

不完全恢复：

一、需要注意实现：

- 1、如果数据库联机日志出现了问题，并且不知道数据库关闭不知道是按正常关闭 (shutdown immediate) 还是不正常关闭 (shutdown abort) 的。首先先用正常关闭的恢复方法恢复数据库。
- 2、正常关闭操作和非正常操作是有区别的，因为 shutdown immediate 执行时会执行检查点，数据已经从内存回写到数据库中，此时如果联机日志出现问题，数据文件其实是正常的。而 abort 却不能。
- 3、shutdown immediate 不用执行 restore 命令拷贝数据文件到制定位置，直接执行 recover 命令恢复数据库即可。
- 4、如果采用日志序号的方式不完全恢复数据，可以用 archive log list 察看当前的日志文件序号，在恢复的时候注意 recover database until sequence 后面跟的日志序号要比刚刚察看到的日志序号的数字大 1。可以多加但是不能少加，少加在恢复的时候会报错。
- 5、在做不完全恢复的时候只能用 restore database 和 recover database，不能用 restore datafile (或者 tablespace)。datafile 和 tablespace 只能在完全恢复的时候用到。
- 6、每次不完全恢复后，都要做一次完全备份。因为日志序列号被重置了。
- 7、在做不完全恢复之前，建议把数据库相关的日志，控制，数据文件拷贝一份。因为没做一次不完全备份，联机日志都被重置。如果想多做几次的话如果没有备份联机日志是没有的。

二、基于日志序列号的恢复举例：

- 1、当联机日志被破坏后 (删除，或者不能读写了) 的不完全恢复。假设在关闭数据库的时候采用的是 shutdown immediate，此时数据已经被回写到数据文件中。

```
startup mount
```

启动实例并且挂在数据库。

```
archive log list
```

察看当前的联机日志号。加入当前的日志序列号是 10

```
RMAN> recover database until sequence 11 thread 1
```

恢复数据库，thread 这个参数在 oracle rac 的时候才能用到，它的值是根据 rac 的配置而设置的，这里采用 1。

```
RMAN> alter database open resetlogs
```

打开数据库，并且把日志重置。此时需要注意，数据库的日志被重置后，他们的序号也被重置了，此时的日志序号是从 1 开始。所以在做一次不完全恢复后，应该给数据库做一次完全的备份。

- 2、当联机日志被破坏后 (删除，或者不能读写了) 的不完全恢复。假设在关闭数据库的时候采用的是 shutdown abort，此时数据库的日志文件已经被损坏，并且数据没有被回写到数据文件中。数据文件和控制文件的 SCN 号不一样。这时就不要用到 restore 命令拷贝数据文件了。

准备工作：

```
insert into test1(a1,a2) values(1,1)
```

在一个表中插入数据，

```
commit
```

```
alter system switch logfile
```

切换归档日志

```
insert into test1(a1,a2) values(2,2)
```

```
commit
```

不切换归档日志。

```
shutdown abort
```

强制关机。

```
host rm /opt/oracle/oradata/orcl/*.log
```

删除联机日志文件。

```
startup mount
```

启动实例并且挂在数据库。

```
archive log list
```

察看当前日志文件的序列号。

```
restore database
```

拷贝数据文件。

```
recover database until sequence 5 thread 1
```

恢复日志文件到制定的日志序列号。

```
alter database open resetlogs;
```

打开数据库并且重置日志。

三、基于时间点的恢复举例：

1、删表空间如果用这种自动备份的方法的话，如果把当前的表空间删除了，那备份文件的表空间也没有了。

```
select to_char(sysdate,'yyyy-mm-dd hh24:mi:ss') from dual;
```

察看当前的系统时间。

```
drop table test1;
```

删除某张表。

```
shutdown immediate;
```

关闭数据库。

```
startup mount
```

启动实例并且挂在数据库。

在新终端下设定系统环境变量

```
export NLS_DATE_FORMAT='yyyy-mm-dd hh24:mi:ss'    在 windows 上只设定这个环境变量就可以了。
```

```
export NLS_LANG=american
```

设定完环境变量后需要启动到 rman 中。rman target /

```
RMAN> restore database
```

```
RMAN> recover database until time '2012-09-16 01:51:04';
```

```
RMAN> alter database open resetlogs;
```

四、基于系统改变号的不完全恢复

```
select dbms_flashback.get_system_change_number from dual;
```

dbms_flashback 是包的名字，后面的是存储过程的名字。

```
RMAN> restore database
```

```
RMAN> recover database until scn 984677
```

```
RMAN> alter database open resetlogs;
```

五、如果控制文件、数据文件和日志文件都被删除了，如果做恢复处理。

准备工作：

1、控制文件在之前必须在 RMAN 中设定成自动备份才可以。

```
RMAN> configure controlfile autobackup on;
```

```
RMAN> backup database plus archivelog;
```

```
insert into test1(a1,a2) values(100,100)
```

在一个表中插入数据，

```
commit
```

```
alter system switch logfile
```

切换归档日志

```
insert into test1(a1,a2) values(200,200)
```

```
commit
```

不切换归档日志。

```
shutdown abort
```

强制关机。

```
rman target /
```

```
RMAN> startup nomount
```

```
RMAN> restore controlfile from autobackup;
```

```
RMAN> alter database mount;
```

```
RMAN> restore database (拷贝备份的数据文件)
```

```
RMAN> recover database
```

```
RMAN> alter database open resetlogs;
```

附加内容：

六、控制文件的操作。

控制文件：

如果控制文件是好的时候可以用 list backup of controlfile 察看控制文件的列表。

自动回复控制文件：

```
RMAN> restore controlfile from autobackup
```

```
RMAN> restore controlfile;
```

七、联机日志文件的保护。

oracle 为了防止联机日志内的数据都被清除，有一个机制是把在线联机日志放在了快速恢复区中。

可以为数据库创建一个日志，日志的保存位置是快速恢复区，路径为

`$ORACLE_BASE/flash_recovery_area/$ORACLE_SID/onlinelog`

八、delete 一个表和 truncate 一个表的差别

1、truncate 是一个 DDL 语句，而 delete 是一个 DML 语句。

2、truncate 执行后立即提交数据被清空，而 delete 会把数据提交到回滚段中如果没有执行 commit 提交就会恢复数据。

3、truncate 执行后立即释放之前表占用的存储空间，而 delete 是不释放的。

九、用 impdp 的时候可以用 REMAP_* 参数来指定导入的位置。把原 users 表空间中的数据转到 sysaux 表空间；把原 sysaux 表空间的数据转到 users 表空间。

```
$ impdp system/oracle schemas=scott directory=d1 dumpfile=scott.dmp  
job_name=import_scott remap_tablespace=users:sysaux,sysaux:users
```