TP 1 : Programmation PL/SQL

L.ZIAD

EST Essaouira

Exercice 1

Ecrivez un programme affectant les valeurs 1 et 2 a deux variables a et b, puis permutant les valeurs de ces deux variables.

```
DECLARE
a NUMBER := 1;
b NUMBER := 2;
temp NUMBER;
BEGIN
DBMS_OUTPUT.PUT_LINE('Avant permutation: a = ' || a || ', b = ' || b);

-- Permutation des valeurs
temp := a;
a := b;
b := temp;

DBMS_OUTPUT.PUT_LINE('Apres permutation: a = ' || a || ', b = ' || b);
END;
/
```

Listing 1 – Permutation de variables

Exercice 2

Ecrivez un programme plaçant la valeur 10 dans une variable a, puis affichant la factorielle de a.

```
DECLARE
a NUMBER := 10;
factorielle NUMBER := 1;
i NUMBER;
BEGIN
FOR i IN 1..a LOOP
factorielle := factorielle * i;
END LOOP;
```

```

DBMS_OUTPUT.PUT_LINE('La factorielle de ' || a || ' est: ' || factorielle);
END;
/
```

Listing 2 – Calcul de factorielle

Exercice 3

Ecrivez un programme PL/SQL permettant d'ajouter un enregistrement dans la table DEPARTMENTS, ayant comme id la plus grande valeur des id + 10.

```
DECLARE
max_id NUMBER;
BEGIN
-- Recuperer le plus grand ID
SELECT MAX(department_id) INTO max_id FROM departments;

-- Inserer un nouveau departement avec ID = max_id + 10
INSERT INTO departments (department_id, department_name, manager_id, location_id)
VALUES (max_id + 10, 'Nouveau Departement', NULL, NULL);

COMMIT;
DBMS_OUTPUT.PUT_LINE('Nouveau departement cree avec ID: ' || (max_id + 10));
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Erreur: ' || SQLERRM);
ROLLBACK;
END;
/
```

Listing 3 – Insertion dans DEPARTMENTS

Exercice 4

Creez un bloc PL/SQL qui selectionne dans la table DEPARTMENTS le plus grand numero de service, et stocke ce dernier dans une variable. Affichez le resultat a l'ecran.

```
DECLARE
max_dept_id NUMBER;
BEGIN
SELECT MAX(department_id) INTO max_dept_id FROM departments;

DBMS_OUTPUT.PUT_LINE('Le plus grand numero de departement est: ' || max_dept_id);
```

```
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Aucun departement trouve');
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Erreur: ' || SQLERRM);
END;
/
```

Listing 4 – Recherche du plus grand numero de departement

Exercice 5

Affichez le nouveau departement que vous avez cree.

```
DECLARE
v_department_id departments.department_id%TYPE;
v_department_name departments.department_name%TYPE;
BEGIN
-- Recuperer le plus grand ID (le dernier cree)
SELECT MAX(department_id) INTO v_department_id FROM departments;

-- Recuperer les informations du departement
SELECT department_name INTO v_department_name
FROM departments
WHERE department_id = v_department_id;

DBMS_OUTPUT.PUT_LINE('Departement cree:');
DBMS_OUTPUT.PUT_LINE('ID: ' || v_department_id);
DBMS_OUTPUT.PUT_LINE('Nom: ' || v_department_name);
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Departement non trouve');
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Erreur: ' || SQLERRM);
END;
/
```

Listing 5 – Affichage du nouveau departement

Exercice 6

Creez un bloc PL/SQL qui met a jour l'ID d'emplacement du nouveau service ajoute au cours de l'exercice precedent (location_id=2500). Affichez le nombre de lignes affectees.

```
DECLARE
v_department_id departments.department_id%TYPE;
rows_affected NUMBER;
BEGIN
```

```
-- Recuperer le plus grand ID
SELECT MAX(department_id) INTO v_department_id FROM departments;

-- Mettre a jour l'emplacement
UPDATE departments
SET location_id = 2500
WHERE department_id = v_department_id;

rows_affected := SQL%ROWCOUNT;

COMMIT;
DBMS_OUTPUT.PUT_LINE('Nombre de lignes affectees: ' || rows_affected);
DBMS_OUTPUT.PUT_LINE('Emplacement mis a jour pour le departement ID: ' || v_department_id);
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Erreur: ' || SQLERRM);
ROLLBACK;
END;
/
```

Listing 6 – Mise a jour de l'emplacement

Exercice 7

Ecrivez le bloc PL/SQL qui saisit (demande a l'utilisateur) un last_name d'un employe, et qui retourne un message indiquant le manager de cet employe.

```
DECLARE
v_last_name employees.last_name%TYPE := '&Entrez_le_nom_de_famille';
v_manager_name VARCHAR2(100);
BEGIN
-- Recuperer le nom du manager
SELECT m.first_name || ' ' || m.last_name INTO v_manager_name
FROM employees e
JOIN employees m ON e.manager_id = m.employee_id
WHERE e.last_name = v_last_name;

DBMS_OUTPUT.PUT_LINE('Le manager de ' || v_last_name || ' est: ' || v_manager_name);
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Aucun employe trouve avec le nom: ' || v_last_name);
WHEN TOO_MANY_ROWS THEN
```

```
DBMS_OUTPUT.PUT_LINE('Plusieurs employes trouves avec ce nom');
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Erreur: ' || SQLERRM);
END;
/
```

Listing 7 – Recherche du manager d'un employe

Exercice 8

Ecrivez le bloc PL/SQL qui affiche les 10 derniers employes embauches.

```
DECLARE
CURSOR c_employees IS
SELECT first_name, last_name, hire_date
FROM employees
ORDER BY hire_date DESC;

v_counter NUMBER := 0;
BEGIN
DBMS_OUTPUT.PUT_LINE('Les 10 derniers employes embauches:');
DBMS_OUTPUT.PUT_LINE('-----------------------------------');

FOR emp IN c_employees LOOP
EXIT WHEN v_counter >= 10;

DBMS_OUTPUT.PUT_LINE(
'Nom: ' || emp.first_name || ' ' || emp.last_name ||
' - Date d''embauche: ' || TO_CHAR(emp.hire_date, 'DD/MM/YYYY')
);

v_counter := v_counter + 1;
END LOOP;

IF v_counter = 0 THEN
DBMS_OUTPUT.PUT_LINE('Aucun employe trouve');
END IF;
END;
/
```

Listing 8 – Affichage des 10 derniers employes embauches

Version alternative avec ROWNUM pour l'Exercice 8

```
DECLARE
CURSOR c_recent_employees IS
SELECT first_name, last_name, hire_date
FROM (
SELECT first_name, last_name, hire_date
FROM employees
ORDER BY hire_date DESC
)
WHERE ROWNUM <= 10;
BEGIN
DBMS_OUTPUT.PUT_LINE('Les 10 derniers employes embauches:');
DBMS_OUTPUT.PUT_LINE('-----------------------------------');

FOR emp IN c_recent_employees LOOP
DBMS_OUTPUT.PUT_LINE(
'Nom: ' || emp.first_name || ' ' || emp.last_name ||
' - Date d''embauche: ' || TO_CHAR(emp.hire_date, 'DD/MM/YYYY')
);
END LOOP;
END;
/
```

Listing 9 – Version alternative avec ROWNUM