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 TP05 ENDO

Travail présenté à Monsieur Selmoune Nazih

Travail présenté par :
AISSANI Anouar
161835024828

2021/2022

1. Ecrire une requête R1 qui donne le nombre de clients de sexe masculin.

```
SQL> SELECT COUNT(*) FROM DClient WHERE SexeClient = 'H';

  COUNT(*)
----------
    504343

Elapsed: 00:00:00.08

Execution Plan
----------------------------------------------------------
Plan hash value: 3011024653

--------------------------------------------------------------------------------
| Id  | Operation          | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |         |     1 |     2 |  2105   (2)| 00:00:01 |
|   1 |  SORT AGGREGATE    |         |     1 |     2 |            |          |
|*  2 |   TABLE ACCESS FULL| DCLIENT |   532K|  1040K|  2105   (2)| 00:00:01 |
--------------------------------------------------------------------------------

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

   2 - filter("SEXECLIENT"='H')
```

2. Créer un index b-arbre de la table DClient sur l'attribut SexeClient

```
SQL> CREATE INDEX SexeClient_index
  2  ON DClient (SexeClient);

Index created.
```

3. Réexécuter R1 en examinant le temps et le plan de l'exécution.

```
SQL> SELECT COUNT(*) FROM DClient WHERE SexeClient = 'H';

  COUNT(*)
----------
    504343

Elapsed: 00:00:00.11

Execution Plan
----------------------------------------------------------
Plan hash value: 2099928318

-------------------------------------------------------------------------------------------
| Id  | Operation             | Name             | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |                  |     1 |     2 |   540   (3)| 00:00:01 |
|   1 |  SORT AGGREGATE       |                  |     1 |     2 |            |          |
|*  2 |   INDEX FAST FULL SCAN| SEXECLIENT_INDEX |   532K|  1040K|   540   (3)| 00:00:01 |
-------------------------------------------------------------------------------------------

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

   2 - filter("SEXECLIENT"='H')
```

4. Supprimer l'index b-arbre, et créer un index bitmap de la même table et sur le même attribut.

```
-- Supprimer l'index b-arbre
SQL> DROP INDEX SexeClient_index;

Index dropped.

-- créer un index bitmap
SQL> CREATE BITMAP INDEX SexeClient_index
  2  ON DClient (SexeClient);

Index created.
```

5. Réexécuter R1 et comparez entre les trois exécutions.

```
SQL> alter system flush shared_pool;

System altered.

SQL> alter system flush buffer_cache;

System altered.

SQL> SELECT COUNT(*) FROM DClient WHERE SexeClient = 'H';

  COUNT(*)
----------
    504343

Elapsed: 00:00:00.06

Execution Plan
----------------------------------------------------------
Plan hash value: 1026537208

-------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name             | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                  |     1 |     2 |    24   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE              |                  |     1 |     2 |            |          |
|   2 |   BITMAP CONVERSION COUNT    |                  |   532K|  1040K|    24   (0)| 00:00:01 |
|*  3 |    BITMAP INDEX SINGLE VALUE| SEXECLIENT_INDEX |       |       |            |          |
-------------------------------------------------------------------------------------------------

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

   3 - access("SEXECLIENT"='H')
```

- comparez entre les trois exécutions :

	Exécution 01	Exécution 02	Exécution 03
Temps d'exécution	00:00:00.08	00:00:00.11	00:00:00.06

- On remarque qu'après la création du l'index **b-arbre** le temps d'exécution n'a pas été optimisé mais il n'a pas été optimisé après la création du l'index **bimap.**
- Le plan d'exécution contient l'opération d'index après la création de l'index.

6. Supprimer le bitmap index.

```
SQL> DROP INDEX SexeClient_index;

Index dropped.
```

7. Ecrire une requête R2 qui donne le nombre d'appel global vers le destinataire Mobilis.

```
SQL> SELECT SUM(a.NBAppels)
  2  FROM FAppel a, DDestinataire d
  3  WHERE a.CodeOperateurDestinataire = d.CodeOperateurDestinataire
  4  AND d.NomOperateurDestinataire = 'Mobilis';

SUM(A.NBAPPELS)
---------------
           8364

Elapsed: 00:00:00.50

Execution Plan
----------------------------------------------------------
Plan hash value: 1158584483

-------------------------------------------------------------------------------------------------------
| Id  | Operation            | Name          | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |               |     1 |    22 |  5356   (3)| 00:00:01 |       |       |
|   1 |  SORT AGGREGATE      |               |     1 |    22 |            |          |       |       |
|*  2 |   HASH JOIN          |               |  8621 |   185K|  5356   (3)| 00:00:01 |       |       |
|*  3 |    TABLE ACCESS FULL | DDESTINATAIRE |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|   4 |    PARTITION LIST ALL|               |  4499K|    30M|  5319   (2)| 00:00:01 |     1 |     4 |
|   5 |     TABLE ACCESS FULL| FAPPEL        |  4499K|    30M|  5319   (2)| 00:00:01 |     1 |     4 |
-------------------------------------------------------------------------------------------------------

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

   2 - access("A"."CODEOPERATEURDESTINATAIRE"="D"."CODEOPERATEURDESTINATAIRE")
   3 - filter("D"."NOMOPERATEURDESTINATAIRE"='Mobilis')
```

8. Créer un index bitmap de jointure entre FAppel et DDestinataire, basé sur l'attribut 'NomOperateurDestinataire'.

```
SQL> CREATE BITMAP INDEX NomOpDest_index
  2  ON FAppel(DDestinataire.NomOperateurDestinataire)
  3  FROM DDestinataire, FAppel
  4  WHERE FAppel.CodeOperateurDestinataire = DDestinataire.CodeOperateurDestinataire;

Index created.
```

9. Réexécuter R2 et comparez entre les deux exécutions.

```
SQL> alter system flush shared_pool;

System altered.

SQL> alter system flush buffer_cache;

System altered.

SQL> SELECT SUM(a.NBAppels)
  2  FROM FAppel a, DDestinataire d
  3  WHERE a.CodeOperateurDestinataire = d.CodeOperateurDestinataire
  4  AND d.NomOperateurDestinataire = 'Mobilis';

SUM(A.NBAPPELS)
--------------
          8364

Elapsed: 00:00:00.20

Execution Plan
----------------------------------------------------------
Plan hash value: 2493943967

-------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name            | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                 |     1 |     4 |     5   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE              |                 |     1 |     4 |            |          |
|   2 |   BITMAP CONVERSION SUM      |                 |  8621 | 34484 |     5   (0)| 00:00:01 |
|*  3 |    BITMAP INDEX SINGLE VALUE| NOMOPDEST_INDEX |       |       |            |          |
-------------------------------------------------------------------------------------------------
```

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

   3 - access("A"."SYS_NC00008$"='Mobilis')
```

- comparez entre les deux exécutions :

	Exécution 01	Exécution 02
Temps d'exécution	00:00:00.50	00:00:00.20

- One remarque que le temps d'exécution a été optimisé après l'ajout de l'index bitmap.
- Le plan d'exécution après la création d'index contient l'opération de l'index bitmap.

10. Ecrire une requête R3 qui donne le nombre d'appels de type 'International'.

```
SQL> SELECT SUM(a.NBAppels) AS nbr_appel_international
  2  FROM FAppel a, DTypeAppel t
  3  WHERE a.CodeTypeAppel = t.CodeTypeAppel
  4  AND t.TypeAppel = 'Internationale';

NBR_APPEL_INTERNATIONAL
-----------------------
                2131593

Elapsed: 00:00:00.52

Execution Plan
----------------------------------------------------------
Plan hash value: 3751911244

-------------------------------------------------------------------------------------
| Id  | Operation           | Name       | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |            |     1 |    19 |  5336   (3)| 00:00:01 |
|   1 |  SORT AGGREGATE     |            |     1 |    19 |            |          |
|*  2 |   HASH JOIN         |            |  2249K|    40M|  5336   (3)| 00:00:01 |
|*  3 |    TABLE ACCESS FULL| DTYPEAPPEL |     1 |    16 |     3   (0)| 00:00:01 |
|   4 |    TABLE ACCESS FULL| FAPPEL     |  4499K|    12M|  5298   (2)| 00:00:01 |
-------------------------------------------------------------------------------------
```

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

   2 - access("A"."CODETYPEAPPEL"="T"."CODETYPEAPPEL")
   3 - filter("T"."TYPEAPPEL"='Internationale')
```

11. Créer un index bitmap de jointure qui améliore le temps de cette requête.

```
SQL> CREATE BITMAP INDEX NbrAppInter_index
  2  ON FAppel(DTypeAppel.TypeAppel)
  3  FROM DTypeAppel, FAppel
  4  WHERE FAppel.CodeTypeAppel = DTypeAppel.CodeTypeAppel;

Index created.
```

12. Réexécuter R3 et comparer les deux exécutions.

```
SQL> alter system flush shared_pool;

System altered.

SQL> alter system flush buffer_cache;

System altered.

SQL> SELECT SUM(a.NBAppels) AS nbr_appel_international
  2  FROM FAppel a, DTypeAppel t
  3  WHERE a.CodeTypeAppel = t.CodeTypeAppel
  4  AND t.TypeAppel = 'Internationale';

NBR_APPEL_INTERNATIONAL
-----------------------
                2131593

Elapsed: 00:00:00.26

Execution Plan
----------------------------------------------------------
Plan hash value: 3530674921
```

```
-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name              | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                   |     1 |     3 |    95   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE              |                   |     1 |     3 |            |          |
|   2 |   BITMAP CONVERSION SUM      |                   |  2249K|  6591K|    95   (0)| 00:00:01 |
|*  3 |    BITMAP INDEX SINGLE VALUE| NBRAPPINTER_INDEX |       |       |            |          |
-------------------------------------------------------------------------------------------

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

   3 - access("A"."SYS_NC00009$"='Internationale')
```

- comparer les deux exécutions :

	Exécution 01	Exécution 02
Temps d'exécution	00:00:00.52	00:00:00.26

- On remarque que le temps d'exécution a été optimisé après l'ajout de l'index bitmap de jointure.
- Le plan d'exécution après la création d'index contient l'opération de l'index bitmap.

13. Créer une table FAppel2 identique à Fappel, en la partitionnant en fonction des codes de type de lignes comme suit : P1{1,3,6}, P2{2,7,8}, P3{4,5}, P4{9,10}.

```
SQL> CREATE TABLE FAppel2 (
  2      CodeClient NUMBER(10),
  3      CodeTypeLigne NUMBER(10),
  4      CodeTypeAppel NUMBER(10),
  5      CodeOperateurDestinataire NUMBER(10),
  6      CodeTemps NUMBER(10),
  7      NBAppels NUMBER,
  8      Duree NUMBER,
  9      CONSTRAINT fk_DClient2 FOREIGN KEY (CodeClient) REFERENCES DClient (CodeClient),
 10      CONSTRAINT fk_DTypeLigne2 FOREIGN KEY (CodeTypeLigne) REFERENCES DTypeLigne
         (CodeTypeLigne),
 11      CONSTRAINT fk_DTypeAppel2 FOREIGN KEY (CodeTypeAppel) REFERENCES DTypeAppel
         (CodeTypeAppel),
 12      CONSTRAINT fk_DDestinataire2 FOREIGN KEY (CodeOperateurDestinataire) REFERENCES
         DDestinataire (CodeOperateurDestinataire),
 13      CONSTRAINT fk_DTemps2 FOREIGN KEY (CodeTemps) REFERENCES DTemps (CodeTemps),
 14      CONSTRAINT pk_FAppel2 PRIMARY KEY (CodeClient, CodeTypeLigne, CodeTypeAppel,
```

```
         CodeOperateurDestinataire, CodeTemps)
 15  )

 16  PARTITION BY LIST (CodeTypeLigne)
 17      (PARTITION partition1 VALUES (1, 3, 6),
 18      PARTITION partition2 VALUES (2, 7, 8),
 19      PARTITION partition3 VALUES (4, 5),
 20      PARTITION partition4 VALUES (9, 10)
 21  );

Table created.
```

14. Remplir FAppel2 les avec les mêmes instances que FAppel.

```
SQL> begin
  2  for i in
  3  ( SELECT DISTINCT c.NumClient , l.CodeTypeLigne, a.CodeTypeAppel,
a.CodeOperateurDstinataire, t.CodeTemps,
  4  count(*) as NBAppels, SUM(a.DureeAppel) as Duree
  5  FROM master.Client c, master.Ligne l,master.Appel a, DTemps t
  6  WHERE  c.NumClient = l.NumClient
  7  AND l.NumeroLigne = a.NumeroLigne
  8  AND t.Jour = TO_CHAR(a.DateAppel,'DD/MM/YYYY')
  9  GROUP BY c.NumClient, l.CodeTypeLigne, a.CodeTypeAppel, a.CodeOperateurDstinataire,
t.CodeTemps)
 10  LOOP
 11  insert into FAppel2 values(i.NumClient, i.CodeTypeLigne, i.CodeTypeAppel,
i.CodeOperateurDstinataire, i.CodeTemps, i.NBAppels, i.Duree);
 12  end loop;
 13  commit ;
 14  end ;
 15  /

PL/SQL procedure successfully completed.
```

15. Ecrire une requête R4 qui donne le nombre d'appels global des lignes de type N°8 en utilisant la table FAppel.

```
SQL> SELECT SUM(NBAppels) FROM FAppel WHERE CodeTypeLigne = 8;

SUM(NBAPPELS)
-------------
       455003

Elapsed: 00:00:00.36

Execution Plan
----------------------------------------------------------
Plan hash value: 850936457

-------------------------------------------------------------------------------
| Id  | Operation          | Name   | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |        |     1 |     3 |  5309   (2)| 00:00:01 |
|   1 |  SORT AGGREGATE    |        |     1 |     3 |            |          |
|*  2 |   TABLE ACCESS FULL| FAPPEL |   449K|  1318K|  5309   (2)| 00:00:01 |
-------------------------------------------------------------------------------

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

   2 - filter("CODETYPELIGNE"=8)
```

16. Modifier R4 pour utiliser la table FAppel2 et comparez les deux exécutions.

```
SQL> alter system flush shared_pool;

System altered.

Elapsed: 00:00:00.16
SQL> alter system flush buffer_cache;

System altered.

Elapsed: 00:00:00.06
SQL> --
```

```
SQL> SELECT SUM(NBAppels) FROM FAppel WHERE CodeTypeLigne = 8;


SUM(NBAppels)
-------------
       455002

Elapsed: 00:00:00.10

Execution Plan
----------------------------------------------------------
Plan hash value: 315761597

-------------------------------------------------------------------------------------------------
| Id  | Operation             | Name    | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |         |     1 |    13 |  2993  (46)| 00:00:01 |       |       |
|   1 |  SORT AGGREGATE       |         |     1 |    13 |            |          |       |       |
|   2 |   PARTITION LIST SINGLE|         |   447K|  5682K|  2993  (46)| 00:00:01 |   KEY |   KEY |
|*  3 |    TABLE ACCESS FULL  | FAPPEL2 |   447K|  5682K|  2993  (46)| 00:00:01 |     2 |     2 |
-------------------------------------------------------------------------------------------------

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

   3 - filter("CODETYPELIGNE"=8)

Note
-----
   - dynamic statistics used: dynamic sampling (level=2)
```

Comparez les deux exécutions :

	Exécution 01	Exécution 02
Temps d'exécution	00:00:00.36	00:00:00.10

- On remarque que le temps d'exécution est plus petit quand on utilise la table FAppel2
- Le plan d'exécution de la deuxième exécution contient la partition qu'on a créé.

17. Y-a-t-il une solution pour partitionner une table existante, si oui l'appliquer pour partitionner vente (selon le même critère que FAppel2).

Oui, il y a une solution pour partitionner une table existante.

```
--1) On supprime les index existant
SQL> drop index NOMOPDEST_INDEX;

Index dropped.

SQL> drop index NBRAPPINTER_INDEX;

Index dropped.

--2) On crée un index sur l'attribut "CodeTypeLigne"
SQL> create index CodeTypeLigne_index on FAppel(CodeTypeLigne);

Index created.

--3) On modifie la table FAppel
SQL> Alter TABLE FAppel MODIFY PARTITION BY LIST (CodeTypeLigne)
  2      (PARTITION partition1 VALUES (1, 3, 6),
  3      PARTITION partition2 VALUES (2, 7, 8),
  4      PARTITION partition3 VALUES (4, 5),
  5      PARTITION partition4 VALUES (9, 10)
  6  ) ONLINE;

Table altered.
```