

**Exercice 1.**

On considère les tables suivantes :

PILOTE(Matricule, Nom, Ville, Age, Salaire).

VOL(Numvol, DateV, Heure\_départ, Heure\_arrivée, Ville\_départ, Ville\_arrivée)

ESCALE(Numescale, Numvol, Ville\_Escale, Durée\_Escale)

- 1) Écrivez la fonction PL/SQL **sommeSalaire()** qui calcule la somme des salaires des pilotes dont l'âge est entre 30 et 40 ans en utilisant un curseur.
- 2) Écrivez un programme PL/SQL qui affiche la moyenne des salaires des pilotes dont l'âge est entre 30 et 40 ans (utiliser la fonction **sommeSalaire**).
- 3) Ecrivez la procédure **itinéraire(Nvol)** qui permet d'afficher l'itinéraire d'un vol dont le numéro est passé en paramètre.

Exemple : Itinéraire(4) Rabat → Casablanca → Montpellier → Paris

- 4) Ecrivez la fonction **nombreVolsDate(laDate)** qui prend en paramètre une date et **retourne** le nombre de vols qui sont programmés pour cette date.

**Exercice 2.**

Le schéma relationnel suivant correspond à la base de données d'une agence de voyage. Les relations décrivent les circuits touristiques proposés par l'agence. Chaque circuit comporte plusieurs étapes. La base contient aussi les lieux touristiques associés aux étapes des circuits.

**Circuit**( identifiant **INTEGER**, descriptif **VARCHAR**, villeDepart **VARCHAR**, paysDepart **VARCHAR**, villeArrivee **VARCHAR**, paysArrivee **VARCHAR**, dateDepart **DATE**, nbPlaceDisponible **INTEGER**, duree **INTEGER**, prixInscription **INTEGER** ) ;

Un circuit possède un identifiant et un descriptif. Il est réalisé entre une ville de départ et une ville d'arrivée. Il a une durée exprimée en jours et coûte le prix *prixInscription* (prix non-incluant le coût des visites de lieux touristiques).

**Etape**( #identifiant **INTEGER**, ordre **INTEGER**, #numLieu **INTEGER**, dateEtape **DATE**, duree **INTEGER** ) ;

Un circuit correspond à une liste de lieux à visiter (étapes) dans un certain ordre. Chaque étape référence un lieu à visiter. L'étape a lieu à la date *dateEtape* et dure le temps *duree* (en minutes).

**LieuAvisiter**( numLieu **INTEGER**, nomLieu **VARCHAR**, ville **VARCHAR**, pays **VARCHAR**, descriptif **VARCHAR**, prix **INTEGER** ) ;

Un lieu (un site, un monument, un musée) à visiter possède un numéro, un nom, un descriptif et est situé dans une ville. La visite coûte un certain prix (pouvant être égal à 0).

Un client, identifié par un numéro de client et un nom, peut réserver une ou plusieurs places.

**Client**(numcli **INTEGER**, nom **VARCHAR**)

**Reservations**(#numcli **INTEGER**, #identifiant **INTEGER**,  
nombrePlaces **INTEGER**)

30

- 1) Écrire la procédure PL/SQL **infoCircuit(identifiantC)** permettant d'afficher pour le circuit dont le numéro est passé en paramètres à la procédure la ville de départ, la ville d'arrivée ainsi que le nombre d'étapes du circuit. Exemple de message affiché **infoCircuit(9)**:

**Information sur le circuit 9: Kénitra, Tinghir, 15 étapes**

- 2) Écrire la procédure PL/SQL **deleteLieu(numL)** supprimant un lieu s'il n'est pas visité (s'il n'est pas associé à aucune étape d'un circuit). Le numéro du lieu est passé en paramètre à la procédure. Si le lieu ne peut pas être supprimé, alors il faut lancer une exception et afficher un message d'erreur.
- 3) Écrire la procédure PL/SQL **bloqueRes(N)** qui permet de bloquer toute réservation (en mettant le nombre de places disponibles à zéro) pour tous les circuits, dont la date de départ est dans moins de  $N$  jours ( $N$  est donné en paramètre à la procédure). Si des circuits sont mis à jour, afficher un message indiquant le nombre de circuits modifiés, sinon afficher qu'aucune ligne n'est modifiée.

--- Bon courage ---

