

TP 2 PL/SQL avec Oracle LiveSQL

Gestion de l'agence « Escale Horizon » –

Pré-requis : les tables DESTINATIONS, CLIENTS et RESERVATIONS du TP SQL « Escale Horizon » doivent déjà exister dans votre schéma Oracle LiveSQL.

Objectifs du TP

- Écrire des blocs anonymes PL/SQL.
- Déclarer et utiliser des variables.
- Utiliser DBMS_OUTPUT.PUT_LINE pour afficher des résultats.
- Manipuler des structures conditionnelles.
- Utiliser des curseurs implicites (SELECT ... INTO).
- Créer des fonctions stockées.
- Créer des procédures stockées.
- Créer des procédures qui utilisent des fonctions.
- Utiliser des curseurs explicites.
- Parcourir un curseur explicite de deux façons : avec FETCH et avec une boucle FOR.

Consignes générales

- Créer un nouveau script dans Oracle LiveSQL.
- Activer l'affichage des sorties avec : SET SERVEROUTPUT ON;
- Tester chaque bloc, fonction ou procédure au fur et à mesure.
- Pour les exercices sur curseurs, réaliser obligatoirement deux versions :
 - une version avec OPEN / FETCH / EXIT WHEN / CLOSE ;
 - une version avec FOR ... IN curseur.

Partie 1 – Prise en main

Exercice 1 – Premier bloc anonyme

Écrire un bloc anonyme qui affiche :

- le message : Début du TP PL/SQL
- la date du jour

Partie 2 – Blocs anonymes

Exercice 2 – Variables simples

- Déclarer une variable v_ville.
- Lui affecter la valeur 'Tokyo'.
- Afficher cette valeur.

Exercice 3 – Calcul simple

- Déclarer une variable v_prix.
- Lui affecter la valeur 120.
- Calculer un prix majoré de 15%.
- Afficher le prix initial et le prix final.

Exercice 4 – Condition

- Prendre un prix de destination.
- Afficher :
- Destination économique si prix < 100
- Destination standard si prix entre 100 et 200
- Destination premium si prix > 200

Exercice 5 – Mise à jour simple

- Modifier le statut de la réservation 1003.
- Remplacer 'En attente' par 'Confirmé'.
- Afficher le nombre de lignes modifiées avec SQL%ROWCOUNT.

Exercice 6 – Suppression contrôlée

- Supprimer les réservations annulées.
- Afficher le nombre de lignes supprimées.

Exercice 7 – Insertion conditionnelle

- Tenter d'insérer une nouvelle destination.
- Vérifier que le prix est strictement supérieur à 0 avant insertion.
- Afficher un message de succès ou d'échec.

Partie 3 – Curseurs implicites (SELECT ... INTO)

Exercice 8 – Récupérer le prix d'une destination

- Récupérer dans une variable le prix de base de la destination 'Paris'.
- Afficher ce prix.

Exercice 9 – Récupérer les informations d'un client

- Récupérer le nom et l'email du client d'identifiant 10.
- Afficher ces informations.

Exercice 10 – Gestion d'erreur simple

- Chercher une destination d'identifiant 999.
- Afficher un message si elle n'existe pas (exception NO_DATA_FOUND).

Partie 4 – Fonctions stockées

Exercice 11 – Fonction simple

- Créer une fonction `prix_ttc(p_prix NUMBER)`.
- Elle reçoit un prix HT.
- Elle retourne le prix TTC avec une majoration de 20%.
- Tester la fonction dans un bloc anonyme.

Exercice 12 – Fonction métier

- Créer une fonction `nb_reservations_client(p_id_client NUMBER)`.
- Elle retourne le nombre de réservations du client passé en paramètre.
- Tester la fonction.

Exercice 13 – Fonction de classification

- Créer une fonction `categorie_destination(p_prix NUMBER)`.
- Retourner 'Economique' si `prix < 100`.
- Retourner 'Standard' si `prix` entre 100 et 200.
- Retourner 'Premium' sinon.

Partie 5 – Procédures stockées

Exercice 14 – Procédure d’affichage

- Créer une procédure `afficher_client(p_id_client NUMBER)`.
- Elle affiche le nom, le prénom et l’email du client.

Exercice 15 – Procédure avec paramètres

- Créer une procédure `augmenter_prix_region(p_region, p_pourcentage)`.
- Elle augmente les prix des destinations de la région passée en paramètre.
- Afficher le nombre de lignes modifiées.

Exercice 16 – Procédure qui utilise une fonction

- Créer une procédure `afficher_categorie_destination(p_id_dest NUMBER)`.
- Elle récupère le prix de la destination.
- Elle appelle la fonction `categorie_destination`.
- Elle affiche la catégorie de la destination.

Partie 6 – Curseurs explicites avec FETCH

Exercice 17 – Parcourir toutes les destinations (version FETCH)

- Créer un curseur explicite qui récupère : `ville, pays, prix_base`.
- Parcourir le curseur avec `OPEN / FETCH / EXIT WHEN / CLOSE`.
- Afficher les informations de chaque destination.

Exercice 18 – Destinations d'Europe (version FETCH)

- Créer un curseur explicite qui récupère uniquement les destinations de la région 'Europe'.
- Parcourir le curseur avec FETCH.
- Afficher la ville et le prix.

Partie 7 – Curseurs explicites avec boucle FOR

Exercice 19 – Parcourir toutes les destinations (version FOR)

- Reprendre l'exercice 17 en utilisant une boucle FOR ... IN curseur.

Exercice 20 – Réservations confirmées (version FOR)

- Créer un curseur qui récupère les réservations confirmées.
- Afficher l'identifiant de réservation, la date de départ et le statut avec une boucle FOR.

Partie 8 – Exercice final de synthèse

Exercice 21 – Procédure bilan_client

Créer une procédure bilan_client(p_id_client NUMBER) qui :

- affiche le nom du client ;
- affiche son nombre de réservations en utilisant la fonction nb_reservations_client ;
- affiche ensuite la liste de ses réservations (identifiant, ville, date de départ, statut) à l'aide d'un curseur explicite.