

Prácticas de

Administración de Bases de Datos

Grado en Ingeniería Informática

**PRÁCTICA 5**

Procesamiento y optimización de consultas

**Ejercicios**

1. Tras una actualización de la base de datos (desde la versión 10.2.0.4), se ha informado de que algunas consultas no parecen ser tan eficientes como anteriormente. Se le pide al administrador que mientras no se encuentren posibles optimizaciones, cambie algún parámetro del optimizador de consultas para que funcione igual que en la versión anterior. ¿Cómo podría realizarse?.

**ALTER SYSTEM SET optimizer\_features\_enable = '10.2.0.4';**

1. ¿Qué tendría que hacerse para que el optimizador no usara estadísticas dinámicas en la sesión actual?.

**ALTER SYSTEM SET optimizer\_dynamic\_sampling = 0;**

1. Un proceso realiza consultas a la base de datos y requiere bajos tiempos de respuesta, aunque no necesita todos los datos en un primer momento. ¿Cómo podría cambiarse el comportamiento del optimizador para conseguirlo?.

**ALTER SYSTEM SET optimizer\_mode = FIRST\_ROWS\_1;**

1. Examina el contenido del script *catplan.sql*.
2. Lanza EXPLAIN PLAN (guardándolo con el statement\_id *sentencia1*) para la siguiente consulta:

SELECT phone\_number FROM hr.employees

WHERE phone\_number LIKE '%16%';

**EXPLAIN PLAN**

**SET STATEMENT\_ID = 'sentencia1' FOR**

**SELECT phone\_number FROM hr.employees**

**WHERE phone\_number LIKE '%16%';**

Muestra el plan mediante UTLXPLS.SQL y mediante DBMS\_XPLAN.DISPLAY.

¿Qué diferencias hay entre hacerlo de una u otra manera?.

**Para mostrarlo mediante DBMS\_XPLAN.DISPLAY:**

**SELECT PLAN\_TABLE\_OUTPUT**

**FROM TABLE(DBMS\_XPLAN.DISPLAY(NULL, 'sentencia1', 'BASIC'));**

**Para mostrarlo mediante UTLXPLS.SQL:**

**select plan\_table\_output from table(dbms\_xplan.display('plan\_table', 'sentencia1','serial'));**

**La diferencia es que la función DBMS\_XPLAN.DISPLAY permite mayor configuración, mientras que UTLXPLS.SQL es un script predefinido pero que aporta rapidez para no tener que configurar ningún comando.**

Muestra el plan con el mayor nivel de detalle y también con el menor nivel de detalle.

1. Lanza EXPLAIN PLAN para la siguiente consulta:

SELECT \*

FROM hr.employees ORDER BY last\_name;

¿Cambia algo del plan si modificas para la actual sesión el valor del parámetro

DB\_FILE\_MULTIBLOCK\_READ\_COUNT a 1? ¿Por qué?.

1. Lanza EXPLAIN PLAN para la siguiente consulta:

SELECT \*

FROM sh.customers

WHERE cust\_city='Los Angeles' AND cust\_state\_province='CA';

Examina el plan resultante. ¿Cambia algo del plan si modificas para la actual sesión el valor del parámetro OPTIMIZER\_DYNAMIC\_SAMPLING a 10?.

1. En cierto código se ha encontrado la siguiente consulta:

SELECT /\*+ FULL(e) \*/ employee\_id, last\_name FROM hr.employees e

WHERE last\_name LIKE '%Smith%';

Analiza su plan de ejecución actual, si el plan de ejecución hace uso de algún índice y cómo podría mejorarse.

1. En cierto código se ha encontrado la siguiente consulta:

SELECT /\*+ NO\_USE\_HASH(o l) \*/

o.customer\_id, l.unit\_price \* l.quantity FROM oe.orders o, oe.order\_items l

WHERE l.order\_id = o.order\_id;

Analiza su plan de ejecución y estudia si podría mejorarse.

1. Para la siguiente consulta:

SELECT \*

FROM hr.employees e, hr.departments d, hr.job\_history j WHERE e.department\_id = d.department\_id

AND e.hire\_date = j.start\_date;

* 1. Examina su plan de ejecución.
  2. Examina los posibles planes resultantes de emplear *LEADING*.

1. Para la siguiente consulta:

SELECT \*

FROM hr.employees, hr.departments

WHERE employees.department\_id = departments.department\_id;

Evalúa si es ventajoso emplear *USE\_MERGE* o *NO\_USE\_MERGE*.

1. En cierto código se ha encontrado la siguiente consulta:

SELECT /\*+ NO\_USE\_MERGE(e d) \*/ \*

FROM hr.employees e, hr.departments d WHERE e.department\_id = d.department\_id ORDER BY d.department\_id;

Analiza su plan de ejecución y estudia si podría mejorarse.

1. Dada la siguiente consulta:

SELECT e1.last\_name, e1.salary, v.avg\_salary FROM hr.employees e1,

(SELECT department\_id, avg(salary) avg\_salary FROM hr.employees e2

GROUP BY department\_id) v

WHERE e1.department\_id = v.department\_id AND e1.salary > v.avg\_salary

ORDER BY e1.last\_name

Compara su plan de ejecución con el correspondiente a usar *MERGE* en esa misma consulta.

1. Dada la siguiente consulta:

SELECT h.customer\_id, l.unit\_price \* l.quantity FROM oe.orders h, oe.order\_items l

WHERE l.order\_id = h.order\_id;

Compara su plan de ejecución con el correspondiente a usar *USE\_NL* en esa misma consulta.

1. Dada la siguiente consulta:

SELECT employee\_id, department\_id FROM hr.employees

WHERE department\_id > 50;

Estudia si su plan de ejecución puede mejorarse mediante el uso del índice

*emp\_department\_ix*.

1. En cierto código se ha encontrado la siguiente consulta:

SELECT /\*+ NO\_INDEX(employees emp\_empid) \*/ employee\_id FROM hr.employees

WHERE employee\_id > 200;

Analiza su plan de ejecución e indica si convendría realizar algún cambio.