

Control de datos en Oracle 11g



Proyecto realizado por Alejandro Valencia Valles para ASIR

Índice

Objetivos.....	7
Almacenado de datos.....	8
Backups Lógicos.....	8
¿Que es Data Pump Export?.....	8
Configuración.....	8
Archivelog.....	11
Que es al modo archive log.....	11
Activar archivelog.....	11
Backups físicos, RMAN.....	14
¿Que es RMAN?.....	14
Ventajas de realizar copias con RMAN:.....	14
Configuración.....	14
Tablespaces UNDO.....	18
Recuperación de datos.....	20
Data Pump Import impdp.....	20
¿Que es Data Pump Import?.....	20
Configuración.....	20
Flashback.....	24
¿Que es Flashback?.....	24
Veamos una prueba.....	24
Ahora con flashback query.....	25
RMAN.....	28
Recuperación de un tablespace.....	28
Recuperación de la base de datos completas incluidos los controlfiles en la base de datos que contiene el catalogo.....	32
Logminer.....	37
Webgrafia.....	42

Objetivos

A lo largo de este proyecto veremos distintos sistemas de control de datos.

Que significa control de datos veremos las herramientas que nos implementa el sistema gestor de bases de datos Oracle 11G R2 para mantener nuestros datos seguros.

Usaremos los siguientes métodos de guardado de datos:

- Realización de backups lógicos, con Data Pump Export (expdp).
- Activar modo archivelog.
- Realización de backups físicos con Rman.
- Tablespace UNDO.

Utilizaremos los siguientes métodos de recuperación:

- Data Pump Import (impdp).
- Flashback
- recovery datafile
- RMAN
- Logminer

Utilizaremos todos estos sistemas para ver como recuperar distintos elementos de nuestra base de datos.

Almacenado de datos.

Backups Lógicos.

¿Que es Data Pump Export?

Para los backups lógicos utilizaremos Data Dump Export (expdp), esta es una herramienta ofrecida por oracle que nos permite realizar copias de nuestros datos y esquemas de una base de datos de una forma sencilla y rápida.

Data Dump Export es una evolución del Export tradicional.

Las principales ventajas de Data Pump Export:

- Podemos exportar en paralelo, escribiendo en múltiples archivos en diferentes discos. Por ejemplo, especificando el parámetro PARALLEL=2 y dos directorios con los nombres de archivos destino, DUMPFILE=dir1:/file1.dp, dir2:/file2.dp.
- Posibilidad de realizar attach y detach del trabajo, monitoreando el trabajo remotamente.
- Más opciones para filtrado de metadatos (parámetros EXCLUDE e INCLUDE).
- Podemos estimar los requerimientos de espacio en disco con el parámetro ESTIMATE_ONLY, antes de ejecutar la tarea.
- Los datos pueden ser exportados desde una BD remota utilizando un DB link.
- Puede especificarse la versión de la BD y exportar solo los objetos compatibles con dicha versión.

Data Pump export crea archivos conocidos como dumps files los cuales tiene un formato propietario que solo Data Pump Import puede entender.

Configuración.

Data Pump es una tecnología servidor. Por esto, los ficheros dump y log son generados en directorios del servidor oracle. Data Pump requiere objetos directorios que para poder acceder al directorio donde se exportará o importarán los datos.

Como declaramos este directorio.

```
SQL> create directory expdp_dir as '/home/oracle/app/oracle/product/11.2.0/dbhome_2/export'  
2 ;
```

```
Directory created.
```

Una vez creado le daremos permisos de lectura y escritura a los usuarios que usen data pump export

y import:

```
SQL> grant read,write on directory expdp_dir to system;  
Grant succeeded.
```

Comenzemos con Data Pump Export.

Data pump puede trabajar desde la linea de comando o bien desde un fichero de configuración.

Data pump se lanza desde la linea de comandos.

```
expdp system/system DIRECTORY=expdp_dir DUMPFILE=expfull.dmp FULL=y  
LOGFILE=expfull.log
```

Esta es la forma mas completa y sencilla de exportar aunque data pump nos permite muchísimas más configuraciones.

La exportación por esquemas.

```
expdp system/system schemas=scott DIRECTORY=expdp_dir DUMPFILE=expscott.dmp  
LOGFILE=expscott.log
```

Exportación por tablas.

```
expdp system/system DIRECTORY=expdp_dir DUMPFILE=expscott-DEPT-EMP.dmp  
LOGFILE=expscott-DEPT-EMP.log tables=SCOTT.DEPT,SCOTT.EMP
```

Nota los Logs son muy amplios pondré el de la exportación de tablas del esquema scott ya que es el mas complejo y a la vez el log mas corto.

```
Export: Release 11.2.0.2.0 - Production on Sat Dec 16 12:39:53 2017  
  
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.  
  
Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options  
Starting "SCOTT"."SYS_EXPORT_TABLE_01": scott/***** DIRECTORY=expdp_dir  
DUMPFILE=expscott-DEPT-EMP.dmp LOGFILE=expscott-DEPT-EMP.log  
tables=SCOTT.DEPT,SCOTT.EMP  
Estimate in progress using BLOCKS method...  
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA  
Total estimation using BLOCKS method: 128 KB  
Processing object type TABLE_EXPORT/TABLE/TABLE  
Processing object type TABLE_EXPORT/TABLE/INDEX/INDEX  
Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/CONSTRAINT  
Processing object type TABLE_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS  
Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT  
Processing object type TABLE_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
```

```

.. exported "SCOTT"."DEPT"          5.929 KB   4 rows
.. exported "SCOTT"."EMP"          8.562 KB  14 rows
Master table "SCOTT"."SYS_EXPORT_TABLE_01" successfully loaded/unloaded
*****
Dump file set for SCOTT.SYS_EXPORT_TABLE_01 is:
/home/oracle/app/oracle/product/11.2.0/dbhome_2/export/expscott-DEPT-EMP.dmp
Job "SCOTT"."SYS_EXPORT_TABLE_01" successfully completed at 12:39:59

```

Veamos los parametros utilizados:

DIRECTORY	nombre del directory de nuestra base de datos que vamos a utilizar
DUMPFILE	nombre del fichero que usaremos para la importacexportacion
FULL	Parametro =y si queremos realizar una exportación completa
LOGFILE	fichero que usara de log
SCHEMAS	nombre de schemas que queremos exportar separados con coma “,”
TABLES	tablas que queremos exportar separados con coma “,”

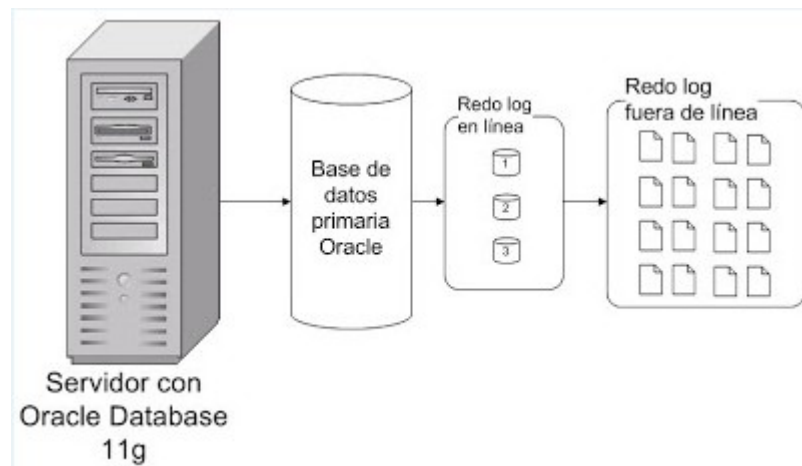
Hay más opciones, como por ejemplo:

CONTENT	tipo de contenido que queremos exportar datos o solo estructura
EXCLUDE	Elementos de la base de dato que no queremos exportar. Ejemplo: STATISTICS excluye las estadísticas, TABLE:=\”=\`tabla\`” excluye la tabla
FILESIZE	El tamaño máximo que deseamos que tenga el fichero dmp al llegar al tamaño máximo se divide en otro fichero que puede ocupar como máximo también este tamaño.
TRANSPORT_ TABLESPACE	Exporta un tablespace para importarlo en otra bbdd.
JOB_NAME	Le damos un nombre al proceso para poder consultarlo o modificarlo en caliente.

Archivelog.

Que es al modo archive log.

Es el modo en que nuestro sistema oracle va almacenando la información de todos los cambios que se van realizando en nuestra base de datos, oracle guarda todas las transacciones que se realizan en unos fichero denominados redo log online (redo log en línea). Cuando oracle no tiene el modo archive log activado cuando estos ficheros se llenan vuelve al primero y va reescribiendolos en cambio cuando ponemos nuestra base de datos en modo archivelog una vez terminado estos redo log online los copia en unos fichero denominado redo log offline (redo log fuera de línea).



De forma que vamos realizando una copia de todas las transacciones que se hacen.

Con este mecanismo tendremos varias ventajas:

- Se podrán realizar copias de seguridad físicas online, copias de los ficheros físicos sin detener la base de datos.
- Se podrá recuperar la base de datos en un momento específico del tiempo, se podrá recuperar la base de datos a una hora y día indicados.

Los dos problemas de este modo es que necesitaremos de espacio en disco y la limpieza cada cierto tiempo de ficheros y que puede ralentizar infinitamente la base de datos.

Activar archivelog.

En primer lugar comprobaremos que nuestra base de datos no está puesta en modo archive log.

Para ello nos conectamos a la base de datos como sys y ejecutamos el siguiente comando.

```
Archive log list
```

Vemos en la salida que no está en modo archive log.

```
SQL> archive log list
```

Database log mode	No Archive Mode
Automatic archival	Disabled
Archive destination	USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence	612
Current log sequence	614

Tambien podemos ver el estado del archive log con la siguiente consulta.

```
Select name, log_mode from v$database;
```

Para activar el modo archive log tendremos que tener la base de datos en modo mount.

Con lo que la apagaremos.

```
Shutdown immediate
```

Arrancamos la base de datos en modo mount.

```
Startup mount;
```

Activamos el modo archive log.

```
Alter database archive log;
```

Abrimos la base de datos.

```
Alter database open;
```

Veamos la salida completa.

```
SQL> Shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Startup mount;
ORACLE instance started.

Total System Global Area 456146944 bytes
```



```
Fixed Size          1344840 bytes
Variable Size       348129976 bytes
Database Buffers    100663296 bytes
Redo Buffers        6008832 bytes
```

Database mounted.

```
SQL> Alter database archivelog;
```

Database altered.

```
SQL> Alter database open;
```

Database altered.

```
SQL>
```

Realizamos la comprobación:

```
SQL> Select name, log_mode from v$database;
```

```
NAME LOG_MODE
```

```
-----
```

```
ORCL ARCHIVELOG
```

```
SQL> Archive log list
```

```
Database log mode      Archive Mode
```

```
Automatic archival    Enabled
```

```
Archive destTablespace created.ination      USE_DB_RECOVERY_FILE_DEST
```

```
Oldest online log sequence 612
```

```
Next log sequence to archive 614
```

```
Current log sequence     614
```

```
SQL>
```

Backups físicos, RMAN.

¿Que es RMAN?

El RMAN (Recovery Manager) es una herramienta de Oracle que sirve para realizar las tareas que están relacionadas con la seguridad de los datos como por ejemplo hacer copias de seguridad, restauraciones, recuperaciones y muchas otras cosas más.

Ventajas de realizar copias con RMAN:

Control sobre las copias: siempre el RMAN guarda información sobre qué copias de seguridad se han hecho y de qué se han hecho las copias de seguridad, donde están ubicadas esas copias y esa información es muy útil para luego hacer restauraciones. Es decir RMAN sabe dónde está ubicada cada copia de la base de datos, archivo dañado, etc.

Lo necesario para recuperar: RMAN posee toda la información necesaria para realizar la recuperación tanto de la base de datos como de archivos dañados, etc.

Restauraciones Directas: RMAN se encarga de ir a buscar la copia de seguridad que corresponde para ser recuperada y restaurarla en el sitio que le corresponde.

Políticas de Seguridad: nos permite ingresar la frecuencia con que tenemos que hacer el backup, cuándo se considera que un backup ya no es necesario guardarlo, etc.

Configuración

Vamos a crear Catalogo de RMAN (Recovery Catalog) ésta estrategia se basa en crear un repositorio de información, un tablespace con un usuario y hacer que allí se guarde toda la información para gestionar las copias de seguridad de una base de datos.

Para que nuestro RMAN funcione correctamente necesitamos que nuestra base de datos este en modo archiveolog

Primero tenemos que crear el usuario y tablespace.

```
CREATE TABLESPACE rman_ts DATAFILE
'/home/oracle/app/oracle/oradata/orcl/rman_ts.dbf' SIZE 300M;
```

Creamos el usuario.

```
CREATE USER rman2_user
IDENTIFIED BY rman2_user
DEFAULT TABLESPACE rman2_ts
QUOTA UNLIMITED ON rman2_ts;
```

Le damos privilegios

```
GRANT RECOVERY_CATALOG_OWNER TO rman2_user;
```

```
GRANT CONNECT, RESOURCE TO rman_user;
```

Ahora realizaremos la conexión de rman con el usuario de la base de datos.

```
[oracle@localhost ~]$ rman
Recovery Manager: Release 11.2.0.2.0 - Production on Sat Dec 16 11:06:18 2017
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
RMAN> CONNECT CATALOG rman_user
recovery catalog database Password:
connected to recovery catalog database
```

Creamos el catalogo. El catálogo es un listado de toda la información que necesita rman para realizar las recuperaciones de datos.

```
RMAN> CREATE CATALOG TABLESPACE rman_ts;
recovery catalog created
```

Una vez creado el catálogo tenemos que conectarnos con nuestro usuario rman_user.

```
[oracle@localhost ~]$ rman target=/ catalog rman_user
Recovery Manager: Release 11.2.0.2.0 - Production on Sat Dec 16 11:09:49 2017
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
connected to target database: ORCL (DBID=1229390655)
recovery catalog database Password:
connected to recovery catalog database
```

Una vez conectados a rman y a nuestro catálogo registramos nuestra base de datos.

```
RMAN> REGISTER DATABASE;
database registered in recovery catalog
starting full resync of recovery catalog
full resync complete
RMAN>
```

Ya tendríamos nuestra base de datos registrada y prerparada.

Ahora solo tendríamos que lanzar una copia de seguridad.

```
RMAN> backup database;
```

```
Starting backup at 17-DEC-17
```

```
allocated channel: ORA_DISK_1
```

```
channel ORA_DISK_1: SID=69 device type=DISK
```

```
channel ORA_DISK_1: starting full datafile backup set
```

```
channel ORA_DISK_1: specifying datafile(s) in backup set
```

```
input datafile file number=00002 name=/home/oracle/app/oracle/oradata/orcl/sysaux01.dbf
```

```
input datafile file number=00001 name=/home/oracle/app/oracle/oradata/orcl/system01.dbf
```

```
input datafile file number=00011 name=/home/oracle/app/oracle/oradata/orcl/rman_ts.dbf
```

```
input datafile file number=00004 name=/home/oracle/app/oracle/oradata/orcl/users01.dbf
```

```
input datafile file number=00003 name=/home/oracle/app/oracle/oradata/orcl/undotbs01.dbf
```

```
input datafile file number=00012 name=/home/oracle/app/oracle/oradata/orcl/undotbs-personal.dbf
```

```
input datafile file number=00005 name=/home/oracle/app/oracle/oradata/orcl/example01.dbf
```

```
input datafile file number=00010
```

```
name=/home/oracle/app/oracle/oradata/orcl/APEX_2614203650434107.dbf
```

```
input datafile file number=00006
```

```
name=/home/oracle/app/oracle/oradata/orcl/APEX_1930613455248703.dbf
```

```
input datafile file number=00007
```

```
name=/home/oracle/app/oracle/oradata/orcl/APEX_2041602962184952.dbf
```

```
input datafile file number=00008
```

```
name=/home/oracle/app/oracle/oradata/orcl/APEX_2610402357158758.dbf
```

```
input datafile file number=00009
```

```
name=/home/oracle/app/oracle/oradata/orcl/APEX_2611417663389985.dbf
```

```
channel ORA_DISK_1: starting piece 1 at 17-DEC-17
```

```
channel ORA_DISK_1: finished piece 1 at 17-DEC-17
```

```
piece
```

```
handle=/home/oracle/app/oracle/flash_recovery_area/ORCL/backupset/2017_12_17/o1_mf_nnndf
```

```
_TAG20171217T101733_f3ff5yol_.bkp tag=TAG20171217T101733 comment=NONE
```

```
channel ORA_DISK_1: backup set complete, elapsed time: 00:02:56
```

```
Finished backup at 17-DEC-17
```

```
Starting Control File and SPFILE Autobackup at 17-DEC-17
```

```
piece
```

```
handle=/home/oracle/app/oracle/flash_recovery_area/ORCL/autobackup/2017_12_17/o1_mf_s_96
```

```
2965233_f3ffclt6_.bkp comment=NONE
```

```
Finished Control File and SPFILE Autobackup at 17-DEC-17
```

Verificamos el estado de la copia.

```
RMAN> RESTORE VALIDATE DATABASE;
```

```
Starting restore at 17-DEC-17
```

```
using channel ORA_DISK_1
```

```
channel ORA_DISK_1: starting validation of datafile backup set
```

```
channel ORA_DISK_1: reading from backup piece
```

```
/home/oracle/app/oracle/flash_recovery_area/ORCL/backupset/2017_12_17/o1_mf_nnndf_TAG2
```

```
0171217T101733_f3ff5yol_.bkp
```

```
channel ORA_DISK_1: piece
```

```
handle=/home/oracle/app/oracle/flash_recovery_area/ORCL/backupset/2017_12_17/o1_mf_nnndf
```

```
_TAG20171217T101733_f3ff5yol_.bkp tag=TAG20171217T101733  
channel ORA_DISK_1: restored backup piece 1  
channel ORA_DISK_1: validation complete, elapsed time: 00:00:45  
Finished restore at 17-DEC-17
```

Vemos como pone que tenemos 1 pieza y nos la valida.

Podemos ver la configuración de RMAN.

```
RMAN> SHOW ALL;
```

```
RMAN configuration parameters for database with db_unique_name ORCL are:
```

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
```

```
CONFIGURE BACKUP OPTIMIZATION OFF; # default
```

```
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
```

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F';  
# default
```

```
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 ENCRYPTION FOR DATABASE OFF; # default
```

```
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
```

```
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT'
```

```
OPTIMIZE FOR LOAD TRUE ; # default
```

```
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
```

```
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
```

```
 '/home/oracle/app/oracle/product/11.2.0/dbhome_2/dbs/snapcf_orcl.f'; # default
```

```
RMAN>
```

Tablespaces UNDO.

Todas las bases de dato oracle deben tener un metodo para mantener la información que es usada para realizar un rollback. Tal información consiste en registros de acciones de transacciones, antes de realizarse su commit correspondiente. A esta colección de registros se les conoce como Undo. Los registros de Undo son usados para:

- Realizar rollback de transacciones cuando se usa la sentencia ROLLBACK.
- Recuperar la base de datos
- Proveer consistencia de lectura
- Hacer una recuperación de una corrupción lógica

Cuando se realiza una sentencia rollback, los registros en undo son usados para deshacer los cambios que fueron hechos en la base de datos por la transacción que aún no ha sido commiteado.

¿Que se almacena en el UNDO?

No se guarda los bloques de datos modificados (error habitual). Lo que se guarda es la información mínima necesaria para cumplir los 2 objetivos del UNDO:

1.- Asegurar la consistencia en lectura. Significa que si hay 100 usuarios trabajando concurrentemente y uno de ellos hace una consulta sobre la base de datos para obtener un informe, aunque la consulta tarde 10 minutos en dar una respuesta Oracle debe darle la información del momento en el que la ha lanzado el usuario. Mientras se ejecutaba esa consulta ha podido haber cambios, pero el usuario necesita saber la situación en el momento de lanzarla.

2.- Permitir la recuperación en caso de rollback. Un usuario puede lanzar varias sentencias que formen una sola transacción. Un cliente compra un billete de avión. Hay que actualizar la tabla de billetes vendidos, la del cliente con sus datos y una cuantas más. Si cualquiera de las actualizaciones falla toda la transacción hay que echarla atrás deshaciendo los cambios realizados.

¿Cual es esa información mínima? Pues depende:

Si el usuario inserta una fila nueva en una tabla al undo sólo irá el rowid del registro insertado, ya que para deshacer lo único que tiene que hacer es una delete de ese rowid.

¿Que se actualiza un campo de un registro? El undo guardará la información de ese campo antes de su modificación.

Como podemos crear un tablespace del tipo UNDO, pues muy sencillo con la siguiente orden.

```
create undo tablespace undotbs datafile '/home/oracle/app/oracle/oradata/orcl/undotbs-
```

```
personal.dbf' size 100M;
```

```
Tablespace created.
```

¿ Como gestiona ORACLE el tablespace UNDO ?

Oracle lo puede gestionar de dos maneras distintas:

1.- SMU (System Managed Undo)- (AUTO*) o lo que es lo mismo, modo automático de gestión, donde no se utilizan los segmentos de rollback externos, la información de UNDO se almacena en un tablespace especial dedicado exclusivamente a este objetivo.

2.- RBU (modo manual de gestión del UNDO) – (MANUAL*) . Se almacena de forma externa en segmentos de rollback. Este era el único método que existía en versiones anteriores de Oracle.

Ambos modos de gestión se establecen mediante el parámetro dinámico UNDO_MANAGEMENT que libera a los DBA's de la administración y monitoreo si lo configuras en modo AUTO.

Ejemplo para setear el parámetro en modo AUTO:

```
SQL> alter system set undo_management = AUTO;
```

¿ Cómo ver que valores de UNDO tenemos configurados en la BBDD ?

```
SQL> show parameters undo
```

NAME	TYPE	VALUE
undo_management	string	AUTO
undo_retention	integer	900
undo_tablespace	string	UNDOTBS1

¿ Que significan ?

undo_management: explicado en el punto anterior.

undo_retention: Parámetro dinámico que indica en segundos (900 por defecto) cuanto tiempo ha de permanecer, al menos, la información de UNDO disponible.

undo_tablespace: Parámetro dinámico que indica el tablespace de almacenamiento de “UNDO” a usar en el arranque.

Recuperación de datos.

Data Pump Import impdp

¿Que es Data Pump Import?

Data Pump Import al igual que el Data Pump Export es una evolución del Import.

Es una herramienta proporcionada por Oracle para la importacion de los ficheros de backups lógicos creados por Data Pump Export, que vimos anteriormente.

Las principales ventajas del Data Pump Import :

- Durante la tarea de impdp podemos cambiar el destino de los datafiles, esquemas y tablespaces (REMAP_DATAFILES, REMAP_SCHEMA, REMAP_TABLESPACE).
- Nos permite filtrar los datos durante la ejecución de impdp.
- Puede importarse desde una BD hacia otra sin escribir a un archivo de dump, especificando el parámetro NETWORK_LINK.
- El status de los trabajos pueden ser consultados directamente desde el data dictionary. Por ejemplo, dba_datapump_jobs, dba_datapump_sessions, etc.

Configuración

La igual que con Data Pump Export necesitamos crear un directorio donde se encuentren los ficheros.

```
SQL> create directory impdp_dir as '/home/oracle/app/oracle/product/11.2.0/dbhome_2/export';  
Directory created.  
  
SQL> grant read,write on directory impdp_dir to system;  
Grant succeeded.
```

Realicemos una importación por ejemplo borramos el usuario SCOTT.

NOTA: Antes de borrar asegúrese de tener un fichero dmp exportado con anterioridad que contenga los datos a recuperar.

```
SQL> drop user scott cascade;  
  
User dropped.
```

Salimos de sqlplus y realizamos una importacion de nuestro fichero full que exportamos anteriormente.


```
[oracle@localhost ~]$ impdp system/system schemas=SCOTT directory=impdp_dir
dumpfile=expfull.dmp logfile=impdpSCOTT.log
```

Import: Release 11.2.0.2.0 - Production on Sun Dec 17 01:47:05 2017

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production

With the Partitioning, OLAP, Data Mining and Real Application Testing options

Master table "SYSTEM"."SYS_IMPORT_SCHEMA_01" successfully loaded/unloaded

Starting "SYSTEM"."SYS_IMPORT_SCHEMA_01": system/***** schemas=SCOTT directory=impdp_dir
dumpfile=expfull.dmp logfile=impdpSCOTT.log

Processing object type DATABASE_EXPORT/SCHEMA/USER

Processing object type DATABASE_EXPORT/SCHEMA/GRANT/SYSTEM_GRANT

Processing object type DATABASE_EXPORT/SCHEMA/ROLE_GRANT

Processing object type DATABASE_EXPORT/SCHEMA/DEFAULT_ROLE

Processing object type DATABASE_EXPORT/SCHEMA/TYPE/TYPE_SPEC

Processing object type DATABASE_EXPORT/SCHEMA/PROCACT_SCHEMA

Processing object type DATABASE_EXPORT/SCHEMA/XMLSCHEMA/XMLSCHEMA

Processing object type DATABASE_EXPORT/SCHEMA/TABLE/TABLE

Processing object type DATABASE_EXPORT/SCHEMA/TABLE/TABLE_DATA

.. imported "SCOTT"."DATA_STAGING_REPOS"	37.02 MB	28 rows
.. imported "SCOTT"."DATA_STAGING_EVO"	10.99 MB	10000 rows
.. imported "SCOTT"."DATA_STAGING_OTN"	11.06 MB	10000 rows
.. imported "SCOTT"."DATA_STAGING_PTN"	11.02 MB	10000 rows
.. imported "SCOTT"."DATA_STAGING_XQY"	10.96 MB	10000 rows
.. imported "SCOTT"."STAT_TABLE"	36.78 KB	178 rows
.. imported "SCOTT"."DEPT"	5.929 KB	4 rows
.. imported "SCOTT"."EMP"	8.562 KB	14 rows
.. imported "SCOTT"."SALGRADE"	5.859 KB	5 rows
.. imported "SCOTT"."BONUS"	0 KB	0 rows

Processing object type DATABASE_EXPORT/SCHEMA/TABLE/INDEX/INDEX

Processing object type DATABASE_EXPORT/SCHEMA/TABLE/CONSTRAINT/CONSTRAINT

Processing object type DATABASE_EXPORT/SCHEMA/TABLE/INDEX/STATISTICS/INDEX_STATISTICS

Processing object type DATABASE_EXPORT/SCHEMA/TABLE/CONSTRAINT/REF_CONSTRAINT

Processing object type DATABASE_EXPORT/SCHEMA/TABLE/STATISTICS/TABLE_STATISTICS

Job "SYSTEM"."SYS_IMPORT_SCHEMA_01" successfully completed at 01:48:13

Esta es la salida.

Impdp como podemos ver comparte muchos parámetros con nuestro expdp aunque modifique un poco su significado al ser una importación en vez de un exportación se comprende perfectamente la diferencia.

Veamos los parametros:

DIRECTORY	nombre del directory de nuestra base de datos que vamos a utilizar
DUMPFIL	nombre del fichero que usaremos para la importación
FULL	Parametro =y si queremos realizar una importación completa
LOGFILE	Fichero que usara de log
SCHEMAS	nombre de schemas que queremos importar separados con coma “,”
TABLES	tablas que queremos importar separados con coma “,”

CONTENT	tipo de contenido que queremos importar datos o solo estructura
SQLFILE	En vez de realizar la importación en la bbdd genera un script sql con la estructura para crearlo
INCLUDE	Elige lo que quiere incluir en la importación.
TRANSPORT_ TABLESPACE	Importar un tablespace para importarlo en otra bbdd.
JOB_NAME	Le damos un nombre al proceso para poder consultarlo o modificarlo en caliente.

Veamos otro ejemplo.

Queremos exportar el codigo sql de un objeto como es la tabla emp del schema de scott.

```
impdp system SQLFILE=tabla-EMP.sql TABLES=SCOTT.EMP DIRECTORY=IMPDP_DIR
DUMPFIL=expscott.dmp LOGFILE=impdp_SCOTT-EMP_sqlfile.log
JOB_NAME=imp_testuser
```

Veamos el codigo sql.

```
[oracle@localhost export]$ cat tabla-EMP.sql
-- CONNECT SYSTEM
ALTER SESSION SET EVENTS '10150 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '10904 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '25475 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '10407 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '10851 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '22830 TRACE NAME CONTEXT FOREVER, LEVEL 192';
-- new object type path: SCHEMA_EXPORT/TABLE/TABLE
CREATE TABLE "SCOTT"."EMP"
(
  "EMPNO" NUMBER(4,0),
  "ENAME" VARCHAR2(10 BYTE),
  "JOB" VARCHAR2(9 BYTE),
  "MGR" NUMBER(4,0),
  "HIREDATE" DATE,
  "SAL" NUMBER(7,2),
  "COMM" NUMBER(7,2),
  "DEPTNO" NUMBER(2,0)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
-- new object type path: SCHEMA_EXPORT/TABLE/INDEX/INDEX
-- CONNECT SCOTT
```

```

CREATE UNIQUE INDEX "SCOTT"."PK_EMP" ON "SCOTT"."EMP" ("EMPNO")
PCTFREE 10 INITRANS 2 MAXTRANS 255
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" PARALLEL 1 ;

ALTER INDEX "SCOTT"."PK_EMP" NOPARALLEL;
-- new object type path: SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
-- CONNECT SYSTEM
ALTER TABLE "SCOTT"."EMP" ADD CONSTRAINT "PK_EMP" PRIMARY KEY ("EMPNO")
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ENABLE;
-- new object type path: SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
DECLARE l_n VARCHAR2(60);
l_o VARCHAR2(60);
nv VARCHAR2(1);
c DBMS_METADATA_T_VAR_COLL;
df varchar2(21) := 'YYYY-MM-DD:HH24:MI:SS';
stmt varchar2(300) := ' INSERT INTO "SYS"."IMPDP_STATS" (type,version,flags,c1,c2,c3,c5,n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,d1,c1) VALUES
('T',6,,1,,2,,3,,4,,5,,6,,7,,8,,9,,10,,11,,12,,13, NULL,,14,,15, NULL,,16,,17);
BEGIN
DELETE FROM "SYS"."IMPDP_STATS";
i_n := 'PK_EMP';
i_o := 'SCOTT';
EXECUTE IMMEDIATE stmt USING 2,l_n,nv,nv,l_o,14,1,1,1,0,14,nv,nv,TO_DATE('2011-10-21 10:06:26',df),nv;

DBMS_STATS.IMPORT_INDEX_STATS('' || i_o || ''','' || i_n || ''','NULL',"IMPDP_STATS",NULL,"SYS");
DELETE FROM "SYS"."IMPDP_STATS";
END;
/

-- new object type path: SCHEMA_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT
ALTER TABLE "SCOTT"."EMP" ADD CONSTRAINT "FK_DEPTNO" FOREIGN KEY ("DEPTNO")
REFERENCES "SCOTT"."DEPT" ("DEPTNO") ENABLE;
-- new object type path: SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
DECLARE
c varchar2(60);
nv varchar2(1);
df varchar2(21) := 'YYYY-MM-DD:HH24:MI:SS';
s varchar2(60) := 'SCOTT';
t varchar2(60) := 'EMP';
p varchar2(1);
sp varchar2(1);
stmt varchar2(300) := 'INSERT INTO "SYS"."IMPDP_STATS" (type,version,c1,c2,c3,c4,c5,n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,d1,r1,r2,ch1,flags,c1) VALUES
(1,6,,2,,3,,4,,5,,6,,7,,8,,9,,10,,11,,12,,13,,14,,15,,16,,17,,18,,19,,20,,21,,22,,23);
BEGIN
DELETE FROM "SYS"."IMPDP_STATS";
INSERT INTO "SYS"."IMPDP_STATS" (type,version,flags,c1,c2,c3,c5,n1,n2,n3,n4,n9,n10,n11,n12,d1) VALUES ('T',6,2,t,p,sp,s,
14,5,38,14,0,NULL,NULL,NULL,
TO_DATE('2011-10-21 10:06:26',df));
c := 'EMPNO';
EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
14,0714285714285714,14,14,0,7369,7934,4,0,nv,nv,
TO_DATE('2011-10-21 10:06:26',df),'C24A46','C25023',nv,2,nv;
c := 'ENAME';
EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
14,0714285714285714,14,14,0,3,38883673419062E+35,4.53054701071074E+35,6,0,nv,nv,
TO_DATE('2011-10-21 10:06:26',df),4144414D53',57415244',nv,2,nv;
c := 'JOB';
EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
5,,2,5,14,0,3,39086497213261E+35,4,32285038678150E+35,8,0,nv,nv,
TO_DATE('2011-10-21 10:06:26',df),414E414C595354',53414C45534D414E',nv,2,nv;
c := 'MGR';
EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
6,,1666666666666667,6,13,1,7566,7902,4,0,nv,nv,
TO_DATE('2011-10-21 10:06:26',df),'C24C43','C25003',nv,2,nv;
END;
/

DECLARE
c varchar2(60);
nv varchar2(1);
df varchar2(21) := 'YYYY-MM-DD:HH24:MI:SS';
s varchar2(60) := 'SCOTT';
t varchar2(60) := 'EMP';
p varchar2(1);
sp varchar2(1);
stmt varchar2(300) := 'INSERT INTO "SYS"."IMPDP_STATS" (type,version,c1,c2,c3,c4,c5,n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,d1,r1,r2,ch1,flags,c1) VALUES
(1,6,,2,,3,,4,,5,,6,,7,,8,,9,,10,,11,,12,,13,,14,,15,,16,,17,,18,,19,,20,,21,,22,,23);
BEGIN
NULL;
c := 'HIREDATE';
EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
13,0769230769230769,13,14,0,2444591,2446939,8,0,nv,nv,
TO_DATE('2011-10-21 10:06:26',df),'77B40C11010101','77BB0517010101',nv,2,nv;
c := 'SAL';
EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
12,,0833333333333333,12,14,0,800,5000,4,0,nv,nv,
TO_DATE('2011-10-21 10:06:26',df),'C209','C233',nv,2,nv;
c := 'COMM';
EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
4,,25,4,4,10,0,1400,2,0,nv,nv,
TO_DATE('2011-10-21 10:06:26',df),'80','C20F',nv,2,nv;
c := 'DEPTNO';
EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
3,,3333333333333333,3,14,0,10,30,3,0,nv,nv,
TO_DATE('2011-10-21 10:06:26',df),'C10B','C11F',nv,2,nv;

DBMS_STATS.IMPORT_TABLE_STATS('SCOTT','EMP',NULL,"IMPDP_STATS",NULL,NULL,"SYS");
DELETE FROM "SYS"."IMPDP_STATS";
END;
/

```

Flashback

¿Que es Flashback?

Desde que existen las base de datos Oracle, existen también los segmentos de rollback (que luego evolucionaron para convertirse en segmentos de UNDO).

Flashback es una herramienta que nos permite consultar estos segmentos de forma que durante un tiempo podríamos acceder a la información de un estado anterior de la base de datos.

Desventaja de este sistema, tenemos un numero de acciones y de tiempo en el que poder actuar antes poder revertir los cambios.

Veamos una prueba.

Tenemos nuestra tabla de pruebas en nuestra base de datos en el esquema de scott.

```
SQL> select * from prueba;
```

```
  CAMPO1
```

```
-----  
      1  
      2  
      3
```

Por error alguien la borra.

```
SQL> drop table prueba
```

```
  2 ;
```

```
Table dropped.
```

```
SQL> select * from prueba
```

```
  2 ;
```

```
select * from prueba
```

```
      *
```

```
ERROR at line 1:
```

```
ORA-00942: table or view does not exist
```

Bien para poder realizar la recuperación lo primero seria mirar si aún contiene nuestro UNDO los datos

```
select OBJECT_NAME, ORIGINAL_NAME, OPERATION, TYPE, TS_NAME, CREATETIME,  
DROPTIME from user_recyclebin;
```

```
OBJECT_NAME          ORIGINAL_NAME          OPERATION
```

```
-----  
TYPE                TS_NAME                CREATETIME
```

```
-----  
DROPTIME  
-----
```

```
BIN$YikUAYKrCTngUAB/AQAwCw==$0 PRUEBA          DROP  
TABLE          USERS          2017-12-17:04:02:18  
2017-12-17:04:07:31
```

```
SQL> flashback table prueba to before drop;
```

```
Flashback complete.
```

Comprobamos que volvimos al punto anterior y que la tabla existe de nuevo con los datos que contenía.

```
SQL> select OBJECT_NAME ,ORIGINAL_NAME, OPERATION, TYPE, TS_NAME,  
CREATETIME, DROPTIME from user_recyclebin;
```

```
no rows selected
```

```
SQL> select * from prueba;
```

```
  CAMPO1  
-----
```

```
  1
```

```
  2
```

```
  3
```

Ahora con flashback query.

En este caso vamos a ver que alguien borra un dato de nuestra tabla Prueba y estos datos son absolutamente necesarios para nuestra base de datos.

```
SQL> select * from prueba;
```

```
  CAMPO1  
-----
```

```
  1
```

2

3

```
SQL> delete from prueba where campo1=2;
```

1 row deleted.

```
SQL> select * from prueba;
```

CAMPO1

1

3

En esta caso usare una utilidad llamada flashback query.

Realizamos una consulta al pasado.

```
SQL> SELECT * FROM prueba AS OF TIMESTAMP TO_TIMESTAMP('17-12-2017 04:30:00',  
'DD-MM-YYYY HH:MI:SS');
```

CAMPO1

1

2

3

La Recuperación de los datos se poría hacer de varias maneras, un create table as select, un insert into la tabla original trás vaciarla o bien guardar los resultados en una 2.tabla e ir comparando el estado anterior con el actual para cambiar solamente los registros maliciosos para evitar que la tabla entera se vaya a una hora para atrás y se pierda datos insertados después.

Nosotros vamos a optar en este caso en generar una consulta que nos devuelva los datos borrados desde aquel momento a este y generara un insert para que podamos reponer nuestra tabla.

```
SQL> select 'insert into prueba values ('||CAMPO1||');' FROM prueba AS OF TIMESTAMP  
TO_TIMESTAMP('17-12-2017 04:30:00', 'DD-MM-YYYY HH:MI:SS') where campo1 not in  
(select campo1 from prueba);
```

```
'INSERTINTOPRUEBAVALUES('||CAMPO1||');
```

```
-----  
insert into prueba values (2);
```

```
SQL>
```

Ahora solo tendríamos que ejecutar la salida resultante.

RMAN.

Recuperación de un tablespace

Veamos como recuperar un tablespace con RMAN y la herramienta Recovery Datafile.

Primero crearemos un tablespace y un usuario que lo vallan a utilizar.

```
SQL> create tablespace prueba_ts datafile '/home/oracle/app/oracle/oradata/orcl/prueba_ts.dbf'  
size 100M;
```

Tablespace created.

```
SQL> CREATE USER prueba  
IDENTIFIED BY prueba  
DEFAULT TABLESPACE prueba_ts  
QUOTA UNLIMITED ON prueba_ts; 2 3 4
```

User created.

```
SQL> GRANT CONNECT, RESOURCE TO prueba;
```

Grant succeeded.

Vemos que contiene un objeto.

```
SQL> select * from tabla2;
```

CAMPO1

1

2

3

1

2

3

6 rows selected.

Ahora provocaremos una perdida del tablespace.

```
[oracle@localhost ~]$ rm /home/oracle/app/oracle/oradata/orcl/prueba_ts.dbf
```

Intentaremos conectarnos al usuario prueba y realizar un truncate.

```
SQL> truncate table tabla2;
truncate table tabla2
      *
ERROR at line 1:
ORA-00603: ORACLE server session terminated by fatal error
ORA-01116: error in opening database file 13
ORA-01110: data file 13: '/home/oracle/app/oracle/oradata/orcl/prueba_ts.dbf'
ORA-27041: unable to open file
Linux Error: 2: No such file or directory
Additional information: 3
ORA-01116: error in opening database file 13
ORA-01110: data file 13: '/home/oracle/app/oracle/oradata/orcl/prueba_ts.dbf'
ORA-27041: unable to open file
Linux Error: 2: No such file or directory
Additional information: 3
Process ID: 24340
Session ID: 67 Serial number: 775
```

Nos muestra un error de que no encuentra nuestro datafile.

Como lo recuperamos.

Primero desctivamos el tablespace.

```
rman target=/ catalog rman_user
RMAN> sql "alter tablespace prueba_ts offline immediate";
```

```
sql statement: alter tablespace prueba_ts offline immediate
```

Restauramos el fichero.

```
RMAN> restore tablespace prueba_ts;
```

```
Starting restore at 17-DEC-17
```

```
using channel ORA_DISK_1
```

```
channel ORA_DISK_1: starting datafile backup set restore
```

```
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
```

```
channel ORA_DISK_1: restoring datafile 00013 to  
/home/oracle/app/oracle/oradata/orcl/prueba_ts.dbf
```

```
channel ORA_DISK_1: reading from backup piece  
/home/oracle/app/oracle/flash_recovery_area/ORCL/backupset/2017_12_17/o1_mf_nnndf_TAG2  
0171217T111531_f3fjlmsd_.bkp
```

```
channel ORA_DISK_1: piece  
handle=/home/oracle/app/oracle/flash_recovery_area/ORCL/backupset/2017_12_17/o1_mf_nnndf  
_TAG20171217T111531_f3fjlmsd_.bkp tag=TAG20171217T111531
```

```
channel ORA_DISK_1: restored backup piece 1
```

```
channel ORA_DISK_1: restore complete, elapsed time: 00:00:07
```

```
Finished restore at 17-DEC-17
```

Y lo recuperamos.

```
RMAN> recover tablespace prueba_ts;
```

```
Starting recover at 17-DEC-17
```

```
using channel ORA_DISK_1
```

```
starting media recovery
```

```
media recovery complete, elapsed time: 00:00:00
```

```
Finished recover at 17-DEC-17
```

Por ultimo volvemos a activar el tablespace.

```
RMAN> sql 'alter tablespace prueba_ts online';
```

```
sql statement: alter tablespace prueba_ts online
```

Verificamos que ahora podemos hacer el truncate.

```
SQL> conn prueba/prueba
```

```
Connected.
```

```
SQL> truncate table tabla2;
```

```
Table truncated.
```

Recuperación de la base de datos completas incluidos los controlfiles en la base de datos que contiene el catalogo.

Veamos el siguiente caso por problemas de discos o por un error humano nuestra base de datos con los controlfile incluidos. Además esta base de datos es el repositorio rman.

Simularemos la pérdida de los controlfile borrándolos. Y a continuación nos pondremos con la restauración completa de la base de datos.

```
[oracle@localhost ~]$ rm /home/oracle/app/oracle/oradata/orcl/control01.ctl  
[oracle@localhost ~]$ rm /home/oracle/app/oracle/flash_recovery_area/orcl/control02.ctl
```

Vemos que nuestra base de datos a perdido los controlfiles.

```
SQL> SHUTDOWN IMMEDIATE  
ORA-00210: cannot open the specified control file  
ORA-00202: control file: '/home/oracle/app/oracle/oradata/orcl/control01.ctl'  
ORA-27041: unable to open file  
Linux Error: 2: No such file or directory  
Additional information: 3
```

Forzamos el apagado de nuestra base de datos.

```
SQL> SHUTDOWN ABORT  
ORACLE instance shut down.
```

La ponemos en modo no montada.

```
SQL> startup nomount  
ORACLE instance started.  
  
Total System Global Area 456146944 bytes  
Fixed Size 1344840 bytes  
Variable Size 373295800 bytes  
Database Buffers 75497472 bytes  
Redo Buffers 6008832 bytes
```

Ahora iriamos a RMAN.

```
[oracle@localhost dbs]$ rman target=/ catalog rman_user
```

```
Recovery Manager: Release 11.2.0.2.0 - Production on Mon Dec 18 18:57:16 2017
```

```
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
```

```
connected to target database: ORCL (not mounted)
```

```
recovery catalog database Password:
```

```
RMAN-00571: =====
```

```
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS
```

```
=====
```

```
RMAN-00571: =====
```

```
RMAN-00554: initialization of internal recovery manager package failed
```

```
RMAN-04004: error from recovery catalog database: ORA-01033: ORACLE initialization or shutdown in progress
```

¿Que ha pasado?

Pues que los controlfile que perdimos eran de la maquina que tenia nuestro catálogo, con lo cual nuestro catalogo no funciona.

No pasa nada aun tenemos las piezas que generamos al realizar los backups.

Localizamos nuestra pieza que contiene los controlfile.

```
[oracle@localhost ~]$ ls -al
```

```
/home/oracle/app/oracle/flash_recovery_area/ORCL/autobackup/2017_12_17/
```

```
total 28856
```

```
drwxrwx--- 2 oracle oracle 4096 Dec 17 20:40 .
```

```
drwxrwx--- 5 oracle oracle 4096 Dec 18 18:40 ..
```

```
-rw-rw---- 1 oracle oracle 9830400 Dec 17 19:55 o1_mf_s_962967313_f3fhdkdx_.bkp
```

```
-rw-rw---- 1 oracle oracle 9830400 Dec 17 20:18 o1_mf_s_962968698_f3fjqt0_.bkp
```

```
-rw-rw---- 1 oracle oracle 9830400 Dec 17 20:40 o1_mf_s_962970018_f3fl12v1_.bkp
```

Y lanzamos rman sin catalogo.

```
[oracle@localhost dbs]$ rman target=/
```

Recovery Manager: Release 11.2.0.2.0 - Production on Mon Dec 18 19:03:21 2017

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

connected to target database: ORCL (not mounted)

RMAN>

Ahora le decimos que restaure los controlfile de nuestra pieza del de la copia de seguridad.

```
RMAN> restore controlfile from
'/home/oracle/app/oracle/flash_recovery_area/ORCL/autobackup/2017_12_17/o1_mf_s_96297001
8_f3fl12v1_.bkp';
```

Starting restore at 18-DEC-17

using target database control file instead of recovery catalog

allocated channel: ORA_DISK_1

channel ORA_DISK_1: SID=19 device type=DISK

channel ORA_DISK_1: restoring control file

channel ORA_DISK_1: restore complete, elapsed time: 00:00:03

output file name=/home/oracle/app/oracle/oradata/orcl/control01.ctl

output file name=/home/oracle/app/oracle/flash_recovery_area/orcl/control02.ctl

Finished restore at 18-DEC-17

Ahora ya podriamos montar nuestra base de datos gracias a que tenemos nuestros controlfile.

```
SQL> alter database mount;
```

Database altered.

```
SQL>
```

Ahora habría que restaurar los ficheros de la pieza.

```
RMAN> restore database;
```

Starting restore at 18-DEC-17

Starting implicit crosscheck backup at 18-DEC-17

allocated channel: ORA_DISK_1

channel ORA_DISK_1: SID=20 device type=DISK

Crosschecked 3 objects

Finished implicit crosscheck backup at 18-DEC-17

Starting implicit crosscheck copy at 18-DEC-17

using channel ORA_DISK_1

Finished implicit crosscheck copy at 18-DEC-17

searching for all files in the recovery area

cataloging files...

cataloging done

List of Cataloged Files

=====

File Name: /home/oracle/app/oracle/flash_recovery_area/ORCL/archivelog/2017_12_18/o1_mf_1_624_f3hzd4wv_.arc

File Name: /home/oracle/app/oracle/flash_recovery_area/ORCL/archivelog/2017_12_18/o1_mf_1_625_f3hzd4mg_.arc

File Name: /home/oracle/app/oracle/flash_recovery_area/ORCL/archivelog/2017_12_18/o1_mf_1_624_f3ht90qh_.arc

File Name: /home/oracle/app/oracle/flash_recovery_area/ORCL/archivelog/2017_12_18/o1_mf_1_626_f3hzd4qx_.arc

File Name: /home/oracle/app/oracle/flash_recovery_area/ORCL/archivelog/2017_12_18/o1_mf_1_625_f3hwftx2_.arc

File Name: /home/oracle/app/oracle/flash_recovery_area/ORCL/autobackup/2017_12_18/o1_mf_s_963081629_f3hzdgoo_.bkp

File Name: /home/oracle/app/oracle/flash_recovery_area/ORCL/autobackup/2017_12_17/o1_mf_s_962970018_f3fl12v1_.bkp

using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile backup set restore

channel ORA_DISK_1: specifying datafile(s) to restore from backup set

channel ORA_DISK_1: restoring datafile 00001 to /home/oracle/app/oracle/oradata/orcl/system01.dbf

channel ORA_DISK_1: restoring datafile 00002 to /home/oracle/app/oracle/oradata/orcl/sysaux01.dbf

channel ORA_DISK_1: restoring datafile 00003 to /home/oracle/app/oracle/oradata/orcl/undotbs01.dbf

channel ORA_DISK_1: restoring datafile 00004 to /home/oracle/app/oracle/oradata/orcl/users01.dbf

channel ORA_DISK_1: restoring datafile 00005 to /home/oracle/app/oracle/oradata/orcl/example01.dbf

channel ORA_DISK_1: restoring datafile 00006 to /home/oracle/app/oracle/oradata/orcl/APEX_1930613455248703.dbf

channel ORA_DISK_1: restoring datafile 00007 to /home/oracle/app/oracle/oradata/orcl/APEX_2041602962184952.dbf

channel ORA_DISK_1: restoring datafile 00008 to /home/oracle/app/oracle/oradata/orcl/APEX_2610402357158758.dbf

channel ORA_DISK_1: restoring datafile 00009 to /home/oracle/app/oracle/oradata/orcl/APEX_2611417663389985.dbf

channel ORA_DISK_1: restoring datafile 00010 to /home/oracle/app/oracle/oradata/orcl/APEX_2614203650434107.dbf

channel ORA_DISK_1: restoring datafile 00011 to /home/oracle/app/oracle/oradata/orcl/rman_ts.dbf

channel ORA_DISK_1: restoring datafile 00012 to /home/oracle/app/oracle/oradata/orcl/undotbs-personal.dbf

channel ORA_DISK_1: restoring datafile 00013 to /home/oracle/app/oracle/oradata/orcl/prueba_ts.dbf

channel ORA_DISK_1: reading from backup piece

/home/oracle/app/oracle/flash_recovery_area/ORCL/backupset/2017_12_17/o1_mf_nnndf_TAG20171217T111531_f3fjlm sd_.bkp

channel ORA_DISK_1: piece

handle=/home/oracle/app/oracle/flash_recovery_area/ORCL/backupset/2017_12_17/o1_mf_nnndf_TAG20171217T111531_f3fjlm sd_.bkp

tag=TAG20171217T111531

channel ORA_DISK_1: restored backup piece 1

```
channel ORA_DISK_1: restore complete, elapsed time: 00:03:17
```

```
Finished restore at 18-DEC-17
```

La salida es muy amplia, reduzco la letra y quito los espacios.

Y realizamos la aplicación de cambios.

```
RMAN> recover database;
```

```
Starting recover at 18-DEC-17
```

```
using channel ORA_DISK_1
```

```
starting media recovery
```

```
archived log for thread 1 with sequence 624 is already on disk as file  
/home/oracle/app/oracle/flash_recovery_area/ORCL/archivelog/2017_12_18/o1_mf_1_624_f3hzd4wv_arc
```

```
archived log for thread 1 with sequence 625 is already on disk as file  
/home/oracle/app/oracle/flash_recovery_area/ORCL/archivelog/2017_12_18/o1_mf_1_625_f3hzd4mg_arc
```

```
archived log for thread 1 with sequence 626 is already on disk as file  
/home/oracle/app/oracle/flash_recovery_area/ORCL/archivelog/2017_12_18/o1_mf_1_626_f3hzd4qx_arc
```

```
archived log file name=/home/oracle/app/oracle/flash_recovery_area/ORCL/archivelog/2017_12_18/o1_mf_1_624_f3hzd4wv_arc thread=1  
sequence=624
```

```
archived log file name=/home/oracle/app/oracle/flash_recovery_area/ORCL/archivelog/2017_12_18/o1_mf_1_625_f3hzd4mg_arc thread=1  
sequence=625
```

```
archived log file name=/home/oracle/app/oracle/flash_recovery_area/ORCL/archivelog/2017_12_18/o1_mf_1_626_f3hzd4qx_arc thread=1  
sequence=626
```

```
media recovery complete, elapsed time: 00:00:03
```

```
Finished recover at 18-DEC-17
```

Comprobamos que nuestra base de datos vuelve a ser completamente funcional.

La abrimos y nos conectamos con un usuario de la base de datos.

```
SQL> alter database open resetlogs;
```

```
Database altered.
```

```
SQL> conn scott/tigger
```

```
Connected.
```

```
SQL> show user
```

```
USER is "SCOTT"
```


Logminer

Logminer es una herramienta que nos ofrece oracle con la cual reconstruir datos queramos recuperar y que hubieran sido validadas con un commit.

Esta herramienta utiliza ficheros Redo Log y archivelog añadiendoles metadatos para una lectura sencilla. Con lo cual nuestra base de datos debe estar en modo archivelog.

Vamos a ver que se han insertado datos que no deberían de estar con lo cual tenemos que revisar la introducción de datos y deshacerla.

Creamos el directorio en el que se encontraran los fichero de logminer.

```
SQL> alter system set utl_file_dir='/logminer' scope=spfile;
System altered.
```

Reiniciamos la base de datos.

```
shutdown immediate
startup
```

Confirmamos que nuestro utl_file_dir es el correcto.

```
SQL> show parameter utl_file_dir
```

NAME	TYPE	VALUE
-----	-----	-----
utl_file_dir	string	/logminer

Le indicamos como queremos que se llame el fichero a construir

```
SQL> exec DBMS_LOGMNR_D.BUILD( DICTIONARY_FILENAME =>'dictionary.ora',
DICTIONARY_LOCATION => '/logminer');

PL/SQL procedure successfully completed.
```

Realizamos una comprobación para ver según el momento de escritura los redo en que grupo debería de estar nuestros datos.

Ponemos el formato de fecha y hora.

```
ALTER SESSION SET NLS_DATE_FORMAT='DD/MM/YYYY HH24:MI:SS';
```

Ahora vemos la escritura de los grupos.

```
SQL> select GROUP# , FIRST_TIME, NEXT_TIME from v$log;
```

GROUP#	FIRST_TIME	NEXT_TIME
1	19/12/2017 01:54:17	19/12/2017 02:06:47
2	19/12/2017 02:06:47	19/12/2017 03:16:34
3	19/12/2017 03:16:34	

Vemos la hora de sistema:

```
SQL> SELECT SYSDATE FROM DUAL;
```

SYSDATE
19/12/2017 03:20:33

Con lo cual nuestros datos deberían de estar en el grupo 2

Comprobemos nuestros Redo log. El uso de los fichero redo o los archive log dependera del tiempo y las transacciones que se realicen si la transacción en mente se realizo antes de que se llenara nuestro archivo redo estará en estos ficheros si es posterior los datos habrán sido trasladados a un archivelog.

```
SQL> select GROUP#, MEMBER from v$logfile;
```

GROUP#	MEMBER
3	/home/oracle/app/oracle/oradata/orcl/redo03.log
2	

```
/home/oracle/app/oracle/oradata/orcl/redo02.log
```

1

```
/home/oracle/app/oracle/oradata/orcl/redo01.log
```

Creamos los metadatos del Redo Log (en nuestro caso) que deseemos. Como tenemos tres los cargaremos todos.

```
SQL> exec
DBMS_LOGMNR.ADD_LOGFILE('/home/oracle/app/oracle/oradata/orcl/redo01.log');

PL/SQL procedure successfully completed.

SQL> exec
DBMS_LOGMNR.ADD_LOGFILE('/home/oracle/app/oracle/oradata/orcl/redo02.log');

PL/SQL procedure successfully completed.

SQL> exec DBMS_LOGMNR.ADD_LOGFILE('/home/oracle/app/oracle/oradata/orcl/redo03.log');

PL/SQL procedure successfully completed.
```

Iniciamos la session de logminer.

```
SQL> exec dbms_logmnr.start_logmnr();

PL/SQL procedure successfully completed.
```

Ahora realizariamos una consulta sobre los ficheros de logminer.

```
SQL> select USERNAME, SQL_REDO from V$LOGMNR_CONTENTS where rownum < 10;

UNKNOWN

update "UNKNOWN"."OBJ# 15" set "COL 2" =
HEXTORAW('5f535953534d55335f31323034333
93036303624'), "COL 3" = HEXTORAW('c102'), "COL 4" = HEXTORAW('c104'), "COL 5" =
HEXTORAW('c2023d'), "COL 6" = HEXTORAW('c40f024c3b'), "COL 7" = HEXTORAW('80'),

USERNAME
```

SQL_REDO

"COL 8" = HEXTORAW('c25f3a'), "COL 9" = HEXTORAW('c2292d'), "COL 10" =
HEXTORAW
(80'), "COL 11" = HEXTORAW('c103'), "COL 12" = HEXTORAW('c103'), "COL 17" = HEX
TORAW('c103') where "COL 2" =
HEXTORAW('5f535953534d55335f3132303433393036303624
) and "COL 3" = HEXTORAW('c102') and "COL 4" = HEXTORAW('c104') and "COL 5" = H
EXTORAW('c2023d') and "COL 6" = HEXTORAW('c40e644126') and "COL 7" =
HEXTORAW('8
0') and "COL 8" = HEXTORAW('c25f39') and "COL 9" = HEXTORAW('c22927') and "COL 1
0" = HEXTORAW('80') and "COL 11" = HEXTORAW('c104') and "COL 12" =
HEXTORAW('c10
3') and "COL 17" = HEXTORAW('c103') and ROWID = 'AAAAAPAABAAAADhAAD';

USERNAME

SQL_REDO

UNKNOWN

commit;

9 rows selected.

Vemos que la salida es bastante compleja. Ahora modificaremos el fichero que lee esta vista hacia el fichero de nuestro logminer.

```
SQL> EXECUTE DBMS_LOGMNR.START_LOGMNR(OPTIONS =>  
DBMS_LOGMNR.DICT_FROM_ONLINE_CATALOG);
```

PL/SQL procedure successfully completed.

Ahora son comprensibles a simple vista los resultados realizamos una consulta dirigida a los cambios hecho en los segmentos del usuario prueba ya que ahí fue donde se realizó el insertado incorrecto de datos.

```
SQL> select SQL_REDO, SQL_UNDO , COMMIT_TIMESTAMP from
V$LOGMNR_CONTENTS where SEG_OWNER='PRUEBA';

SQL_REDO
-----

SQL_UNDO
-----

COMMIT_TI
-----

CREATE USER prueba
IDENTIFIED BY VALUES 'E552C348C0B39E0E' DEFAULT TABLESPACE prueba_ts
QUOTA UNLIMITED ON prueba_ts;

create table tabla1 (campo1 number);

insert into "PRUEBA"."TABLA1"("CAMPO1") values ('1');
delete from "PRUEBA"."TABLA1" where "CAMPO1" = '1' and ROWID =
'AAAb4zAAMAAAACDA
AA';

insert into "PRUEBA"."TABLA1"("CAMPO1") values ('2');
delete from "PRUEBA"."TABLA1" where "CAMPO1" = '2' and ROWID =
'AAAb4zAAMAAAACDA
AB';

insert into "PRUEBA"."TABLA1"("CAMPO1") values ('3');
delete from "PRUEBA"."TABLA1" where "CAMPO1" = '3' and ROWID =
```

```
'AAAb4zAAMAAAACDA  
AC';
```

Como vemos la columna "SQL_UNDO" nos muestra como deshacer ese cambio realizado en la tabla. Con lo cual simplemente procedemos al borrado de los datos.

```
delete from "PRUEBA"."TABLA1" where "CAMPO1" = '2' and ROWID =  
'AAAb4zAAMAAAACDA  
AB';
```

Webgrafia

Archivelog

<http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=559>

<http://www.orasite.com/oracle-rac/poner-base-de-datos-oracle-rac-11g-r2-en-modo-archivelog>

UNDO

http://cursos.atica.um.es/DBA9i1/pdf/contenidos/CursoDBA9i1_parte2_4p.pdf

<http://subway-shop.com/blogcolacios/2013/07/crecimiento-undoscriminado-tablespace-undo/>

<http://adminbd-dcueva.blogspot.com.es/2009/04/que-son-los-tablespace-del-tipo-undo-y.html>

http://www.runningoracle.com/product_info.php?products_id=48

https://docs.oracle.com/cd/B28359_01/server.111/b28310/undo005.htm#ADMIN11470

EXPORT IMPORT

<https://cajondesastreoracle.wordpress.com/tag/ventajas-de-datapump/>

<http://hypsoluciones.com/2015/05/como-usar-data-pump-expdp-e-impdp-da-oracle/>

<https://www.adictosaltrabajo.com/tutoriales/tutorial-impdp/>

https://docs.oracle.com/cd/B19306_01/server.102/b14215/dp_export.htm

<http://rincondeundba.blogspot.com.es/2010/09/expdp-fails-with-ora-31626-ora-31633.html>

<https://oracle-base.com/articles/10g/oracle-data-pump-10g>

<http://rincondeundba.blogspot.com.es/2010/09/data-pump-export-expdp-and-data-pump.html>

<https://oracle-base.com/articles/10g/oracle-data-pump-10g#SchemaExpImp>

http://www.dba-oracle.com/t_impdp_parameters.htm

RMAN

<http://rene-ace.com/funciona-respaldo-de-oracle-rman/>

<http://www.orasite.com/backup-de-base-de-datos/guia-rapida-para-rman>

<http://oracleprof.blogspot.com.es/2010/02/rman-and-sbttape-backup-pieces-how-to.html>

<https://juanmercadoit.com/2009/07/28/recuperando-ambiente-con-rman/>

Flashback query

https://docs.oracle.com/cd/B28359_01/appdev.111/b28424/adfns_flashback.htm#g1026131

<https://blog.avanttic.com/2011/07/22/operaciones-de-flashback-en-oracle-database/>

<https://www.toadworld.com/platforms/oracle/b/weblog/archive/2015/01/07/un-viaje-a-trav-233-s-del-tiempo-con-flashback-query>

LOGMINER

<http://oraclexperto.blogspot.com.es/2006/03/uso-de-logminer.html>

<http://www.oracleflash.com/24/How-to-read-or-analyze-redo-log-files-using-LogMiner.html>

Esta Webgrafia no contempla en la totalidad las páginas revisadas además de estudios en profundidad de los sistemas de forma propia.