Nom: Akrouchi Rihab
Matricule: 202031050231
groupe: Master 2 IL groupe 2

# Rapport tp sded 2

## TP 3

## Introduction :

Les vues matérialisées jouent un rôle crucial dans l'optimisation des performances des bases de données en permettant le stockage physique des résultats de requêtes complexes. Ce rapport examine le processus de création, d'actualisation, et de gestion des vues matérialisées en utilisant différentes méthodes de rafraîchissement, notamment les options FAST et COMPLETE. Chaque étape s'inscrit dans l'optimisation des performances des requêtes, particulièrement pour les systèmes traitant de grands volumes de données où la rapidité et l'efficacité sont primordiales.

1.création d'une vue matérialisée VM1 :

```
SQL> CREATE MATERIALIZED VIEW VM1
  2  BUILD IMMEDIATE
  3  REFRESH COMPLETE ON DEMAND
  4  AS
  5  SELECT CodeTrajet, datetrajet
  6  FROM trajet
  7  WHERE datetrajet BETWEEN TO_DATE('01-Jun-23', 'DD-MON-YY') AND TO_DATE('30-Jun-23', 'DD-MON-YY');

Materialized view created.
```

2.création d'une vue matérialisée VM2 :

```
SQL> CREATE MATERIALIZED VIEW log on trajet ;

Materialized view log created.

SQL>
SQL> CREATE MATERIALIZED VIEW VM2
  2  BUILD IMMEDIATE
  3  REFRESH fast ON DEMAND
  4  AS
  5  SELECT CodeTrajet, datetrajet
  6  FROM trajet
  7  WHERE datetrajet BETWEEN TO_DATE('01-Jun-23', 'DD-MON-YY') AND TO_DATE('30-Jun-23', 'DD-MON-YY');

Materialized view created.
```

3.Tester les MAJ (AJOUT,SUPPRESSION,MODIFICATION)

```
Elapsed: 00:00:00.00
SQL> INSERT INTO trajet (CodeTrajet, datetrajet) VALUES ('12300000', TO_DATE('25-Jun-23', 'DD-MON-YY'));

1 row created.

Elapsed: 00:00:00.00
SQL> UPDATE trajet SET datetrajet = TO_DATE('26-Jun-23', 'DD-MON-YY') WHERE CodeTrajet = '12300000';

1 row updated.
```

```
Elapsed: 00:00:00.00
SQL> DELETE FROM trajet WHERE CodeTrajet = '12300000';

1 row deleted.

Elapsed: 00:00:00.01
```

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

PL/SQL procedure successfully completed.

Elapsed: 00:00:03.48
SQL> EXEC DBMS_MVIEW.REFRESH('VM2');

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.06
```

comparaison : la vm2 est plus rapide que vm1
car on a utilisé fast dans vm2 et complete dans vm1

## 4.Insertion d'une nouvelle ligne

```
SQL> INSERT INTO trajet (CodeTrajet, datetrajet)
  2  VALUES ('12500000', TO_DATE('15-Jun-24', 'DD-MON-YY'));

1 row created.
```

```
SQL> INSERT INTO trajet (CodeTrajet, datetrajet)
  2  VALUES ('12500000', TO_DATE('15-Jun-24', 'DD-MON-YY'));

1 row created.

Elapsed: 00:00:00.00
```

## Consulter le journal :

```
SQL> SELECT * FROM MLOG$_TRAJET;

CODETRAJET SNAPTIME$ D O
---------- --------- - -
CHANGE_VECTOR$$
--------------------------------------------------------------------------------
     XID$$
----------
  12300000 01-JAN-00 I N
FEFF
1.9703E+15

  12300000 01-JAN-00 U U
0400
```

```
CODETRAJET SNAPTIME$ D O
---------- --------- - -
CHANGE_VECTOR$$
--------------------------------------------------------------------------------
     XID$$
----------
1.9703E+15


Elapsed: 00:00:00.00
```

-Refresher la vue VM2

```
SQL> EXEC DBMS_MVIEW.REFRESH('VM2');

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.68
```

**5.CRÉER ET ALIMENTER UNE VUE VM3**

```
SQL> CREATE MATERIALIZED VIEW VM3
  2  BUILD DEFERRED
  3  REFRESH COMPLETE ON DEMAND
  4  AS
  5  SELECT t.CodeTrajet, t.DateTrajet
  6  FROM Trajet t
  7  JOIN Localisation l ON t.CodeDepart = l.GPS
  8  WHERE l.CODEVILLE = (SELECT v.CodeVille FROM Ville v WHERE v.NomVille = 'Alger');

Materialized view created.
```

6.Alimenter VM3

```
SQL> EXEC DBMS_MVIEW.REFRESH('VM3');

PL/SQL procedure successfully completed.
```

7. CRÉER LA VUE VM4

```
SQL> CREATE MATERIALIZED VIEW VM4
  2  BUILD IMMEDIATE
  3  REFRESH COMPLETE ON DEMAND
  4  AS
  5  SELECT EXTRACT(MONTH FROM DateTrajet) AS MoisTrajet,
  6         EXTRACT(YEAR FROM DateTrajet) AS AnneeTrajet,
  7         COUNT(*) AS NBTrajets
  8  FROM Trajet
  9  GROUP BY EXTRACT(MONTH FROM DateTrajet), EXTRACT(YEAR FROM DateTrajet);

Materialized view created.
```

8. Création de Vues Matérialisées Identiques à VM3 et VM4 avec l'Option Fast

```
SQL> CREATE MATERIALIZED VIEW VM3_FAST
  2  BUILD IMMEDIATE
  3  REFRESH FAST ON DEMAND
  4  AS
  5  SELECT t.CodeTrajet, t.DateTrajet
  6  FROM Trajet t
  7  JOIN Localisation l ON t.CodeDepart = l.GPS
  8  WHERE l.CODEVILLE = (SELECT v.CodeVille FROM Ville v WHERE v.NomVille = 'Alger');
JOIN Localisation l ON t.CodeDepart = l.GPS
                                      *
ERROR at line 7:
ORA-12015: cannot create a fast refresh materialized view from a complex query
```

```
SQL> CREATE MATERIALIZED VIEW VM4_FAST
  2  BUILD IMMEDIATE
  3  REFRESH FAST ON DEMAND
  4  AS
  5  SELECT EXTRACT(MONTH FROM t.DateTrajet) AS MoisTrajet,
  6         EXTRACT(YEAR FROM t.DateTrajet) AS AnneeTrajet,
  7         COUNT(*) AS NBTrajets
  8  FROM Trajet t
  9  GROUP BY EXTRACT(MONTH FROM t.DateTrajet),
 10           EXTRACT(YEAR FROM t.DateTrajet);
FROM Trajet t
     *
ERROR at line 8:
ORA-12032: cannot use rowid column from materialized view log on
"MASTER22"."TRAJET"
```

## Problème:

On trouve un probleme lors la création des vues avec l'option fast

On peut pas créer une vue avec une requête complexe

Je propose de créer un journal de vue matérialisée approprié ou utiliser des sous-requêtes simples sinon utiliser l'option complete

```
SQL> CREATE MATERIALIZED VIEW LOG ON Localisation
  2  WITH PRIMARY KEY, ROWID
  3  INCLUDING NEW VALUES;

Materialized view log created.
```

```
SQL> CREATE MATERIALIZED VIEW VM4
  2  BUILD IMMEDIATE
  3  REFRESH COMPLETE ON DEMAND
  4  AS
  5  SELECT EXTRACT(MONTH FROM t.DateTrajet) AS MoisTrajet,
  6         EXTRACT(YEAR FROM t.DateTrajet) AS AnneeTrajet,
  7         COUNT(*) AS NBTrajets
  8  FROM Trajet t
  9  GROUP BY EXTRACT(MONTH FROM t.DateTrajet),
 10           EXTRACT(YEAR FROM t.DateTrajet);

Materialized view created.
```

```
SQL> CREATE MATERIALIZED VIEW VM3
  2  BUILD IMMEDIATE
  3  REFRESH COMPLETE ON DEMAND
  4  AS
  5  SELECT t.CodeTrajet, t.DateTrajet
  6  FROM Trajet t
  7  JOIN Localisation l ON t.CodeDepart = l.GPS
  8  WHERE l.CODEVILLE = (SELECT v.CodeVille FROM Ville v WHERE v.NomVille = 'Alger');

Materialized view created.
```

Ces étapes devraient aider à créer et à maintenir votre vue matérialisée `VM3` avec un rafraîchissement rapide.

**Conclusion :**

L'expérimentation des vues matérialisées VM1, VM2, VM3, et VM4 a permis d'illustrer l'impact des différentes stratégies de rafraîchissement sur les performances des requêtes. L'option FAST s'est montrée plus efficace que COMPLETE dans les cas de vues simples, bien qu'elle présente des limitations avec les requêtes complexes. Pour surmonter ces contraintes, il est recommandé de privilégier les requêtes simples ou de maintenir un journal de vues approprié pour optimiser le rafraîchissement rapide. Ces pratiques contribuent à une gestion optimale des ressources et garantissent des performances améliorées dans les bases de données.