

Développement des Bases de Données

Arquillière Mathieu

3 février 2020

1 TP1

1.1 Exercice 1

Code :

```
1 DECLARE
2     nom emp.ename%TYPE;
3     salaire emp.sal%TYPE;
4     commission emp.comm%TYPE;
5     departement dept.dname%TYPE;
6 BEGIN
7     SELECT ename,sal,comm,dname INTO nom,salaire,commission,departement FROM Emp NATURAL JOIN Dept
8     WHERE ename='MILLER';
9     DBMS_OUTPUT.PUT_LINE('Nom : ' || nom || ' Salaire : ' || salaire || ' Commission : ' ||
10    commission || 'Departement : ' || departement);
11 END;
12 /
```

Résultat :

1.2 Exercice 2

Code :

```
1 DECLARE
2     num1 temp.num_col1%TYPE;
3     num2 temp.num_col2%TYPE;
4     char temp.char_col%TYPE;
5 BEGIN
6     FOR i IN 1..10 LOOP
7         IF MOD(i, 2) = 0 THEN
8             INSERT INTO temp VALUES (i, i * 100, CONCAT(TO_CHAR(i), ' est pair'));
9         ELSE
10            INSERT INTO temp VALUES (i, i * 100, CONCAT(TO_CHAR(i), ' est impair'));
11        END IF;
12    END LOOP;
13    COMMIT;
14 END;
15 /
```

Résultat :

1.3 Exercice 3

Code :

```
1 DECLARE
2     Cursor c IS SELECT sal, empno, ename FROM emp ORDER BY sal DESC;
3     salaire emp.sal%TYPE;
4     numero emp.empno%TYPE;
5     nom emp.ename%TYPE;
6 BEGIN
7     OPEN c;
8     FOR i IN 1..5 LOOP
9         FETCH c INTO salaire, numero, nom;
10        INSERT INTO temp VALUES (salaire, numero, nom);
11    END LOOP;
12 END;
13 /
```

Résultat :

1.4 Exercice 4

Code :

```
1 DECLARE
2   Cursor c IS SELECT UNIQUE sal, NVL(comm, 0), empno, ename FROM emp WHERE sal + NVL(comm, 0) >
3     2000;
4   salaire emp.sal%TYPE;
5   numero emp.empno%TYPE;
6   nom emp.ename%TYPE;
7   comm emp.comm%TYPE;
8 BEGIN
9   OPEN c;
10  LOOP
11    FETCH c INTO salaire, comm, numero, nom;
12    INSERT INTO temp VALUES (salaire + comm, numero, nom);
13    EXIT WHEN (c%notfound);
14  END LOOP;
15 END;
16 /
```

Résultat :

1.5 Exercice 5

Code :

```
1 DECLARE
2   Cursor c IS SELECT sal, ename, empno, mgr FROM emp;
3   salaire emp.sal%TYPE;
4   nom emp.ename%TYPE;
5   empno emp.empno%TYPE;
6   mgrEmp emp.mgr%TYPE;
7   chaineMgr emp.mgr%TYPE;
8 BEGIN
9   OPEN c;
10  LOOP
11    FETCH c INTO salaire, nom, empno, mgrEmp;
12    EXIT WHEN(c%NOTFOUND);
13    IF salaire >= 4000 THEN
14      DBMS_OUTPUT.PUT_LINE('Salaire: ' || salaire || ' Nom: ' || nom || ' no: ' || empno || ' mgr:
15      ' || mgrEmp);
16      SELECT mgr INTO chaineMgr FROM emp WHERE empno=7902;
17      LOOP
18        EXIT WHEN(chaineMgr IS NULL OR chaineMgr=mgrEmp);
19        SELECT mgr INTO chaineMgr FROM emp where empno=chaineMgr;
20      END LOOP;
21      IF chaineMgr=mgrEmp OR mgrEmp IS NULL THEN
22        INSERT INTO temp VALUES (null, salaire, nom);
23      END IF;
24    END LOOP;
25 END;
26 /
```

Résultat :

2 TP2

2.1 Exercice 1

Code :

```
1 CREATE OR REPLACE PROCEDURE createdept_zangla(num IN NUMBER, name IN VARCHAR2, loc IN VARCHAR2)
2 IS
3     d NUMBER;
4 BEGIN
5     SELECT deptno INTO d FROM dept WHERE deptno = num;
6     RAISE_APPLICATION_ERROR(-20001, 'Numero de departement deja existant');
7     EXCEPTION
8         WHEN NO_DATA_FOUND THEN
9             INSERT INTO dept VALUES(num, name, loc);
10 END;
11 /
12
13 /*exec createdept_zangla(12, 'TEst', 'Aubiere');*/
14
15 CREATE OR REPLACE FUNCTION salok_zangla(jobselect IN VARCHAR2, salaire IN NUMBER) RETURN NUMBER
16 IS
17     j VARCHAR2(9);
18     mi NUMBER;
19     ma NUMBER;
20 BEGIN
21     SELECT job, lsal, hsal INTO j, mi, ma FROM salintervalle_f2 WHERE mi <= salaire AND ma >=
22     salaire;
23     RETURN 1;
24     EXCEPTION WHEN NO_DATA_FOUND THEN RETURN 0;
25 END;
26 /
27 select salok_zangla(job, 2800) FROM salintervalle_f2;
```

Résultat :