TP2 ENDO

Nom : ACHACHI

Pénom : El hadj Ali

Mat : 181832008538

Sujet : View Materialized

1) Créer une vue matérialisée VM1 avec les options (IMMEDIATE, COMPLETE, ON DEMAND) :

```
SQL> CREATE MATERIALIZED VIEW VM1
  2  BUILD IMMEDIATE
  3  REFRESH COMPLETE
  4  ON DEMAND
  5  AS
  6  SELECT * FROM appel
  7  WHERE DATEAPP between '01/06/21' and '30/06/21';

Vue matÚrialisÚe crÚÚe.

SQL>
```

2) Créer une vue matérialisée VM2 avec les options (IMMEDIATE, FAST, ON DEMAND) :
(Premièrement on créer Un fichier de journalisation des données modifiées LOG)

```
SQL> CREATE MATERIALIZED VIEW LOG ON appel;

Journal de vue matÚrialisÚe crÚÚ.

SQL> CREATE MATERIALIZED VIEW VM2
  2  BUILD IMMEDIATE
  3  REFRESH FAST
  4  ON DEMAND
  5  AS
  6  SELECT * FROM appel
  7  WHERE DATEAPP between '01/06/2021' and '30/06/2021' ;

Vue matÚrialisÚe crÚÚe.
```

3) Testez les répercussions des mises à jour de la base de données, sur les deux vues : 1- La suppression

```
SQL> DELETE FROM appel
  2  WHERE codeappel between 500 and 510;

11 ligne(s) supprimÚe(s).

EcoulÚ : 00 :00 :01.15
SQL>
SQL> EXECUTE DBMS_MVIEW.REFRESH('VM1');

ProcÚdure PL/SQL terminÚe avec succPs.

EcoulÚ : 00 :01 :18.09
SQL> EXECUTE DBMS_MVIEW.REFRESH('VM2');

ProcÚdure PL/SQL terminÚe avec succPs.

EcoulÚ : 00 :00 :07.77
```

2- L’ajout :

:

```
SQL> set timing on;
SQL>
SQL> UPDATE appel
  2  SET duree = duree+ 77
  3  WHERE codeappel between 600 and 650;

51 ligne(s) mise(s) Ó jour.

EcoulÚ : 00 :00 :00.01
SQL> EXECUTE DBMS_MVIEW.REFRESH('VM1');

ProcÚdure PL/SQL terminÚe avec succPs.

EcoulÚ : 00 :00 :38.17
SQL> EXECUTE DBMS_MVIEW.REFRESH('VM2');

ProcÚdure PL/SQL terminÚe avec succPs.

EcoulÚ : 00 :00 :10.49
```

On remarque:

Que le temps d'exécution du rafraichissement du vue d'option FAST est toujours inférieur que le temps d'exécution du rafraichissement du vue d'option COMPLETE

5) R1 : la liste des clients (CodeCl, NomCl) ayant effectué des appels de type international :

```
 NUMCLIENT NOMCLIENT
---------- ------------------------------
     51242 FZILQQVL
    183703 CPFQVJIC
    771775 RTJSVWKN
    580907 LECZKGMF
    279843 GVTSDIWV
    419132 MXQRWLPM
   1005655 AYPHBVIT
```

Jusqu'à 1894542 client.

6) <u>le temps et le plan d'exécution du R1</u> :

```
2368641 ligne(s) sÚlectionnÚe(s).

EcoulÚ : 00 :13 :14.34

Plan d'exÚcution
----------------------------------------------------------
Plan hash value: 881011081

-------------------------------------------------------------------------------
----------

| Id  | Operation           | Name      | Rows  | Bytes |TempSpc| Cost (%CPU)|
Time     |

-------------------------------------------------------------------------------
----------

|   0 | SELECT STATEMENT    |           |  2565K|   129M|       | 18644   (4)|
00:03:44 |

|*  1 |  HASH JOIN          |           |  2565K|   129M|       | 18644   (4)|
00:03:44 |

|*  2 |   TABLE ACCESS FULL | TYPEAPPEL |     1 |    15 |       |     2   (0)|
00:00:01 |

|*  3 |   HASH JOIN         |           |  5131K|   185M|    60M| 18550   (4)|
00:03:43 |

|*  4 |    HASH JOIN        |           |  1534K|    42M|    29M|  5030   (4)|
00:01:01 |

|   5 |     TABLE ACCESS FULL| CLIENT    |  1063K|    17M|       |   901   (5)|
00:00:11 |

|   6 |     TABLE ACCESS FULL| LIGNE     |  1500K|    17M|       |   882   (6)|
00:00:11 |

|   7 |    TABLE ACCESS FULL | APPEL     |  4999K|    42M|       |  5349   (5)|
00:01:05 |
```

7) Création du Vue 3 (VM3) :

```
SQL> CREATE MATERIALIZED VIEW VM3
  2  BUILD IMMEDIATE
  3  REFRESH COMPLETE ON DEMAND
  4  ENABLE QUERY REWRITE
  5  AS
  6  SELECT c.NUMCLIENT,c.NOMCLIENT, t.CODETA ,t.TYPEAP
  7  FROM client c, ligne l ,appel a, typeappel t
  8  where c.numclient = l.numclient
  9  and l.numligne = a.numligne
 10  and t.codeta = a.codeta;

Vue matÚrialisÚe crÚÚe.
```

8) Ré exécuter la requête R1 et Examiner le temps et le plan d'exécution :

```
2368641 ligne(s) sÚlectionnÚe(s).

EcoulÚ : 00 :10 :01.84

Plan d'exÚcution
----------------------------------------------------------
Plan hash value: 646864740

-------------------------------------------------------------------------------
-----

| Id  | Operation                    | Name | Rows  | Bytes | Cost (%CPU)| Time
    |

-------------------------------------------------------------------------------
-----

|   0 | SELECT STATEMENT             |      |  2719K|   121M|  5805   (6)| 00:01
:10 |

|*  1 |  MAT_VIEW REWRITE ACCESS FULL| VM3  |  2719K|   121M|  5805   (6)| 00:01
:10 |

-------------------------------------------------------------------------------
-----


Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("VM3"."TYPEAP"='internationale')

Note
-----
   - dynamic sampling used for this statement
```

Comparaison:

Le temps d'exécution du R1 après la création du VM3 et inférieur qu'avant et on a

Sans VM4 : temps = 13 m 14 s

Avec VM4 : temps = 10 m 1 s

9) Ecrire une requête R2 pour obtenir le nombre d'appels par mois, année (Mois, Année,NBApp) :

```
SQL> select count(CodeAppel) as nombre_appel,extract(Month from DateApp) as Mois,extract(year from DateApp) as Annee
  2  from appel a
  3  inner join typeappel t on t.codeta = a.codeta
  4  group by t.typeap,extract(Month from DateApp) ,extract(year from DateApp) ;

NOMBRE_APPEL       MOIS      ANNEE
------------ ---------- ----------
       89228          1       2021
       89113          5       2020
       78174         11       2020
       80306          1       2021
       80650          8       2020
       80325         12       2020
       77245          6       2021
       86396          6       2021
       89160          1       2020
       80834          3       2020
       77868          9       2020

NOMBRE_APPEL       MOIS      ANNEE
------------ ---------- ----------
       86536          4       2020
       72893          2       2021
       89381         10       2021
       77562          4       2021
       89317          8       2020
       80893          8       2021
       86260          4       2021
       78034          6       2020
       80228          1       2020
       80754          7       2020
       77977          9       2021

NOMBRE_APPEL       MOIS      ANNEE
------------ ---------- ----------
       77301         11       2021
       80443          5       2021
       89470         10       2020
       86496         11       2020
       89206          5       2021
       86448          9       2021
       89230          3       2020
       80672          5       2020
       86746         12       2021
       78067         12       2021
       80344         10       2020

NOMBRE_APPEL       MOIS      ANNEE
------------ ---------- ----------
       78275          4       2020
       89629         12       2020
       83635          2       2020
       89772          7       2020
       79871          7       2021
       89472          3       2021
       80844          2       2021
       80388          3       2021
       75259          2       2020
       89682          8       2021
       86902          6       2020

NOMBRE_APPEL       MOIS      ANNEE
------------ ---------- ----------
       86211          9       2020
       89734          7       2021
       80229         10       2021
       86529         11       2021

48 ligne(s) sÚlectionnÚe(s).
```

10) le temps et le plan d'exécution :

```
48 ligne(s) sÚlectionnÚe(s).

EcoulÚ : 00 :00 :10.25

Plan d'exÚcution
----------------------------------------------------------
Plan hash value: 3117717671

-------------------------------------------------------------------------------
-
| Id  | Operation           | Name      | Rows  | Bytes | Cost (%CPU)| Time
|

-------------------------------------------------------------------------------
-
|   0 | SELECT STATEMENT    |           |  1033 | 26858 |  6312  (19)| 00:01:16
|
|   1 |  HASH GROUP BY      |           |  1033 | 26858 |  6312  (19)| 00:01:16
|
|*  2 |   HASH JOIN         |           |  4999K|   123M|  5423   (6)| 00:01:06
|
|   3 |    TABLE ACCESS FULL| TYPEAPPEL |     2 |    30 |     2   (0)| 00:00:01
|
|   4 |    TABLE ACCESS FULL| APPEL     |  4999K|    52M|  5331   (5)| 00:01:04
|

-------------------------------------------------------------------------------
-
```

11) Créer une vue matérialisée VM4 (Mois, Année, NBApp) en utilisant les options (IMMEDIATE, COMPLETE, ON DEMAND, ENABLE QUERY REWRITE) :

```
SQL> CREATE MATERIALIZED VIEW VM4
  2  BUILD IMMEDIATE
  3  REFRESH COMPLETE ON DEMAND
  4  ENABLE QUERY REWRITE
  5  AS
  6  SELECT extract(Month from DateApp) as Mois , extract(year from DateApp) as Annee, count(CodeAppel) as NBApp
  7  FROM appel
  8  group by extract(Month from DateApp) ,extract(year from DateApp) ;

Vue matÚrialisÚe crÚÚe.
```

12) Ré exécuter la requête R2 :

```
48 ligne(s) sÚlectionnÚe(s).

EcoulÚ : 00 :00 :04.04

Plan d'exÚcution
----------------------------------------------------------
Plan hash value: 3117717671

-------------------------------------------------------------------------------
-
| Id  | Operation           | Name      | Rows  | Bytes | Cost (%CPU)| Time
|

-------------------------------------------------------------------------------
-
|   0 | SELECT STATEMENT    |           |  1033 | 26858 |  6312  (19)| 00:01:16
|
|   1 |  HASH GROUP BY      |           |  1033 | 26858 |  6312  (19)| 00:01:16
|
|*  2 |   HASH JOIN         |           |  4999K|   123M|  5423   (6)| 00:01:06
|
|   3 |    TABLE ACCESS FULL| TYPEAPPEL |     2 |    30 |     2   (0)| 00:00:01
|
|   4 |    TABLE ACCESS FULL| APPEL     |  4999K|    52M|  5331   (5)| 00:01:04
|

-------------------------------------------------------------------------------
```

Comparaison:
Le temps d'exécution du R2 après la création du VM3 et inférieur qu'avant on a :
Sans VM4 : temps = 10 s
Avec VM4 : temps = 4 s

13) 1- Augmenter le nombre d'instances d'Appel de5000001 à 6000000 :

```
SQL> DECLARE
  2  CodeAppel number; Duree number ; DateApp date ; codeta number ; numligne number; codedo int;
  3  begin
  4    for CodeAppel in 5000001.. 6000000 loop
  5    Select floor(dbms_random.value(1,60.9)) into Duree from dual;
  6    Select TO_DATE( TRUNC(DBMS_RANDOM.VALUE(TO_CHAR(DATE'2020-01-01','J'), TO_CHAR(DATE'2021-12-31','J'))), 'J') into DateApp FROM DUAL;
  7    Select floor (dbms_random.value(1,2.9)) into codeta from dual;
  8    Select floor (dbms_random.value(1,1500255.9)) into numligne FROM DUAL;
  9    Select floor (dbms_random.value(1,522.9)) into codedo from dual;
 10    insert into Appel VALUES(CodeAppel,Duree,DateApp,codeta,numligne,codedo);
 11    end loop;
 12   commit;
 13  end;
 14  /

ProcÚdure PL/SQL terminÚe avec succPs.

EcoulÚ : 00 :07 :03.10
```


Temps exécution Sans VM4 :

```
48 ligne(s) sÚlectionnÚe(s).

EcoulÚ : 00 :00 :07.90

Plan d'exÚcution
----------------------------------------------------------
Plan hash value: 3117717671

-------------------------------------------------------------------------------------
-
| Id  | Operation           | Name      | Rows  | Bytes | Cost (%CPU)| Time
|

-------------------------------------------------------------------------------------
-
|   0 | SELECT STATEMENT    |           |  1033 | 26858 |  6312  (19)| 00:01:16
|
|   1 |  HASH GROUP BY      |           |  1033 | 26858 |  6312  (19)| 00:01:16
|
|*  2 |   HASH JOIN         |           |  4999K|   123M|  5423   (6)| 00:01:06
|
|   3 |    TABLE ACCESS FULL| TYPEAPPEL |     2 |    30 |     2   (0)| 00:00:01
|
|   4 |    TABLE ACCESS FULL| APPEL     |  4999K|    52M|  5331   (5)| 00:01:04
|

-------------------------------------------------------------------------------------
-
```

Temps exécution Avec VM4 :

```
48 ligne(s) sÚlectionnÚe(s).

EcoulÚ : 00 :00 :05.07

Plan d'exÚcution
----------------------------------------------------------
Plan hash value: 3117717671

-------------------------------------------------------------------------------
-
| Id  | Operation            | Name       | Rows  | Bytes | Cost (%CPU)| Time
|
-------------------------------------------------------------------------------
-
|   0 | SELECT STATEMENT     |            |  1033 | 26858 |  6312  (19)| 00:01:16
|
|   1 |  HASH GROUP BY       |            |  1033 | 26858 |  6312  (19)| 00:01:16
|
|*  2 |   HASH JOIN          |            |  4999K|   123M|  5423   (6)| 00:01:06
|
|   3 |    TABLE ACCESS FULL| TYPEAPPEL  |     2 |    30 |     2   (0)| 00:00:01
|
|   4 |    TABLE ACCESS FULL| APPEL      |  4999K|    52M|  5331   (5)| 00:01:04
|
-------------------------------------------------------------------------------
```

2- Augmenter le nombre d'instances d'Appel de 6000001 à 6500000 :

```
SQL> DECLARE
  2  CodeAppel number; Duree number ; DateApp date ; codeta number ; numligne number; codedo int;
  3  begin
  4    for CodeAppel in 6000001.. 6500000 loop
  5    Select floor(dbms_random.value(1,60.9)) into Duree from dual;
  6    Select TO_DATE( TRUNC(DBMS_RANDOM.VALUE(TO_CHAR(DATE'2020-01-01','J'), TO_CHAR(DATE'2021-12-31','J'))), 'J') into DateApp FROM DUAL;
  7    Select floor (dbms_random.value(1,2.9)) into codeta from dual;
  8    Select floor (dbms_random.value(1,1500255.9)) into numligne FROM DUAL;
  9    Select floor (dbms_random.value(1,522.9)) into codedo from dual;
 10    insert into Appel VALUES(CodeAppel,Duree,DateApp,codeta,numligne,codedo);
 11    end loop;
 12
 13   commit;
 14  end;
 15  /

ProcÚdure PL/SQL terminÚe avec succPs.

EcoulÚ : 00 :03 :28.39
```

Temps exécution Sans VM4 :

```
48 ligne(s) sÚlectionnÚe(s).

EcoulÚ : 00 :00 :07.27

Plan d'exÚcution
----------------------------------------------------------
Plan hash value: 3117717671

-------------------------------------------------------------------------------
-
| Id  | Operation           | Name      | Rows  | Bytes | Cost (%CPU)| Time
|

-------------------------------------------------------------------------------
-
|   0 | SELECT STATEMENT    |           |  1033 | 26858 |  6312  (19)| 00:01:16
|
|   1 |  HASH GROUP BY      |           |  1033 | 26858 |  6312  (19)| 00:01:16
|
|*  2 |   HASH JOIN         |           |  4999K|   123M|  5423   (6)| 00:01:06
|
|   3 |    TABLE ACCESS FULL| TYPEAPPEL |     2 |    30 |     2   (0)| 00:00:01
|
|   4 |    TABLE ACCESS FULL| APPEL     |  4999K|    52M|  5331   (5)| 00:01:04
|

-------------------------------------------------------------------------------
-
```

Temps exécution Avec VM4 :

```
48 ligne(s) sÚlectionnÚe(s).

EcoulÚ : 00 :00 :05.75

Plan d'exÚcution
----------------------------------------------------------
Plan hash value: 3117717671

-------------------------------------------------------------------------------
-
| Id  | Operation           | Name      | Rows  | Bytes | Cost (%CPU)| Time
|

-------------------------------------------------------------------------------
-
|   0 | SELECT STATEMENT    |           |  1033 | 26858 |  6312  (19)| 00:01:16
|
|   1 |  HASH GROUP BY      |           |  1033 | 26858 |  6312  (19)| 00:01:16
|
|*  2 |   HASH JOIN         |           |  4999K|   123M|  5423   (6)| 00:01:06
|
|   3 |    TABLE ACCESS FULL| TYPEAPPEL |     2 |    30 |     2   (0)| 00:00:01
|
|   4 |    TABLE ACCESS FULL| APPEL     |  4999K|    52M|  5331   (5)| 00:01:04
|

-------------------------------------------------------------------------------
-
```

14) Tableau comparatif des temps d'exécution (avec et sans la vue matérialisée) :

nombre instance	avec_vueM	sans_vueM
6000000	5.1 s	7.9 s
6500000	5.75 s	7.27 s

15) <u>Conclusions</u> : dans ce TP on conclus que :

- l'exécution du rafraichissement du vue d'option FAST est rapide que vue d'option COMPLETE
- l'exécution des requetés avec les vues est rapides que l'exécution sans vues
- Plus le nombre d'instance et petit l'exécution est rapide .