

# PRÁCTICA 3

APARTADO 1 Preparar la BDejemplo: ejecuta BDejemplo.sql que viene en esta práctica.

Ejecutamos la BDejemplo.

APARTADO 2 Provocar que Oracle aplique reglas del optimizador diferentes.

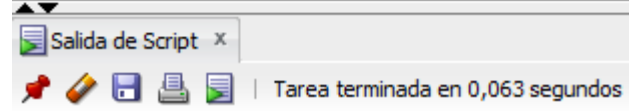
Se desea hacer un estudio comparativo de varias consultas, algunas ya vistas en la Teoría. Para ello, generamos el Plan de Ejecución de las consultas de Oracle, que se genera con el comando EXPLAIN.

Para ejecutarlas sigue estos pasos:

1. Se necesita crear una tabla donde Oracle almacena los resultados de las 'explicaciones'. Debe estar ya creada. Si al hacer los siguientes pasos no la encuentra, ejecuta este archivo: utlxplan.sql

Ejecutamos utlxplan.sql.

```
create table PLAN_TABLE (  
    statement_id    varchar2(30),  
    timestamp       date,  
    remarks         varchar2(80),  
    operation       varchar2(30),  
    options         varchar2(30),  
    object_node     varchar2(128),  
    object_owner    varchar2(30),  
    object_name     varchar2(30),  
    object_instance numeric,  
    object_type     varchar2(30),  
    optimizer       varchar2(255),  
    search_columns  number,  
    id             numeric,  
    parent_id      numeric,  
    position       numeric,  
    cost           numeric,  
    cardinality     numeric,  
    bytes          numeric,  
    other_tag       varchar2(255),  
    partition_start varchar2(255),  
    partition_stop  varchar2(255),  
    partition_id    numeric,  
    other          long);
```



Salida de Script x | Tarea terminada en 0,063 segundos

Table PLAN\_TABLE creado.

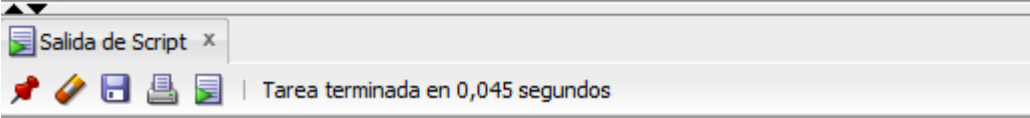
2. Para tener una máscara de edición que formatee el resultado del plan, ejecuta: (sale vacía ahora) MASCPPLAN13.sql.

Ejecutamos MASCPPLAN13.sql.

```
tttitle ' INFORME DEL PLAN '

col operation heading 'OPERACION' format a12 word_wrapped
col options heading 'OPCIONES' format a12 word_wrapped
col object_name heading 'TABLA' format a12 word_wrapped
col cost heading 'Coste' format a5
col cardinality heading 'Filas' format a5
col parent_id heading 'PADRE' format a5

select operation, options, object_name, cost, cardinality, parent_id, id
from plan_table
connect by prior id=parent_id      -- and statement_id= 'actuaT'
start with id = 1                 -- and statement_id= 'actuaT'
order by id;
```



Salida de Script x

Tarea terminada en 0,045 segundos

no se ha seleccionado ninguna fila

3. Ejecuta las siguientes explicaciones siguiendo estos pasos:

- En el SQL Developer con “Ejecutar Script” F5 para copiar los resultados en formato de texto (usa fuente courier new para que cuadren las columnas)
- las tres instrucciones para cada consulta: borrar lo que había, explicar consulta, ver el resultado.

-- CONSULTA 1 --

```
delete plan_table; /* primero borra las tuplas */
EXPLAIN PLAN
INTO plan_table
FOR (select * from cliente where DNI < '00000005') union
(select * from moroso where NombreC = 'Client E');
```

Ejecutar MASCPLAN13.sql

```
tttitle ' INFORME DEL PLAN '

col operation heading 'OPERACION' format a12 word_wrapped
col options heading 'OPCIONES' format a12 word_wrapped
col object_name heading 'TABLA' format a12 word_wrapped
col cost heading 'Coste' format a5
col cardinality heading 'Filas' format a5
col parent_id heading 'PADRE' format a5

select operation, options, object_name, cost, cardinality, parent_id, id
from plan_table
connect by prior id=parent_id      -- and statement_id= 'actuaT'
start with id = 1                  -- and statement_id= 'actuaT'
order by id;
```

Salida de Script x

Tarea terminada en 0,058 segundos

INFORME DEL PLAN						
OPERACION	OPCIONES	TABLA	Coste	Filas	PADRE	ID
SORT	UNIQUE		7	5	0	1
UNION-ALL					1	2
TABLE ACCESS BY INDEX ROWID		CLIENTE	2	4	2	3
INDEX	RANGE SCAN	SYS_C007374	2	1	3	4
TABLE ACCESS FULL		MOROSO	3	1	2	5

-- 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');
```

Ejecutar MASCPLAN13.sql

```
tttitle ' INFORME DEL PLAN '

col operation heading 'OPERACION' format a12 word_wrapped
col options heading 'OPCIONES' format a12 word_wrapped
col object_name heading 'TABLA' format a12 word_wrapped
col cost heading 'Coste' format a5
col cardinality heading 'Filas' format a5
col parent_id heading 'PADRE' format a5

select operation, options, object_name, cost, cardinality, parent_id, id
from plan_table
connect by prior id=parent_id      -- and statement_id= 'actuaT'
start with id = 1                  -- and statement_id= 'actuaT'
order by id;
```

Salida de Script x

Tarea terminada en 0,053 segundos

INFORME DEL PLAN						
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_C007374	1	1	3	4
TABLE ACCESS FULL		MOROSO	3	1	2	5

-- 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');
```

Ejecutar MASCPLAN13.sql

```
tttitle ' INFORME DEL PLAN '

col operation heading 'OPERACION' format a12 word_wrapped
col options heading 'OPCIONES' format a12 word_wrapped
col object_name heading 'TABLA' format a12 word_wrapped
col cost heading 'Coste' format a5
col cardinality heading 'Filas' format a5
col parent_id heading 'PADRE' format a5

select operation,options,object_name,cost,cardinality,parent_id,id
from plan_table
connect by prior id=parent_id      -- and statement_id= 'actuaT'
start with id = 1                  -- and statement_id= 'actuaT'
order by id;
```

Salida de Script x

Tarea terminada en 0,05 segundos

OPERACION	OPCIONES	TABLA	Coste	Filas	PADRE	ID
NESTED LOOPS					0	1
NESTED LOOPS			4	1	1	2
TABLE ACCESS	FULL	MOROSO	3	1	2	3
INDEX	UNIQUE SCAN	SYS_C007374	0	1	2	4
TABLE ACCESS	BY INDEX	CLIENTE	1	1	1	5
	ROWID					

-- CONSULTA 4 -- ----

```
delete plan_table;
EXPLAIN PLAN
INTO plan_table
FOR (select * from cliente where dni in
(select dni from invierte));
```

Ejecutar MASCPLAN13.sql

```
tttitle ' INFORME DEL PLAN '

col operation heading 'OPERACION' format a12 word_wrapped
col options heading 'OPCIONES' format a12 word_wrapped
col object_name heading 'TABLA' format a12 word_wrapped
col cost heading 'Coste' format a5
col cardinality heading 'Filas' format a5
col parent_id heading 'PADRE' format a5

select operation, options, object_name, cost, cardinality, parent_id, id
from plan_table
connect by prior id=parent_id      -- and statement_id= 'actuaT'
start with id = 1                  -- and statement_id= 'actuaT'
order by id;
```

Salida de Script x

Tarea terminada en 0,053 segundos

INFORME DEL PLAN						
OPERACION	OPCIONES	TABLA	Coste	Filas	PADRE	ID
HASH JOIN	SEMI		6	3	0	1
TABLE ACCESS	FULL	CLIENTE	3	6	1	2
INDEX	FAST FULL	CLAVE_INVIRT	2	8	1	3
	SCAN	E_PRIM				

-- CONSULTA 5 --

```
delete plan_table; /* borra las tuplas de explicación anterior*/
EXPLAIN PLAN
INTO plan_table
FOR (select * from cliente where dni in
(select dni from invierte where cantidad < 30000));
```

Ejecutar MASCPLAN13.sql

```
tttitle ' INFORME DEL PLAN '

col operation heading 'OPERACION' format a12 word_wrapped
col options heading 'OPCIONES' format a12 word_wrapped
col object_name heading 'TABLA' format a12 word_wrapped
col cost heading 'Coste' format a5
col cardinality heading 'Filas' format a5
col parent_id heading 'PADRE' format a5

select operation, options, object_name, cost, cardinality, parent_id, id
from plan_table
connect by prior id=parent_id      -- and statement_id= 'actuaT'
start with id = 1                  -- and statement_id= 'actuaT'
order by id;
```

Salida de Script x

Tarea terminada en 0,044 segundos

INFORME DEL PLAN						
OPERACION	OPCIONES	TABLA	Coste	Filas	PADRE	ID
-----						
NESTED LOOPS					0	1
NESTED LOOPS			5	1	1	2
SORT	UNIQUE		3	1	2	3
TABLE ACCESS	FULL	INVIERTE	3	1	3	4
INDEX	UNIQUE SCAN	SYS_C007374	0	1	2	5
TABLE ACCESS	BY INDEX	CLIENTE	1	1	1	6
	ROWID					

-- CONSULTA 6 --

```
delete plan_table;
EXPLAIN PLAN 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);
```

Ejecutar MASCPLAN13.sql

```
tttitle ' INFORME DEL PLAN '

col operation heading 'OPERACION' format a12 word_wrapped
col options heading 'OPCIONES' format a12 word_wrapped
col object_name heading 'TABLA' format a12 word_wrapped
col cost heading 'Coste' format a5
col cardinality heading 'Filas' format a5
col parent_id heading 'PADRE' format a5

select operation, options, object_name, cost, cardinality, parent_id, id
from plan_table
connect by prior id=parent_id      -- and statement_id= 'actuaT'
start with id = 1                 -- and statement_id= 'actuaT'
order by id;
```

Salida de Script x

Tarea terminada en 0,044 segundos

INFORME DEL PLAN						
OPERACION	OPCIONES	TABLA	Coste	Filas	PADRE	ID
HASH	UNIQUE		10	3	0	1
HASH JOIN			9	3	1	2
HASH JOIN			7	4	2	3
TABLE ACCESS	FULL	COMPRAS	3	4	3	4
TABLE ACCESS	FULL	CLIENTE	3	6	3	5
INDEX	FULL SCAN	CLAVE_INVIRT	2	4	2	6
		E_PRIM				

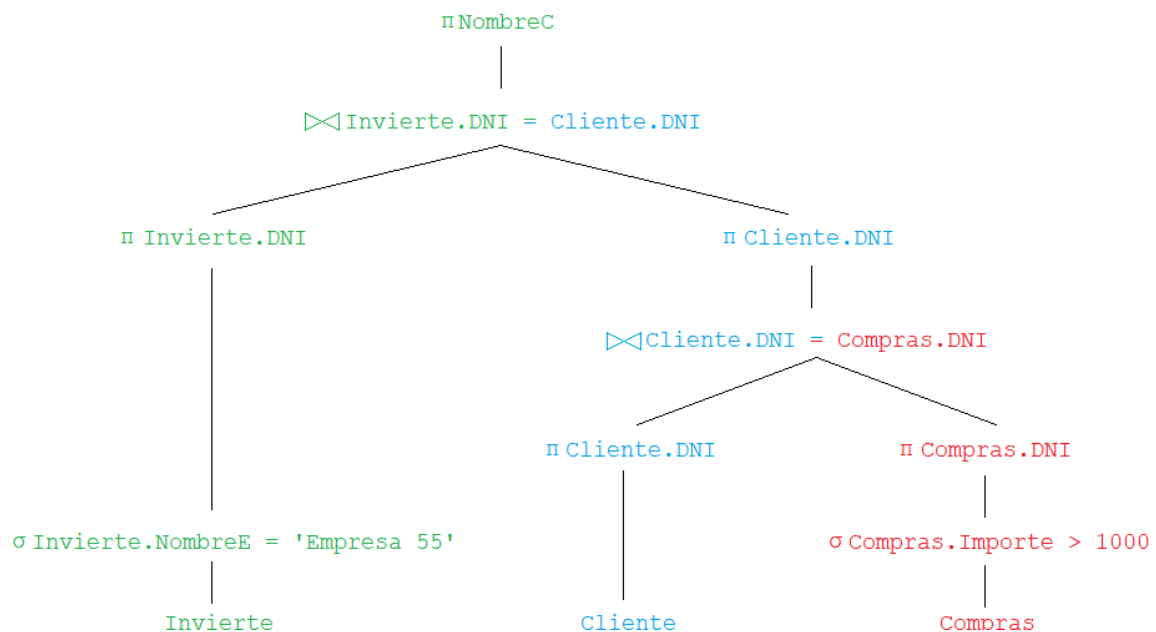


Entrega lo siguiente:

1. Para cada consulta:

- El resultado con las operaciones del explain plan (corta y pega con fuente courier new).
- Explica brevemente porqué ha ejecutado esas operaciones de bajo nivel y no otras.
- Solo para la consulta 6: Dibuja 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.

```
EXPLAIN PLAN 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);
```



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

(esta comparación es una “super-simplificación” de todo lo que hay que tener en cuenta para comparar dos consultas)

- Por qué CONSULTA 2 será más eficiente que CONSULTA 1?  
¿Cómo se sabe que es más eficiente?

Porque la CONSULTA 2 encuentra estrictamente 00000005. Mientras que la CONSULTA 1 tiene un rango mayor.

- Por qué en CONSULTA 1 no accede por el índice a CLIENTE?  
En la CONSULTA 1

- Por qué en CONSULTA 3 sí accede por el índice a CLIENTE?  
Es un select dentro de otro.  
`select DNI from moroso where NombreC = 'Client E'`

- Por qué en CONSULTA 3 no accede por el índice a MOROSO?  
Es un select dentro de otro.  
`select DNI from moroso where NombreC = 'Client E'`

c) Crea consultas nuevas (no vistas antes), con la BDEjemplo, que provoquen en Oracle las operaciones siguientes:

Full table scan  
`SELECT *  
FROM Clientes;`

Index unique scan  
`SELECT *  
FROM Cliente  
WHERE DNI = '00000001';`

Full Index scan  
`SELECT *  
FROM Cliente  
WHERE DNI = '00000001';`

Cartesian join  
`SELECT *  
FROM Emp, Dept  
WHERE Dept.deptno = 10  
AND Emp.sal > 10000;`

Crea otras consultas que provoquen en Oracle (ver Reglas de Optimización) que:

- NO use los índices

```
SELECT *
  FROM Emp, Dept
 WHERE Dept.deptno = 10
        AND Emp.sal > 10000;
```

- Sí use índices

```
SELECT *
  FROM Emp, Dept
 WHERE Dept.deptno = 10
        SORT Emp.sal;
```

d) Ejecutar en sql Developer: con icono “Rastreo Automático (F6)” la consulta 6.

- Compara con la salida obtenida con MASCPLAN13.sql:

¿Son ambas funciones iguales?

Da error.

INFORME DEL PLAN						
OPERACION	OPCIONES	TABLA	Coste	Filas	PADRE	ID
HASH	UNIQUE		9	4	0	1
HASH JOIN			8	4	1	2
MERGE JOIN			4	4	2	3
TABLE ACCESS	BY INDEX	CLIENTE	2	6	3	4
	ROWID					
INDEX	FULL SCAN	SYS_C007374	1	6	4	5
SORT	JOIN		2	4	3	6
INDEX	SKIP SCAN	CLAVE_INVIRT	1	4	6	7
		E_PRIM				

INFORME DEL PLAN						
OPERACION	OPCIONES	TABLA	Coste	Filas	PADRE	ID
TABLE ACCESS	FULL	COMPRAS	3	4	2	8

8 filas seleccionadas.

- Incluye en la entrega: un fichero html con el árbol del plan (botón dcho sobre raíz de salida de F6).

```
<TABLE class="TreeTableWidget" border=0 cellpadding=0 cellspacing=0 width=100% bgcolor=#efefef><tr><td>
```

```
<!-- .....data begin..... -->
```

```
<TABLE class=TreeTableWidget border=0 cellspacing=0 cellpadding=1 align=center width=100%" bgcolor=#ffffff>
```

```
<tr class="TableCell">
```

```
<td valign=top bgcolor=#bbbbbb>&nbsp;<font color=#444477>OPERATION</font>&nbsp;</td>
```

```
<td valign=top bgcolor=#aaaaaa>&nbsp;<font color=#444477>OBJECT_NAME</font>&nbsp;</td>
```

```
<td valign=top bgcolor=#bbbbbb>&nbsp;<font color=#444477>OPTIONS</font>&nbsp;</td>
```

```
<td valign=top bgcolor=#aaaaaa>&nbsp;<font color=#444477>CARDINALITY</font>&nbsp;</td>
```

```
<td valign=top bgcolor=#bbbbbb>&nbsp;<font color=#444477>COST</font>&nbsp;</td>
```

```
</tr>
```

```
<tr>
```

```
<td width=45% valign=top bgcolor=#ffffff ><font size=-1><table width=100% border=0 cellspacing=0 cellpadding=0><tr><td width=1%><table CELLSPACING=0 cellpadding=0><tr><td><IMG SRC="images/minus.gif" wWIDTH=16 hHEIGHT=16 BORDER=0><IMG SRC="images/gray.png" wWIDTH=16 hHEIGHT=16 BORDER=0></td><tr></table></td> <td width=99%>&nbsp;<DELETE STATEMENT&nbsp;</td></tr></table></font></td>
```

```
<td width=11% valign=top bgcolor=#dfdfdf >&nbsp;&nbsp;&nbsp;</td>
```

```
<td width=11% valign=top bgcolor=#dfd5d1 >&nbsp;&nbsp;&nbsp;</td>
```

```
<td width=11% valign=top bgcolor=#dfdfdf >&nbsp;&nbsp;7</font></td>
```

```
<td width=11% valign=top bgcolor=#dfd5d1 >&nbsp;&nbsp;3</font></td>
```

```
</tr>
```

```
<tr>
```

```
<td width=45% valign=top bgcolor=#ffffff ><table width=100% border=0 cellspacing=0 cellpadding=0><tr><td width=1%><table CELLSPACING=0 cellpadding=0><tr><td>&nbsp;&nbsp;&nbsp;&nbsp;</td><td><IMG SRC="images/minus.gif" wWIDTH=16 hHEIGHT=16 BORDER=0><IMG
```

[illegible]

```
-- CONSULTA 4 --
select HECTOR.PELISAHORA.TITULO
from HECTOR.PELISAHORA
where HECTOR.PELISAHORA.TITULO in (select
HECTOR.PELISHIST.TITULO from
HECTOR.PELISHIST);
```

```
-- CONSULTA 5 --
select HECTOR.PELISAHORA.TITULO
from HECTOR.PELISAHORA
where HECTOR.PELISAHORA.TITULO in
(select HECTOR.PELISHIST.TITULO from HECTOR.PELISHIST
where HECTOR.PELISAHORA.TITULO = HECTOR.PELISHIST.TITULO);
```

(Entrega lo que se pide en cada apartado)

- Esta tabla en Word con los datos de los planes para las consultas, una fila por consulta.
- Indica Qué observas en la relación de cada operación, su Coste y sus Filas utilizadas?

CONSULTA	Coste (total)	Num. Filas (total)	Num. Operaciones	¿Qué observas?
1	2	285	3	
2	107	627	3	
3	4	483	3	
4	7	823	4	
5	7	823	4	

b) Explicar cuál es más eficiente si tienes en cuenta sólo el coste y las filas usadas.

La primera por tener el menor coste.

#### APARTADO 4. (EXTRA)

Entregar las instrucciones necesarias en sql y los resultados de las consultas.  
Queremos hacer un diccionario que relacione jerárquicamente los conceptos. Ej.: el concepto "select" con todos los tipos de select. Para ello hacer lo siguiente:

- a) Crear una tabla diccionario DICCION, que tenga los siguientes atributos:
- PalID , será como máximo de 20 caracteres. Identifica la palabra.
  - Descripción, de 50 caracteres.
  - PadreID, de 20 char. Representa un concepto más genérico que PalID, en las filas insertadas en b) se ve que 'select compuesta' es el PadreID de 'select jerarquica' y de 'select correlativa'

```
CREATE TABLE DICCION (  
    PalID          varchar2(20) ,  
    Descrip        varchar2(50) ,  
    PadreID        varchar2(50) ,  
    PRIMARY KEY (PalID)  
);
```

- b) Incluir las siguientes filas: (estas comillas son del word, no válidas en oracle)
- ('select jerarquica','estructura tabla en arbol','select compuesta');
- ('fecha sistema','es la fecha que tiene el ordenador','fecha');
- ('fecha','tipo de dato , en oracle : DATE','nada');
- ('select compuesta','consultas con varias partes','select');
- ('select simple','consultas con una sola instruccion','select');
- ('select','hacer consulta','nada');
- ('sql','lenguaje de consultas estructuradas','nada');
- ('select correlativa','coordina resultado subconsulta','select compuesta');

```
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','consultas  
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');
```

c) Hacer una consulta jerárquica conectada por PalID y PadreID que empiece con la palabra 'select' (es como la de 'mascpian.sql').

```
select PALID2, DESCRIP2, PADREID2
  from DICCION
 start with PALID = 'select'
 connect by prior id = PADREID;
```

d) Insertar una fila con valores PalID = 'select anidada', descripción = 'consulta dentro de consulta' y PadreID = 'select compuesta'. Lo importante de esta inserción es que se quiere hacer solo en el caso que el padre exista, es decir solo hacerla en caso de que una consulta de PalID='select compuesta' devuelve algo. Si no devuelve nada no se debe crear. (es una inserción condicionada).