Rapport
TP 3

Maîtrise des vues matérialisées

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

TP 3

Objectif du TP

Maîtrise des vues matérialisées

Introduction

À travers ce TP, nous allons comprendre les caractéristiques des vues matérialisées, qui sont différentes des vues, afin de les utiliser à bon escient.

Réponse 1

- **Création d'une vue matérialisée VM1 :**

```
SQL> CREATE MATERIALIZED VIEW VM1
  2       BUILD IMMEDIATE REFRESH COMPLETE ON DEMAND
  3       AS select CodeOp, DateOp
  4         from Operation
  5         where TypeOp = '2'
  6         and DateOp>='01/01/2018'
  7         and DateOp<='31/01/2018';

Vue matérialisée créée.

SQL>
```

Réponse 2

- **Création du journal LOG pour utiliser FAST sur la table Operation:**

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

Journal de vue matérialisée créé.

SQL>
```

- **Création d'une vue matérialisée VM2 :**

```
SQL> CREATE MATERIALIZED VIEW VM2
  2       BUILD IMMEDIATE REFRESH FAST ON DEMAND
  3       AS select CodeOp, DateOp
  4         from Operation
  5         where TypeOp = '2'
  6         and DateOp>='01/01/2018'
  7         and DateOp<='31/01/2018';

Vue matérialisée créée.

SQL>
```

Réponse 3

- **L'ajout d'un retrait en janvier 2018 dans la table Operation**

```
SQL> INSERT INTO Operation
  2    VALUES (999999,'21/01/2018',
  3           '19:23','2',90000.21,NULL,28);

1 ligne créée.

SQL>
```

- **Consultation de l'état du journal log avant le rafraîchissement :**

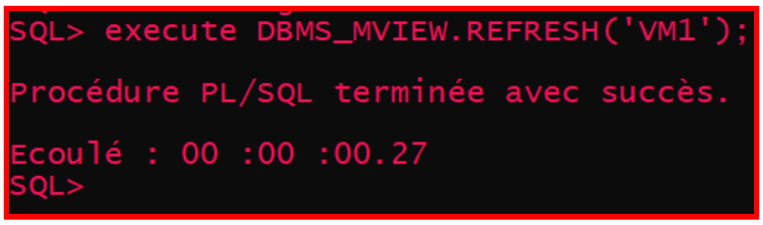
```
SQL> select * from MLOG$_Operation;

    CODEOP SNAPTIME D O
---------- -------- - -
CHANGE_VECTOR$$
--------------------------------
     XID$$
----------
    999999 01/01/00 D O
00
2,8148E+15

    999999 01/01/00 I N
FE
2,8148E+15

    CODEOP SNAPTIME D O
---------- -------- - -
CHANGE_VECTOR$$
--------------------------------
     XID$$
----------
```

Remarque 1 : Le journal log a stocké la ligne décrivant la modification apportée aux données de la table Operation (ajout d'un retrait en janvier 2018)

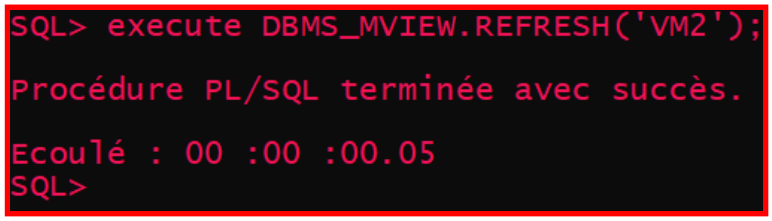
- **Utilisation de commande set timing on pour examiner les temps de rafraîchissement :**

```
SQL> set timing on
SQL>
```

- **Rafraichissement de la vue matérialisée VM1 :**

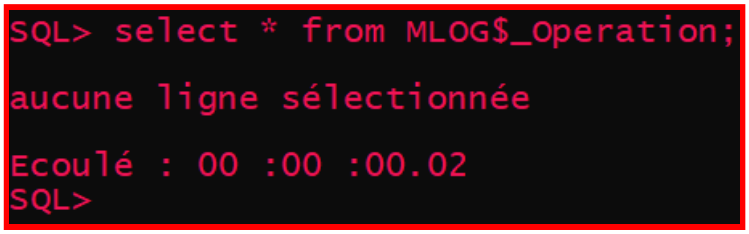


- **Rafraichissement de la vue matérialisée VM2 :**



Remarque 2 : Nous remarquons qu'avec l'option FAST, le temps d'exécution du rafraîchissement est beaucoup plus rapide.

- **Consultation de l'état du journal log après le rafraîchissement :**



Remarque 3 : Le journal est vidé après le rafraîchissement des deux vues. Le journal log sert donc à actualiser les vues matérialisées en fonction de la table principale.

- Répercussion de l'ajout d'un retrait en janvier 2018 sur la vue VM1

```
SQL> select * from VM1 where CodeOp=999999;

    CODEOP DATEOP
---------- --------
    999999 21/01/18

Ecoulé : 00 :00 :00.01
SQL>
```

- Répercussion de l'ajout d'un retrait en janvier 2018 sur la vue VM2

```
SQL> select * from VM2 where CodeOp=999999;

    CODEOP DATEOP
---------- --------
    999999 21/01/18

Ecoulé : 00 :00 :00.03
SQL>
```

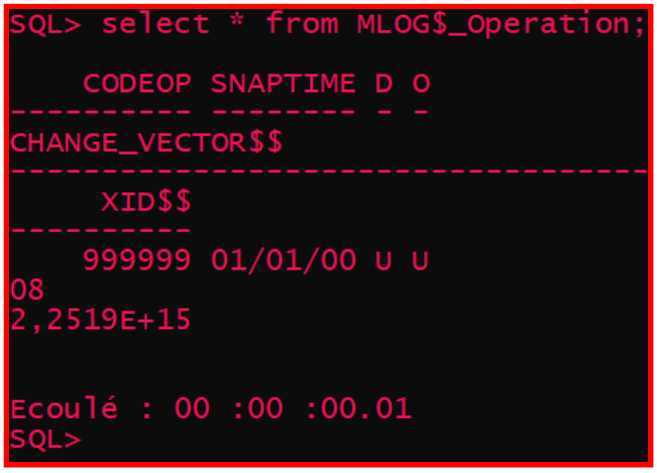
Réponse 4

Cas 1 : Modification

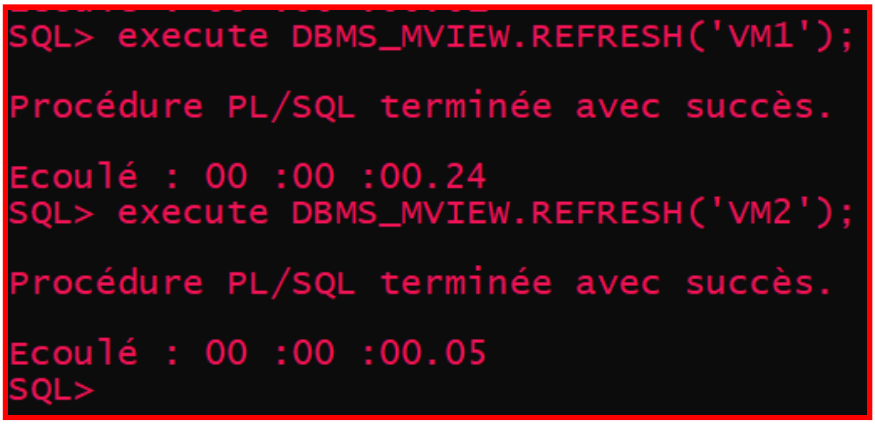
- La modification d'un tuple de la table Operation

```
SQL> UPDATE Operation
  2    SET DateOp='07/01/2018'
  3    Where CodeOp=999999;

1 ligne mise à jour.

Ecoulé : 00 :00 :00.01
SQL>
```

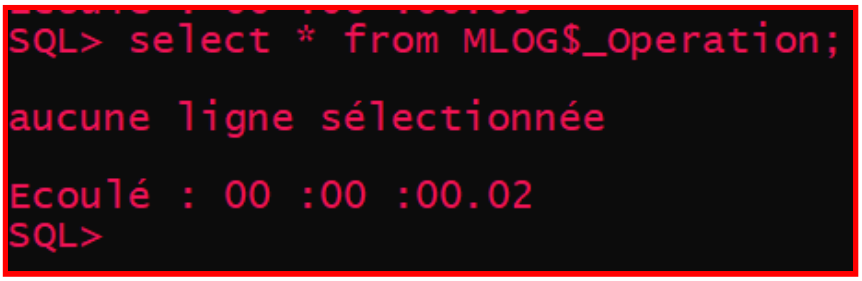
- **Consultation de l'état du journal log avant le rafraîchissement :**



- **Rafraîchissement des vues matérialisée VM1 et VM2 :**



- **Consultation de l'état du journal log après le rafraîchissement :**



- Répercussion de la modification sur les vues VM1 et VM2

```
SQL> select * from VM1 where CodeOp=999999;

    CODEOP DATEOP
---------- --------
    999999 07/01/18

Ecoulé : 00 :00 :00.03
SQL> select * from VM2 where CodeOp=999999;

    CODEOP DATEOP
---------- --------
    999999 07/01/18

Ecoulé : 00 :00 :00.02
SQL>
```

Cas 2 : Suppression

- La suppression d'un tuple de la table Operation

```
SQL> DELETE FROM Operation
  2    Where CodeOp=999999;

1 ligne supprimée.

Ecoulé : 00 :00 :00.02
SQL>
```

- Consultation de l'état du journal log avant le rafraîchissement :

```
SQL> select * from MLOG$_Operation;

    CODEOP SNAPTIME D O
---------- -------- - -
CHANGE_VECTOR$$
--------------------------------
     XID$$
----------
    999999 01/01/00 D O
00
1,1260E+15


Ecoulé : 00 :00 :00.09
SQL>
```

- **Rafraîchissement des vues matérialisée VM1 et VM2 :**

```
SQL> execute DBMS_MVIEW.REFRESH('VM1');

Procédure PL/SQL terminée avec succès.

Ecoulé : 00 :00 :00.23
SQL> execute DBMS_MVIEW.REFRESH('VM2');

Procédure PL/SQL terminée avec succès.

Ecoulé : 00 :00 :00.05
SQL>
```

- **Consultation de l'état du journal log après le rafraîchissement :**

```
SQL> select * from MLOG$_Operation;

aucune ligne sélectionnée

Ecoulé : 00 :00 :00.00
SQL>
```

- **Répercussion de la suppression sur les vues VM1 et VM2**

```
SQL> select * from VM1 where CodeOp=999999;

aucune ligne sélectionnée

Ecoulé : 00 :00 :00.02
SQL> select * from VM2 where CodeOp=999999;

aucune ligne sélectionnée

Ecoulé : 00 :00 :00.01
SQL>
```

- Tableau comparatif des temps d'exécution des rafraîchissement des deux vues VM1 ET VM2 :

Temps d'exécution du rafraîchissement	VM1	VM2
Insertion	**00 : 00 : 00.27**	**00 : 00 : 00.05**
Modification	**00 : 00 : 00.24**	**00 : 00 : 00.05**
Suppression	**00 : 00 : 00.23**	**00 : 00 : 00.05**

Remarque 4 : Nous remarquons qu'avec l'option FAST, le temps d'exécution du rafraîchissement est beaucoup plus rapide dans tous les cas (insertion, modification et suppression).

Réponse 5

- Création d'une vue matérialisée VM3

```
SQL> CREATE MATERIALIZED VIEW VM3
  2       BUILD IMMEDIATE REFRESH COMPLETE ON COMMIT
  3       AS select CodeOp, DateOp
  4         from Operation O, Compte C
  5        where
  6        (O.VersementCompte=C.NumCompte or O.RetraitCompte=C.NumCompte)
  7        and C.est_domicileAg>=1 and C.est_domicileAg<=100;

Vue matérialisée créée.

Ecoulé : 00 :00 :00.21
SQL>
```

Réponse 6

- Recherche d'un compte de l'agence 88

```
SQL> SELECT NumCompte From Compte Where est_domicileAg=88;

 NUMCOMPTE
----------
     40195
     65453
    137210
    139449
    146922
    147940
    120642
    159510
    186255

9 ligne(s) sélectionnée(s).

Ecoulé : 00 :00 :00.08
SQL>
```

Nous prenons le compte numéro 40195.

- Ajout d'une opération dans l'agence 88

```
SQL> INSERT INTO Operation
  2    VALUES (999999,'04/11/2019','10:25',
  3          '1',1000000.21,40195,NULL);

1 ligne créée.

Ecoulé : 00 :00 :00.01
SQL>
```

- Validation avec COMMIT

```
SQL> COMMIT;

Validation effectuée.

Ecoulé : 00 :00 :00.22
SQL>
```

- Test de la mise à jour de VM3

```
SQL> SELECT CodeOp FROM VM3 WHERE CodeOp=999999;

    CODEOP
----------
    999999

Ecoulé : 00 :00 :00.03
SQL>
```

Réponse 7

- Création d'une vue matérialisée VM4

```
SQL> CREATE MATERIALIZED VIEW VM4
  2       BUILD IMMEDIATE REFRESH COMPLETE ON DEMAND
  3       AS select est_domicileAg as CodeAgence,
  4                  count(NumCompte) as NBComptes
  5         from Compte
  6        group by est_domicileAg;

Vue matérialisée créée.

Ecoulé : 00 :00 :00.12
SQL>
```

Réponse 8

- Création du journal LOG pour utiliser FAST sur la table Compte :

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

Journal de vue matérialisée créé.

Ecoulé : 00 :00 :00.06
SQL>
```

- **Création d'une vue matérialisée identiques à VM3 avec l'option FAST**

```
SQL> CREATE MATERIALIZED VIEW VM33
  2       BUILD IMMEDIATE REFRESH FAST ON COMMIT
  3       AS select CodeOp, DateOp
  4         from Operation O, Compte C
  5        where
  6        (O.VersementCompte=C.NumCompte or O.RetraitCompte=C.NumCompte)
  7        and C.est_domicileAg>=1 and C.est_domicileAg<=100;
     and C.est_domicileAg>=1 and C.est_domicileAg<=100
                                                  *
ERREUR à la ligne 7 :
ORA-12052: Régénération de type Fast impossible sur la vue matérialisée
MASTER.VM33


Ecoulé : 00 :00 :00.03
SQL>
```

- **Création d'une vue matérialisée identiques à VM4 avec l'option FAST**

```
SQL> CREATE MATERIALIZED VIEW VM44
  2       BUILD IMMEDIATE REFRESH FAST ON DEMAND
  3       AS select est_domicileAg as CodeAgence,
  4                 count(NumCompte) as NBComptes
  5         from Compte
  6        group by est_domicileAg;
      from Compte
           *
ERREUR à la ligne 5 :
ORA-12032: impossible d'utiliser la colonne d'ID de ligne (rowid) du journal
des vues matérialisées sur "MASTER"."COMPTE"


Ecoulé : 00 :00 :00.02
SQL>
```

- **Le problème**

En comparant la création de la vue VM2 avec la création des vues matérialisées identiques à VM3 et à VM4 avec l'option **FAST** et en consultant la documentation d'oracle, nous déduisons que :

- On n'a pas pu créer une vue matérialisée identiques à VM3 avec l'option **FAST** car la requête contient une **jointure** entre deux tables(Compte et Operation) .

- On n'a pas pu créer une vue matérialisée identiques à VM4 avec l'option **FAST** car la requête contient un **count** et un **group by**.

- Solution pour créer une vue matérialisée identique à VM3

Selon la documentation d'oracle, la définition de requêtes pour les vues matérialisées avec des jointures uniquement doit respecter les conditions suivantes pour le rafraîchissement rapide:

- Ils ne peuvent pas avoir de group by ou d'agrégats.
- Les rowID de toutes les tables interrogées doivent figurer dans la des attributs sélectionnés de la requête.
- Les rowID de toutes les tables interrogées doivent figurer dans les journaux de la vue matérialisée.
- Les journaux log doivent être présents sur toutes les tables référencées dans la requête qui définit la vue matérialisée.

L'une des solutions que nous avons trouvé est représentée dans la capture suivante :

- Ajout de rowid dans le journal des tables Operation et Compte :

```
SQL> Drop MATERIALIZED VIEW LOG ON Operation;

Journal de vue matérialisée supprimé.

Ecoulé : 00 :00 :00.02
SQL> Drop MATERIALIZED VIEW LOG ON Compte;

Journal de vue matérialisée supprimé.

Ecoulé : 00 :00 :00.03
SQL> CREATE MATERIALIZED VIEW LOG ON Operation with rowid;

Journal de vue matérialisée créé.

Ecoulé : 00 :00 :00.03
SQL> CREATE MATERIALIZED VIEW LOG ON Compte with rowid;

Journal de vue matérialisée créé.

Ecoulé : 00 :00 :00.01
SQL>
```

- Création de la vue matérialisée identique à VM3 :

```
SQL> CREATE MATERIALIZED VIEW VM33
  2       NOLOGGING CACHE
  3       BUILD IMMEDIATE REFRESH FAST ON COMMIT
  4       AS select Operation.CodeOp, Operation.DateOp,
  5                 Operation.ROWID AS OROW,
  6                 Compte.ROWID AS CROW
  7         from Operation, Compte
  8        where
  9        (Operation.VersementCompte=Compte.NumCompte
 10         or Operation.RetraitCompte=Compte.NumCompte)
 11        and Compte.est_domicileAg>=1
 12        and Compte.est_domicileAg<=100;

Vue matérialisée créée.

Ecoulé : 00 :00 :00.14
SQL>
```

Remarque 5 : Selon la documentation d'oracle, la clause **logging** nous permet de spécifier si certaines opérations DML seront enregistrées dans le fichier journal redo (LOGGING) ou non (NOLOGGING).

- Solution pour créer une vue matérialisée identique à VM4

Selon la documentation d'oracle, la définition de requêtes pour les vues matérialisées avec des agrégats est soumise aux restrictions de rafraîchissement rapide suivantes:

- Toutes les tables de la vue matérialisée doivent avoir des journaux log, et ces journaux log doivent:
- Contenir toutes les colonnes de la table référencée dans la vue matérialisée.
- Spécifier **SEQUENCE** pour indiquer qu'une valeur de séquence fournissant des informations de commande supplémentaires doit être enregistrée dans le journal de la vue matérialisée.
- être Spécifiés avec **ROWID** et **INCLUDING NEW VALUES**
- COUNT(*) doit être spécifié.

La solution pour la création de la table est représentée les captures suivantes :

- Ajout des **rowid** nécessaires dans le journal des tables Operation et Compte ainsi que l'option **including new values** et **with sequence** :

```
SQL> Drop MATERIALIZED VIEW LOG ON Operation;

Journal de vue matérialisée supprimé.

Ecoulé : 00 :00 :00.01
SQL> Drop MATERIALIZED VIEW LOG ON Compte;

Journal de vue matérialisée supprimé.

Ecoulé : 00 :00 :00.02
SQL> CREATE MATERIALIZED VIEW LOG ON Operation
  2  WITH SEQUENCE, ROWID (CodeOp, DateOp,
  3                        VersementCompte,
  4                        RetraitCompte)
  5  INCLUDING NEW VALUES;

Journal de vue matérialisée créé.

Ecoulé : 00 :00 :00.02
SQL> CREATE MATERIALIZED VIEW LOG ON Compte
  2  WITH SEQUENCE, ROWID (NumCompte,
  3                        est_domicileAg)
  4  INCLUDING NEW VALUES;

Journal de vue matérialisée créé.

Ecoulé : 00 :00 :00.02
SQL>
```

- Création de la vue matérialisée identique à VM4 :

```
SQL> CREATE MATERIALIZED VIEW VM44
  2       BUILD IMMEDIATE REFRESH FAST ON DEMAND
  3       AS select est_domicileAg as CodeAgence,
  4                  count(NumCompte) as NBComptes
  5         from Compte
  6        group by est_domicileAg;

Vue matérialisée créée.

Ecoulé : 00 :00 :00.08
SQL>
```

Conclusion

Après avoir réalisé ce TP, nous pouvons conclure que les vues matérialisée, contrairement à des vues standards, ont une existence concrète et permettent de stocker le résultat **SELECT**.
Elle diffèrent aussi des vues car elles sont copiées à partir des tables de base et par conséquent, elles ont besoin d'un mécanisme de rafraîchissement qui dépend des options choisies lors de la création.