TEMA 8
Respaldos y Recuperación

8.1. IMPORTANCIA DE LA CREACIÓN DE RESPALDOS.

- En general, el objetivo del proceso de respaldos y recuperación (backup and Recovery) es proteger a la base de datos ante una pérdida de datos así como la capacidad de reconstruir una base de datos posterior a la ocurrencia de la falla.
- Este proceso incluye las siguientes actividades.
    - Planear y probar las respuestas y acciones a seguir ante la presencia de algún tipo de falla.
    - Configurar la base de datos para realizar tanto respaldos como procesos de recuperación.
    - Planear una programación de backups.
    - Monitoreo de backups
    - Resolución de problemas asociados con backups.
    - Recuperar una base de datos ante una pérdida de datos.
    - Archivado de datos – Mover copias de la base de datos a medios de almacenamiento permanentes por periodos de tiempo largos.
    - Transmisión de datos hacia bases de datos remotas.
- La realización de las actividades anteriores, le permite a un administrador cumplir con los siguientes puntos:
    - Incrementar el llamado MTBF: Mean Time Between Failures
    - Decrementar el llamado MTTR: Mean time To Recover

8.1.1. Backups.

- Representa una copia de la base de datos que puede ser empleada para reconstruir el estado de una base de datos en un instante del tiempo. Un backup se clasifica en 2 tipos:

    - Backup físico
    - Backup lógico

8.1.1.1. Backups lógicos

- Contienen datos lógicos, como son: tablas, procedimientos almacenados, etc.
- Para realizar este tipo de backups se emplean herramientas como Oracle Data Pump, empleada para crear archivos binarios a partir de datos lógicos (***export***) los cuales pueden ser ***importados*** en la BD.
- Oracle data pump hace uso de los comandos `impdp` y `expdp` p para realizar operaciones de import y export.
- Para realizar una recuperación, un backup lógico generalmente no es suficiente. Se requiere de uno físico.

8.1.1.2. Backups físicos

- Son copias de los archivos físicos de la base de datos que son empleados para guardar y recuperar una BD: data files, control files, archive Redo Logs.
- Representan a un elemento fundamental para realizar una recuperación de la base de datos.

Por lo anterior, típicamente el término backup se emplea para referirse únicamente a backups ***físicos*** administrados por una herramienta llamada RMAN. (Recovery Manager).

8.1.1.3. Herramientas para realizar backups y operaciones de recuperación.

- Recovery Manager (RMAN).

Herramienta integrada con la base de datos empleada para realizar diversos tipos de backups así como diversas actividades asociadas con recuperación. Se encarga de administrar el llamado ***RMAN repository*** el cual contiene datos históricos acerca de un backup. RMAN puede ser empleado a línea de comandos o de forma gráfica a través de Oracle Enterprise Manager.

RMAN es una de las herramientas más utilizadas para la generación de Backups. Adicionalmente existen otras soluciones como:

- Oracle Manager Enterprise Cloud Control
- Zero Data Loss Recovery Appliance (Recovery Appliance)
- User-managed backup and recovery. Combinación de comandos del Sistema operativo y comandos de SQL *Plus.

8.2. ARQUITECTURA DE RMAN.

El entorno de RMAN está formado por un conjunto de aplicaciones y bases de datos que juegan roles específicos como parte de la estrategia de respaldos y recuperación de una base de datos. La siguiente tabla muestra sus principales componentes.

Componente	Descripción

Cliente RMAN	Aplicación que se ejecuta del lado de la maquina cliente encargada de realizar la administración de respaldos y procesos de recuperación de una base de datos. El cliente RMAN puede conectarse de forma remota a la base de datos que se desea respaldar.
	Para efectos del curso se empleará la interfaz a línea de comandos `rman`.
	`$ rman`
	`RMAN >`

Target Database	Base de datos que contiene todos los archivos que se desean respaldar: archivos de control, archivos de parámetros, data files, archive redo logs. • RMAN hace uso del archivo de control de la *target database* para obtener información de ella. • En el archivo de control se almacena la información de las operaciones que realiza RMAN.
Recovery catalog database	• Contiene metadatos que emplea RMAN para realizar backups y operaciones de recovery. • Es posible crear un *recovery catalog database* para almacenar datos de varias *target databases.*

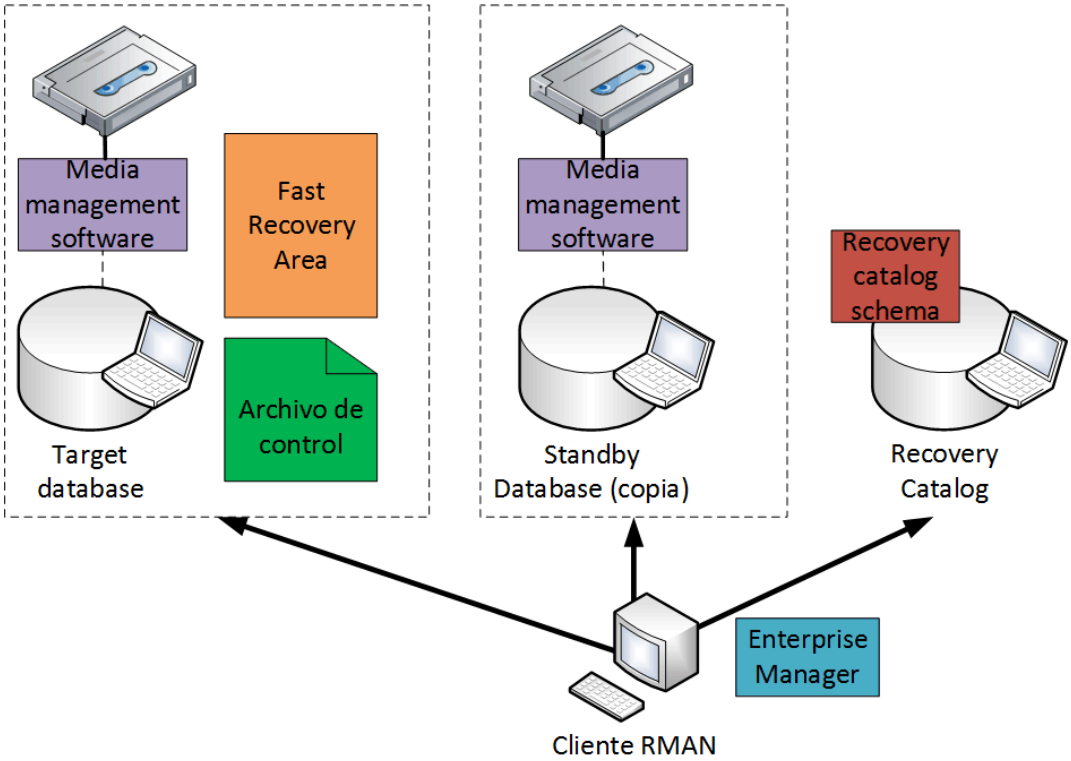
Componente	Descripción
	• Este componente es opcional ya que estos metadatos se almacenan por default en el archivo de control sin embargo, si se hace uso de una *base de datos standby* (se revisa más adelante), este componente será obligatorio
Recovery catalog schema	• Se refiere al usuario dueño de las tablas que emplea RMAN para almacenar sus metadatos dentro del *recovery catalog.*
Physical standby database	• Copia de una base de datos primaria que es actualizada con datos Redo generados por la base de datos primaria. • Es posible hacer un *switch* a esta base de datos en caso de falla ocurrida en la base de datos primaria. • RMAN puede realizar backups, o procesos de recovery sobre una base de datos *standby* • Los respaldos que se hacen sobre esta base de datos *standby* pueden ser empleados para recuperar una base de datos primaria. • En este caso, el *recovery catalog* es requerido para poder realizar respaldos de la base de datos *standby.*
Fast recovery area	Ubicación o directorio en disco empleado para almacenar todos los archivos que produce RMAN: • Backups de todos los archivos antes mencionados (control files, archivos de parámetros, data files, archive redo logs). • Backups de backups • Flashback logs (se revisa más adelante). La administración de esta área se realiza de forma automática por lo que se recomienda ampliamente su uso (se revisa más adelante).
Media management Software	Se refiere al software (utilidades, drivers, etc.) que le permiten a RMAN acceder a otros medios de almacenamiento, por ejemplo, ***cintas.***
Enterprise manager	Interface Web que permite realizar la administración de respaldos de forma gráfica.

Como mínimo, los únicos 2 componentes obligatorios que requiere RMAN para trabajar son:

- Cliente RMAN
- Target Database.

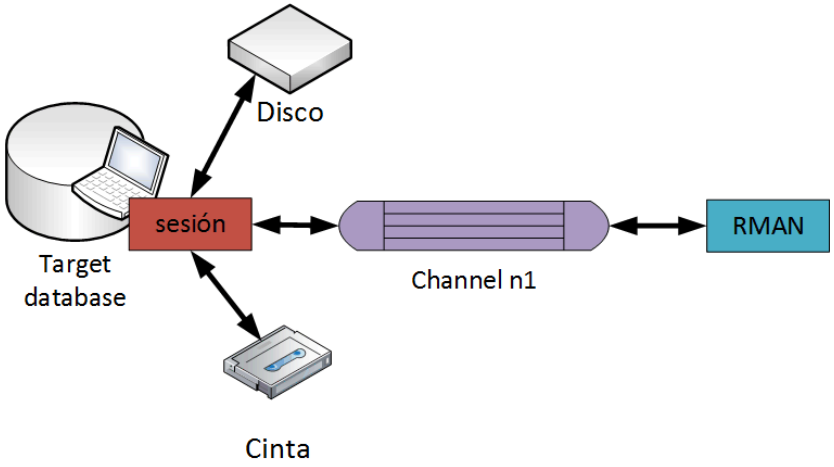
Sin embargo, en sistemas productivos la configuración es mucho más complicada.

En la siguiente imagen se describe la arquitectura de RMAN. Notar la existencia de una base de datos standby o copia, así como la existencia de una BD que actúa como repositorio externo para almacenar los metadatos que requiere RMAN.



8.2.1. RMAN channels.

- Representan un flujo de datos hacia un dispositivo de almacenamiento (disco o cinta).
- Cada channel es iniciado por una sesión.
- Durante la creación de un backup o de una operación de `restore`, los datos pueden ser leídos o escritos en el dispositivo de almacenamiento a través de uso de un *channel.*
- RMAN soporta 2 tipos de dispositivos:
  - Discos
  - Cintas SBT (System Backup to Type). Este tipo de almacenamiento require hacer uso del llamado *media software management* el cual representa al conjunto de librerías, drivers y utilidades requeridas para poder realizar las operaciones que RMAN requiera.
- Para realizar la configuración de un channel se emplea el comando `configure channel`.

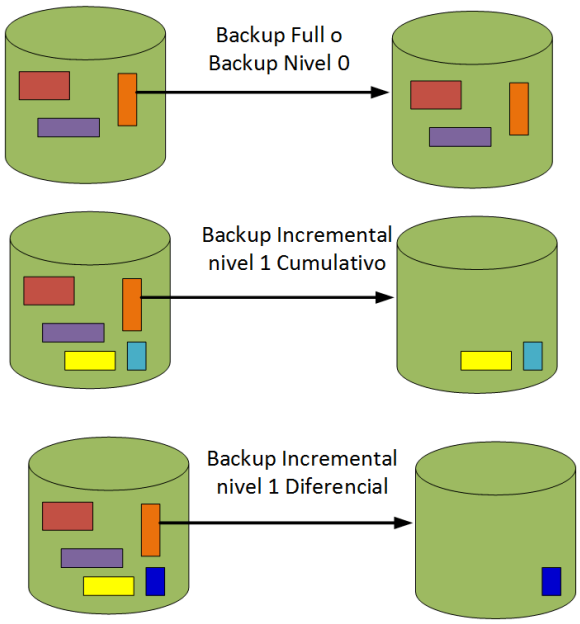


- El Cliente RMAN en realidad no se encarga de realizar el respaldo. El cliente delega esta tarea a un server process.

- La mayoría de los comandos de RMAN requieren hacer uso de un *channel.* Es posible configurarlo de manera global o a nivel de sesión.
- Un *channel* es el encargado de establecer la comunicación entre el cliente RMAN y la *target database* o a una *standby database.*

8.3. CONCEPTOS BÁSICOS DE RMAN.

8.3.1. Términos comúnmente empleados para realizar un backup.



Estrategia del backup

- Backup completo de una base de datos (Whole)
- Backup parcial de una base de datos (Partial solo ciertos tablespaces).

Tipo de backup

- Full backup: Backup de todos los bloques de datos
- Incremental: Respaldar únicamente la información que ha cambiado.
  - Cumulativo: (cambios desde el último nivel 0).
  - Diferencial: (cambios desde el último nivel 0 o 1).

Modo del Backup

- Offline (Consistent, cold)
- Online (Inconsistent, hot)

8.3.2. Backups consistentes e inconsistentes con RMAN

- Se emplea el comando `backup`
- Para ambos tipos, RMAN soporta los siguientes archivos
  - Data files
  - Control files
  - Server parameter file
  - Archived Redo Logs
  - RMAN Backups

8.3.2.1. Backup consistente

- Ocurren cuando la BD está cerrada.
- Los data files deben estar completamente sincronizados. Es decir, la BD debe detenerse en modo `immediate`, `transactional`, o `normal`.
- Si la instancia se inicia en modo `mount` y se hace un backup consistente, la base de datos puede ser restaurada empleando este backup y podrá ser abierta sin realizar media recovery. Esto se debe a que la base fue cerrada de forma ordenada provocando un full checkpoint.
- Ojo: Este backup puede no contener los cambios realizados a la BD posterior a su creación.

8.3.2.2. Backup inconsistente

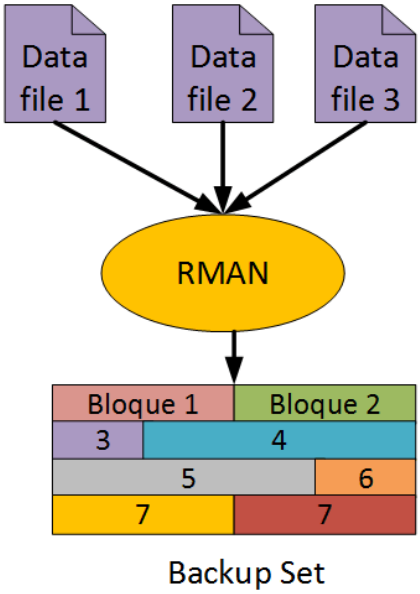
- El backup se realiza cuando la base de datos en modo `open`, o el backup se realiza cuando la base de datos fue detenida con `shutdown abort`.

- Al restaurar la BD con un backup inconsistente, es necesario aplicar media recovery.
- RMAN no permite realizar bakups inconsistentes si la BD está configurada en modo `noarchivelog`.
- De lo anterior, si la BD está en modo archivelog, los archive redo logs y los data files han sido respaldadosm un backup inconsistente puede emplearse para aplicar recovery sin problemas.
- Esta técnica es útil para casos donde una base de datos no puede detenerse para realizar respaldos.

8.3.3. Backup Set e Image Copies

- Por default RMAN realiza backups generando un conjunto de archivos llamados ***backup sets.*** Esos archivos tiene un formato muy particular entendibles únicamente por RMAN.
- Si se especifica el comando `backup as copy`, no se generan *backup sets*. Se realiza una copia del archivo como si fuera una copia con el comando `cp` del sistema operativo. La copia será idéntica a la existente en la BD target.

8.3.3.1. Backup sets



- Los datos del backup se guarda en una estructura lógica llamada *backup set.*
- Un backup set contiene datos de uno o más archivos: data files, control files, redo logs, server parameter file.
- A cada uno de los archivos que contiene el *backup set* se le conoce como ***backup piece.***
- Su formato es binario y solo lo entiende RMAN
- El formato para nombrar a cada uno de los archivos binarios (backup piece) del *backup set* puede ser asignado por RMAN o empleando el parámetro `format` del comando `backup`.
- Por default cada *backup set* contiene un solo *backup piece*. Su tamaño se puede limitar empleando la opción `maxpiecesize` en el parámetro de configuración `configure cannel` o `allocate channel`.
- En un backup set, RMAN lee bloques de varios archivos, por ejemplo, data files de forma simultánea y los escribe en el mismo backup set. A esta técnica se le llama *backup multiplexing*.
- El número de archivos simultáneos que se pueden leer se controla con el parámetro `maxopenfiles`, el default es 8.

8.3.3.2. Image copies.

- A diferencia de un backup set, Image copies son copias idénticas de los archivos de la base de datos. No son almacenadas con formato específico de RMAN.
- Se emplea el comando backup as copy para crear *image copies*.
- Una desventaja que tiene un Image copies sobre un backup set, es que el tamaño del respaldo es mayor. Un backup set ***únicamente*** respalda los bloques que se contienen datos mientras que un image copy respalda el archivo completo.

8.3.4. Generación de múltiples copias de backups.

- RMAN permite la generación de múltiples copias de un backup.
- Existen 2 técnicas para generar copias:

- Duplexed backup Sets
- Backups of backup sets.

8.3.4.1. Duplexed backup sets.

- SI un backup se crea empleando backup sets, RMAN puede crear un ***duplexed backup set***. Esta característica permite crear hasta 4 copias idénticas de cada *backup piece* en ubicaciones diferentes empleando un solo comando `backup`.
- Para configurar el número de copias se emplea el parámetro copies en los comandos `configure`, `set` o `backup`.
- Típicamente estas copias se realizan en cintas.
- El parámetro `format` se emplea para indicar el destino de cada una de las copias.

Ejemplo:

```
backup device type disk copies 3 datafile 7
  format '/disk1/%U','?/oradata/%U','?/%U';
```

- ***Ojo***: En este ejemplo RMAN genera un solo backup set con un solo ID y genera 3 copias idénticas de cada *backup piece* en el mismo backup set.

8.3.4.2. Backups of backup sets

- Como el título lo indica, en esta técnica RMAN realiza un nuevo backup del resultado de haber realizado un backup previo a través de backup sets.
- Para este caso se emplea el comando `backup backupset`.
- Si RMAN detecta que una de las copias del backup no existe o se ha dañado, RMAN busca otra copia en alguna otra ubicación.

Ejemplo:

Generar un respaldo de la base de datos y de los archive redo logs y escribirlos tanto en disco como en cinta.

```
backup device type disk as backupset
  database plus archivelog;
backup
  device type sbt
  backupset all; # copies backup sets on disk to tape
```

8.3.5. Backups del archivo de control y del archivo de parámetros.

- Respaldar tanto los archivos de control como el archivo de parámetro es una tarea importante, en especial cuando se requieren hacer labores de recuperación.
- Para el caso del archivo de control, este es respaldado de forma automática posterior a la ejecución del comando `backup`, incluso si dicho comando se usa para respaldar el actual archivo de control.
- Si le parámetro `configure controlfile autobackup` tiene el valor `on`, RMAN realizará un backup automático del archivo de control y del server parameter file cada vez que se ejecute el comando `backup` de forma exitosa.
- Si la base de datos está en modo `archivelog`, RMAN realiza auto backups de del archivo de control cuando se realiza un cambio estructural a la base de datos que afecte al archivo de control.

8.3.6. Backups incrementales.

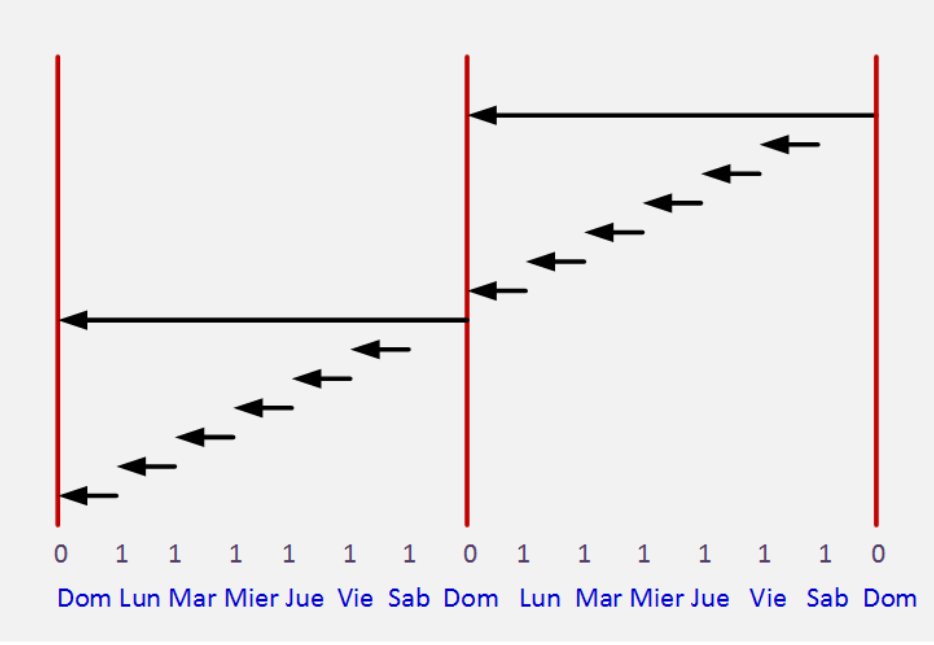
- A nivel general, un backup incremental toma cambios únicamente ocurridos después del último backup.
- Por default RMAN realiza full backups, y un full backup NO puede ser el punto inicial o el backup padre para realizar backups incrementales.
- RMAN puede realizar backups incrementales multi – nivel. Cada nivel es etiquetado como nivel 0 o nivel 1.
- Un backup Incremental nivel 0 representa el punto de partida o backup padre para generar backups subsecuentes. La única diferencia con un full backup es justamente la restricción de no ser considerado como backup de partida para crear backups incrementales.
- Un backup incrementan nivel 1 puede ser de 2 tipos:

    - *Backup Incremental diferencial*
    - *Backup Incremental cumulativo*.

8.3.6.1. Backup Incremental diferencial.

- Realiza el respaldo de todos los bloques que han cambiado desde el último backup incremental nivel 1 o nivel 0 en caso que el nivel 1 no exista.
- RMAN determina el backup nivel 0 o 1 más reciente y realiza un respaldo de todos los bloques que han cambiado posterior a dicho backup. Si no existe backup nivel 1, RMAN copiará todos los bloques que cambiaron del backup nivel 0. En caso de no existir nivel 0, respalda todos los bloques que cambiaron desde que el archivo fue creado.

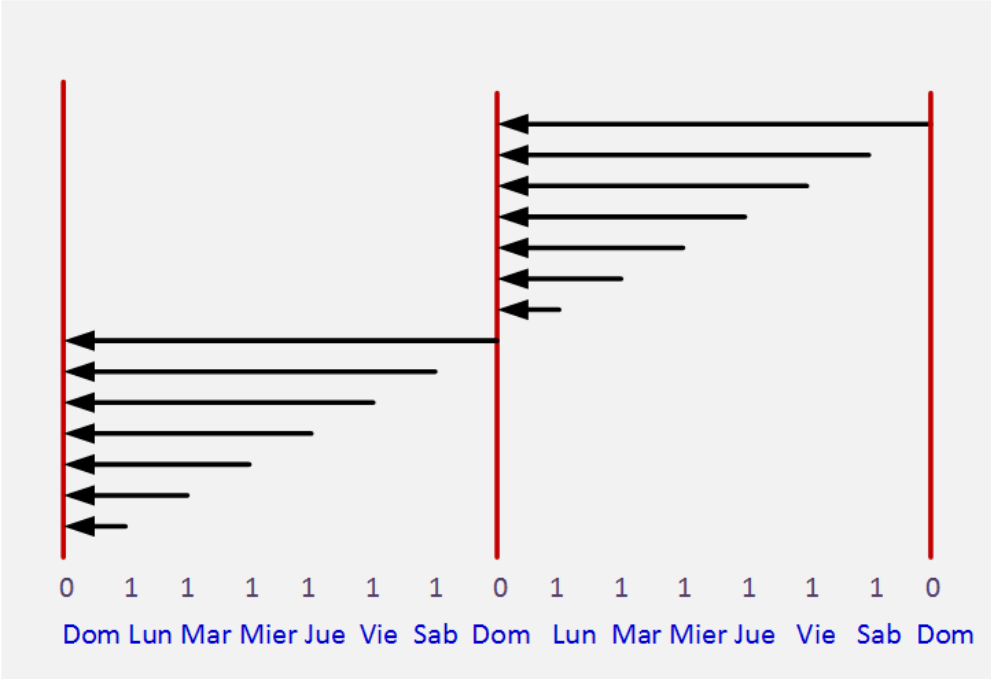
Ejemplo:



- En este ejemplo se crea un backup nivel 0 los domingos. De lunes a sábado se crea un backup incremental nivel 1 que respalda únicamente los cambios realizados durante el día. Al término de la semana se vuelve a crear un backup nivel 0.

8.3.6.2. Backups Incrementales cumulativos.

- Para este tipo de backups se incluyen todos los cambios desde el último backup nivel 0 realizado.
- Este tipo de backups reduce el trabajo que se requiere realizar una posible operación de restauración de algún archivo ya que solo se requiere aplicar a lo más ***un solo*** backup incremental cumulativo.
- Este tipo de respaldo requiere una mayor cantidad de espacio y de tiempo para generarse ya que parte del trabajo realizado en el backup anterior se tiene que realizar nuevamente.



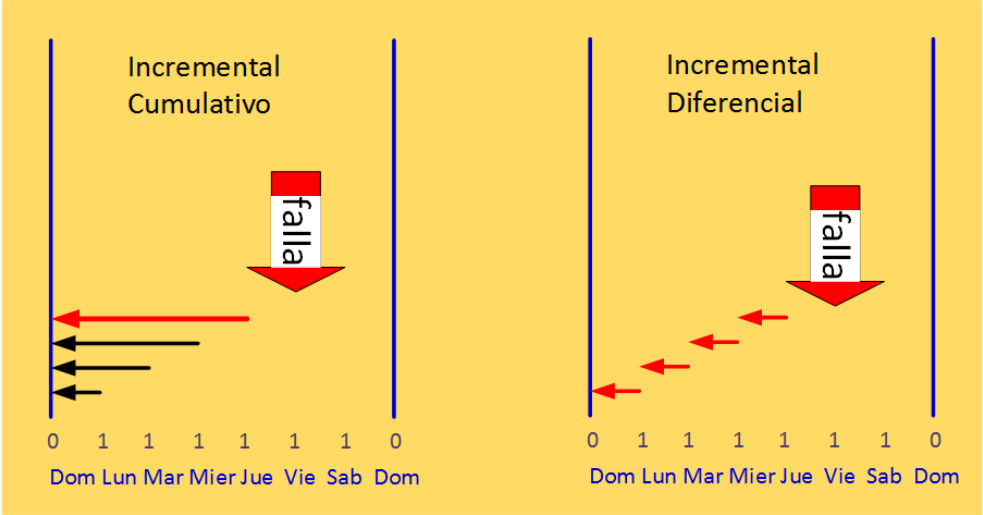
- Como se puede observar en la imagen anterior, cada día se realiza un backup incremental nivel 1 que contiene los cambios acumulados desde el último nivel 0.

Escenario de uso:

Suponer que ocurre una falla el día viernes.

- Si la estrategia seleccionada es backups incrementales diferenciales, RMAN tendría que realizar la restauración de todos los backups empezando por el backup del día lunes hasta día jueves, haciendo uso de un total de 4 backups. Esto requiere mayor trabajo y tiempo.
- Si la estrategia seleccionada es backups incrementales cumulativos, RMAN solo tendría que restaurar el último backup incremental cumulativo existente el cual corresponde al del día jueves.

Ojo: En ambos casos, la operación de recovery no se considera completa ya que los cambios que ocurrieron posterior al respaldo del día jueves y hasta el punto de falla aún no se han aplicado. Para este periodo de tiempo se emplean los *archive redo logs*.



Notar en la imagen anterior, las flechas marcadas en rojo corresponden a los respaldos que se tienen que aplicar para realizar la restauración de un archivo. En un esquema Incremental cumulativo el trabajo requerido es menor respecto al diferencial.

8.3.6.3. Block change tracking.

Para implementar la creación de backups incrementales RMAN realiza un escaneo del data file para determinar los bloques de datos que han cambiado.
Este escaneo se puede omitir si se habilita ***block change tracking.***

- Si se habilita, el proceso de background Change Tracking Writer CTWR se encarga de escribir la dirección de cada bloque de datos que ha cambiado en un archivo llamado *change tracking file.*
- Por default, el archivo se guarda en la ruta especificada por el parámetro `db_create_file_dest`. El archivo tiene formato de bitMap.

```
alter database enable block change tracking using file
  '/unam-bda/backups/block-tracking/change_tracking.dbf'
```

- Una forma de monitorear la efectividad del concepto de ***block change tracking*** es realizar una consulta a la vista `v$backup_datafile`.
- Esta vista es actualizada cada vez que un data file es involucrado en un *backup set*.
- La columna `datafile_blocks` indica el número de bloques que contiene el datafile.
- La columna `blocks_read` indica el número de bloques que fueron leídos para ser incluidos en el último backup.

```
select file#,datafile_blocks,
 (blocks_read / datafile_blocks) * 100 as porcentaje
from v$backup_datafile
where used_change_tracking='YES'
and incremental_level > 0;
```

El proceso de background CTWR es el encargado de realizar este monitoreo. Para revisar si está activo:

```
select program
from v$process
where program like '%CTWR%';
```

8.3.6.4. Algoritmo para construir un respaldo incremental.

Para construir un backup nivel 1 RMAN realiza las siguientes acciones:

- Cada data file cuenta con un *data file checkpoint SCN* el cual se puede consultar en `v$datafile.checkpoint_change#` Todos los cambios con un SCN menor a este SCN se garantizan estar contenidos en el data file.
- Cuando un backup nivel 0 se usa para restaurar a un archivo, el archivo restaurado contiene el SCN que tenía cuando el backup fue realizado.
- Cuando un backup nivel 1 es aplicado a un archivo, el SCN del archivo avanza hasta el SCN del backup Incremental nivel 1. Si hay más backups incrementales, se aplican de forma sucesiva.
- En una operación de recovery RMAN puede aplicar backups o Archive Redo Logs. Si existen ambas opciones, RMAN siempre hará uso de backups ya que estos son más rápidos para ser aplicados con respecto a datos de Redo.

Ejercicio 1.

En este ejercicio se realiza la preparación del ambiente de base de datos para trabajar con respaldos.

A. Con la finalidad de ahorrar la mayor cantidad de espacio, antes de comenzar con los ejercicios posteriores, realizar el borrado de los siguientes tablespaces incluyendo sus data files. Incluir las sentencias correspondientes: `usertbs`, `apps_tbs`, `indx_tbs`, `store_tbs1`, `store_tbs_multiple`, `store_tbs_custom`,`undotbs2`. Al final de este ejercicio, solo deberán existir los tablespaces `system sysaux`, `undotbs1`, `tempts1`, y `users`.

B. Generar una consulta empleando el diccionario de datos que muestre el total de archive redo logs existentes así como en número de destino (dest_id) existentes hasta el momento, es decir, la consulta deberá mostrar el número de archive redo logs existentes agrupados por en número de destino (1 o 2). Se deben obtener únicamente 2 registros. Hacer uso de `v$archived_log`.

C. Debido a los ejercicios realizados hasta el momento, seguramente existirá un número considerable de archive redo logs. Debido a la eliminación de los tablespaces, estos archivos ya no son necesarios, por lo que pueden eliminarse. Realizar las siguientes acciones para realizar la eliminación de estos archivos a través de RMAN.

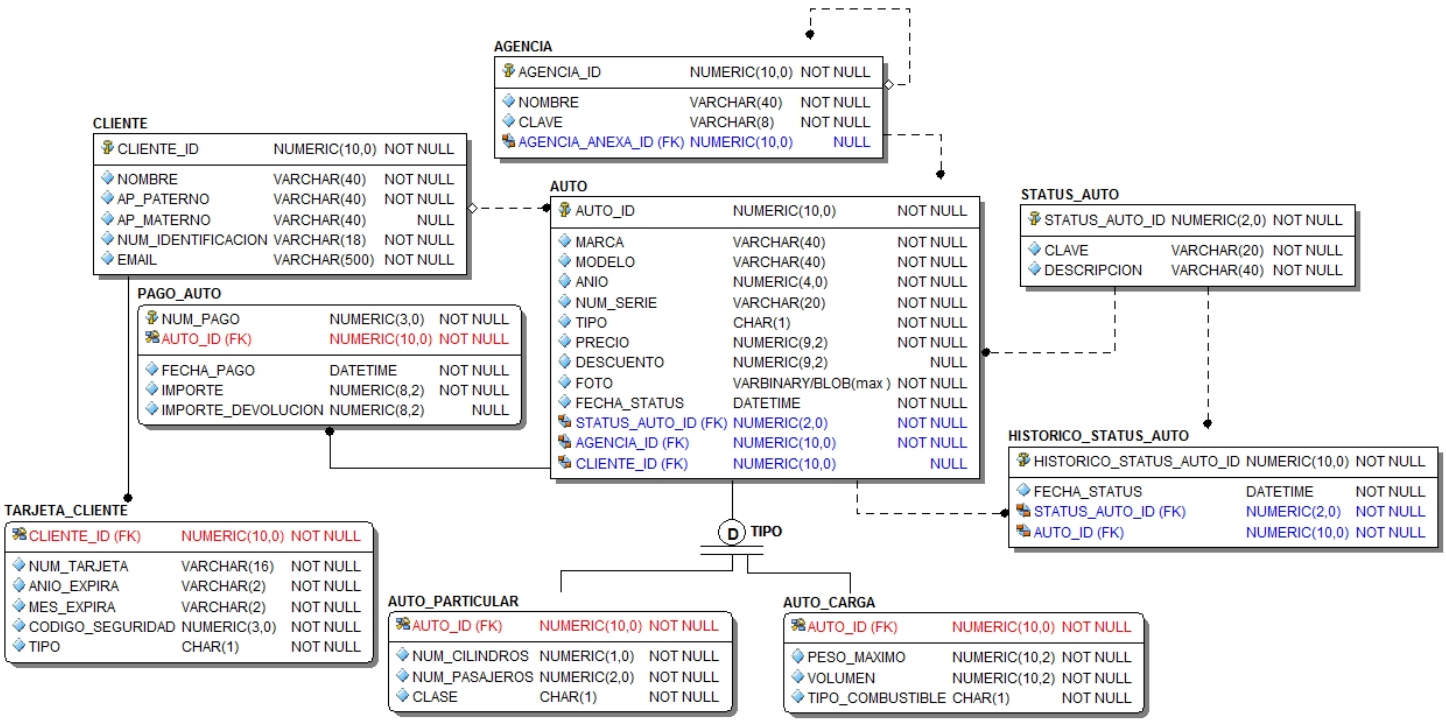
   La estrategia es la siguiente: Hacer un Respaldo completo de la base de datos incluyendo los archive redo logs. Esto permitirá la existencia de al menos un backup. Posteriormente se le indicará a RMAN que elimine a todos los archivos considerados como obsoletos. Esta última acción permitirá realizar una limpieza de todos los archivos que ya no son necesarios y por lo tanto salvar espacio en disco. Ojo: Para realizar estas acciones, se deberá contar con al menos 5GB de espacio en disco. Se proporcionan las instrucciones las cuales serán estudiadas a detalle en secciones posteriores.

```
---1. Abrir una terminal como usuario oracle y entrar a rman
su -l oracle
export ORACLE_SID=jrcbda2
rman

RMAN>
```

```
---2. Conectarse a la base de datos target, ajustar el nombre de la  BD.
 connect target "sys@jrcbda2 as sysdba"

---3.  Realizar un backup contemplando archive redo logs. Observar la ruta
---    donde seran almacenados /unam-bda/backups. En caso de no existir dicha
---    ruta deberá ser creada.
RMAN> backup database plus archivelog format "/unam-bda/backups/backup_%U"

--- 4.  Realizar una consulta al contenido de este directorio para
---     verificar el tamaño del backup (usar otra terminal) , incluir
---     los resultados.  ¿Cuánto espacio se ocupó ?
oracle@pc-jrc-ora backups]$ ls -ltrh

--- 5. Eliminar los archivos obsoletos empleando RMAN
RMAN > delete obsolete


--- 6. Al terminar,  a nivel de sistema operativo mostrar el contenido de los
---    directorios donde se encuentran los archive redo logs.  ¿ Qué sucedió?
---
[oracle@pc-jrc-ora JRCBDA2]$ pwd
/unam-bda/archivelogs/JRCBDA2
[oracle@pc-jrc-ora JRCBDA2]$ ls -ltrh disk_*/*


--- 7. Mostrar ahora el contenido del directorio donde se encuentran los
---    backups. Explicar lo sucedido.

[oracle@pc-jrc-ora backups]$ pwd
/unam-bda/backups
[oracle@pc-jrc-ora backups]$ ls -ltrh *
```

D. Ejecutar nuevamente la consulta del inciso C. Notar que los resultados no varían. Este no es un resultado esperado debido a que los archive redo logs han sido eliminados de sus ubicaciones. La consulta sigue mostrando el mismo número de registros debido a que en el archivo de control la información de los archive redo logs permanece por un periodo predeterminado de 7 días, configurado en el parámetro `control_file_record_keep_time`. Sin embargo existe un cambio. Mostrar la columna `name`. Notar que su valor es nulo. Esto se debe a que los archivos ya no existen físicamente. Para obtener los archivos que realmente existen, incluir como respuesta de este inciso una consulta que vuelva a realizar el conteo pero sin considerar a los archivos que ya no existen físicamente. Incluir el SQL y la salida.

E. Suponer el siguiente modelo relacional el cual describe una base de datos de una empresa automotriz. Para almacenar sus datos se ha decidido crea los siguientes tablespaces:



Nombre	Ubicación	Tamaño inicial de su data file	Requisitos adicionales.
autos_tbs	/u01/app/oracle/oradata/<ORACLE_SID>/autos01.dbf	50mb	Auto crecimiento en intervalos de 10mb con un límite de 100 MB
clientes_tbs	/u01/app/oracle/oradata/JRCBDA2/clientes01.dbf	30mb	Auto crecimiento en intervalos de 10mb con un límite de 100 MB
indexes_tbs	/u01/app/oracle/oradata/JRCBDA2/indexes01.dbf	2mb	Auto crecimiento en intervalos de 10mb con un límite de 100 MB

Generar las sentencias SQL necesarias para crear estos tablespaces.

F. De la carpeta compartida en Drive obtener todos los scripts SQL. Considerar el script `s-02-autos-ddl.sql` Este script contiene la definición del modelo relacional. Realizar los cambios necesarios al modelo para que los datos tanto de las tablas como de los índices se guarden en los siguientes tablespaces.

Objetos	Tablespace
cliente, tarjeta_cliente, agencia, pago_auto	`clientes_tbs`
auto, status_auto, historico_status_auto, auto_particular, auto_carga	`autos_tbs`
Todos los índices de las llaves primarias	`Indexes_tbs`
Todos los índices tipo unique incluidos en el script	`Indexes_tbs`
Todos los índices non unique de las FKs (estos índices no fueron incluidos en el script, se deberán agregar).	`Indexes_tbs`

Ejemplos:

```
create table cliente(
  cliente_id  number(10, 0)    not null,
  . . .
  constraint cliente_pk primary key (cliente_id)
  using index (
    create unique index cliente_pk on cliente(cliente_id)
    tablespace indexes_tbs
  )
) tablespace clientes_tbs;

create index agencia_agencia_anexa_id_ix on agencia(agencia_anexa_id)
tablespace indexes_tbs;
```

Observar en los ejemplos anteriores las instrucciones resaltadas para configurar el tablespace donde los datos serán almacenados. Asegurarse que todas las tablas y todos los índices tanto implícitos como los explícitos queden almacenados en los tablespaces correspondientes. No se necesita incluir estos cambios en las respuestas de este ejercicio.

- Crear un usuario llamado `<iniciales>_autos_bda` con los privilegios necesarios para poder implementar el modelo anterior.
- Asignarle cuota ilimitada al usuario para que pueda almacenar datos en los 3 tablespaces configurados anteriormente.
- Realizar las configuraciones necesarias para que el tablespace por default asignado al usuario sea el tablespace `autos_tbs`.
- Ejecutar el script `s-02-autos-ddl.sql` para crear los objetos con los cambios solicitados.

Hasta este punto no será necesario incluir los resultados para este inciso. Lo único que se deberá incluir son las siguientes consultas (SQL y salida).

- Generar una consulta que muestre el dueño, nombre de las tablas, y el nombre del tablespace asignado a cada una de las tablas de modelo relacional para verificar los resultados. Comprobar que el nombre del tablespace sea el esperado.
- Generar una consulta que muestre los siguientes datos de los índices: el nombre del dueño, nombre del índice, nombre del tablespace donde se almacena, tipo de índice, nombre de la tabla donde se aplica, columna que indica si el índice es `unique` o `non unique`. Comprobar que el nombre del tablespace sea el esperado.

G. Realizar la carga de datos. Generar un programa PL/SQL llamado `s-03-carga-inicial.sql` encargado de ejecutar los scripts SQL que contienen los datos iniciales para este caso de estudio (ubicados en drive).

Ejemplo:

```
--si ocurre un error, se hace rollback de los datos y
--se sale de SQL *Plus
whenever  sqlerror exit rollback

Prompt creando usuario jrc_autos_bda
connect jrc_autos_bda/jorge

set define off

Prompt realizando la carga de datos
--completar

set define on

Prompt confirmando cambios
commit;
```

H. Generar una consulta SQL que muestre el nombre del segmento, y el número de extensiones que contiene para el usuario en cuestión. Con base a los resultados obtenidos ¿Qué tabla es la que hasta el momento ocupa el mayor espacio?

8.4. TECNOLOGÍA FLASHBACK

- Proporciona un conjunto de herramientas que complementan los backups físicos y estrategias de recuperación.
- Proporciona una capa adicional para proteger datos, en especial, para visualizar datos en un punto del tiempo anterior.
- Esta tecnología se divide en 2 principales funcionalidades:
  - Logical Flashback features
  - Flashback Database (Physical).

8.4.1. Logical Flashback features.

- Conjunto de funcionalidades que no dependen de RMAN, están disponibles sin importar si RMAN fue seleccionada como parte de la estrategia de backups.
- Este conjunto de funcionalidades trabajan a nivel lógico, operando sobre los objetos de la base de datos. La mayoría de estas herramientas trabajan con datos ***undo.***
- La siguiente lista muestra las principales funcionalidades de ese grupo.
  - *Oracle flashback query*
    Permite realizar consultas a la base de datos considerando un punto específico en el tiempo. Estas consultas pueden ser recuperadas y restauradas en el estado actual de la base de datos.
  - *Oracle Flashback Version Query*
    Permite visualizar todas las versiones de un conjunto de registros a partir de un instante en el tiempo. Permite la recuperación de cambios no deseados, así como realizar auditoría de cambios.
  - *Oracle Flashback transaction.*
    Permite realizar operaciones de "reversa" en transacciones. Es decir, permite deshacer transacciones que hicieron `commit` en tiempo pasado incluyendo dependencias con otras transacciones, como si estas nunca hubieran ocurrido.
  - *Oracle flashback table*

Permite recuperar una tabla en un instante en el tiempo sin la necesidad de detener la instancia o sin la necesidad de modificar el estado a offline de ciertas partes de la BD.

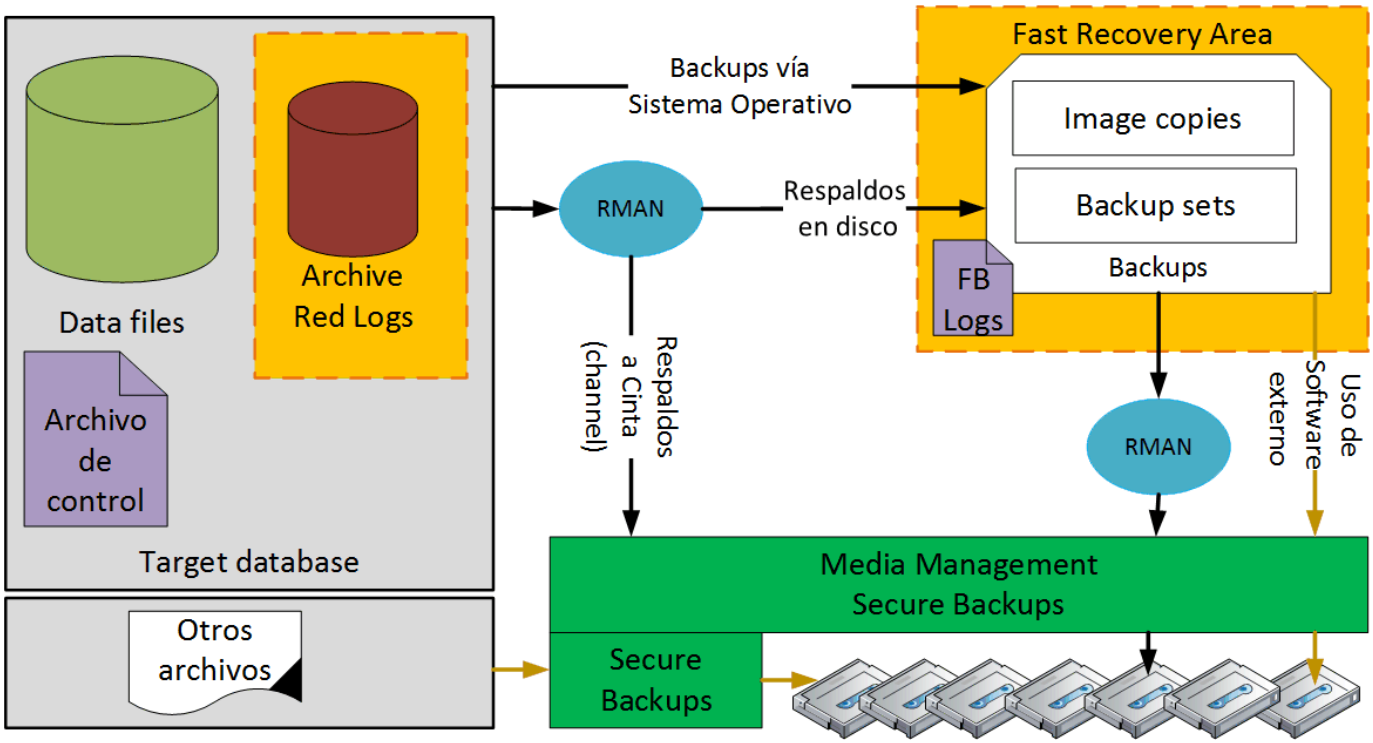
- *Oracle flashback drop.*
  Permite deshacer los efectos de la sentencia `drop`. Util cuando se elimina una tabla por error.

8.4.1.1. Flashback Database

- Permite revertir una base de datos a un punto específico en el tiempo, pero a nivel físico (a nivel archivos).
- Representa una alternativa de protección de datos llamada ***database point –in time recovery (DBPITR).***
- Dependen de RMAN. Si los data files contienen cambios no deseados, se emplea el comando de RMAN `flashback database` para restaurar los archivos en un instante en el tiempo.
- Generalmente esta estrategia es mucho más rápida que una restauración de data files a través de un backup.
- Flashback Database hace uso de ***flashback logs*** para acceder a versiones anteriores de los bloques de datos así como cierta información de los Archived Redo Logs.
- Para habilitar esta funcionalidad es indispensable configurar un área llamada ***fast recovery area*** ya que estos logs solo pueden ser almacenados aquí. El uso de estos logs no está habilitado por default.
- Otra funcionalidad soportada es la posibilidad de crear ***puntos de restauración.*** Un punto de restauración en realidad corresponde a un SCN (System Change Number) generado en un punto en el tiempo.

8.4.2. Configuración de la Fast Recovery Area (FRA).

- FRA permite establecer un área en disco donde la base de datos puede crear, y administrar diversos tipos de archivos relacionados con el tema de ***respaldos*** y ***recuperación***.
- Su uso es altamente recomendado. Su configuración debería ser considerada como el primer paso para para implementar una ***estrategia general de respaldos.***
- Esta área puede contener los siguientes tipos de archivos:
  - Control files
  - Online Redo Logs
  - Archive Redo Logs
  - Flashback Logs
  - RMAN Backups
- Los archivos contenidos en la FRA pueden ser tanto *permanentes* como *temporales*.
- La base de datos puede eliminar de forma automática archivos que adquieren el estado de ***obsoletos***. Este estado dependerá de la configuración de una ***política de retención de respaldos*** o que estos archivos hayan sido respaldados en *cintas*.
- En realidad no elimina estos archivos mientras haya espacio disponible para continuar con la operación. Si el espacio está próximo a terminarse, la base de datos comienza a eliminar estos archivos.
- Los archivos que se guardan en esta área se administran por la propia base de datos. Los nombres de los archivos se guardan empleando el formato OMF (Oracle Managed Format).



La siguiente tabla muestra la lista de archivos que puede contener el FRA, su tipo, así como el efecto que producen hacia la base de datos.

Archivo	Tipo	Comportamiento de la BD cuando la FRA no está disponible.
Copias multiplexadas de los archivos de control.	Permanentes	La instancia falla si la base de datos no puede escribir la copia de los archivos en la FRA. La falla ocurre aunque exista la copia fuera de la FRA.
Online Redo Logs	Permanentes	La instancia no se afecta si al menos existe una copia fuera de la FRA. Esto significa que la copia puede fallar al escribirse en la FRA y la instancia sigue viva.
Archive Redo Logs	Transitorios	Mismo comportamiento respecto a los Online Redo Logs
Image Copies de data files y control files	Transitorios	Image Copy: Copia de un archivo que se realiza a través de comandos del sistema operativo y se almacena en la FRA. La instancia no se afecta si estas copias no pueden ser almacenadas. El concepto de *Image copy* se revisa más adelante.
Backup pieces	Transitorios	Se refieren a los archivos que genera un backup al emplear RMAN en el modo de *backup set.* La instancia no se afecta si los archivos no se pueden almacenar. El concepto de *backup set* concepto se revisa más adelante.
Flashback logs	Transitorios	La instancia no se afecta a menos que se hayan configurado la habilitación de *puntos de restauración garantizados*. Lo anterior significa que la base de datos debe garantizar la funcionalidad para poder realizar puntos de restauración. Si esto ocurre, la instancia fallará. De lo contrario, la funcionalidad *flashback database* será deshabilitada, escribe mensaje de error en el *alert log,* pero la instancia continua con su operación.

8.4.2.1. Habilitar la Fast Recovery Area.

La siguiente tabla muestra los parámetros que deben ser configurados para habilitar esta área.

Parámetro	Descripción
`db_recovery_file_dest_size`	Indica el tamaño máximo en disco que será empleado para almacenar el contenido de la FRA. Este parámetro debe ser definido antes que el parámetro `db_recovery_file_dest`. Aumentar un 10% respecto al valor estimado.
`db_recovery_file_dest`	Especifica la ubicación donde se almacenarán los archivos de la FRA. Puede ser una ubicación en disco o un grupo de discos ASM.
`db_flashback_retention_target`	Especifica el número máximo en minutos dentro del cual una base de datos podría hacer flashback hacia el pasado. Por ejemplo, si su valor es 30, significa que la base de datos garantiza recuperar el estado anterior de los datos hasta por un periodo máximo de 30 minutos. Es un parámetro opcional empleado únicamente por flashback database (*guaranteed restore point*).

8.4.3. Consideraciones de la FRA.

- Entre más grande sea, más útil será. Lo ideal es que esta área sea capaz de contener los archivos de control, Online Redo Logs, archive Redo Logs y flashback logs, y de preferencia que contenga también todos los data files, backups, backups incrementales.
- Si se carece de grandes cantidades de espacio, se puede optar por guardar únicamente los tablespaces más importantes, y todos los archive redo Logs que aún no han sido respaldados en cintas.
- Cómo mínimo, la FRA debería contener todos los archived redo Logs no respaldados en cintas.

8.4.3.1. Fórmulas para calcular el tamaño de la FRA.

Los siguientes aspectos son empleados para determinar el tamaño de la FRA:

- La BD tiene un tamaño pequeño o grande número de bloques de datos que cambia de forma frecuente.
- Los respaldos se guardan únicamente en disco o en cinta.
- Se ha habilitado el parámetro `db_flashback_retention_target` para garantizar operaciones *flashback database* por cierto periodo de tiempo.
- Se ha configurado una política de retención de backups
- Si se planea hace uso de flashback logs, el volumen generado es aproximadamente del mismo orden en magnitud de la cantidad de Redo generado.
- Suponer que se establece `db_flashback_retention_target` a 24 hrs, la BD genera 20GB de redo al día, entonces la regla generar sería que el espacio requerido para guardar flashback logs sea de 20 a 30 GB.

Ejemplo:

Suponer que se desea determinar el tamaño de la FRA con las siguientes condiciones:

- Backup retention policy = `redundancy 1`.
- Se desea emplear la estrategia sugerida de *incremental forever* (se revisa más adelante)
- Considerar que n = número de días que transcurren para realizar backups incrementales.

$$diskQuota = tamaño\ de\ una\ copia\ de\ la\ bd + \\ tamaño\ de\ un\ backup\ incremental + \\ tamaño\ de\ (n+1)\ dias\ \ de\ archive\ redo\ logs + \\ tamaño\ del\ control\ file + \\ tamaño\ de\ un\ miembro\ de\ redo\ logs * número\ de\ mienbros + \\ tamaño\ de\ los\ \ flashback\ logs$$

- La FRA debe estar ubicada en discos diferentes a los discos donde se almacenan los archivos activos de la base de datos
- Por lo anterior, los valores de los parámetros `db_recovery_file_dest`, `db_create_file_dest`, `db_create_online_log_dest_n` deben tener ubicaciones diferentes.

8.4.4. Deshabilitar la FRA.

Las siguientes instrucciones se emplean para realizar esta tarea.

```
alter database flasback off;
```

- En caso que las ubicaciones de los online Redo Logs hayan sido configuradas para ser almacenados dentro de la FRA, modificar los parámetros `log_archive_dest_n` para que apunten a ubicaciones externas a la FRA.
- Deshabilitar el parámetro `db_recovery_file_dest`

```
alter system set db_recovery_file_dest='';
```

8.4.5. Configuración de RMAN para guardar archivos en la FRA.

- Al emplear el comando `backup` de RMAN sin hacer uso del parámetro `format` provoca que los archivos se almacenen en la FRA siempre y cuando la FRA esté habilitada.
- Para el caso del archivo de control, RMAN puede realizar respaldos de forma automática. Para ello, se deberá asegurar que el siguiente parámetro de RMAN no esté configurado con rutas que apunten a ubicaciones fuera de la FRA.

```
configure controlfile autobackup format for device type disk clear
```

- Para el caso de los archive redo logs, es posible configurar alguno de los parámetros `log_archive_dest_n` para que la copia se guarde en la FRA.

Ejemplo:

```
log_archive_dest_n='LOCATION=USE_DB_RECOVERY_FILE_DES'
```

- Adicional a la configuración anterior, se recomienda guardar otra copia de los Archive Redo Logs para proteger la copia que se encuentra en la FRA.
- Los comandos `recover database`, `recover tablespace`, `recover .. block` y `flashback database` se encargan de hacer una operación de *restore* (reemplazo de archive redo logs) tomados

de algún backup externo a la FRA para aplicar `recovery`. RMAN toma estos archivos para hacer el recovery, los copia a la FRA y posteriormente los elimina de la ubicación anterior.

8.5. CONFIGURACIÓN DE LA POLÍTICA DE RETENCIÓN DE RESPALDOS Y ARCHIVED REDO LOGS.

Permite especificar qué backups deben ser conservados basados en los requerimientos de retención de cada base de datos. Para realizar esta configuración se emplea el comando de RMAN
`configure retention policy`

- Toda base de datos productiva debe definir su política de retención.
- El espacio de almacenamiento puede crecer rápidamente, por lo que backups deben ser eliminados cuando estos se consideren obsoletos. Esta política le permite identificar a la base de datos el momento en el que un backup se puede ya considerar como obsoleto.
- La configuración se puede realizar empleando 2 técnicas:
  - Redundancy based policy
  - Recovery window based policy

8.5.1. Redundancy based policy

- En este escenario se configura el número mínimo de backups completos nivel 0 deben existir de cada data file así como del archivo de control.
- La política por default para este caso es 1.
- Se emplea el comando `configure retention policy to redundancy n`
- Si el número de archivos excede esta política, los archivos se consideran obsoletos.
- RMAN mantiene todos los archived redo logs así como los backups incrementales que son necesarios para hacer un posible recovery de backups (archivos) marcados como no obsoletos.

8.5.2. Recovery Window based policy

- En este caso se configura el número de días hacia atrás a partir de la fecha actual como periodo máximo en el que la base de datos puede aplicar un recovery.
- RMAN no considerará a un backup nivel 0 como obsoleto si este fue realizado durante esta ventana de tiempo.
- RMAN mantendrá todos los archive redo logs y backups nivel 1 incrementables que son necesarios para para hacer la recuperación en cualquier punto dentro de esta ventana de tiempo.

Ejemplo:

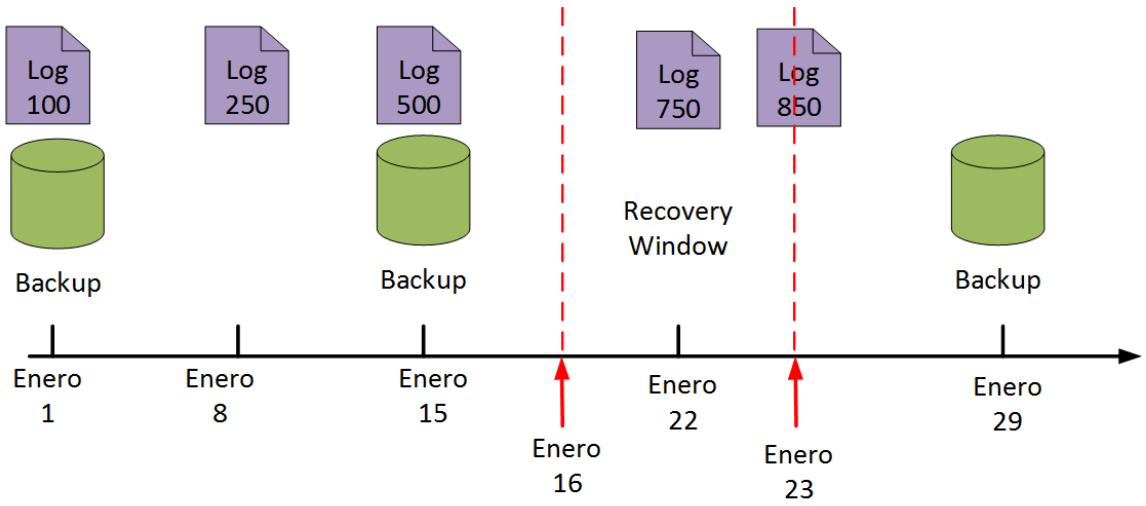
```
configure retention policy to recovery window of 7 days;
```

En este caso, la base de datos podrá ser recuperada hacia cualquier punto dentro de la última semana.

- RMAN no elimina de forma automática los backups obsoletos, solo se marcan como tal en `v%backup_files`.
- Los archivos pueden eliminarse empleando el comando `delete obsolete`.
- El comando `run obsolete` puede ser empleado para determinar los backups considerados como obsoletos.

Ejemplo:

Suponer la siguiente programación de eventos



- En este escenario se ha configurado una ventana de recovery de 7 dias que se extiende del 16 a 23 de enero. Asumiendo que la fecha actual es el día 23 de enero, RMAN debe conservar los archive redo logs 750, 850 y 500 así como el backup del día 15 de enero. Estos últimos 2 son especialmente importantes en caso de solicitar una recuperación de la base de datos entre el día 16 y 22 de enero, por ejemplo, recuperar la base de datos al día 20.
- El backup del día enero 1 y los archive logs 100 y 250 son marcados como obsoletos ya que no se requieren para realizar un recovery dentro de la ventana configurada.
- Notar que si existiera un backup el día 22 de enero, aun así el backup del día 15 tendría que conservarse por la misma razón antes mencionada. Este backup no permitiría hacer recovery entre los días 16 y 22.

8.5.3. Deshabilitar la política de retención

Si la política de retención se deshabilita, RMAN no marcará a los archivos como obsoletos. Para realizar esta acción se emplea el siguiente comando:

```
configure retention policy to none;
```

La siguiente tabla muestra los criterios que utiliza RMAN para determiner si un archive puede ser omitido al realizar backups.

Tipo de archivo	Criterio
Data file	El archivo debe tener el mismo DBId, Checkpoint SCN, creation SCN, Reset Logs SCN. El data file debe estar offline-normal, read-only, o cerrado de forma normal.
Archived Log	El archivo debe tener el mismo DBID, número de secuencia, resetlogs SCN
Backup Set	Mismo DBID, backup set ID.

- La configuración de la política de retención también influye en esta decisión.

8.5.4. Política de retención para Archive Redo Logs

- Es posible emplear RMAN para crear una configuración que permita administrar la permanencia de los Archive redo Logs así como su marcado como obsoletos.
- Se emplea el siguiente comando `configure archivelog deletion policy`.
- Esta configuración puede ser aplicada a todos los destinos de los archive logs incluyendo aquellos existentes en la FRA.
- Los archive redo logs pueden ser eliminados automáticamente o a través de comandos de RMAN.
- Solo los archivos que se encuentran en la FRA pueden ser eliminados de forma automática. Para este caso, la BD trata de mantenerlos el mayor tiempo posible, y los elimina cuando se requiere más espacio para almacenar otros archivos.
- Para los demás archivos, se emplean los comandos `backup … delete input`,`delete archivelog`.
- Por default no existe política de retención. En este caso, los archive Redo Logs contenidos en la FRA son elegibles para ser eliminados únicamente cuando existe al menos un respaldo en disco o en citas.

La sintaxis general para habilitar esta política es:

```
configure archivelog deletion policy backed up integer times to device type
```

Ejemplo:

Configurar la siguiente política de retención de archive redo logs: Eliminar todos los archive redo logs tanto de la FRA como de ubicaciones externas cuando los archive redo logs han sido respaldados al menos 2 veces en disco.

```
configure archivelog deletion policy
  to backed up 2 times to disk;
```

8.5.5. Optimización de backups.

Esta optimización se refiere a la posibilidad de saltar ú omitir el respaldo de archivos cuando estos ya han sido respaldado anteriormente bajo ciertas circunstancias. Para habilitar esta verificación se emplea el siguiente comando:

```
configure backup optimization on
```

8.6. ADMINISTRACIÓN DE BACKUPS CON RMAN

- Cualquier operación con RMAN requiere conexión a la BD target.
- Se emplea el comando `rman`:

```
[jorge@pc-jrc-ora admin]$ rman

RMAN> connect target "jorge@jrcbd2 as sysbackup"

target database Password:
connected to target database: JRCBD2 (DBID=693525726)

RMAN>
```

- En el ejemplo anterior, el usuario `jorge` se conecta a la target DB con service name "`jrcbda2`".
- Notar que el usuario requiere el privilegio `sysbackup` o `sysdba`.

Por default, RMAN es pre-configurado con ciertos valores. Para mostrar esta configuración se emplea en siguiente comando:

```
RMAN> show all;

RMAN configuration parameters for database with db_unique_name JRCBDA2 are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP ON; # 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 COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR LOAD
TRUE ; # default
CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/app/oracle/product/18.0.0/dbhome_1/dbs/snapcf_jrcbd2.f'; # default
```

Para modificar un valor se emplea el comando `configure`

Ejemplos:

```
configure device type disk paralelism 4
backup type to compressed backupset;
configure backup optimization on;
configure retention policy to redundancy 3;
configure channel device type disk format '/disk1/ora_df%t_s%s_s%p';
```

8.6.1. Respaldar una base de datos.

- Para realizar un respaldo de la base de datos, se emplea el comando `backup`.

```
RMAN> backup database;
```

La instrucción anterior realiza un respaldo completo de la base de datos empleado todos los valores de los parámetros configurados a través del comando `configure` o haciendo su de sus valores por default. La instancia debe estar en modo `mount` o `open`.

- De forma adicional es posible crear un script RMAN

```
run {
  shutdown immediate;
  startup mount;
  backup database;
  alter database open;
}
```

Típicamente estos scripts se almacenan con extensión `.rman` y se ejecuta de la siguiente manera:

```
rman target sys @offline_full_whole.rman
```

La instrucción anterior realiza un backup de la base de datos local. Para realizar un backup de otra BD configurada en el archivo `tnsnames.ora` con el nombre `jrcbd3`, el comando será:

```
rman target sys@jrcbd3 @offline_full_whole.rman
```

Para eliminar un backup se emplea el comando `delete backup`.

Ejemplos:

```
rman> delete backup;
rman> delete backupset 23;
rman> delete noprompt backup;
```

8.6.1.1. Configurar tipo de almacenamiento.

```
backup device type disk database;
```

En este ejemplo se hace uso de un disco (default). El valor `sbt` se emplea para escribir un respaldo en cintas. Este valor se puede configurar con el siguiente comando:

```
configure default device type
```

8.6.1.2. Configurar backup set / image copy

- Respaldar empleando un image copy

```
backup as copy device type disk database;
```

- Respaldar empleando backup sets.

```
backup as backupset database;

backup as backupset device type disk database;

backup as backupset device type sbt database;
```

8.6.1.3. Configurar el formato para nombrar archivos.

Las siguientes reglas se emplean para para determinar el formato que será empleado por RMAN para nombrar a los archivos de un backup. Las opciones se listan en orden de precedencia.

1. Empleando el parámetro `format` del comando `backup`.

```
backup database format "/disk1/backup_%U";
```

- El formato especifica la ubicación donde el backup será almacenado. `%U` permite generar nombres únicos formados por la siguiente estructura (data files empleando backup sets, image copies, archive redo logs, y control files):

Backup set
`%u_%p_%c`

Image copy
```
data-D-%d_id-%I_TS-%N_FNO-%f_%u
```

Archive redo log.
```
arch-D_%d-id-%I_S-%e_T-%h_A-%a_%u
```

Control file.
```
cf-D_%d-id-%I_%u
```

`%D` día del mes
`%d` nombre de la BD
`%I` Id de la BD
`%N` nombre del tablespace
`%f` Nombre absoluto del data file
`%u` cadena de 8 caracteres constituida por una representación del backup set o image copy así como un timestamp que corresponde a la fecha de creación.
`%e` Secuencia del archive redo log.
`%h` Archive redo log thread number
`%a` Activation ID de la base de datos

8.6.1.4. Configurar formato en múltiples discos.

- La siguiente configuración permite generar un backup realizando escrituras en paralelo para mejorar desempeño. El comando allocate channel permite establecer la escritura hacia cada dispositivo.

```
run {
  allocate channel disk1 device type disk format '/disk1/%d_backups/%u';
  allocate channel disk2 device type disk format '/disk2/%d_backups/%u';
  allocate channel disk3 device type disk format '/disk3/%d_backups/%u';
  backup as copy database;
}
```

- Si se desea configurar estos parámetros haciendo uso del comando configure:

```
configure device type disk parallelism 3;
configure default device type to disk;
configure channel 1 device type disk format '/disk1/%d_backups/%u';
configure channel 2 device type disk format '/disk2/%d_backups/%u';
configure channel 3 device type disk format '/disk3/%d_backups/%u';
```

- El comando backup se reduce a lo siguiente:

```
backup as copy database;
```

8.6.1.5. Configurar tags.

- Un tag representa a una cadena definida por el usuario que puede ser especificada a un respaldo para describir su propósito, uso u objetivo.
- Los tags pueden ser empleados para backup sets, data files, control files.
- Los tags pueden ser empleados inclusive para identificar backups en una operación restore.

- En la práctica los tags se emplean para distinguir a un conjunto de respaldos creados como parte de una estrategia configurada: backups incrementales, full, etc. Ejemplos: `weekly_incremental`, etc.

Ejemplo:

```
backup as backupset
  copies 1
  datafile 7
  tag Monday_backup;
```

8.6.1.6. Respaldar archive redo logs.

El siguiente comando realiza un respaldo completo de la base de datos, realiza switch de los Redo logs e incluye los archive redo logs en el backup.

```
backup database plus archivelog;
```

8.6.1.7. Respaldar tablespaces

```
backup device type disk tablespace users, tools;
```

8.6.1.8. Respaldar data files.

```
backup device type sbt datafile 1,2,3,4 datafilecopy '/tmp/system01.dbf';
```

8.6.1.9. Respaldar el archive de control.

En caso de que la configuración `configure controlfile autobackup` no esté habilitada, se deben emplear alguna de las siguientes instrucciones para incluirlo en un backup.

```
backup current controlfile

backup device type sbt tablespace users include current controlfile;

backup as copy current controlfile format '/tmp/control01.ctl';

backup as copy
current controlfile
format '/tmp/control01.ctl';
backup device type sbt
controlfilecopy '/tmp/control01.ctl';
```

8.6.1.10. Respaldar el archive de parámetros

```
backup device type sbt spfile;
```

8.6.2. Respaldar a través de backups incrementales.

- Como punto inicial para hacer uso de estas funcionalidades, se debe contar con un diseño de estrategia de respaldos que contemple a los distintos tipos de respaldos incrementales.
- Por ejemplo, Diseño de un esquema de 3 niveles de respaldos:

- Nivel 0 cada mes.
- Cumulativo cada semana
- Diferencial tomado cada día.

- Con un esquema como el anterior, nunca se tendría que aplicar más allá de un día de Redo Logs para realizar un recovery exitoso.
- Como regla general, se recomienda tomar un backup nivel 0 cuando los datos han cambiado en un 20% o más.
- Una forma de ver qué tanto cambiaron los datos es verificando el tamaño de los backups incrementales es a través de la siguiente consulta:

```
select file#, incremental_level, completion_time,
  blocks, datafile_blocks
from v$backup_datafile
where incremental_level > 0
and blocks / datafile_blocks > .2
order by completion_time;
```

- En esta consulta se comparan el número de bloques en nivel 1 y en nivel 0. Se recomienda crear un nuevo backup nivel 0 cuando el backup nivel 1 más reciente tiene aproximadamente la mitad del backup de nivel 0.

8.6.2.1. Crear un respaldo incremental nivel 0

```
backup as backupset incremental level 0 database;
```

8.6.2.2. Crear un respaldo incremental nivel 1 diferencial.

```
backup as backupset incremental level 1 database;
```

8.6.2.1. Crear un respaldo incrementan nivel 1 cumulativo.

```
backup as backupset incremental level 1 cumulative database;
```

8.6.3. Principales vistas asociadas con la generación de backups.

```
v$backup_files
v$backup_set
v$backup_piece
v$backup_redolog
v$backup_spfile
v$backup_datafile
v$backup_device
v$rman_configuration
v$backup_piece_details
```

Ejercicio 2.

A. Aplicar las siguientes configuraciones en RMAN para establecer el directorio por default donde se ubicarán todos los backups así como el respaldo automático del archivo de control.

```
configure channel device type disk
format '/unam-bda/backups/backup_%U.bkp' maxpiecesize 2G;

configure controlfile autobackup
format for device type disk to '/unam-bda/backups/ctl_file%F.bkp';
```

B. A partir del modelo relacional creado en el ejercicio anterior realizar un *full backup* considerando la carga inicial realizada. Para ello emplear la siguiente instrucción. Notar el uso del parámetro tag en el cual se especifica una etiqueta para identificar fácilmente al backup que se realizará.

```
backup database plus archivelog
tag autos_full_inicial;
```

C. Mostrar la información de los backups. Existen varias formas para visualizar los datos de los backups realizados hasta el momento. El comando `backup list` permite mostrar todos los backups acumulados hasta el momento. Dicho comando cuenta con diversos parámetros para personalizar el detalle. Ejecutar el comando `list backup`, identificar los backups realizado en el punto anterior con base al tag asignado y a la fecha, incluir la salida. Considerar los siguientes puntos para entender la salida, contestar las preguntas.

- El comando `backup` ejecutado en el inciso anterior genera varios backup sets. Dependiendo el tamaño de los datos a respaldar se pueden generar varios archivos.
- Observar que en la salida a cada backup set se le asigna un ID o Key (BS Key). **Indicar la lista de IDs que se generaron**.
- Observar que se generan diferentes tipos de backup sets los cuales se pueden identificar por las siguientes líneas incluidas en la salida:
    - List of Archived Logs in backup set (Pueden generarse varios backup set de este tipo) **¿Cuántos backup sets se generaron de este tipo?**
    - List of Datafiles in backup set 20 (Pueden generarse varios backup set de este tipo) **¿Cuántos backup sets se generaron de este tipo?**
    - SPFILE Included, Control File Included. Este Backup set incluye al control file y al SPFILE. **¿Cuántos backup sets se generaron de este tipo?**

D. Ejecutar el comando `list backup summary`. Este comando permite identificar de forma más sencilla los backup set generados anteriormente. Incluir la salida. Observar que algunos de los data sets que se obtuvieron tienen el tag asignado anteriormente.

E. Consultar ahora los data sets a través de la vista `v$backup_piece_details`. La vista muestra cada uno de los archivos generados durante la ejecución de un backup. Con base a los BS keys y a la fecha, mostrar los registros que corresponden al backup realizado anteriormente. Por la cantidad de columnas, dividir la consulta en 2:

- Para la primera consulta mostrar `session_key,bs_key, set_count,handle,tag`

- Para la segunda consulta mostrar `session_key,bs_key, status,start_time, completion_time, elapsed_seconds` (quitar los decimales), `elapsed_seconds, deleted, size_bytes_display`

De las consultas anteriores se destaca lo siguiente:

- Observar la columna `handle` la cual contiene las ubicaciones y los nombres de los archivos del respaldo realizado. Confirmar a nivel de sistema operativo que estos archivos existen.
- Notar que todos los archivos están asociados al mismo `session_key`.
- La columna `size_bytes_display` permite conocer el tamaño en disco que cada uno de estos archivos requiere. Estos datos serán útiles más adelante para calcular el valor inicial de la FRA. Con base A los resultados obtenidos, llenar la siguiente tabla:

BS_KEY	Ubicación del archivo	Tamaño que ocupa en disco	Contenido del archivo (data files, archived redo logs o control file / SPFILE)

F. Una vez que el backup ha sido ejecutado con éxito es posible verificar si existen archivos o backups que pudieran ser considerados como obsoletos con base a la política de retención configurada. Por default su valor es 1 como se indica en el parámetro de configuración `configure retention policy to redundancy 1;` Para revisar qué archivos pueden eliminarse, ejecutar el siguiente comando:

```
RMAN> report obsolete;
```

- Revisar la salida del comando anterior y generar una lista que indique los backup sets o Backup pieces que pueden ser ya eliminados. Notar que la mayoría de los registros mosteados corresponden a archived redo logs.
- De la consulta anterior, ¿aparece algún archivo que pertenece al backup realizado en pasos anteriores? De ser así, explicar por qué razón se ha marcado como obsoleto a pesar de ser creado hace solo unos instantes.
- Ejecutar el comando `delete obsolete` para eliminar a los archivos obsoletos.

Ejercicio 3.

En este ejercicio se procederá con la configuración y habilitación de la FRA. Para ello, los primeros incisos de este ejercicio se enfocan al cálculo aproximado del espacio para posteriormente realizar su configuración y verificación de su correcto funcionamiento. Para poder calcular el espacio aproximado se requiere calcular la cantidad de Redo Logs que la base de datos produciría en un día de producción normal, así como la generación de un backup incremental considerando un esquema de respaldos incrementales diferenciales de un día.

A. Revisar la lista de archived redo log existentes. Para realizar el cálculo se requiere partir de un esquema limpio. Para asegurar esta condición, realizar nuevamente un full backup similar al realizado en el ejercicio anterior y ejecutar una limpieza de archivos obsoletos. Posteriormente volver a ejecutar la consulta en `v$backup_piece_details`.

B. Simular la operación diaria. De la carpeta compartida obtener el `script s-05-genera-redo.sql` Este script ejecuta una serie de bloques PL/SQL anónimos que provocan la generación aleatoria de instrucciones DML sobre la mayoría de las tablas relativas al control de autos. Todos los cambios que realice este script serán considerados como la operación diaria en un día normal productivo. Ejecutar el script, e incluir la salida como parte de la respuesta de este ejercicio.

C. Realizar un backup nivel 0 incluyendo los archive redo logs. Ejecutar nuevamente la consulta en realizada en ejercicios anteriores, incluir la salida como parte de este ejercicio para los backup sets generados en este último backup. Posteriormente realizar una limpieza de archivos obsoletos. Emplear el `tag autos_backup_nivel_0_1`.

D. Ejecutar nuevamente el script para simular la carga diaria. Posteriormente, realizar un backup Incremental nivel 1 cumulativo incluyendo los archived redo logs. Ejecutar nuevamente la consulta en `v$backup_piece_details` realizada en ejercicios anteriores mostrando únicamente los archivos de este backup. Emplear el tag `autos_backup_nivel_1_1`;

E. Ejecutar nuevamente el comando `list backup` para identificar a cada backup set. Llenar la siguiente tabla.

BS_KEY	Ubicación del archivo	Tamaño que ocupa en disco	Contenido del archivo (data files, archived redo logs o control file / SPFILE)

F. Con base a la tabla del inciso anterior y al resultado del backup nivel 0, llenar la siguiente tabla.

- Asumir un esquema en el que se realizará un backup nivel tipo 0 cada domingo, y un incremental diferencial diario. Por lo tanto, N = 6.
- Tamaño de una copia de la BD se obtendrá el valor de la columna `size_bytes_display` para el backup set que corresponde al respaldo de los data files. Incluir este valor como parte de la respuesta de este inciso.

Variable	Tamaño
Tamaño de una copia de la base de datos (tamaño del backup set nivel 0 que contiene a los data files)	
Tamaño de un backup incremental nivel 1 (tamaño del backup set nivel 1 que contiene a los data files)	
Tamaño de los archive redo logs que se producen en un día productivo	
Tamaño de los archive redo Logs que se producirán en N + 1 días donde N = 6 (como se mencionó).	
Tamaño del backup set que contiene al archivo de control	
Tamaño de los flashback logs Suponer que se desea configurar el parámetro `db_flashback_retention_target` a 24 hrs. (Calcular con base a lo revisado en clase).	
Tamaño de uno de los miembros de redo logs * (N+1)	
Total de espacio estimado para la FRA. (Aumentar 10% a la suma de las variables)	

G. Ejecutar nuevamente el script sql empleado para generar datos REDO. Posteriormente generar un backup incremental nivel 1 diferencial. Emplear el tag `autos_backup_nivel_1_2`. Mostrar los detalles que reporta RMAN al ejecutar el comando `list backup` solo para este backup set. **¿Cuánto espacio requiere un incremental diferencial el cual representa la actividad de un día?** Posteriormente realizar limpieza de archivos obsoletos.

H. Configurar los siguientes parámetros para habilitar la FRA con base a la siguiente tabla. Incluir las instrucciones SQL. Antes de aplicar los cambios, crear un pfile a partir del spfile. Notar que estas configuraciones no requieren reiniciar la instancia por lo que se recomienda emplear el valor del cláusula `scope` a `'both'`.

Parámetro	Valor
`db_recovery_file_dest_size`	El valor calculado anteriormente (redondear al siguiente número entero).
`db_recovery_file_dest`	Emplear la ruta `/unam-bda/fast-reco-area`
`db_flashback_retention_target`	Especificar el equivalente a 24 hrs.

I. Una vez que la FRA ha sido configurada, deberán realizarse las siguientes configuraciones. Generar las sentencias SQL necesarias, incluirlas como parte de la respuesta de este ejercicio.

- Modificar la ruta de la segunda copia de los archived redo logs para que esta sea almacenada en la FRA.
- Aplicar el comando de RMAN correspondiente para provocar que el archivo de control y el SPFILE se almacenen en la FRA.
- Aplicar el comando de RMAN correspondiente para provocar que los backup set se guarden en la FRA.
- Generar un nuevo backup nivel 1 cumulativo considerando archive logs para comprobar resultados.

8.7. DIAGNÓSTICO Y REPARACIÓN DE FALLAS.

Las palabras ***restore*** y ***recover*** tienen significado diferente, las cuales se aplican posterior a la ocurrencia de una falla.

- *Restore*: Significa restaurar un archivo dañado a través de la extracción de un backup previamente realizado.
- *Recover*: Aplicar vectores de cambios obtenidos de los Redo Logs con la finalidad de aplicar los últimos cambios a la base de datos (ponerla al día).

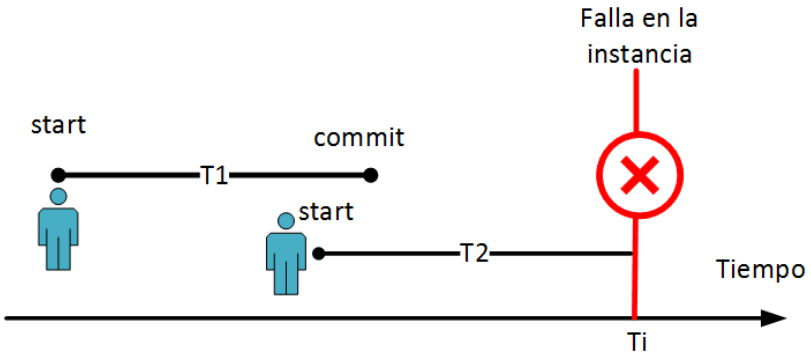
8.7.1. Tipos de fallas.

A nivel general, las fallas que pueden ocurrir en una base de datos son:

- *Statement failure:*
  Falla una sentencia SQL (insert, update, etc.).
- *User Process failure*:
  Falla en una sesión de un usuario.
- Network failure:
  La conectividad hacia la base de datos se pierde.
- *User error*:
  El usuario concluye correctamente una operación, pero dicha operación es incorrecta. Por ejemplo, el usuario elimina datos equivocadamente.
- *Instance failure*:
  La base de datos se detiene repentinamente.
- *Media failure*:
  Pérdida de alguno de los archivos físicos requeridos para el correcto funcionamiento de la base de datos.
- *Application errors*:
  Fallas en el software que accede a la base de datos, típicamente provoca corrupción de bloques de datos.

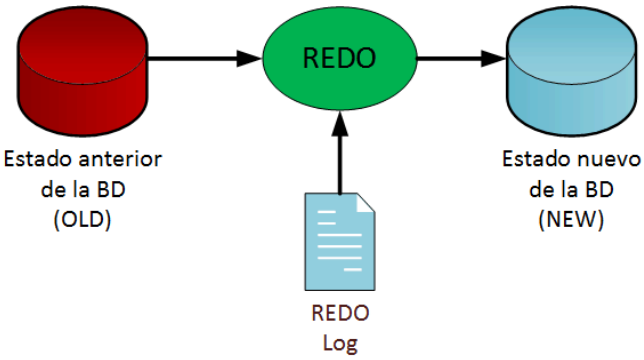
8.7.2. Instance recovery

- Se produce posterior a una falla inesperada a nivel de instancia al intentar abrir una base de datos cuyos archivos no fueron sincronizados al momento de realizar una operación *shutdown*.

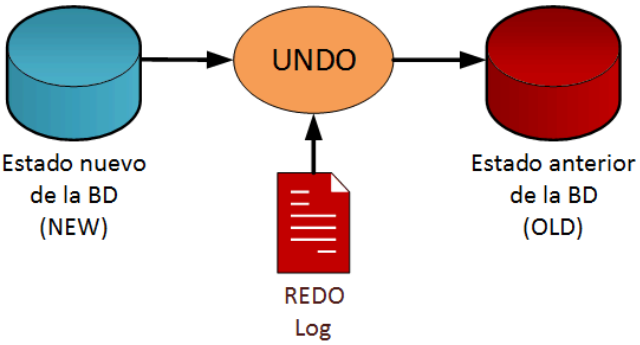


- Se emplea únicamente la información almacenada en los Online Redo Logs para realizar esta sincronización (ojo: no se usan archive redo logs).
- Instance recovery ocurre de forma automática al intentar abrir una base de datos.
- Instance recovery puede requerir de 2 operaciones fundamentales:

Rolling forward:
Consiste en aplicar cambios contenidos en los Online Redo Logs a los Data files ***Protocolo REDO***



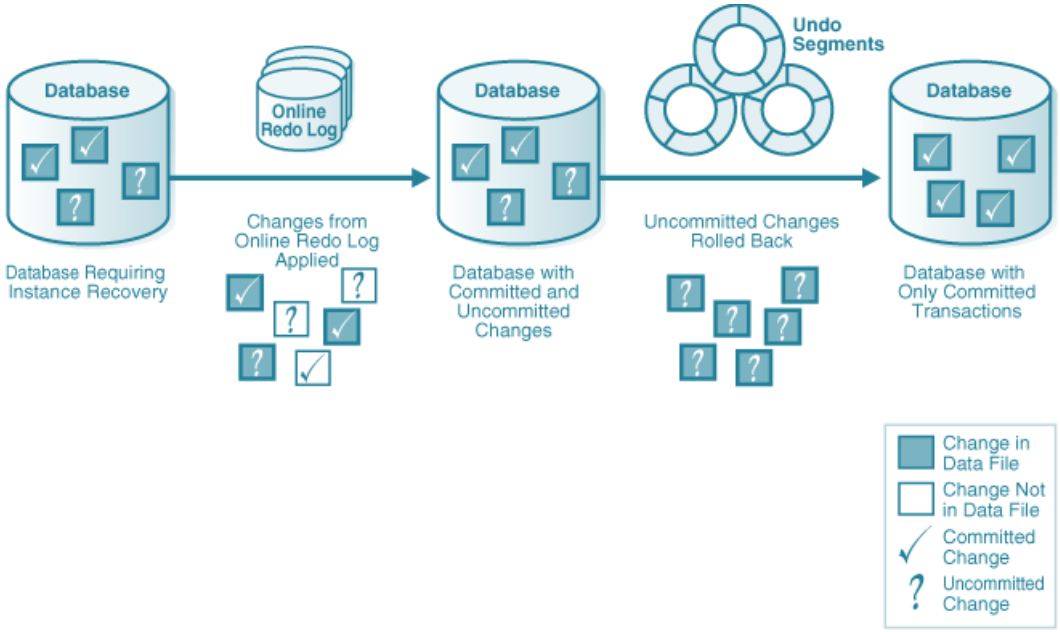
Rolling back:
Cambios que fueron aplicados a los data files sin haber sido confirmados deben regresar a su estado original. ***Protocolo UNDO***



8.7.2.1. Fases en un proceso de Instance recovery.

En un proceso de Instance recovery se aplican las siguientes etapas:

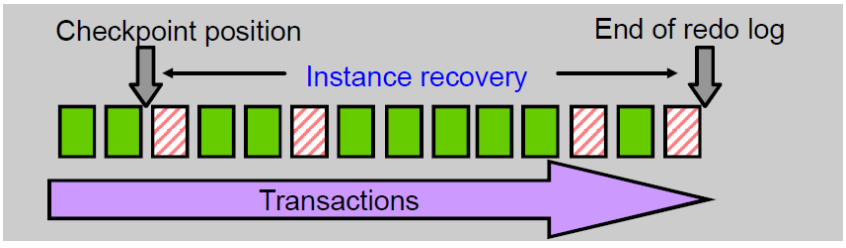
1. Se invoca la instrucción startup
2. Se detecta desincronización de archivo
3. Se aplica una operación Roll forward (protocolo REDO) tanto para transacciones confirmadas como para no confirmadas.
4. La base de datos es abierta.
5. Se aplica una operación Roll back (protocolo UNDO).
6. Los datos contenidos en el DB buffer cache que contiene el resultado del proceso de recovery en su momento serán sincronizados con Data files.



- Para que una base de datos pueda abrirse, el SCN contenido en los headers de los data files deben corresponder con en SCN actual que es almacenado en el control file.
- Si estos números no coinciden, la instancia aplica los cambios almacenados en los Redo logs de forma secuencial para ***rehacer*** las transacciones hasta que estos SCN coincidan, justo hasta el punto en el que ocurrió la falla (protocolo REDO)
- Este proceso incluye la aplicación de cambios que pertenecen a transacciones que aún no han hecho `commit`.
- Una vez que la base de datos se abre, todas estas transacciones se les aplica una operación de `rollback` (protocolo undo).

8.7.2.2. Importancia de un checkpoint en un instance recovery.

- Las operaciones descritas anteriormente se aplican a partir del último checkpoint realizado.
- Recordando, en los Redo Logs, se almacena la posición de este último checkpoint.
- Recordando, todos los cambios que están atrás o antes de esta última marca de checkpoint corresponden a cambios confirmados con un SCN asignados y guardados correctamente en los data files. Por lo tanto, el proceso de recovery solo considera a la última posición.



- Cada 3 segundos el proceso de background checkpoint guarda información en el archivo de control acerca de la posición del checkpoint grabado en los Online Redo logs.
- De esta forma la base de datos conoce el punto a partir del cual se debe aplicar un recovery en caso de una falla.

- El tiempo requerido para completar un proceso de Instance recovery depende de la cantidad de datos REDO que se tiene que aplicar a partir del último checkpoint y hasta el último SCN almacenado en el control file.
- Este tiempo puede ser especificado a través de un parámetro MTTR (Mean time to recover) en segundos.
- Otro aspecto a considerar es el tamaño de los grupos de Redo Logs.
- Para 2 grupos de Redo Logs, la distancia entre la posición del checkpoint y el final de los Redo Logs no puede ser más del 90% de la capacidad del grupo de redo log.
- El parámetro `fast_start_mttr_target` permite establecer la cantidad de segundos máxima en la que la recuperación de la instancia deberá realizarse. Hay que considerar los siguientes efectos:
    - Valores pequeños incrementa la actividad de operaciones I/O ¿por qué razón? Para reducir los tiempos, se requiere reducir la cantidad de datos Redo por aplicar. Esto implica más operaciones checkpoint y por lo tanto, más escrituras a los data files.
    - Valores grandes incrementa el tiempo que requiere la instancia para completar un proceso de Instance recovery posterior a la ocurrencia de una falla.
- Por default este parámetro está deshabilitado (0).

8.7.3. Media recovery

A nivel general, los archivos de la BD se consideran como:

- *Críticos*: Si la base de datos está abierta, esta será detenida y no podrá ser iniciada hasta que el daño sea reparado
    - Copias de un control file
    - Data file que es pate del tablespace `system`.
    - Data file que es parte del tablespace `undo` actualmente en uso.
- *No Críticos*: La base de datos permanece abierta o puede ser abierta a pesar de la correncia de la falla.
- Online Redo Logs multiplexados
- Archivos temporales
- Data files que no son parte del tablespace `system`.

Para cualquiera de los 2 casos, no debería existir razón para perder datos, y ser posible realizar una recuperación completa empleando *backups* y *Archive redo logs*.

- Como regla general, para recuperar cualquier data file se emplean las 2 operaciones siguientes:
    - Realizar una operación de `restore` (copiar el data file de un backup)
    - Aplicar `recovery` (aplicar Redo Logs para poner el archivo al día).
- Al proceso anterior se le conoce como ***complete media recovery.***
- Este proceso se realiza en 4 pasos:
    - Poner el archivo dañado como offline
    - Aplicar operación `restore`
    - Aplicar operación `recovery`
    - Poner el archivo online.

8.7.4. Recovery de datafiles en modo noarchivelog.

- En este caso, NO habrá manera de realizar un complete recovery.
- Solo se podrá realizar un ***incomplete recovery***, es decir, solo se podrá restaurar el datafile con el backup más reciente, pero no podrán ser aplicados los últimos cambios ya que los archive REDO logs no existen.

- Un datafile que no contiene los últimos cambios ¡NO podrá abrirse!
- Por lo tanto, la única opción en este caso es realizar una ***restauración de la base completa.***
- Esta restauración se tendrá que obtener de un ***whole offline backup.***
- Posterior a la restauración, la base de datos podrá ser abierta, pero los últimos cambios posteriores al backup, *¡se habrán perdido!*
- Existe otro error antes de que la BD pueda ser abierta: Esta no cuenta con su actual Online Redo Log debido a que estos nunca fueron respaldados.
- De lo anterior se debe ejecutar la instrucción en modo `mount`:`alter database clear logfile group <group_number>`.
- En RMAN los comandos que permiten realizar una restauración completa de la base de datos en modo `noarchivelog` son:

```
shutdown abort ;
startup mount ;
restore database;
alter database open resetlogs;
```

- Si existen backups incrementales, la secuencia de instrucciones es la siguiente:

```
shutdown abort ;
startup mount ;
restore database;
recover database noredo;
alter database open resetlogs;
```

- Esta instrucción adicional permite localizar todos los backups acumulativos y diferenciales incrementales nivel 1 para ser aplicados.
- La opción `noredo` es importante para indicarle a RMAN que no intente aplicar Redo Logs debido a que la BD no está en modo archivelog.

8.7.5. Incomplete recovery

- Como se comentó anteriormente, una recuperación incompleta ocurre cuando se detecta la inexistencia de algún archive redo log o cuando todas las copias del actual Online Redo log se han perdido.
- En algunos escenarios se pude decidir realizar un recovery incompleto de forma ***deliberada***.
- Lo anterior puede generarse debido a errores cometidos por el usuario:
    - Transacciones confirmadas con cambios no deseados (`delete` sin `where`, etc.).
- Posterior a la ocurrencia de un error del usuario, es posible realizar un restore completo de la BD hasta el punto justo antes de la ocurrencia del error cometido por el usuario.
- Esto es posible, pero, si existen otros cambios que son correctos posterior al error del usario, estos cambios se perderán.
    - La alternativa para evitar esta situación es emplear ***flashback technologies***.
    - Esta técnica permite recuperarse de un error del usuario sin tener que realizar un restore incompleto.

8.7.5.1. Recovery incompleto de un control file.

- En temas anteriores se revisó el ejercicio para realizar una recuperación del archivo de control.
- Si por alguna razón ese proceso no es posible, o si se desea recuperar el archivo en otro punto o estado diferente de la BD, se puede realizar un recovery incompleto.
- Un ejemplo de lo anterior es el borrado accidental de un tablesplace. El último control file no contendrá información de dicho tablespace.
- Los 4 pasos para realizar esta actividad son:
    - Iniciar en modo mount
    - Restaurar todos los datafiles
    - Aplicar recover hasta cierto punto
    - Abrir la BD haciendo reset de los REDO logs.
- Para aplicar el tercer paso se tienen las siguientes opciones:
    - Hasta un punto en el tiempo
    - Hasta un determinado System Change Number (SCN).
    - Hasta un determinado número de secuencia de un Redo Log.

Para realizar alguna de estas 3 acciones se requiere aplicar archive redo log y Redo Logs de ser necesario.

Ejemplo:

```
run {
  startup mount;
  set until time = "to_date('27-10-08 10:00:00','dd-mm-yy hh24:mi:ss')";
  restore database;
  recover database;
  alter database open resetlogs;
}
```

Otras herramientas para realizar protección de datos:

Backup and Recovery Objective	Recovery Time Objective (RTO)	Oracle Solution
Physical data protection	Hours/Days	Recovery Manager Oracle Secure Backup
Logical data protection	Minutes/Hours	Flashback Technologies
Recovery analysis	Minimize time for problem identification and recovery planning	Data Recovery Advisor

Disaster Recovery Objective	Recovery Time Objective (RTO)	Oracle Solution
Physical data protection	Seconds/Minutes	Data Guard Active Data Guard

8.8. OPERACIONES RECOVER Y RESTORE CON RMAN

RMAN puede ayudar a realizar estas 2 actividades ofreciendo varios beneficios como son automatización, simplicidad y reducción de posible error o falla.

- Existe otro componente llamado *Data Recovery Advisor* (DRA) encargado de diagnosticar problemas así como proporcionar recomendaciones o sugerencias para resolverlos.

8.8.1. Health Monitor

- Formado por un conjunto de validaciones que se ejecutan de forma automática cuando ocurre alguna condición de error, o de forma manual a partir de la ejecución de ciertos comandos.
- El resultado de estas validaciones no son almacenadas en la BD. Se emplea el sistema de archivos. Esto derivado a que posterior a la ocurrencia de una falla, la BD no necesariamente estará disponible.
- Por lo anterior se requiere de la existencia de un repositorio externo llamado ***Automatic Diagnostic Repository (ADR)***. Su ubicación depende del valor del parámetro `diagnostic_dest`.

Ejercicio 3.

En este ejercicio se hará uso del DRA para diagnosticar y corregir problemas. Incluir el resultado de cada operación.

A. Entrar a la interfaz de RMAN.

```
rman target /
```

B. Confirmar que existe un whole full backup para el tablespace sysaux.

```
list backup of tablespace sysaux;
```

C. En caso de no existir, crearlo:

```
backup as backupset tablespace sysaux;
```

D. Salir de RMAN y detener la instancia.
E. Provocar una falla. Eliminar manualmente el archivo dbf que corresponde al tablespace sysaux.
F. Intentar levantar la instancia con el comando `startup`.
G. El comando anterior fallará debido a la pérdida del archivo dejando a la instancia en modo `mount`.

```
ORA-01157: cannot identify/lock data file 2 - see DBWR trace file
ORA-01110: data file 2: '/u01/app/oracle/oradata/JRCBD2/sysaux01.dbf'
```

H. Entrar nuevamente a la interface RMAN, ejecutar el siguiente comando para diagnosticar el problema:

```
list failure
```

I. El comando anterior mostrará información en la que se indica la inexistencia de un datafile.

```
RMAN> list failure; using target database control file instead of recovery
catalog Database Role: PRIMARY

List of Database Failures
=========================

Failure ID Priority Status Time Detected Summary
---------- -------- --------- ------------- -------
242 HIGH OPEN 03-NOV-19 One or more non-system datafiles are missing
```

J. Generar sugerencias para resolver el problema:

```
advise failure;
```

K. La instrucción anterior va a sugerir una operación de `restore` y `recover` de un datafile. Se generará un script el cual se podrá ejecutar para reparar la falla.

```
RMAN> advise failure;

Database Role: PRIMARY

List of Database Failures
=========================

Failure ID Priority Status Time Detected Summary
---------- -------- --------- ------------- -------
242 HIGH OPEN 03-NOV-19 One or more non-system datafiles are missing

analyzing automatic repair options; this may take some time
allocated channel: ORA_DISK_1 channel ORA_DISK_1: SID=383 device type=DISK
analyzing automatic repair options complete

Mandatory Manual Actions
========================
no manual actions available

Optional Manual Actions
=======================
1. If file /u01/app/oracle/oradata/JRCBD2/sysaux01.dbf was unintentionally renamed
or moved, restore it

Automated Repair Options
========================
Option Repair Description
------ ------------------
1 NOARCHIVELOG mode restore datafile 2
Strategy: The repair includes complete media recovery with no data loss
Repair script: /u01/app/oracle/diag/rdbms/jrcbd2/jrcbd2/hm/reco_2546202009.hm
```

Contenido del script:

```
# NOARCHIVELOG mode restore datafile
restore datafile 2;
recover datafile 2;
sql 'alter database datafile 2 online';
```