SE APUNTARÁ EN ROJO LO ACORDADO CON EL PROFESOR EN LA TUTORÍA

A.- Para cada consulta*:*
- Ejecuta cada instrucción por separado y copia resultado para compararlos en los siguientes apartados

A0)

	OPERATION	OPTIONS	OBJECT_NAME	COST	CARDINALITY	PARENT_ID	ID
1	SORT	UNIQUE	(null)	7	7	0	1
2	UNION-ALL	(null)	(null)	(null)	(null)	1	2
3	TABLE ACCESS	FULL	CLIENTE	3	6	2	3
4	TABLE ACCESS	FULL	MOROSO	3	1	2	4

A1)

	OPERATION	OPTIONS	OBJECT_NAME	COST	CARDINALITY	PARENT_ID	ID
1	SORT	UNIQUE	(null)	7	5	0	1
2	UNION-ALL	(null)	(null)	(null)	(null)	1	2
3	TABLE ACCESS	FULL	CLIENTE	3	4	2	3
4	TABLE ACCESS	FULL	MOROSO	3	1	2	4

A2)

	OPERATION	OPTIONS	OBJECT_NAME	COST	CARDINALITY	PARENT_ID	ID
1	SORT	UNIQUE	(null)	4	2	0	1
2	UNION-ALL	(null)	(null)	(null)	(null)	1	2
3	TABLE ACCESS	BY INDEX ROWID	CLIENTE	1	1	2	3
4	INDEX	UNIQUE SCAN	SYS_C007781	1	1	3	4
5	TABLE ACCESS	FULL	MOROSO	3	1	2	5

A3)

	OPERATION	OPTIONS	OBJECT_NAME	COST	CARDINALITY	PARENT_ID	ID
1	HASH JOIN	(null)	(null)	4	1	0	1
2	NESTED LOOPS	(null)	(null)	4	1	1	2
3	NESTED LOOPS	(null)	(null)	4	1	2	3
4	STATISTICS COLLECTOR	(null)	(null)	(null)	(null)	3	4
5	TABLE ACCESS	FULL	MOROSO	3	1	4	5
6	INDEX	UNIQUE SCAN	SYS_C007781	0	1	3	6
7	TABLE ACCESS	BY INDEX ROWID	CLIENTE	1	1	2	7
8	TABLE ACCESS	FULL	CLIENTE	1	1	1	8

A4)

	OPERATION	OPTIONS	OBJECT_NAME	COST	CARDINALITY	PARENT_ID	ID
1	HASH JOIN	SEMI	(null)	5	3	0	1
2	NESTED LOOPS	SEMI	(null)	5	3	1	2
3	STATISTICS COLLECTOR	(null)	(null)	(null)	(null)	2	3
4	TABLE ACCESS	FULL	CLIENTE	3	7	3	4
5	INDEX	RANGE SCAN	CLAVE_INVIRTE_PRIM	2	3	2	5
6	INDEX	FAST FULL SCAN	CLAVE_INVIRTE_PRIM	2	8	1	6

A5)

Primera ejecución:

	OPERATION	OPTIONS	OBJECT_NAME	COST	CARDINALITY	PARENT_ID	ID
1	HASH JOIN	(null)	(null)	5	1	0	1
2	NESTED LOOPS	(null)	(null)	5	1	1	2
3	NESTED LOOPS	(null)	(null)	5	1	2	3
4	STATISTICS COLLECTOR	(null)	(null)	(null)	(null)	3	4
5	SORT	UNIQUE	(null)	3	1	4	5
6	TABLE ACCESS	FULL	INVIERTE	3	1	5	6
7	INDEX	UNIQUE SCAN	SYS_C007781	0	1	3	7
8	TABLE ACCESS	BY INDEX ROWID	CLIENTE	1	1	2	8
9	TABLE ACCESS	FULL	CLIENTE	1	1	1	9

Tras ejecutar la misma consulta al día siguiente me sale una cosa diferente a la del primer día, por petición del profesor inserto las dos fotos.

Después de la tutoría del jueves 23-04-2020:

	OPERATION	OPTIONS	OBJECT_NAME	COST	CARDINALITY	PARENT_ID	ID
1	NESTED LOOPS	SEMI	(null)	3	1	0	1
2	TABLE ACCESS	FULL	CLIENTE	3	1	1	2
3	TABLE ACCESS	BY INDEX ROWID BATCHED	INVIERTE	0	1	1	3
4	INDEX	RANGE SCAN	CLAVE_INVIRTE_PRIM	0	1	3	4

A6)

	OPERATION	OPTIONS	OBJECT_NAME	COST	CARDINALITY	PARENT_ID	ID
1	HASH	UNIQUE	(null)	9	2	0	1
2	HASH JOIN	SEMI	(null)	8	2	1	2
3	NESTED LOOPS	SEMI	(null)	8	2	2	3
4	STATISTICS COLLECTOR	(null)	(null)	(null)	(null)	3	4
5	HASH JOIN	(null)	(null)	6	4	4	5
6	NESTED LOOPS	(null)	(null)	6	4	5	6
7	STATISTICS COLLECTOR	(null)	(null)	(null)	(null)	6	7
8	TABLE ACCESS	FULL	COMPRAS	3	4	7	8
9	TABLE ACCESS	BY INDEX ROWID	CLIENTE	3	1	6	9
10	INDEX	UNIQUE SCAN	SYS_C007781	(null)	(null)	9	10
11	TABLE ACCESS	FULL	CLIENTE	3	7	5	11
12	INDEX	RANGE SCAN	CLAVE_INVIRTE_PRIM	2	2	3	12
13	INDEX	FULL SCAN	CLAVE_INVIRTE_PRIM	2	4	2	13

Segunda ejecución de la misma consulta:

	OPERATION	OPTIONS	OBJECT_NAME	COST	CARDINALITY	PARENT_ID	ID
1	HASH	UNIQUE	(null)	2	1	0	1
2	NESTED LOOPS	SEMI	(null)	1	1	1	2
3	NESTED LOOPS	(null)	(null)	1	1	2	3
4	INDEX	SKIP SCAN	CLAVE_INVIRTE_PRIM	1	1	3	4
5	TABLE ACCESS	BY INDEX ROWID	CLIENTE	0	1	3	5
6	INDEX	UNIQUE SCAN	SYS_C007781	0	1	5	6
7	TABLE ACCESS	BY INDEX ROWID BATCHED	COMPRAS	0	1	2	7
8	INDEX	RANGE SCAN	SYS_C007803	0	1	7	8

B.- Contestar a las siguientes preguntas basándote en las operaciones de los resultados anteriores:

B1)

La diferencia está en:

Consulta 1: `select * from cliente where DNI < '00000005'`

Consulta 2: `select * from cliente where DNI = '00000005'`

En la consulta 2 simplemente accede a una sola fila única por índice único, mientras la primera lee más de una fila.

B2)

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

Porque no está accediendo por PK de moroso debería acceder por DNI de moroso para acceder por índice.

B3)

Accede a la clave primaria de cliente DNI porque es su PK

3	TABLE ACCESS	BY INDEX ROWID	CLIENTE
4	INDEX	UNIQUE SCAN	SYS_C007781

B4)

Porque busca por el dni que sea distinto a eso y por lo tanto va leer todas las filas de la tabla en vez de acceder a una única fila.

B5)

La 1 más eficiente ya que devolverá menos filas un < (que acortará el resultado final) que un <> que sacará más filas.

C).- Dibuja a mano el árbol del Plan de Ejecución: incluye el núm. de operación y su nombre en cada nodo. Incluye las condiciones de selección y las proyecciones de los atributos en el lugar que corresponda para obtener el árbol optimizado.

Miramos las tablas para ver cuántas filas nos devolvería cada una para obtener la condición más restrictiva. Nos fijamos que la condición de importe > 100 de la tabla compras nos devuelve 6 filas mientras que la condición de NOMBREE = ‘Empresa 55’ de la tabla invierte nos devuelve 4 filas, por lo tanto, la condición más restrictiva será NOMBREE = ‘Empresa 55’ porque nos devuelve menos filas.

Paso 1: para selecciones con condiciones conjuntivas, σ $c1 \wedge c2 \wedge ... cn(r)$, generamos selecciones con condiciones simples σ $c1$(σ $c2$(σ...cn(r))

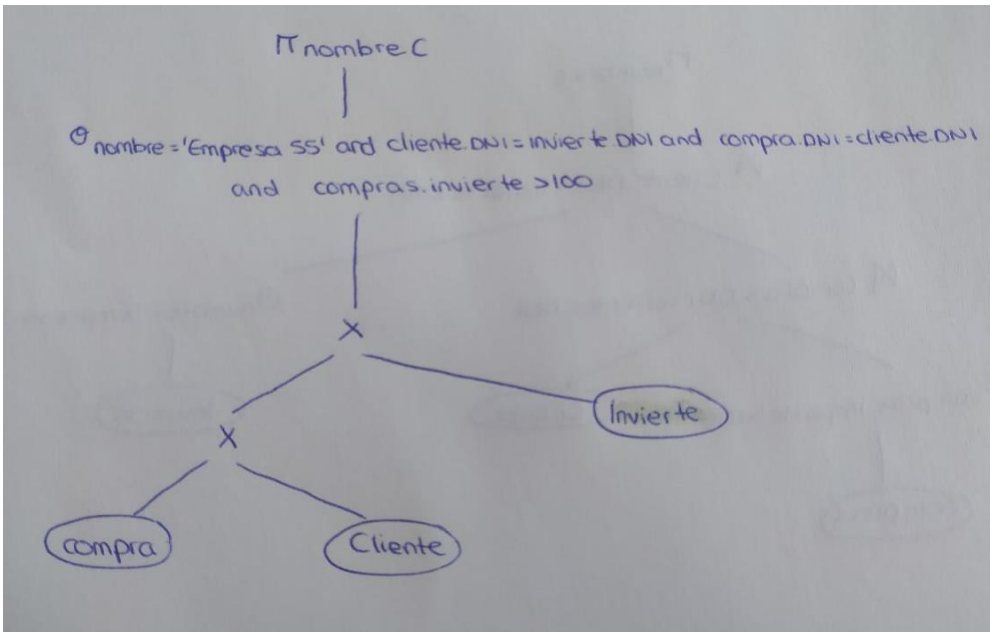
Paso 2: Ejecutar primero las selecciones: Bajándolas en el árbol. Operaciones σ más restrictivas hacia los nodos hoja. Asegurarse de que el orden de los nodos hoja no provocan operaciones “X”.

Paso 3: Poner juntas las selecciones y uniones

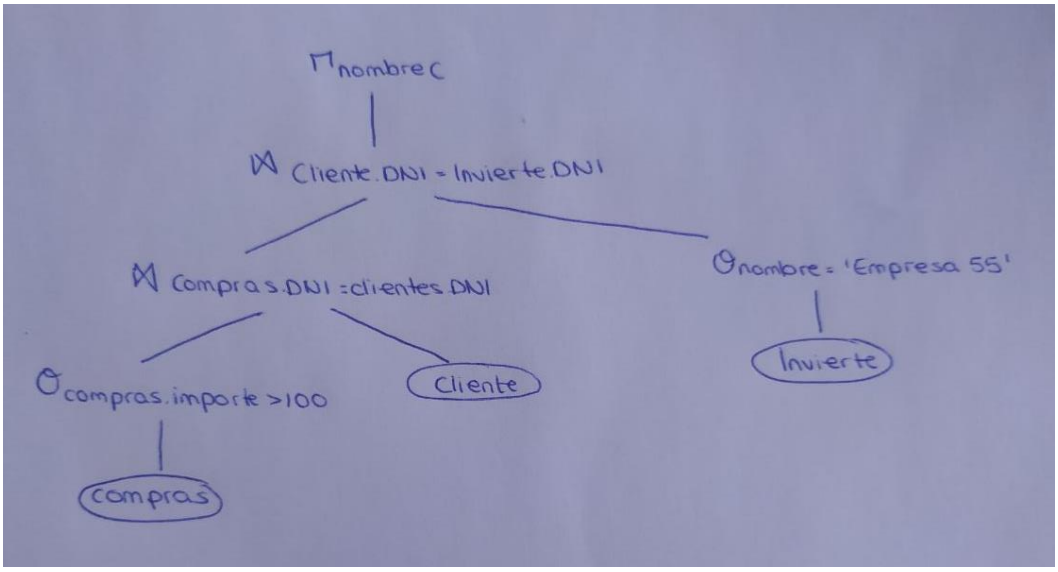
Paso 4: Cambiar (selección + prod. Cartesiano) por theta unión.

Paso 5: Deshacer Proyecciones con lista atributos de varias relaciones. Añadir π para evitar que suban atributos innecesarios.

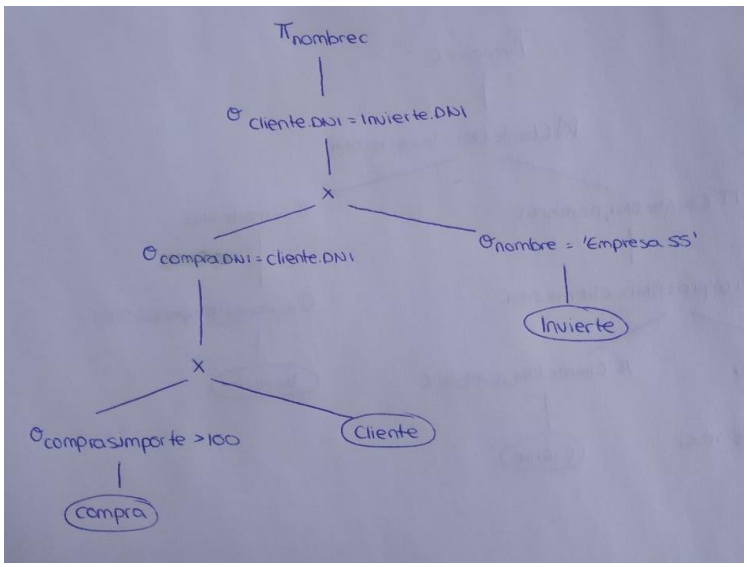
Con los pasos 1 y 2:



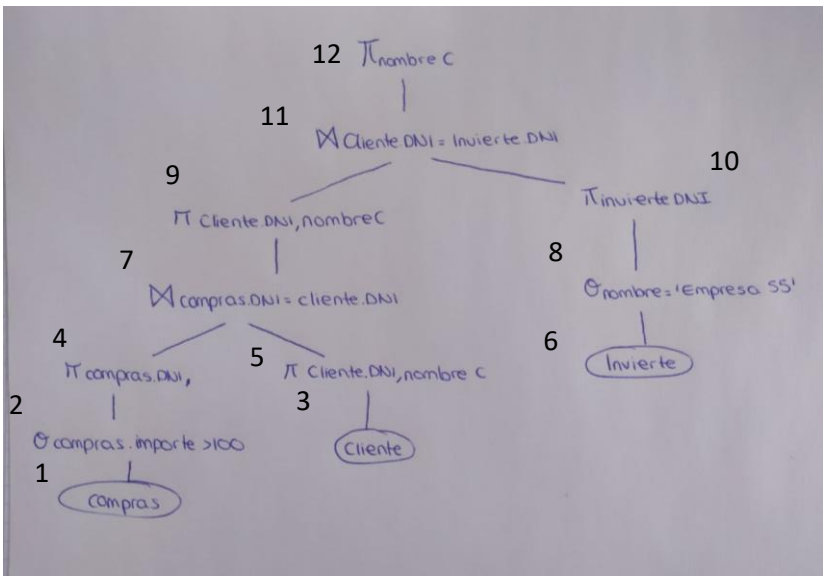
Con el paso 3:



Con el paso 4:



Con el paso 5:



D.- Compáralo con el árbol que muestra SQLDeveloper al marcar la consulta y pulsa F10. Indica qué diferencias hay?

(Esta imagen es tras la segunda ejecución de la instrucción)

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	2
HASH		UNIQUE	1	2
NESTED LOOPS		SEMI	1	1
NESTED LOOPS			1	1
INDEX	CLAVE_INVIRTE_PRIM	SKIP SCAN	1	1
Access Predicates				
INVIERTE.NOMBREE='Empresa 55'				
Filter Predicates				
INVIERTE.NOMBREE='Empresa 55'				
TABLE ACCESS	CLIENTE	BY INDEX ROWID	1	0
INDEX	SYS_C007781	UNIQUE SCAN	1	0
Access Predicates				
CLIENTE.DNI=INVIERTE.DNI				
TABLE ACCESS	COMPRAS	BY INDEX ROWID BATCHED	1	0
Filter Predicates				
COMPRAS.IMPORTE>1000				
INDEX	SYS_C007803	RANGE SCAN	1	0
Access Predicates				
COMPRAS.DNI=CLIENTE.DNI				

Se accede y se filtra por la tabla invierte, nombre = ‘Empresa 55’, luego se accede se busca en la tabla cliente mediante el dni obtenido por dicha empresa, esto se realiza en el bucle interno, en el externo se realiza el filtro por importe > 1000 en la tabla compras. Por último, buscará en compras por el dni obtenido en la tabla de cliente.

E.- Genera otro árbol con F6 y compáralo con el obtenido usando F10. Son ambas funciones iguales?. ¿Qué significan las diferencias?

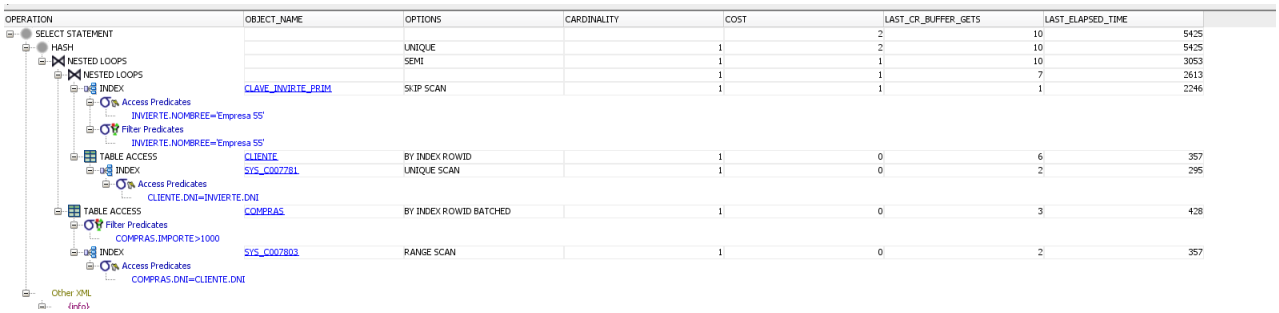
La diferencia encontrada está en las columnas de:

LAST_CR_BUFFER_GETS
Número de buffers recuperados en modo consistente, durante la última ejecución. Los buffers generalmente se recuperan en modo consistente para consultas.

LAST_ELAPSED_TIME

Tiempo transcurrido (en microsegundos) correspondiente a esta operación, durante la última ejecución.

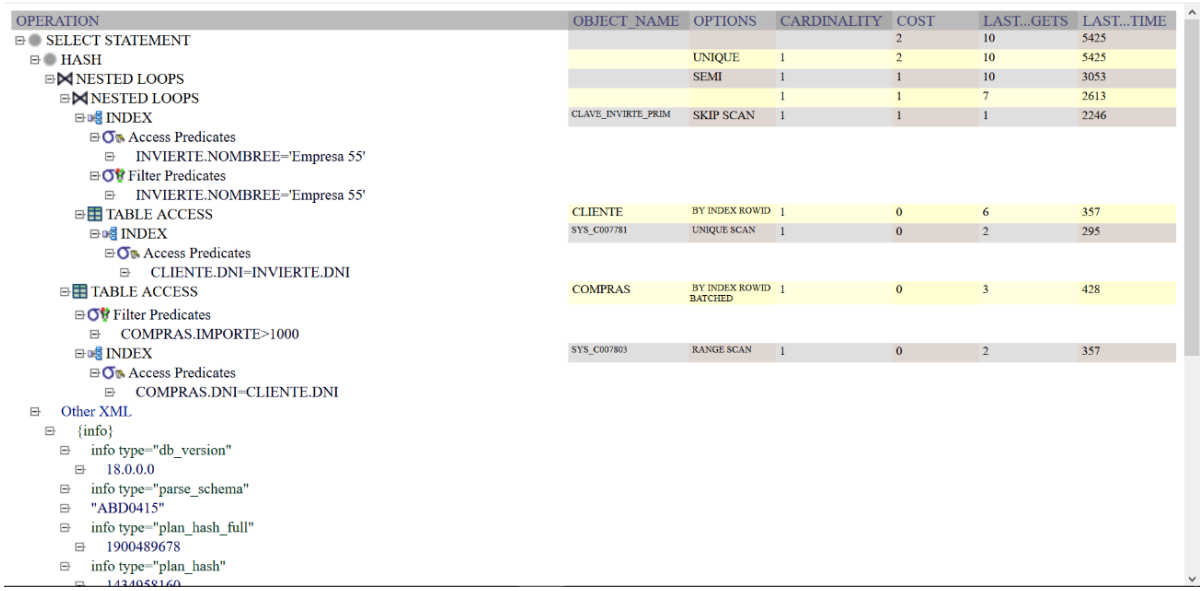
(Esta imagen es tras la segunda ejecución de la instrucción)



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	LAST_CR_BUFFER_GETS	LAST_ELAPSED_TIME
SELECT STATEMENT				2	10	5425
HASH		UNIQUE	1	2	10	5425
NESTED LOOPS		SEMI	1	1	10	3053
NESTED LOOPS			1	1	7	2613
INDEX	CLAVE_INVIRTE_PRIM	SKIP SCAN	1	1	1	2246
Access Predicates						
INVIERTE.NOMBREE='Empresa 55'						
Filter Predicates						
INVIERTE.NOMBREE='Empresa 55'						
TABLE ACCESS	CLIENTE	BY INDEX ROWID	1	0	6	357
INDEX	SYS_C007781	UNIQUE SCAN	1	0	2	295
Access Predicates						
CLIENTE.DNI=INVIERTE.DNI						
TABLE ACCESS	COMPRAS	BY INDEX ROWID BATCHED	1	0	3	428
Filter Predicates						
COMPRAS.IMPORTE>1000						
INDEX	SYS_C007803	RANGE SCAN	1	0	2	357
Access Predicates						
COMPRAS.DNI=CLIENTE.DNI						
Other XML						
<info>						

A continuación, adjunto el html de hacer f6 de este apartado para que se vea mejor:

(Esta imagen es tras la segunda ejecución de la instrucción)



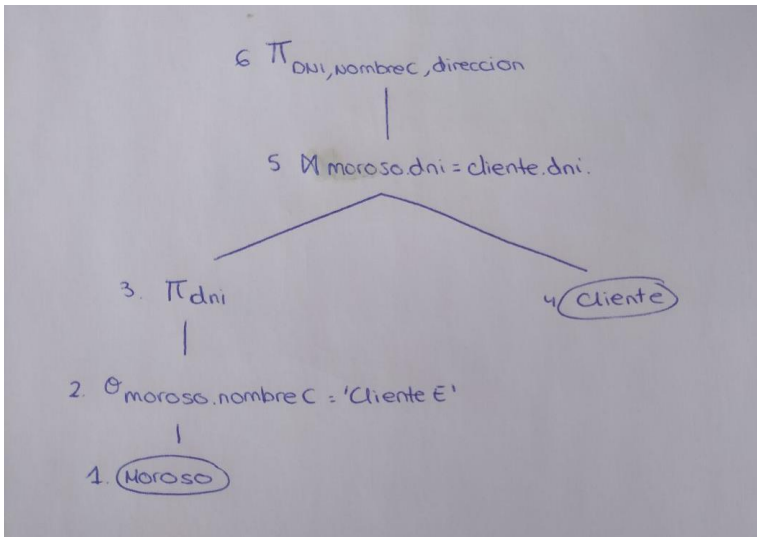
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	LAST...GETS	LAST...TIME
SELECT STATEMENT				2	10	5425
HASH		UNIQUE	1	2	10	5425
NESTED LOOPS		SEMI	1	1	10	3053
NESTED LOOPS			1	1	7	2613
INDEX	CLAVE_INVIRTE_PRIM	SKIP SCAN	1	1	1	2246
Access Predicates						
INVIERTE.NOMBREE='Empresa 55'						
Filter Predicates						
INVIERTE.NOMBREE='Empresa 55'						
TABLE ACCESS	CLIENTE	BY INDEX ROWID	1	0	6	357
INDEX	SYS_C007781	UNIQUE SCAN	1	0	2	295
Access Predicates						
CLIENTE.DNI=INVIERTE.DNI						
TABLE ACCESS	COMPRAS	BY INDEX ROWID BATCHED	1	0	3	428
Filter Predicates						
COMPRAS.IMPORTE>1000						
INDEX	SYS_C007803	RANGE SCAN	1	0	2	357
Access Predicates						
COMPRAS.DNI=CLIENTE.DNI						
Other XML						
{info}						
info type="db_version"						
18.0.0.0						
info type="parse_schema"						
"ABD0415"						
info type="plan_hash_full"						
1900489678						
info type="plan_hash"						
1434958160						

F.- *Exporta* a un fichero html el árbol del plan (botón dcho sobre raíz de salida de F10). (así lo puedes copiar al archivo para entregar)

(Esta imagen es tras la segunda ejecución de la instrucción)

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	2
HASH		UNIQUE	1	2
NESTED LOOPS		SEMI	1	1
NESTED LOOPS			1	1
INDEX	CLAVE_INVIRTE_PRIM	SKIP SCAN	1	1
Access Predicates				
INVIERTE.NOMBREE='Empresa 55'				
Filter Predicates				
INVIERTE.NOMBREE='Empresa 55'				
TABLE ACCESS	CLIENTE	BY INDEX ROWID	1	0
INDEX	SYS_C007781	UNIQUE SCAN	1	0
Access Predicates				
CLIENTE.DNI=INVIERTE.DNI				
TABLE ACCESS	COMPRAS	BY INDEX ROWID BATCHED	1	0
Filter Predicates				
COMPRAS.IMPORTE>1000				
INDEX	SYS_C007803	RANGE SCAN	1	0
Access Predicates				
COMPRAS.DNI=CLIENTE.DNI				
Other XML				
{info}				
info type="db_version"				
18.0.0.0				
info type="parse_schema"				
"ABD0415"				
info type="plan_hash_full"				
1900489678				
info type="plan_hash"				
1434958160				
info type="plan_hash_2"				
1900489678				
{hint}				
PARTIAL_JOIN(@"SEL$1" "COMPRAS"@"SEL$1")				
USE_HASH_AGGREGATION(@"SEL$1")				
USE_NL(@"SEL$1" "COMPRAS"@"SEL$1")				
USE_NL(@"SEL$1" "CLIENTE"@"SEL$1")				
LEADING(@"SEL$1" "INVIERTE"@"SEL$1" "CLIENTE"@"SEL$1" "COMPRAS"@"SEL$1")				
BATCH_TABLE_ACCESS_BY_ROWID(@"SEL$1" "COMPRAS"@"SEL$1")				
INDEX_RS_ASC(@"SEL$1" "COMPRAS"@"SEL$1" ("COMPRAS" "DNI" "COMPRAS" "NUMT" "COMPRAS" "NUMF"))				
INDEX_RS_ASC(@"SEL$1" "CLIENTE"@"SEL$1" ("CLIENTE" "DNI"))				
INDEX_SS(@"SEL$1" "INVIERTE"@"SEL$1" ("INVIERTE" "DNI" "INVIERTE" "NOMBREE" "INVIERTE" "TIPO"))				
OUTLINE_LEAF(@"SEL$1")				
ALL_ROWS				
DB_VERSION('18.1.0')				
OPTIMIZER_FEATURES_ENABLE('18.1.0')				
IGNORE_OPTIM_EMBEDDED_HINTS				

G.- *Dibuja* el árbol a mano de CONSULTA **3** igual que para la consulta 6, y obtén el árbol con F10:



Al hacer f10:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	3
NESTED LOOPS			1	3
NESTED LOOPS			1	3
TABLE ACCESS	MOROSO	FULL	1	3
Filter Predicates				
NOMBREC='Client E'				
INDEX	SYS_C007781	UNIQUE SCAN	1	0
Access Predicates				
DNI=DNI				
TABLE ACCESS	CLIENTE	BY INDEX ROWID	1	0
Other XML				
{info}				
info type="db_version"				
18.0.0.0				
info type="parse_schema"				
"ABD0415"				
info type="plan_hash_full"				
3942512477				
info type="plan_hash"				
700661067				
info type="plan_hash_2"				
3942512477				
{hint}				
NLJ_BATCHING(@"SEL$5DA710D3" "CLIENTE"@"SEL$1")				
USE_NL(@"SEL$5DA710D3" "CLIENTE"@"SEL$1")				
LEADING(@"SEL$5DA710D3" "MOROSO"@"SEL$2" "CLIENTE"@"SEL$1")				
INDEX(@"SEL$5DA710D3" "CLIENTE"@"SEL$1" ("CLIENTE"."DNI"))				
FULL(@"SEL$5DA710D3" "MOROSO"@"SEL$2")				

G1)

Las correspondientes columnas de estos nested loops están a null.

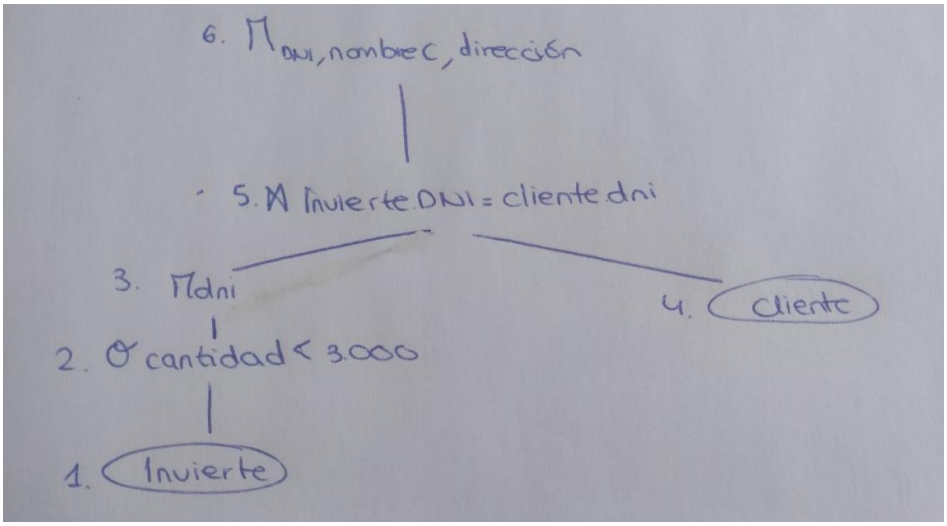
G2)

El nested loop interno es un bucle que busca en moroso el dni del que tenga NOMBREC=’Cliente E’ mientras que el nested loop externo es un bucle que busca en cliente el dni del moroso seleccionado

G3)

Porque hay dos bucles anidados que acceden a la tabla cliente y dni.

J.- *Dibuja* el árbol a mano de CONSULTA **5** igual que para en g.- (para pistas usa F10)



Al hacer f10:

(Imagen de la segunda ejecución)

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	3
NESTED LOOPS		SEMI	1	3
TABLE ACCESS	CLIENTE	FULL	1	3
TABLE ACCESS	INVIERTE	BY INDEX ROWID BATCHED	1	0
Filter Predicates				
CANTIDAD<30000				
INDEX	CLAVE_INVIRTE_PRIM	RANGE SCAN	1	0
Access Predicates				
DNI=DNI				

Other XML

```
{info}
  info type="db_version"
    18.0.0.0
  info type="parse_schema"
  "ABD0415"
  info type="plan_hash_full"
    4150203500
  info type="plan_hash"
    1363506709
  info type="plan_hash_2"
    4150203500
  {hint}
    USE_NL(@"SEL$5DA710D3" "INVIERTE"@"SEL$2")
    LEADING(@"SEL$5DA710D3" "CLIENTE"@"SEL$1" "INVIERTE"@"SEL$2")
    BATCH_TABLE_ACCESS_BY_ROWID(@"SEL$5DA710D3" "INVIERTE"@"SEL$2")
    INDEX_RS_ASC(@"SEL$5DA710D3" "INVIERTE"@"SEL$2" ("INVIERTE"."DNI" "INVIERTE"."NOMBREE" "INVIERTE"."TIPO"))
    FULL(@"SEL$5DA710D3" "CLIENTE"@"SEL$1")
    OUTLINE(@"SEL$2")
```

J1) El sort se realiza sobre la tabla invierte, en el atributo DNI.

J2) La tabla invierte y la de cliente.

J3) Si ordenamos por DNI al hacer la búsqueda podremos tardar menos ya que estos estarán ordenados y sabremos si tenemos que ir hacia abajo o hacia arriba.