# Apartado 1.A

|  |  |  |  |
| --- | --- | --- | --- |
| CONCEPTO | COSTE(TOTAL) | NUM.FILAS(TOTAL) | NUM. OPERACIONES |
| -- CONSULTA 1 -- | 15 | 20 | 4 |
| -- CONSULTA 2 -- | 10 | 5 | 5 |
| -- CONSULTA 3 -- | 10 | 6 | 8 |
| -- CONSULTA 4 -- | - | - | - |
| -- CONSULTA 5 -- | 7 | 24 | 6 |
|  |  |  |  |

-- CONSULTA 1 –

delete plan\_table; /\* primero borra las tuplas \*/

EXPLAIN PLAN

INTO plan\_table

FOR (select \* from cliente) union (select \* from moroso);

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| OPERACION | OPCIONES | TABLA | | Coste | Filas | PADRE | ID |
|  |  |  | |  |  |  |  |
| SORT | UNIQUE |  | | 8 | 10 | 0 | 1 |
| UNION-ALL |  | |  |  |  | 1 | 2 |
| TABLE ACCESS | FULL | | CLIENTE | 3 | 6 | 2 | 3 |
| TABLE ACCESS | FULL | | MOROSO | 3 | 4 | 2 | 4 |

ID 1

SORT

CLIENTE

ID 4

TABLE-ACS FULL

MOROSO

ID 2

UNION-ALL

CLIENTE

ID 3

TABLE-ACS FULL

CLIENTE

**Se han ejecutado estas operaciones por:**

* **Al realizarse un select \* , se realiza un TABLE-ACCESS FULL en ambas tablas**
* **Al ejecutarse un union ALL se realiza la selección de todas las filas de las consultas.**
* **SORT siempre se ejecuta para ordenar y retirar duplicados**

-- CONSULTA 2 --

delete plan\_table; /\* borra las tuplas de explicación anterior\*/

EXPLAIN PLAN

INTO plan\_table

FOR (select \* from cliente where DNI = '00000005') union

(select \* from moroso where NombreC = 'Client E');

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| OPERACION | OPCIONES | TABLA | | Coste | Filas | PADRE | ID |
| SORT | UNIQUE |  | | 6 | 2 | 0 | 1 |
| UNION-ALL |  |  | |  |  | 1 | 2 |
| TABLE ACCESS | BY INDEX  ROWID | | CLIENTE | 1 | 1 | 2 | 3 |
| INDEX | UNIQUE SCAN | | SYS\_C0027175 | 0 | 1 | 3 | 4 |
| TABLE ACCESS | FULL | | MOROSO | 3 | 1 | 2 | 5 |

ID 5

TABLE-ACS FULL

MOROSO

ID 3

TABLE ACS BY INDEX ROWID

CLIENTE



ID 2

UNION-ALL

CLIENTE

ID 4

INDEX UNIQUE SCAN

SYS\_C0027175

**Se han ejecutado estas operaciones por:**

* **Al realizarse un select \* , se realiza un TABLE-ACCESS FULL en la tabla moroso que no tiene ningún índice sobre nombreC.**
* **En el caso de la tabla de clientes se realiza una búsqueda por un solo ídice ya que se utiliza el campo DNI que es PK e índice (reduce el nº de filas que se seleccionan).**
* **Al ejecutarse un union ALL se realiza la selección de todas las filas de las consultas.**
* **SORT siempre se ejecuta para ordenar y retirar duplicados.**

-- CONSULTA 3 -- anidados ----

delete plan\_table;

EXPLAIN PLAN

INTO plan\_table

FOR select \* from cliente where DNI in

(select DNI from moroso where NombreC = 'Client E');

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| OPERACION | OPCIONES | TABLA | | Coste | Filas | PADRE | ID |
| HASH JOIN |  |  | | 3 | 1 | 0 | 1 |
| NESTED LOOPS |  |  | |  |  | 1 | 2 |
| NESTED LOOPS |  | |  | 3 | 1 | 2 | 3 |
| STATISTICS COLLECTOR |  | |  |  |  | 3 | 4 |
| TABLE ACCESS | FULL | | MOROSO | 2 | 1 | 4 | 5 |
| INDEX | UNIQUE SCAN | | SYS\_C009880 | 0 | 1 | 3 | 6 |
| TABLE ACCESS | BY INDEX ROWID | | CLIENTE | 1 | 1 | 2 | 7 |
| TABLE ACCES | FULL | | CLIENTE | 1 | 1 | 1 | 8 |

ID 1

HASSHS JOIN

ID 8

TABLE ACCES FULL MOROSO

ID 5

TABLE ACCES FULL MOROSO

CLIENTE

ID 4

STATISTICS COLLECTOR

ID 6

TABLE UNIQUE SCAN CLIENTE

CLIENTE

ID 3

NESTED LOOPS

ID 7

TABLE ACS BY INDEX ROWID CLIENTE

CLIENTE

ID 2

NESTED LOOPS

CLIENTE

**Se han ejecutado estas operaciones por:**

* **Al realizarse un select….in , se realiza un TABLE-ACCESS FULL en la tabla morosos ya que se busca por el nombre del cliente y este no es un índice.**
* **Sobre esta consulta Oracle ha realizado una recolección de estadísticas que le ayudan a escoger el plan de ejecución.**
* **Por otro lado (nodo 6) se ha realizado un búsqueda por un único índice ya que el dni de la tabla cliente contiene un índice.**
* **A continuación aparecen dos nested loops ya que accede a las segunda (moroso) tabla a través de los índices de la primera (cliente), se puede apreciar en el nodo 7.**
* **Como último paso se ha creado una tabla hash en memoria con la clave de join entre cliente y moroso ya que la tabla es lo suficientemente grande.**

-- CONSULTA 4 – que pasa en esta consulta? ----

delete plan\_table;

EXPLAIN PLAN

SET STATEMENT\_ID= 'mipruxx'

INTO plan\_table

FOR (select \* from cliente) union (select \* from invierte);

**NO SE HA PODIDO REALIZAR ESTA CONSULTA POR QUE LOS ATRIBUTOS DE LAS TABLAS NO COINCIDEN.**

-- CONSULTA 5 --

delete plan\_table;

EXPLAIN PLAN

SET STATEMENT\_ID= 'mipru05'

INTO plan\_table

FOR (select \* from cliente where dni in

(select dni from invierte));

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| OPERACION | OPCIONES | TABLA | | Coste | Filas | PADRE | ID |
| HASH JOIN | SEMI |  | | 3 | 3 | 0 | 1 |
| NESTED LOOPS | SEMI |  | |  | 3 | 1 | 2 |
| STATISTICS COLLECTOR |  | |  |  |  | 2 | 3 |
| TABLE ACCESS | FULL | | CLIENTE | 2 | 6 | 3 | 4 |
| INDEX | RANGE SCAN | | CLAVE\_INVIRTE\_PRIM | 1 | 4 | 2 | 5 |
| INDEX | FULL SCAN | | CLAVE\_INVIRTE\_PRIM | 1 | 8 | 1 | 6 |

ID 1

HASSHS JOIN SEMI

ID 6

INDEX FULL SCAN CLAVE\_INVIRTE\_PRIM

ID 4

TABLE ACCESS FULL CLIENTE

ID 3

STATISTICS COLLECTOR

ID 5

INDEX RANGE SCAN CLAVE\_INVIRTE\_PRIM

CLIENTE

ID 2

NESTED LOOPS SEMI

CLIENTE

**Se han ejecutado estas operaciones por:**

* **TABLE-ACCESS FULL en la tabla clientes ya que se están consultando todas las filas y atributos.**
* **Sobre esta consulta Oracle ha realizado una recolección de estadísticas que le ayudan a escoger el plan de ejecución.**
* **A continuación aparecen un nested loop ya que accede a las segunda (cliente) tabla a través de los índices de la primera (invierte), se puede apreciar en el nodo 2.**
* **De invierte para poder realizar el paso de nested loop ha hecho una búsqueda por un rango del índice, dado ya que el índice no es único.**
* **Se ha realizado un escaneo de índice completo de la tabla invierte por que tiene todos sus atributos en índices.**
* **Como último paso se ha creado una tabla hash en memoria con la clave de join entre cliente e invierte ya que la tabla es lo suficientemente grande.**

**Preguntas:**

1. **Porqué en CONSULTA 2 el acceso será más eficiente que en CONSULTA 1?**
   1. **Por que al ‘estrechar’ la búsqueda solo a una fila concreta y que además es índice la selección de filas es menor que si se buscan todas.**
2. **Porqué en CONSULTA 2 no accede por el índice a moroso?**
   1. **Por que el atributo NombreC no tiene índice.**
3. **Porqué en CONSULTA 3 sí accede por el índice a CLIENTE?**
   1. **Por que lo ha utilizado para acceder a cada fila de la tabla necesaria.**
4. **Porqué en CONSULTA 3 no accede por el índice a MOROSO?**
   1. **Por que NombreC no tiene índice.**

# Apartado 1.B

-- CONSULTA 1 --

EXPLAIN PLAN

SET STATEMENT\_ID= 'mipru05'

INTO plan\_table

FOR select distinct NombreC

from Cliente, Compras, Invierte

where Cliente.DNI = Invierte.DNI and

Invierte.NombreE = 'Empresa 55' and

Compras.DNI = Cliente.DNI and

Compras. Importe >1000;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| OPERACION | OPCIONES | TABLA | | Coste | Filas | PADRE | ID |
| HASH | UNIQUE |  | | 5 | 1 | 0 | 1 |
| HASH JOIN | SEMI |  | | 5 | 1 | 1 | 2 |
| HASH JOIN |  |  | | 3 | 2 | 2 | 3 |
| NESTED LOOPS |  |  | |  |  | 3 | 4 |
| NESTED LOOPS |  | |  | 3 | 2 | 4 | 5 |
| STATISTICS COLLECTOR |  | |  |  |  | 5 | 6 |
| INDEX | SKIP SCAN | | CLAVE\_INVIRTE\_PRIM | 1 | 2 | 6 | 7 |
| INDEX | UNIQUE SCAN | | SYS\_C009880 | 0 | 1 | 5 | 8 |
| TABLE ACCESS | BY INDEX ROWID | | CLIENTE | 1 | 1 | 4 | 9 |
| TABLE ACCES | FULL | | CLIENTE | 1 | 1 | 3 | 10 |
| TABLE ACCESS | FULL | | COMPRAS | 2 | 11 | 2 | 11 |

ID 1

HASH UNIQUE

ID 11

TABLE ACCESS FULL COMPRAS

ID 10

TABLE ACCESS FULL CLIENTE

CLIENTE

ID 9

TABLE ACCES BY INDEX ROWID CLIENTE

CLIENTE

ID 8

INDEX UNIQUE SCAN SYS\_C009880

CLIENTE

ID 7

INDEX UNIQUE SCAN CLAVE\_INVIRTE\_PRIM

ID 6

STATISTICS COLLETOR

ID 5

NESTED LOOPS

ID 4

NESTED LOOPS

ID 3

HASH JOIN

ID 2

HASH JOIN SEMI

CLIENTE

**Se han ejecutado estas operaciones por:**

* **En el nodo 7 se ha realizado una búsqueda por índice único sobre la tabla invierte para leer cada fila y consultar solo las que tengan el nombre de empresa ‘Empresa 55’.**
* **De estas filas se ha realizado un análisis de estadísticas para decidir el mejor plan de ejecución.**
* **Se ha realizado una búsqueda por índice único (dni) en cliente para posteriormente usar cada uno de estos índices para con la tabla invierte, y comprar fila a fila (nested loop).**
* **Con el resultado de estas filas que satisfagan la condición de tener el dni igual se accede a la tabla clientes fila a fila para obtenerlos usando el índice de cliente sobre la otra tabla.**
* **Se genera una tabla en memoria para realizar comprobaciones más rápido (nodo 3) con la tabla clientes y los resultados de las consultas anteriores.**
* **Se consultan las compras y se unen a la tabla hash en memoria.**
* **Esta tabla es filtrada para que no contenga filas con datos repetidos o tabla única en memoria.**

-- CONSULTA 1 – sin distinct

EXPLAIN PLAN

SET STATEMENT\_ID= 'mipru05'

INTO plan\_table

FOR select distinct NombreC

from Cliente, Compras, Invierte

where Cliente.DNI = Invierte.DNI and

Invierte.NombreE = 'Empresa 55' and

Compras.DNI = Cliente.DNI and

Compras. Importe >1000;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| OPERACION | OPCIONES | TABLA | | Coste | Filas | PADRE | ID |
| HASH JOIN |  |  | | 5 | 4 | 0 | 1 |
| HASH JOIN |  |  | | 3 | 2 | 1 | 2 |
| NESTED LOOPS |  |  | |  |  | 2 | 3 |
| NESTED LOOPS |  | |  | 3 | 2 | 3 | 4 |
| STATISTICS COLLECTOR |  | |  |  |  | 4 | 5 |
| INDEX | SKIP SCAN | | CLAVE\_INVIRTE\_PRIM | 1 | 2 | 5 | 6 |
| INDEX | UNIQUE SCAN | | SYS\_C009880 | 0 | 1 | 4 | 7 |
| TABLE ACCESS | BY INDEX ROWID | | CLIENTE | 1 | 1 | 3 | 8 |
| TABLE ACCES | FULL | | CLIENTE | 1 | 1 | 2 | 9 |
| TABLE ACCESS | FULL | | COMPRAS | 2 | 11 | 1 | 10 |

ID 6I

NDEX SKIP SCAN CLAVE\_INVIRTE\_PRIM

ID 1

HASH JOIN

ID 7

INDEX UNIQUE SCAN SYS\_C009880

CLIENTE

ID 8

TABLE ACCES BY INDEX ROWID CLIENTE

CLIENTE

ID 9

TABLE ACCESS FULL CLIENTE

CLIENTE

ID 10

TABLE ACCESS FULL COMPRAS

ID 5

STATISTICS COLLETOR

ID 4

NESTED LOOPS

ID 3

NESTED LOOPS

ID 2

HASH JOIN

CLIENTE

**Se han ejecutado estas operaciones por:**

* **Es igual que la anterior a excepción de que no se realiza el hash unique sobre la consulta de compras y la tabla hash. Y la tabla hash creada después de la consulta de la tabla compras no es semi.**

-- CONSULTA 1 –- Equivalente anidada

EXPLAIN PLAN

SET STATEMENT\_ID= 'mipru05'

INTO plan\_table

FOR select NombreC from cliente where cliente.DNI in

( select compras.dni from compras where importe > 1000)

and cliente.dni in ( select invierte.dni from invierte where invierte.nombreE = 'Empresa 55');

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| OPERACION | OPCIONES | TABLA | | Coste | Filas | PADRE | ID |
| HASH JOIN |  |  | | 5 | 4 | 0 | 1 |
| HASH JOIN |  |  | | 3 | 2 | 1 | 2 |
| NESTED LOOPS |  |  | |  |  | 2 | 3 |
| NESTED LOOPS |  | |  | 3 | 2 | 3 | 4 |
| STATISTICS COLLECTOR |  | |  |  |  | 4 | 5 |
| SORT | UNIQUE | |  | 1 | 2 | 5 | 6 |
| INDEX | SKIP SCAN | | CLAVE\_INVIRTE\_PRIM | 1 | 2 | 6 | 7 |
| INDEX | UNIQUE SCAN | | SYS\_C009880 | 0 | 1 | 4 | 8 |
| TABLE ACCESS | BY INDEX ROWID | | CLIENTE | 1 | 1 | 3 | 9 |
| TABLE ACCES | FULL | | CLIENTE | 1 | 1 | 2 | 10 |
| TABLE ACCESS | FULL | | COMPRAS | 2 | 11 | 1 | 11 |

ID 1

HASH JOIN

ID 6

SORT UNIQUE

ID 7

INDEX SKIP SCAN CLAVE\_INVIRTE\_PRIM

ID 7

INDEX UNIQUE SCAN SYS\_C009880

CLIENTE

ID 8

TABLE ACCES BY INDEX ROWID CLIENTE

CLIENTE

ID 9

TABLE ACCESS FULL CLIENTE

CLIENTE

ID 10

TABLE ACCESS FULL COMPRAS

ID 5

STATISTICS COLLETOR

ID 4

NESTED LOOPS

ID 3

HASH JOIN

ID 2

HASH JOIN

CLIENTE

**Se han ejecutado estas operaciones por:**

* **Las diferencias con la primera son:**
  + **Ya no se realiza un hash semi ni un hash unique al final del plan de ejecución.**
  + **Y que se hace un sort unique de la consulta por el índice en la tabla invierte (nodo 7).**

d) En ambas sale un número de operaciones de 11, se realizan dos accesos a la tabla cliente y uno a invierte y a compras; en ambas se realiza acceso FULL sobre clientes y compras, ambas tienen dos NESTED LOOPS, y acceden por índice unique a cliente, y por skip scan a invierte.

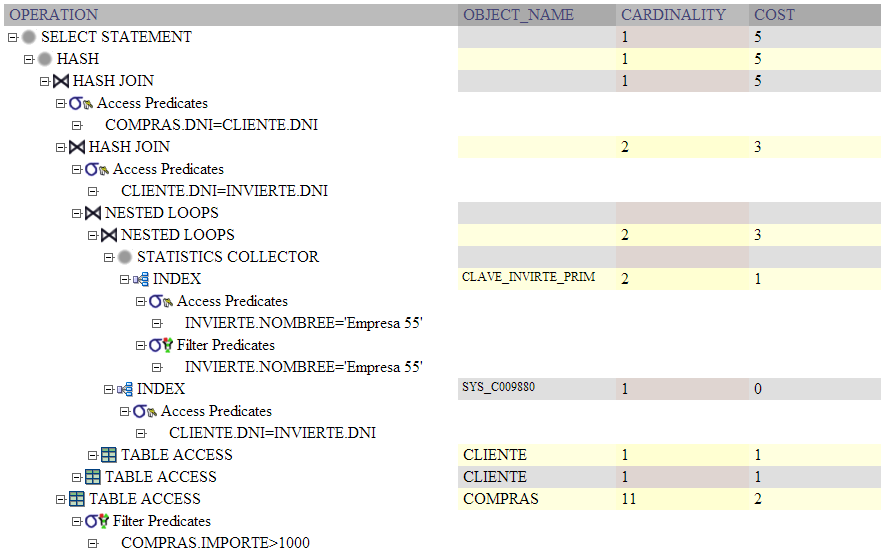
Por lo tanto ambas tienen el mismo coste en término de estos datos.

e) Atendiendo al número de filas y al coste de las consultas: la consulta a) tiene un coste de 21 y un número de filas de 22, mientras que b) tiene un coste de 17 y un número de filas de 26. Siendo por tanto b) más eficiente pero consultando más filas.

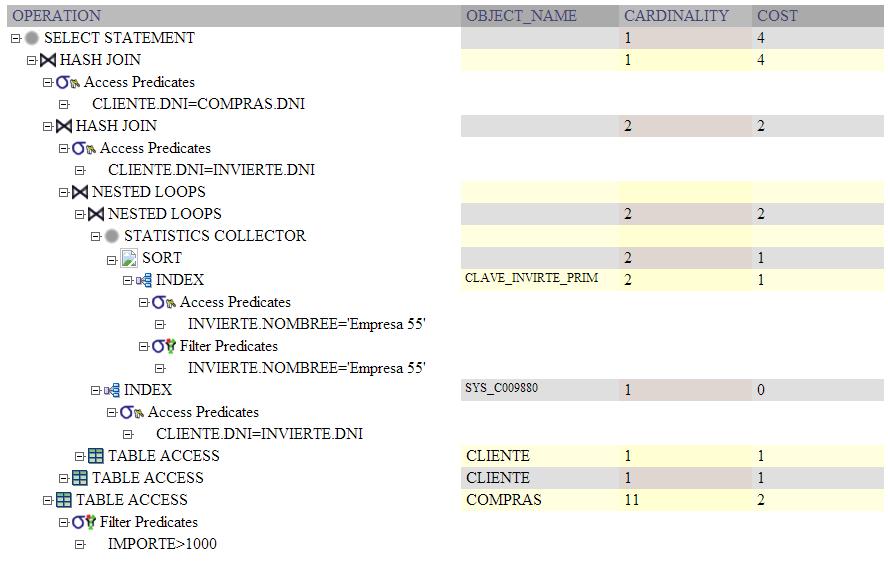
f) HASH UNIQUE

g)

consulta a)



consulta b)



Difieren en que en a) se realiza una operación HASH para hacer el distinct y en b) no, también en b) se hace una ordenación (SORT) antes de la recolección de estadísticas que en a) no ocurre.

h)

FULL TABLE SCAN: select \* from tarjeta;

TABLE ACCES BY ROWID y INDEX UNIQUE SCAN: select \* from tarjeta where numt=’00000001’

CARTESIAN JOIN: select \* from cliente, tienet;

SORT: select \* from moroso order by dni asc;

NESTED LOOPS Y FULL INDEX SCAN: select \* from moroso where dni in (select dni from tienet);

# Apartado 2

1. create table diccion ( PalID varchar2 (20) primary key, Descripcion varchar2 (50), PadreID char(20));
2. insert into diccion values ('select jerarquica', 'estructura tabla en arbol', 'select compuesta');

insert into diccion values ('fecha sistema', 'es la fecha que tiene el ordenador', 'fecha');

insert into diccion values ('fecha', 'tipo de dato, en oracle : DATE', 'nada');

insert into diccion values ('select compuesta', 'consulta con varias partes', 'select');

insert into diccion values ('select simple', 'consultas con una sola instruccion', 'select');

insert into diccion values ('select', 'hacer consulta', 'nada');

insert into diccion values ('sql', 'lenguaje de consultas estructuradas', 'nada');

insert into diccion values ('select correlativa', 'coordina resultado subconsulta','select compuesta');