

Prácticas de CACHE

PL/SQL 12c-18c avanzado

NOTA: Aunque siempre pongo las soluciones, os recomiendo que primero intentéis hacer el ejemplo por vosotros mismos y luego lo comparéis con el mío. ¡¡¡En muchas ocasiones, el mismo proceso se puede hacer de varias formas!!!!

Caché Parte 2

1. Creamos la siguiente consulta:

```
SELECT * FROM LOCATIONS WHERE COUNTRY_ID='JP';
```

2. En primer lugar comprobamos su plan de ejecución



3. Con el HINT RESULT_CACHE, modificamos la consulta anterior:

```
SELECT /*+ RESULT_CACHE*/ * FROM LOCATIONS WHERE COUNTRY_ID='JP';
```

4. Debe haber generado una caché para la consulta



5. Comprobamos con un report de la memoria caché que se ha usado. Hay que cambiarse al usuario SYSTEM si no tenemos permisos. Podemos comprobar que al menos hay un bloque de SQL que está usando la memoria

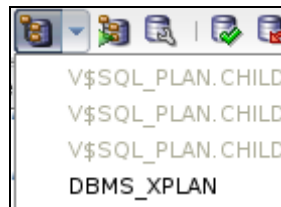
```
EXECUTE DBMS_RESULT_CACHE.MEMORY_REPORT;
```

```
Procedimiento PL/SQL terminado correctamente.
Result Cache Memory Report
[Parameters]
Block Size           = 1K bytes
Maximum Cache Size   = 200M bytes (200K blocks)
Maximum Result Size   = 10M bytes (10K blocks)
[Memory]
Total Memory = 216400 bytes [0.072% of the Shared Pool]

... Fixed Memory = 1040 bytes [0.000% of the Shared Pool]
... Dynamic Memory = 215360 bytes [0.071% of the Shared Pool]
..... Overhead = 182592 bytes
..... Cache Memory = 32K bytes (32 blocks)
..... Unused Memory = 2 blocks
..... Used Memory = 30 blocks
..... Dependencies = 17 blocks (17 count)
..... Results = 13 blocks
..... SQL = 1 blocks (1 count)
..... CDB = 12 blocks (10 count)

Procedimiento PL/SQL terminado correctamente.
```

6. Comprobamos las estadísticas para esa memoria cache. Evidentemente, cada uno tendréis la vuestra. Yo estoy usando la mía para el ejemplo
7. Nos ponemos en el botón “Generar plan” y seleccionamos DBMS_XPLAN.



8. Debe generar algo parecido a lo siguiente:

```
SELECT * FROM LOCATIONS WHERE COUNTRY_ID='JP'

select * from table(dbms_xplan.display_cursor(sql_id=>'96vhjj3j1khrx', format=>'ALLSTATS
LAST'))
```

9. Copiamos el último comando a una sesión del usuario SYSTEM y lo lanzamos
10. Seleccionamos la RESULT_CACHE que indica
- 11.

SQL_ID 96vhjj3j1khrx, child number 0			

SELECT /*+ RESULT_CACHE*/ * FROM LOCATIONS WHERE COUNTRY_ID='JP'			
Plan hash value: 916683258			

Id	Operation	Name	E-Rows

0	SELECT STATEMENT		
1	RESULT CACHE	5aawmswkg96cw1ua9s12d59bd3	
2	TABLE ACCESS BY INDEX ROWID	LOCATIONS	2
* 3	INDEX RANGE SCAN	LOC_COUNTRY_IX	2

12. Comprobamos el estado de esa caché

```
SELECT * FROM V$RESULT_CACHE_OBJECTS WHERE
CACHE_ID='5aawmswkg96cw1ua9s12d59bd3';
```

TYPE	STATUS	BUCKET_NO	HASH	NAME
Result	Published	737	481055457	SELECT /*+ RESULT_CACHE*/ * FROM LOCATIONS WHERE COUNTRY_ID='JP'

13. Podemos ver que tiene un bloque, 2 filas y 6 columnas cacheadas

BLOCK_COUNT	SCN	COLUMN_COUNT	PIN_COUNT	SCAN_COUNT	ROW_COUNT
1	9800218	6	0	1	2

14. Vamos a crear ahora una función que cachea los resultados de una función que luego podemos usar en una SQL normal.

15. Para ello vamos a usar el paquete GEST_EMPLE que hemos usado durante las prácticas de la sección de SQL dinámico.

16. Vamos a usar una función que devuelva el salario neto de un empleado y la vamos a cachear. La llamamos SALARIO_NETO

17. Es bastante simple. Recibe un EMPLOYEE_ID y devuelve el salario neto (por ejemplo, le quitamos el 15% de impuestos,

```
FUNCTION SALARIO_NETO(CODIGO NUMBER) RETURN NUMBER;
```

18. Dentro de la cabecera del paquete debemos indicar que se usará la cache

```
FUNCTION SALARIO_NETO(CODIGO NUMBER) RETURN NUMBER RESULT_CACHE ;
```

19. Un ejemplo sería el siguiente

```
FUNCTION SALARIO_NETO(CODIGO NUMBER) RETURN NUMBER
RESULT_CACHE RELIES_ON(EMPLOYEES)
IS
NETO NUMBER;
```

```
BEGIN
    SELECT SALARY-(SALARY*15/100) INTO NETO FROM EMPLOYEES WHERE
    EMPLOYEE_ID=CODIGO;
    RETURN NETO;
END;
```

20. Ahora la usamos en una consulta SQL normal

```
SELECT FIRST_NAME,SALARY,GEST_EMPL.SALARIO_NETO(EMPLOYEE_ID) FROM
EMPLOYEES;
```

21. Nos cambiamos a SYSTEM y comprobamos que se ha generado cache. Podemos ver también que se ha usado el namespace PLSQLm, no el de SQL

```
SELECT * FROM V$RESULT_CACHE_OBJECTS WHERE NAME LIKE '%SALARIO_NETO%';
```

ID	TYPE	STATUS	BUCKET_NO	HASH	NAME	NAMESPAC
50	Result	Published	3955	4101328755	"HR"."GEST_EMPL"::11."SALARIO_NETO"#3048d2af80817a01 #112	PLSQL
55	Result	Published	3788	3216174796	"HR"."GEST_EMPL"::11."SALARIO_NETO"#3048d2af80817a01 #112	PLSQL
56	Result	Published	3764	1883614900	"HR"."GEST_EMPL"::11."SALARIO_NETO"#3048d2af80817a01 #112	PLSQL
80	Result	Published	3735	1849626263	"HR"."GEST_EMPL"::11."SALARIO_NETO"#3048d2af80817a01 #112	PLSQL
82	Result	Published	3723	1188961931	"HR"."GEST_EMPL"::11."SALARIO_NETO"#3048d2af80817a01 #112	PLSQL
70	Result	Published	3700	3959037556	"HR"."GEST_EMPL"::11."SALARIO_NETO"#3048d2af80817a01 #112	PLSQL
60	Result	Published	3607	960925207	"HR"."GEST_EMPL"::11."SALARIO_NETO"#3048d2af80817a01 #112	PLSQL
86	Result	Published	3528	3456601544	"HR"."GEST_EMPL"::11."SALARIO_NETO"#3048d2af80817a01 #112	PLSQL
68	Result	Published	3502	4261297582	"HR"."GEST_EMPL"::11."SALARIO_NETO"#3048d2af80817a01 #112	PLSQL
37	Result	Published	3496	1855499688	"HR"."GEST_EMPL"::11."SALARIO_NETO"#3048d2af80817a01 #112	PLSQL
53	Result	Published	3268	2948586692	"HR"."GEST_EMPL"::11."SALARIO_NETO"#3048d2af80817a01 #112	PLSQL
88	Result	Published	3147	2149723211	"HR"."GEST_EMPL"::11."SALARIO_NETO"#3048d2af80817a01 #112	PLSQL