

TP 5 ENDO

NOM : ACHACHI

Prénom : Elhadj Ali

Mat : 181832008538

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

```
SQL> select count(*) from dpcclient where SEXECLIENT = 'M';
```

COUNT(*)
562003

Ecoulé : 00 :02 :07.26

Plan d'exécution

Plan hash value: 3114477839

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	2	1724 (4)	00:00:21
1	SORT AGGREGATE		1	2		
* 2	TABLE ACCESS FULL	DPCCLIENT	530K	1035K	1724 (4)	00:00:21

Predicate Information (identified by operation id):

2 - filter("SEXECLIENT"='M')

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

```
SQL> CREATE INDEX IBA_dpcclient_w  
2 ON dpcclient (SEXECLIENT);
```

Index créé.

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

```
SQL> select count(*) from dpcclient where SEXECLIENT = 'M';
```

COUNT(*)
562003

Exécution : 00 :00 :04.81

Plan d'exécution

Plan hash value: 632737982

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	2	459 (8)	00:00:06
1	SORT AGGREGATE		1	2		
* 2	INDEX FAST FULL SCAN	IBA_DPCCLIENT_W	530K	1035K	459 (8)	00:00:06

Predicate Information (identified by operation id):

2 - filter("SEXECLIENT"='M')

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

```
SQL> DROP INDEX IBA_dpcclient_w;
```

Index supprimé.

```
SQL> CREATE BITMAP INDEX IBM_dpcclient_w  
2 ON dpcclient (SEXECLIENT);
```

Index créé.

- 5) Réexécuter R1 et comparez entre les trois exécutions. :

```
SQL> select count(*) from dpcclient where SEXECLIENT = 'M';
```

COUNT(*)
562003

Execution time : 00 :00 :00.93

Plan d'exécution

Plan hash value: 3954610813

Id	Operation	Name	Rows	Bytes	Cost (%C)
0	SELECT STATEMENT		1	2	24
1	SORT AGGREGATE		1	2	
2	BITMAP CONVERSION COUNT		530K	1035K	24
3	BITMAP INDEX SINGLE VALUE	IBM_DPCCLIENT_W			

Predicate Information (identified by operation id):

3 - access("SEXECLIENT"='M')

6) Supprimer les deux index :

```
SQL> DROP INDEX IBM_dpcclient_w;
```

Index supprimé.

7) Ecrire une requête R2 qui donne le nombre d'appel global vers le destinataire KRLXWFN :

```
SQL> select count(A.NBAppels) AS Nombres_Appels from FAppel A, DDestinataire D
  2  where NomOperateurDestinataire = 'KRLXCWFN'
  3  and A.CodeOperateurDestinataire = D.CodeOperateurDestinataire;

NOMBRES_APPELS
-----
          6015

Ecoulu : 00 :01 :57.10

Plan d'exécution
-----
Plan hash value: 3033830088

-----
-----
| Id | Operation          | Name          | Rows  | Bytes | Cost (%CPU)| Time |
|----|-----|-----|-----|-----|-----|-----|
-----
|  0 | SELECT STATEMENT    |               |      1 |    19 |    3377  (7)| 00:00:41 |
|  1 |   SORT AGGREGATE    |               |      1 |    19 |           |      |
|*  2 |    HASH JOIN        |               |   6680 |  123K |    3377  (7)| 00:00:41 |
|*  3 |     TABLE ACCESS FULL| DDESTINATAIRE |        1 |     12 |          2  (0)| 00:00:01 |
|  4 |     TABLE ACCESS FULL| FAPPEL        |   3493K |   23M |    3313  (6)| 00:00:40 |
-----
:

```

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

```
SQL> CREATE BITMAP INDEX IBM_FApp_Ddes_libj
  2  ON FAppel(D.NOMOPERATEURDESTINATAIRE)
  3  FROM FAppel A, DDestinataire D
  4  WHERE D.CODEOPERATEURDESTINATAIRE = A.CODEOPERATEURDESTINATAIRE;

Index créé.
```

- 9) Réexécuter R2 et comparez entre les deux exécutions :

```

SQL> select count(A.NBAppels) AS Nombres_Appels from FAppel A, DDestinataire D
  2  where NomOperateurDestinataire = 'KRLXCWFN'
  3  and A.CodeOperateurDestinataire = D.CodeOperateurDestinataire;

NOMBRES_APPELS
-----
          6015

Ecoulu : 00 :00 :04.97

Plan d'exécution
-----
Plan hash value: 3707432149

-----
-----
| Id | Operation | Name | Rows | Bytes | Cost |
|---|---|---|---|---|---|
| 0 | SELECT STATEMENT | | 1 | 7 | 11 |
| 1 | SORT AGGREGATE | | 1 | 7 | |
| 2 | TABLE ACCESS BY INDEX ROWID | FAPPEL | 6693 | 46851 | 11 |
| 3 | BITMAP CONVERSION TO ROWIDS | | | | |
|* 4 | BITMAP INDEX SINGLE VALUE | IBM_FAPP_DDES_LIBJ | | | |
-----
-----

```

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

```
SQL> select count(A.NBAppels) from FAppel A , DTypeAppel T
  2  where A.CODETYPEAPPEL = T.CODETYPEAPPEL
  3  and T.TYPEAPPEL = 'internationale';

COUNT(A.NBAPPELS)
-----
          1657794

Ecoulu : 00 :00 :41.71

Plan d'exécution
-----
Plan hash value: 3751911244

-----
--
| Id | Operation          | Name          | Rows  | Bytes | Cost (%CPU)| Time
|-----|-----|-----|-----|-----|-----|-----|
| 0 | SELECT STATEMENT    |               |      1 |      21 | 3377 (7)| 00:00:41
| 1 | SORT AGGREGATE      |               |      1 |      21 |           |
|* 2 | HASH JOIN           |               | 1746K | 34M | 3377 (7)| 00:00:41
|* 3 | TABLE ACCESS FULL | DTYPEAPPEL   |      1 |      15 | 2 (0)| 00:00:01
| 4 | TABLE ACCESS FULL | FAPPEL       | 3493K | 19M | 3313 (6)| 00:00:40
|-----|-----|-----|-----|-----|-----|
```

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

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

Index créé.
```

12) Réexécuter R3 et comparer les deux exécutions :

```

SQL> select count(A.NBAppels) from Fappel A , DTypeAppel T
  2  where T.TypeAppel = 'internationale'
  3  and A.CodeTypeAppel = T.CodeTypeAppel;

COUNT(A.NBAPPELS)
-----
          1657794

Ecoulu : 00 :00 :03.37

Plan d'exécution
-----
Plan hash value: 3751911244

-----
--
| Id | Operation          | Name          | Rows  | Bytes | Cost (%CPU)| Time |
|----|-----|-----|-----|-----|-----|-----|-----|
| 0  | SELECT STATEMENT   |               |      1 |      21 | 3377  (7)| 00:00:41 |
| 1  | SORT AGGREGATE     |               |      1 |      21 |           |      |
|*  2  | HASH JOIN          |               | 1746K | 34M   | 3377  (7)| 00:00:41 |
|*  3  | TABLE ACCESS FULL| DTYPEAPPEL   |      1 |      15 | 2  (0)| 00:00:01 |
|  4  | TABLE ACCESS FULL| FAPPEL       | 3493K | 19M   | 3313  (6)| 00:00:40 |
-----

```

- 13) Créer une table Fappel2 identique à Fappel, en la partitionnant en fonction des code 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 int,
  3   CodeTypeLigne int,
  4   CodeTypeAppel int,
  5   CodeOperateurDestinataire int,
  6   CodeTemps NUMBER(6),
  7   NBAppels int,
  8   Duree int,
  9   CONSTRAINT uk_Fappel2 PRIMARY KEY (CodeClient, CodeTypeLigne, CodeTypeAppel, CodeOperateurDestinataire, CodeTemps),
 10   CONSTRAINT FK_FH_DC2 FOREIGN KEY (CodeClient) REFERENCES DPCCClient,
 11   CONSTRAINT FK_FH_DL2 FOREIGN KEY (CodeTypeLigne) REFERENCES DTypeLigne ,
 12   CONSTRAINT FK_FH_DD2 FOREIGN KEY (CodeOperateurDestinataire) REFERENCES DDestinataire ,
 13   CONSTRAINT FK_FH_DT2 FOREIGN KEY (CodeTypeAppel) REFERENCES DTypeAppel,
 14   CONSTRAINT FK_FH_DCT2 FOREIGN KEY ( CODETEMPS ) REFERENCES DTemps )
 15   PARTITION BY LIST (CodeTypeLigne)
 16   (PARTITION P1 VALUES (1,3,6),
 17    PARTITION P2 VALUES (2,7,8),
 18    PARTITION P3 VALUES (4,5),
 19    PARTITION P4 VALUES (9,10)
 20  );

Table créée.

Ecoulu : 00 :00 :02.22

```


14) Remplir FAppel2 les avec les mêmes instances que FAppel :

```
SQL> BEGIN
2 FOR i IN
3 (SELECT L.numclient, L.codetp, A.codeta, A.codedo, D.CodeTemps, count(A.CodeAppel) as NBAppel, sum(A.Duree) as Dure
4 FROM MSETR.ligne L, DTemps D, MSETR.Appel A
5 WHERE D.Jour = A.DateApp
6 AND L.numligne = A.numligne
7 GROUP BY numclient, codetp, codeta, codedo, CodeTemps
8 )
9 LOOP
10 INSERT INTO FAppel2 VALUES (i.numclient, i.codetp, i.codeta, i.codedo, i.CodeTemps, i.NBAppel, i.Dure) log errors into err$_FAppel reject limit unlimited;
11 END LOOP;
12 COMMIT;
13 END;
14 /

Procédure PL/SQL terminée avec succès.

Ecoulu : 00 :19 :49.33
```

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 count(A.codetypeappel) as Nombre_appel_globale
2 from FAppel A, DTypeLigne T
3 where A.CodeTypeLigne = T.CodeTypeLigne
4 And T.CodeTypeLigne = 8;
```

NOMBRE_APPEL_GLOBALE

0

Ecoulu : 00 :00 :58.00

Plan d'exécution

Plan hash value: 4281336041

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	6	3294 (5)	00:00:40
1	SORT AGGREGATE		1	6		
2	NESTED LOOPS		1	6	3294 (5)	00:00:40
* 3	INDEX UNIQUE SCAN	PK_DTPELIGNE	1	3	0 (0)	00:00:01
* 4	TABLE ACCESS FULL	FAPPEL	1	3	3294 (5)	00:00:40

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

```
SQL> select count(A.codetypeappel) as Nombre_appel_globale
  2   from FAppel2 A , DTypeLigne T
  3   where A.CodeTypeLigne = T.CodeTypeLigne
  4   And T.CodeTypeLigne = 8;

NOMBRE_APPEL_GLOBALE
-----
                      0

Ecoulu : 00 :00 :07.84
```

17) Y-a-t-il une solution pour partitionner une table existante : oui
appliquer pour partitionner vente (selon le même critère que FAppel2)

après la suppression de la table fappel :

```
SQL> CREATE TABLE Fappel (
  2   CodeClient int,
  3   CodeTypeLigne int,
  4   CodeTypeAppel int,
  5   CodeOperateurDestinataire int,
  6   CodeTemps NUMBER(6),
  7   NBAppels int,
  8   Duree int,
  9   CONSTRAINT qK_Fappe PRIMARY KEY (CodeClient, CodeTypeLigne, CodeTypeAppel, CodeOperateurDestinataire, CodeTemps),
10   CONSTRAINT dK_FH_D FOREIGN KEY (CodeClient) REFERENCES DPClient,
11   CONSTRAINT sK_FH_D FOREIGN KEY (CodeTypeLigne) REFERENCES DTypeLigne ,
12   CONSTRAINT wK_FH_D FOREIGN KEY (CodeOperateurDestinataire) REFERENCES DDestinataire ,
13   CONSTRAINT K_FH_D FOREIGN KEY (CodeTypeAppel) REFERENCES DTypeAppel,
14   CONSTRAINT pK_FH_DC FOREIGN KEY ( CODETEMPS ) REFERENCES DTemps )
15   PARTITION BY LIST (CodeTypeLigne)
16   (PARTITION P1 VALUES (1,3,6),
17    PARTITION P2 VALUES (2,7,8),
18    PARTITION P3 VALUES (4,5),
19    PARTITION P4 VALUES (9,10)
20  );

Table créée.
```