

# Acrostiche 2.0

M1 MIAGE - 2015/2016

February 3, 2016

## 1 Coût

$$\text{cost} = (\#SRDS * \text{sreadtm} + \#MRDS * \text{mreadtm} + \#CPUCycles / \text{cpuspeed}) / \text{sreadtm}$$

où

#SRDS est le nombre des lectures E/S en mode mono bloc,

sreadtm est le temps nécessaire à la lecture d'un bloc

#MRDS est le nombre des lectures en mode multiblocs,

mreadtm est le temps de lecture en mode multiblocs

CPUCycles est le nombre de cycles CPU,

cpuspeed est le nombre de cycle CPU par seconde

Par ailleurs

$$\text{sreadtm} = \text{ioseektm} + \text{db\_block\_size} / \text{iotfrspeed}$$

$$\text{mreadtm} = \text{ioseektm} + \text{MBRC} * \text{db\_block\_size} / \text{iotfrspeed}$$

$$\text{cpuspeed} = \text{cpuspeedNW}$$

où

ioseektime : temps de recherche d'un bloc sur le disque en ms, valeur par défaut

10

iotfrspeed : la vitesse de transfert de données en Bytes par ms, valeur par défaut

4096

cpuspeedNW : le nombre moyen de cycles CPU par seconde, valeur par défaut qui dépend du système.

MBRC : le nombre des blocs lus en moyenne dans une opération E/S en mode multiblocs

Les valeurs de ces statistiques sont ensuite utilisées pour calculer les composants nécessaires à l'équation de coût.

## 2 Acrotische “Optimisé”

Quelques exemples:

```
CREATE OR REPLACE PROCEDURE
```

```
ACROSTICHE2 (mot adjectifs.libelle%type, lang adjectifs.langue%type) AS
```

```
i NUMBER;
```

```

j NUMBER;

TYPE papapouik IS TABLE OF VARCHAR2(500) INDEX BY BINARY_INTEGER;
my_table PAPAPOUIK;

BEGIN
FOR libelle_record IN (SELECT libelle FROM adjectifs WHERE adjectifs.langue = lang)
LOOP
    i := instr(mot, SUBSTR(libelle_record.libelle, 1 ,1));
    IF i <> 0 THEN
        my_table(i) := libelle_record.libelle;
    END IF;
END LOOP;

FOR i IN my_table.first..my_table.last LOOP
DBMS_OUTPUT.put_line(my_table(i));
END LOOP;
END ACROSTICHE2;

CREATE OR REPLACE PROCEDURE ACRO2(chaine IN VARCHAR2) AS
CURSOR c_mots IS SELECT libelle FROM adjectifs ORDER BY DBMS_RANDOM.RANDOM;
TYPE char_tab IS TABLE OF VARCHAR2(50) INDEX BY BINARY_INTEGER;
TYPE check_tab IS TABLE OF NUMBER INDEX BY BINARY_INTEGER;
tab char_tab;
tabCheck check_tab;
mot c_mots%ROWTYPE;
BEGIN
    for k in 1..length(chaine)loop
        tabCheck(k) := 0;
    end loop;
    FOR mot IN c_mots LOOP
        for i in 1.. length(chaine) LOOP
            if SUBSTR(mot.libelle,1,1) = SUBSTR(chaine,i,1) and tabCheck(i) = 0
            then tabCheck(i) := 1;
            tab(i) := mot.libelle;
            end if;
        end loop;
    END LOOP;
    for j in 1..length(chaine)loop
        DBMS_OUTPUT.PUT_LINE(tab(j));
    end loop;
END ACRO2;

```

**Question 1** Compléter votre schéma relationnel de façon à pouvoir stocker en base des acrostiches d'une locution donnée, on considère que plusieurs acrostiches peuvent être associés à une même locution (éventuellement dans différentes

languages).

Définir les contraintes de clés primaires et étrangères qui s'imposent dans la conception de votre modèle de données<sup>1</sup>.

Constater que des index ont été créés.

Retrouver les caractéristiques de ces index.

Donner quelques exemples de requêtes où il fait usage de ces index.

**Question 2** Réécrire la procédure **Acrostiche** en fonction de sorte qu'elle génère un acrostiche et le mémorise en base s'il n'existe pas d'acrostiche pour la locution passée en paramètre, et qu'elle renvoie dans tous les cas l'identifiant d'un acrostiche de la locution passée en paramètre.

Analyser le plan d'évaluation qui renvoie la composition d'un acrostiche à partir d'un identifiant d'acrostiche.

---

<sup>1</sup>utiliser la commande ALTER TABLE