

# TP

M1 MIAGE - 2015/2016

January 27, 2016

## Exemple1:

```
create or replace procedure acrostiche(mot varchar2,languechoisie varchar2) is
lettre varchar2(2);
renvoi varchar2(20);
longueur integer;
randomnumber integer;
begin

for i in 1 .. length(mot)
loop
    lettre := SUBSTR(mot, i,1);
    select count(*) into longueur
    from adjectifs where langue=languechoisie and libelle like concat(lettre,'%') ;
    randomnumber := SYS.DBMS_RANDOM.VALUE(1,longueur);
    select libelle into renvoi from( select libelle ,rownum rn
    from (select libelle from adjectifs
    where langue=languechoisie and libelle like concat(lettre,'%'))
    where rn =randomnumber;
    DBMS_OUTPUT.put_line(renvoi );
end loop;
end;
```

## Exemple2:

```
create or replace PROCEDURE Acro (pLibelle IN VARCHAR2, pLangue IN VARCHAR2) AS

mots SYS_REFCURSOR;
TYPE lMot IS TABLE OF VARCHAR2(50) INDEX BY BINARY_INTEGER;
lResult lMot;
lAnagram lMot;
vLibelle adjectifs.libelle%Type;
identif number(10);
valeurTab adjectifs.libelle%Type;
```

```

BEGIN

    identif := 0;

    FOR i in 1 .. length(pLibelle) LOOP
        lAnagram(i) := SUBSTR(pLibelle, i, 1);
    END LOOP;

    FOR j in lAnagram.FIRST .. lAnagram.LAST LOOP
        OPEN mots FOR SELECT libelle
        from adjectifs where langue = pLangue and libelle like lAnagram(j)||'%';
        FETCH mots into valeurTab;
        lResult(identif) := valeurTab;
        identif := identif + 1;
        Close mots;
    end loop;

    FOR k in lResult.FIRST .. lResult.LAST Loop
        DBMS_OUTPUT.PUT_LINE(lResult(k));
    end loop;

```

END;

### Exemple3:

```

CREATE OR REPLACE FUNCTION genere_mot(lettre IN VARCHAR2)
    RETURN VARCHAR2
AS
    TYPE char_tab IS TABLE OF VARCHAR2(1) INDEX BY BINARY_INTEGER;
    expression VARCHAR2(30) := CONCAT('~',lettre);
    CURSOR c_mots IS SELECT libelle FROM adjectifs
    WHERE REGEXP_LIKE (libelle, expression) AND LANGUE = 'français'
    ORDER BY DBMS_RANDOM.RANDOM;
    mot c_mots%ROWTYPE;
BEGIN
    OPEN c_mots;
    FETCH c_mots INTO mot;
    RETURN mot.libelle;
END genere_mot;
/

```

```

-- Affiche dans la sortie l'acrostiche du mot passé en paramètre
CREATE OR REPLACE PROCEDURE acrostiche(chaine IN VARCHAR2)
AS
    TYPE char_tab IS TABLE OF VARCHAR2(1) INDEX BY BINARY_INTEGER;
    tab char_tab;

```

```

    tmp VARCHAR2(1);
    resultat VARCHAR2(100) := '';
BEGIN
    FOR i IN 1 .. LENGTH(chaine) LOOP
        DBMS_OUTPUT.PUT_LINE(genere_mot(substr(chaine, i, 1)));
    END LOOP;
END acrotische;

```

#### Exemple4:

```

procedure acrostiche (mot adjectifs.libelle%type, lang adjectifs.langue%type) as
    current_mot varchar2(50);
    CURSOR mots (car IN VARCHAR2) IS
    SELECT libelle FROM
    (SELECT libelle FROM adjectifs
     WHERE LIBELLE LIKE car || '%' and langue = lang ORDER BY dbms_random.value)
    WHERE rownum = 1;
    begin
        for i in 1..length(mot) loop
            OPEN mots(substr(mot,i,1));
            LOOP
                FETCH mots INTO current_mot;
                EXIT WHEN mots%NOTFOUND;
            END LOOP;
            DBMS_OUTPUT.put_line(current_mot);
            CLOSE mots;
        end loop;
    end acrostiche;

```

**Question 1** Peut-on comparer ces différentes propositions d'extraire d'un acrostiche ?

Peut-on estimer leur coût (à la manière de l'estimation des coûts d'un plan d'évaluation par le CBO).

Y-a-t-il un plus efficace que les autres ?

**Question 2** Mesurer les temps d'exécution de votre solution d'acrostiche en faisant varier la longueur du mot (utiliser la fonction GET\_TIME du package DBMS\_UTILITY).

```

DECLARE
    v_start NUMBER;
    v_stop NUMBER;
BEGIN
    v_start := DBMS_UTILITY.GET_TIME;
    v_stop := DBMS_UTILITY.GET_TIME;
    DBMS_OUTPUT.PUT_LINE(((v_stop-v_start)*10) || ' ms');
END;

```

Comparer le temps d'exécution de :

```
select min(mot) from adjectifs ;
```

```
select max(mot) from adjectifs suivi de select min(mot) from adjectifs ;
```

```
select min(mot), max(mot) from adjectifs
```

**Question 3** Produire une autre version de la procédure ACROSTICHE qui s'appuie sur au plus **un** parcours de la table **Adjectifs**.