



Parte 3

Estudiantes:

Nicole Araya Ballesterero

Ariana Solano Vallejos

Parte 1

Aumento de Data base buffer

Sentencias encontradas de posible ayuda:

```
-----  
  
ALTER SYSTEM SET DB_CACHE_SIZE = 518767104 SCOPE=SPFILE;
```

```
SELECT name, value, ISSES_MODIFIABLE FROM v$parameter  
WHERE name = 'db_cache_size';
```

```
-----  
  
SELECT * FROM v$sgainfo WHERE name = 'Buffer Cache Size';
```

SHOW PARAMETER DB_CACHE_ADVICE

```
select name, value
from v$parameter
where name in
('sga_max_size',
'sga_target',
'shared_pool_size',
'large_pool_size',
'java_pool_size'
)
or name like 'db%cache_size'
```

- Este comando `ALTER SYSTEM` se utiliza para aumentar el tamaño de la memoria caché del búfer. Además, estás utilizando la opción `SCOPE=SPFILE` para que este cambio se refleje en el archivo de parámetros de inicio (`spfile`) y persista después de reiniciar la base de datos.
- `SELECT name, value, ISSES_MODIFIABLE FROM v$parameter WHERE name = 'db_cache_size';` : Aquí estás consultando la vista dinámica `v$parameter` para obtener información sobre el parámetro 'db_cache_size'. Esto te dará el valor actual del tamaño de la memoria caché del búfer y si es modificable en la sesión actual.
- `SELECT * FROM v$sgainfo WHERE name = 'Buffer Cache Size';` : Esta consulta muestra información sobre la memoria caché del búfer en la base de datos. Estás consultando la vista dinámica `v$sgainfo` para encontrar la entrada con el nombre 'Buffer Cache Size', que debería proporcionar el tamaño actual de la memoria caché del búfer.
- `SHOW PARAMETER DB_CACHE_ADVICE` : Este comando muestra la configuración del parámetro 'db_cache_advice', que se utiliza para habilitar o deshabilitar el asesoramiento de tamaño de la memoria caché del búfer. Esta opción te ayuda a tomar decisiones sobre el tamaño óptimo de la memoria caché del búfer en función de la actividad de la base de datos.
- La ultima consulta, se utiliza para obtener una visión general de la configuración de la memoria y la caché en la base de datos Oracle. Los

componentes clave del Sistema Global del Área de Memoria Compartida (SGA) de Oracle, que gestiona el uso de la memoria en la base de datos.

Alterando el cache buffer

Aplicando el `Alter system` :

```
CREATE OR REPLACE PROCEDURE aumentar_db_cache_size IS
    v_current_size NUMBER;
    v_new_size NUMBER;
BEGIN
    -- Obtener el valor actual de DB_CACHE_SIZE
    SELECT TO_NUMBER(value) INTO v_current_size
    FROM v$parameter
    WHERE name = 'db_cache_size';

    -- Calcular el nuevo tamaño aumentado en un 25%
    v_new_size := v_current_size * 1.25;

    -- Ejecutar el comando ALTER SYSTEM para actualizar el tamaño en
    SPFILE
    --EXECUTE IMMEDIATE 'ALTER SYSTEM SET DB_CACHE_SIZE = ' || v_
    new_size || ' SCOPE=SPFILE';

    -- Confirmar el cambio en el valor
    COMMIT;

    -- Mostrar el valor actualizado
    DBMS_OUTPUT.PUT_LINE('DB_CACHE_SIZE actualizado a: ' || v_new_
    size);

EXCEPTION
    WHEN OTHERS THEN
        -- Manejo de errores
        DBMS_OUTPUT.PUT_LINE('Error al actualizar DB_CACHE_SIZE: ' ||
```

```
SQLERRM);  
END aumentar_db_cache_size;  
/  
  
SET SERVEROUTPUT ON;  
BEGIN  
    aumentar_db_cache_size;  
END;  
/  

```

```
Procedure AUMENTAR_DB_CACHE_SIZE compilado
```

```
DB_CACHE_SIZE actualizado a: 398458880
```

```
Procedimiento PL/SQL terminado correctamente.
```

Antes:

```
Total System Global Area 1610610072 bytes  
Fixed Size                 9856408 bytes  
Variable Size              738197504 bytes  
Database Buffers          855638016 bytes  
Redo Buffers               6918144 bytes  
Base de datos montada.  
Base de datos abierta.
```

Despues:

```
Total System Global Area 1610610072 bytes  
Fixed Size                 9856408 bytes  
Variable Size              553648128 bytes  
Database Buffers          1040187392 bytes  
Redo Buffers               6918144 bytes  
Base de datos montada.  
Base de datos abierta.
```

—Si modifico el tamaño del buffer, se visualizan en el spfile, con este comando encuentro la ubicacion:

```
SELECT name, value FROM v$parameter WHERE name = 'spfile';
```

Me da la dirección:

Nombre	Fecha de modificación	Tipo	Tamaño
cm_xe.dat	4/9/2023 22:16	Archivo DAT	4 KB
hc_xe.dat	25/8/2023 21:47	Archivo DAT	2 KB
initXE.ora	25/8/2023 21:53	Archivo ORA	1 KB
PWDXE.ora	25/8/2023 21:48	Archivo ORA	2 KB
SPFILEXE.ORA	4/9/2023 22:16	Archivo ORA	4 KB

```
SPFILEXE.ORA [x]
1 C:\NUL\NUL\SOHNUL\NUL\NUL\NUL\NUL\NUL\NUL
2 xe.__db_cache_size=956301312
3 xe.__inmemory_ext_roarea=0
4 xe.__inmemory_ext_rwarea=0
5 xe.__java_pool_size=0
6 xe.__large_pool_size=33554432
7 xe.__oracle_base='C:\app\Nicole\prod
8 xe.__pga_aggregate_target=536870912
9 xe.__sga_target=1610612736
10 xe.__shared_io_pool_size=83886080
11 xe.__shared_pool_size=520093696
12 xe.__streams_pool_size=0
13 xe.__unified_pga_pool_size=0
14 *.audit_file_dest='C:\app\Nicole\prod
15 *.audit_trail='db'
16 *.compatible='21.0.0'
17 *.control_files='C:\app\Nicole\produ
18 *.db_block_size=8192
19 *.db_cache_size=398458880
20 *.db_name='XE'
21 *.diagnostic_dest='C:\app\Nicole\prod
22 *.dispatchers='(PROTOCOL=TCP) (SERVI
23 *.enable_pluggable_database=true
24 *.local_listener='LISTENER_XE'
```

Referencias usadas:

<http://oracle11gnotes.blogspot.com/2012/07/72-crear-y-administrar-tablespaces.html>

<https://docs.oracle.com/en/database/oracle/oracle-database/21/tgdba/tuning-database-buffer-cache.html#GUID-74EF352E-2C42-4C39-9F82-00BC7DC616AD>

<https://www.dataprix.com/es/blog-it/ilmasacratore/obtener-lista-tablas-mas-movimiento-insertupdate-oracle>

<https://www.dataprix.com/es/forum/oracle-database/recopilacion-scripts-y-consultas-utiles-oracle>

PARTE 2

MOVER A OTRO TABLESPACE

Creando tablas:

```
create table t1(a int, b int, c int, constraint pkt1 primary key (a));  
create table t4(x int, y int, z int, constraint pkt4 primary key (x));  
create table t5(m int, n int, o int, constraint pkt5 primary key (m));
```

```
create table t3(g int, h int,  
constraint fkt3_1 foreign key (g) references t4(x),  
constraint fkt3_2 foreign key (h) references t5(m),  
constraint pkt3 primary key (g, h));
```

```
create table t2(d int, e int, f int,  
constraint fkt2_1 foreign key (d) references t1(a),
```

```
constraint fkt2_2 foreign key (e, f) references t3(g, h),  
constraint pkt2 primary key (d, e, f));
```

Table T1 creado.

Table T4 creado.

Table T5 creado.

Table T3 creado.

Table T2 creado.

Con esta instrucción vemos donde estan las tablas almacenadas:

```
SELECT table_name, tablespace_name  
FROM user_tables  
WHERE table_name IN ('T1', 'T4', 'T5', 'T3', 'T2');
```

	TABLE_NAME	TABLESPACE_NAME
1	T1	USERS
2	T2	USERS
3	T3	USERS
4	T4	USERS
5	T5	USERS

Creo los tablespaces

```
create tablespace T1 Datafile 'T1.BDF' Size 4M;  
create tablespace T2 Datafile 'T2.BDF' Size 4M;
```

```
create tablespace T3 Datafile 'T3.BDF' Size 4M;  
create tablespace T4 Datafile 'T4.BDF' Size 4M;  
create tablespace T5 Datafile 'T5.BDF' Size 4M;
```

Y muevo las tablas a esos tablespaces:

```
ALTER TABLE t1 MOVE TABLESPACE T1;  
ALTER TABLE t2 MOVE TABLESPACE T2;  
ALTER TABLE t3 MOVE TABLESPACE T3;  
ALTER TABLE t4 MOVE TABLESPACE T4;  
ALTER TABLE t5 MOVE TABLESPACE T5;
```

	TABLE_NAME	TABLESPACE_NAME
1	T1	T1
2	T2	T2
3	T3	T3
4	T4	T4
5	T5	T5

Con esta instruccion puedo ver las llaves PK y FK de una tabla:

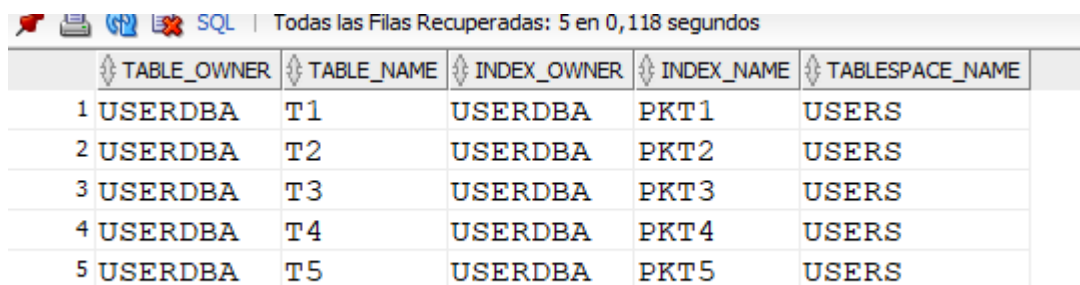
```
SELECT constraint_name, constraint_type  
FROM user_constraints  
WHERE table_name = 'T3';
```

La 'R' son FK y las 'P' son PK:

	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	FKT3_1	R
2	FKT3_2	R
3	PKT3	P

Y en sys puedes ver la informacion completa de un Tablespace:

```
SELECT i.TABLE_OWNER, t.TABLE_NAME, i.OWNER as INDEX_OWNER, i.IN  
DEX_NAME, i.TABLESPACE_NAME  
FROM dba_tables t, dba_indexes i  
WHERE t.OWNER = i.TABLE_OWNER  
AND t.TABLE_NAME = i.TABLE_NAME  
AND t.TABLESPACE_NAME IN ('T1', 'T2', 'T3', 'T4', 'T5')  
ORDER BY i.TABLE_OWNER, t.TABLE_NAME, i.OWNER, i.INDEX_NAME;
```



	TABLE_OWNER	TABLE_NAME	INDEX_OWNER	INDEX_NAME	TABLESPACE_NAME
1	USERDBA	T1	USERDBA	PKT1	USERS
2	USERDBA	T2	USERDBA	PKT2	USERS
3	USERDBA	T3	USERDBA	PKT3	USERS
4	USERDBA	T4	USERDBA	PKT4	USERS
5	USERDBA	T5	USERDBA	PKT5	USERS

Mover indices a los tablespaces que queremos:

```
alter index USERDBA.PKT1 rebuild tablespace T1;  
alter index USERDBA.PKT2 rebuild tablespace T2;  
alter index USERDBA.PKT3 rebuild tablespace T3;  
alter index USERDBA.PKT4 rebuild tablespace T4;  
alter index USERDBA.PKT5 rebuild tablespace T5;
```

Index USERDBA.PKT1 alterado.

Index USERDBA.PKT2 alterado.

Index USERDBA.PKT3 alterado.

Index USERDBA.PKT4 alterado.

Index USERDBA.PKT5 alterado.

	TABLE_OWNER	TABLE_NAME	INDEX_OWNER	INDEX_NAME	TABLESPACE_NAME
1	USERDBA	T1	USERDBA	PKT1	T1
2	USERDBA	T2	USERDBA	PKT2	T2
3	USERDBA	T3	USERDBA	PKT3	T3
4	USERDBA	T4	USERDBA	PKT4	T4
5	USERDBA	T5	USERDBA	PKT5	T5

Aparte ahora si hago inserts en las tablas me va a salir el tamaño que ocupa ese tablespace:

```
insert into t1 values (1,2,3);
insert into t4 values (9,2,3);
commit;
```

```
SELECT OWNER, TABLESPACE_NAME, SUM(BYTES)
FROM DBA_SEGMENTS
WHERE TABLESPACE_NAME IN ('T1', 'T2', 'T3', 'T4', 'T5')
GROUP BY OWNER, TABLESPACE_NAME;
```

	OWNER	TABLESPACE_NAME	SUM(BYTES)
1	USERDBA	T1	196608
2	USERDBA	T4	131072

CALCULAR TABLESPACE

-En SYS

Consultas:

```

---Informacion general del tablespace
SELECT
    t.tablespace_name AS "Tablespace",
    t.status AS "Estado",
    ROUND(MAX(d.bytes) / 1024 / 1024, 2) AS "MB Tamaño",
    ROUND((MAX(d.bytes) / 1024 / 1024) - (SUM(DECODE(f.bytes, NULL,
0, f.bytes)) / 1024 / 1024), 2) AS "MB Usados",
    ROUND(SUM(DECODE(f.bytes, NULL, 0, f.bytes)) / 1024 / 1024, 2) AS
"MB Libres"
FROM
    DBA_FREE_SPACE f,
    DBA_DATA_FILES d,
    DBA_TABLESPACES t
WHERE
    t.tablespace_name = d.tablespace_name
    AND f.tablespace_name(+) = d.tablespace_name
    AND f.file_id(+) = d.file_id
    AND t.tablespace_name IN (
        SELECT DISTINCT tablespace_name
        FROM dba_indexes
        WHERE owner = 'USERDBA'
    )
GROUP BY
    t.tablespace_name, t.status

```

```
ORDER BY
1, 3 DESC;
```

----Informacion de tablespace enfocada en todos sus segmentos (PK, FK , tablas)

```
SELECT TABLESPACE_NAME,
      SUM(CASE WHEN SEGMENT_TYPE IN ('INDEX', 'TABLE', 'FOREIGN KE
Y') THEN BYTES ELSE 0 END) AS TOTAL_SIZE_BYTES
FROM DBA_SEGMENTS
WHERE OWNER = 'USERDBA'
GROUP BY TABLESPACE_NAME
ORDER BY TABLESPACE_NAME;
```

Informacion general del tablespace:

	Tablespace	Estado	MB Tamaño	MB Usados	MB Libres
1	ADMIBD	ONLINE	8	1,13	6,88
2	T1	ONLINE	4	1,13	2,88
3	T2	ONLINE	4	1	3
4	T3	ONLINE	4	1	3
5	T4	ONLINE	4	1,13	2,88
6	T5	ONLINE	4	1	3
7	USERBD	ONLINE	8	1,13	6,88
8	USERS	ONLINE	5	2,88	2,13

Informacion de tablespace enfocada en todos sus segmentos (PK, FK , tablas):

	TABLESPACE_NAME	TOTAL_SIZE_BYTES
1	ADMIBD	262144
2	PRUEBA1	65536
3	T1	262144
4	T4	131072
5	USERBD	131072
6	USERS	851968

Procedures:

-----PROCEDURE INFO GENERAL-----

```

CREATE OR REPLACE PROCEDURE obtener_info_tablespaces(p_cursor OUT SYS_REFCURSOR) AS
BEGIN
    OPEN p_cursor FOR
        SELECT
            t.tablespace_name AS "Tablespace",
            t.status AS "Estado",
            ROUND(MAX(d.bytes) / 1024 / 1024, 2) AS "MB Tamaño",
            ROUND((MAX(d.bytes) / 1024 / 1024) - (SUM(DECODE(f.bytes, NULL, 0, f.bytes)) / 1024 / 1024), 2) AS "MB Usados",
            ROUND(SUM(DECODE(f.bytes, NULL, 0, f.bytes)) / 1024 / 1024, 2) AS "M
B Libres"
        FROM
            DBA_FREE_SPACE f,
            DBA_DATA_FILES d,
            DBA_TABLESPACES t
        WHERE
            t.tablespace_name = d.tablespace_name
            AND f.tablespace_name(+) = d.tablespace_name
            AND f.file_id(+) = d.file_id
            AND t.tablespace_name IN (
                SELECT DISTINCT tablespace_name
                FROM dba_indexes

```

```

        WHERE owner = 'USERDBA'
    )
    GROUP BY
        t.tablespace_name, t.status
    ORDER BY
        1, 3 DESC;

END obtener_info_tablespaces;
/

SET SERVEROUTPUT ON
VAR rc REFCURSOR
EXEC obtener_info_tablespaces(:rc);
PRINT rc

-----PROCEDURE todos sus segmentos -----
----

--EN USERDBA

CREATE TABLE segmentos (
    codigo int,
    fecha_hora VARCHAR2(50),
    tablespace_name VARCHAR2(50),
    total_size_bytes NUMBER,
    constraint pkseg primary key (codigo)
)tablespace USERBD;

CREATE SEQUENCE segmentos_seq MINVALUE 1 START WITH 1 INCREME
NT BY 1 CACHE 200;

--EN SYS
CREATE OR REPLACE PROCEDURE insertar_segmentos AS
c SYS_REFCURSOR;
-- vfecha_hora VARCHAR2(50);
vtotal_size_bytes NUMBER;
vtablespace_name VARCHAR2(50);

```

```

BEGIN
  OPEN c FOR
    SELECT TABLESPACE_NAME,
           SUM(CASE WHEN SEGMENT_TYPE IN ('INDEX', 'TABLE', 'FOREIG
N KEY') THEN BYTES ELSE 0 END) AS TOTAL_SIZE_BYTES
    FROM DBA_SEGMENTS
   WHERE OWNER = 'USERDBA'
   GROUP BY TABLESPACE_NAME
   ORDER BY TABLESPACE_NAME;

  LOOP
    FETCH c INTO vtablespace_name, vtotal_size_bytes;
    EXIT WHEN c%NOTFOUND;

    INSERT INTO userdba.segmentos (codigo, fecha_hora, tablespace_na
me, total_size_bytes)
      VALUES (userdba.segmentos_seq.NEXTVAL, TO_CHAR(SYSDATE, 'YY
YY/MM/DD HH24:MI:SS'), vtablespace_name, vtotal_size_bytes);
    END LOOP;
    COMMIT;
    CLOSE C;
  END insertar_segmentos;
/

BEGIN
  insertar_segmentos;
END;
/

```

HWM:

```

--dar permiso para usar ese procedure en userdba
GRANT EXECUTE ON obtener_info_tablespaces TO userdba;

--EN USERDBA
CREATE TABLE ALERTAS_TABLESPACES (

```

```

    alerta_id NUMBER,
    tablespace_name VARCHAR2(50),
    mbusados NUMBER,
    dias NUMBER,
    fecha_alerta VARCHAR2(30),
    constraint pkAler primary key (alerta_id)
)tablespace USERBD;

CREATE SEQUENCE alerta_seq START WITH 1 INCREMENT BY 1;

CREATE OR REPLACE PROCEDURE ALERTA(v_mb_usados NUMBER, v_nombre VARCHAR2, v_mb_libres NUMBER) AS
    v_fecha_alerta VARCHAR2(50);
    promedio NUMBER;
    v_dias NUMBER;
BEGIN
    -- Calcular el promedio de MB usados para tablas con el nombre que entra
    --AVG: calcular el promedio
    SELECT AVG(total_size_bytes)
    INTO promedio
    FROM segmentos
    WHERE tablespace_name = v_nombre;

    v_dias := ROUND((v_mb_libres/ (promedio / (1024 * 1024) ) ),3);

    v_fecha_alerta := TO_CHAR(SYSTIMESTAMP, 'YYYY-MM-DD HH24:MI');
    -- Convertir SYSTIMESTAMP a cadena

    INSERT INTO ALERTAS_TABLESPACES (alerta_id, tablespace_name, mbusados, dias, fecha_alerta)
    VALUES (alerta_seq.NEXTVAL, v_nombre, v_mb_usados, v_dias ,v_fecha_alerta);
END ALERTA;
/

```



```

DECLARE
rc SYS_REFCURSOR;
v_limite_mb NUMBER := 1.10; --en MB, HWM
v_mb_tamano NUMBER;
v_mb_usados NUMBER;
v_mb_libres NUMBER;
v_nombre VARCHAR2(50);
v_status VARCHAR2(50);

BEGIN

    sys.obtener_info_tablespaces(rc);

    LOOP
        FETCH rc INTO v_nombre, v_status, v_mb_tamano, v_mb_usados, v_mb
_libres;
        EXIT WHEN rc%NOTFOUND;

        IF v_mb_usados > v_limite_mb THEN
            ALERTA(v_mb_usados, v_nombre, v_mb_libres );
        END IF;
    END LOOP;

    CLOSE rc;
    COMMIT;
END;
/

```

Primero va a la información general de los tablespaces y revisa quien se pasa 1.10 MB (establecidos por ahora por mi para ver como funciona el alerta):

Tablespace	Estado	MB Tamaño	MB Usados	MB Libres
ADMIBD	ONLINE	8	1,13	6,88
T1	ONLINE	4	1,13	2,88
T2	ONLINE	4	1	3
T3	ONLINE	4	1,13	2,88
T4	ONLINE	4	1,13	2,88
T5	ONLINE	4	1,13	2,88
USERBD	ONLINE	8	1,38	6,63
USERS	ONLINE	5	2,88	2,13

Detecta varios pero usaremos a T1 como ejemplo:

Entonces al ya saber que se paso el tablespace T1, el va a la tabla segmentos y busca a los T1 que guardan el registro de crecimiento de sus llaves y tablas de diferentes días:

	CODIGO	FECHA_HORA	TABLESPACE_NAME	TOTAL_SIZE_BYTES
1	201	2023/09/06 13:54:22	ADMIBD	262144
2	202	2023/09/06 13:54:22	PRUEBA1	65536
3	203	2023/09/06 13:54:22	T1	262144
4	204	2023/09/06 13:54:22	T3	131072
5	205	2023/09/06 13:54:22	T4	131072
6	206	2023/09/06 13:54:22	T5	131072
7	207	2023/09/06 13:54:22	USERBD	524288
8	208	2023/09/06 13:54:22	USERS	851968
9	3	2023/09/06 03:37:44	ADMIBD	262144
10	4	2023/09/06 03:37:44	PRUEBA1	65536
11	5	2023/09/06 03:37:44	T1	262144
12	6	2023/09/06 03:37:44	T4	131072
13	7	2023/09/06 03:37:44	USERBD	393216
14	8	2023/09/06 03:37:44	USERS	851968

Despues con la información general y de la tabla segmentos, el calcula aproximadamente cuantos días le faltan para estar lleno.

Formula: $v_dias := (v_mb_libres / (promedio / (1024 * 1024)))$;

Y esta información se guardara en la tabla Alertas_tablespaces:

	ALERTA_ID	TABSPACE_NAME	MBUSADOS	DIAS	FECHA_ALERTA
1	41	ADMIBD	1,13	27,52	2023-09-06 14:25
2	42	T1	1,13	11,52	2023-09-06 14:25
3	43	T3	1,13	23,04	2023-09-06 14:25
4	44	T4	1,13	23,04	2023-09-06 14:25
5	45	T5	1,13	23,04	2023-09-06 14:25
6	46	USERBD	1,38	15,154	2023-09-06 14:25
7	47	USERS	2,88	2,622	2023-09-06 14:25