

Consulta 1)

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi

NESTED LOOPS			1	143	3718	0	1
INDEX	FULL SCAN	PK_PELISACTUAL	1	143	1859	1	2
INDEX	UNIQUE SCAN	PK_PELISHIST	0	1	13	1	3

Consulta 2)

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi

HASH JOIN			87	152	608608	0	1
TABLE ACCESS	FULL	PELISAHORA	19	143	286286	1	2
TABLE ACCESS	FULL	PELISHIST	68	484	968968	1	3

Consulta 3)

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi

NESTED LOOPS			4	143	18876	0	1
INDEX	FAST FULL SCAN	NOR_IND_PELISHIST	4	484	31944	1	2
INDEX	RANGE SCAN	NOR_IND_PELISAHORA	0	1	66	1	3

Consulta 4)

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi

NESTED LOOPS			5	143	18876	0	1
SORT	UNIQUE		4	484	31944	1	2
INDEX	FAST FULL SCAN	NOR_IND_PELISHIST	4	484	31944	2	3
INDEX	RANGE SCAN	NOR_IND_PELISAHORA	0	1	66	1	4

Consulta 5)

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste Filas		BYTES PADRE	Id_Fi	
-----	-----	-----	-----	-----	-----	-----	-----
NESTED LOOPS			5	143	18876	0	1
SORT	UNIQUE		4	484	31944	1	2
INDEX	FAST FULL	NOR_IND_PELI	4	484	31944	2	3
	SCAN	SHIST					
INDEX	RANGE SCAN	NOR_IND_PELI	0	1	66	1	4
		SAHORA					

A)

CONSULTA	COSTE(TOTAL)	NUM.FILAS(TOTAL)	NUM.OPERACIONES	BYTES(TOTAL)
C1	1	287	3	5590
C2	87	779	3	1863862
C3	4	628	3	50886
C4	5	1112	4	82830
C5	5	1112	4	82830

A1)

Porque accede por un índice que es su PK.

A2)

Tal y como se expresa en la teoría, el criterio principal será escoger el plan de ejecución más óptimo (el que tenga menos coste) para conseguir una mejor eficiencia.

A3)

Que en la consulta 1 estamos buscando por su pk mientras que en la consulta 2 busca por la descripción. Por lo tanto, esto hará que haga una búsqueda completa a la tabla y nos devuelva más filas aumentando su coste.

A4)

Porque C3 accede por índice (titulo) el cual hemos creado al principio

A5)

Porque al no acceder por índice tiene que hacer una búsqueda más completa

A6)

La explicación aprendida en la teoría sería que el hash join se usa cuando se une una tabla pequeña a otra bastante más grande, usando la menor para crear una tabla hash en memoria con la clave join.

A7)

La consulta 4 hace un sort por el where in mientras que la 3 realiza un join.

A8)

Porque hacer una consulta añadiendo una condición de where no implica que tenga que haber una operación más de bajo nivel.

B)

B.1) Sólo filas y coste

Las más eficientes serían C1 y C3, ya que al devolver menos filas tendrán un coste menor. Entre la C1 y C3 la más eficiente sería C1 ya que tiene un coste de 1 y un total de filas de 287.

B.2) Sólo bytes y coste

Las más eficientes vuelve a ser C1 y C3, siendo la más eficientes entre las dos C1 ya que tiene un coste de 1 y 5590 bytes totales, mientras que C3 tiene 4 de coste y 50886 bytes.

C)

C1)

```
CREATE BITMAP INDEX NOR_IND_GENERO ON PELISAHORA(GENERO);
```

```
select titulo, round(drama) from pelisahora
```

```
where genero = 'Drama' and round(drama) > 35
```

Plan de ejecución de consulta:

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi
TABLE ACCESS BY INDEX ROWID BATCHED		PELISAHORA	2	24	2256	0	1
BITMAP CONVERSION TO ROWIDS						1	2
BITMAP INDEX SINGLE VALUE		NOR_IND_GENE				2	3
		RO					

Resultado de ejecutar consulta:

	TITULO	ROUND(DRAMA)
1	Criadas y señoras (The help)	42
2	La voz dormida	55
3	Margin call	54
4	5 metros cuadrados	64
5	Anonymous	45
6	La saga Crepúsculo: Amanecer - Parte 1	50
7	Todos los días de mi vida	56
8	Miel de naranjas	43
9	¡Por fin solos! (2012)	39
10	Adiós a la reina	41
11	Cuando te encuentre	50
12	Tengo ganas de ti	50
13	El fraude	37
14	Elefante blanco	47
15	Fénix 11.23	43
16	Amor	53
17	Golpe de efecto	57
18	La saga Crepúsculo: Amanecer - Parte 2	42
19	Persiguiendo Mavericks	40
20	El lado bueno de las cosas	43
21	El vuelo (Flight)	37
22	Volver a nacer	56
23	El bosque (El bosc)	42

C2)

1.-

INFORME DEL PLAN

OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi
TABLE ACCESS FULL		PELISAHORA	19	1	79	0	1

2.-

CREATE INDEX ROUNDDRAMA ON PELISAHORA(ROUND(drama));

3.-

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi
TABLE ACCESS	BY INDEX	PELISAHORA	1	1	79	0	1
	ROWID						
	BATCHED						
INDEX	RANGE SCAN	ROUNDDRAMA	1	1		1	2

4.-

No accede por Index ya que no tiene un = si no un >.

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi
TABLE ACCESS	FULL	PELISAHORA	19	40	3160	0	1

D)

D1)

select count(round(drama)) from pelisahora;

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi
TABLE ACCESS	BY INDEX	PELISAHORA	1	1	79	0	1
	ROWID						
INDEX	RANGE SCAN	ROUNDDRAMA	1	143	1859	1	2

D2)

select * from pelisahora where id= 1

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi
TABLE ACCESS	BY INDEX	PELISAHORA	1	1	2239	0	1
	ROWID						
INDEX	UNIQUE SCAN	PK_PELISACTUAL	0	1		1	2

D3)

```
select id from pelisahora;
```

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi
INDEX	FULL SCAN	PK_PELISACTUAL	1	142	568	0	1

D4)

```
SELECT * FROM PELISAHORA, PELISHIST;
```

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi
MERGE JOIN	CARTESIAN		9263	73982	348011328	0	1
TABLE ACCESS	FULL	PELISHIST	68	521	1160788	1	2
BUFFER	SORT		9195	142	351592	1	3
TABLE ACCESS	FULL	PELISAHORA	18	142	351592	3	4

D5)

```
select PELISAHORA.DESCRIPCION
```

```
from PELISAHORA, pelishist
```

```
where PELISAHORA.genero = pelishist.genero and pelisahora.genero = 'accion' and  
pelisahora.accion >50
```

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi
TABLE ACCESS	FULL	PELISHIST	68	11	99	1	5

D6)

```
select PELISAHORA.DESCRIPCION
```

```
from PELISAHORA, pelishist
```

```
where PELISAHORA.titulo = pelishist.titulo and pelisahora.genero = 'accion' and  
pelisahora.accion >50;
```

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste Filas		BYTES PADRE	Id_Fi	

CONVERSION							
BITMAP INDEX SINGLE VALUE		NOR_IND_GENE RO				5	6
INDEX	RANGE SCAN	NOR_IND_PELI SHIST	1	1	21	2	7
INDEX	FAST FULL SCAN	NOR_IND_PELI SHIST	1	1	21	1	8

E)

INFORME DEL PLAN							
OPERACION	OPCIONES	TABLA	Coste	Filas	BYTES	PADRE	Id_Fi
-----	-----	-----	-----	-----	-----	-----	-----
COLLECTION ITERATOR	PICKLER FETCH	DISPLAY	29	8168	16336	0	1

Consulta 1

INFORME DEL PLAN
PLAN_TABLE_OUTPUT

Plan hash value: 3786695621

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	

0	SELECT STATEMENT		142	1136	1 (0)	00:00:01	
1	NESTED LOOPS		142	1136	1 (0)	00:00:01	
2	INDEX FULL SCAN	PK_PELISACTUAL	142	568	1 (0)	00:00:01	
* 3	INDEX UNIQUE SCAN	PK_PELISHIST	1	4	0 (0)	00:00:01	

INFORME DEL PLAN
PLAN_TABLE_OUTPUT

Predicate Information (identified by operation id):

3 - access("PELISAHORA"."ID"="PELISHIST"."ID")

15 filas seleccionadas.

Consulta 2

INFORME DEL PLAN
PLAN_TABLE_OUTPUT

Plan hash value: 2200754562

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		144	643K	87 (0)	00:00:01
* 1	HASH JOIN		144	643K	87 (0)	00:00:01
2	TABLE ACCESS FULL	PELISAHORA	142	334K	19 (0)	00:00:01
3	TABLE ACCESS FULL	PELISHIST	521	1102K	68 (0)	00:00:01

INFORME DEL PLAN
PLAN_TABLE_OUTPUT

Predicate Information (identified by operation id):

1 - access("PELISAHORA"."DESCRIPCION"="PELISHIST"."DESCRIPCION")

15 filas seleccionadas.

Consulta 3

INFORME DEL PLAN
PLAN_TABLE_OUTPUT

Plan hash value: 3928350524

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		142	6106	3 (0)	00:00:01
1	NESTED LOOPS		142	6106	3 (0)	00:00:01
2	INDEX FAST FULL SCAN	NOR_IND_PELISHIST	521	10941	3 (0)	00:00:01
* 3	INDEX RANGE SCAN	NOR_IND_PELISAHORA	1	22	0 (0)	00:00:01

INFORME DEL PLAN
PLAN_TABLE_OUTPUT

Predicate Information (identified by operation id):

3 - access("PELISAHORA"."TITULO"="PELISHIST"."TITULO")

15 filas seleccionadas.

Consulta 4

INFORME DEL PLAN
PLAN_TABLE_OUTPUT

Plan hash value: 2674791169

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		142	6106	4 (25)	00:00:01
1	NESTED LOOPS		142	6106	4 (25)	00:00:01
2	SORT UNIQUE		521	10941	3 (0)	00:00:01
3	INDEX FAST FULL SCAN	NOR_IND_PELISHIST	521	10941	3 (0)	00:00:01
* 4	INDEX RANGE SCAN	NOR_IND_PELISAHORA	1	22	0 (0)	00:00:01

INFORME DEL PLAN
PLAN_TABLE_OUTPUT

Predicate Information (identified by operation id):

4 - access("PELISAHORA"."TITULO"="PELISHIST"."TITULO")

16 filas seleccionadas.

Consulta 5

INFORME DEL PLAN
PLAN_TABLE_OUTPUT

Plan hash value: 2674791169

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		142	6106	4 (25)	00:00:01
1	NESTED LOOPS		142	6106	4 (25)	00:00:01
2	SORT UNIQUE		521	10941	3 (0)	00:00:01
3	INDEX FAST FULL SCAN	NOR_IND_PELISHIST	521	10941	3 (0)	00:00:01
* 4	INDEX RANGE SCAN	NOR_IND_PELISAHORA	1	22	0 (0)	00:00:01

INFORME DEL PLAN
PLAN_TABLE_OUTPUT

Predicate Information (identified by operation id):

4 - access("PELISAHORA"."TITULO"="PELISHIST"."TITULO")

16 filas seleccionadas.

APARTADO 4

- A) create table DICCION(PalID varchar2(20), Descripcion varchar2(50), PadreID varchar2(20));

B) 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 correlativ','coordina resultado subconsulta', 'select compuesta');

C) select palid,descripcion,padreid

from diccion

connect by prior palid=padreid -- conectada por PalID y PadreID

start with palid = 'select' -- empieza por la palabra select

order by padreid;

	⚡ PALID	⚡ DESCRIPCION	⚡ PADREID
1	select	hacer consulta	nada
2	select compuesta	consultas con varias partes	select
3	select simple	consultas con una sola instruccion	select
4	select jerarquica	estructura tabla en arbol	select compuesta
5	select correlativ	coordina resultado subconsulta	select compuesta

INFORME DEL PLAN

PALID	DESCRIPCION	PADREID
select	hacer consulta	nada
select compuesta	consultas con varias partes	select
select correlativ	coordina resultado subconsulta	select compuesta
select jerarquica	estructura tabla en arbol	select compuesta
select simple	consultas con una sola instruccion	select

D)

INSERT INTO DICCION select 'select anidada','consulta dentro de consulta','select compuesta' from dual WHERE EXISTS (SELECT * FROM DICCION WHERE padreid = 'select compuesta');

E)

PALID	DESCRIPCION	PADREID
select	hacer consulta	nada
→ select anidada	consulta dentro de consulta	select compuesta
select compuesta	consultas con varias partes	select
select correlativ	coordina resultado subconsulta	select compuesta
select jerarquica	estructura tabla en arbol	select compuesta
select simple	consultas con una sola instruccion	select