

AMPLIACIÓN de BASES DE DATOS

(Profesor: Héctor Gómez Gauchía)

Práctica 4 SEMANA 8 y 9 - Optimización de Consultas

Resultado:

- En un archivo reúne Instrucciones, Resultados de las ejecuciones y tus respuestas a las preguntas.
- Haz lista de dudas concretas sin resolver sobre tus respuestas para consultar con el profesor

Modo de entrega:

- Si terminas alguno de los apartados indicados, avisa al profesor para puntuar la parte de participación en clase y sigue sus instrucciones.
- Los conceptos de esta práctica se evalúan en el examen de control

APARTADO 1 Preparación para esta práctica

- Ejecuta crear-BDejemplo.sql que viene en esta práctica

APARTADO 2 → Cómo obtener la información de Ejecución de Consultas y comparar la eficiencia

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.

a.- Para cada consulta:

- Ejecuta cada instrucción por separado y copia resultado para los siguientes apartados

```
-- 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 “el contenido” (no el fichero) del MASCPPLAN13.sql para ver la salida formateada
→ Copia el resultado para comparar con los otros. Para cuadrar columnas usa fuente courier new

-- 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 MASCPPLAN13.sql
→ Copia el resultado para comparar con los otros

-- 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 MASCPPLAN13.sql
→ Copia el resultado para comparar con los otros
```

```

-- CONSULTA 4 - ----
delete plan_table;

EXPLAIN PLAN
    INTO plan_table
    FOR (select * from cliente where dni in
        (select dni from invierte));
→ Ejecutar MASCPLAN13.sql
→ Copia el resultado para comparar con los otros

-- 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
→ Copia el resultado para comparar con los otros

-- 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
→ Copia el resultado para comparar con los otros

```

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

b.1 Comparando CONSULTA 2 y CONSULTA 1:

¿Qué reglas aplica en ambas consultas para que no coincidan las operaciones que realizan?
(Repasa la teoría : Reglas de Optimización en Selección)

b.2 Porqué en CONSULTA 1 no accede por el índice a MOROSO ?

(Repasa la teoría : Reglas de Optimización, Combinación dos tablas)

b.3 Porqué en CONSULTA 2 sí accede por el índice a CLIENTE ?

b.4 Porqué en CONSULTA 3 no accede por el índice a CLIENTE ?

c.- - Solo para la consulta 6: (las demás las puedes hacer en casa)

- 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*.

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

e.- Genera otro árbol con F6 y compáralo con el obtenido usando F10. Son ambas funciones iguales?

f.- Exporta a un fichero html el árbol del plan (botón dcho sobre raíz de salida de F10).

Práctica 4 SEMANA 9

APARTADO 3

Preparación:

- Si el tablespace se te llena mira la nota al final del apartado .
 - Importa las tablas en formato excel de la /BDpelis/: Sobre Tablas: con B.dcho + importar Datos
 - Deja la cabecera
 - Conserva el mismo nombre del atributo/ columna
 - El tipo que escojas debe ser razonable y en las dos tablas igual: id (integer). Título, género y descripción (VARCHAR2) y el resto (NUMBER: tamaño 5 con 2 de escala y no Null)
 - Conviene pulsar botón verificar antes de ejecutar la importación.
 - Crea una Clave primaria en cada tabla para el atributo ID con ALTER TABLE (asígnale un nombre, como PK_pelisactual)
 - Crear índice único B+ con CREATE INDEX sobre el Título para agilizar las búsquedas y las uniones.
- Ejecutar los tres pasos del apartado anterior (borrar lo que había, explicar consulta y ver el resultado), para hacer una explicación de estas consultas y *completa cada apartado*:

```
-- CONSULTA 1 --

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

-- CONSULTA 2 --
select PELISAHORA.DESCRIPCION
from PELISAHORA, PELISHIST
where PELISAHORA.DESCRIPCION = PELISHIST.DESCRIPCION;

-- CONSULTA 3 --
select PELISAHORA.TITULO
from PELISAHORA, PELISHIST
where PELISAHORA.TITULO = PELISHIST.TITULO;

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

-- CONSULTA 5 --
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 un documento Word* con los datos de los planes para las consultas indicadas, una fila por consulta.

NOTA: El *coste total* de un plan es el coste de la operación raíz, porque indica el coste acumulado de sus hijos. Igual pasa con el total de las filas utilizadas.

- Indica en “**Qué observas?**” tus conclusiones sobre cada operación, su Coste y sus Filas utilizadas

CONSULTA	Coste (total)	Num. Filas (total)	Num. Operaciones	Qué observas?
1				
2				
. . . .				

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

- c)- **c.1** Queremos acceder rápidamente por el género. Asume que tiene pocos valores distintos. - ¿Qué índice conviene crear? - Créalo. - Aplica el plan sobre esta consulta a ver si lo usa: 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 con el valor redondeando del atributo drama. Qué índice debes crear para agilizar esta consulta:

```
select titulo, ROUND(Drama)
from pelisahora
where ROUND(Drama) = 10 ;
```

Sigue estos pasos:

- 1.- Comprueba el plan con esa consulta
- 2.- Crea el índice adecuado
- 3.- Comprueba ahora que el plan con esa consulta usa dicho índice.
- 4.- Genera el plan de la consulta con la condición "> 10". ¿Qué diferencia hay en el plan? ¿porqué ?

- d)-** Crea consultas nuevas sobre las tablas de películas (no vistas antes),, que provoquen en Oracle las operaciones siguientes: (consulta en teoría qué características las provocan)
Full table scan, Index unique scan, Full Index scan y Cartesian join.

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

- NO use los índices
- SÍ use índices

- e)-** Prueba otro modo de conseguir los datos de el plan usando el paquete XPLAN:

```
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 de 'masplan.sql')

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)