Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Informatique

Spécialité : MASTER 1 Big Data Analytics

Rapport de TP2 ENDO

Travail présenté à Monsieur SELMOUNE Nazih

Travail présenté par :
AISSANI Anouar
161835024828

Année Universitaire: 2021/2022

1. Créer une vue matérialisée VM1 contenant les appels réalisée en Juin 2021 en utilisant les options (IMMEDIATE, COMPLETE, ON DEMAND)

```
SQL> CREATE MATERIALIZED VIEW VM1
  2  BUILD IMMEDIATE
  3  REFRESH COMPLETE ON DEMAND
  4  AS SELECT CodeAppel
  5  FROM Appel
  6   WHERE EXTRACT(YEAR FROM DateAppel) = 2021
  7  AND EXTRACT(MONTH FROM DateAppel) = 6;

Materialized view created.
```

2. Créer une vue matérialisée VM2 identique à VM1, en utilisant les options (IMMEDIATE, FAST, ON DEMAND)

```
SQL> CREATE MATERIALIZED VIEW
  2  LOG ON Appel;

Materialized view log created.

SQL> CREATE MATERIALIZED VIEW VM2
  2  BUILD IMMEDIATE
  3  REFRESH FAST ON DEMAND
  4  AS SELECT *
  5  FROM Appel WHERE EXTRACT(YEAR FROM DateAppel) = 2021
  6  AND EXTRACT(MONTH FROM DateAppel) = 6;

Materialized view created.
```

3. Testez les répercussions des mises à jour de la base de données, sur les deux vues (ajout, suppression, modification), en examinant et comparant les temps d'exécution du rafraichissement des deux vues.

```
--Insertion:
SQL> INSERT INTO APPEL VALUES(3500221, 20, to_date('2021-03-17','yyyy-mm-dd'), 15,
300, 2);

1 row created.

SQL> EXECUTE DBMS_MVIEW.REFRESH('VM1');

PL/SQL procedure successfully completed.

Elapsed: 00:00:01.31
SQL> EXECUTE DBMS_MVIEW.REFRESH('VM2');

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.05

--Modification
SQL> UPDATE APPEL SET DureeAppel = 30 WHERE CodeAppel = 3500221;

1 row updated.

SQL> EXECUTE DBMS_MVIEW.REFRESH('VM1');

PL/SQL procedure successfully completed.

Elapsed: 00:00:01.31
SQL> EXECUTE DBMS_MVIEW.REFRESH('VM2');

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.05
```

```
--Suppression
SQL> DELETE FROM APPEL WHERE CodeAppel = 3500221;

1 row deleted.

Elapsed: 00:00:00.00

SQL> EXECUTE DBMS_MVIEW.REFRESH('VM1');

PL/SQL procedure successfully completed.

Elapsed: 00:00:01.29
SQL> EXECUTE DBMS_MVIEW.REFRESH('VM2');

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.05
```

	Insertion	Modification	Suppresion
VM1	00:00:01.31	00:00:01.31	00:00:01.29
VM2	00:00:00.05	00:00:00.05	00:00:00.05

On remarque que le temps d'exécution du rafraichissement de la première vue **"VM1"** est toujours plus grand que le temps d'exécution du rafraichissement de la deuxième vue **"VM2".**

4. Activer les options autotrace, et timing de oracle.

```
SQL> SET TIMING ON
SQL> SET AUTOTRACE ON EXPLAIN
SQL>
```

5. Ecrire une requête R1 pour obtenir la liste des clients (CodeCl, NomCl) ayant effectué des appels de type internaional.

```
SQL> SELECT c.NumClient AS CodeCl, c.NomClient AS NomCl
  2  FROM Client c, Ligne l, Appel a, TypeAppel t
  3  WHERE t.TypeAppel = 'Internationale'
  4  AND t.CodeTypeAppel = a.CodeTypeAppel
  5  AND a.NumeroLigne = l.NumeroLigne
  6  AND l.NumClient = c.NumClient;

1657775 rows selected.

Elapsed: 00:05:44.48

Execution Plan
----------------------------------------------------------
Plan hash value: 881011081

-------------------------------------------------------------------------------------------
| Id  | Operation            | Name      | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |           |  1750K|    90M|       | 16583   (2)| 00:00:01 |
|*  1 |  HASH JOIN           |           |  1750K|    90M|       | 16583   (2)| 00:00:01 |
|*  2 |   TABLE ACCESS FULL  | TYPEAPPEL |     1 |    16 |       |     3   (0)| 00:00:01 |
|*  3 |   HASH JOIN          |           |  3500K|   126M|    58M| 16553   (2)| 00:00:01 |
|*  4 |    HASH JOIN         |           |  1500K|    41M|    29M|  5386   (2)| 00:00:01 |
|   5 |     TABLE ACCESS FULL| CLIENT    |  1065K|    17M|       |  1107   (2)| 00:00:01 |
|   6 |  TABLE ACCESS FULL| LIGNE     |  1500K|    17M|       |  1079   (3)| 00:00:01
|
|   7 |  TABLE ACCESS FULL| APPEL     |  3500K|    30M|       |  4711   (2)| 00:00:01
|
-------------------------------------------------------------------------------------------
-------

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

   1 - access("T"."CODETYPEAPPEL"="A"."CODETYPEAPPEL")
```

```
  2 - filter("T"."TYPEAPPEL"='Internationale')
  3 - access("A"."NUMEROLIGNE"="L"."NUMEROLIGNE")
  4 - access("L"."NUMCLIENT"="C"."NUMCLIENT")
```

6. Examiner le temps et le plan d'exécution.

- **Temps d'execution : `00:05:44.48`**
- **Le plan d'exécution contient les tables: TypeAppel, Client, Ligne, Appel.**

7. Créer une vue matérialisée VM3 (CodeCl, NomCl, CodeTypeAppel, TypeAppel) en utilisant les options (IMMEDIATE, COMPLETE, ON DEMAND, ENABLE QUERY REWRITE) contenant une jointure entre les tables Client, Ligne, Appel et TypeAppel.

```
SQL> CREATE MATERIALIZED VIEW VM3
  2  BUILD IMMEDIATE
  3  REFRESH COMPLETE ON DEMAND
  4  ENABLE QUERY REWRITE AS
  5  SELECT c.NumClient AS CodeCl, c.NomClient AS NomCl,
  6  t.CodeTypeAppel AS CodeTypeAppel, t.TypeAppel AS TypeAppel
  7  FROM Client c, Ligne l, Appel a, TypeAppel t
  8  WHERE t.CodeTypeAppel = a.CodeTypeAppel
  9  AND a.NumeroLigne = l.NumeroLigne
 10  AND l.NumClient = c.NumClient;

Materialized view created.
```

8. Ré exécuter la requête R1. Examiner le temps et le plan d'exécution et comparer avec (6).

```
SQL> alter system flush shared_pool;

System altered.

SQL> alter system flush buffer_cache;

System altered.

SQL> SELECT c.NumClient AS CodeCl, c.NomClient AS NomCl
  2  FROM Client c, Ligne l, Appel a, TypeAppel t
  3  WHERE t.TypeAppel = 'Internationale'
  4  AND t.CodeTypeAppel = a.CodeTypeAppel
  5  AND a.NumeroLigne = l.NumeroLigne
  6  AND l.NumClient = c.NumClient;

1657775 rows selected.

Elapsed: 00:04:33.14

Execution Plan
----------------------------------------------------------
Plan hash value: 646864740

-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |      |  1750K|    50M|  4911   (2)| 00:00:01 |
|*  1 |  MAT_VIEW REWRITE ACCESS FULL| VM3  |  1750K|    50M|  4911   (2)| 00:00:01 |
-------------------------------------------------------------------------------------------

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

   1 - filter("VM3"."TYPEAPPEL"='Internationale')
```

Temps d'exécution	Avant la création de VM3	Après la création de VM3
R1	00:05:44.48	00:04:33.14

Le temps d'exécution de la reqeuête "R1" après la création de la vue "VM3" est inférieur au temps d'exécution avant la création de la vue "VM3".

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

```
SQL> SELECT COUNT(*) AS NBApp, EXTRACT(MONTH FROM DateAppel) AS Mois,
EXTRACT(YEAR FROM DateAppel) AS Annee
  2  FROM Appel
  3  GROUP BY EXTRACT(MONTH FROM DateAppel), EXTRACT(YEAR FROM DateAppel);

    NBAPP |       MOIS |       ANNEE
--------- | ---------- | ----------
   144096 |         12 |       2021
   134547 |          2 |       2021
   149213 |          3 |       2020
   143142 |          4 |       2020
   142856 |          9 |       2020
   148588 |          5 |       2021
   143519 |          4 |       2021
   144718 |         11 |       2021
   148469 |          7 |       2021
   148083 |          1 |       2021
   148418 |         10 |       2021
   148915 |          8 |       2021
   143695 |         11 |       2020
   148499 |          1 |       2020
   148706 |          7 |       2020
   144460 |          9 |       2021
   148931 |          8 |       2020
   144330 |          6 |       2021
   148820 |          3 |       2021
```

```
    147949 |          5 |       2020
    139364 |          2 |       2020
    148440 |         12 |       2020
    148667 |         10 |       2020
    143795 |          6 |       2020

24 rows selected.

Elapsed: 00:00:01.32

Execution Plan
----------------------------------------------------------
Plan hash value: 585468147

-------------------------------------------------------------------------------
| Id  | Operation          | Name  | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |       |   730 |  5840 |  4957   (7)| 00:00:01 |
|   1 |  HASH GROUP BY     |       |   730 |  5840 |  4957   (7)| 00:00:01 |
|   2 |   TABLE ACCESS FULL| APPEL |  3500K|    26M|  4695   (2)| 00:00:01 |
-------------------------------------------------------------------------------
```

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 AS
  5  SELECT COUNT(*) AS NBApp, EXTRACT(MONTH FROM DateAppel) AS Mois, EXTRACT(YEAR
FROM DateAppel) AS Annee
  6  FROM Appel
  7  GROUP BY EXTRACT(MONTH FROM DateAppel), EXTRACT(YEAR FROM DateAppel);

Materialized view created.
```

12. Ré exécuter la requête R2. Examiner le temps et le plan d'exécution et comparer avec (10).

```
SQL> alter system flush shared_pool;

System altered.

SQL> alter system flush buffer_cache;

System altered.

SQL> SELECT COUNT(*) AS NBApp, EXTRACT(MONTH FROM DateAppel) AS Mois, EXTRACT(YEAR FROM
DateAppel) AS Annee
  2  FROM Appel
  3  GROUP BY EXTRACT(MONTH FROM DateAppel), EXTRACT(YEAR FROM DateAppel);

    NBAPP |       MOIS |      ANNEE
--------- | ---------- | ----------
   144096 |         12 |       2021
   134547 |          2 |       2021
   149213 |          3 |       2020
   143142 |          4 |       2020
   142856 |          9 |       2020
   148588 |          5 |       2021
   143519 |          4 |       2021
   144718 |         11 |       2021
   148469 |          7 |       2021
   148083 |          1 |       2021
   148418 |         10 |       2021
   148915 |          8 |       2021
   143695 |         11 |       2020
   148499 |          1 |       2020
```

```
    148706 |           7 |        2020
    144460 |           9 |        2021
    148931 |           8 |        2020
    144330 |           6 |        2021
    148820 |           3 |        2021
    147949 |           5 |        2020
    139364 |           2 |        2020
    148440 |          12 |        2020
    148667 |          10 |        2020
    143795 |           6 |        2020

24 rows selected.

Elapsed: 00:00:00.01

Execution Plan
----------------------------------------------------------
Plan hash value: 3157804869

-------------------------------------------------------------------------------------------
| Id  | Operation                     | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |      |    24 |   288 |     3   (0)| 00:00:01 |
|   1 |  MAT_VIEW REWRITE ACCESS FULL| VM4  |    24 |   288 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------
```

Temps d'exécution	Avant la création de vue	Après la création de vue
R2	00:00:01.32	00:00:00.01

- **Le temps d'exécution de la reqeuête "R2" après la création de la vue "VM4" est inférieur au temps de d'exécution avant la création de la vue.**

13. Augmenter le nombre d'instances de Appel à 4000000 , puis à 4500000 et retester la requête R2 avec et sans la vue matérialisée VM4 à chaque fois. (N'oubliez pas de rafraichir la vue après chaque augmentation du nombre d'instances : commande : Execute DBMS_MVIEW.Refersh('VM4') ;).

- **Augmenter le nombre d'instances de Appel à 4000000:**

```
SQL> DECLARE
  2       CodeApp number;
  3       Duree number;
  4       DateApp date;
  5       codeTA number;
  6       NumL number;
  7       CodeOD number;
  8  BEGIN
  9       FOR CodeApp IN 3500221 .. 4000000 LOOP
 10           SELECT floor(dbms_random.value(1, 60.9)) into Duree from dual;
 11           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;
 12           SELECT floor(dbms_random.value(1,2.9)) into codeTA from dual;
 13           SELECT floor(dbms_random.value(1,1500255.9)) into NumL from dual;
 14           SELECT floor(dbms_random.value(1,522.9)) into codeOD from dual;
 15           INSERT INTO Appel VALUES (CodeApp, Duree, DateApp, NumL, codeOD, CodeTA);
 16       END LOOP;
 17       COMMIT;
 18       END;
 19       /

PL/SQL procedure successfully completed.
```

- **Rafraichir la vue VM4 après la première augmentation et retester la requête R2:**

```
SQL> EXECUTE DBMS_MVIEW.REFRESH('VM4');

PL/SQL procedure successfully completed.

SQL> alter system flush shared_pool;

System altered.

SQL> alter system flush buffer_cache;

System altered.
```

```
SQL> alter system flush shared_pool;

System altered.

SQL> alter system flush buffer_cache;

System altered.

SQL> SELECT COUNT(*) AS NBApp, EXTRACT(MONTH FROM DateAppel) AS Mois, EXTRACT(YEAR FROM
DateAppel) AS Annee
  2  FROM Appel
  3  GROUP BY EXTRACT(MONTH FROM DateAppel), EXTRACT(YEAR FROM DateAppel);

    NBAPP |       MOIS |      ANNEE
--------- | ---------- | ----------
   164618 |         12 |       2021
   153646 |          2 |       2021
   170498 |          3 |       2020
   163886 |          4 |       2020
   163325 |          9 |       2020
   169724 |          5 |       2021
   164068 |          4 |       2021
   165190 |         11 |       2021
   169647 |          7 |       2021
   169248 |          1 |       2021
   169886 |         10 |       2021
   170102 |          8 |       2021
   164088 |         11 |       2020
   169735 |          1 |       2020
   169884 |          7 |       2020
   164885 |          9 |       2021
   170122 |          8 |       2020
   164782 |          6 |       2021
   170052 |          3 |       2021
   169482 |          5 |       2020
   159249 |          2 |       2020
   169657 |         12 |       2020
   169902 |         10 |       2020
   164324 |          6 |       2020
```

```
24 rows selected.

Elapsed: 00:00:00.00

Execution Plan
----------------------------------------------------------
Plan hash value: 3157804869

-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |      |    24 |   288 |     3   (0)| 00:00:01 |
|   1 |  MAT_VIEW REWRITE ACCESS FULL| VM4  |    24 |   288 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------
```

- **L'exécution de R2 après la suppression de la vue VM4:**

```
SQL> DROP MATERIALIZED VIEW VM4;

Materialized view dropped.

SQL> alter system flush shared_pool;

System altered.

SQL> alter system flush buffer_cache;

System altered.

-----------------------------------
SQL> SELECT COUNT(*) AS NBApp, EXTRACT(MONTH FROM DateAppel) AS Mois, EXTRACT(YEAR
FROM DateAppel) AS Annee
  2  FROM Appel
  3  GROUP BY EXTRACT(MONTH FROM DateAppel), EXTRACT(YEAR FROM DateAppel);

    NBAPP |       MOIS |      ANNEE
```

```
---------- | ---------- | ----------
    164618 |         12 |       2021
    153646 |          2 |       2021
    170498 |          3 |       2020
    163886 |          4 |       2020
    163325 |          9 |       2020
    169724 |          5 |       2021
    164068 |          4 |       2021
    165190 |         11 |       2021
    169647 |          7 |       2021
    169248 |          1 |       2021
    169886 |         10 |       2021
    170102 |          8 |       2021
    164088 |         11 |       2020
    169735 |          1 |       2020
    169884 |          7 |       2020
    164885 |          9 |       2021
    170122 |          8 |       2020
    164782 |          6 |       2021
    170052 |          3 |       2021
    169482 |          5 |       2020
    159249 |          2 |       2020
    169657 |         12 |       2020
    169902 |         10 |       2020
    164324 |          6 |       2020

24 rows selected.

Elapsed: 00:00:00.52

Execution Plan
----------------------------------------------------------
Plan hash value: 585468147

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

```
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |       |   730 |  5840 |  4957   (7)| 00:00:01 |
|   1 |  HASH GROUP BY     |       |   730 |  5840 |  4957   (7)| 00:00:01 |
|   2 |   TABLE ACCESS FULL| APPEL |  3500K|    26M|  4695   (2)| 00:00:01 |
-------------------------------------------------------------------------------
```

- **Recréation de la vue VM4:**

```
SQL> CREATE MATERIALIZED VIEW VM4
  2  BUILD IMMEDIATE
  3  REFRESH COMPLETE ON DEMAND
  4  ENABLE QUERY REWRITE AS
  5  SELECT COUNT(*) AS NBApp, EXTRACT(MONTH FROM DateAppel) AS Mois, EXTRACT(YEAR
FROM DateAppel) AS Annee
  6  FROM Appel
  7  GROUP BY EXTRACT(MONTH FROM DateAppel), EXTRACT(YEAR FROM DateAppel);

Materialized view created.
```

- **Deuxième augmentation:**

```
SQL> DECLARE
  2      CodeApp number;
  3      Duree number;
  4      DateApp date;
  5      codeTA number;
  6      NumL number;
  7      CodeOD number;
  8  BEGIN
  9      FOR CodeApp IN 4000001 .. 4500000 LOOP
 10          SELECT floor(dbms_random.value(1, 60.9)) into Duree from dual;
 11          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;
 12          SELECT floor(dbms_random.value(1,2.9)) into codeTA from dual;
 13          SELECT floor(dbms_random.value(1,1500255.9)) into NumL from dual;
 14          SELECT floor(dbms_random.value(1,522.9)) into codeOD from dual;
 15          INSERT INTO Appel VALUES (CodeApp, Duree, DateApp, NumL, codeOD, CodeTA);
 16      END LOOP;
 17      COMMIT;
 18      END;
```

```
 19      /

PL/SQL procedure successfully completed.
```

- **Retester la requête R2:**

```
SQL> alter system flush shared_pool;

System altered.

SQL> alter system flush buffer_cache;

System altered.

SQL> ------------------------------
SQL>
SQL> SELECT COUNT(*) AS NBApp, EXTRACT(MONTH FROM DateAppel) AS Mois, EXTRACT(YEAR FROM
DateAppel) AS Annee
  2  FROM Appel
  3  GROUP BY EXTRACT(MONTH FROM DateAppel), EXTRACT(YEAR FROM DateAppel);

    NBAPP |       MOIS |      ANNEE
--------- | ---------- | ----------
   185222 |         12 |       2021
   172593 |          2 |       2021
   191738 |          3 |       2020
   184444 |          4 |       2020
   183851 |          9 |       2020
   190762 |          5 |       2021
   184496 |          4 |       2021
   185756 |         11 |       2021
   190987 |          7 |       2021
   190545 |          1 |       2021
   191414 |         10 |       2021
   191169 |          8 |       2021
   184565 |         11 |       2020
   191111 |          1 |       2020
   191177 |          7 |       2020
   185482 |          9 |       2021
   191293 |          8 |       2020
```

```
    185329 |          6 |       2021
    191304 |          3 |       2021
    190593 |          5 |       2020
    179155 |          2 |       2020
    190986 |         12 |       2020
    191179 |         10 |       2020
    184849 |          6 |       2020

24 rows selected.

Elapsed: 00:00:00.22

Execution Plan
----------------------------------------------------------
Plan hash value: 3157804869

-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |      |    24 |   288 |     3   (0)| 00:00:01 |
|   1 |  MAT_VIEW REWRITE ACCESS FULL| VM4  |    24 |   288 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------
```

- **Après la suppression de la vue VM4:**

```
SQL> DROP MATERIALIZED VIEW VM4;

Materialized view dropped.

SQL> alter system flush shared_pool;

System altered.
SQL> alter system flush buffer_cache;

System altered.

SQL> --------------------------------
SQL> SELECT COUNT(*) AS NBApp, EXTRACT(MONTH FROM DateAppel) AS Mois, EXTRACT(YEAR
FROM DateAppel) AS Annee
```

```
  2  FROM Appel
  3  GROUP BY EXTRACT(MONTH FROM DateAppel), EXTRACT(YEAR FROM DateAppel);

    NBAPP |       MOIS |      ANNEE
--------- | ---------- | ----------
   185222 |         12 |       2021
   172593 |          2 |       2021
   191738 |          3 |       2020
   184444 |          4 |       2020
   183851 |          9 |       2020
   190762 |          5 |       2021
   184496 |          4 |       2021
   185756 |         11 |       2021
   190987 |          7 |       2021
   190545 |          1 |       2021
   191414 |         10 |       2021
   191169 |          8 |       2021
   184565 |         11 |       2020
   191111 |          1 |       2020
   191177 |          7 |       2020
   185482 |          9 |       2021
   191293 |          8 |       2020
   185329 |          6 |       2021
   191304 |          3 |       2021
   190593 |          5 |       2020
   179155 |          2 |       2020
   190986 |         12 |       2020
   191179 |         10 |       2020
   184849 |          6 |       2020

24 rows selected.

Elapsed: 00:00:02.39

Execution Plan
----------------------------------------------------------
```

```
Plan hash value: 585468147

-------------------------------------------------------------------------------
| Id  | Operation          | Name  | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |       |   730 |  5840 |  4957   (7)| 00:00:01 |
|   1 |  HASH GROUP BY     |       |   730 |  5840 |  4957   (7)| 00:00:01 |
|   2 |   TABLE ACCESS FULL| APPEL |  3500K|    26M|  4695   (2)| 00:00:01 |
```

14. Donner un tableau comparatif des temps d'exécution (avec et sans la vue matérialisée) en fonction du nombre d'instances.

Nombre d'instances	Sans la vue	Avec la vue
3500220	**00:00:01.32**	**00:00:00.01**
4000000	**00:00:52**	**00:00:00**
4500000	**00:00:02.39**	**00:00:00.22**

15. Quelles Conclusions tirez-vous de ce TP?

- **Une vue matérialisée est un objet de base de données qui contient les résultats d'une requête qui permet de conserver des copies de données distantes sur la mémoire. Ces copiers peuvent être mises à jour.**
- **Les vues matérialisées sert à optimiser le temps d'execution des requêtes qui affichent les mêmes valeurs contenant dans la vue.**