**AMPLIACIÓN de BASES DE DATOS**
(Profesor : Héctor Gómez Gauchía)

# Práctica 3: Optimizacion de Consultas

*Fecha entrega:* **24-04-2014 después de la clase de Laboratorio**
*Resultado*: Los ficheros indicados en cada apartado, todos en un directorio comprimido.
*Modo de entrega:* Subir un zip con el directorio al Campus Virtual.

**APARTADO 1.A**

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

→ **Entrega** un fichero *prac31.docx* con lo siguiente:
- Una tabla con los datos de los planes para las consultas siguientes, una fila por consulta.
  - Las columnas son: Coste (total), Num. Filas(total), Num. Operaciones.
  - Qué observas en la relación de cada operación, su Coste y sus Filas utilizadas?
- Para cada consulta:
  - El resultado con las operaciones del explain plan
  - Dibuja el árbol del Plan de Ejecución: incluye el núm. De operación y su nombre en cada nodo
  - Explica brevemente porqué ha ejecutado esas operaciones y no otras.
- Responde a las *preguntas que hay después de las ejecuciones*.

*Para ejecutarlas sigue estos pasos*: (ver NOTA al final del enunciado sobre ejecutar en sql Developer)

**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`

**2)-** Para tener una mascara de edición que formatee el resultado del plan , **ejecuta**: (sale vacía ahora)
`MASCPLAN13.sql`

**3)- Ejecuta** las siguientes explicaciones siguiendo estos pasos**:**
(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) union (select * from moroso);
```
→ Ejecutar `MASCPLAN13.sql`

```
-- 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`

```
-- 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`

```
-- CONSULTA 4 – que pasa en esta consulta? ----
delete plan_table;

EXPLAIN PLAN
SET STATEMENT_ID= 'mipruxx'
INTO plan_table
     FOR (select * from cliente) union (select * from invierte);
```
→ Ejecutar `MASCPLAN13.sql`

```
-- CONSULTA 5 –   compara con la 3  ----
delete plan_table;

EXPLAIN PLAN
SET STATEMENT_ID= 'mipru05'
INTO plan_table
     FOR (select * from cliente where dni in
               (select dni from  invierte));
```
→ Ejecutar `MASCPLAN13.sql`

---

→ *Contestar a las siguientes preguntas sobre los resultados anteriores*:
Porqué en -- CONSULTA 2 --  el acceso será más eficiente que en -- CONSULTA 1-- ?
Porqué en  -- CONSULTA 2 –  no accede por el indice a moroso ?
Porqué en  -- CONSULTA 3 –  sí accede por el indice a CLIENTE ?
Porqué en  -- CONSULTA 3 –  no accede por el indice a MOROSO ?

*Entregar:ver al principio del apartado 1.*

**APARTADO 1.B-** (incluye lo solicitado en el fichero de entrega prac31.docx)

→ *Ejecutar  las tres instrucciones de la explicación de la consulta: borrar lo que había, explicar consulta y ver el resultado,  para*:

**a)**- Hazlo de las dos formas, sin 'distinct' y con 'distinct':
- Incluye en la entrega los resultados del explain de ambas
- Qué diferencias hay?

```
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;
```

**b)**- Escribe la explicación de otra consulta *equivalente*  a la de **a)** usando una subconsulta anidada.

→ *Compara los dos resultados haciendo lo siguiente*:

**c)**- Dibuja los dos árboles del plan de ejecución, para **a)** *con distinct* y para **b)**

**d)**- Explica cual sería más eficiente sin tener en cuenta el coste, basándote en esta información:
- *Criterios*: Cuántas operaciones y de qué tipo son
- *DATOS a tener en cuenta*: Hay muchas menos filas en Cliente que en Inversiones
  Nested loop es más costoso que Filter.
  Tipos de acceso:  mas costoso secuencial (full)  que
  con indices (dos tipos: más costoso *unique* que *range*).

**e)**- Explicar cual es más eficiente si ahora tienes en cuenta el coste y la filas usadas (cardinalidad)?

**f)** - Que instrucción ha ejecutado oracle para hacer el ' *distinct'?*

**g) - Ejecutar en *sql Developer*** con iconos "Rastreo Automático (F6)" y con "Explicación del plan (F10)" la misma consultas y compara los resultados (marcar la consulta para ejecutar ambas operaciones). Se puede obtener un fichero html con el árbol del plan (botón dcho sobre la raíz de la salida de F6 o F10).
- Qué diferencias hay? Son ambas funciones iguales?

**h) - Escribe consultas nuevas (no vistas antes), con la BDEjemplo, que provoquen en Oracle las operaciones siguientes**:

Full table scan, Table access by ROWID, Index unique scan, Full Index scan,
Nested Loops, Sort, y Cartesian join.

**i)- Escribe consultas nuevas (no vistas antes), con la BDEjemplo y tablas PelisAhora, Pelishist, que provoquen en Oracle (ver Reglas de Optimización) que:**

- NO use los índices
- SÍ use índices

*Entregar: incluye en el fichero prac31.docx lo pedido en cada sección*

## APARTADO 2.-

**a)-** Se desea 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)-** Se desean 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 'mascplan.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.

*Entregar: Fichero prac32.sql con las instrucciones necesarias en sql.*
*Si has tenido que hacer algo diferente a lo pedido, explícalo brevemente.*

## APARTADO 3

Vamos a probar con tablas con más filas: Hector.pelisahora y Hector.pelishist

1.- Hacer los planes de ejecución, dibujar los árboles, comenta la eficiencia de las operaciones y justificar porque se usan esas operaciones para las consultas siguientes:

```
a) (select * from hector.pelisahora where ID in
            (select ID from  hector.pelishist where ID like '%1%');
```

```
b) select * from hector.pelisahora where ID in
            (select descripcion from  hector.pelishist where ID like '%1%');

c) select * from hector.pelisahora where ID in
            (select descripcion from  hector.pelishist where
                                                ID like '%1%' or ID ='3');
```

*Entregar: Fichero prac33.docx con la explicación*

**<u>NOTA sobre ejecución en SQL Developer</u>: Hay dos modos de ejecutar** *MASCPLAN13.sql*:

- **consulta(icono: triángulo) resultado sale en forma de tabla**
- **script (icon: triángulo dentro de rectángulo con rayas) sale en forma de texto. Así se puede cortar y pegar o salvar como fichero de texto.**

**APARTADO 4**

Para determinar **problemas de rendimiento en consultas**, además de ver el Plan de Ejecución de una consulta, es útil analizar estadísticas estimadas y reales de ejecución de consultas, así como de índices y tablas.

**Sección 1.- Paquete DBM_XPLAN: Estadísticas de ejecución real y estimaciones de una Consulta**

Este paquete sirve cuando se necesitan estadísticas con más detalles que con el EXPLAIN, así como para obtener datos reales de ejecución (hasta ahora eran estimados). Para ello completa los siguientes apartados:

***a)*** *Estadísticas con datos reales de* ***ejecución***

Teniendo en cuenta el significado de los atributos:

- E-Rows : Filas estimadas
- A-Rows : Filas reales (cuando ejecuta la consulta)
- Buffers: memoria que ha usado para las operaciones

SE PIDE:

- Ejecuta las siguientes instrucciones y entrega los resultados
- Haz una tabla comparando los totales de esos atributos para las dos consultas.
- Si tuvieras que escoger entre las dos consultas, cual escogerías? Justifícalo con los datos de la tabla

---

```
select /*+ GATHER_PLAN_STATISTICS */ NombreC
 from cliente where DNI in
          (select DNI from moroso where NombreC = 'Client D');

SELECT plan_table_output
 FROM table(DBMS_XPLAN.DISPLAY_CURSOR (FORMAT=> 'ALLSTATS LAST'));
```

---

```
select /*+ GATHER_PLAN_STATISTICS */ cl.NombreC
 from cliente cl, moroso mo
 where cl.DNI = mo.DNI and
      mo.DNI < '00000006';

SELECT plan_table_output
 FROM table(DBMS_XPLAN.DISPLAY_CURSOR (FORMAT=> 'ALLSTATS LAST'));
```

NOTA: La instrucción /*+ GATHER_PLAN_STATISTICS */ es una "hint", para que Oracle almacene estadísticas para esa consulta *mientras la ejecuta*

***b)*** *Para una consulta más compleja*

Esta consulta sirve para ver qué tablas están bloqueadas, se verá mejor en PRAC4

La estimación que da el Plan de ejecución, no siempre es buena. Observa la diferencia en las filas estimadas y las de ejecución.

SE PIDE:

- Ejecuta las siguientes instrucciones y entrega los resultados
- Observa las diferencias entre E-Rows (estimación) y A-Rows (ejecución). Indica en qué operaciones (de las que tienen bastantes filas) ambos atributos son iguales o casi. Por qué es?
- Qué operaciones consumen más memoria (atributo used-Mem). Porqué?

```
SELECT /*+ GATHER_PLAN_STATISTICS */
      DO.owner,
      DO.object_name,
```

```
      DO.object_type,
      lo.session_id,
      lo.oracle_username
  FROM dba_objects DO, v$locked_object lo
WHERE DO.object_id = lo.object_id;

SELECT plan_table_output
 FROM table(DBMS_XPLAN.DISPLAY_CURSOR (FORMAT=>'ALLSTATS LAST'));

select *
from all_tables
where table_name ='PELISHIST' and owner='HECTOR';
```

**Sección 2.- Características de tus tablas e índices consultando USER_INDEXES y USER_TABLES**

***1.- Tabla user_indexes : (si son índices de otro usuario trabaja con: all_indexes )***

SE PIDE:

- Ejecuta cada una de las instrucciones, copia el resultado de ejecutar la instrucción
- Describe qué obtienes con cada una, describe sus atributos y valores obtenidos Necesitarás consultar la web.
- Para los atributos DISTINCT_KEYS, AVG_LEAF_BLOCKS_PER_KEY puedes recordar la explicación que vimos en clase de teoría.
- Porqué los dos índices tienen valores diferentes en DISTINCT_KEYS?

**1.a)** desc user_indexes

**1.b)** select OWNER,INDEX_NAME,INDEX_TYPE, TABLE_NAME, UNIQUENESS
from all_indexes
Where TABLE_NAME = 'PELISHIST';

**1.c)** select OWNER, INDEX_NAME, INITIAL_EXTENT, BLEVEL, LEAF_BLOCKS, NUM_ROWS
from all_indexes
Where TABLE_NAME = 'PELISHIST';

**1.d)** select OWNER, INDEX_NAME, DISTINCT_KEYS, AVG_LEAF_BLOCKS_PER_KEY,
AVG_DATA_BLOCKS_PER_KEY, CLUSTERING_FACTOR
from all_indexes
Where TABLE_NAME = 'PELISHIST';

***2.- Tabla user_tables : (si son tablas de otro usuario trabaja con: all_tables)***

SE PIDE:

- Ejecuta cada una de las instrucciones, copia el resultado de ejecutar la instrucción
- Describe qué obtienes con cada una, describe sus atributos (los más importantes en **2.c**) y valores obtenidos. Necesitarás consultar documentación en la web.

**2.a)** desc user_tables

**2.b)** select table_name, num_rows, blocks, empty_blocks, avg_row_len
from user_tables;

**2.c)** select *
from **all_tables**
where table_name ='PELISHIST';

**Sección 3 .- Paquete DBMS_STATS: Estadísticas del Optimizador de Consultas**

Gestión de estadísticas de tablas e índices que usa el Optimizador para calcular costes en la decisión de escoger su Plan de Ejecución. El optimizador usa las estadísticas que tengas para decidir el plan de ejecución de las operaciones. Si no se cambian nunca, puede que estén obsoletas. Conviene obtenerlas de nuevo cada cierto tiempo.

SE PIDE:

- Ejecuta cada una de las instrucciones, copia el resultado de ejecutar la instrucción
- Describe qué obtienes con cada una, describe sus atributos y valores obtenidos
  Necesitarás consultar la web.

**a)** EXEC DBMS_STATS.GATHER_INDEX_STATS('HECTOR','INDEXDESCRIPPELISHIST');

**b)** EXEC DBMS_STATS.GATHER_TABLE_STATS('HECTOR','PELISHIST');

**c)**
```
select OWNER,INDEX_NAME,NUM_ROWS,LAST_ANALYZED,
       BLEVEL,LEAF_BLOCKS,DISTINCT_KEYS
from dba_indexes
where owner = 'HECTOR'
  and index_name ='INDEXDESCRIPPELISHIST';
```

**Sección 4 : Estadísticas de una Tabla y de Columna desde el SQL DEVELOPER**

SE PIDE:

- Explora estas instrucciones, ejecútalas e indica qué obtienes
  Explica los atributos que reconoces, que creas más importantes

**a)** Encima de una tabla botón derecho: menu contexto "Estadísticas" + “Valida Estructura”

**b)** Encima de una tabla + B.dcho: menu contexto "Estadísticas" + Recopilar Estadística
(equivale al ANALIZE)

**c)** Ahora se puede consultar, un vez abierta esa tabla, en la ventana derecha con pestañas:
en la pestaña "estadísticas"

**d)** Desde la misma ventana con pestañas de una tabla: pestaña "Estadística"
- Abajo en ventana "Estadísticas de Columna"

**e)** En "Refrescar = 5" refresca datos cada 5 segundos

**f)** Desde la misma ventana con pestañas de una tabla: pestaña "Detalles"

*Entregar: Fichero prac34.docx con todo lo pedido*