

PL/SQL

TP2

1 Aide mémoire PL/SQL

Bloc PL/SQL	<pre> DECLARE -- Déclaration constantes/variables BEGIN -- Commandes/instructions EXCEPTION -- Traitement des erreurs à l'exécution END; / -- Exécution automatique lors du « start » </pre>
Déclaration de variable	<code>Nom_Variable TYPE_VARIABLE;</code>
Affectation	<code>Nom_Variable := valeur;</code>
Tests	<pre> SELECT attribut INTO Nom_Variable FROM table; IF condition1 THEN -- Instructions ELSIF condition2 THEN -- (Optionnel) -- Instructions ELSE -- (Optionnel) -- Instructions END IF; </pre>
Boucles	<pre> FOR compteur IN [REVERSE] min..max LOOP -- Instructions END LOOP; WHILE condition LOOP -- Instructions END LOOP; </pre>
Curseurs - Déclaration - Utilisation	<pre> CURSOR Nom_Curseur IS Requête_SQL; FOR nuplet IN Nom_Curseur LOOP -- Instructions -- Ex. Nom_Variable := nuplet.attribut; END LOOP;-- NB : nuplet est de type Nom_Curseur%ROWTYPE </pre>
Exceptions - Déclarer - Lever - Traiter	<pre> Nom_Exception EXCEPTION; RAISE Nom_Exception; WHEN Nom_Exception THEN -- Instruction; </pre>
Documentation PL/SQL	http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28370/toc.htm

2 Quel cirque !

Soit la table **PERSONNEL**(Nom, Role) qui rassemble les membres du personnel d'un cirque. On souhaite déterminer la proportion de jongleurs parmi eux.

1. Créer et alimenter cette table avec un jeu de données.

2. Ecrire un bloc PL/SQL anonyme permettant de :

- compter le nombre de n-uplets dans la table **PERSONNEL** et stocker le résultat dans une variable ;
- compter le nombre d'employés dont le rôle est « Jongleur » dans la table **PERSONNEL** et stocker le résultat dans une deuxième variable ;
- calculer la proportion (en pourcentage), stocker le résultat dans une troisième variable et afficher le résultat à l'écran.

3. Inclure dans le programme précédent un traitement d'exception permettant de détecter si la table **PERSONNEL** est vide, c'est-à-dire, que le nombre total de n-uplets dans **PERSONNEL** est égal à zéro. Dans ce cas, déclencher une erreur fatale. Tester en effaçant tout le contenu de la table **PERSONNEL**.

3 Catalogue

Ecrire un programme PL/SQL (bloc PL/SQL anonyme) permettant d'afficher votre catalogue système (liste des tables et des vues de votre compte) disponible grâce à la vue système TAB sous la forme :

L'objet UNE_TABLE est de type TABLE.

L'objet UNE_AUTRE_TABLE est de type TABLE.

L'objet UNE_VUE est de type VIEW.

...

NB : Tester au préalable sous SQL l'exécution de la requête `SELECT * FROM TAB;` pour avoir une idée du contenu de la vue système TAB.

4 Les gros navions

Soit le schéma suivant :

AVION (AvNum, AvNom, Capacite, Localisation)

PILOTE (PINum, PINom, PIPrenom, Ville, Salaire)

VOL (VolNum, PInum#, AvNum#, VilleDep, VilleArr, HeureDep, HeureArr)

Les performances des avions de marque Airbus évoluent, aussi souhaite-t-on faire des mises à jour de la table VOL. Les temps de vol des avions de type A300 (avions n° 1 et 4) doivent être réduits de 10 % et ceux des avions de type A310 (avions n° 2 et 8) de 15 %. Il s'agit de définir un programme PL/SQL permettant ces modifications.

1. Créer les tables présentées ci-dessus et alimentez-les avec un jeu de données cohérent.
2. Dans un bloc PL/SQL anonyme, déclarer un curseur permettant de lire les données suivantes : numéro de vol, numéro d'avion, heure de départ et heure d'arrivée des vols pour lesquels l'avion utilisé est le n° 1, 2, 4 ou 8. Pour chaque vol lu par le curseur, calculer le temps de vol, le réduire dans la proportion voulue selon l'avion utilisé, puis mettre à jour l'attribut HEUREARR de ce vol dans la table VOL.
3. Tester !