

# AMPLIACIÓN de BASES DE DATOS

(Profesor: Héctor Gómez Gauchía)

## Práctica 4 Apdos 3 y 4 - Optimización de Consultas

### Respuestas:

- **Abre una Hoja Excel**, copia el resultado de los resultados del Explain de cada apartado.

- **A la vez, en un archivo word:**

- Incluye cada pregunta y tu respuesta con la explicación de lo pedido.
- Si usas el explain gráfico (F10) puedes guardar una captura en este archivo.

*Te va a servir para estudiar para los exámenes y completar el Cuestionario.*

Siempre que trabajes, haz Lista de Dudas concretas para consultar con el profesor, online o por email o en clase o en el laboratorio.

Modo de entrega: No se Entrega

**(para ganar apoyo en la nota del Examen Final: debes ejecutar la práctica en el Oracle de la Facultad y rellenar el CUESTIONARIO de la práctica cuando esté disponible: se avisará)**

- Si terminas algún apartado y tienes dudas en tu LISTA, avisa al profesor para resolverlas.
- Los conceptos de esta práctica se evalúan en el examen Final.

## APARTADO 3

### Preparación:

- Vamos a usar más volumen de datos para que los valores estadísticos tengan más sentido:  
-> Si el tablespace se te llena mira la nota al final del apartado .

Paso 1 - Importa las tablas PELISAHORA y PELISHIST en formato excel de la carpeta /BDpelis/ :

PASOS: en SQLDeveloper, sobre carpeta *Tablas*: con B.dcho + importar Datos

- 1. *Vista previa*: Deja la cabecera y resto igual. Escoger el fichero.

(para cada paso: avanzar con <siguiente>)

- 2. *Método de importación*: todo como está, salvo el nombre de la tabla, usa el mismo del fichero

- 3. *Selección de columnas*: dejarlas todas tal cual

- Conserva el mismo nombre del atributo/ columna.

- 4. *Definición de columnas*: Asignar tipo y tamaño a cada columna:

- Quitar el tick de “¿NULL?” A todos los atributos ←←

- En los tipos, los valores por defecto NO son iguales en algunos atributos de las dos tablas.

El tipo que escojas debe ser igual en las dos tablas: id (integer), título (varchar2 256), género (varchar2 128), descripción (varchar2 4000) y el resto de % de cada género (number tamaño 5 y escala 2)

NOTA: los géneros dan un warning, no pasa nada.

- 5. Pulsa “Terminar”

Paso 2 - Crea una Clave primaria en cada tabla para el atributo ID con ALTER TABLE (asígnale un nombre, como PK\_pelisactual y PK\_pelishist), algo así para cada tabla.

```
ALTER TABLE pelishist (Y LA pelisahora también )  
ADD CONSTRAINT pk_pelishist PRIMARY KEY (id);
```

Paso 3 – Crear índice único B+ en cada tabla, con CREATE INDEX sobre el Título para agilizar búsquedas y uniones.

Paso 4 → Ejecutar los tres pasos del apartado Apdo2 (borrar lo que había, explicar consulta y ver el resultado), para hacer una explicación de estas consultas y *completa cada apartado*:

**NOTA:** incluimos la columna de **bytes** para ver el volumen de datos usado en cada operación (y **cost**):

los tres pasos del apartado Apdo2:

```
delete plan_table; /* primero borra las filas de explicación anterior */
```

```
EXPLAIN PLAN  
INTO plan_table
```

```

FOR [escribe aquí la consulta];

select operation,options,object_name,cost,cardinality, bytes, parent_id,id
  from plan_table connect by prior id=parent_id
 start with id = 1 order by id;
-----

-- CONSULTA C1 -

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

-- CONSULTA C2 --
select PELISAHORA.DESCRIPCION
from PELISAHORA, PELISHIST
where PELISAHORA.DESCRIPCION = PELISHIST.DESCRIPCION;

-- CONSULTA C3 --
select PELISAHORA.TITULO
from PELISAHORA, PELISHIST
where PELISAHORA.TITULO = PELISHIST.TITULO;

-- CONSULTA C4 --
select PELISAHORA.TITULO
from PELISAHORA
where PELISAHORA.TITULO in (select PELISHIST.TITULO from PELISHIST);

-- CONSULTA C5 --
select PELISAHORA.TITULO
from PELISAHORA
where PELISAHORA.TITULO in
      (select PELISHIST.TITULO from PELISHIST
       where PELISAHORA.TITULO = PELISHIST.TITULO) ;

```

a)- Para comparar la eficiencia en coste y filas manejadas crea esta tabla en el Word con los datos de los planes para las consultas indicadas, una fila por consulta.

CONSULTA	Coste (total)	Num. Filas (total)	Num. Operaciones	Bytes (total)	Es Fiable la estimación?
C1 ....					
C2 ....					
....					

**NOTA:** El *coste total* de un plan es el coste de la operación raíz, porque indica el coste acumulado de sus hijos (cada hijo tiene su coste particular). No pasa igual con el total de las filas (**Cardinality**) y bytes utilizados, que es la suma de las cantidades individuales de todas las operaciones.

**NOTA:** ¿Cómo saber si tenemos una buena estimación de los valores del plan?: Una medida simple es comprobar que el Num. Filas (Cardinality) es igual o muy parecido a las filas reales que tenemos en las tablas. Si no es así, es que la estimación está fallando y no es fiable.

**NOTA:** si alguna operación no está en la teoría, puedes buscar en internet, ej.: Oracle index fast full.

### Contesta a las siguientes preguntas consultando la teoría:

- a.1) Sobre C1: ¿Porqué no accede a la tabla PELISHIST?
- a.2) Sobre C1: ¿Cuál es el criterio principal aplicado para escoger un plan de ejecución u otro?
- a.3) Compara C1 y C2: ¿Qué problema hay con los índices para que el coste sea tal alto en C2?
- a.4) Compara C2 y C3: ¿Porqué, en C2, accede a las tablas completas?
- a.5) Compara C2 y C3: ¿Porqué, en C2, el coste es tal alto?

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

**a.7)** Compara C3 y C4: ¿Porqué ambas NO hacen las mismas operaciones de bajo nivel?

**a.8)** Compara C4 y C5: ¿Porqué ambas SI hacen las mismas operaciones de bajo nivel?

**b)- Explica** cuál de las consultas es más eficiente si tienes en cuenta:

**b.1)** Sólo el *coste* y las *filas usadas*.

**b.2)** Sólo el *coste* y los *bytes usados*.

**c)- Crear diferentes tipos de índices dependiendo de las situaciones:**

**c.1** Queremos acceder rápidamente por el género. Asume que tiene pocos valores distintos. - ¿Qué índice conviene crear? - Créalo. - Obtener el plan sobre esta consulta a ver si usa el índice: Queremos la pelis de la tabla pelisahora que sean del género Drama y tengan más de un 35% de drama.

**c.2** Vamos a hacer muchas consultas como la del ejemplo, así que queremos agilizarlas mediante un índice. EJ.:

```
select titulo, round(Drama)
from pelisahora
where round(Drama) = 50 ;
```

Para responder sigue estos pasos:

1.- Obtener el plan de ejecución con esa consulta

2.- Crear el índice adecuado

3.- Ahora, obtener el plan de ejecución otra vez y comprobar si ahora usa dicho índice.

4.- Genera el plan de la consulta con la condición "> 10". ¿Qué diferencia hay en el plan? ¿porqué ?

**d)- (OPCIONAL)** Crea consultas diferentes sobre las tablas de películas, que provoquen en Oracle las operaciones siguientes: (consulta en teoría qué características las provocan)

**d.1)** Full table scan, **d.2)** Index unique scan, **d.3)** Full Index scan y **d.4)** Cartesian join.

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

**d.5)** - NO use los índices

**d.6)** - SÍ use índices

*Prueba a forzar el uso de un tipo de operación concreta como este ejemplo:*

```
d.7) Select /*+ Full(Empleado) */ NumEmp, NomEmp from Empleado;
```

**e)- (OPCIONAL)** Prueba otro modo de conseguir los datos del plan usando el paquete XPLAN (basado en estimaciones, como el explain plan) :

```
SELECT PLAN_TABLE_OUTPUT FROM TABLE(DBMS_XPLAN.DISPLAY()) ;
```

NOTA: Para ampliar tu tablespace: (desde el usuario ADMINUSER)

```
alter DATABASE DATAFILE 'ESPACIOU-tuyo' autoextend ON
next 521k maxsize 25M;
```

#### APARTADO 4.- Consultas jerárquicas

*(Entregar las instrucciones necesarias en sql y los resultado de las consulta)*

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’

**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');
```

**c)-** Hacer una *consulta jerárquica* conectada por PalID y PadreID que empiece con la palabra ‘select’  
(es como la que usamos para ver la operaciones de la PLAN\_TABLE)

**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”).

**e)** Ejecuta de nuevo la *consulta jerárquica* y comprueba que sale correctamente.