Alejandro Ruiz Martín

PRACTICA 4 – SEMANA 9

**APARTADO 3**

Consulta 1:

```
select PELISAHORA.ID
from PELISAHORA, PELISHIST
where PELISAHORA.ID = PELISHIST.ID;
```

```
INFORME DEL PLAN
OPERACION    OPCIONES     TABLA        Coste Filas      BYTES PADRE Id_Fi
------------ ------------ ------------ ----- ----- ---------- ----- -----
NESTED LOOPS                              1   142       3692     0     1
INDEX        FULL SCAN    SYS_C007495     1   142       1846     1     2
INDEX        UNIQUE SCAN  SYS_C007496     0     1         13     1     3
```

Consulta 2:

```
select PELISAHORA.DESCRIPCION
from PELISAHORA, PELISHIST
where PELISAHORA.DESCRIPCION = PELISHIST.DESCRIPCION;
```

```
INFORME DEL PLAN
OPERACION    OPCIONES     TABLA        Coste Filas      BYTES PADRE Id_Fi
------------ ------------ ------------ ----- ----- ---------- ----- -----
HASH JOIN                                73   142     568568     0     1
TABLE ACCESS FULL         PELISAHORA     18   142     284284     1     2
TABLE ACCESS FULL         PELISHIST      55   588    1177176     1     3
```

Consulta 3:

```
select PELISAHORA.TITULO
from PELISAHORA, PELISHIST
where PELISAHORA.TITULO = PELISHIST.TITULO;
```

```
INFORME DEL PLAN
OPERACION    OPCIONES     TABLA        Coste Filas      BYTES PADRE Id_Fi
------------ ------------ ------------ ----- ----- ---------- ----- -----
NESTED LOOPS                              1   142      18744     0     1
INDEX        FULL SCAN    PELISA_INDEX    1   142       9372     1     2
INDEX        UNIQUE SCAN  PELISH_INDEX    0     1         66     1     3
```

Consulta 4:

```
select PELISAHORA.TITULO
from PELISAHORA
where PELISAHORA.TITULO in (select PELISHIST.TITULO from PELISHIST);
```

```
INFORME DEL PLAN
OPERACION    OPCIONES     TABLA        Coste Filas      BYTES PADRE Id_Fi
------------ ------------ ------------ ----- ----- ---------- ----- -----
NESTED LOOPS                              1   142      18744     0     1
INDEX        FULL SCAN    PELISA_INDEX    1   142       9372     1     2
INDEX        UNIQUE SCAN  PELISH_INDEX    0     1         66     1     3
```

Consulta 5:

```
select PELISAHORA.TITULO
from PELISAHORA
where PELISAHORA.TITULO in
(select PELISHIST.TITULO from PELISHIST
where PELISAHORA.TITULO = PELISHIST.TITULO);
```

```
INFORME DEL PLAN
OPERACION    OPCIONES     TABLA        Coste Filas      BYTES PADRE Id_Fi
------------ ------------ ------------ ----- ----- ---------- ----- -----
NESTED LOOPS                              1   142      18744     0     1
INDEX        FULL SCAN    PELISA_INDEX    1   142       9372     1     2
INDEX        UNIQUE SCAN  PELISH_INDEX    0     1         66     1     3
```

**APARTADO A**

**A.0.** Tabla de planes

| CONSULTA | Coste (total) | Num. Filas (total) | Num. Operaciones | Bytes (total) |
|---|---|---|---|---|
| C1 | 1 | 285 | 3 | 5551 |
| C2 | 73 | 872 | 3 | 2030028 |
| C3 | 1 | 284 | 3 | 28149 |
| C4 | 1 | 284 | 3 | 28149 |
| C5 | 1 | 284 | 3 | 28149 |

**A.1.** Sobre C1: ¿Por qué no accede a la tabla PELISHIST?

SOL: Al crear las tablas y definir sus PK, genera un índice en ambas tablas que guarda la columna de los ID de las tablas, por lo que mira dentro del índice “SYS_C007495” los ID que coinciden

**A.2**. Sobre C1: ¿Cuál es el criterio principal para escoger un plan de ejecución u otro?

SOL: El plan de ejecución que resulte más eficiente en cuanto a coste

**A.3.** Compara C1 y C2: ¿Qué problema hay con los índices para que el coste sea tal alto en C2?

SOL: Que los índices que tienen solo engloban uno la columna del ID y el generado por nosotros la del Nombre. En esta consulta no utiliza ninguno de esos dos campos, por lo que debe cargar toda la tabla y hacer Join por la columna de Descripción

**A.4.** Compara C2 y C3: ¿Por qué, en C2, accede a las tablas completas?

SOL: al no tener el campo "descripción" ningún índice, como se explica en el apartado a.3, necesita cargar las tablas completas, ya que todas las operaciones se hacen con esas columnas

**A.5.** Compara C2 y C3: ¿Porqué, en C2, el coste es tal alto?

SOL: El coste es tan elevado porque tiene que hacer cargar las dos tablas completas y hacer Join entre ellas por el campo "descripción"

**A.6.** Compara C2 y C3: ¿En C2, porqué el Hash Join solo usa 142 filas si ha leído las 521 filas de pelishist?

SOL: El Join lo hace aplicándolo a la tabla "PELISAHORA", que tiene 142 filas, no necesita hacer Join en todas las filas de "PELISHIST",

**A.7.** Compara C3 y C4: ¿Por qué ambas NO hacen las mismas operaciones de bajo nivel?

SOL: Salen las mismas

**A.8.** Compara C4 y C5: ¿Por qué ambas SI hacen las mismas operaciones de bajo nivel?

SOL: Las consultas son muy parecidas añadiendo una condición más en el segundo select, usando datos ya recopilados.

**APARTADO B**

**B.1.** Consultas más eficientes teniendo en cuenta sólo el *coste* y las *filas usadas*.

SOL: La consulta 3,4 o 5 son más eficientes según su coste y el número de filas utilizado

**B.2.** Consultas más eficientes teniendo en cuenta sólo el *coste* y los *bytes usadas*.

SOL: La consulta 1 es la más eficiente según su coste y los bytes usados.

**APARTADO C**

* Adjunto los .html en la entrega

**C.1.**

1) Índice:

```
CREATE INDEX PELISA_DrIndex ON PELISAHORA (ROUND(DRAMA));
```

2) Plan con índice:

```
INFORME DEL PLAN
OPERACION    OPCIONES     TABLA        Coste Filas      BYTES PADRE Id_Fi
------------ ------------ ------------ ----- ----- ---------- ----- -----
TABLE ACCESS BY INDEX     PELISAHORA       2     7        245     0     1
             ROWID
             BATCHED

INDEX        RANGE SCAN   PELISA_DRIND     1     1                1     2
                          EX
```

---

**C.2.**

1) Plan sin índice (EJ):

```
INFORME DEL PLAN
OPERACION    OPCIONES     TABLA        Coste Filas      BYTES PADRE Id_Fi
------------ ------------ ------------ ----- ----- ---------- ----- -----
TABLE ACCESS FULL         PELISAHORA      18     1         25     0     1
```

2) Índice:

CREATE INDEX PELISA_DrIndex ON PELISAHORA (ROUND(DRAMA));

3) Plan con índice:

```
INFORME DEL PLAN
OPERACION    OPCIONES     TABLA        Coste Filas      BYTES PADRE Id_Fi
------------ ------------ ------------ ----- ----- ---------- ----- -----
TABLE ACCESS BY INDEX     PELISAHORA       2     1         35     0     1
             ROWID
             BATCHED

INDEX        RANGE SCAN   PELISA_DRIND     1     1                1     2
                          EX
```

4)Plan con " > 10": Accede a varios mientras que el otro accede a uno, por lo que el número de filas es mayor en la segunda consulta

```
INFORME DEL PLAN
OPERACION    OPCIONES     TABLA        Coste Filas      BYTES PADRE Id_Fi
------------ ------------ ------------ ----- ----- ---------- ----- -----
TABLE ACCESS BY INDEX     PELISAHORA       2     7        245     0     1
             ROWID
             BATCHED

INDEX        RANGE SCAN   PELISA_DRIND     1     1                1     2
                          EX
```