Rapport
TP 4

Réécriture des requêtes à travers les vues matérialisées

Réalisé en binôme par :
BENSALAH Kawthar / ABBACI Khaled
Numero du binôme : 22
Master 2 IL - Groupe 1
USTHB 2019/2020

TP 4

Objectif du TP

Réécriture des requêtes à travers les vues matérialisées

Introduction

À travers ce TP, nous allons comprendre l'utilité des vues matérialisées et les avantages qu'elles offrent lors de l'interrogation d'une base de données.

Réponse 1

- **Activation des options autotrace et timing de oracle :**

```
SQL> set autotrace on explain;
SQL> set timing on;
SQL>
```

Réponse 2

- **Changer le nom de la Wilaya numéro 31 à Oran :**

```
SQL> update Wilaya
  2     set NomWilaya = 'Oran'
  3   where CodeWilaya = 31;

1 ligne mise à jour.
```

- **Vider les buffers :**

```
SQL> alter system flush shared_pool;

Système modifié.

Ecoulé : 00 :00 :00.29
SQL> alter system flush buffer_cache;

Système modifié.

Ecoulé : 00 :00 :00.07
SQL>
```

- **Ecriture de la requête R1 qui permet d'obtenir la liste des clients ayant un compte dans une agence située à la wilaya d'Oran :**

```
SQL>   select C.NumClient, C.NomClient
  2      from Client C, Compte CC, Agence A, Ville V, Wilaya W
  3     where C.NumClient = CC.PossedeClient
  4     and CC.est_domicileAg = A.NumAgence
  5     and A.SeTrouveVille = V.CodeVille
  6     and V.DependWilaya = W.CodeWilaya
  7     and W.NomWilaya = 'Oran';

 NUMCLIENT NOMCLIENT
---------- --------------------------------------------------
       218 DAANEZQX
        54 LBABUVWP
       149 SGRFQPAJ
       175 NHELIAEZ
```

Réponse 3

- **Examination du temps d'exécution :**

```
3325 ligne(s) sélectionnée(s).

Ecoulé : 00 :00 :31.60
```

Le temps d'exécution de la requête est égale à 31.60 secondes.

- **Examination du plan d'exécution :**

```
Plan d'exécution
----------------------------------------------------------
Plan hash value: 2604713717

-------------------------------------------------------------------------------
--
| Id  | Operation                | Name    | Rows  | Bytes | Cost (%CPU)| Time
 |
-------------------------------------------------------------------------------
--
|   0 | SELECT STATEMENT         |         |  4167 |   227K|   815   (1)| 00:00:10
 |
|*  1 |  HASH JOIN               |         |  4167 |   227K|   815   (1)| 00:00:10
 |
|*  2 |   HASH JOIN              |         |  4167 |   162K|   268   (2)| 00:00:04
 |
|*  3 |    HASH JOIN             |         |   256 |  7680 |    26   (4)| 00:00:01
 |
|   4 |     TABLE ACCESS FULL    | VILLE   |   330 |  2310 |     3   (0)| 00:00:01
 |
|   5 |     MERGE JOIN CARTESIAN|         | 12300 |   276K|    22   (0)| 00:00:01
 |
|*  6 |      TABLE ACCESS FULL   | WILAYA |     1 |    14 |     3   (0)| 00:00:01
 |
|   7 |      BUFFER SORT         |         | 12300 |   108K|    19   (0)| 00:00:01
 |
|   8 |       TABLE ACCESS FULL | AGENCE | 12300 |   108K|    19   (0)| 00:00:01
 |
|   9 |    TABLE ACCESS FULL     | COMPTE |   200K|  1953K|   240   (1)| 00:00:03
 |
|  10 |   TABLE ACCESS FULL      | CLIENT |   100K|  1562K|   547   (1)| 00:00:07
 |
```

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


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

   1 - access("C"."NUMCLIENT"="CC"."POSSEDECLIENT")
   2 - access("CC"."EST_DOMICILEAG"="A"."NUMAGENCE")
   3 - access("A"."SETROUVEVILLE"="V"."CODEVILLE" AND
              "V"."DEPENDWILAYA"="W"."CODEWILAYA")
   6 - filter("W"."NOMWILAYA"='Oran')

SQL>
```

Remarque 1 : Le plan d'exécution permet de savoir de quelle manière le SGBD va exécuter la requête et quelles sont les tables utilisées.

Réponse 4

- **Création de la vue matérialisée VM5 :**

```
SQL> CREATE MATERIALIZED VIEW VM5
  2       BUILD IMMEDIATE REFRESH COMPLETE ON DEMAND
  3       enable query rewrite
  4       AS select C.NumClient, C.NomClient
  5          from Client C, Compte CC, Agence A, Ville V, Wilaya W
  6         where C.NumClient = CC.PossedeClient
  7         and CC.est_domicileAg = A.NumAgence
  8         and A.SeTrouveVille = V.CodeVille
  9         and V.DependWilaya = W.CodeWilaya;

Vue matérialisée créée.

Ecoulé : 00 :00 :00.84
SQL>
```

Réponse 5

- **Vider les buffers :**

```
SQL> alter system flush shared_pool;

Système modifié.

Ecoulé : 00 :00 :00.20
SQL> alter system flush buffer_cache;

Système modifié.

Ecoulé : 00 :00 :00.03
SQL>
```

- **Réexécution de la requête R1 :**

```
SQL>    select C.NumClient, C.NomClient
  2       from Client C, Compte CC, Agence A, Ville V, Wilaya W
  3      where C.NumClient = CC.PossedeClient
  4      and CC.est_domicileAg = A.NumAgence
  5      and A.SeTrouveVille = V.CodeVille
  6      and V.DependWilaya = W.CodeWilaya
  7      and W.NomWilaya = 'Oran';

 NUMCLIENT NOMCLIENT
---------- --------------------------------------------------
       218 DAANEZQX
        54 LBABUVWP
       149 SGREOPAJ
```

- **Examination du temps d'exécution :**

```
3325 ligne(s) sélectionnée(s).

Ecoulé : 00 :00 :29.14
```

- **Examination du plan d'exécution :**

```
Plan d'exécution
----------------------------------------------------------
Plan hash value: 2604713717

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

-------------------------------------------------------------------------------
--
|   0 | SELECT STATEMENT        |         |  4167 |   227K|   815   (1)| 00:00:10
 |
|*  1 |  HASH JOIN              |         |  4167 |   227K|   815   (1)| 00:00:10
 |
|*  2 |   HASH JOIN             |         |  4167 |   162K|   268   (2)| 00:00:04
 |
|*  3 |    HASH JOIN            |         |   256 |  7680 |    26   (4)| 00:00:01
 |
|   4 |     TABLE ACCESS FULL   | VILLE   |   330 |  2310 |     3   (0)| 00:00:01
 |
|   5 |     MERGE JOIN CARTESIAN|         | 12300 |   276K|    22   (0)| 00:00:01
 |
|*  6 |      TABLE ACCESS FULL  | WILAYA  |     1 |    14 |     3   (0)| 00:00:01
 |
|   7 |      BUFFER SORT        |         | 12300 |   108K|    19   (0)| 00:00:01
 |
|   8 |       TABLE ACCESS FULL | AGENCE  | 12300 |   108K|    19   (0)| 00:00:01
 |
|   9 |    TABLE ACCESS FULL    | COMPTE  |   200K|  1953K|   240   (1)| 00:00:03
 |
|  10 |   TABLE ACCESS FULL     | CLIENT  |   100K|  1562K|   547   (1)| 00:00:07
 |
```

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


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

   1 - access("C"."NUMCLIENT"="CC"."POSSEDECLIENT")
   2 - access("CC"."EST_DOMICILEAG"="A"."NUMAGENCE")
   3 - access("A"."SETROUVEVILLE"="V"."CODEVILLE" AND
              "V"."DEPENDWILAYA"="W"."CODEWILAYA")
   6 - filter("W"."NOMWILAYA"='Oran')

SQL>
```

Remarque 2: On remarque que le temps d'exécution de la requête R1 est presque égale à celui examiné avant la création de la vue VM5 (question 3).

Ainsi , le plan d'exécution de la requête R1 après la création de la vue VM5 n'a pas changé : consultation des tables Ville, Willaya, Agence, Compte et Client au lieu de VM5.

Réponse 6

- Création de la vue matérialisée VM6

```
SQL> CREATE MATERIALIZED VIEW VM6
  2       BUILD IMMEDIATE REFRESH COMPLETE ON DEMAND
  3       enable query rewrite
  4       AS select C.NumClient, C.NomClient
  5          from Client C, Compte CC, Agence A, Ville V, Wilaya W
  6         where C.NumClient = CC.PossedeClient
  7         and CC.est_domicileAg = A.NumAgence
  8         and A.SeTrouveVille = V.CodeVille
  9         and V.DependWilaya = W.CodeWilaya
 10         and W.NomWilaya = 'Oran';

Vue matérialisée créée.

Ecoulé : 00 :00 :00.53
SQL>
```

Réponse 7

- Vider les buffers :

```
SQL> alter system flush shared_pool;

Système modifié.

Ecoulé : 00 :00 :00.20
SQL> alter system flush buffer_cache;

Système modifié.

Ecoulé : 00 :00 :00.04
SQL>
```

- Réexécution de la requête R1 :

```
SQL>    select C.NumClient, C.NomClient
  2       from Client C, Compte CC, Agence A, Ville V, Wilaya W
  3      where C.NumClient = CC.PossedeClient
  4      and CC.est_domicileAg = A.NumAgence
  5      and A.SeTrouveVille = V.CodeVille
  6      and V.DependWilaya = W.CodeWilaya
  7      and W.NomWilaya = 'Oran';

 NUMCLIENT NOMCLIENT
---------- --------------------------------------------------
       218 DAANEZQX
        54 LBABUVWP
       149 SGRFQPAJ
       175 NHELIAEZ
```

- **Examination du temps d'exécution :**

```
3325 ligne(s) sélectionnée(s).

Ecoulé : 00 :00 :02.84
```

- **Examination du plan d'exécution :**

```
Plan d'exécution
----------------------------------------------------------
Plan hash value: 1412056132

-------------------------------------------------------------------------------
-----
| Id  | Operation                     | Name | Rows  | Bytes | Cost (%CPU)| Time
    |
-------------------------------------------------------------------------------
-----
|   0 | SELECT STATEMENT              |      |  3325 |   129K|     5   (0)| 00:00
:01 |
|   1 |  MAT_VIEW REWRITE ACCESS FULL| VM6  |  3325 |   129K|     5   (0)| 00:00
:01 |
-------------------------------------------------------------------------------
-----


Note
-----
   - dynamic sampling used for this statement (level=2)

SQL>
```

Remarque 3 : On remarque une baisse du temps d'exécution de la requête R1 par rapport à celui examiné avant la création de VM5 et celui d'après. De plus, le plan d'exécution de la requête R1 n'est plus le même : consultation de la vue

VM6 seulement contrairement au plans examinés précédemment (question 3 et 5) où figuraient les tables Ville, Willaya, Agence, Compte et Client.

Réponse 8

- **Vider les buffers :**

```
SQL> alter system flush shared_pool;

Système modifié.

Ecoulé : 00 :00 :00.22
SQL> alter system flush buffer_cache;

Système modifié.

Ecoulé : 00 :00 :00.14
SQL>
```

- **Ecriture de la requête R2 qui permet d'obtenir le nombre d'opérations par Banque :**

```
SQL> select B.CodeBanque, B.NomBanque, count(O.CodeOp) as NbOperations
  2    from Banque B, Compte C, Agence A, Operation O
  3   where (O.VersementCompte = C.NumCompte or O.RetraitCompte = C.NumCompte)
  4   and C.est_domicileAg = A.NumAgence
  5   and A.AppartientBanque = B.CodeBanque
  6   group by B.CodeBanque, B.NomBanque;

CODEBANQUE NOMBANQUE                                           NBOPERATIONS
---------- -------------------------------------------------- ------------
         9 BNA 9                                                     62742
         2 BNA 2                                                     60769
         5 BNA 5                                                     62195
         3 BNA 3                                                     62144
         8 BNA 8                                                     62536
         1 BNA 1                                                     62644
         4 BNA 4                                                     63620
        10 BNA 10                                                    54587
         7 BNA 7                                                     60830
         6 BNA 6                                                     58247
```

Réponse 9

- **Examination du temps d'exécution :**

```
10 ligne(s) sélectionnée(s).

Ecoulé : 00 :00 :01.07
```

Le temps d'exécution de la requête est égale à 01.07 secondes.

- **Examination du plan d'exécution :**

```
Plan d'exécution
----------------------------------------------------------
Plan hash value: 1361406634

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

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

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

|   0 | SELECT STATEMENT       |           |    71 |  2769 |       |  3317   (2)
| 00:00:40 |

|   1 |  HASH GROUP BY         |           |    71 |  2769 |       |
|          |

|   2 |   CONCATENATION        |           |       |       |       |
|          |

|*  3 |    HASH JOIN           |           |   289K|    10M|       |  1637   (2)
| 00:00:20 |

|   4 |     TABLE ACCESS FULL  | BANQUE    |    10 |   140 |       |     3   (0)
| 00:00:01 |

|*  5 |     HASH JOIN          |           |   289K|  7064K|       |  1632   (2)
| 00:00:20 |

|   6 |      TABLE ACCESS FULL | AGENCE    | 12300 | 98400 |       |    19   (0)
| 00:00:01 |

|*  7 |      HASH JOIN         |           |   289K|  4804K|  4304K|  1611   (2)
| 00:00:20 |

|   8 |       TABLE ACCESS FULL| COMPTE    |   200K|  1953K|       |   240   (1)
| 00:00:03 |

|*  9 |       TABLE ACCESS FULL| OPERATION |   289K|  1978K|       |   897   (2)
| 00:00:11 |

|* 10 |    HASH JOIN           |           |   320K|    11M|       |  1666   (2)
| 00:00:20 |
```

```
|  11 |     TABLE ACCESS FULL  | BANQUE    |    10 |   140 |       |     3   (0)
| 00:00:01 |

|* 12 |     HASH JOIN          |           |   320K|  7835K|       |  1661   (2)
| 00:00:20 |

|  13 |      TABLE ACCESS FULL | AGENCE    | 12300 | 98400 |       |    19   (0)
| 00:00:01 |

|* 14 |      HASH JOIN         |           |   320K|  5328K|  4304K|  1639   (2)
| 00:00:20 |

|  15 |       TABLE ACCESS FULL| COMPTE    |   200K|  1953K|       |   240   (1)
| 00:00:03 |

|* 16 |       TABLE ACCESS FULL| OPERATION |   320K|  2193K|       |   897   (2)
| 00:00:11 |

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


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

   3 - access("A"."APPARTIENTBANQUE"="B"."CODEBANQUE")
   5 - access("C"."EST_DOMICILEAG"="A"."NUMAGENCE")
   7 - access("O"."RETRAITCOMPTE"="C"."NUMCOMPTE")
   9 - filter("O"."RETRAITCOMPTE" IS NOT NULL)
  10 - access("A"."APPARTIENTBANQUE"="B"."CODEBANQUE")
  12 - access("C"."EST_DOMICILEAG"="A"."NUMAGENCE")
  14 - access("O"."VERSEMENTCOMPTE"="C"."NUMCOMPTE")
       filter(LNNVL("O"."RETRAITCOMPTE"="C"."NUMCOMPTE") OR
              LNNVL("O"."RETRAITCOMPTE" IS NOT NULL))
  16 - filter("O"."VERSEMENTCOMPTE" IS NOT NULL)

SQL>
```

Réponse 10

- **Création de la vue matérialisée VM7 :**

```
SQL>  CREATE MATERIALIZED VIEW VM7
  2      BUILD IMMEDIATE REFRESH COMPLETE ON DEMAND
  3      enable query rewrite
  4      AS select B.CodeBanque, B.NomBanque, count(O.CodeOp) as NbOperations
  5    from Banque B, Compte C, Agence A, Operation O
  6   where (O.VersementCompte = C.NumCompte or O.RetraitCompte = C.NumCompte)
  7   and C.est_domicileAg = A.NumAgence
  8   and A.AppartientBanque = B.CodeBanque
  9   group by B.CodeBanque, B.NomBanque;

Vue matérialisée créée.

Ecoulé : 00 :00 :26.66
SQL>
```

Réponse 11

- **Vider les buffers :**

```
Ecoulé : 00 :00 :26.66
SQL> alter system flush shared_pool;

Système modifié.

Ecoulé : 00 :00 :00.21
SQL> alter system flush buffer_cache;

Système modifié.

Ecoulé : 00 :00 :00.38
SQL>
```

- **Réexécution de la requête R2 :**

```
SQL> select B.CodeBanque, B.NomBanque, count(O.CodeOp) as NbOperations
  2    from Banque B, Compte C, Agence A, Operation O
  3   where (O.VersementCompte = C.NumCompte or O.RetraitCompte = C.NumCompte)
  4   and C.est_domicileAg = A.NumAgence
  5   and A.AppartientBanque = B.CodeBanque
  6   group by B.CodeBanque, B.NomBanque;

CODEBANQUE NOMBANQUE                                           NBOPERATIONS
---------- -------------------------------------------------- ------------
         9 BNA 9                                                     63285
         8 BNA 8                                                     62683
         2 BNA 2                                                     60733
         5 BNA 5                                                     62347
         3 BNA 3                                                     62086
         1 BNA 1                                                     62536
        10 BNA 10                                                    54801
         4 BNA 4                                                     63368
         7 BNA 7                                                     60669
         6 BNA 6                                                     57806

10 ligne(s) sélectionnée(s).
```

- **Examination du temps d'exécution :**

```
10 ligne(s) sélectionnée(s).

Ecoulé : 00 :00 :00.23
```

- **Examination du plan d'exécution :**

```
Plan d'exécution
----------------------------------------------------------
Plan hash value: 432514644

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

-------------------------------------------------------------------------------
-----
|   0 | SELECT STATEMENT              |      |    10 |   530 |     3   (0)| 00:00
:01 |
|   1 |  MAT_VIEW REWRITE ACCESS FULL| VM7  |    10 |   530 |     3   (0)| 00:00
:01 |
-------------------------------------------------------------------------------
-----


Note
-----
   - dynamic sampling used for this statement (level=2)

SQL>
```

Remarque 4 : On remarque que le temps d'exécution de la requête R2 a diminué comparé à celui examiné avant la création de la vue VM7. Le plan d'exécution a également changé et ne contient plus les tables : Agence, Banque, Compte , Opération ainsi que les jointures entre celles ci. Il contient seulement la vue VM7.

Réponse 12

- **Augmentation du nombre d'instance de la table Operation à 800000 :**

```
SQL> DECLARE
  2  d DATE;
  3  heure char(5);
  4  typeop char(1);
  5  montant number;
  6  cpt number;
  7  I number;
  8  begin
  9    for i in 610315..800000  loop
 10      select TO_DATE(TRUNC(dbms_random.value(
 11        TO_CHAR(DATE'2015-01-01','J'),
 12        TO_CHAR(DATE'2018-12-31','J'))),'J')
 13        into d
 14        from dual;
 15      select trunc(dbms_random.value(5000,100000),2)
 16        into montant
 17        from dual;
 18      select dbms_random.string('U',5)
 19        into heure
 20        from dual;
 21      select FLOOR(dbms_random.value(1,2.9))
 22        into typeop
 23        from dual;
 24      select FLOOR(dbms_random.value(1,200000))
 25        into cpt
 26        from dual;
 27      insert into Operation
 28      values (i,d,heure,typeop,montant,
 29        decode(typeop,1,cpt,null),
 30        decode(typeop,2,cpt,null));
 31    end loop
 32  COMMIT;
 33  end;
 34  /

Procédure PL/SQL terminée avec succès.

Ecoulé : 00 :01 :23.51
SQL>
```

- **Rafraichissement de la vue VM7 :**

```
SQL> execute DBMS_MVIEW.REFRESH('VM7');

Procédure PL/SQL terminée avec succès.

Ecoulé : 00 :00 :08.18
SQL>
```

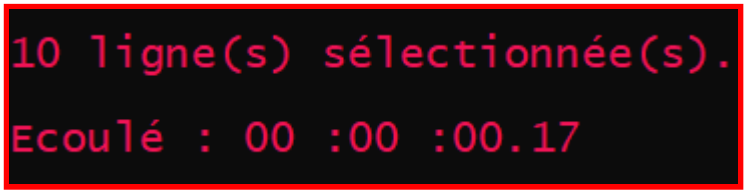
- **Vider les buffers :**

```
SQL> alter system flush shared_pool;

Système modifié.

Ecoulé : 00 :00 :00.20
SQL> alter system flush buffer_cache;

Système modifié.

Ecoulé : 00 :00 :00.18
SQL>
```

- **Réexécution de la requête R2 avec VM7 et 800000 instances :**

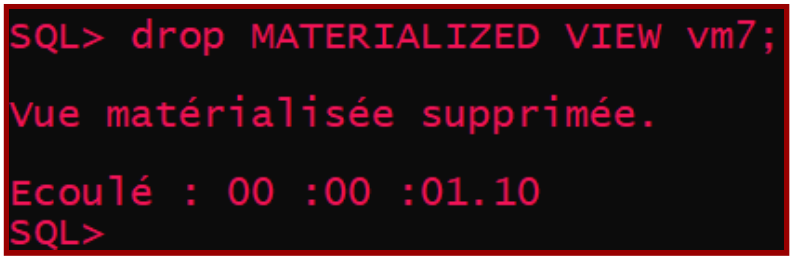
```
SQL> select B.CodeBanque, B.NomBanque, count(O.CodeOp) as NbOperations
  2    from Banque B, Compte C, Agence A, Operation O
  3   where (O.VersementCompte = C.NumCompte or O.RetraitCompte = C.NumCompte)
  4   and C.est_domicileAg = A.NumAgence
  5   and A.AppartientBanque = B.CodeBanque
  6   group by B.CodeBanque, B.NomBanque;

CODEBANQUE NOMBANQUE                                           NBOPERATIONS
---------- -------------------------------------------------- ------------
         9 BNA 9                                                     82954
         5 BNA 5                                                     81613
         2 BNA 2                                                     79719
         8 BNA 8                                                     81950
         3 BNA 3                                                     81310
         1 BNA 1                                                     81775
         4 BNA 4                                                     83299
        10 BNA 10                                                    71943
         7 BNA 7                                                     79593
         6 BNA 6                                                     75844

10 ligne(s) sélectionnée(s).
```

- **Examination du temps d'exécution avec VM7 et 800000 instances :**



- **Examination du plan d'exécution avec VM7 et 800000 instances :**

- **Suppression de la vue VM7 :**



- **Vider les buffers :**

```
SQL> alter system flush shared_pool;

Système modifié.

Ecoulé : 00 :00 :00.18
SQL> alter system flush buffer_cache;

Système modifié.

Ecoulé : 00 :00 :00.06
SQL>
```

- **Réexécution de la requête R2 sans VM7 et 800000 instances :**

```
SQL> select B.CodeBanque, B.NomBanque, count(O.CodeOp) as NbOperations
  2    from Banque B, Compte C, Agence A, Operation O
  3   where (O.VersementCompte = C.NumCompte or O.RetraitCompte = C.NumCompte)
  4   and C.est_domicileAg = A.NumAgence
  5   and A.AppartientBanque = B.CodeBanque
  6   group by B.CodeBanque, B.NomBanque;

CODEBANQUE NOMBANQUE                                          NBOPERATIONS
---------- -------------------------------------------------- ------------
         9 BNA 9                                                     82954
         5 BNA 5                                                     81613
         2 BNA 2                                                     79719
         8 BNA 8                                                     81950
         3 BNA 3                                                     81310
         1 BNA 1                                                     81775
         4 BNA 4                                                     83299
        10 BNA 10                                                    71943
         7 BNA 7                                                     79593
         6 BNA 6                                                     75844

10 ligne(s) sélectionnée(s).
```

- **Examination du temps d'exécution sans VM7 et 800000 instances :**

```
10 ligne(s) sélectionnée(s).

Ecoulé : 00 :00 :01.73
```

- **Examination du plan d'exécution sans VM7 et 800000 instances :**

```
Plan d'exécution
----------------------------------------------------------
Plan hash value: 1361406634

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

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

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

|   0 | SELECT STATEMENT       |           |    71 |  2769 |       |  3317   (2)
| 00:00:40 |

|   1 |  HASH GROUP BY         |           |    71 |  2769 |       |
|          |

|   2 |   CONCATENATION        |           |       |       |       |
|          |

|*  3 |    HASH JOIN           |           |   289K|    10M|       |  1637   (2)
| 00:00:20 |

|   4 |     TABLE ACCESS FULL  | BANQUE    |    10 |   140 |       |     3   (0)
| 00:00:01 |

|*  5 |     HASH JOIN          |           |   289K|  7064K|       |  1632   (2)
| 00:00:20 |

|   6 |      TABLE ACCESS FULL | AGENCE    | 12300 | 98400 |       |    19   (0)
| 00:00:01 |

|*  7 |      HASH JOIN         |           |   289K|  4804K|  4304K|  1611   (2)
| 00:00:20 |

|   8 |       TABLE ACCESS FULL| COMPTE    |   200K|  1953K|       |   240   (1)
| 00:00:03 |

|*  9 |       TABLE ACCESS FULL| OPERATION |   289K|  1978K|       |   897   (2)
| 00:00:11 |

|* 10 |    HASH JOIN           |           |   320K|    11M|       |  1666   (2)
```

```
|  11 |     TABLE ACCESS FULL  | BANQUE    |    10 |   140 |       |     3   (0)
| 00:00:01 |

|* 12 |     HASH JOIN          |           |   320K|  7835K|       |  1661   (2)
| 00:00:20 |

|  13 |      TABLE ACCESS FULL | AGENCE    | 12300 | 98400 |       |    19   (0)
| 00:00:01 |

|* 14 |      HASH JOIN         |           |   320K|  5328K|  4304K|  1639   (2)
| 00:00:20 |

|  15 |       TABLE ACCESS FULL| COMPTE    |   200K|  1953K|       |   240   (1)
| 00:00:03 |

|* 16 |       TABLE ACCESS FULL| OPERATION |   320K|  2193K|       |   897   (2)
| 00:00:11 |

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


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

   3 - access("A"."APPARTIENTBANQUE"="B"."CODEBANQUE")
   5 - access("C"."EST_DOMICILEAG"="A"."NUMAGENCE")
   7 - access("O"."RETRAITCOMPTE"="C"."NUMCOMPTE")
   9 - filter("O"."RETRAITCOMPTE" IS NOT NULL)
  10 - access("A"."APPARTIENTBANQUE"="B"."CODEBANQUE")
  12 - access("C"."EST_DOMICILEAG"="A"."NUMAGENCE")
  14 - access("O"."VERSEMENTCOMPTE"="C"."NUMCOMPTE")
       filter(LNNVL("O"."RETRAITCOMPTE"="C"."NUMCOMPTE") OR
              LNNVL("O"."RETRAITCOMPTE" IS NOT NULL))
  16 - filter("O"."VERSEMENTCOMPTE" IS NOT NULL)

SQL>
```

- **Augmentation du nombre d'instance de la table Operation à 1000000 :**

```
SQL> DECLARE
  2  d DATE;
  3  heure char(5);
  4  typeop char(1);
  5  montant number;
  6  cpt number;
  7  I number;
  8  begin
  9    for i in 800001..1000000  loop
 10      select TO_DATE(TRUNC(dbms_random.value(
 11        TO_CHAR(DATE'2015-01-01','J'),
 12        TO_CHAR(DATE'2018-12-31','J'))),'J')
 13        into d
 14        from dual;
 15      select trunc(dbms_random.value(5000,100000),2)
 16        into montant
 17        from dual;
 18      select dbms_random.string('U',5)
 19        into heure
 20        from dual;
 21      select FLOOR(dbms_random.value(1,2.9))
 22        into typeop
 23        from dual;
 24      select FLOOR(dbms_random.value(1,200000))
 25        into cpt
 26        from dual;
 27      insert into Operation
 28      values (i,d,heure,typeop,montant,
 29        decode(typeop,1,cpt,null),
 30        decode(typeop,2,cpt,null));
 31    end loop
 32  COMMIT;
 33  end;
 34  /

Procédure PL/SQL terminée avec succès.
```

- **Vider les buffers :**

```
SQL> alter system flush shared_pool;

Système modifié.

Ecoulé : 00 :00 :00.26
SQL> alter system flush buffer_cache;

Système modifié.

Ecoulé : 00 :00 :00.11
SQL>
```

- **Réexécution de la requête R2 sans VM7 et 1000000 instances :**

```
SQL> select B.CodeBanque, B.NomBanque, count(O.CodeOp) as NbOperations
  2    from Banque B, Compte C, Agence A, Operation O
  3   where (O.VersementCompte = C.NumCompte or O.RetraitCompte = C.NumCompte)
  4   and C.est_domicileAg = A.NumAgence
  5   and A.AppartientBanque = B.CodeBanque
  6   group by B.CodeBanque, B.NomBanque;

CODEBANQUE NOMBANQUE                                           NBOPERATIONS
---------- -------------------------------------------------- ------------
         9 BNA 9                                                    103517
         5 BNA 5                                                    101902
         2 BNA 2                                                     99653
         8 BNA 8                                                    102402
         3 BNA 3                                                    101737
         1 BNA 1                                                    102413
         4 BNA 4                                                    104133
        10 BNA 10                                                    89858
         7 BNA 7                                                     99566
         6 BNA 6                                                     94819

10 ligne(s) sélectionnée(s).
```

- **Examination du temps d'exécution sans VM7 et 1000000 instances :**

```
10 ligne(s) sélectionnée(s).

Ecoulé : 00 :00 :02.17
```

- **Examination du plan d'exécution sans VM7 et 1000000 instances :**

```
Plan d'exécution
----------------------------------------------------------
Plan hash value: 1361406634

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

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

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

|   0 | SELECT STATEMENT       |           |    71 |  2769 |       |  3317   (2)
| 00:00:40 |

|   1 |  HASH GROUP BY         |           |    71 |  2769 |       |
|          |

|   2 |   CONCATENATION        |           |       |       |       |
|          |

|*  3 |    HASH JOIN           |           |   289K|    10M|       |  1637   (2)
| 00:00:20 |

|   4 |     TABLE ACCESS FULL  | BANQUE    |    10 |   140 |       |     3   (0)
| 00:00:01 |

|*  5 |     HASH JOIN          |           |   289K|  7064K|       |  1632   (2)
| 00:00:20 |

|   6 |      TABLE ACCESS FULL | AGENCE    | 12300 | 98400 |       |    19   (0)
| 00:00:01 |

|*  7 |      HASH JOIN         |           |   289K|  4804K|  4304K|  1611   (2)
| 00:00:20 |

|   8 |       TABLE ACCESS FULL| COMPTE    |   200K|  1953K|       |   240   (1)
| 00:00:03 |

|*  9 |       TABLE ACCESS FULL| OPERATION |   289K|  1978K|       |   897   (2)
| 00:00:11 |

|* 10 |    HASH JOIN           |           |   320K|    11M|       |  1666   (2)
```

```
|  11 |     TABLE ACCESS FULL  | BANQUE    |    10 |   140 |       |     3   (0)
| 00:00:01 |

|* 12 |     HASH JOIN          |           |   320K|  7835K|       |  1661   (2)
| 00:00:20 |

|  13 |      TABLE ACCESS FULL | AGENCE    | 12300 | 98400 |       |    19   (0)
| 00:00:01 |

|* 14 |      HASH JOIN         |           |   320K|  5328K|  4304K|  1639   (2)
| 00:00:20 |

|  15 |       TABLE ACCESS FULL| COMPTE    |   200K|  1953K|       |   240   (1)
| 00:00:03 |

|* 16 |       TABLE ACCESS FULL| OPERATION |   320K|  2193K|       |   897   (2)
| 00:00:11 |

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


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

   3 - access("A"."APPARTIENTBANQUE"="B"."CODEBANQUE")
   5 - access("C"."EST_DOMICILEAG"="A"."NUMAGENCE")
   7 - access("O"."RETRAITCOMPTE"="C"."NUMCOMPTE")
   9 - filter("O"."RETRAITCOMPTE" IS NOT NULL)
  10 - access("A"."APPARTIENTBANQUE"="B"."CODEBANQUE")
  12 - access("C"."EST_DOMICILEAG"="A"."NUMAGENCE")
  14 - access("O"."VERSEMENTCOMPTE"="C"."NUMCOMPTE")
       filter(LNNVL("O"."RETRAITCOMPTE"="C"."NUMCOMPTE") OR
              LNNVL("O"."RETRAITCOMPTE" IS NOT NULL))
  16 - filter("O"."VERSEMENTCOMPTE" IS NOT NULL)

SQL>
```

- **Création de la vue VM7 :**

```
SQL>  CREATE MATERIALIZED VIEW VM7
  2     BUILD IMMEDIATE REFRESH COMPLETE ON DEMAND
  3     enable query rewrite
  4     AS select B.CodeBanque, B.NomBanque, count(O.CodeOp) as NbOperations
  5    from Banque B, Compte C, Agence A, Operation O
  6   where (O.VersementCompte = C.NumCompte or O.RetraitCompte = C.NumCompte)
  7   and C.est_domicileAg = A.NumAgence
  8   and A.AppartientBanque = B.CodeBanque
  9   group by B.CodeBanque, B.NomBanque;

Vue matérialisée créée.

Ecoulé : 00 :00 :02.16
SQL>
```

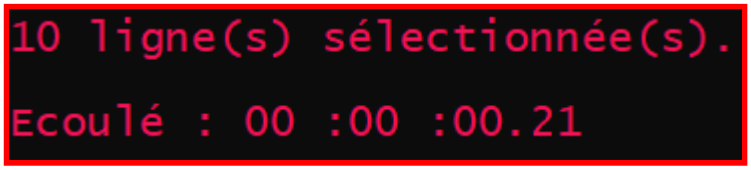
- **Vider les buffers :**

```
SQL> alter system flush shared_pool;

Système modifié.

Ecoulé : 00 :00 :00.21
SQL> alter system flush buffer_cache;

Système modifié.

Ecoulé : 00 :00 :00.08
SQL>
```

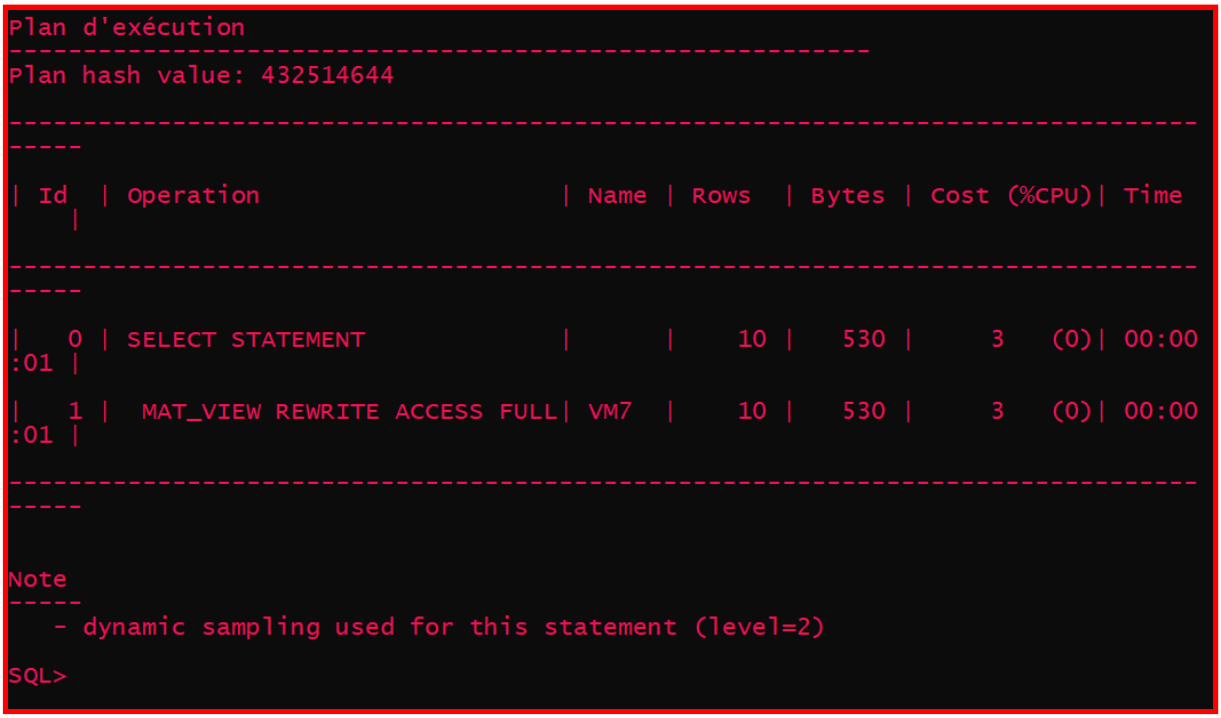
- **Réexécution de la requête R2 avec VM7 et 1000000 instances :**

```
SQL> select B.CodeBanque, B.NomBanque, count(O.CodeOp) as NbOperations
  2    from Banque B, Compte C, Agence A, Operation O
  3   where (O.VersementCompte = C.NumCompte or O.RetraitCompte = C.NumCompte)
  4   and C.est_domicileAg = A.NumAgence
  5   and A.AppartientBanque = B.CodeBanque
  6   group by B.CodeBanque, B.NomBanque;

CODEBANQUE NOMBANQUE                                          NBOPERATIONS
---------- -------------------------------------------------- ------------
         9 BNA 9                                                    103517
         5 BNA 5                                                    101902
         2 BNA 2                                                     99653
         8 BNA 8                                                    102402
         3 BNA 3                                                    101737
         1 BNA 1                                                    102413
         4 BNA 4                                                    104133
        10 BNA 10                                                    89858
         7 BNA 7                                                     99566
         6 BNA 6                                                     94819

10 ligne(s) sélectionnée(s).
```

- **Examination du temps d'exécution avec VM7 et 1000000 instances :**



- **Examination du plan d'exécution avec VM7 et 1000000 instances :**



Réponse 13

- **Tableau comparatif des temps d'exécution**

Nombre d'instance	610314	800000	1000000
Sans la vue matérialisée	00 : 00 : 01.07	00 : 00 : 01.73	00 : 00 : 02.17
Avec la vue matérialisée	00 : 00 : 00.23	00 : 00 : 00.17	00 : 00 : 00.21

Remarque 6 : D'après le tableau ci-dessus, on remarque que les temps d'exécution de la requête R2 avec l'utilisation de VM7 sont moins important que son temps d'exécutions sans utilisation de VM7 et cela quel que soit le nombre d'instance de la table Operation.

Réponse 14

- **Conclusions :**

  - Les vues matérialisée jouent un rôle important dans l'optimisation des temps d'exécutions des requêtes.

  - L'option **ENABLE QUERY REWRITE** active la fonction de réécriture de requête de l'optimiseur afin d'améliorer les performances et ceci en utilisant les informations des vues matérialisées dans l'exécution des requêtes.

  - Lors de l'exécution des requêtes, l'optimiseur choisit d'accéder aux vues matérialisées même si les requêtes elles même ne faisant aucune mention de ces vues (question 11, 12 et 13) . Sans ces dernières, l'optimiseur doit utiliser les tables d'origine mentionnées dans la requête ce qui est très coûteux.

  - L'optimiseur essaie de réécrire une requête en comparant son texte avec le texte de la définition des vues matérialisées. Dans le cas où les textes ne sont pas identiques, l'optimiseur utilisent les tables mentionnées dans la requête (question 5).

Conclusion

Après la réalisation de ce TP, nous pouvons dire que les vues matérialisées sont très utiles et permettent un gain de performance assez important et ceci via les méthodes de réécriture des requêtes.