目录

[1 引言 2](#_Toc253060082)

[**1.1** 数据库的运行模式： 2](#_Toc253060083)

[**1.1.1** 归档方式（ARCHIVELOG） 2](#_Toc253060084)

[**1.1.2** 非归档方式(NOARCHIVELOG) 2](#_Toc253060085)

[2 数据库备份 2](#_Toc253060086)

[**2.1** 逻辑备份： 2](#_Toc253060087)

[**2.1.1** exp 2](#_Toc253060088)

[**2.1.2** Expdp 5](#_Toc253060089)

[**2.2** 物理备份 7](#_Toc253060090)

[**2.2.1** 冷备份 7](#_Toc253060091)

[**2.2.2** 热备份 7](#_Toc253060092)

[3 数据库恢复 14](#_Toc253060093)

[**3.1** 逻辑恢复 14](#_Toc253060094)

[**3.1.1** Imp导入 14](#_Toc253060095)

[**3.1.2** Impdp导入 15](#_Toc253060096)

[**3.2** Rman恢复 17](#_Toc253060097)

[**3.2.1** 恢复案例 17](#_Toc253060098)

# 引言

ORACLE数据库是一种大型关系型的数据库，可以存贮达到存贮TB的数据，那么如何保证这些数据的安全尤其至关重要，需要制定一整套的ORACLE数据库的备份机制

## 数据库的运行模式：

ORACLE数据库有两种运行模式

### 归档方式（ARCHIVELOG）

归档方式的目的是当数据库发生故障时最大限度恢复所有已提交的事物；

### 非归档方式(NOARCHIVELOG)

恢复数据库到最近的正常点。

我们根据数据库的高可用性和用户可承受丢失的工作量的多少，一般情况把正式运行的数据库采用为归档方式；那些正在开发和调试的数据库采用不归档方式。

我们公司平台的数据库，这两种运行方式都有，所以我们采取的备份方式都是有区别的；

# 数据库备份

## 逻辑备份：

数据库的逻辑备份有三种模式：完全备份，用户模式备份，表备份

### exp

exp是oracle软件附带的一个数据导出的命令行工具，他的位置一般是位于$ORACLE\_HOME/bin/exp

这个命令行本身为我们提供了详细的参数说明，可以使用下面的方式看帮助说明：

$exp help=y

#### 参数说明

USERID username/password 指定执行导出操作的用户名和口令

BUFFER size of data buffer 指定导出的buffer 大小

FULL export entire file (N) 是否全量导出默认是N

OWNER list of owner usernames 用户模式导出

TABLES list of table names 表导出

FILE output files (EXPDAT.DMP) 导出的文件名，习惯上后缀名为.dmp

LOG log file of screen output 导出时的日志记录文件

ROWS export data rows (Y) 导出表的数据行

CONSISTENT cross-table consistency(N) 导出时是否保持关联表的数据一致性，默认为N

COMPRESS import into one extent (Y) 是否导出到一个区

GRANTS export grants (Y) 导出授权，默认Y

INCTYPE incremental export type 增量导出，注：已经不用做增量备份，没有实际意义

INDEXES export indexes (Y) 导出所以，默认Y

DIRECT direct path (N) 直接路径导出

TRIGGERS export triggers (Y) 导出触发器，默认Y

PARFILE parameter filename 导出参数文件，可以把控制参数写入到一个文件

CONSTRAINTS export constraints (Y) 导出约束关系，默认Y

#### 全量导出

导出数据库所有的数据对象和数据行

$exp userid=sms/smspass full=Y buffer=102400 file=/database/oracle\_backup/sms\_exp\_full\_20100201.dmp log=/database/oracle\_backup/sms\_exp\_full\_20100201.log

如果是不导出数据行可以使用控制参数rows=N

#### 用户模式导出

导出数据库用户sms中的所有对象和数据行

$exp userid=sms/smspass owner=sms buffer=102400 file=/database/oracle\_backup/sms\_exp\_owner\_20100201.dmp log=/database/oracle\_backup/sms\_exp\_owner\_20100201.log

如果是不导出数据行可以使用控制参数rows=N

#### 表导出

导出指定表的数据

$exp userid=sms/smspass buffer=102400 tables=(sms\_mt\_report,sms\_batch\_log) file=/database/oracle\_backup/sms\_exp\_tbs\_20100201.dmp log=/database/oracle\_backup/sms\_exp\_tbs\_20100201.log

如果是不导出数据行可以使用控制参数rows=N

#### 适用环境

Exp导出备份，适用于所有oracle数据库版本和应用；

1. 常用于数据量小的数据库做逻辑导出备份，
2. 生成的文件为二进制格式，对于跨平台的，小规模数据库数据移植时也比较常用
3. 可以在oracle服务器上运行exp，也可以在能够连接到数据库的另外一台服务器上运行exp，exp是作为客户端运行；
4. 运行exp的的客户端的NLS\_LANG，此环境变量的值需要和数据库的字符集相同；

#### 备份策略

对于小规模的数据库，业务空闲的时间段，使用exp进行数据库逻辑导出备份；可以根据业务情况定时进行操作，可以每天一次，也可以每天两次，备份频率需要根据业务和服务器的情况来定！

我们平台的数据库逻辑备份，基本上是备份策略：

##### 对于数据量不大的应用，一般在15G以内的数据量

我们采用用户模式导出！用户下的所有对象和数据行；本数据库服务器上保留3天的导出文件，同时需要把导出备份文件传输到其他的服务器上保留7天的备份

##### 对于数据量超过15G的应用

我们采用用户模式导出：

先导出备份用户下面的所有对象不导数据行；

再做重点表的数据行导出；

本数据库服务器上保留3天的导出文件，同时需要把导出备份文件传输到其他的服务器上保留7天的备份

#### 备份部署

通过shell脚本配合crontab 定时执行备份

具体备份脚本可以参考下面的例子：

|  |
| --- |
| #!/bin/sh  FILENAME="\_BJG\_`date +%Y%m%d.dmp`"  username="xxx"  password="xxxpass"  ftpuser="ftp1"  ftppass="ftp1pass!#"  exp\_dir="/backup/exp\_backup/xqw"  FILENAME1="$username$FILENAME.gz"  FILENAME2="BJG$username`date +%Y%m%d.log`"  PF="$exp\_dir/$username$FILENAME"  echo $PF  echo "starting export,this will take a few minutes,please wait"  exp userid=$username/$password owner=$username buffer=10240000 direct=y file=$PF rows=Y log=$exp\_dir/$FILENAME2  if [ -f $PF ]; then  echo "full export successfully, now compress the export file"  gzip $PF  PF2=$PF.gz  echo $PF2  if [ -f $PF2 ]; then  echo "`date +%F' '%r` export successful, and export file is $FILENAME ">>/$exp\_dir/fullbackup\_log  echo "Now starting to FTP ..........."  ftp -i -n 192.168.236.9<<FTP  user $ftpuser $ftppass  bin  lcd $exp\_dir  cd /data/exp\_backup/orcbak/xqw  put $FILENAME1  bye  FTP  echo "ftp successfully"  echo "ftp successfully" >>$exp\_dir/fullbackup\_log  else  echo "`date +%F' '%r` ERROR! when gzip the export file." >>$exp\_dir/fullbackup\_log  fi  else  echo "`date +%F' '%r` ERROR! when export file." >>$exp\_dir/fullbackup\_log  fi |

### Expdp

Oracle10G版本，提供的一个高速的数据泵技术，在oracle服务端运行！

#### 参数说明

USERID 必须是命令行中的第一个参数。

关键字 说明 (默认)

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

ATTACH 连接到现有作业, 例如 ATTACH [=作业名]。

COMPRESSION 减小有效的转储文件内容的大小

关键字值为: (METADATA\_ONLY) 和 NONE。

CONTENT 指定要卸载的数据, 其中有效关键字为:

(ALL), DATA\_ONLY 和 METADATA\_ONLY。

DIRECTORY 供转储文件和日志文件使用的目录对象。

DUMPFILE 目标转储文件 (expdat.dmp) 的列表,

例如 DUMPFILE=scott1.dmp, scott2.dmp, dmpdir:scott3.dmp。

ENCRYPTION\_PASSWORD 用于创建加密列数据的口令关键字。

ESTIMATE 计算作业估计值, 其中有效关键字为:

(BLOCKS) 和 STATISTICS。

ESTIMATE\_ONLY 在不执行导出的情况下计算作业估计值。

EXCLUDE 排除特定的对象类型, 例如 EXCLUDE=TABLE:EMP。

FILESIZE 以字节为单位指定每个转储文件的大小。

FLASHBACK\_SCN 用于将会话快照设置回以前状态的 SCN。

FLASHBACK\_TIME 用于获取最接近指定时间的 SCN 的时间。

FULL 导出整个数据库 (N)。

HELP 显示帮助消息 (N)。

INCLUDE 包括特定的对象类型, 例如 INCLUDE=TABLE\_DATA。

JOB\_NAME 要创建的导出作业的名称。

LOGFILE 日志文件名 (export.log)。

NETWORK\_LINK 链接到源系统的远程数据库的名称。

NOLOGFILE 不写入日志文件 (N)。

PARALLEL 更改当前作业的活动 worker 的数目。

PARFILE 指定参数文件。

QUERY 用于导出表的子集的谓词子句。

SAMPLE 要导出的数据的百分比;

SCHEMAS 要导出的方案的列表 (登录方案)。

STATUS 在默认值 (0) 将显示可用时的新状态的情况下,

要监视的频率 (以秒计) 作业状态。

TABLES 标识要导出的表的列表 - 只有一个方案。

TABLESPACES 标识要导出的表空间的列表。

TRANSPORT\_FULL\_CHECK 验证所有表的存储段 (N)。

TRANSPORT\_TABLESPACES 要从中卸载元数据的表空间的列表。

VERSION 要导出的对象的版本, 其中有效关键字为:

(COMPATIBLE), LATEST 或任何有效的数据库版本。

下列命令在交互模式下有效。

注: 允许使用缩写

命令 说明

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

ADD\_FILE 向转储文件集中添加转储文件。

CONTINUE\_CLIENT 返回到记录模式。如果处于空闲状态, 将重新启动作业。

EXIT\_CLIENT 退出客户机会话并使作业处于运行状态。

FILESIZE 后续 ADD\_FILE 命令的默认文件大小 (字节)。

HELP 总结交互命令。

KILL\_JOB 分离和删除作业。

PARALLEL 更改当前作业的活动 worker 的数目。

PARALLEL=<worker 的数目>。

START\_JOB 启动/恢复当前作业。

STATUS 在默认值 (0) 将显示可用时的新状态的情况下,

要监视的频率 (以秒计) 作业状态。

STATUS[=interval]

STOP\_JOB 顺序关闭执行的作业并退出客户机。

STOP\_JOB=IMMEDIATE 将立即关闭

数据泵作业。

创建目录对象expdir

create or replace directory expdir as '/home/oracle/expdir';

[grant](http://www.adp-gmbh.ch/ora/sql/grant.html) read, write on directory backup to user;

#### 全量导出

导出数据库所有的对象和数据行

$ expdp sms/smspass DIRECTORY=expdir DUMPFILE=sms\_expdp\_full\_20100201.dmp full=y JOB\_NAME=expdp\_job1 content=all PARALLEL=2 logfile=sms\_expdp\_full\_20100201.log

如果仅导出数据库对象不导出数据，可以使用content= METADATA\_ONLY

#### 用户模式导出

导出数据库sms用户的对象和数据行

$ expdp sms/smspass DIRECTORY=expdir DUMPFILE=sms\_expdp\_owner\_20100201.dmp SCHEMAS=sms JOB\_NAME=expdp\_job1 content=all PARALLEL=2 logfile=sms\_expdp\_owner\_20100201.log

如果仅导出数据库对象不导出数据，可以使用content= METADATA\_ONLY

#### 表导出

导出数据库表和数

$ expdp sms/smspass DIRECTORY=expdir DUMPFILE=sms\_expdp\_tabs\_20100201.dmp TABLES=(sms\_mt\_report,sms\_batch\_log) JOB\_NAME=expdp\_job1 content=all PARALLEL=2 logfile=sms\_expdp\_tabs\_20100201.log

如果仅导出数据库对象不导出数据，可以使用content= METADATA\_ONLY

#### 适用环境

Expdp导出备份，适用于oracle数据库10G；

1. 常用于数据量小的数据库做逻辑导出备份，
2. 生成的文件为二进制格式，对于跨平台的，小规模数据库数据移植时也比较常用
3. 可以在oracle服务器上运行exp，也可以在能够连接到数据库的另外一台服务器上运行exp，exp是作为客户端运行；
4. 运行exp的的客户端的NLS\_LANG，此环境变量的值需要和数据库的字符集相同；

#### 备份策略

同exp的备份策略一样！

## 物理备份

物理备份就是转储ORACLE物理文件(如数据文件、控制文件、归档日志文件等)，一旦数据库发生故障，可以利用这些文件进行还原；

### 冷备份

所谓冷备份，如果业务运行中断，那么可以把数据库关闭，然后把数据库相关的文件（数据文件，重做文件，控制文件，参数文件等）进行打包备份到其他的磁盘或者磁带上！

#### 使用环境

这种备份方式需要停止数据库，一般在线平台很少运行中断服务！所以这种方式的备份很少；我们平台的数据库都没有用这种方式备份！

### 热备份

热备份可以理解为不需要停机，就可以进行的的物理备份，这个只是个相对的叫法而已！

#### RMAN简介

RMAN(Recovery Manager)，也是一种物理备份，不是对象级的逻辑备份，RMAN 可以用来备份和恢复数据库文件、归档日志和控制文件，也可以用来执行完全或不完全的数据库恢复。

是DBA的一个重要工具，用于备份、还原和恢复oracle数据库

#### 适用环境

1. 大规模数据库，对数据的可恢复性要求较高，或者说是不允许丢失数据的情况下
2. 目标数据库
3. Rman执行命令
4. Rman备份必须使目的数据库运行在归档模式，否则不能够使用rman进行备份！

#### RMAN 功能特点

1）功能类似物理备份，但比物理备份强大，从下面的特点可以看到；

2）可以压缩空块；

3）可以在块水平上实现增量；

4）可以把备份的输出打包成备份集，也可以按固定大小分割备份集；

5）备份与恢复的过程可以自动管理；

6）可以使用脚本（存在Recovery catalog 中）

7）可以做坏块监测

#### 术语解释

##### Backup sets 备份集

备份集顾名思义就是一次备份的集合，它包含本次备份的所有备份片，以oracle专有的格式保存。一个备份集根据备份的类型不同，可能构成一个完全备份或增量备份。

##### Backup pieces 备份片

一个备份集由若干个备份片组成。每个备份片是一个单独的输出文件。一个备份片的大小是有限制的；如果没有大小的限制， 备份集就只由一个备份片构成。备份片的大小不能大于你的文件系统所支持的文件长度的最大值。

##### Image copies 映像备份

映像备份不压缩、不打包、直接COPY 独立文件（数据文件、归档日志、控制文件），类似操作系统级的文件备份。而且只能COPY 到磁盘，不能到磁带。可以作为增量备份的0级，一般用来备份控制文件。

##### Full backup 全备份

全备份是指对数据文件中使用过的数据块进行备份，没有使用过的数据块是不做备份的，也就是说，RMAN 进行备份是进行压缩的。

##### Incremental backup 增量备份

增量备份是指备份数据文件中自从上一次同一级别的或更低级别的备份以来被修改过的数据块。与完全备份相同，增量备份也进行压缩。增量备份虽然概念很好，备份也很简单，但恢复起来却是另外一回事，不仅麻烦而且容易出错，所以实际中越来越少使用，到了Oracle 9i已经建议不再使用，以后版本会逐渐取消对增量备份的支持。

##### Recovery catalog 恢复目录

恢复目录用于记录RMAN 使用过程中的控制信息，恢复目录应该经常被同步。尽管我们可以使用nocatalog 方式来使用RMAN，此时控制信息记录在目标数据库的控制文件中，但这样毕竟不安全，因为一旦目标数据库的控制文件损坏就意味着所有的RMAN 备份失效。同样的道理恢复目录应该建立在另外一个数据库中

#### Rman使用

##### Rman配置参数

我们可以通过使用show all 查看rman默认的配置参数

RMAN> show all;

RMAN configuration parameters are:

CONFIGURE RETENTION POLICY TO REDUNDANCY 2;

CONFIGURE BACKUP OPTIMIZATION ON;

CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default

CONFIGURE CONTROLFILE AUTOBACKUP ON;

CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/database/oradir/rbackup/level0\_%F.ctl';

CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default

CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default

CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default

CONFIGURE MAXSETSIZE TO UNLIMITED; # default

CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default

CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/home/oracle/product/dbs/snapcf\_orc1.f'; # default

我们可以使用configure命令来修改默认的配置

###### 备份策略保持,分两个保持策略,一是时间策略,二是冗余策略, 注：这个两个策略是不能同时生效的！

RMAN>configure retention policy to recovery window of 5 days;

RMAN>configure retention policy to redundancy 3;

RMAN>configure retention policy to none;

RMAN>configure retention policy to clear;

###### 控制文件的自动备份

RMAN>configure controlfile autobackup on;

###### 设置并行备份

RMAN>configure device type disk parallelism 4;

###### 通道配置与自动通道配置;

自动配置通道

RMAN> CONFIGURE DEFAULT DEVICE TYPE TO DISK;

手动配置通道，会覆盖自动配置

RMAN>allocate channel c1 device type disk；

###### 配置备份优化

RMAN>configure backup optimization on;

##### 常用操作

###### Report命令

report命令,用来检查哪些文件需要备份,哪些文件能被删除,以及哪些文件能不能获得信息;

如报告数据库的所有能备份的数据文件对象;

RMAN>report schema;

or

RMAN>report schema at time 'sysdate-14';

RMAN>report schema at scn 1000;

报告需要备份的数据文件

RMAN>report need backup [ redundancy|days|incremental n ];

报告过期了的数据文件或者不可用的备份和拷贝;

RMAN>report obsolete

报告不能获得的或者不能达到的数据文件信息;

RMAN>report unrecoverable [ database ]

###### Backup命令

最简单的执行数据库备份操作

RMAN>BACKUP DATABASE;

备份表空间

RMAN>backup tablespace system;

备份归档日志

RMAN>backup archivelog all delete input;

指定备份集存储路径并且备份归档日志

RMAN>backup database include current controlfile

format '/database/rman/%U\_%s.bak' tag = 'full’

PLUS ARCHIVELOG format '/database/rman/%U\_%s.archbak' tag = 'archivelog'

DELETE ALL INPUT;

###### List 命令

一般用来查看备份与拷贝信息,

RMAN>list backup;

查看备份汇总信息;

RMAN>list backup summary;

查看文件拷贝信息

RMAN>list copy;

###### crosscheck命令

检查磁盘或磁带上的备份或者拷贝是否正确,并更新备份或者拷贝的状态,如果不正确,将标记为expired过期;

RMAN>crosscheck backup;

RMAN>crosscheck archivelog all;

###### delete命令

可以用来删除指定的备份或者用来删除废弃或者是过期的备份集

RMAN>delete expired backup;

RMAN>delete noprompt obsolete;

RMAN>delete noprompt archivelog until time 'sysdate-2';

rman>crosscheck archivelog all;

rman>delete noprompt expired copy;

###### Exclude命令

排除某些不需要备份的表空间。

这样可以缩减备份的容量，对备份进行适当优化和调整。排除某个表空间可以使用类似如下命令：

RMAN>configure exclude for tablespace users;

其设置可以在RMAN中通过show exclude命令来查看。如果将来想临时增加排除表空间的备份，可以通过以下命令实现：

RMAN>backup database noexclude;

如果想取消排除设置则可以使用如下命令：

RMAN>CONFIGURE EXCLUDE FOR TABLESPACE users CLEAR;

###### Restore命令

验证数据库备份

RMAN>restore database validate;

###### Recover命令

恢复数据库

RMAN>RECOVER DATABASE

##### 不使用恢复目录

登录连接,操作系统认证

$Rman target / nocatalog

口令认证需要使用tnsname建立的服务名 如服务名testdb 能够连接到目的数据库，testrman能够连到恢复目录所在的数据库

$Rman target system/system@testdb nocatalog

##### 使用恢复目录

登录连接,操作系统认证

$Rman target / catalog rman/rmanpass

口令认证需要使用tnsname建立的服务名 如服务名testdb 能够连接到目的数据库，testrman能够连到恢复目录所在的数据库

$Rman target system/system@testdb catalog rman/rmanpass@testrman

###### 创建恢复目录的表空间和用户

最好的建议是把恢复目录创建在和目标数据不同的server上，具体操作如下（以下操作是在同一个数据库进行的演示！）

创建恢复目录存成的表空间

SQL> create tablespace rmants datafile '/home/oracle/oradata/orc1/rman01.dbf' size 100M autoextend on maxsize 500M;

创建用户rman

SQL> create user rman identified by rmanpass default tablespace rmants;

用户授权

SQL> grant connect,resource,recovery\_catalog\_owner to rman;

###### 创建恢复目录并注册数据库，

首先连接到目标数据库和恢复目录

$rman target / catalog rman/rmanpass

Recovery Manager: Release 10.1.0.2.0 - Production

Copyright (c) 1995, 2004, Oracle. All rights reserved.

connected to target database: ORC1 (DBID=1195607395)

connected to recovery catalog database

recovery catalog is not installed

RMAN>

这里提示说recovery catalog is not installed,说明我们还没有创建恢复目录，使用下面的命令创建；

RMAN> create catalog tablespace rmants;

recovery catalog created

RMAN> register database;

database registered in recovery catalog

starting full resync of recovery catalog

full resync complete

注册完成！

可以使用恢复目录进行数据库的备份和恢复操作了！

#### 备份策略

##### 全备份

每次执行ramn生成数据库的全量备份！全量备份会备份数据库所有占用的数据块！在备份归档日志的时候不会跳过数据块，备份集往往比较大，需要较多的备份存储空间！

这种全备不能作为增量备份的基础备份！

##### 增量备份

增量备份可以在一段时间内，比如从周日到周六，周一做一次全量的rman备份，其后每天可以做增量备份，只备份产生变化的数据块！根据对变化的数据块的不同参照可以分成下面两种；

增量备份分成好多级别，等级数字是0，1，2，3等整数表示，其中0级增量备份是其他级别增量备份的基础；如果在等级增量备份是0级的备份集部存在，那么rman会自动产生0级别的增量备份。

下表是一个礼拜的增量备份策略

|  |  |  |
| --- | --- | --- |
| 星期 | 差异增量 | 累积增量 |
| 星期天 | 0级 | 0级 |
| 星期一 | 2级 | 2级 |
| 星期二 | 2级 | 2级 |
| 星期三 | 1级 | 1级 |
| 星期四 | 2级 | 2级 |
| 星期五 | 2级 | 2级 |
| 星期六 | 2级 | 2级 |
| 星期天 | 0级 | 0级 |

###### 累积增量备份

是备份上级及同级备份以来所有变化的数据块，差异增量是默认增量备份方式；所以累积增量需要更多的备份时间，同时需要较小的恢复时间

零级备份

RMAN>backup incremental level 0 database;

一级累计增量

RMAN>backup incremental level 1 cumulative database;

###### 差量增量备份

是备份上级备份以来所有变化的块，可以备份同级备份以来变化的数据块。所以会减少备份时间，但需要更多的恢复时间

零级备份

RMAN>backup incremental level 0 database;

一级差异增量

RMAN>backup incremental level 1 database;

公司有部分平台使用的rman进行备份，是用的差异的增量备份策略,每周日凌晨2点做全备份，周一到周六凌晨2点做差异增量备份！

#### 部署备份

Rman增量备份的部署案例，可以参考下面的

|  |
| --- |
| #!/sbin/sh  if [ `date +%u` -eq 7 ];  then  echo "------------------------------Backup Staring -----------------------------L0";date  $ORACLE\_HOME/bin/rman msglog=/app/backup/rman/44backup.log << EOF  connect target /  delete noprompt obsolete;  backup database include current controlfile  format '/database/44rman/%U\_%s.bak' tag = 'level 0' filesperset = 2  PLUS ARCHIVELOG format '/database/44rman/%U\_%s.archbak' tag = 'level 0 archivelog' filesperset = 2  DELETE ALL INPUT;  list backup;  restore database VALIDATE;  exit  EOF  echo "------------------------------end-----------------------------L0";date  else  echo "------------------------------Backup Staring -----------------------------L1";date  $ORACLE\_HOME/bin/rman msglog=/app/backup/rman/44backup.log << EOF  connect target /  delete noprompt obsolete;  backup incremental level 1 database include current controlfile format '/database/44rman/level1\_%U\_%s.bak' tag = 'level 1 db';  sql 'alter system archive log current';  backup archivelog all format '/database/44rman/level1\_%d\_%s\_%p\_%u.arcbak' tag = 'level 1 archivelog' delete all input;  exit  EOF  echo "------------------------------end-----------------------------L1";date  fi |

# 数据库恢复

## 逻辑恢复

### Imp导入

由exp导出的dmp文件，在进行数据移植，或者恢复时，需要使用Imp 工具导入到数据库中！

#### 参数说明：

USERID 必须是命令行中的第一个参数。

关键字 说明 (默认值) 关键字 说明 (默认值)

USERID 用户名/口令 FULL 导入整个文件 (N)

BUFFER 数据缓冲区大小 FROMUSER 所有者用户名列表

FILE 输入文件 (EXPDAT.DMP) TOUSER 用户名列表

SHOW 只列出文件内容 (N) TABLES 表名列表

IGNORE 忽略创建错误 (N) RECORDLENGTH IO 记录的长度

GRANTS 导入权限 (Y) INCTYPE 增量导入类型

INDEXES 导入索引 (Y) COMMIT 提交数组插入 (N)

ROWS 导入数据行 (Y) PARFILE 参数文件名

LOG 屏幕输出的日志文件 CONSTRAINTS 导入限制 (Y)

DESTROY 覆盖表空间数据文件 (N)

INDEXFILE 将表/索引信息写入指定的文件

SKIP\_UNUSABLE\_INDEXES 跳过不可用索引的维护 (N)

FEEDBACK 每 x 行显示进度 (0)

TOID\_NOVALIDATE 跳过指定类型 ID 的验证

FILESIZE 每个转储文件的最大大小

STATISTICS 始终导入预计算的统计信息

RESUMABLE 在遇到有关空间的错误时挂起 (N)

RESUMABLE\_NAME 用来标识可恢复语句的文本字符串

RESUMABLE\_TIMEOUT RESUMABLE 的等待时间

COMPILE 编译过程, 程序包和函数 (Y)

STREAMS\_CONFIGURATION 导入流的一般元数据 (Y)

STREAMS\_INSTANTIATION 导入流实例化元数据 (N)

VOLSIZE 磁带的每个文件卷上的文件的字节数

#### 全量导入

导入数据库所有的数据对象和数据行

$imp userid=sms/smspass full=Y buffer=102400 file=/database/oracle\_backup/sms\_exp\_full\_20100201.dmp log=/database/oracle\_backup/sms\_imp\_full\_20100201.log

如果是不导入数据行可以使用控制参数rows=N

#### 用户模式导入

导入数据库用户sms中的所有对象和数据行

$imp userid=sms/smspass fromuser=sms touser=sms buffer=102400 file=/database/oracle\_backup/sms\_exp\_owner\_20100201.dmp log=/database/oracle\_backup/sms\_imp\_owner\_20100201.log

如果是不导出数据行可以使用控制参数rows=N

#### 表导入

导出指定表的数据

$imp userid=sms/smspass buffer=102400 tables=(sms\_mt\_report,sms\_batch\_log) file=/database/oracle\_backup/sms\_exp\_tabs\_20100201.dmp log=/database/oracle\_backup/sms\_imp\_tabs\_20100201.log

如果是不导出数据行可以使用控制参数rows=N

#### 适用环境

仅适用于exp同版本或低版本exp导出的dmp文件， 对于高于imp版本的exp导出的dmp文件，imp不能执行导入操作

运行imp的的客户端的NLS\_LANG，此环境变量的值需要和数据库的字符集相同；

### Impdp导入

#### 参数说明

USERID 必须是命令行中的第一个参数。

关键字 说明 (默认)

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

ATTACH 连接到现有作业, 例如 ATTACH [=作业名]。

CONTENT 指定要加载的数据, 其中有效关键字为:

(ALL), DATA\_ONLY 和 METADATA\_ONLY。

DIRECTORY 供转储文件, 日志文件和 sql 文件使用的目录对象。

DUMPFILE 要从 (expdat.dmp) 中导入的转储文件的列表,

例如 DUMPFILE=scott1.dmp, scott2.dmp, dmpdir:scott3.dmp。

ENCRYPTION\_PASSWORD 用于访问加密列数据的口令关键字。

此参数对网络导入作业无效。

ESTIMATE 计算作业估计值, 其中有效关键字为:

(BLOCKS) 和 STATISTICS。

EXCLUDE 排除特定的对象类型, 例如 EXCLUDE=TABLE:EMP。

FLASHBACK\_SCN 用于将会话快照设置回以前状态的 SCN。

FLASHBACK\_TIME 用于获取最接近指定时间的 SCN 的时间。

FULL 从源导入全部对象 (Y)。

HELP 显示帮助消息 (N)。

INCLUDE 包括特定的对象类型, 例如 INCLUDE=TABLE\_DATA。

JOB\_NAME 要创建的导入作业的名称。

LOGFILE 日志文件名 (import.log)。

NETWORK\_LINK 链接到源系统的远程数据库的名称。

NOLOGFILE 不写入日志文件。

PARALLEL 更改当前作业的活动 worker 的数目。

PARFILE 指定参数文件。

QUERY 用于导入表的子集的谓词子句。

REMAP\_DATAFILE 在所有 DDL 语句中重新定义数据文件引用。

REMAP\_SCHEMA 将一个方案中的对象加载到另一个方案。

REMAP\_TABLESPACE 将表空间对象重新映射到另一个表空间。

REUSE\_DATAFILES 如果表空间已存在, 则将其初始化 (N)。

SCHEMAS 要导入的方案的列表。

SKIP\_UNUSABLE\_INDEXES 跳过设置为无用索引状态的索引。

SQLFILE 将所有的 SQL DDL 写入指定的文件。

STATUS 在默认值 (0) 将显示可用时的新状态的情况下,

要监视的频率 (以秒计) 作业状态。

STREAMS\_CONFIGURATION 启用流元数据的加载

TABLE\_EXISTS\_ACTION 导入对象已存在时执行的操作。

有效关键字: (SKIP), APPEND, REPLACE 和 TRUNCATE。

TABLES 标识要导入的表的列表。

TABLESPACES 标识要导入的表空间的列表。

TRANSFORM 要应用于适用对象的元数据转换。

有效的转换关键字: SEGMENT\_ATTRIBUTES, STORAGE

OID 和 PCTSPACE。

TRANSPORT\_DATAFILES 按可传输模式导入的数据文件的列表。

TRANSPORT\_FULL\_CHECK 验证所有表的存储段 (N)。

TRANSPORT\_TABLESPACES 要从中加载元数据的表空间的列表。

仅在 NETWORK\_LINK 模式导入操作中有效。

VERSION 要导出的对象的版本, 其中有效关键字为:

(COMPATIBLE), LATEST 或任何有效的数据库版本。

仅对 NETWORK\_LINK 和 SQLFILE 有效。

下列命令在交互模式下有效。

注: 允许使用缩写

命令 说明 (默认)

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

CONTINUE\_CLIENT 返回到记录模式。如果处于空闲状态, 将重新启动作业。

EXIT\_CLIENT 退出客户机会话并使作业处于运行状态。

HELP 总结交互命令。

KILL\_JOB 分离和删除作业。

PARALLEL 更改当前作业的活动 worker 的数目。

PARALLEL=<worker 的数目>。

START\_JOB 启动/恢复当前作业。

START\_JOB=SKIP\_CURRENT 在开始作业之前将跳过

作业停止时执行的任意操作。

STATUS 在默认值 (0) 将显示可用时的新状态的情况下,

要监视的频率 (以秒计) 作业状态。

STATUS[=interval]

STOP\_JOB 顺序关闭执行的作业并退出客户机。

STOP\_JOB=IMMEDIATE 将立即关闭

数据泵作业。

#### 全量导入

导出数据库所有的对象和数据行

$ impdp sms/smspass DIRECTORY=expdir DUMPFILE=sms\_expdp\_full\_20100201.dmp full=y JOB\_NAME=impdp\_job1 content=all PARALLEL=2 logfile=sms\_impdp\_full\_20100201.log

如果仅导入数据库对象不导入数据，可以使用content= METADATA\_ONLY

#### 用户模式导入

导出数据库sms用户的对象和数据行

$ impdp sms/smspass DIRECTORY=expdir DUMPFILE=sms\_expdp\_owner\_20100201.dmp SCHEMAS=sms remap\_schema=sms:sms JOB\_NAME=impdp\_job1 content=all PARALLEL=2 logfile=sms\_impdp\_owner\_20100201.log

如果仅导入数据库对象不导入数据，可以使用content= METADATA\_ONLY

如果sms用户导出的文件，需要导入到ly用户下面，那么可以用下面的：

$ impdp sms/smspass DIRECTORY=expdir DUMPFILE=sms\_expdp\_owner\_20100201.dmp SCHEMAS=sms remap\_schema=sms:ly JOB\_NAME=impdp\_job1 content=all PARALLEL=2 logfile=sms\_expdp\_owner\_20100201.log

#### 表导入

导出数据库表和数

$ impdp sms/smspass DIRECTORY=expdir DUMPFILE=sms\_expdp\_tabs\_20100201.dmp TABLES=(sms\_mt\_report,sms\_batch\_log) JOB\_NAME=impdp\_job1 content=all PARALLEL=2 logfile=sms\_impdp\_tabs\_20100201.log

如果仅导入数据库对象不导入数据，可以使用content= METADATA\_ONLY

#### 适用环境

仅适用于expdp同版本或低版本expdp导出的dmp文件， 对于高于impdp版本的expdp导出的dmp文件，impdp不能执行导入操作！

运行impdp的的客户端的NLS\_LANG，此环境变量的值需要和数据库的字符集相同；

## Rman恢复

我们使用rman备份的最终目的就是保证数据库损坏，或者丢失数据时，能够进行完全恢复，或者不完全恢复！所以rman恢复也是非常重要的；需要经常验证你的rman备份集是不是有效的！

Restore 的文件类型

数据库，表空间，控制文件，spfile

### 恢复案例

不使用catalog目录，使用目标库的控制文件备份，当目标数据库的控制文件和数据文件都丢失了，那么

怎么使用之前的rman全备来恢复？

第一步：先恢复控制文件，

前提准确知道目标库的控制文件的路径和文件名字；和目标库的DBID=XXX

操作步骤：

1）

$rman nocatalog target /

RMAN>SET DBID=XXXX

RMAN>restore controlfile to '/oradata/db2/orcltest/control01.ctl' from '/oradata/db2/rmanback/ctl/ctl\_1.1111.00';

RMAN>restore controlfile to '/oradata/db2/orcltest/control02.ctl' from '/oradata/db2/rmanback/ctl/ctl\_1.1111.00';

RMAN>restore controlfile to '/oradata/db2/orcltest/control03.ctl' from '/oradata/db2/rmanback/ctl/ctl\_1.1111.00';

2)如果控制文件成功恢复的没有问题,那么接下来可以进行数据文件的恢复了;条件是启动实力到mount状态;

>shutdown immediate

>startup mount;

restore controlfile to '/database/oracle/oradata/orcl/control01.ctl' from '/database/rman/controlfile/c-1223900367-20090921-00';

RMAN>restore database;

rman>recover database;