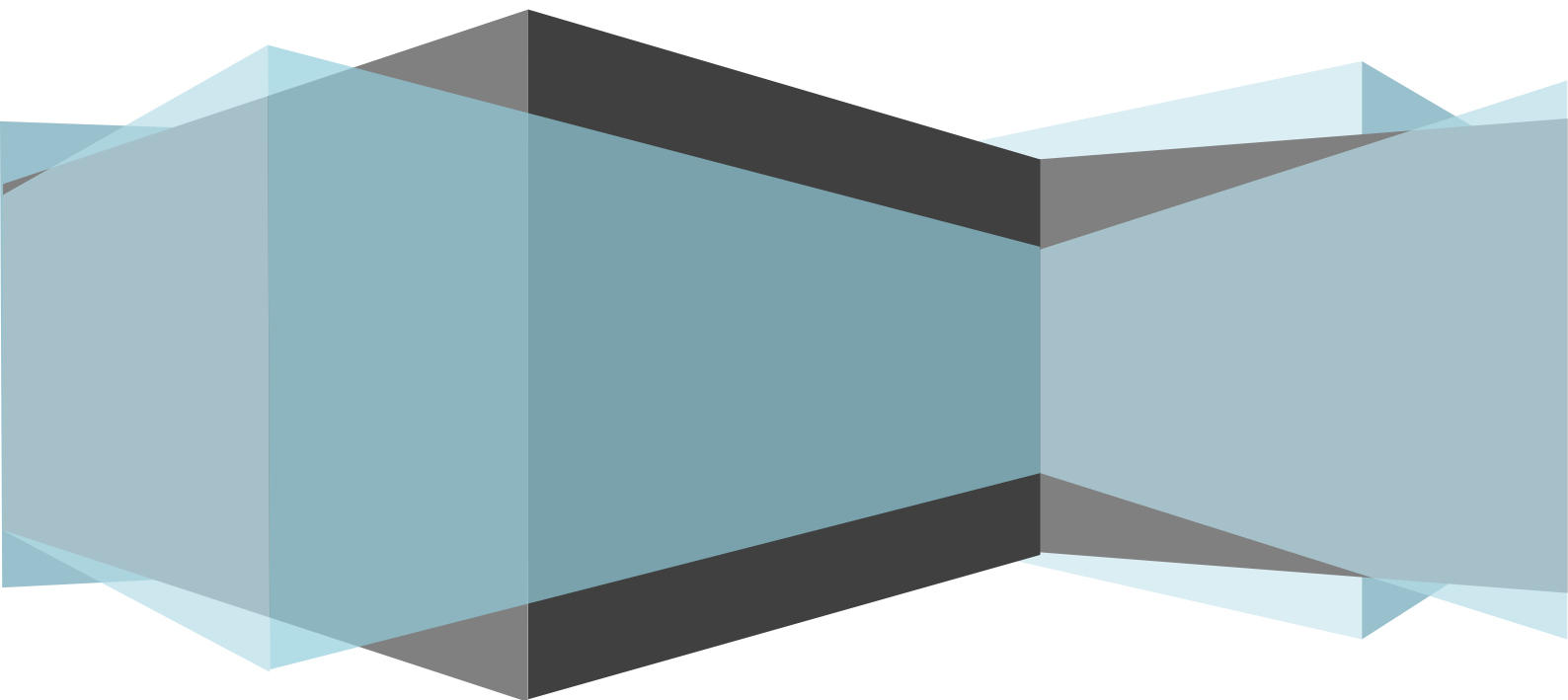


TD – Optimisation de requêtes

03/12/2012

Polizzi Florent



I. Les plans d'exécution d'Oracle – Requêtes simples

1. Donner le nombre de clients

- **Requête SQL** : `select count(*) FROM client;`
- **EXPLAIN PLAN** :

Id	Operation	Name	Rows	Cost (%CPU)	Time
0	SELECT STATEMENT		1	53 (0)	00:00:01
1	SORT AGGREGATE		1		
2	INDEX FAST FULL SCAN	SYS_C007136	116K	53 (0)	00:00:01

- Oracle a parcouru entièrement la table à travers l'INDEX (INDEX FAST FULL SCAN), a appliqué la fonction d'agrégation COUNT et a renvoyé le résultat.

2. Calculer l'âge moyen des clients

- **Requête SQL** : `select AVG(age) from client;`
- **EXPLAIN PLAN** :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	13	300 (1)	00:00:04
1	SORT AGGREGATE		1	13		
2	TABLE ACCESS FULL	CLIENT	116K	1480K	300 (1)	00:00:04

- Oracle a parcouru entièrement la table (TABLE ACCESS FULL) a appliqué la fonction d'agrégation AVG et a renvoyé le résultat.

3. Donner le nombre de clients par ville

- **Requête SQL** : select ville,COUNT(*) FROM client group BY ville;
- **EXPLAIN PLAN** :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		116K	3074K	305 (3)	00:00:04
1	HASH GROUP BY		116K	3074K	305 (3)	00:00:04
2	TABLE ACCESS FULL	CLIENT	116K	3074K	299 (1)	00:00:04

- Oracle a parcouru entièrement la table (TABLE ACCESS FULL), puis a fait le GROUP BY et a renvoyé le résultat.

4. Calculer l'âge moyen des clients par ville

- **Requête SQL** : select ville,AVG(age) FROM client group BY ville;
- **EXPLAIN PLAN** :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		116K	4554K	305 (3)	00:00:04
1	HASH GROUP BY		116K	4554K	305 (3)	00:00:04
2	TABLE ACCESS FULL	CLIENT	116K	4554K	300 (1)	00:00:04

- Oracle a parcouru entièrement la table (TABLE ACCESS FULL), puis a fait le GROUP BY et a renvoyé le résultat.

5. Donner le nombre de clients par ville qui sont des femmes

- **Requête SQL :** select ville,COUNT(*) FROM client WHERE genre='F' group BY ville;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		43286	1268K	303 (2)	00:00:04
1	HASH GROUP BY		43286	1268K	303 (2)	00:00:04
* 2	TABLE ACCESS FULL	CLIENT	43286	1268K	301 (1)	00:00:04

Predicate Information (identified by operation id):

2 - filter("GENRE"='F')

- Oracle a parcouru entièrement la table (TABLE ACCESS FULL) déjà filtré (2 - filter("GENRE"='F')), puis a fait le GROUP BY et a renvoyé le résultat. (c'est pour ça qu'oracle a travaillé avec seulement une table de 43286 lignes)

6. Donner le nombre de clients par ville qui ne sont pas des hommes

- **Requête SQL :** select ville,COUNT(*) FROM client WHERE genre<>'M' group BY ville;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		43286	1268K	303 (2)	00:00:04
1	HASH GROUP BY		43286	1268K	303 (2)	00:00:04
* 2	TABLE ACCESS FULL	CLIENT	43286	1268K	301 (1)	00:00:04

Predicate Information (identified by operation id):

2 - filter("GENRE"<>'M')

- Oracle a parcouru entièrement la table (TABLE ACCESS FULL) déjà filtré (2 - filter("GENRE"<>'M')), puis a fait le GROUP BY et a renvoyé

le résultat. (c'est pour ça qu'oracle a travaillé avec seulement une table de 43286 lignes)

7. Donner le nombre de clients par ville qui sont des femmes mais uniquement si la ville compte plus de 3000 femmes clientes

- **Requête SQL :** select ville,COUNT(*) FROM client WHERE (select count(*) FROM client where genre='F')>3000 group BY ville;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		116K	3074K	606 (2)	00:00:08
1	HASH GROUP BY		116K	3074K	606 (2)	00:00:08
* 2	FILTER					
3	TABLE ACCESS FULL	CLIENT	116K	3074K	299 (1)	00:00:04
4	SORT AGGREGATE		1	3		
* 5	TABLE ACCESS FULL	CLIENT	43286	126K	301 (1)	00:00:04

Predicate Information (identified by operation id):

2 - filter((SELECT COUNT(*) FROM "CLIENT" "CLIENT" WHERE "GENRE"='F')>3000)

5 - filter("GENRE"='F')

- Oracle a parcouru la table client pour filtrer par rapport au parametre Genre = 'F' (5) puis a parcouru une nouvelle fois la table afin de récupérer la valeur donnée par la sous requête (2) puis a renvoyé le résultat.

8. Donner le nombre de clients dans la ville de Dijon

- **Requête SQL :** select count(*) from client where ville='Dijon';
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	27	300 (1)	00:00:04
1	SORT AGGREGATE		1	27		
* 2	TABLE ACCESS FULL	CLIENT	12734	335K	300 (1)	00:00:04

Predicate Information (identified by operation id):

2 - filter("VILLE"='Dijon')

- Oracle a parcouru entièrement la table filtré avec 2 (Ville = 'Dijon'), a appliqué la fonction d'agregation COUNT et a renvoyé le résultat.

9. Afficher la liste des clients triés par ordre croissant de nom

- **Requête SQL :** Select * from client ORDER BY nom ASC;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		116K	12M		3191 (1)	00:00:39
1	SORT ORDER BY		116K	12M	14M	3191 (1)	00:00:39
2	TABLE ACCESS FULL	CLIENT	116K	12M		300 (1)	00:00:04

- Oracle a parcouru entièrement la table (TABLE ACCESS FULL), puis a fait le SORT désiré (Nom croissant)

10. Afficher la liste des clients triés par ordre décroissant de nom

- **Requête SQL** : `Select * from client ORDER BY nom DESC;`
- **EXPLAIN PLAN** :

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		116K	12M		3191 (1)	00:00:39
1	SORT ORDER BY		116K	12M	14M	3191 (1)	00:00:39
2	TABLE ACCESS FULL	CLIENT	116K	12M		300 (1)	00:00:04

- Oracle a parcouru entièrement la table (TABLE ACCESS FULL), puis a fait le SORT désiré (Nom décroissant).

II. Les plans d'exécution d'Oracle – Jointures

1. Afficher la date de la dernière commande

- **Requête SQL :** select MAX(jour) from commande;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	9	1028 (1)	00:00:13
1	SORT AGGREGATE		1	9		
2	TABLE ACCESS FULL	COMMANDE	1058K	9305K	1028 (1)	00:00:13

- Oracle a parcouru entièrement la table (TABLE ACCESS FULL), a effectué un sort et a fait le select.

2. Afficher la date de la dernière commande pour chaque client

- **Requête SQL :** select client,MAX(jour) from commande GROUP BY client;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1058K	22M	1074 (6)	00:00:13
1	HASH GROUP BY		1058K	22M	1074 (6)	00:00:13
2	TABLE ACCESS FULL	COMMANDE	1058K	22M	1029 (1)	00:00:13

- Oracle a parcouru entièrement la table (TABLE ACCESS FULL), puis a fait le GROUP BY et a effectué le select.

3. Afficher la date de la dernière commande de la ville de Toulouse

- **Requête SQL :** `select MAX(commande.jour) from commande,client where commande.client = client.id and client.ville='Toulouse';`
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	36	8461K (1)	28:12:17
1	SORT AGGREGATE		1	36		
2	MERGE JOIN CARTESIAN		8725M	292G	8461K (1)	28:12:17
* 3	TABLE ACCESS FULL	CLIENT	8241	217K	300 (1)	00:00:04
4	BUFFER SORT		1058K	9305K	8461K (1)	28:12:14
5	TABLE ACCESS FULL	COMMANDE	1058K	9305K	1027 (1)	00:00:13

Predicate Information (identified by operation id):

3 - filter("CLIENT"."VILLE"='Toulouse')

- Oracle a parcouru entièrement la table commande puis a parcouru entièrement la table client en sélectionnant seulement ceux qui sont de Toulouse, a effectué une jointure, puis a appliqué la fonction d'agrégation MAX et a renvoyé le résultat.

4. Afficher la date de la dernière commande pour chaque client de la ville de Toulouse

- **Requête SQL :** `select client.id,MAX(commande.jour) from commande,client where client.ville='Toulouse' and commande.client=client.id GROUP BY client.id;`

• EXPLAIN PLAN :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		95968	5810K	1339 (2)	00:00:17
1	HASH GROUP BY		95968	5810K	1339 (2)	00:00:17
* 2	HASH JOIN		95968	5810K	1335 (2)	00:00:17
* 3	TABLE ACCESS FULL	CLIENT	8241	321K	300 (1)	00:00:04
4	TABLE ACCESS FULL	COMMANDE	1058K	22M	1029 (1)	00:00:13

Predicate Information (identified by operation id):

2 - access("COMMANDE"."CLIENT"="CLIENT"."ID")

3 - filter("CLIENT"."VILLE"='Toulouse')

- Oracle a récupéré la table client avec seulement les clients de Toulouse (3) , puis a récupéré la table commande, il a effectué une jointure sur l'id des clients (2), a regroupé par client et a renvoyé le résultat.

5. Afficher le nom du client et le nom de la ville pour chaque client qui a passé plus de 3 commandes en septembre dernier

- **Requête SQL :** select client.nom,client.ville from client where (Select count(*) from commande where jour>'31/08/12' and jour<'01/10/12')>3 ;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		116K	6148K	1343 (2)	00:00:17
* 1	FILTER					
2	TABLE ACCESS FULL	CLIENT	116K	6148K	299 (1)	00:00:04
3	SORT AGGREGATE		1	9		
* 4	FILTER					
* 5	TABLE ACCESS FULL	COMMANDE	65641	576K	1044 (3)	00:00:13

Predicate Information (identified by operation id):

```

1 - filter( (SELECT COUNT(*) FROM "COMMANDE" "COMMANDE" WHERE
TO_DATE('31/08/12')<TO_DATE('01/10/12') AND "JOUR">'31/08/12' AND
"JOUR"<'01/10/12')>3)
4 - filter(TO_DATE('31/08/12')<TO_DATE('01/10/12'))
5 - filter("JOUR">'31/08/12' AND "JOUR"<'01/10/12')

```

- Oracle a récupéré la table commande en ayant filtré seulement les commandes du mois de Septembre. Oracle a renvoyé le résultat du nombre de commande du client puis a récupéré la table client en filtrant avec (1) puis a retourné le résultat.

6. Afficher le nom du client et le nom de la ville pour chaque client qui a acheté un produit vert

- **Requête SQL :** select client.nom,client.ville from client,commande,produit where client.id=commande.client and commande.produit = produit.id and produit.couleur='vert' ;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		294K	37M		5183 (1)	00:01:03
* 1	HASH JOIN		294K	37M	9000K	5183 (1)	00:01:03
2	TABLE ACCESS FULL	CLIENT	116K	7628K		299 (1)	00:00:04
* 3	HASH JOIN		294K	18M	1736K	3358 (1)	00:00:41
* 4	TABLE ACCESS FULL	PRODUIT	34170	1334K		330 (1)	00:00:04
5	TABLE ACCESS FULL	COMMANDE	1058K	26M		1030 (2)	00:00:13

Predicate Information (identified by operation id):

1 - access("CLIENT"."ID"="COMMANDE"."CLIENT")
 3 - access("COMMANDE"."PRODUIT"="PRODUIT"."ID")
 4 - filter("PRODUIT"."COULEUR"='vert')

- Oracle a récupéré la table client puis la table produit filtré avec seulement les produits de couleur vert, a fait une jointure sur ces deux tables avec l'id du produit, a récupéré la table des clients, a fait une jointure sur l'id client et a renvoyé le résultat

7. Afficher le nom du client et le nom de la ville pour chaque client qui a acheté au moins 5 produits verts.

- **Requête SQL :** select client.nom,client.ville from client,commande where client.id=commande.client and (select count(*) from commande,produit where commande.produit=produit.id and produit.couleur='vert')>4 group by client.nom,client.ville;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		1058K	80M		5780 (2)	00:01:10
1	HASH GROUP BY		1058K	80M		5780 (2)	00:01:10
* 2	FILTER						
* 3	HASH JOIN		1058K	80M	9000K	3028 (1)	00:00:37
4	TABLE ACCESS FULL	CLIENT	116K	7628K		299 (1)	00:00:04
5	TABLE ACCESS FULL	COMMANDE	1058K	13M		1029 (1)	00:00:13
6	SORT AGGREGATE		1	53			
* 7	HASH JOIN		294K	14M	1736K	2707 (1)	00:00:33
* 8	TABLE ACCESS FULL	PRODUIT	34170	1334K		330 (1)	00:00:04
9	TABLE ACCESS FULL	COMMANDE	1058K	13M		1030 (2)	00:00:13

Predicate Information (identified by operation id):

```

2 - filter( (SELECT COUNT(*) FROM "PRODUIT" "PRODUIT","COMMANDE" "COMMANDE"
WHERE "COMMANDE"."PRODUIT"="PRODUIT"."ID" AND "PRODUIT"."COULEUR"='vert')>4)
3 - access("CLIENT"."ID"="COMMANDE"."CLIENT")
7 - access("COMMANDE"."PRODUIT"="PRODUIT"."ID")
8 - filter("PRODUIT"."COULEUR"='vert')

```

- Oracle a parcouru la table commande puis la table produit trié avec seulement les produits de couleur vert (8), puis a fait une jointure de ces deux tables par l'id du produit, puis a appliqué la fonction d'agrégation COUNT après Oracle récupère les tables client et commande et a fait une jointure sur id client et a regroupé par le nom du client et a renvoyé le résultat.

III. L'optimisation par les clusters

1. Vitesse d'insertion

En reprenant le script de création sans clusters, l'insertion s'effectue en 162 secondes (2 min et 42 secondes) alors que le script de création avec clusters, l'insertion s'effectue en 17 secondes.

2. Afficher la date de la dernière commande

- **Requête SQL :** select MAX(jour) from commande;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	9	3663 (1)	00:00:44
1	SORT AGGREGATE		1	9		
2	TABLE ACCESS FULL	COMMANDE	91721	806K	3663 (1)	00:00:44

- Oracle a parcouru entièrement la table (TABLE ACCESS FULL), a effectué un sort et a fait le select.

3. Afficher la date de la dernière commande pour chaque client

- **Requête SQL :** select client,MAX(jour) from commande GROUP BY client;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		91721	1970K	3667 (1)	00:00:45
1	HASH GROUP BY		91721	1970K	3667 (1)	00:00:45
2	TABLE ACCESS FULL	COMMANDE	91721	1970K	3663 (1)	00:00:44

- Oracle a parcouru entièrement la table (TABLE ACCESS FULL), puis a fait le GROUP BY et a effectué le select.

4. Afficher la date de la dernière commande de la ville de Toulouse

- **Requête SQL :** `select MAX(commande.jour) from commande,client where commande.client = client.id and client.ville='Toulouse';`
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	34	3964 (1)	00:00:48
1	SORT AGGREGATE		1	34		
* 2	HASH JOIN		88868	2950K	3964 (1)	00:00:48
* 3	TABLE ACCESS FULL	CLIENT	7290	87480	300 (1)	00:00:04
4	TABLE ACCESS FULL	COMMANDE	91721	1970K	3663 (1)	00:00:44

Predicate Information (identified by operation id):

```

2 - access("COMMANDE"."CLIENT"="CLIENT"."ID")
3 - filter("CLIENT"."VILLE"='Toulouse')

```

Note

```

-----
- dynamic sampling used for this statement (level=2)

```

- Oracle a parcouru entièrement la table commande puis a parcouru entièrement la table client en sélectionnant seulement ceux qui sont de Toulouse, a effectué une jointure, puis a appliqué la fonction d'agrégation MAX et a renvoyé le résultat.

5. Afficher la date de la dernière commande pour chaque client de la ville de Toulouse

- **Requête SQL :** `select client.id,MAX(commande.jour) from commande,client where client.ville='Toulouse' and commande.client=client.id GROUP BY client.id;`
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	34	3968 (1)	00:00:48
1	HASH GROUP BY		1	34	3968 (1)	00:00:48
* 2	HASH JOIN		88868	2950K	3964 (1)	00:00:48
* 3	TABLE ACCESS FULL	CLIENT	7290	87480	300 (1)	00:00:04
4	TABLE ACCESS FULL	COMMANDE	91721	1970K	3663 (1)	00:00:44

Predicate Information (identified by operation id):

```

2 - access("COMMANDE"."CLIENT"="CLIENT"."ID")
3 - filter("CLIENT"."VILLE"='Toulouse')

```

Note

```

- dynamic sampling used for this statement (level=2)

```

- Oracle a récupéré la table client avec seulement les clients de Toulouse (3) , puis a récupéré la table commande, il a effectué une jointure sur l'id des clients (2), a regroupé par client et a renvoyé le résultat.

6. Afficher le nom du client et le nom de la ville pour chaque client qui a passé plus de 3 commandes en septembre dernier

- **Requête SQL :** select client.nom,client.ville from client where (Select count(*) from commande where jour>'31/08/12' and jour<'01/10/12')>3 ;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		100K	3515K	3964 (1)	00:00:48
* 1	FILTER					
2	TABLE ACCESS FULL	CLIENT	100K	3515K	299 (1)	00:00:04
3	SORT AGGREGATE		1	9		
* 4	FILTER					
* 5	TABLE ACCESS FULL	COMMANDE	5572	50148	3665 (1)	00:00:44

Predicate Information (identified by operation id):

```

1 - filter( (SELECT COUNT(*) FROM "COMMANDE" "COMMANDE" WHERE
            TO_DATE('31/08/12')<TO_DATE('01/10/12') AND "JOUR">'31/08/12' AND
            "JOUR"<'01/10/12')>3)
4 - filter(TO_DATE('31/08/12')<TO_DATE('01/10/12'))
5 - filter("JOUR">'31/08/12' AND "JOUR"<'01/10/12')

```

Note

```

-----
- dynamic sampling used for this statement (level=2)

25 lignes sélectionnées

```

- Oracle a récupéré la table commande en ayant filtré seulement les commandes du mois de Septembre. Oracle a renvoyé le résultat du nombre de commande du client puis a récupéré la table client en filtrant avec (1) puis a retourné le résultat.

7. Afficher le nom du client et le nom de la ville pour chaque client qui a acheté un produit vert

- **Requête SQL :** `select client.nom,client.ville from client,commande,produit where client.id=commande.client and commande.produit = produit.id and produit.couleur='vert' ;`
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		91679	9579K		903 (1)	00:00:11
* 1	HASH JOIN		91679	9579K	5176K	903 (1)	00:00:11
2	TABLE ACCESS FULL	CLIENT	100K	4003K		299 (1)	00:00:04
3	NESTED LOOPS		91679	5908K		12 (0)	00:00:01
4	TABLE ACCESS BY INDEX ROWID	PRODUIT	2143	85720		1 (0)	00:00:01
* 5	INDEX RANGE SCAN	IDX_CLUSTER_COULEUR	46			1 (0)	00:00:01
6	TABLE ACCESS CLUSTER	COMMANDE	43	1118		1 (0)	00:00:01
* 7	INDEX UNIQUE SCAN	IDX_CLUSTER_PRODUIT_COMMANDE	1			0 (0)	00:00:01

Predicate Information (identified by operation id):

```

1 - access("CLIENT"."ID"="COMMANDE"."CLIENT")
5 - access("PRODUIT"."COULEUR"='vert')
7 - access("COMMANDE"."PRODUIT"="PRODUIT"."ID")

```

Note

```

- dynamic sampling used for this statement (level=2)

```

25 lignes sélectionnées

- Ici on voit bien que la requête a utilisé les clusters créés en (7) `idx_cluster_produit_commande` et en (5) `idx_cluster_couleur`.

8. Afficher le nom du client et le nom de la ville pour chaque client qui a acheté au moins 5 produits verts.

- **Requête SQL :** select client.nom,client.ville from client,commande where client.id=commande.client and (select count(*) from commande,produit where commande.produit=produit.id and produit.couleur='vert')>4 group by client.nom,client.ville;
- **EXPLAIN PLAN :**

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		91721	4836K		5557 (1)	00:01:07
1	HASH GROUP BY		91721	4836K	5792K	5557 (1)	00:01:07
* 2	FILTER						
* 3	HASH JOIN		91721	4836K	2240K	4325 (1)	00:00:52
4	TABLE ACCESS FULL	COMMANDE	91721	1164K		3663 (1)	00:00:44
5	TABLE ACCESS FULL	CLIENT	100K	4003K		299 (1)	00:00:04
6	SORT AGGREGATE		1	53			
7	NESTED LOOPS		91679	4745K		12 (0)	00:00:01
8	TABLE ACCESS BY INDEX ROWID	PRODUIT	2143	85720		1 (0)	00:00:01
* 9	INDEX RANGE SCAN	IDX_CLUSTER_COULEUR	46			1 (0)	00:00:01
10	TABLE ACCESS CLUSTER	COMMANDE	43	559		1 (0)	00:00:01
* 11	INDEX UNIQUE SCAN	IDX_CLUSTER_PRODUIT_COMMANDE	1			0 (0)	00:00:01

Predicate Information (identified by operation id):

```

2 - filter( (SELECT COUNT(*) FROM "PRODUIT" "PRODUIT","COMMANDE" "COMMANDE" WHERE
"COMMANDE"."PRODUIT"="PRODUIT"."ID" AND "PRODUIT"."COULEUR"='vert')>4)
3 - access("CLIENT"."ID"="COMMANDE"."CLIENT")
9 - access("PRODUIT"."COULEUR"='vert')
11 - access("COMMANDE"."PRODUIT"="PRODUIT"."ID")

```

Note

```

- dynamic sampling used for this statement (level=2)

```

31 lignes sélectionnées

- Ici on voit bien que la requête a utilisé les clusters créés en (11) idx_cluster_produit_commande et en (9) idx_cluster_couleur.