Recyclebin flashback 차이점 찾기…

플래시백 끄기…

조회

**SELECT owner, original\_name, object\_name,**

**type, ts\_name, droptime, related, space**

**FROM dba\_recyclebin**

**WHERE can\_undrop = 'YES';**

**SELECT original\_name, object\_name,**

**type, ts\_name, droptime, related, space**

**FROM user\_recyclebin**

**WHERE can\_undrop = 'YES';**

**SHOW RECYCLEBIN**

**Flash back 설정**

**SQL> SHUTDOWN IMMEDIATE;**

**SQL> STARTUP MOUNT EXCLUSIVE;**

**SQL> ALTER SYSTEM SET**

**2 DB\_FLASHBACK\_RETENTION\_TARGET=2880 SCOPE=BOTH;**

**SQL> ALTER DATABASE FLASHBACK ON;**

**SQL> ALTER DATABASE OPEN;**

**ALTER TABLESPACE <ts\_name> FLASHBACK {ON|OFF}**

Flashback 제외시킬 테이블스페이스 설정

* + **查看快速恢复区磁盘限额：**
* **SQL> SELECT estimated\_flashback\_size,**
* **2 flashback\_size  
   3 FROM V$FLASHBACK\_DATABASE\_LOG;**
  + **确定当前闪回窗口：**

**SQL> SELECT oldest\_flashback\_scn,**

**2 oldest\_flashback\_time**

**3 FROM V$FLASHBACK\_DATABASE\_LOG;**

* + **监视闪回数据库日志中的事件记录：**
* **SQL> SELECT \*  
   2 FROM V$FLASHBACK\_DATABASE\_STAT;**

**===========================**

**块损坏故障现象：ORA-01578**

**ORA-01578：“ORACLE data block corrupted (file # %s, block # %s)”：**

**ANALYZE 命令**

**执行逻辑块检查**

**不将块标记为软损坏；只报告软损坏情况**

**验证索引项和表项**

**SQL> ANALYZE TABLE *table\_name* VALIDATE   
 2 STRUCTURE CASCADE;   
SQL> ANALYZE INDEX *index\_name* VALIDATE   
 2 STRUCTURE;**

**实时验证块完整性：**

Show parameter DB\_BLOCK\_CHECKING

보통 false로 설정

**使用 DBMS\_REPAIR**

**1. 检测并报告损坏。**

**SET SERVEROUTPUT ON**

**DECLARE num\_corrupt INT;**

**BEGIN**

**num\_corrupt := 0;**

**DBMS\_REPAIR.CHECK\_OBJECT (**

**schema\_name => ‘HR',**

**object\_name => 'DEPARTMENTS',**

**repair\_table\_name => 'REPAIR\_TABLE',**

**corrupt\_count => num\_corrupt);**

**END;**

**2. 评估 DBMS\_REPAIR 的成本和优势。**

**使用 DBMS\_REPAIR**

在使用 DBMS\_REPAIR 之前，您的首要任务应当是检测并报告损坏。报告不仅要表明出错的块，还要给出相关的修复指令。使用本课中先前介绍的选项之后，可以考虑采用 DBMS\_REPAIR 程序包来修复损坏。CHECK\_OBJECT 过程检查指定的对象，并使用有关损坏以及修复指令的信息来填写修复表。在执行任何 DBMS\_REPAIR 过程之前，必须使用

1. create ADMIN\_TABLES sdfdf
2. **CHECK\_OBJECT**

**SET SERVEROUTPUT O N**

**DECLARE num\_corrupt INT;**

**BEGIN**

**num\_corrupt := 0;**

**DBMS\_REPAIR.CHECK\_OBJECT (**

**schema\_name => ‘HR',**

**object\_name => 'DEPARTMENTS',**

**repair\_table\_name => 'REPAIR\_TABLE',**

**corrupt\_count => num\_corrupt);**

**END;**

1. **使对象变为可用。**

**SET SERVEROUTPUT ON**

**DECLARE num\_fix INT;**

**BEGIN**

**num\_fix := 0;**

**DBMS\_REPAIR.FIX\_CORRUPT\_BLOCKS (**

**schema\_name => 'HR',**

**object\_name => 'DEPARTMENTS',**

**object\_type => DBMS\_REPAIR.TABLE\_OBJECT,**

**repair\_table\_name => 'REPAIR\_TABLE',**

**fix\_count => num\_fix);**

**END;**

END;

1. **修复损坏并重建丢失的数据。**

**SET SERVEROUTPUT ON**

**DECLARE num\_orphans INT;**

**BEGIN**

**num\_orphans := 0;**

**DBMS\_REPAIR.DUMP\_ORPHAN\_KEYS (**

**schema\_name => 'SCOTT',**

**object\_name => 'PK\_DEPT',**

**object\_type => DBMS\_REPAIR.INDEX\_OBJECT,**

**repair\_table\_name => 'REPAIR\_TABLE',**

**orphan\_table\_name => 'ORPHAN\_KEY\_TABLE',**

**key\_count => num\_orphans);**

**DBMS\_OUTPUT.PUT\_LINE('orphan key count: ' ||   
 TO\_CHAR(num\_orphans));**

**END;**

043=> les\_07\_corrupt

按备份标记限制 BMR：

**BLOCKRECOVER TABLESPACE SYSTEM DBA 4194404, 4194405   
FROM TAG "weekly\_backup";**

下面的示例恢复 SYSTEM 表空间中的两个块，并强制从创建时间至少在两天以前的备份中进行还原：

**BLOCKRECOVER TABLESPACE SYSTEM DBA 4194404, 4194405 RESTORE UNTIL TIME 'SYSDATE-2';**

**使用 BLOCKRECOVER 的示例（续）**

下面的示例恢复两个块，并强制使用在 SCN 100 之前执行的备份进行还原：

**BLOCKRECOVER DATAFILE 9 BLOCK 13 DATAFILE 2 BLOCK 19 RESTORE UNTIL SCN 100;**

下面的示例恢复两个块，并强制使用在日志序列 7024 之前执行的备份进行还原：

**BLOCKRECOVER DATAFILE 9 BLOCK 13 DATAFILE 2 BLOCK 19 RESTORE UNTIL SEQUENCE 7024;**

==================================

**V$DATABASE\_BLOCK\_CORRUPTION 视图显示当前损坏的数据库块的列表。**

内存一般。总体的 50%

Sga : 高速缓存区 70%

Process / 3000~2000

Show parameter ca

Alter system set sga\_target 4000m scpe=spfile;

Select \* from v$sgainfo;

Show parameter shared;

Show parameter db\_cache; (db\_cache);

Create pfile from spfile;

**Oracle 内存结构**

与 Oracle 实例关联的基本内存结构包括：

* + - **系统全局区 (SGA)：**由所有服务器进程和后台进程共享。
    - **程序全局区 (PGA)：**由每个服务器和后台进程专用，每个进程都有一个 PGA。

系统全局区 (SGA) 是共享内存区，包含实例的数据和控制信息，由以下各项组成：

* + - **数据库缓冲区高速缓存：**用于缓存从磁盘检索到的数据块。
    - **重做日志缓冲区：**用于缓存重做信息，直到其可以写入磁盘为止。
    - **共享池：**用于缓存可在用户间共享的各种结构。
    - **大型池：**用于缓冲大型 I/O 请求的可选区域，以便支持并行查询、共享服务器、Oracle XA 以及某些类型的备份操作。
    - **Java 池：**用于存放 Java 虚拟机 (JVM) 中特定于会话的 Java 代码和数据。 (升级时用)
    - **流池：**由 Oracle Streams 使用。 （不设置的话一般在共享池）
    - **保留缓冲区高速缓存：**用于存放会尽可能长地保留在缓冲区高速缓存中的数据。
    - **循环缓冲区高速缓存：**用于存放缓冲区高速缓存中很快过期的数据。
    - ***n*K 块大小缓冲区高速缓存：**用于缓存大小与默认数据库块大小不同的数据块，用来支持可传输的表空间。

Show parameter keep **保留缓冲区高速缓存：**

**Show parameter db\_**

**指定 DB\_CACHE\_SIZE 缓冲区高速缓存(sga 内存的70%)**

**DB\_BLOCK\_SIZE**

**DB\_CACHE\_SIZE**

**DB\_RECYCLE\_CACHE\_SIZE**

**DB\_KEEP\_CACHE\_SIZE**

**使用多个缓冲区池**

**CREATE INDEX cust\_idx …**

**STORAGE (BUFFER\_POOL KEEP …);**

**ALTER TABLE oe.customers**

**STORAGE (BUFFER\_POOL RECYCLE);**

**ALTER INDEX oe.cust\_lname\_ix**

**STORAGE (BUFFER\_POOL KEEP);**

**大型池：**

* + - **备份和还原操作**
    - **共享服务器的会话数据**
    - **并行查询消息传送**

**Java 池**

**普通升级时　５１２ｍ**

**一般的时候　０ｍ**

**重做日志缓冲区　（ｒｅｄｏ）**

Show parameter cur open\_cursors

Open\_cursors value 一般　设为２０００

**库高速缓存的内存优化准则**