



Servicio Andaluz de Salud
CONSEJERÍA DE SALUD

*Oficina Técnica para la Gestión y Supervisión de
Servicios TIC
Subdirección de Tecnologías de la Información*

Best Practices Flashback

*Referencia documento:
InfV5_JASAS_Flashback_BestPractices_TestingEnv_v910.doc
Fecha: 16 de junio de 2018
Versión: 9.1.0*

Registro de Cambios

Fecha	Autor	Versión	Notas
16 de Julio de 2015	Pablo Fernández Riejos	6.1	Revisión de Julio de 2015
16 de Diciembre de 2015	Pablo Fernández Riejos	6.2	Revisión de Diciembre de 2015
16 de Junio de 2016	Pablo Fernández Riejos	7.1	Revisión de Junio 2016
16 de Noviembre de 2016	Pablo Fernández Riejos	7.2	Revisión de Noviembre de 2016
16 de Junio de 2017	Pablo Fernández Riejos	8.1	Revisión de Junio de 2017
16 de Noviembre de 2017	Pablo Fernández Riejos	8.2	Revisión de Noviembre de 2017
16 de Junio de 2018	Pablo Fernández Riejos	9.1	Revisión de Junio de 2018
16 de Noviembre de 2018	Jonathan Ortiz	9.2	Revisión de Noviembre de 2018

Revisiones

Nombre	Role
Jonathan Ortiz	ACS Service Engineer
Gregorio Adame	ACS Service Engineer
José María Gómez	ACS Service TAM

Distribución

Copia	Nombre	Empresa
1	Subdirección de Tecnologías de la Información	Consejería de Salud, Junta de Andalucía
2	Servicio de Coordinación de Informática de la Consejería de Hacienda y Administración Pública	Consejería de Hacienda y Administración Pública, Junta de Andalucía

Índice de Contenidos

CONTROL DE CAMBIOS	4
INTRODUCCIÓN	5
FLASHBACK.....	6
<i>Parametrización para uso FLASHBACK en base datos</i>	6
<i>Flashback Database</i>	8
<i>Flashback Drop</i>	9
<i>Flashback Table</i>	11
<i>Flashback Query</i>	12
BEST PRACTICES FLASHBACK.....	14
<i>Best practices de configuración</i>	14
<i>Best practices de operación</i>	16
<i>Notas de rendimiento</i>	17
ESCENARIOS DE RECUPERACIÓN CON FUNCIONALIDAD FLASHBACK.....	18
<i>Recuperación de una tabla borrada (con DROP) con Flashback Drop</i>	18
<i>Recuperación de la filas de una tabla a un instante anterior</i>	19

Control de cambios

Cambio	Descripción	Página
	No se realizan cambios en esta versión	

Introducción

En el presente informe se presenta la funcionalidad Oracle Flashback, listando las características, tipos y best practices.

Flashback

A partir de versión de base de datos Oracle 10g Release 2, se libera la funcionalidad Oracle Flashback, la tecnología flashback se compone de las siguientes partes:

- Flashback Database
- Flashback Drop
- Flashback Table
- Flashback Versions Query
- Flashback Transaction Query

Flashback Table, Flashback Query, Flashback Transaction Query y Flashback Version Query consultan los datos necesarios para realizar flashback de los datos en los segmentos de undo, esto es, registros de los efectos de cada actualización en una base de datos Oracle y los valores sobreescritos en la actualización, usados en un primer momento para el propósito de proporcionar lecturas consistentes para las sentencias SQL y para realizar rollbacks de las transacciones, los registros de undo contienen la información necesaria para reconstruirlos datos puesto que los mantienen en un tiempo pasado así como todos los cambios desde dicho momento pasado.

Se dispone además de Flashback Database y Flashback Drop, hay dos diferencias principales de estas tecnologías de flashback y las comentadas en el párrafo anterior. Flashback Database consulta los datos de imágenes anteriores en los flashback logs, a diferencia de las tecnologías flashback que consultan los datos de los segmentos de undo. También Flashback Drop es construido sobre un nuevo mecanismo denominado Recycle Bin, el cual es usado por Oracle para gestionar los objetos de base de datos borrados hasta que el espacio ocupado por el objeto borrado sea reclamado para almacenar un nuevo objeto.

Parametrización para uso FLASHBACK en base datos

La tecnología flashback de base de datos permite la recuperación rápida de datos ante diferentes escenarios de recuperación y de forma más sencilla que aplicando recuperación con RMAN.

Toda política de backup y recuperación tiene que tener en cuenta a la tecnología FLASHBACK como método de recuperación ante los escenarios de recuperación presentados a continuación, sobre los que se aplicarían las distintas tecnologías de flashback.

Nivel de Objeto	Escenario de recuperación	Tecnología Flashback	Origen de la información
Base de datos	Truncate table; Cambios no deseados en tablas	Database	Flashback logs
A o Tabla n t i n Transacción	Drop table	Drop	Recycle bin
	Actualización de datos de forma no correcta	Tabla	Datos undo
	Comparación datos actuales con datos antiguos	Query	Datos undo
	Comparación de versiones de una fila	Version	Datos undo
Transacción	Investigación de estados de los datos históricamente	Transaction	Datos undo

ción se presenta la parametrización necesaria en la base de datos para la configuración de las distintas tecnologías flashback.

Tecnología Flashback	Configuración
Database	<pre>SQL> SHUTDOWN IMMEDIATE; SQL> STARTUP MOUNT EXCLUSIVE; SQL> ALTER SYSTEM SET DB_FLASHBACK_RETENTION_TARGET=2880 SCOPE=BOTH; -- El tiempo se especifica en minutos SQL> ALTER DATABASE FLASHBACK ON; SQL> ALTER DATABASE OPEN;</pre>
Drop	<pre>ALTER SYSTEM SET recyclebin=ON scope=BOTH</pre>
Tabla	<p>Usar el parámetro UNDO_RETENTION para especificar en segundos la cantidad de información commit que será retenida en la base de datos.</p> <p>Además se habilitará ROW MOVEMENT en aquellas tablas sobre las que se desea que se permita FLASHBACK TABLE.</p> <pre>ALTER TABLE <table_name> ENABLE ROW MOVEMENT;</pre>
Query	Usar el parámetro UNDO_RETENTION para especificar en segundos la cantidad de información commit que será retenida en la base de datos.
Version	Usar el parámetro UNDO_RETENTION para especificar en segundos la cantidad de información commit que será retenida en la base de datos.
Transaction	Usar el parámetro UNDO_RETENTION para especificar en segundos la cantidad de información commit que será retenida en la base de datos.

Flashback Database

Permite rebobinar la base de datos completa a un punto en el tiempo en el pasado. Es usado como sustituto de una recuperación incompleta para la base de datos entera. El principal beneficio de Oracle Flashback Database sobre la recuperación incompleta de la base de datos es que Flashback Database es mucho más rápido y eficiente, puesto que Flashback Database no se base en los segmentos de undo sino en los flashback logs.

Si Flashback Database está habilitado, los flashback logs son almacenados en la Flash Recovery Area (FRA).

Los Flashback Logs son escritos de forma secuencial durante el funcionamiento normal de la base de datos, y no son archivados, Oracle de forma automática, crea, borra y redimensiona los Flashback logs en la FRA. El administrador de base de datos sólo necesita monitorizar el rendimiento de la base de datos (al activar la recogida de los flashback logs) y cuanto espacio reservar en la FRA para los flashback logs.

El tiempo que se tarda en hacer Flashback de una base de datos es proporcional al tiempo que se necesite rebobinar, pero más que el tiempo que se tardaría en restaurar y recuperar una base de datos completa. Las imágenes anteriores en los flashback logs son sólo usadas para restaurar la base de datos a un punto en el tiempo pasado, y se usa forward recovery es usado para poner la base de datos en un estado consistente en el tiempo pasado. Oracle pondrá los datafiles al punto previo en el tiempo, pero no los ficheros auxiliares, como los ficheros de parámetros.

DB_FLASHBACK_RETENTION_TARGET, es el parámetro cuyo valor determina hasta que momento en el pasado puede recuperarse la base de datos con un flashback database, el valor se determina en minutos.

La configuración del parámetro DB_FLASHBACK_RETENTION_TARGET determina, de forma indirecta, cuantos datos de flashback log mantendrá la base de datos. El tamaño de los flashback logs generados por la base de datos para un periodo de tiempo puede variar considerablemente, dependiendo de la carga específica de la base de datos. Usando la vista v\$flashback_database_log puede estimarse cuanto espacio añadir a la FRA para los flashback logs. Después de haber habilitado Flashbask Database, el parámetro db_flashback_retention_target y dejar a la base de datos ejecutándose de forma normal durante un periodo de tiempo, puede consultarse el espacio estimado de flashback.

```
SQL> SELECT ESTIMATED_FLASHBACK_SIZE FROM  
V$FLASHBACK_DATABASE_LOG;
```

Prerrequisitos de uso de Flashback Database

- Tienen que usarse los privilegios de SYSDBA.
- La FRA tiene que estar configurada en la base de datos.

- La base de datos tiene que haberse puesto en modo FLASHBACK mediante la sentencia ALTER DATABASE FLASHBACK ON, a no ser que se esté rebobinando a un punto de restauración garantizado.
- La base de datos estará montada pero no abierta.

Limitaciones de Flashback Database

- Flashback Database puede solamente deshacer los cambios de los datafiles que se han realizado por una base de datos Oracle. No se pueden reparar fallos en los discos o recuperar borrados accidentales de los datafiles.
- No puede usarse Flashback Database para deshacer una operación de shrink de datafile.
- Si el fichero de control de la base de datos es restaurado desde un backup o recreado, todos los flashback logs acumulados se pierden. Tampoco puede usarse Flashback Database para volver a un punto de restauración después de la recreación de un controlfile.
- Cuando se usa Flashback Database a un tiempo objetivo, durante el cual se realizaron operaciones en modo NOLOGGING, **es probable que se produzcan corrupción de bloques** en los objetos de base de datos y datafiles afectados por la operación NOLOGGING.
- Shrinking un datafile o borrar un tablespace pueden impedir que se puede aplicar el flashback database.
- Si es posible, evita usar Flashback Database con un tiempo pasado objetivo o SCN que coincida con un operación NOLOGGING. También, como recomendación, siempre realiza un full o incremental backup de los datafiles afectados inmediatamente después de cualquier operación NOLOGGING.

Flashback Drop

Esta funcionalidad proporciona una manera para restaurar una tabla que ha sido borrada accidentalmente. Flashback Drop proporciona un método seguro cuando se borran objetos en bases de datos 10g. Cuando un usuario borra una tabla, Oracle coloca dicho objeto en la recycle bin. Los objetos en la recycle bin permanecerán allí hasta que el usuario decida borrarlos de forma permanente o hasta que las limitaciones de espacio empiecen a producirse en el tablespace donde se encuentre la tabla. La recycle bin es un contenedor virtual donde todos los objetos borrados existen. Los usuarios podrán ver el contenido de la recycle bin y recuperar las tablas borradas y sus objetos dependientes.

La funcionalidad Flashback Drop estará disponible justo después del reinicio de la base de datos.

Prerrequisitos

- Para realizar un flashback de una tabla justo antes de la operación DROP TABLE, solamente se necesitan los privilegios de borrado sobre la tabla que se borró.

Limitaciones y restricciones de uso de Flashback Drop

- La funcionalidad recycle bin no está disponible para tablespace de sistemas, sólo para tablespace de no-sistemas y gestionados localmente. Si la tabla está en un tablespace de no-sistema y gestionado localmente, pero tiene uno o más objetos dependientes en un tablespace gestionado por diccionario, dicho objeto no podrá recuperarse de la recycle bin.
- No hay una cantidad de espacio fija reservada en la recycle bin, y no hay garantía de cuando tiempo los objetos borrados permanecerán en la recycle bin. Dependiendo de la actividad del sistema, un objeto borrado puede permanecer en la recycle bin, bien segundos, bien meses.
- Mientras que Oracle permite las consultas sobre objetos almacenados en las recycle bin, no pueden usarse sentencias DML o DDL sobre los objetos de la recycle bin, dichas acciones fallarán con el mensaje

ORA-38301: can not perform DDL/DML over objects in Recycle Bin

Y en el fichero de alertas de la base de datos se encontrará el mensaje

performing DML/DDDL operation over object in bin.
performing DML/DDDL operation over object in bin.

- Puede realizarse Flashback Query sobre las tablas en la recycle bin, pero sólo usando el nombre asociado en la recycle bin. No puede usarse el nombre original de la tabla.
- Una tabla y sus objetos dependientes (índices (excepto los bitmap join index), LOB segments, nested tables, triggers, constraints...) van a la recycle bin juntos cuando se borra una tabla. De tal manera, que cuando ejecuta un Flashback Drop, los objetos son generados de nuevo todos juntos, es posible, sin embargo que algunos objetos dependientes como los índices puedan haber sido reclamados por falta de espacio, en dichos casos, los objetos dependientes que han sido reclamados no serán recuperados de la recycle bin.
- Debido a cuestiones de seguridad, tablas que tengan activada la auditoría Fine Grained (FGA) y políticas de base de datos virtual privada (VPD) no serán almacenadas en la recycle bin si son borradas.
- Las tablas particionadas de index-organized no son almacenadas en la recycle bin.
- La recycle bin no preserva la restricciones de integridad de la tabla. Si una tabla tenía una integridad referencial antes que fuera borrada, tras el flashback drop tendrá que recrearse dicha integridad referencial.
- Cuando se borra una tabla con drop table, todas las vistas log materializadas definidas sobre la tabla son también borradas pero no se colocarán en la recycle bin. Por tanto, sobre las vistas logs materializadas no puede aplicarse flashback table junto con la tabla.
- Cuando se borra una tabla con drop table, cualquier índice sobre la tabla es borrado y puesto en la recycle bin junto con la tabla, sin embargo si debido a presiones de espacio se usa el espacio de la recycle bin en la que se encuentran los

índices, cuando se aplique flashback table sobre la tabla, los índices no serán recuperados.

- No puede aplicarse flashback table sobre una tabla que ha sido purgada, bien por un usuario, bien por un proceso de reclamación de espacio.
- La base de datos Oracle recupera todos los índices definidos sobre la tabla recuperada desde la recycle bin excepto para los bitmap join index y los domain index.

Flashback Table

El flashback table tiene lugar realizando rollback de los cambios realizados en la tabla o tablas y sus objetos dependientes, como los índices. Tener en cuenta que Flashback Table es diferente de Flashback Drop: Flashback Table deshace los cambios de transacciones recientes a una tabla existente a diferencia de Flashback Drop que realiza una recuperación de una tabla sobre la que se ha aplicado un Flashback Drop; Flashback Table usa los datos de los segmentos de undo para la recuperación y flashback drop usa la recycle bin.

El comando flashback table pone una o más tablas a un punto en el tiempo antes de cualquier número de corrupciones lógicas hayan ocurrido en las tablas. Para poder realizar un flashback table sobre las tablas, tiene que habilitarse row movement para dichas tablas; puesto que las operaciones DML suelen poner la tabla en su primer estado, los ROWIDs en la tabla cambian. Por tanto, Flashback Table no es una opción viable para las aplicaciones que dependen que el ROWID de la tabla permanezca constante.

Antes de realizar una operación de flashback table, primero debe habilitarse el row movement sobre las tablas afectadas.

```
SQL> alter table <table_name> enable row movement;
```

Prerrequisitos uso Flashback Table

- Para aplicar flashback table a un SCN o timestamp anterior, el usuario de base de datos debe tener el privilegio de objetos FLASHBACK TABLE sobre la tabla o el privilegio de sistema FLASHBACK ANY TABLE, además de los privilegios SELECT, INSERT, DELETE y ALTER sobre la tabla.
- Row movement tiene que estar habilitado para todas las tablas en la lista de tablas sobre la que se va a aplicar el flashback table, a no ser que se esté ejecutando sobre la tabla flashback TO BEFORE DROP. Ese operación es llamada una operación flashback drop, y usa los datos borrados almacenados en la recycle bin en lugar de los datos de undo.
- Para realizar flashback de una tabla a un punto de restauración, el usuario tendrá los privilegios de sistema SELECT ANY DICTIONARY o FLASHBACK ANY TABLE o bien el rol SELECT_CATALOG_ROLE.

Limitaciones y restricciones en el uso de Flashback Table

- Las operaciones flashback table no son válidas para los siguientes tipos de objetos: tablas que son partes de un cluster, vistas materializadas, tablas AQ, tablas estáticas del diccionario de oracle, tablas de sistema, tablas remotas, objetos tablas, nested tables, o particiones y subparticiones.
- Las siguientes operaciones DDL cambian la estructura de una tabla, por lo que a continuación de dicha operación no podrá realizarse un flashback tabla anterior en el tiempo a dicha operación: upgrading, mover o truncar una tabla, añadir una restricción sobre una tabla, añadir una tabla al cluster, modificar o borrar una columna; añadir, borrar, merging, splitting, coalescing, o truncar una partición o subpartición (con la excepción de añadir una partición por rango)
- No se puede realizar un ROLLBACK de una sentencia FLASHBACK TABLE. Sin embargo, puede realizarse un FLASHBACK TABLE y especificarse un momento justo antes al actual, por lo tanto es recomendable anotar el SCN de la tabla antes de realizar una operación flashback table.

Flashback Query

Flashback Query, disponible desde versión 9i, ha sido mejorado desde 10gR2 para incluir dos nuevos tipos de consultas: Flashback Version Query y Flashback Transaction Query

Flashback Version Query permite a un usuario o al DBA ver todas las versiones de los valores de las filas de una tabla entre dos instantes de tiempo, y con Flashback Transaction Query pueden verse todas las transacciones que han cambiado una fila entre dos instantes de tiempo.

Flashback Versions Query proporciona un método fácil para mostrar todas las versiones de todas las filas de una tabla entre dos SCN o dos timestamp, si las filas fueran insertadas, borradas o actualizadas. Incluso si una fila fue borrada o reinsertada varias veces, todos esos cambios estarán disponibles para la operación Flashback Versions Query.

Flashback Transaction Query, en contraste, analiza el histórico de los cambios de la tabla basados en el transaction ID. Usando, Flashback Versions Query, puede encontrarse que transacciones cambiaron por ejemplo la información del salario, pero no se conocerá quién hizo el cambio.

Flashback Transaction Query usa los datos de la vista del diccionario FLASHBACK_TRANSACTION_QUERY para recuperar la información de la transacción para todas las tablas envueltas en la transacción. La vista proporciona las sentencias SQL que pueden usarse para deshacer los cambios realizados por una transacción particular.

Prerrequisitos:

- Para realizar Oracle Flashback Version Query, el usuario tendrá los privilegios SELECT sobre las tablas, FLASHBACK TABLE sobre las tablas o bien FLASHBACK ANY TABLE.
- Para realizar Oracle Flashback Transaction Query, el usuario tendrá FLASHBACK_TABLE, o bien SELECT ANY TRANSACTION.

- Para el uso del paquete DBMS_FLASHBACK, habrá que otorgar privilegios de EXECUTE sobre DBMS_FLASHBACK, este paquete proporciona la misma funcionalidad que Flashback Query, pero Flashback Query es en algunas ocasiones más conveniente.
- Para usar Flashback Transaction Query en Oracle 10g, la base de datos estará en ejecución con el parámetro compatible mayor a 10.0, y tener supplemental logging habilitado:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

Limitaciones y restricciones sobre Flashback Query

- No pueden especificarse una subquery en la expresión de la cláusula AS OF
- No puede usarse la cláusula VERSIONS en flashback queries sobre temporales, tablas externas, tablas fijas o tablas que sean partes de un cluster.
- Flashback Query no deshace ningún cambio. Sólo es un mecanismo de consulta.
- Flashback Query no indica que ha cambiado, para eso usar LogMiner.
- Flashback Query no indicará información sobre tablas anteriormente a una operación DDL.

Best Practices Flashback

A continuación se listan las funcionalidades de flashback y las versiones de base de datos en las que han aparecido:

- Oracle 10g (2003)
 - FEATURE: Release inicial en la que aparece la capacidad de flashback database
- Oracle 10g Release 2
 - FEATURE: Normal y Guaranteed Restore Points
 - FEATURE: Flashback database across "OPEN RESETLOGS"
- Oracle 11g Release 1
 - FEATURES: Snapshot standby, Data Guard Fast-Start Failover con reinstanciación de vieja primaria con flashback
 - Funcionalidad de Rendimiento: la funcionalidad Block new optimization fue introducida para entornos single-instance (no para RAC) y redujo el impacto del rendimiento para operaciones de insert y direct load. La optimización con flashback block new optimization realiza un bypass del logging de las operaciones para los insert a nuevos bloques. Para las operaciones direct write, la optimización puede evitar lecturas adicionales de bloques físicos al buffer cache.
- Oracle 11g Release 2
 - FEATURE: Flashback database puede habilitarse con la base de datos abierta.
 - PERFORMANCE FEATURE: la funcionalidad Block new optimization está disponible para RAC
 - PERFORMANCE FEATURE: El overhead inicial cuando se reservan los flashback logs antes de alcanzar DB_FLASHBACK_RETENTION_TARGET ha sido reducido y en algunos casos eliminados intentando reservar previamente antes de bloquear cualquier proceso.
 - PERFORMANCE FEATURE: Mejoras en la lectura de flashback logs durante las operaciones de flashback.

Best practices de configuración

1. **Configurar DB_FLASHBACK_RETENTION_TARGET correctamente.** Configurar el parámetro DB_FLASHBACK_RETENTION_TARGET al valor mayor indicado por algunas de las condiciones siguientes:
 - Para conseguir que flashback database realice la restauración de una base de datos primaria después de un DataGuard Failover, en la mayoría de los casos el parámetro DB_FLASHBACK_RETENTION_TARGET debe ser configurado a un mínimo de 60 minutos.
 - Considera los casos donde haya múltiples pérdidas de servicio que podrían resultar en aumentos en el tiempo de transporte o pérdidas de comunicación entre las bases de datos primaria y standby, para dichos casos configurar el parámetro DB_FLASHBACK_RETENTION_TARGET a un valor igual a 60 minutos más el tiempo de la pérdida de comunicación máximo que podría permitirse. Esto asegura que una base de datos primaria con errores puede

aplicar flashback database al SCN previo al SCN en el que la standby pasó a ser primaria

- Si se usa Flashback Database para una recuperación a un punto en el tiempo por un error de usuario o corrupción lógica, configura DB_FLASHBACK_RETENTION_TARGET a un valor igual al tiempo más lejano en el tiempo al que se desea recuperar ante este tipo de fallos.
 - Configurar el parámetro DB_FLASHBACK_RETENTION_TARGET igual tanto en la primaria como en la standby
2. **Tamaño del Fast Recovery Area (FRA).** Asegurar que el fast recovery area ha reservado espacio suficiente para almacenar los flashback logs para conseguir el target retention time y para picos de cargas masivas. Por regla general es que el volumen de los flashback logs es aproximadamente del mismo orden de magnitud que la generación de redo, por lo tanto, se recomienda usar la siguiente fórmula para dar una aproximación del tamaño.

$\text{Target FRA} = \text{Current FRA} + \text{DB_FLASHBACK_RETENTION_TARGET} \times 60 \times \text{Peak Redo Rate (MB/sec)}$
--

Otro método adicional para determinar el tamaño del fast recovery area es habilitar flashback database y permitir a la aplicación que se ejecuta por unas dos o tres horas y posteriormente consultar el valor de la columna V\$FLASHBACK_DATABASE_STAT.ESTIMATED_FLASHBACK_SIZE. Tener en cuenta que el DB_FLASHBACK_RETENTION_TARGET es un objetivo y no hay garantía que se pueda conseguir el flashback de la base de datos a ese punto.

En algunos casos en los que se produzca falta de espacio en la FRA en la que existen flashback logs, puede producirse que los flashback logs más antiguos sean borrados.

3. **Configurar suficiente ancho de banda I/O para FRA.** Si no existe suficiente ancho de banda de I/O en FRA con bases de datos con Flashback Database se indica con el evento de espera "FLASHBACK BUF FREE BY RVWR" y con una latencia de más de 30 ms en el "FLASHBACK LOG FILE WRITE" para operaciones de insert masivos. En general, las operaciones de flashback IO son de 1MB de tamaño y el total throughput de escritura será igual a la generación de redo si la base de datos tiene forcé logging habilitado. Para mayor simplicidad, la recomendación es configurar un almacenamiento compartido grande, teniendo los discos DATA en la porción externa de los discos de la LUN y RECO (para la FRA) en la porción interna de los discos de las LUNs
4. **Recomendación de parametrización de Log_buffer .** para dar a flashback database más espacio de buffer en la memoria. La recomendación es configurar log_buffer al valor máximo del que se especifique para dicha versión de base de datos.

La recomendación de Oracle es 64 MB para 32-bit systems y 256 MB para 64-bit systems

5. **Configura el parámetro `_DB_FLASHBACK_LOG_MIN_SIZE` = <redo log size>** en las versiones 11.2.0.2., desde versión 11.2.0.3 el parámetro oculto ya se configura como mínimo al valor del tamaño del redo log group el cual es el prescrito como best practice. Esta mejora no está disponible en versión Oracle 10g, por lo que podría producirse problemas de rendimiento en bases de datos 10g hasta que se alcance el `DB_FLASHBACK_RETENTION_TARGET`.
6. **Configurar `_DB_FLASHBACK_LOG_MIN_TOTAL_SPACE` =< projected flashback size> de forma temporal**, si se desea prealocar espacio flashback logs en la FRA para preveer el espacio necesario antes de una carga masiva y así la carga no se vea afectada por el tiempo necesario para reservar el espacio de los flashback logs.

Best practices de operación

1. Salva las estadísticas de rendimiento de la base de datos, bien con AWR o Statspack, antes y después de habilitar flashback database, de tal manera que pueda medirse el impacto de habilitar flashback database.
2. Monitorizar el espacio libre en la Fast Recovery Area (FRA).
3. Desde 11.2, puede habilitarse flashback database mientras la base de datos está abierta. Sin embargo esta operación puede fallar si no hay espacio de memoria contiguo suficiente. Es por ello, que se recomienda habilitar flashback database en estado mount.
4. Monitar el progreso de las operaciones flashback usando la vista `V$SESSION_LONGOPS`

```
select * from v$session_longops where opname like 'Flashback%';
```

5. Si se necesita más detalle de las operaciones de flashback database puede configurarse el parámetro oculto `_FLASHBACK_VERBOSE_INFO=TRUE` el cual generará trazas en el `DIAGNOSTIC_DEST`
6. Cuando se usa la funcionalidad flashback database para realizar pruebas en bases de datos de test que tienen que repetirse, se recomienda el uso de puntos de restauración garantizados solo sin activar explícitamente flashback database. Para minimizar el espacio usado y el rendimiento de flashback, seguir la siguiente recomendación.

```
Create Guaranteed Restore Point (GRP)
Execute test
loop

    Flashback database to GRP
    Open resetlogs
    Create new GRP
    Drop old GRP
    Execute test

End loop
```

Notas de rendimiento

Tras la implementación de las best practices de configuración y de operación, Oracle indica las siguientes observaciones de rendimiento cuando se habilita flashback database en la base de datos primaria o en la base de datos standby.

El impacto en el rendimiento en bases de datos primarias OLTP es normalmente de menos del 5%.

El impacto de insert masivos u operaciones de carga en la base de datos primaria es normalmente menos del 5% si la nueva optimización flashback block está configurada; en otro caso el impacto puede llegar al 40%.

Escenarios de Recuperación con funcionalidad Flashback

Recuperación de una tabla borrada (con DROP) con Flashback Drop

```
-- Creamos una tabla para el ejemplo
SQL> Create table TESTFLASH (id number);

SQL> Insert into TESTFLASH values (1);
SQL> Insert into TESTFLASH values (2);
SQL> Insert into TESTFLASH values (3);
SQL> Insert into TESTFLASH values (4);
SQL> Insert into TESTFLASH values (5);
SQL> Commit;

SQL> select * from TESTFLASH;

          ID
-----
          1
          2
          3
          4
          5

5 rows selected.

Consultemos la tabla en el presente schema
=====

SQL> select * from tab;

TNAME                TABTYPE  CLUSTERID
-----
TESTFLASH            TABLE

Supongamos, se produce un borrado accidental de la tabla
=====

SQL> drop table TESTFLASH;

Table dropped.

Vemos el estado de la tabla en el schema

SQL> select * from tab;

TNAME                TABTYPE  CLUSTERID
-----
BIN$04LhpcndanfgMAAAAAANPw==$0 TABLE

-- Mostramos el contenido de la recycle bin
SQL> show recyclebin

ORIGINAL NAME      RECYCLEBIN NAME      OBJECT TYPE  DROP TIME
-----
--
TESTFLASH         BIN$04LhpcndanfgMAAAAAANPw==$0 TABLE      2004-02-
16:21:13:31

Se observa en la recycle bin el nombre original de la tabla, así como el
nombre dentro de la recycle bin

-- Ejecutamos el comando para la recuperación de la tabla
```

```
SQL> FLASHBACK TABLE TESTFLASH TO BEFORE DROP;
```

```
FLASHBACK COMPLETE.
```

```
-- Consultamos de nuevo la table TESTFLASH
```

```
SQL> select * from TESTFLASH;
```

```
      ID
-----
      1
      2
      3
      4
      5
```

```
5 rows selected.
```

```
Consultamos la tabla en el schema del usuario actual
```

```
SQL> SELECT * FROM TAB;
```

```
TNAME                                TABTYPE  CLUSTERID
-----                                -
TESTFLASH                            TABLE
```

```
Tras el flashback de la tabla, la recycle bin estará vacía.
```

```
SQL> Show Recyclebin
```

Recuperación de la filas de una tabla a un instante anterior

```
-- Habilitamos row movement sobre la tabla (esto se realizará una sólo vez,
si y solo si no está habilitado previamente.
```

```
SQL> ALTER TABLE flash_test_table enable row movement;
```

```
Table altered.
```

```
-- Consultamos los datos de la tabla en el instance de tiempo que se realiza
la consulta
```

```
SQL> select * from flash_test_table;
```

```
      EMPNO EMPNAME
-----
      1 Kiran
      2 Scott
      3 Tiger
      4 Jeff
```

```
-- También es recomendable almacenar el SCN de la base de datos antes de
ejecutar el FLASHBACK TABLE
```

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

```
CURRENT_SCN
-----
332348
```

```
-- Con otro usuario, nos logamos en la base de datos y modificamos la tabla
flash_test_table
```

```
SQL> connect scott/tiger
```

```
Connected.
```

```
SQL> insert into flash_test_table values(5,'Jane');
```

```
1 row created.
```

```
SQL> insert into flash_test_table values(6,'John');

1 row created.

SQL> commit;

Commit complete.

-- Como SYSDBA volvemos a conectarnos a la base de datos para consultar el
nuevo SCN de la base de datos.

SQL> connect / as sysdba
Connected.

SQL> select current_scn from v$database;

CURRENT_SCN
-----
332376

-- Como scott damos marcha atrás a los cambios realizados sobre las filas de
las tablas.

SQL> connect scott/tiger
Connected.

SQL> select * from flash_test_table;

  EMPNO EMPNAME
-----
       1 Kiran
       2 Scott
       3 Tiger
       4 Jeff
       5 Jane
       6 John

6 rows selected.

SQL> flashback table flash_test_table to scn 332348;

Flashback complete.

SQL> select * from flash_test_table;

  EMPNO EMPNAME
-----
       1 Kiran
       2 Scott
       3 Tiger
       4 Jeff

-- También la operación de flashback puede realizarse a un SCN posterior

SQL> flashback table flash_test_table to scn 332376;

Flashback complete.

SQL> select * from flash_test_table;

  EMPNO EMPNAME
-----
       1 Kiran
       2 Scott
       3 Tiger
       4 Jeff
       5 Jane
       6 John

6 rows selected.
```