

Série d'exercices PL/SQL



hamdi.aloulou@isima.u-monastir.tn



Exercice 1

Soit la relation:

EMPLOYE (ID, NOM, DEPARTEMENT, AGE, SALAIRE).

Écrivez un bloc PLSQL qui effectue une augmentation de 200 euros du salaire des employés du département 'Commercial' et qui utilise le dernier curseur implicite pour afficher le nombre d'employés affectés par ce changement.



Solution

- › Les curseurs implicites sont créés par PLSQL lors de l'exécution des commandes SQL INSERT, UPDATE, DELETE, SELECT (renvoi une seule ligne).
- › Les attributs du dernier curseur implicite utilisé sont accessibles par le préfixe SQL : SQL%FOUND, SQL%NOTFOUND, SQL%ROWCOUNT



Solution

```
SET SERVEROUTPUT ON  
DECLARE
```

```
    total NUMBER(2);
```

```
BEGIN
```

```
    UPDATE EMPLOYE SET salaire = salaire + 200
```

```
    WHERE DEPARTEMENT = 'Commercial';
```

```
    IF SQL%NOTFOUND THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Aucun salaire augmenté');
```

```
    ELSIF SQL%FOUND THEN
```

```
        total := SQL%ROWCOUNT;
```

```
        DBMS_OUTPUT.PUT_LINE(total || 'salaires ont été augmentés');
```

```
    ENDIF;
```

```
END;
```



Exercice 2

Soit la relation EMPLOYE

EMPLOYE(ID, NOM, DEPARTEMENT, AGE, SALAIRE).

Écrivez un bloc PL/SQL qui affiche les noms des employés du département 'Commercial' qui sont âgés de plus de 40 ans. Utilisez un curseur explicite déclaré dans une boucle FOR dédiée.



Solution

SET SERVEROUTPUT ON

BEGIN

FOR rec_emp IN (SELECT * FROM EMPLOYE

WHERE AGE >= 40 AND DEPARTEMENT = 'Commercial') LOOP

DBMS_OUTPUT.PUT_LINE(rec_emp.NOM);

END LOOP;

END;



Exercice 3

Soit la table E_EMP :

E_EMP(No, Nom, Salaire, Service)

1. Ecrivez un programme PL/SQL qui permet d'afficher les enregistrements (champs No, Nom et Salaire) de la table E_EMP qui ont un salaire supérieur à la moyenne des salaires des employés
2. Ecrivez un programme PL/SQL qui met à jour les salaires des employés comme suit:
 - ☐ Salaire = 1000 si service =1
 - ☐ Salaire = 2000 si service =2
 - ☐ Salaire = 3000 sinon



Solution

DECLARE

```
v_mean E_EMP.SALAIRE%TYPE;
```

```
CURSOR c(p_mean E_EMP.SALAIRE%TYPE) IS
```

```
SELECT No, Nom, Salaire FROM E_EMP WHERE SALAIRE > p_mean;
```

BEGIN

```
SELECT AVG(SALAIRE) INTO v_mean FROM E_EMP;
```

```
DBMS_OUTPUT.PUT_LINE('La moyenne des salaires est : ' || v_mean);
```

```
FOR rec_c IN c(v_mean) LOOP
```

```
    DBMS_OUTPUT.PUT_LINE('NO: ' || rec_c.NO || ' NOM: ' || rec_c.NOM || ' SALAIRE: ' ||  
        rec_c.SALAIRE);
```

```
END LOOP;
```

```
END;
```



Solution

DECLARE

CURSOR c **IS SELECT** * **FROM** E_EMP **FOR UPDATE OF** SALAIRE

BEGIN

FOR rec_c **IN** c **LOOP**

IF rec_c.SERVICE_NO = 1 **THEN**

UPDATE E_EMP **SET** SALAIRE = 1000 **WHERE CURRENT OF** c;

ELSIF rec_c.SERVICE_NO = 2 **THEN**

UPDATE E_EMP **SET** SALAIRE = 2000 **WHERE CURRENT OF** c;

ELSE

UPDATE E_EMP **SET** SALAIRE = 3000 **WHERE CURRENT OF** c;

END IF;

END LOOP;

COMMIT;

END;



Exercice 4

- › On considère la base de données suivante :
 - COMPETITION (CODE_COMP, NOM_COMP)
 - PARTICIPANT (NO_PART, NOM_PART, DATENAISSANCE, ADRESSE, EMAIL)
 - SCORE (NO_PART#, CODE_COMP#, NO_JUGE#, NOTE)
 - JUGE (NO_JUGE, NON_JUGE, NATIONALITÉ)
- › Écrire un bloc PLSQL qui lit à la console le nom d'une compétition et qui affiche les participants avec leur score total (la somme de tous les scores par tous les juges). Utilisez un curseur avec paramètre.



Solution

SET SERVEROUTPUT ON

ACCEPT nom_c varchar(20) **PROMPT** 'Nom de la compétition'

DECLARE

CURSOR c1(p_nom_c COMPETITION.NOM_COMP%TYPE) **IS**

SELECT DISTINCT P.No_part, P.Nom_part

FROM Participant P, Competition C, Score S

WHERE S.No_part = P.No_part **AND** S.Code_Comp = C.Code_Comp **AND**
C.Nom_Comp = p_nom_comp;

V_sum Score.Note%TYPE;

BEGIN

FOR rec_c1 **IN** c1(&nom_c) **LOOP**

SELECT SUM(NOTE) **INTO** V_sum **FROM** Score

WHERE NO_PART=rec_c1.No_part and CODE_COMP= &nom_c;

DBMS_OUTPUT.PUT_LINE(rec_c1.Nom_part || ' ' || V_sum);

END LOOP;

END;



Solution 2

SET SERVEROUTPUT ON

ACCEPT nom_c varchar(20) **PROMPT** 'Nom de la compétition'

DECLARE

CURSOR c1(p_nom_c COMPETITION.NOM_COMP%TYPE) **IS**

SELECT P.Nom_part , **SUM**(S.Note) AS TOTAL

FROM Participant P, Competition C, Score S

WHERE S.No_part = P.No_part **AND** S.Code_Comp = C.Code_Comp **AND**
C.Nom_Comp = p_nom_comp

Group By P.Nom_part;

BEGIN

FOR rec_c1 **IN** c1(&nom_c) **LOOP**

DBMS_OUTPUT.PUT_LINE(rec_c1.Nom_part || ' ' || rec_c1.total);

END LOOP;

END;