

Taller de Seguridad, Backup y Recover en una Base de Datos Oracle 10g

Instructor: Ing. Francisco Riccio.

OCA Oracle Database 10g

OCP Oracle Database 10g

MCTS SQL Server 2005

Email: francisco@friccio.com

Fecha: 27 de Setiembre del 2008.

Seguridad & Auditoria

Creando Usuarios y Roles

- Cada usuario que se conecta a nuestra base de datos debe contar con un usuario con fines de auditoría.
- Un esquema es un usuario más sus objetos.
- Existen 3 modos de autenticar a un usuario y son: Password authentication (El password se almacena en el data dictionary), external authentication (Confía en el sistema operativo su autenticación) y global authentication (La autenticación se hace por mecanismos de seguridad avanzada como kerberos, radius, biometrica y certificados X.509).
- Cada usuario tiene asignado por default un tablespace donde almacenará sus objetos, a un tablespace temporal y a un profile (Sirve para limitar recursos al usuario y para asignarle políticas de password).

Creando un usuario con autenticación de base de datos:

```
create user nombre identified by password;
```

Creando un usuario con autenticación por sistema operativo:

Previamente validar el parámetro: os_authent_prefix

```
create user ops$nombre identified externally;
```

Donde nombre es un usuario de sistema operativo.

Desbloqueando / Bloqueando un usuario:

```
alter user usuario account unlock;
```

```
alter user usuario account lock;
```

Expirando un password:

```
alter user usuario password expire;
```

Configurando cuotas sobre tablespaces:

```
alter user xx identified by xx quota 100M | unlimited on USERS
```

Garantizando y Revocando Privilegios

Tipos de privilegios:

- **Objetos:** Permisos sobre objetos: tablas, índices, stored procedures, etc.
Podemos permitir garantizar un privilegio de objeto a un usuario y este pueda también dar este privilegio (WITH GRANT OPTION), no está disponible cuando el que va hacer garantizado es un rol.

Observación: Cuando revocamos un privilegio a un usuario y este a la vez entregó privilegios a un tercero, el tercero perderá el privilegio de acceder al objeto del primero.

Privilegio de Objetos: select, insert, update, delete, alter table, debug, execute, etc.

- **Sistema:** Permisos sobre operaciones de base de datos tales como: crear usuarios, alterar la base de datos, etc.
Podemos permitir garantizar un privilegio de sistema a un usuario y este pueda también dar este privilegio (WITH ADMIN OPTION),

Observación: Si un usuario otorga privilegios de sistema a otro usuario con la opción with admin option y a la vez este segundo usuario le da privilegios a un tercero.

Luego cuando se haga un revoke al segundo, el tercero sigue manteniendo el privilegio otorgado.

Privilegio de Sistema: alter database, alter system, audit system, create any (table – index – job – procedure – sequence – trigger - view), debug any procedure, execute any program, create procedure, create profile, create role, create session, lock any table, flashback any table, analyze any, sysdba, sysoper, etc.

- **Roles:** Permite agrupar privilegios de objetos y sistemas.
Para habilitar un rol: set role nombre_rol identified by password.

El password se setea de la siguiente manera: create rol nombre_rol identified by password.

Con: alter user nombre default role lista_rols (carga automáticamente los roles del usuario)

Recordemos que las vistas más utilizadas son: dba_role_privs (Privilegio de roles), dba_tab_privs (Privilegio de objetos de tablas) y dba_sys_privs (Privilegio de Sistemas).

Adicionalmente tenemos:

session_privs = Muestra los privilegios de sistema y objetos asignados a la sesión.

sessions_roles = Muestra los roles asignados a la sesión.

Nota:

Podemos crear roles con password:

```
create role rol1 identified by mipassword;
```

Para habilitar un rol con password en una sesión:

```
set role rol1 identified by mipassword, rol2;
```

```
set role all except rol1
```

```
set role none => Deshabilita todos.
```

Alterando los roles activados de un usuario determinado:

Cuando creamos un rol, el usuario creador adquiere el rol automáticamente como activo, pero podemos cambiar su comportamiento de la siguiente manera:

```
alter user xxx default role rol1, rol2; (si el rol tiene password en esta alteración no es necesario ponerlo)
```

Controlando Recursos a los Usuarios - Perfiles

En Oracle 10g nos permite limitar a un recurso mediante cuotas al consumo de espacio, CPU, memoria etc.

Los perfiles permiten limitar diversos recursos del CPU tales como: CPU time, memoria, cantidad de logical reads, etc.

Para habilitarlo: alter system set resource_limit = true scope=both

Opciones:

CONNECT TIME = Indica el tiempo de conexión que puede estar una sesión, en caso supere el tiempo en minutos aplica rollback a sus transacciones.

create profile nombre LIMIT CONNECT_TIME #min | UNLIMITED;

CPU_PER_CALL = Limita la cantidad de CPU que puede ser consumida por cualquier llamada de base de datos con ese profile. El valor a ingresar es en ciento de segundos en un ciclo.

create profile nombre LIMIT CPU_PER_CALL # | UNLIMITED;

CPU_PER_SESSION = Limita la cantidad de CPU time que puede ser consumida en cualquier sesión. El tiempo está en ciento de segundos.

IDLE_TIME = Limita la duración de tiempo muerto de una sesión. El tiempo está en minutos.

create profile nombre LIMIT IDLE_TIME #min | UNLIMITED;

LOGICAL_READS_PER_CALL = Por cada llamada hacia la base de datos restringe el número de logical reads que puede consumir la llamada.

create profile nombre LIMIT LOGICAL_READS_PER_CALL #;

LOGICAL_READS_PER_SESSION el cual indica el máximo número de logical reads durante toda la sesión.

PRIVATE_SGA = Limita la cantidad de UGA en bytes que una conexión con shared server puede ubicar.

create profile nombre LIMIT PRIVATE_SGA #;

SESSIONS_PER_USER = Restringue a un usuario al número de sesiones que puede abrir.

create profile nombre LIMIT SESSIONS_PER_USER #;

Aplicando Políticas de Passwords

Forzando a n intentos fallidos de password el bloqueo del user por un tiempo determinado:

```
ALTER PROFILE nombre LIMIT
```

```
FAILED_LOGIN_ATTEMPTS    #intentos
```

```
PASSWORD_LOCK_TIME       nMinutos (Vuelve a desbloquearlo) | UNLIMITED
```

```
PASSWORD_LIFE_TIME       x
```

```
PASSWORD_GRACE_TIME      y
```

El `password_life_time` especifica el máximo número de días que un password se puede mantener vigente.

El `password_grace_time` es el número de días después del primer login exitoso comenzará a recordarle del vencimiento del password.

Limita el password a m días y durante n días recordará esto.

```
ALTER PROFILE nombre LIMIT
```

```
PASSWORD_LIFE_TIME       m-n dias
```

```
PASSWORD_GRACE_TIME      n dias
```

Reuso de Passwords

Especifica el número de días que un password puede ser reusado y después de cuantos passwords cambiados.

```
CREATE PROFILE nombre LIMIT
```

```
PASSWORD_REUSE_TIME      #dias | UNLIMITED
```

```
PASSWORD_REUSE_MAX       n | UNLIMITED
```

Requiere que al menos n passwords cambien y durante #dias para poder reusar un password.

Nota:

Existe un archivo en llamado \$ORACLE_HOME/rdbms/admin/utlpwdmg.sql el cual contiene una función PL/SQL con complejidad alta para poderla asociar a un profile.

Ejemplo:

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION verify_function;
```

Auditoria

Para activar la auditoria:

ALTER SYSTEM SET AUDIT_TRAIL= DB; DB,EXTENDED, OS, XML, XML, EXTENDED SCOPE=SPFILE;

DB,EXTENDED: Registra en la base de datos junto con las sentencias SQL que son auditadas.

OS: Se registra en \$ORACLE_BASE/admin ó event viewer en Windows

Ejemplo: alter System set audit_trail=db scope=spfile;

Oracle permite 4 niveles de auditoría.

- Statement
- Privilegio
- Objeto
- Fine grained Access

Statement

Permite auditar 1 tipo de SQL.

Ejemplo:

audit table; (todo lo relacionado a la palabra table, create, drop, truncate, etc)

audit table by xxx

audit table by xxx whenever not successful

audit alter table | procedure | database | profile | system; grant (todas sus formas, ejemplo: grant sequence); execute (todas sus formas, ejemplo: execute procedure); procedure (create y drop para procedure, function, package); lock table; role (create, drop, set role, alter); select (select table); system grant (grant, revoke); tablespace (alter, create, drop); trigger (alter, create, drop); user (alter user, create, drop); view (alter, create, drop); session (connect).

Privilegio

Audita un privilegio de sistema, ejemplo:

```
audit create any table
```

```
audit delete any table
```

Tips:

Para auditar un GRANT ANY PRIVILEGE debemos primero ejecutar el siguiente audit en el usuario que va a lanzar los audit grant any privilege:

```
AUDIT SYSTEM GRANT BY usuario_auditor
```

Objeto

Audita un select, insert, update o execute de un objeto particular.

```
audit select on hr.employees;
```

```
audit select on hr.employees by access whenever successful;
```

Puede definírsele también a nivel de:

S = Session (Ejemplo: Si hacemos 10 inserts, los 10 inserts generaran 1 entrada de auditoría)

A = Access (Ejemplo: Si hacemos 10 inserts, cada insert generará 1 entrada de auditoría)

Nota:

Oracle 11g audita cualquier operación DML sobre la sys.aud\$

No puede ser restringido por usuario.

Si definimos por BY SESSION entonces en la tabla DBA_AUDIT_TRAIL (campo SES_ACTIONS) tendrá un conjunto de bits ordenados que representarán la lista de actividades que se hicieron sobre el objeto.

Cada posición del arreglo de bits está definida en la siguiente tabla:

<i>Position</i>	<i>Action</i>
1	ALTER
2	AUDIT
3	COMMENT
4	DELETE
5	GRANT
6	INDEX
7	INSERT
8	LOCAL
9	RENAME
10	SELECT
11	UPDATE
12	REFERENCE
13	EXECUTE

Donde cada celda tiene 3 valores: S (Success), F (Failure), B (Both).

Tengamos presente que una auditoria por sesión solo guardará la primera sentencia ejecutada y las demás sentencias relacionadas al mismo objeto ya no son guardadas en esa sesión. Para saber el historial de actividades debemos revisar la columna SES_ACTIONS.

Vistas importantes:

DBA_AUDIT_TRAIL = Muestra el reporte de las auditorias.

(Pero podemos filtrar con las vistas: DBA_AUDIT_STATEMENT, DBA_AUDIT_OBJECT)

DBA_COMMON_AUDIT_TRAIL = Muestra el reporte de las auditorias y FGA.

DBA_STMT_AUDIT_OPTS = Muestra auditorias activadas de tipo STATEMENT.

DBA_PRIV_AUDIT_OPTS = Muestra auditorias activadas de tipo PRIVILEGIO DE SISTEMA.

DBA_OBJ_AUDIT_OPTS = Muestra auditorias activadas de tipo OBJETO.

Nota:

Si queremos auditar al SYS debemos colocar en true el parámetro **audit_sys_operations**;

Las actividades de subir y bajar la base de datos y al loguarse como sysdba o sysoper son siempre escritas en disco.

Toda la auditoría se guarda internamente en la tabla SYS.AUD\$ la cual debemos ir depurándola en el tiempo y con su previo backup (export).

Fine Grained Auditing

Permite monitorear el acceso a la información basado en el contenido de la data.

Nos apoyamos del package: DBMS_FGA

```
DBMS_FGA.ADD_POLICY('esquema','objeto_monitorear','un_alias','condicion_logica*','columnas_visualizar*');
```

Parámetros:

condicion_logica = Si devuelve true o null crea un registro en la tabla de auditoría.

columnas_visualizar = Las columnas que se proyecten en el select no deben aparecer en la condicion_logica.

Aquí deben ir las columnas que se proyecten en un select, si es así y condicion_logica es true entonces se crea un registro de auditoría.

Si colocamos NULL en este parámetro indica que cualquier campo se procederá a auditar. Va de la mano con el resultado de la condicion_logica.

Todos los Parámetros:

object_schema

object_name

policy_name

audit_condition

audit_column = Lo definimos cuando nos interesa que solo se genere una entrada de auditoría cuando las columnas sean accedidas.

handler_schema

handler_module (UTL_MAIL.SEND_ATTACH_RAW)

enable = true/false

statement_types = select, insert, update, delete

audit_column_ops = DBMS_FGA.ALL_COLUMNS | DBMS_FGA.ANY_COLUMNS (Indica si todos los campos referenciados en audit_column deben figurar o solo algunos)

Ejemplos:

Ejemplo 1:

```
DBMS_FGA.ADD_POLICY  
(object_schema=>'SCOTT',object_name=>'EMP',policy_name=>'EMP_ACCESS');
```

Al no tener condiciones Oracle auditará toda operación DML y SELECT.

Ejemplo 2:

```
DBMS_FGA.ADD_POLICY  
(object_schema=>'SCOTT',object_name=>'EMP',policy_name=>'EMP_ACCESS',audit_column=>'SAL,  
COMM');
```

Indicamos que se generará cualquier entrada de auditoría si hay cambios y lecturas sobre el campo SAL ó COMM.

Ejemplo 3:

```
DBMS_FGA.ADD_POLICY  
(object_schema=>'SCOTT',object_name=>'EMP',policy_name=>'EMP_ACCESS',audit_column=>'SAL,  
COMM', statement_types=>'INSERT, UPDATE', audit_column_opts=>DBMS_FGA.ANY_COLUMNS);
```

Solamente sentencias INSERT y UPDATE que incluyan cualquiera (DBMS_FGA.ANY_COLUMNS “default”) de los campos listados en audit_column.

Ejemplo 4:

```
DBMS_FGA.ADD_POLICY  
(object_schema=>'SCOTT',object_name=>'EMP',policy_name=>'EMP_ACCESS',audit_column=>'SAL',  
audit_condition=>'SAL>9999',statement_types=>'SELECT',audit_column_opts=>DBMS_FGA.ANY_CO  
LUMNS,audit_trail=>DBMS_FGA.DB + DBMS_FGA.EXTENDED);
```

Si cumple la condición del audit_condition se generará una entrada de auditoría.

Ejemplo 5:

```
DBMS_FGA.ADD_POLICY (object_schema=>'SCOTT',object_name=>'EMP',  
policy_name=>'xxx',audit_column=>'CAMPO1,CAMPO2',enable=>FALSE,statement_types=>'SELECT')
```

Ejemplo 6:

```
DBMS_FGA.ENABLE_POLICY(object_schema=>'HR',object_name=>'EMPLOYEES',policy_name=>'xxx')
```

Ejemplo 7:

```
DISABLE_POLICY(object_schema=>'HR',object_name=>'EMPLOYEES',policy_name=>'xxx');
```

Ejemplo 8:

```
DROP_POLICY(object_schema=>'HR',object_name=>'EMPLOYEES',policy_name=>'xxx');
```

Vistas importantes:

DBA_AUDIT_POLICIES = Reporte de las reglas FGA

DBA_FGA_AUDIT_TRAIL = Reporte del resultado de la auditoria FGA.

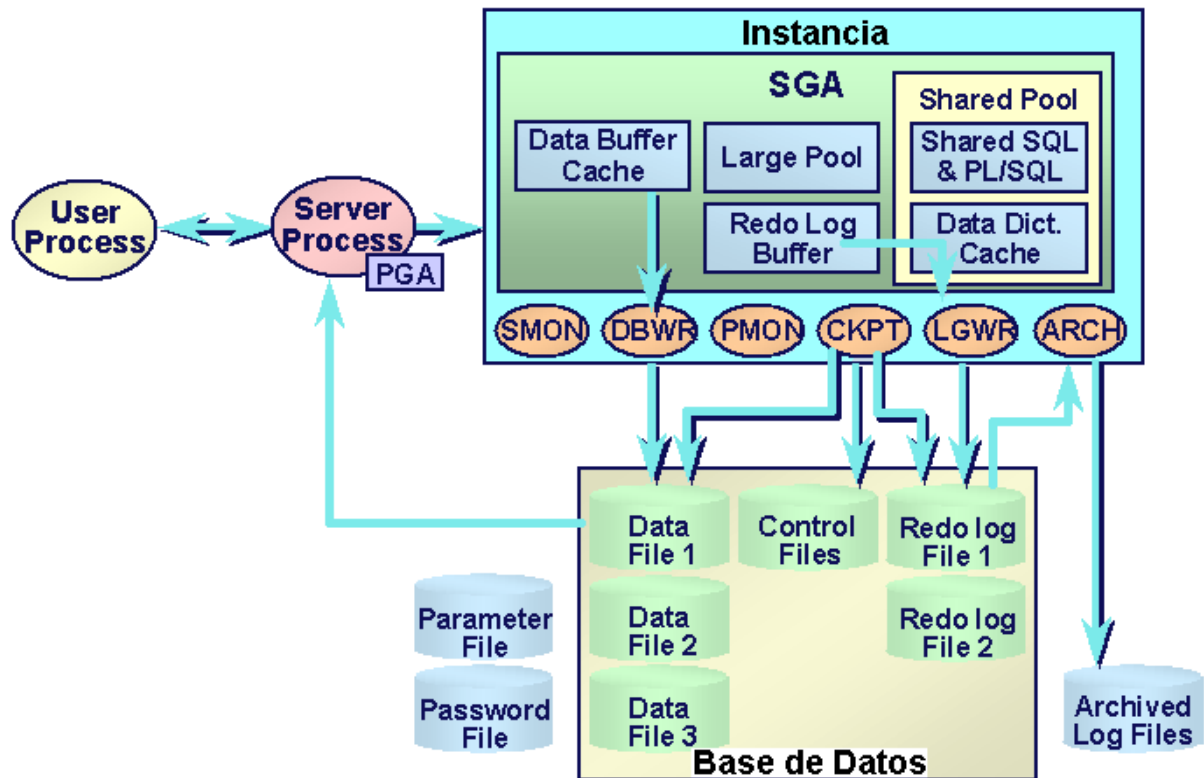
Recomendaciones y Best Practices

1. Tener el parámetro de base de datos: **07_DICTIONARY_ACCESSIBILITY = FALSE**
2. Revisar al public el permiso de execute de los siguientes paquetes:
 - UTL_TCP
 - UTL_SMTP
 - UTL_HTTP
 - UTL_FILE
 - DBMS_OBFUSCATION_TOOLKIT
 - DBMS_CRYPTO
3. No garantizar el privilegio de SYSDBA ni entregar el rol DBA ni ningún privilegio de sistema con ANY.
4. Tener el parámetro de base de datos: **REMOTE_OS_AUTHENT = FALSE**
5. Los archivos del \$ORACLE_HOME y \$ORACLE_BASE deben tener privilegios 750 a nivel de S.O.
6. Auditar toda acción en la SYS.AUD\$: AUDIT ALL ON SYS.AUD\$ BY ACCESS.
7. Auditar cualquier DROP ANY y GRANT ANY.

Backup

Overview Arquitectura Oracle

Una base de datos Oracle se compone de sus Users Process, su Instancia y su base de datos.



- El user process, es usado para almacenar la conexión del usuario. Las variables e información de cada usuario se almacenan en el PGA.
- La instancia es el SGA + background process
El SGA se compone del:
 - Data Buffer Cache, almacena los bloques leídos de disco en memoria.
 - Shared Pool, almacena los planes de ejecución y código compilado.
 - Large Pool, es utilizado como buffer para los backups y para ejecución de queries con paralelismo.
 - Redo Log Buffer, almacena las transacciones antes de cada commit.

Background process:

- PMON:

- Libera todos los recursos que una sesión usó después que se cierra dicha sesión.
 - Responsable de indicarle al Listener cuando una sesión se ha llegado a conectar con un dispatcher en una conexión compartida para que el Listener aumente en 1 su contador.
 - Responsable de registrar dinámicamente las instancias en los listener con configuración por default.
 - SMON:
 - Maneja espacios para operaciones de ordenamiento.
 - Es el responsable de un instance recovery (en caso de una caída).
 - MMON:
 - Recolecta estadísticas y los almacena en el buffer de AWR.
 - Limpia los traces e inputs de incidentes con origen de más de 30 días (Oracle 11g).
 - MMNL (Memory Monitor Light): Baja del buffer del AWR a disco cada segundo.
 - Job Queue Monitor (CJQ): Asigna jobs a la cola de jobs. (Job Coordinator)
 - Job Queue (J): Ejecuta jobs de la base de datos que han sido schedule. (Job Slave)
 - MMAN (Memory Manager): Es activado en el Database Control y esto permite que el ADDM ajuste cada componente del SGA según las estadísticas.
 - Recovery Writer (RVWR): Escribe recovery información cuando se habilita el flash recovery.
 - Change Tracking Writer (CTWR): Mantiene tracks de los bloques cambiados desde un backup incremental por el RMAN.
 - DBWriter baja a disco en las siguientes circunstancias:
 - No hay espacio libre en el buffer cache para nueva data.
 - Demasiados commits que no han bajado a disco.
 - Usando shutdown o usando checkpoint.
 - El tablespace es puesto en modo backup, offline o readonly.
 - Un segmento es borrado.
 - LOGWriter baja a disco en las siguientes circunstancias:
 - Cada 3 segundos.
 - Cuando se hace un commit.
 - Cuando el redo buffer tiene 1 MB de redo información.
 - Checkpoint.
 - Checkpoint (CKPT)
 - Actualiza las cabeceras del control file y de los datafiles igualando el mismo SCN.
 - DMnn_IDProceso: Es el master control process en un trabajo de Oracle Data Pump.
 - DWnn_IDProceso: Es el worker process en un trabajo de Oracle Data Pump.
- Base de Datos
 - Redo Logs, contienen las transacciones de la base de datos.
 - Datafiles, contienen la información de nuestra base de datos. La base de datos debe tener al menos un datafile y cada datafile pertenece a un tablespace.
 - Log files, contienen las transacciones commiteadas utilizadas para recovery.
 - Control file: Mantiene la metadata de la base de datos, como donde se ubican sus datafiles, el nombre de la base de datos y algunas configuraciones de retención de los backups.
 - SPFILE y PFILE: Mantiene la configuración de la instancia, donde se define el nombre de la instancia, la memoria RAM de cada componente del SGA etc.

- Password file: Permite indica cuantas cuentas pueden tener el privilegio SYSDBA.

Multiplexación de Redo Logs y Control Files

Redo Logs.

Para visualizar cuantos **redo log groups** tenemos, consultamos la vista: **v\$log** y para saber la descripción de los **miembros de cada grupo** en toda la base de datos consultamos la vista **v\$logfile**.

Ejemplo:

```
SQL> select GROUP#, SEQUENCE#, bytes/1024/1024 as TAM_SIZE_MB, archived, members, status from v$log;
```

GROUP#	SEQUENCE#	TAM_SIZE_MB	ARC	MEMBERS	STATUS
1	8	50	YES	1	INACTIVE
2	9	50	NO	1	CURRENT
3	7	50	YES	1	INACTIVE

```
SQL> select group#, member from v$logfile;
```

GROUP#	MEMBER
1	/u02/oradata/ORCL/redo01.log
2	/u02/oradata/ORCL/redo02.log
3	/u02/oradata/ORCL/redo03.log

Es importante resaltar lo siguiente:

- Oracle solo requiere 1 miembro por redo log group para seguir operando.
- Solo existe un redo log group activo, el cual recibe los redo entries del redo log buffer.
- Los redo log groups tienen 4 estados: ACTIVO (Se requiere para instance recovery ya que aún no ha ocurrido un checkpoint), CURRENT (Redo log group que recibe los redo entries del redo log buffer), UNUSED (Redo log recién creado y no ha sido utilizado) e INACTIVO (No es requerido en caso de un instance recovery).
- Al llenarse un redo log group se genera el evento de switch log file el cual genera 2 eventos importantes: CHECKPOINT y ARCHIVE (Si está habilitado), ambos ocurren asincrónicamente.

Operaciones sobre los redo logs groups

Creando y eliminando redo logs groups.

```
SQL> alter database add logfile group 4 ('/u02/oradata/ORCL/redo04a.log', '/u02/oradata/ORCL/redo04b.log') size 50M;
```

```
Database altered.
```

```
SQL> alter database drop logfile group 4;
```

```
Database altered.
```

Agregando y eliminando miembro de un redo log group.

```
SQL> alter database add logfile member '/u02/oradata/ORCL/redo03a.log' to group 3;
Database altered.

SQL> alter database drop logfile member '/u02/oradata/ORCL/redo03a.log';
Database altered.
```

Nota:

- Es una buena recomendación tener más de 1 miembro por grupo y estén en carpetas diferentes o en filesystems diferentes si es posible.
- Se puede trabajar al menos con 2 redo log groups pero Oracle recomienda al menos 3 y si la base de datos es muy transaccional es recomendable generar varios redo log groups porque provoca lentitud en el sistema al no poderse utilizar un redo log group libre para escribir ya que todos podrían estar haciendo la operación de archive todavía.

Control Files.

Para visualizar la ubicación de nuestros control files. (**show parameter control_files**)

```
SQL> show parameter control_files
```

NAME	TYPE	VALUE
control_files	string	/u02/oradata/ORCL/control01.ct 1, /u02/oradata/ORCL/control02 .ctl, /u02/oradata/ORCL/contro l03.ctl

```
SQL> select VALUE from v$parameter where name='control_files';
```

```
VALUE
-----
/u02/oradata/ORCL/control01.ctl, /u02/oradata/ORCL/control02.ctl, /u02/oradata/ORCL/control03.ctl
```

Se puede apreciar en este ejemplo que tenemos 3 control files llamados control01.ctl, control02.ctl y control03.ctl.

Para agregar más de un control file se debe realizar las siguientes labores:

- Modificar el parámetro control_file para indicarle que tendrá otro control file más (scope=spfile).
- Bajar la instancia de base de datos.
- Copiar un control file a otro destino.
- Levantar la base de datos.

Ejemplo:

```

SQL> alter system set control_files='/u02/oradata/ORCL/control01.ctl',
2  '/u02/oradata/ORCL/control02.ctl', '/u02/oradata/ORCL/control03.ctl',
3  '/u02/oradata/ORCL/control04.ctl' scope=spfile;

System altered.

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> !
[oracle@pcoracle ~]$ cp /u02/oradata/ORCL/control01.ctl /u02/oradata/ORCL/control04.ctl
[oracle@pcoracle ~]$ exit
exit

SQL> startup
ORACLE instance started.

Total System Global Area 184549376 bytes
Fixed Size 1218412 bytes
Variable Size 71305364 bytes
Database Buffers 109051904 bytes
Redo Buffers 2973696 bytes
Database mounted.
Database opened.
SQL> show parameter control_files

```

NAME	TYPE	VALUE
control_files	string	/u02/oradata/ORCL/control01.ctl, /u02/oradata/ORCL/control02.ctl, /u02/oradata/ORCL/control03.ctl, /u02/oradata/ORCL/control04.ctl

La misma idea sería para retirar un control file de la base de datos.

Oracle recomienda tener al menos 3 control files dispersos en diferentes carpetas o si es posible en diferentes filesystems.

Realizando User Managed Backups (Online / Offline)

OFFLINE

Tenemos 2 métodos para realizar backups offline de tipo User Managed.

- a) Bajando la base de datos y hacer una copia de los files a nivel de sistema operativo.
- b) Bajando la base de datos y subiéndola en modo MOUNT y hacer el backup con RMAN o con comandos de sistema operativo.

Ejemplo:

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> !
[oracle@pcoracle ~]$ cp -r /u02/oradata/ORCL /u03/.
```

ONLINE

Esté método nos permite sacar backups en caliente a la base de datos es decir sin generar indisponibilidad de servicios.

Requisito principal: La base de datos debe estar en modo ARCHIVE.

Validando si la base de datos está en modo archive:

```
SQL> select log_mode from v$database;
```

```
LOG_MODE
-----
ARCHIVELOG
```

Si deseamos sacar un backup completo en modo online a la base de datos:

```
SQL> alter database begin backup;
```

Database altered.

```
SQL> !
[oracle@pcoracle ~]$ cp -r /u02/oradata/ORCL /u03/.
[oracle@pcoracle ~]$ exit
exit
```

```
SQL> alter database end backup;
```

Database altered.

Si deseamos sacar backup en modo online a un Tablespace:

```
SQL> alter tablespace USERS begin backup;
```

Tablespace altered.

```
SQL> !
```

```
[oracle@pcoracle ~]$ cp /u02/oradata/ORCL/users01.dbf /u03/ORCL/users01.dbf
```

```
[oracle@pcoracle ~]$ exit
```

```
exit
```

```
SQL> alter tablespace USERS end backup;
```

Tablespace altered.

Nota:

- Cuando un Tablespace se coloca en modo begin backup, Oracle internamente hace que los datafiles comprendidos en el Tablespace no sean escritos por el proceso DBWRITER y todos los cambios ocurridos son escritos en los redo logs (tendrá la copia de cada bloque cambiado), por lo tanto genera gran cantidad de redo logs, porque se está escribiendo más información de lo normal.
- Oracle recomienda ejecutar este tipo de backups en horarios más libres.
- Cuando un Tablespace está en modo backup sus datafiles están en estado backup y se puede apreciar en el siguiente query (v\$backup):

```
SQL> alter tablespace USERS begin backup;
```

Tablespace altered.

```
SQL> select * from v$backup;
```

FILE#	STATUS	CHANGE#	TIME
1	NOT ACTIVE	498997	26-SEP-09
2	NOT ACTIVE	498997	26-SEP-09
3	NOT ACTIVE	498997	26-SEP-09
4	ACTIVE	499337	26-SEP-09

```
SQL> alter tablespace USERS end backup;
```

Tablespace altered.

Configuración del Flash Recovery Area

El Flash Recovery Area es un espacio de disco que se le ha asignado a Oracle para poder almacenar datafiles, control files, online redo logs y principalmente archivers y backups.

El flash recovery area puede albergar más de una base de datos de modo que se tendría centralizado.

Disponible desde Oracle Database 10g.

Para saber la ubicación de nuestro flash recovery area y su capacidad es consultando los siguientes parámetros:

```
SQL> show parameter db_recovery_file_dest
```

NAME	TYPE	VALUE
db_recovery_file_dest	string	
db_recovery_file_dest_size	big integer	0

En este ejemplo no lo tenemos setado.

Para configurarlo:

```
SQL> !  
[oracle@pcoracle ~]$ mkdir /u02/flash_recovery_area  
[oracle@pcoracle ~]$ exit  
exit
```

```
SQL> alter system set db_recovery_file_dest_size=3G;
```

System altered.

```
SQL> alter system set db_recovery_file_dest='/u02/flash_recovery_area';
```

System altered.

Tener en consideración lo siguiente:

- Si se setea el Flash Recovery Area y no esta seteando ningún parámetro LOG_ARCHIVE_DEST_n el LOG_ARCHIVE_DEST_10 equivale a DB_RECOVERY_FILE_DEST.

Si queremos hacerlo de modo explícito sería:

```
alter system set log_archive_dest_n='LOCATION=USE_DB_RECOVERY_FILE_DEST';
```

Y si está seteado al menos un LOG_ARCHIVE_DEST_n en otras rutas y no se setea algún LOG_ARCHIVE_DEST_n en el Flash Recovery Area, los archivers nunca irán al Flash Recovery Area.

- Tener nuestros archivers y backups en el flash recovery area nos da la ventaja que Oracle generará una autolimpieza y nos alerta. Cuando está al 87% de consumo el flash recovery area

recibimos alertas en el alert.log y en la vista **DBA_OUTSTANDING_ALERTS** (Devuelve alertamientos de métricas que están alcanzando un umbral determinado).

- Para monitorear el espacio consumido lo podemos hacer consultando la vista **V\$FLASH_RECOVERY_FLASH_USAGE**.

```
SQL> select * from v$flash_recovery_area_usage;
```

FILE_TYPE	PERCENT_SPACE_USED	PERCENT_SPACE_RECLAIMABLE	NUMBER_OF_FILES
CONTROLFILE	0	0	0
ONLINELOG	0	0	0
ARCHIVELOG	0	0	0
BACKUPPIECE	0	0	1
IMAGECOPY	0	0	0
FLASHBACKLOG	0	0	0

```
6 rows selected.
```

Configurando RMAN

Ingresamos a RMAN y se verifica su configuración.

```
[oracle@pcoracle u03]$ rman target /
Recovery Manager: Release 10.2.0.1.0 - Production on Sat Sep 26 08:13:53 2009
Copyright (c) 1982, 2005, Oracle. All rights reserved.
connected to target database: ORCL (DBID=1217299163)
RMAN> show all;

using target database control file instead of recovery catalog
RMAN configuration parameters are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default
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 ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/u01/app/oracle/db/10.2/dbs/snapcf_ORCL.f'; # default
```

También se puede visualizar la configuración de RMAN consultando la vista **V\$RMAN_CONFIGURATION**.

Configurando retención de Backups

En RMAN podemos configurar 3 tipos de backups: Por número de copias, por tiempo y no configurado por RMAN.

Por número de copias: Indicamos a Oracle que debe existir n copias de un backup para considerarlo obsoleto el más antiguo.

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

Por días de retención: Indicamos a Oracle el número de días que debe tener un backup para ser considerado vigente todavía.

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF # DAYS;
```

Si un backup no cumple el período de días de ser considerado vigente pero es necesario para tener un backup completo de la base de datos; Oracle no lo marcará como obsoleto.

Asimismo siempre Oracle mantendrá un backup completo de la base de datos.

El número de días de retención no puede ser mayor al especificado en el parámetro **CONTROL_FILE_RECORD_KEEP_TIME** si no estamos utilizando Catalogo.

Sin retención: Indica que el backup nunca será obsoleto, pero le da el derecho a la herramienta de terceros que realiza el backup a hacerlo.

Compression

Solo disponible desde la versión Oracle Database 11g, el cual Oracle nos permite tener 2 tipos de compresiones: ZLIB y BZIP2.

Por default está habilitada la BZIP2 y consume mayor procesamiento de CPU pero comprime más, mientras ZLIB comprime consumiendo menos recursos pero comprime con menor ratio.

Ejemplo: `CONFIGURE COMPRESSION ALGORITHM 'ZLIB';`

Encriptación

Nota: 316886.1

Está disponible desde la versión 10gR2 y solo se puede realizar encriptación a backupset si es a disco y si es por tape solo utilizando Oracle Secure Backup.

Modos de encriptación:

Transparent: Requiere un Oracle Key Management Infrastructure disponible como Oracle Wallet y no requiere intervención del DBA cuando está configurado.

Password: Requiere un Password el backup para setearlo.

Dual: Soporta Transparent y Password.

Pasos:

Validamos los algoritmos disponibles:

```
select ALGORITHM_NAME from V$RMAN_ENCRYPTION_ALGORITHMS;
```

Seteándolo en el RMAN:

```
CONFIGURE ENCRYPTION ALGORITHM 'AES128';
```

Ejecutando un backup con Password:

```
set encryption on identified by password only;
```

Ejemplo:

```
RMAN> set encryption on identified by friccio only;
```

executing command: SET encryption

```
RMAN> backup as compressed backupset database;
```

```
Starting backup at 13-OCT-09
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=159 devtype=DISK
channel ORA_DISK_1: starting compressed full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00001 name=/u02/oradata/ORCL/system.dbf
input datafile fno=00002 name=/u02/oradata/ORCL/undotbs01.dbf
input datafile fno=00003 name=/u02/oradata/ORCL/sysaux01.dbf
input datafile fno=00004 name=/u02/oradata/ORCL/users01.dbf
channel ORA_DISK_1: starting piece 1 at 13-OCT-09
□
```

Para Restaurar:

```
set decryption identified by password;
```

```
restore database;
```

Nota:

Si habilitamos la opción:

```
CONFIGURE ENCRYPTION FOR DATABASE ON
```

Debemos configurar un wallet de modo que **todo backup siempre saldrá encriptado**.

BACKUP OPTIMIZATION

RMAN tiene la facilidad de no backupear files que no cambian como los datafiles read only si ya fue backupeado anteriormente, logrando disminuir el tiempo del backup.

```
CONFIGURE BACKUP OPTIMIZATION ON;
```

Aplica a: datafiles, archive logs y backup sets.

Si queremos forzar que de todos modos haga un backup a los datafiles read only a pesar que está con la opción de optimización habilitada:

```
BACKUP DATABASE FORCE;
```

CONTROLFILE AUTOBACKUP

Podemos indicar a RMAN que siempre que ejecute un backup incluya al CONTROLFILE y al SPFILE (Si está usando SPFILE en la base de datos) en cada backup que se realice.

Por default un backup database incluye el control file.

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

SNAPSHOT CONTROLFILE

Oracle cuando ejecuta un backup y necesita re sincronizar con una consistente versión del control file se apoya de una copia del control file que la realiza cuando inicia el backup.

También lo realiza cuando desea realizar un backup del control file.

```
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/u02/oradata/ORCL/SNAPORA.CTL';
```

Si 2 backups desean realizar la misma operación fallará (Después de 5 intentos al querer tener la exclusividad del archivo) entregando el siguiente error:

```
RMAN-08512: Waiting For Snapshot ControlFile Enqueue
```

Especificando el canal (DISK o TAPE)

Especificando el canal por default.

```
TAPE = SBT
```

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

Especificando algunos canals:

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 1;
```

```
CONFIGURE DEVICE TYPE SBT PARALLELISM 2;
```

Configurando el tamaño de un backup pieces - MAXPIECESIZE

El backup pieces es un archive, donde un conjunto de backup pieces forman el backup de un datafile.

```
CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE = 1G;
```

Nota:

Existe un bug en las versiones 10.1.0.2 hasta la 10.2.0.3 entregando el siguiente error cuando se realiza un backup image cuando esta seteado el MAXPIECESIZE.

During an RMAN backup, a tracefile is generated with this error:

ORA-00600: internal error code, arguments: [ksfdfetch1], [], [], [], [], [], [], []

Se soluciona en la versión 10.2.0.4 o retirando el MAXPIECESIZE ó colocando el parche 4541506.

Configurando el tamaño de un backupset - MAXSETSIZE

Un backupset es igual a la suma de cada backup piezas que son parte del backupset.

CONFIGURE MAXSETSIZE TO 1G;

Ó

RMAN> BACKUP DATABASE MAXSETSIZE=1G;

Nota:

No podemos especificar la cantidad de backuppieces en un backupset.

Si un archivo ocupa más que el backupset, el backup fallará.

Podemos especificar el número de archivos por backupset con la propiedad filesperset=#.

Ejemplo:

```
RMAN> backup database filesperset 2;
```

```
Starting backup at 13-OCT-09
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00001 name=/u02/oradata/ORCL/system.dbf
input datafile fno=00003 name=/u02/oradata/ORCL/sysaux01.dbf
channel ORA_DISK_1: starting piece 1 at 13-OCT-09
channel ORA_DISK_1: finished piece 1 at 13-OCT-09
piece handle=/u02/flash_recovery_area/ORCL/backupset/2009_10_13/o1_mf_nnndf_TAG2
0091013T141456_5f9nfjvl_.bkp tag=TAG20091013T141456 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:45
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00002 name=/u02/oradata/ORCL/undotbs01.dbf
input datafile fno=00004 name=/u02/oradata/ORCL/users01.dbf
channel ORA_DISK_1: starting piece 1 at 13-OCT-09
```

ARCHIVELOG DELETION POLICY

Indicamos la política de eliminación de los archive logs.

Puede tener 3 valores:

- APPLIED ON STANDBY, Habilita al Flash Recovery Area eliminar al archive log cuando el archive ya fue aplicado en el Standby.
- NONE (default), Habilita al Flash Recovery Area a eliminar los archive logs que son backupeados y que son obsoletos según en la retención del backup.

Si realizamos un backup eliminando los archive log (delete input) y esta seteado alguna alguna política de eliminación de archivelog hará que el comando no elimine el archivelog a menos que coloquemos forcé.

```

RMAN> backup as compressed backupset archivelog all delete input force;

```

```

Starting backup at 13-OCT-09
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting compressed archive log backupset
channel ORA_DISK_1: specifying archive log(s) in backup set
input archive log thread=1 sequence=13 recid=36 stamp=700145283
channel ORA_DISK_1: starting piece 1 at 13-OCT-09
channel ORA_DISK_1: finished piece 1 at 13-OCT-09
piece handle=/u02/flash_recovery_area/ORCL/backupset/2009_10_13/o1_mf_anr
NE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
channel ORA_DISK_1: deleting archive log(s)
archive log filename=/u03/1_13_689523803.dbf recid=36 stamp=700145283
Finished backup at 13-OCT-09

```

Realizando Backups con RMAN

Backup Image:

Backup Base de Datos: BACKUP AS COPY DATABASE;

Backup Tablespace: BACKUP AS COPY TABLESPACE xxx;

Backup Datafile: BACKUP AS COPY DATAFILE #;

Backup ControlFile: BACKUP AS COPY CURRENT CONTROLFILE;

Backup SPFILE: BACKUP AS COPY SPFILE;

```
RMAN> backup as copy tablespace USERS;
```

```
Starting backup at 13-OCT-09
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
input datafile fno=00004 name=/u02/oradata/ORCL/users01.dbf
output filename=/u02/flash_recovery_area/ORCL/datafile/o1_mf_users_5f9j
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 13-OCT-09
```

```
RMAN> backup as copy datafile 2;
```

```
Starting backup at 13-OCT-09
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
input datafile fno=00002 name=/u02/oradata/ORCL/undotbs01.dbf
output filename=/u02/flash_recovery_area/ORCL/datafile/o1_mf_undotbs1_5
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:15
Finished backup at 13-OCT-09
```

```
RMAN> backup as copy current controlfile;
```

```
Starting backup at 13-OCT-09
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
copying current control file
output filename=/u02/flash_recovery_area/ORCL/controlfile/o1_mf_TAG2009
8
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:02
Finished backup at 13-OCT-09
```

```
RMAN> backup as copy spfile;
```

```
Starting backup at 13-OCT-09
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
including current SPFILE in backupset
channel ORA_DISK_1: starting piece 1 at 13-OCT-09
channel ORA_DISK_1: finished piece 1 at 13-OCT-09
piece handle=/u02/flash_recovery_area/ORCL/backupset/2009_10_13/o1_mf_r
NE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
Finished backup at 13-OCT-09
```

Nota: Recordemos que el backup image es un backup como si fuera una copia de sistema operativo y no puede ser enviado directamente a tape.

Backupset:

Backup Base de Datos: BACKUP AS BACKUPSET DATABASE;

Backup Tablespace: BACKUP AS BACKUPSET TABLESPACE xxx;

Backup Datafile: BACKUP AS BACKUPSET DATAFILE #;

Backup ControlFile: BACKUP CURRENT CONTROLFILE;

Backup SPFILE: BACKUP AS BACKUPSET SPFILE;

```
RMAN> BACKUP AS BACKUPSET TABLESPACE USERS;
```

```
Starting backup at 13-OCT-09
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00004 name=/u02/oradata/ORCL/users01.dbf
channel ORA_DISK_1: starting piece 1 at 13-OCT-09
channel ORA_DISK_1: finished piece 1 at 13-OCT-09
piece handle=/u02/flash_recovery_area/ORCL/backupset/2009_10_13/o1_mf_n
NE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-OCT-09
```

```
RMAN> BACKUP AS BACKUPSET DATAFILE 2;
```

```
Starting backup at 13-OCT-09
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00002 name=/u02/oradata/ORCL/undotbs01.dbf
channel ORA_DISK_1: starting piece 1 at 13-OCT-09
channel ORA_DISK_1: finished piece 1 at 13-OCT-09
piece handle=/u02/flash_recovery_area/ORCL/backupset/2009_10_13/o1_mf_n
NE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
Finished backup at 13-OCT-09
```

```
RMAN> BACKUP CURRENT CONTROLFILE;
```

```
Starting backup at 13-OCT-09
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
including current control file in backupset
channel ORA_DISK_1: starting piece 1 at 13-OCT-09
channel ORA_DISK_1: finished piece 1 at 13-OCT-09
piece handle=/u02/flash_recovery_area/ORCL/backupset/2009_10_13/o1_mf_n
NE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
Finished backup at 13-OCT-09
```

```
RMAN> BACKUP AS BACKUPSET SPFILE;
```

```
Starting backup at 13-OCT-09
```

Nota:

El backupset es un backup con formato solo legible por RMAN y puede ser enviado a disco o a tape.

El backupset lo podemos comprimir con la opción BACKUPSET COMPRESSED (Desde Oracle 10g).

Backup del Control File con diferentes Métodos.

- a) Backup en forma de script: alter database backup controlfile to trace;
Se almacena en el UDUMP como un trace de usuario (USER_DUMP_DEST)
- b) Backup en forma binaria: alter database backup controlfile to 'destino';
- c) Utilizando RMAN: backup current controlfile;

Ejemplo:

```
SQL> alter database backup controlfile to trace;
```

```
Database altered.
```

```
SQL> alter database backup controlfile to '/tmp/control_bk';
```

```
Database altered.
```

```
SQL> !
```

```
[oracle@pcoracle u03]$ rman target /
```

```
Recovery Manager: Release 10.2.0.1.0 - Production on Tue Oct 13 13:26:49 2009
```

```
Copyright (c) 1982, 2005, Oracle. All rights reserved.
```

```
connected to target database: ORCL (DBID=1217299163)
```

```
RMAN> backup current controlfile;
```

```
Starting backup at 13-OCT-09
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=139 devtype=DISK
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
including current control file in backupset
channel ORA_DISK_1: starting piece 1 at 13-OCT-09
channel ORA_DISK_1: finished piece 1 at 13-OCT-09
piece handle=/u02/flash_recovery_area/ORCL/backupset/2009_10_13/o1_mf_ncnnf_TA
NE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
Finished backup at 13-OCT-09
```

Planificando Backups Incrementales

Conceptos:

- a) Whole Backup = Es un backup completo de toda la base de datos (Datafiles + Control files, no incluye online redo logs).
- b) Partial Database = Es cualquier elemento a backuppear de una base de datos mientras que no sea toda la base de datos.
- c) Full = Incluye todos los data blocks de cada datafile que se backuppea en un WHOLE o PARTIAL DATABASE.
- d) Incremental = Trae todos los nuevos data blocks modificados o creados desde el último backup. Existe 5 niveles de backups (De 0 a 4), 1 es el más usado y 0 (**LO**) es el base line (es un full backup).
- e) Consistente = Es un offline backup. (El SCN de cada datafile coincide con el SCN de cada control file).
- f) Inconsistente = Online backups. SCN en el control file es inconsistente con respecto a las cabeceras de los data files. Inconsistente backups requieren recovery cuando ellos son usados para una restauración.
- g) Backup Set = Es un conjunto de datafiles o archived redo logs que son almacenados en 1 solo archivo en un formato que solo es leído por RMAN. (**No incluye data blocks no usados**).
- h) Image Copy = Copia data blocks usados y no usados, siendo una copia exacta de cada datafile. (No puede ser enviado a TAPE directamente).

Nivel 0 = Full Backup

RMAN> BACKUP INCREMENTAL LEVEL 0 AS COMPRESSED BACKUPSET DATABASE;

Nivel 1 = Incremental

RMAN> BACKUP INCREMENTAL LEVEL 1 AS COMPRESSED BACKUPSET DATABASE;

(Copia los últimos data blocks cambiados desde el último backup L0 o L1 más próximo).

Nivel 1 Acumulativo

RMAN> BACKUP INCREMENTAL LEVEL 1 CUMULATIVE AS COMPRESSED BACKUPSET DATABASE;

(Copia los bloques modificados desde ese momento hacia el backup L0 o Ln-1)

Ejemplo:

```
RMAN> backup incremental level 0 as compressed backupset tablespace USERS;

Starting backup at 13-OCT-09
using channel ORA_DISK_1
channel ORA_DISK_1: starting compressed incremental level 0 datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00004 name=/u02/oradata/ORCL/users01.dbf
channel ORA_DISK_1: starting piece 1 at 13-OCT-09
channel ORA_DISK_1: finished piece 1 at 13-OCT-09
piece handle=/u02/flash_recovery_area/ORCL/backupset/2009_10_13/o1_mf_nnnd0_TAG200910:
NE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-OCT-09

RMAN> backup incremental level 1 as compressed backupset tablespace USERS;

Starting backup at 13-OCT-09
using channel ORA_DISK_1
channel ORA_DISK_1: starting compressed incremental level 1 datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00004 name=/u02/oradata/ORCL/users01.dbf
channel ORA_DISK_1: starting piece 1 at 13-OCT-09
channel ORA_DISK_1: finished piece 1 at 13-OCT-09
piece handle=/u02/flash_recovery_area/ORCL/backupset/2009_10_13/o1_mf_nnnd1_TAG200910:
NE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-OCT-09

RMAN> backup incremental level 1 cumulative as compressed backupset tablespace USERS;

Starting backup at 13-OCT-09
using channel ORA_DISK_1
channel ORA_DISK_1: starting compressed incremental level 1 datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00004 name=/u02/oradata/ORCL/users01.dbf
channel ORA_DISK_1: starting piece 1 at 13-OCT-09
channel ORA_DISK_1: finished piece 1 at 13-OCT-09
piece handle=/u02/flash_recovery_area/ORCL/backupset/2009_10_13/o1_mf_nnnd1_TAG200910:
NE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-OCT-09
```

Habilitando Block Change Tracking

Está opción permite habilitar un nuevo background process (**CTWR**) que guardará en un archivo plano la dirección de los bloques modificados desde el último backup, con la intención de cuando se haga un backup se lea ese file y sea más rápido el backup.

Habilitar:

```
alter database (enable-disable) block change tracking using file 'ruta';
```

Validando en la vista V\$BLOCK_CHANGE_TRACKING.

Ejemplo:

```
SQL> alter database enable block change tracking using file '/u03/bct.dat';
```

```
Database altered.
```

```
SQL> select * from v$block_change_tracking;
```

```
STATUS
```

```
-----
```

```
FILENAME
```

```
-----
```

```
      BYTES
```

```
-----
```

```
ENABLED
```

```
/u03/bct.dat
```

```
11599872
```

Nota:

El Block Change Tracking File solo guarda hasta 7 backups incrementales., después del 7mo backup incremental se sobrescribo el file.

Si sacamos un backup incremental L1 y no contamos con el backup L0 el backup no fallará, sino el L1 se convertirá en L0.

Resolviendo problemas con el crosscheck RMAN

RMAN tiene un catalogo que tiene mapeado los archive logs y backups presentes en disco., si uno de los backups o archivers logs han sido eliminados por herramientas del S.O RMAN devolverá error en ciertas actividades.

Ejemplo: Si borramos un archive a nivel de S.O y sacamos un backup de archives (backup archive log all), RMAN tratará de sacar backup al archive log que nos borramos por lo tanto fallará el backup.

Para estos escenarios debemos refrescar lo que RMAN visualiza y actualizarle su catalogo respecto lo que está actualmente en disco.

Existen 3 modalidades de crosscheck:

- CROSSCHECK BACKUP
- CROSSCHECK COPY
- CROSSCHECK ARCHIVELOG ALL

Eliminando Backups

delete all obsolete = Elimina aquellos backups que son más viejos lo que la retención permite.

delete all expired = Elimina del catalogo aquellos archivos y backups que ya no están disponibles, detectados por el crosscheck. (DELETE EXPIRED ARCHIVELOG ALL / DELETE EXPIRED BACKUP)

delete copy = Elimina todos los backups de tipo image.

delete backupset = Elimina todos los backups de tipo backupset.

Reportes de RMAN

Para listar todos los backups ejecutados.

Lista de Backups Copy: RMAN> list copy *summary*;

Lista de Backups Set: RMAN> list backupset | backup *summary*;

Más reportes: RMAN> list backupset | backup | copy of tablespace <tablespace>;

RMAN> list backupset | backup | copy by file;

RMAN> report schema; (Devuelve la estructura de la base de datos).

Reporte de Backups obsoletos: RMAN> report obsolete;

Lista de Backups Copy: RMAN> list copy;

Schedule de Backups

1.

Lo primero que debemos hacer antes de realizar un schedule de un backup es contar con un script de tipo RMAN que contenga la acción a realizar.

Ejemplo de un script:

```
run{  
  
  allocate channel c1 type disk format '/u03/backup_full.bk';  
  
  backup as compressed backupset database;  
  
}
```

Lo guardamos en un archivo llamado BACKUP_FULL.rman

2.

Luego:

Creamos un archivo de extensión sh (Unix) ó cmd (Windows), con el siguiente contenido:

```
rman target / cmdfile=BACKUP_FULL.rman log=BACKUP_FULL.log
```

3.

Finalmente lo programamos en el schedule de tareas de Windows o en el cron en sistemas Unix.

- crontab -e
- Ingresar los siguiente: 0 0 * * * sh BACKUP_FULL.sh

Nota: En caso de los ambientes Unix debemos hacer que el sh también cargue las variables de ambiente ORACLE_BASE, ORACLE_HOME, ORACLE_SID y PATH.

Demo configuración del Recovery Catalog

RMAN Recovery Catalogo es un esquema que se ubica en un Oracle Database con la finalidad de ser el repositorio para los backups.

Nos permite:

- Almacenar scripts y llamarlo desde las BD.
- Ser un punto focal de todos los backups.
- Es el único modo de tener backups keep forever o por periodo más grande que un año.

Pasos:

- **Crear un usuario y un tablespace especial para el Catalogo en una base de datos nueva.**

```
create tablespace tbs_rman datafile '/u02/ORCL/tbs_rman.dbf' size 200M;  
  
create user rman identified by oracle default tablespace tbs_rman quota unlimited on  
tbs_rman;
```
- **Garantizamos el rol RECOVERY_CATALOG en la base de datos que servirá como Catalogo:**

```
grant recovery_catalog_owner to rman;
```
- **Nos conectamos al RMAN en la base de datos que servirá como Catalogo:**

```
rman catalog=rman/oracle;  
  
rman> create catalog;
```
- Nos conectamos al Catalogo desde nuestras OTRAS Bases de Datos.

```
rman / catalog=user/pass@conexion
```
- Registramos la Base de Datos en el Catalogo:

```
rman> register database;
```

Para retirar una base de datos del Catálogo:

```
rman> unregister database;
```

Algunos trabajos extras:

- a) Manipulación de scripts centralizados:

```
create global script nombre { .... }
```

Otras opciones:

```
replace global script nombre { ... }
```

```
delete global script nombre
```

```
print global script nombre;
```

```
list script names;
```

Para ejecutarlo:

```
run {execute script nombre;}
```

- b) Registrar un backup que no ha sido registrado en el Catalogo y se desea registrar en el Catalogo.

```
rman> catalog datafilecopy '/.....dbf';
```

ó

```
rman> catalog backupiece '/...bkp';
```

ó

```
rman> catalog archivelog '...arc','...arc','...arc';
```

- c) Sincronizando el control file con el Catálogo de modo manual:

Durante una operación de RMAN, el catalogo será sincronizado con el control file.

Nuevos registros serán actualizados o agregados durante este proceso de sincronización.

Podemos sincronizarlo manualmente:

```
rman> resync catalog;
```

Puede ayudarnos en solucionar problemas de performance.

Revisando las 10 mejores prácticas de Oracle sobre Backups

Las 10 mejores prácticas recomendado por Oracle son las siguientes:

1. Habilitar Block Checking (Verifica si un data block está corrupto al hacer una operación de I/O).

```
alter system set db_block_checking=true;
```

2. Habilitar el feature block change tracking para mejorar el tiempo de los backups incrementales.
3. Multiplexar los archiver logs y tenerlos en diferentes filesystems.
4. Backuppear con la opción de revisión lógica (Validará RMAN cada backup que realiza backup, es el mejor método de saber que tenemos un backup bueno).

```
backup check logical database;
```

5. Validar nuestros backups de vez en cuando con la opción *validate*:

```
restore validate database; (Ojo: Esto no restaura, solo hace un test).
```

6. Tener backupsets con 1 file (Con esto disminuye el tiempo de restore ya que RMAN no leerá todo el backupset que tendrá una serie de files para traer el file solicitado).

```
backup database filesperset 1;
```

7. Mantenimiento al Catalogo.

Ejecutar periódicamente operaciones de mantenimiento como:

```
7.1 delete obsolete;
```

```
7.2 crosscheck backup;
```

```
delete expired backup;
```

Sino el catalogo crecerá hasta consumir una gran cantidad de disco.

8. Habilitar la opción de *autoback controlfile*.
9. Debemos realizar pruebas de *recover* y saber hasta que punto podemos llegar:

```
rman> recover database test;
```

10. No utilizar la opción *delete all input* cuando realicemos backups.

Recover

Recuperando Base de Datos en modo NO ARCHIVE

Una base de datos en modo NO ARCHIVE solo puede ser recuperada hasta el último backup offline ejecutado.

Debemos restaurar TODA la base de datos y levantarla.

Recuperando Base de Datos en modo ARCHIVE

Perdiendo datafiles de tablespaces no críticos.

1. Colocar el datafile perdido o dañado en modo offline
2. Restaurar el datafile y hacer recover.
3. Colocar el datafile en modo online.

```
[oracle@pcoracle ORCL]$ rm users01.dbf  
[oracle@pcoracle ORCL]$ sqlplus / as sysdba
```

```
SQL*Plus: Release 10.2.0.1.0 - Production on Tue Oct 13 14:57:55 2009
```

```
Copyright (c) 1982, 2005, Oracle. All rights reserved.
```

```
Connected to:
```

```
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production  
With the Partitioning, OLAP and Data Mining options
```

```
SQL> alter database datafile '/u02/oradata/ORCL/users01.dbf' offline;
```

```
Database altered.
```

```
SQL> !
```

```
[oracle@pcoracle ORCL]$ rman target /
```

```
Recovery Manager: Release 10.2.0.1.0 - Production on Tue Oct 13 14:58:07 2009
```

```
Copyright (c) 1982, 2005, Oracle. All rights reserved.
```

```
connected to target database: ORCL (DBID=1217299163)
```

```
RMAN> restore datafile '/u02/oradata/ORCL/users01.dbf';
```

```
RMAN> recover datafile '/u02/oradata/ORCL/users01.dbf';

Starting recover at 13-OCT-09
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 13-OCT-09

RMAN> exit

Recovery Manager complete.
[oracle@pcoracle ORCL]$ exit
exit

SQL> alter database datafile '/u02/oradata/ORCL/users01.dbf' online;

Database altered.
```

Perdiendo datafiles de tablespaces críticos.

Los tablespaces críticos son: SYSTEM y UNDO.

1. Bajar la base de datos.
2. Subir la base de datos hasta modo MOUNT.
3. Colocar el datafile perdido o dañado en modo offline
4. Restaurar el datafile, hacer recover y colocar el datafile en modo online.
5. Abrir la base de datos.

```
[oracle@pcoracle ORCL]$ rm system.dbf  
[oracle@pcoracle ORCL]$ sqlplus / as sysdba
```

SQL*Plus: Release 10.2.0.1.0 - Production on Tue Oct 13 15:04:19 2009

Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

```
SQL> shutdown abort;  
ORACLE instance shut down.
```

```
SQL> startup mount  
ORACLE instance started.
```

```
Total System Global Area 184549376 bytes  
Fixed Size 1218412 bytes  
Variable Size 79693972 bytes  
Database Buffers 100663296 bytes  
Redo Buffers 2973696 bytes  
Database mounted.
```

```

SQL> alter database datafile '/u02/oradata/ORCL/system.dbf' offline;

Database altered.

SQL> !
[oracle@pcoracle ORCL]$ rman target /

Recovery Manager: Release 10.2.0.1.0 - Production on Tue Oct 13 15:07:37 2009

Copyright (c) 1982, 2005, Oracle. All rights reserved.

connected to target database: ORCL (DBID=1217299163, not open)

RMAN> restore datafile '/u02/oradata/ORCL/system.dbf';

Starting restore at 13-OCT-09
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=155 devtype=DISK

channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to /u02/oradata/ORCL/system.dbf
channel ORA_DISK_1: reading from backup piece /u02/flash_recovery_area/ORCL/ba
channel ORA_DISK_1: restored backup piece 1
piece handle=/u02/flash_recovery_area/ORCL/backupset/2009_10_13/o1_mf_nnndf_TA
channel ORA_DISK_1: restore complete, elapsed time: 00:00:25
Finished restore at 13-OCT-09

RMAN> exit

Recovery Manager complete.
[oracle@pcoracle ORCL]$ exit
exit

SQL> recover datafile '/u02/oradata/ORCL/system.dbf';
Media recovery complete.
SQL> alter database datafile '/u02/oradata/ORCL/system.dbf' online;

Database altered.

SQL> alter database open;

```

Perdiendo temp files.

En el caso de los temp files no es crítico, podemos crear otros tempfiles al Tablespace y se arregla el problema.

En caso la base de datos esté abajo y no existe los temp files, el mismo Oracle los recrea automáticamente.

Nota:

Si deseamos restaurar archive logs debemos ejecutar los siguientes comandos:

```
RESTORE ARCHIVELOG SEQUENCE n
```

```
RESTORE ARCHIVELOG FROM SEQUENCE n UNTIL SEQUENCE m;
```

Incomplete Recovery

Un incomplete recovery es recuperar nuestra base de datos hasta un punto en el tiempo que nosotros indiquemos.

Podemos recuperar nuestra base de datos por 4 medidas: time, scn, log sequence y cancel.

Paso 1: Bajar la base de datos y subirla en modo MOUNT.

Paso 2: Restaurar en el tiempo.

```
RMAN> RESTORE DATABASE UNTIL TIME "TO_DATE('10/15/2009 13:00','MM/DD/YYYY HH24:MI');
```

Ó

```
RMAN> RESTORE DATABASE UNTIL TIME "sysdate-1/24*60";
```

Existe:

UNTIL TIME, UNTIL SCN, UNTIL LOG SEQUENCE, UNTIL CANCEL

Pasos:

Paso 3: Hacer recover en el tiempo

```
RMAN> RECOVER DATABASE UNTIL TIME "TO_DATE('10/15/2009 13:00','MM/DD/YYYY HH24:MI');
```

Ó

```
RMAN> RECOVER DATABASE UNTIL TIME "sysdate-1/24*60";
```

Paso 4: Subir la base de datos en OPEN NO RESETLOGS.

```
SQL> ALTER DATABASE OPEN RESETLOGS
```

Nota:

OPEN RESETLOGS generará una nueva encarnación en la base de datos, lo cual significa que la secuencia de archive logs inician en 1, los online redo los son limpiados y los backups pre existentes ya no sirven para este nuevo ciclo de vida de la base de datos.

Recuperación de una base de datos utilizando un backup image

La restauración de un image copy puede realizarse con RMAN como se ve en el siguiente ejemplo:

```
RMAN> restore datafile '/u02/oradata/ORCL/users01.dbf';

Starting restore at 13-OCT-09
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=159 devtype=DISK

channel ORA_DISK_1: restoring datafile 00004
input datafile copy recid=20 stamp=700154768 filename=/u02/flash_recovery_
destination for restore of datafile 00004: /u02/oradata/ORCL/users01.dbf
channel ORA_DISK_1: copied datafile copy of datafile 00004
output filename=/u02/oradata/ORCL/users01.dbf recid=22 stamp=700154853
Finished restore at 13-OCT-09
```

```
RMAN> recover datafile '/u02/oradata/ORCL/users01.dbf';

Starting recover at 13-OCT-09
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:02

Finished recover at 13-OCT-09
```

También se puede restaurar dirigiéndonos donde se ubica el backup y copiarlo mediante comandos de S.O y luego hacer el restore.

Otra técnica más avanzada en caso de una pérdida de un datafile es indicar a la base de datos que haga un switch del datafile dañado al backup image.

Esto lo podemos realizar con el comando: switch datafile # to copy;

```
[oracle@pcoracle ORCL]$ rman target /

Recovery Manager: Release 10.2.0.1.0 - Production on Tue Oct 13 15:36:34 2009
Copyright (c) 1982, 2005, Oracle. All rights reserved.

connected to target database: ORCL (DBID=1217299163)

RMAN> switch datafile '/u02/oradata/ORCL/users01.dbf' to copy;

using target database control file instead of recovery catalog
datafile 4 switched to datafile copy "/u02/flash_recovery_area/ORCL/datafile/01
```



```
RMAN> recover datafile '/u02/flash_recovery_area/ORCL/datafile/o1_mf_users_5f9rm03v_.dbf';

Starting recover at 13-OCT-09
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 13-OCT-09

RMAN> exit

Recovery Manager complete.
[oracle@pcoracle ORCL]$ sqlplus / as sysdba

SQL*Plus: Release 10.2.0.1.0 - Production on Tue Oct 13 15:37:18 2009

Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> alter database datafile '/u02/flash_recovery_area/ORCL/datafile/o1_mf_users_5f9rm03v_.dbf'

Database altered.
```

Recovery data block

Un data block se puede poner en modo corrupto por corrupción de memoria escrita a disco o problemas en I/O.

Frente a esto cuando Oracle trata de leer un data block corrupto devolverá un error y lo escribirá en el alert.log el data block dañado y también los registra en la tabla **v\$database_block_corruption** automáticamente.

RMAN desde la versión 9i nos permite repara bloques corruptos en un datafile incluso en modo online. Este feature solo disponible en la versión Enterprise Edition.

Pasos:

Si encontramos un bloque dañado podemos ejecutar la recuperación en modo MOUNT u OPEN.

En Oracle 9i/10g

```
RMAN> blockrecover corruption list;
```

En Oracle 11g

```
RMAN> recover corruption list;
```

Con corruption list recuperará todos los bloques identificados en la table v\$database_block_corruption.

Si solo queremos restaurar un bloque determinado:

En Oracle 9i/10g

```
blockrecover datafile 3 block 3;
```

En Oracle 11g

```
recover datafile 3 block 3;
```

Si deseamos saber cual segment pertenece el bloque podemos ejecutar este query:

```
select count(*), e.segment_type
FROM dba_extents e, v$database_block_corruption c
WHERE e.file_id = c.FILE# and c.BLOCK# BETWEEN e.block_id AND e.block_id + e.blocks - 1
group by e.segment_type;
```

Validando nuestra base de datos de bloques corruptos:

- RMAN> BACKUP CHECK LOGICAL VALIDATE DATABASE; (No genera un backup y finalizado la revisión el resultado lo podemos encontrar en v\$database_block_corruption)
- Utilizando DBV: dbv file=/u02/oradata/ORCL/system.dbf blocksize=8192
- Analizando los objetos: SQL> analyze table nombre validate structure cascade (finalizado la revisión el resultado lo podemos encontrar en v\$database_block_corruption)

Nota:

El recover de data blocks fallará si el bloque físico corrupto está en un redo log file.

Recuperación del control file (Recreación y restauración)

Recreación del control file.

Requisito: Debemos contar un backup del control file del tipo:

alter database backup controlfile to trace as '/tmp/backup_controlfile.sql';

```
SQL> alter database backup controlfile to trace as  '/tmp/backup_controlfile.sql';  
Database altered.
```

Abrimos el archive backup_controlfile.sql

El script tiene 2 secciones:

Una que tiene la sentencia NORESETLOGS y la debemos aplicar cuando tengamos los online redo logs disponibles y la otra sección es RESETLOGS y la debemos usar cuando no tengamos los online redo logs disponibles.

En el ejemplo tenemos los online redo logs entonces se hará con la sección NORESETLOGS.

```
--      Set #1. NORESETLOGS case  
--  
-- The following commands will create a new control file and use it  
-- to open the database.  
-- Data used by Recovery Manager will be lost.  
-- Additional logs may be required for media recovery of offline  
-- Use this only if the current versions of all online logs are  
-- available.  
  
-- After mounting the created controlfile, the following SQL  
-- statement will place the database in the appropriate  
-- protection mode:  
-- ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE  
  
STARTUP NOMOUNT  
CREATE CONTROLFILE REUSE DATABASE "ORCL" NORESETLOGS  ARCHIVELOG  
  MAXLOGFILES 16  
  MAXLOGMEMBERS 3  
  MAXDATAFILES 100  
  MAXINSTANCES 8  
  MAXLOGHISTORY 292  
LOGFILE  
  GROUP 1 '/u02/oradata/ORCL/redo01.log'  SIZE 50M,  
  GROUP 2 '/u02/oradata/ORCL/redo02.log'  SIZE 50M,  
  GROUP 3 '/u02/oradata/ORCL/redo03.log'  SIZE 50M  
-- STANDBY LOGFILE  
  
DATAFILE  
  '/u02/oradata/ORCL/system.dbf',  
  '/u02/oradata/ORCL/undotbs01.dbf',  
  '/u02/oradata/ORCL/sysaux01.dbf',  
  '/u02/flash_recovery_area/ORCL/datafile/o1_mf_users_5f9rm03v_.dbf'  
CHARACTER SET WE8ISO8859P1  
;  
  
-- Configure RMAN configuration record 1  
VARIABLE RECNO NUMBER;  
EXECUTE :RECNO := SYS.DBMS_BACKUP_RESTORE.SETCONFIG('RETENTION POLICY','TO RECOVERY WINDOW OF 2 DAYS');  
-- Configure RMAN configuration record 2  
VARIABLE RECNO NUMBER;  
EXECUTE :RECNO := SYS.DBMS_BACKUP_RESTORE.SETCONFIG('ENCRYPTION FOR DATABASE','OFF');  
-- Configure RMAN configuration record 3  
VARIABLE RECNO NUMBER;  
EXECUTE :RECNO := SYS.DBMS_BACKUP_RESTORE.SETCONFIG('ENCRYPTION ALGORITHM','AES128');
```

Entonces:

Connected to an idle instance.

```
SQL> STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "ORCL" NORESETLOGS ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 100
    MAXINSTANCES 8
    MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/u02/oradata/ORCL/redo01.log' SIZE 50M,
  GROUP 2 '/u02/oradata/ORCL/redo02.log' SIZE 50M,
  GROUP 3 '/u02/oradata/ORCL/redo03.log' SIZE 50M
-- STANDBY LOGFILE
DATAFILE
  '/u02/oradata/ORCL/system.dbf',
  '/u02/oradata/ORCL/undotbs01.dbf',
  '/u02/oradata/ORCL/sysaux01.dbf',
  '/u02/flash_recovery_area/ORCL/datafile/o1_mf_users_5f9rm03v_.dbf'
CHARACTER SET WE8ISO8859P1
;

-- Configure RMAN configuration record 1
VARIABLE RECNO NUMBER;
EXECUTE :RECNO := SYS.DBMS_BACKUP_RESTORE.SETCONFIG('RETENTION POLICY','TO RECOVERY WINDOW OF 2 DAYS');
-- Configure RMAN configuration record 2
VARIABLE RECNO NUMBER;
EXECUTE :RECNO := SYS.DBMS_BACKUP_RESTORE.SETCONFIG('ENCRYPTION FOR DATABASE','OFF');
-- Configure RMAN configuration record 3
VARIABLE RECNO NUMBER;
EXECUTE :RECNO := SYS.DBMS_BACKUP_RESTORE.SETCONFIG('ENCRYPTION ALGORITHM','AES128');
-- Configure RMAN configuration record 5
VARIABLE RECNO NUMBER;
EXECUTE :RECNO := SYS.DBMS_BACKUP_RESTORE.SETCONFIG('DEFAULT DEVICE TYPE TO','DISK');
-- Configure RMAN configuration record 6
VARIABLE RECNO NUMBER;
EXECUTE :RECNO := SYS.DBMS_BACKUP_RESTORE.SETCONFIG('ARCHIVELOG DELETION POLICY','TO NONE');
-- Configure RMAN configuration record 8
VARIABLE RECNO NUMBER;
EXECUTE :RECNO := SYS.DBMS_BACKUP_RESTORE.SETCONFIG('BACKUP OPTIMIZATION','ON');
-- Commands to re-create incarnation table
-- Below log names MUST be changed to existing filenames on
-- disk. Any one log file from each branch can be used to
-- re-create incarnation records.
-- ALTER DATABASE REGISTER LOGFILE '/u03/1_1_689518427.dbf';
-- ALTER DATABASE REGISTER LOGFILE '/u03/1_1_689523803.dbf';
```

Con esto se reconstruye el control file.

Restauración del control file.

Debemos contar con una copia del control file.

Ejemplo: SQL> alter database backup controlfile to '/tmp/control.ct!';

Si perdemos **todos los controlfiles**, entonces procedemos a bajar la base de datos con:

```
SQL> shutdown abort;
```

Luego copiamos el backup del controlfile en las ubicaciones de los controlfiles perdidos y levantamos la base de datos pero con **OPEN RESETLOGS**.

Recuperación de una Base de Datos habiendo perdido todos los files con la mínima pérdida de información

Requisito: Debemos contar con un backup completo con los archivers y el online redo log;

Fases:

- 1) Subimos la base de datos en estado nomount (dummy)
- 2) Recuperamos el spfile y levantamos en modo nomount.
- 3) Recuperamos el controlfile del autobackup y levantamos en modo mount.
- 4) Restauramos la base de datos.
- 5) Hacemos recover.
- 6) Levantamos la base de datos si tenemos los online redo logs sino levantamos en modo RESETLOGS.

Sacando un Backup completo a la base de datos.

RMAN> backup as compressed backupset database plus archivelog;

Bajamos la base de datos y subimos en modo nomount (dummy)

```
[oracle@pcoracle dbs]$ rman target /  
Recovery Manager: Release 10.2.0.1.0 - Production on Sat Sep 26 18:00:32 2009  
Copyright (c) 1982, 2005, Oracle. All rights reserved.  
connected to target database (not started)  
RMAN> startup nomount force;  
startup failed: ORA-01078: failure in processing system parameters  
LRM-00109: could not open parameter file '/u01/app/oracle/db/10.2/dbs/initORCL.ora'  
starting Oracle instance without parameter file for retrieval of spfile  
Oracle instance started  
  
Total System Global Area      159383552 bytes  
  
Fixed Size                      1218268 bytes  
Variable Size                  54528292 bytes  
Database Buffers              100663296 bytes  
Redo Buffers                    2973696 bytes
```

Restauramos el SPFILE.

RMAN> restore spfile from 'ruta'

```
RMAN> restore spfile from '/u02/flash_recovery_area/ORCL/autobackup/2009_09_26/o1_mf_s_698608651_5cx72wqx_.bkp';  
Starting restore at 26-SEP-09  
using channel ORA_DISK_1  
  
channel ORA_DISK_1: autobackup found: /u02/flash_recovery_area/ORCL/autobackup/2009_09_26/o1_mf_s_698608651_5cx72wqx_.bkp  
channel ORA_DISK_1: SPFILE restore from autobackup complete  
Finished restore at 26-SEP-09
```

Restauramos el Control File

RMAN> startup nomount forcé; (Para inicializar con el spfile recuperado)

RMAN> restore controlfile from 'ruta';

Este restore controlfile from pudo ser cambiado seteando el DBID:

RMAN> SET DBID=xxx

RMAN> restore controlfile from autobackup;

```
RMAN> startup nomount force;

Oracle instance started

Total System Global Area      184549376 bytes
Fixed Size                    1218412 bytes
Variable Size                 75499668 bytes
Database Buffers             104857600 bytes
Redo Buffers                   2973696 bytes

RMAN> restore controlfile from '/u02/flash_recovery_area/ORCL/autobackup/2009_09_26/o1_mf_s_698608651_5cx72wqx_.bkp';

Starting restore at 26-SEP-09
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=156 devtype=DISK
```

Restauramos la Base de Datos y hacemos Recover

RMAN> sql 'alter database mount';

sql statement: alter database mount
released channel: ORA_DISK_1

RMAN> restore database;

```
Starting restore at 26-SEP-09
Starting implicit crosscheck backup at 26-SEP-09
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=159 devtype=DISK
Crosschecked 3 objects
Finished implicit crosscheck backup at 26-SEP-09

Starting implicit crosscheck copy at 26-SEP-09
using channel ORA_DISK_1
Crosschecked 2 objects
Finished implicit crosscheck copy at 26-SEP-09

searching for all files in the recovery area
cataloging files...
cataloging done

List of Cataloged Files
=====
File Name: /u02/flash_recovery_area/ORCL/autobackup/2009_09_26/o1_mf_s_698608651_5cx72wqx_.bkp
using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to /u02/oradata/ORCL/system.dbf
restoring datafile 00002 to /u02/oradata/ORCL/undotbs01.dbf
restoring datafile 00003 to /u02/oradata/ORCL/sysaux01.dbf
restoring datafile 00004 to /u02/oradata/ORCL/users01.dbf
channel ORA_DISK_1: reading from backup piece /u02/flash_recovery_area/ORCL/backupset/2009_09_2
```

```
RMAN> recover database;
```

```
Starting recover at 26-SEP-09  
using channel ORA_DISK_1
```

```
starting media recovery
```

```
channel ORA_DISK_1: starting archive log restore to default destination  
channel ORA_DISK_1: restoring archive log  
archive log thread=1 sequence=15  
channel ORA_DISK_1: reading from backup piece /u02/flash_recovery_area/c  
channel ORA_DISK_1: restored backup piece 1  
piece handle=/u02/flash_recovery_area/ORCL/backupset/2009_09_26/o1_mf_ar  
channel ORA_DISK_1: restore complete, elapsed time: 00:00:02  
archive log filename=/u02/flash_recovery_area/ORCL/archivelog/2009_09_26  
channel default: deleting archive log(s)  
archive log filename=/u02/flash_recovery_area/ORCL/archivelog/2009_09_26  
unable to find archive log
```

Levantamos la Base de Datos en RESETLOGS

No contamos con los online redo logs por lo tanto aperturamos con RESETLOGS.

```
[oracle@pcoracle dbs]$ rman target /
```

```
Recovery Manager: Release 10.2.0.1.0 - Production on Sat Sep 26 18:05:42 2009
```

```
Copyright (c) 1982, 2005, Oracle. All rights reserved.
```

```
connected to target database: ORCL (DBID=1217299163, not open)
```

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

```
using target database control file instead of recovery catalog  
sql statement: alter database open resetlogs
```


Tablespace Point-Time Recovery (TSPITR)

Oracle Database desde la versión 8i nos permite llevar en el tiempo todo un Tablespace sin alterar los demás tablespaces y sin bajar servicios.

La estrategia que hace RMAN para conseguir esto es clonar la base de datos en un directorio que indiquemos y esa base de datos clonada le hace un recovery en el tiempo y luego exporta los objetos del tablespace que necesitamos llevar en el tiempo y lo importa en nuestro tablespace. Después de todo este trabajo RMAN procede a eliminar la base de datos clonada.

Pasos:

1. Contar con un backup completo de la base de datos.
2. Crear un directorio vacío.
3. recover tablespace nombre_tablespace
until time TO_DATE(fecha)
auxiliary destination 'ruta';
4. Poner el tablespace en modo online.

Ejemplo:

```
RMAN> recover tablespace USERS until time "TO_DATE('26-09-2009 18:22:13','DD-MM-YYYY HH24:MI:SS')" auxiliary destination '/u03/tbs';

Starting recover at 26-SEP-09
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=159 devtype=DISK
RMAN-05026: WARNING: presuming following set of tablespaces applies to specified point in time

List of tablespaces expected to have UNDO segments
tablespace SYSTEM
tablespace UNDOTBS1

Creating automatic instance, with SID='fbzc'

Removing automatic instance
Automatic instance removed
auxiliary instance file /u03/tbs/cntrl_tspitr_ORCL_fbzc.f deleted
auxiliary instance file /u03/tbs/TSPITR_ORCL_FBZC/datafile/o1_mf_system_5cx8mtm2_.dbf deleted
auxiliary instance file /u03/tbs/TSPITR_ORCL_FBZC/datafile/o1_mf_undotbs1_5cx8mtog_.dbf deleted
auxiliary instance file /u03/tbs/TSPITR_ORCL_FBZC/datafile/o1_mf_temp_5cx8oxjz_.tmp deleted
auxiliary instance file /u03/tbs/TSPITR_ORCL_FBZC/onlinelog/o1_mf_1_5cx8or7s_.log deleted
auxiliary instance file /u03/tbs/TSPITR_ORCL_FBZC/onlinelog/o1_mf_2_5cx8ortz_.log deleted
auxiliary instance file /u03/tbs/TSPITR_ORCL_FBZC/onlinelog/o1_mf_3_5cx8osly_.log deleted
Finished recover at 26-SEP-09

RMAN> sql 'alter tablespace users online';

sql statement: alter tablespace users online
```

Si deseamos saber que objetos se borrarán después del recovery en el tiempo consultamos la vista:

TS_PITR_OBJECTS_TO_BE_DROPPER (Filtrar por el campo
CREATION_TIME>Tiempo_Recovery_Incomplete)

Restricciones:

1. No podemos hacer recovery en el tiempo a un tablespace donde tenga objetos que dependan de otros y estos se ubiquen en otro tablespace. En este escenario hay que mover el tablespace y los tablespaces dependientes.

Para saber de antemano la dependencia debemos consultar la vista: **SYS.TS_PITR_CHECK**

2. Si renombramos un tablespace ya no podemos retroceder en el tiempo a menos que saquemos otro backup y ese sería nuestro punto de partida.
3. Tablespaces que contienen los siguientes objetos no pueden aplicársele el Recovery Incomplete: Tablas master de replicación, particiones de tablas particionadas (debe ser toda la tabla y sus particiones), tablas externas, tablas con tipo de datos VARRAY y Nested Table, tablespaces con undo o rollback segments y objetos del owner SYS.