[http://docs.oracle.com/cd/E11882\_01/server.112/e25494/repair.htm#ADMIN022](http://docs.oracle.com/cd/E11882_01/server.112/e25494/repair.htm" \l "ADMIN022)

修复损坏的数据块，最正确的做法是使用rman备份实现BLOCK MEDIA RECOVERY (BMR)

BLOCK MEDIA RECOVERY (BMR)

RMAN> blockrecover datafile 14 block 56,107,276,517;

RMAN> run {blockrecover datafile 5 block 114 to 118;}

RMAN> recover corruption list ;      --根据v$database\_block\_corruption视图的内容，修复所有的坏块

当数据库没有rman备份的情况下，只能采用一些特殊的方式workaround，根据坏块所影响的对象，有不同的解决方式

找到坏坏所在的段信息

select segment\_name, segment\_type, owner from dba\_extents where file\_id = <Absolute file number> and <corrupted block number> between block\_id and block\_id + blocks -1;

或

rman命令

结合dba\_lobs,dba\_segments，确认坏块影响的对象

如果块不属于任何对象，查询dba\_free\_space 确认坏块是否属于数据文件的可用空间。

Select \* from dba\_free\_space where file\_id= <Absolute file number> and <corrupted block number> between block\_id and block\_id + blocks -1;

示例

SQL> select segment\_name, segment\_type, owner from dba\_extents where file\_id = 6 and 131 between block\_id and block\_id + blocks -1;

SEGMENT\_NAME               SEGMENT\_TYPE      OWNER

------------------------------ ------------------ ------------------------------

EMP                   TABLE          HR

SQL> select owner,segment\_name,segment\_type,bytes,blocks from dba\_segments where segment\_name ='EMP' and owner='HR';

OWNER                   SEGMENT\_NAME              SEGMENT\_TYPE          BYTES    BLOCKS

------------------------------ ------------------------------ ------------------ ---------- ----------

HR                   EMP                  TABLE              65536         8

SQL>

1 如果坏块没有影响任何对象，建议创建虚拟表，并向虚拟表中插入数据以格式化块

mos： How to Format Corrupted Block Not Part of Any Segment (文档 ID 336133.1)

2 如果坏块在对象索引上，删除重建，不会丢失任何数据

For fixing the corruption on the index, we can recreate (not rebuild)

重建完成后使用analyze命令验证

For table:

SQL>analyze table <owner>.<table\_name> validate structure cascade online;

For index:

SQL>analyze index <owner>.<index> validate structure online;

3 如果坏块在对象表上，可以选择重建表或跳过坏块，会丢失坏块所在的数据，只有容忍数据丢失的情况下才可使用此方法

使用datapump导出数据，创建新表，导入数据

示例

SQL> select table\_name,NUM\_ROWS from user\_tables where table\_name='EMP';

TABLE\_NAME             NUM\_ROWS

------------------------------ ----------

EMP          107

[oracle@db ~]$ expdp \'/ as sysdba\' directory=DATA\_PUMP\_DIR dumpfile=hr02.dmp logfile=hr02.expdp tables=HR.EMP

Export: Release 11.2.0.3.0 - Production on Wed Apr 12 11:15:54 2017

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production

With the Partitioning, Automatic Storage Management, OLAP, Data Mining

and Real Application Testing options

Starting "SYS"."SYS\_EXPORT\_TABLE\_01":  "/\*\*\*\*\*\*\*\* AS SYSDBA" directory=DATA\_PUMP\_DIR dumpfile=hr02.dmp logfile=hr02.expdp tables=HR.EMP

Estimate in progress using BLOCKS method...

Processing object type TABLE\_EXPORT/TABLE/TABLE\_DATA

Total estimation using BLOCKS method: 64 KB

Processing object type TABLE\_EXPORT/TABLE/TABLE

Processing object type TABLE\_EXPORT/TABLE/INDEX/INDEX

Processing object type TABLE\_EXPORT/TABLE/INDEX/STATISTICS/INDEX\_STATISTICS

. . exported "HR"."EMP"                                  9.867 KB       9 rows

Master table "SYS"."SYS\_EXPORT\_TABLE\_01" successfully loaded/unloaded

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dump file set for SYS.SYS\_EXPORT\_TABLE\_01 is:

  /u02/app/oracle/admin/orcl/dpdump/hr02.dmp

Job "SYS"."SYS\_EXPORT\_TABLE\_01" successfully completed at 11:16:04

只能导出未损坏的9行数据库，损坏的数据无法导出

drop table emp;

[oracle@db tmp]$ impdp \'/ as sysdba\' directory=DATA\_PUMP\_DIR dumpfile=hr02.dmp logfile=hr02.impdp

Import: Release 11.2.0.3.0 - Production on Wed Apr 12 18:26:05 2017

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production

With the Partitioning, Automatic Storage Management, OLAP, Data Mining

and Real Application Testing options

Master table "SYS"."SYS\_IMPORT\_FULL\_01" successfully loaded/unloaded

Starting "SYS"."SYS\_IMPORT\_FULL\_01":  "/\*\*\*\*\*\*\*\* AS SYSDBA" directory=DATA\_PUMP\_DIR dumpfile=hr02.dmp logfile=hr02.impdp

Processing object type TABLE\_EXPORT/TABLE/TABLE

Processing object type TABLE\_EXPORT/TABLE/TABLE\_DATA

. . imported "HR"."EMP"                                  9.867 KB       9 rows

Processing object type TABLE\_EXPORT/TABLE/INDEX/INDEX

Processing object type TABLE\_EXPORT/TABLE/INDEX/STATISTICS/INDEX\_STATISTICS

Job "SYS"."SYS\_IMPORT\_FULL\_01" successfully completed at 18:26:08

[oracle@db tmp]$

使用DBMS\_REPAIR PLSQL程序包跳过坏块

Oracle提供了DBMS\_REPAIR包用来发现、标识并修改数据文件中的坏块。

这里的修复是有损修复也就是说将受损的数据块标记为坏块，不对其进行访问而已。

任何工具都不是万能的，使用这个包的同时会带来数据丢失、表和索引返回数据不一致，完整性约束破坏等其他问题。因此当出现错误时，应当首先从物理备份或逻辑备份恢复，使用dbms\_repair只是在没有备份的情况下使用的一种手段，这种方式一般都会造成数据的丢失。

有一点需要注意，dbms\_repair包没有进行授权，只有sys用户可以执行

注意：不能用于lob字段！！！

1.创建相应的表对象

使用DBMS\_REPAIR.ADMIN\_TABLES过程创建一个表对象，用于记录需要被修复的表

BEGIN

  DBMS\_REPAIR.ADMIN\_TABLES(TABLE\_NAME => 'REPAIR\_TABLE',

                           TABLE\_TYPE => dbms\_repair.repair\_table,

                           ACTION     => dbms\_repair.create\_action,

                           TABLESPACE => 'USERS');

END;

/

使用DBMS\_REPAIR.ADMIN\_TABLES过程创建一个表对象，用于记录在表块损坏后那些孤立索引，也就是指向坏块的那些索引

BEGIN

  DBMS\_REPAIR.ADMIN\_TABLES(TABLE\_NAME => 'ORPHAN\_KEY\_TABLE',

                           TABLE\_TYPE => DBMS\_REPAIR.ORPHAN\_TABLE,

                           ACTION     => DBMS\_REPAIR.CREATE\_ACTION,

                           TABLESPACE => 'USERS');

END;

/

这两个表的删除可以通过下列存储过程完成

BEGIN

DBMS\_REPAIR.ADMIN\_TABLES (

TABLE\_NAME => 'REPAIR\_TABLE',

TABLE\_TYPE => dbms\_repair.repair\_table,

ACTION => dbms\_repair.drop\_action);

END;

/

BEGIN

DBMS\_REPAIR.ADMIN\_TABLES (

TABLE\_NAME => 'ORPHAN\_KEY\_TABLE',

TABLE\_TYPE => dbms\_repair.orphan\_table,

ACTION => dbms\_repair.drop\_action);

END;

/

2.使用CHECK\_OBJECT过程检测受损对象。

SET SERVEROUTPUT ON

DECLARE num\_corrupt INT;

BEGIN

  num\_corrupt := 0;

  DBMS\_REPAIR.CHECK\_OBJECT(SCHEMA\_NAME       => 'HR',

                           OBJECT\_NAME       => 'EMP2',

                           REPAIR\_TABLE\_NAME => 'REPAIR\_TABLE',

                           CORRUPT\_COUNT     => num\_corrupt);

  DBMS\_OUTPUT.PUT\_LINE('number corrupt: ' || TO\_CHAR(num\_corrupt));

END;

/

number corrupt: 1

--下面我们可以从repair\_table查询到受损的块

--从下面的查询中可以看出列marked\_corrupt全部为true，表明我们在CHECK\_OBJECT已经标注了坏块

COLUMN object\_name FORMAT a10

COLUMN repair\_description FORMAT a28

SET LINES 10000

SELECT object\_name, block\_id, corrupt\_type,marked\_corrupt,repair\_description  FROM repair\_table;

OBJECT\_NAM   BLOCK\_ID CORRUPT\_TYPE MARKED\_COR REPAIR\_DESCRIPTION

---------- ---------- ------------ ---------- ----------------------------

EMP2          171          6148 TRUE       mark block software corrupt

注意：在8i下，check\_object只会检查坏块，MARKED\_CORRUPT为false，故需要进行 定位坏块（fix\_corrupt\_blocks） ，修改MARKED\_CORRUPT为true，同时更新CHECK\_TIMESTAMP。 9i以后经过check\_object，MARKED\_CORRUPT的值已经标识为TRUE了。所以可以直接进行第四步了。

3 标记坏块

--过程FIX\_CORRUPT\_BLOCKS用于标记坏块，在这个演示中，我们在CHECK\_OBJECT已经被标注了，如没有执行下面的过程

--由于上一步已经标注，所以下面的输出为0

SET SERVEROUTPUT ON

DECLARE num\_fix INT;

BEGIN

  num\_fix := 0;

  DBMS\_REPAIR.FIX\_CORRUPT\_BLOCKS(SCHEMA\_NAME       => 'HR',

                                 OBJECT\_NAME       => 'EMP2',

                                 OBJECT\_TYPE       => dbms\_repair.table\_object,

                                 REPAIR\_TABLE\_NAME => 'REPAIR\_TABLE',

                                 FIX\_COUNT         => num\_fix);

  DBMS\_OUTPUT.PUT\_LINE('num fix: ' || TO\_CHAR(num\_fix));

END;

/

num fix: 0

PL/SQL 过程已成功完成

4.执行SKIP\_CORRUPT\_BLOCKS过程，使后续DML操作跳过坏块，我们前面虽然定位了坏块，但是，如果我们访问table依然报错：

SQL> select \* from emp2;

select \* from emp2

              \*

ERROR at line 1:

ORA-01578: ORACLE data block corrupted (file # 6, block # 171)

ORA-01110: data file 6: '/u02/app/oracle/tbs\_block\_corrupt01.dbf'

BEGIN

  DBMS\_REPAIR.SKIP\_CORRUPT\_BLOCKS(SCHEMA\_NAME => 'HR',

                                  OBJECT\_NAME => 'EMP2',

                                  OBJECT\_TYPE => dbms\_repair.table\_object,

                                  FLAGS       => dbms\_repair.skip\_flag);

END;

/

SQL> select employee\_id from emp2;

EMPLOYEE\_ID

-----------

    189

    190

    191

    192

    193

    194

    195

    196

    197

9 rows selected.

DML可正常操作出未受影响的行，但坏块所在的行已经丢失。原表107行数据，丢失了98行数据

注意：针对影响system/sysaux表空间的的数据块损坏，原则是重建数据库！

注意：本文没有涉及坏块影响lob段，lob段移动方法

参考：

example <http://blog.csdn.net/leshami/article/details/10616687>

syntax <http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/d_repair.htm#CHDHHIAB>

**Extracting Data from a Corrupt Table using DBMS\_REPAIR or Event 10231 (Doc ID 33405.1)1**

**DBMS\_REPAIR SCRIPT (Doc ID 556733.1)**

DBMS\_REPAIR PL/SQL package

<http://docs.oracle.com/cd/B19306_01/server.102/b14231/repair.htm#ADMIN022>