UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES "ACATLAN"

TÉCNICAS DE RESPALDO Y RECUPERACIÓN DE BASES DE DATOS IMPLEMENTADAS CON EL DBMS ORACLE.

U.N.A.M. FES ACATLAN
JUN. 22 2005
DEPTO. DE TÍTULOS
PROFESIONALES Y CERTIFICACIÓN
UNIDAD DE ADMINISTRACIÓN ESCOLAR

T E S I S

QUE PARA OBTENER EL TÍTULO DE :

LICENCIADO EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

P R E S E N T A :

RICARDO DOMÍNGUEZ ESQUERRO

ASESOR:
LIC. ALEJANDRO RUBIO PÉREZ

ACATLAN ESTADO DE MÉXICO, JUNIO 2005

m. 345723

Universidad Nacional Autónoma de México
UNAM
Biblioteca Central
Dirección General de Bibliotecas de la UNAM

UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.
NOMBRE: Ricardo Domínguez
FECHA: 24/VI/05
FIRMA:

Técnicas de respaldo y recuperación de bases de datos implementadas con el DBMS Oracle.
Versión 1.1.1

Presenta:
Ricardo Domínguez Esquerro
Número de Cuenta 9106785-2

Director de Tesis:
Lic. Alejandro Roberto Rubio Pérez

Agradecimiento

Muchas gracias Dios, porque me permitiste tener vida para poder sentir todo lo maravilloso que hay en ella.

Te agradezco mucho los padres que diste, mi papá Luis y a mi mamá Sara, que siempre me han guiado por el camino correcto sin importar lo que a ellos les costara, y además me diste a mi hermano Guillermo, que siempre ha sido mi mejor amigo y mi soporte en los momentos dificiles.

Me diste el coraje y las ganas para estudiar una carrera en la UNAM, y aparte me pusiste en la su mejor carrera MAC.

Me diste la oportunidad de conocer a gente que fue una influencia muy grande en mi persona como: Alan Garduño, Fernando Gonzalez , Judith Jaramillo y Juan Francisco Flores. Ellos han sido grandes ejemplos en mi vida profesional.

Y por ultimo solo quiero agradecerte que pusiste en mi camino a una persona que es y sera siempre parte de mi vida. Ella me enseño a amar con todo el corazón sin pedir nada a cambio. Espero algún volverte a ver Adriana.

Muchas, muchas gracias Dios.

Tabla de Contenido

Objetivo

El mantener una base de datos disponible y consistente en todo momento requiere de la capacidad de enfrentar diversas situaciones inesperadas en las que pudiera perderse información. El tema a tratar en este trabajo se enfoca al análisis de las diferentes técnicas de respaldo de una base de datos a fin de definir un método de recuperación de información que resulte adecuado a las necesidades del servicio, a partir de los distintos factores que inciden en un sistema, entre los que destacan el hardware, software, número de usuarios, tipo de disponibilidad, entre otros.

El presente trabajo conlleva el esfuerzo de intentar puntualizar, de manera metódica, la recuperación de una base de datos basada en el *DBMS* Oracle, en virtud de que ese sistema ha demostrado poseer las características óptimas para el respaldo de la información y su recuperación ante cualquier condición, además porque la Facultad de Estudios Superiores Campus Acatlán cuenta con el software de mérito, por lo que la aspiración personal del autor, es ofrecer mediante este esfuerzo, una breve referencia a quienes se interesen en el estudio de las bases de datos y en general a la comunidad estudiantil.

Introducción.

La competencia en el mercado dentro de cualquier ámbito comercial o profesional resulta desmesurada, lo cual se debe a la gran diversificación de los productos y servicios que ofrecen las diferentes compañías a sus clientes. Sin embargo, existe un factor que indudablemente marca la diferencia: la disponibilidad y consistencia de su información. Esta peculiaridad puede ser determinante en el desarrollo de la contienda.

La información es el elemento más importante dentro de cualquier compañía u organización. Con ella, se pueden definir estrategias, decisiones, soluciones a diferentes problemas, etcétera. Sin embargo, la posibilidad de una falla siempre está presente.

Dentro de cualquier computadora, la amplia variedad de hardware y software puede ocasionar pérdida de información. Desastres naturales, fallas en el suministro eléctrico (*v.gr. desconectar el cable de poder accidentalmente*), problemas con el software (*actualizaciones incorrectas*) y fallas en hardware, son difíciles de predecir. Especial mención merecen otras cuestiones que demandan ser consideradas, como la corrupción intencional y accidental de datos.

Cada aplicación conlleva requerimientos especiales. Algunas están diseñadas para la captura de información, como pueden ser las máquinas registradoras de un supermercado; otras están orientadas a la recuperación de información, tales como un típico datawarehouse. En los eventos de fallas de hardware y software, la habilidad para restaurar los datos después del siniestro es vital. Un respaldo o copia de seguridad, vincula la elaboración de una reproducción electrónica de los programas y la información, en un medio removible.

Un respaldo puede implicar la copia de todos los programas, datos y archivos de configuración en un dispositivo específico, como puede ser una cinta o un disco magnético.

La frecuencia y la magnitud de los respaldos dependen de la aplicación y las necesidades de los datos en el negocio. Por ejemplo, una base de datos que es alimentada cada tarde para una simple actualización, quizá no necesite ser respaldada. En contraste, ciertas aplicaciones no permiten la mínima pérdida de información; a manera de muestreo, un sistema bancario.

Cuando se decide cómo respaldar y recuperar la información de una base de datos se deben entender las necesidades del negocio. Esto debe incluir el tiempo en el que el equipo no podrá ser operado por mantenimiento o reparación; el tiempo que la compañía puede soportar sin poder accesar a los datos y la aceptación de la pérdida de información; si ésta puede reconstruirse o si se perdiera durante la falla. Cada uno de los escenarios tiene diferentes necesidades de respaldo y recuperación.

Asimismo la recuperación también necesita ser tomada en cuenta. El duplicar el hardware es demasiado costoso y el proceso de restauración desde una cinta es excesivamente lento. Estos procesos requieren consideraciones cuidadosas respecto de las aplicaciones del negocio.

A efecto de cumplir esta tarea, existe una persona denominada administrador de base de datos o DBA (*DataBase Administrator* por sus siglas en inglés), quien es responsable tanto del diseño y mantenimiento de la base de datos, como de la evaluación, selección e implementación del DBMS (*Data Base Management System*).

La principal responsabilidad del DBA es la habilidad de corregir, en el menor tiempo posible, cualesquiera eventualidades que se presentaran, tales como fallas en el hardware, software, la red o incluso el propio DBMS. Si algún error afectara el funcionamiento de éste, es prioritario recuperar a la brevedad la base de datos y regresar a la operación normal. La restauración de la base de datos podría proteger la información y a los usuarios del sistema de problemas innecesarios y evitar o reducir la probabilidad de tener que realizar de nueva cuenta el trabajo.

El presente trabajo está basado en el *DBMS* Oracle, ya que ha demostrado poseer las características óptimas para el respaldo de la información y su recuperación ante cualquier condición, además porque la Facultad de Estudios Superiores Campus Acatlán cuenta con el referido software, por lo que la aspiración personal del autor, es que el trabajo que aquí se ofrece, sirva en el futuro como referencia a la comunidad estudiantil.

Capítulo I Arquitectura del DBMS Oracle y sus bases de datos.

1.1 Definiciones de bases de datos.

1.1.1 Definición de base de datos.

El objetivo general de una base de datos es relacionar hechos y situaciones que previamente estaban separados en términos conceptuales o de método. Más específicamente, una base de datos es *"un conjunto de información interrelacionada, almacenada con la menor cantidad posible de redundancia*[1]*"*. Una base de datos debe estar orientada a varias aplicaciones.

Sólo resulta exitosa, si cumple con las necesidades de los usuarios. Esto radica en la raíz del diseño de las bases de datos. Por tanto, la presencia de especialistas temáticos es crucial.

[1] **JARAMILLO LÓPEZ**, Judith. *"Técnicas para la recuperación de información y búsqueda de texto en bases de datos relacionales"*. México, UNAM, 2001.

Una base de datos es la agrupación de informes sobre objetos así como de las relaciones entre ellos (*idea de sistema*). Una base de datos actúa como un modelo de la realidad, en un concepto amplio de modelo.

1.1.2 Definición de una base de datos "Oracle".

Se define a una base de datos Oracle, como una compilación de datos que son conformados en una unidad[2]. Su propósito general es almacenar y recuperar la información relacionada entre sí. Está formada por estructuras lógicas y estructuras físicas .

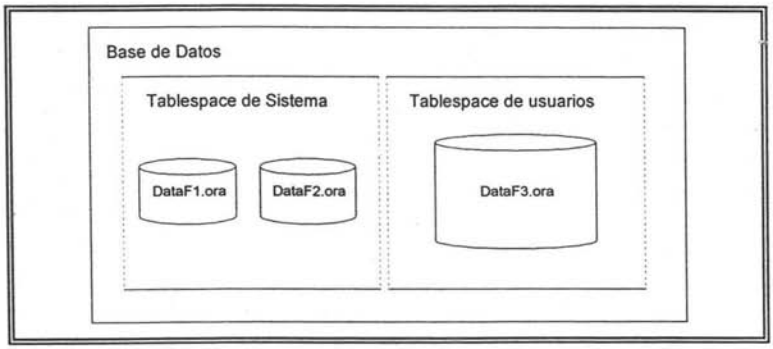


Figura 1.1 Una base de datos Oracle.

La relación entre las estructuras lógicas (*Tablespace*) y las físicas (*Datafiles*) se destaca en la figura 1.1. La conceptualización de ambas se ensayará en el punto número tres *"Arquitectura del manejador de bases de datos Oracle 8I"*.

[2] **BURLESON**, Donald Keith. "*High-performance oracle database applications*". Scottdale, Arizona: Coriolis, c 1996.

1.1.3 *Definición de una instancia Oracle.*

Cada vez que una base de datos Oracle es puesta en marcha, se le asigna un segmento de memoria denominado Área Global del Sistema *(SGA System Global Area,* por sus siglas en Inglés) en tanto que los procesos de background de Oracle son inicializados.

La SGA es un área de memoria cuya finalidad es compartir la información de la base de datos a sus usuarios. La combinación de los procesos de background y de los búffers de memoria se le denominan "instancia Oracle".

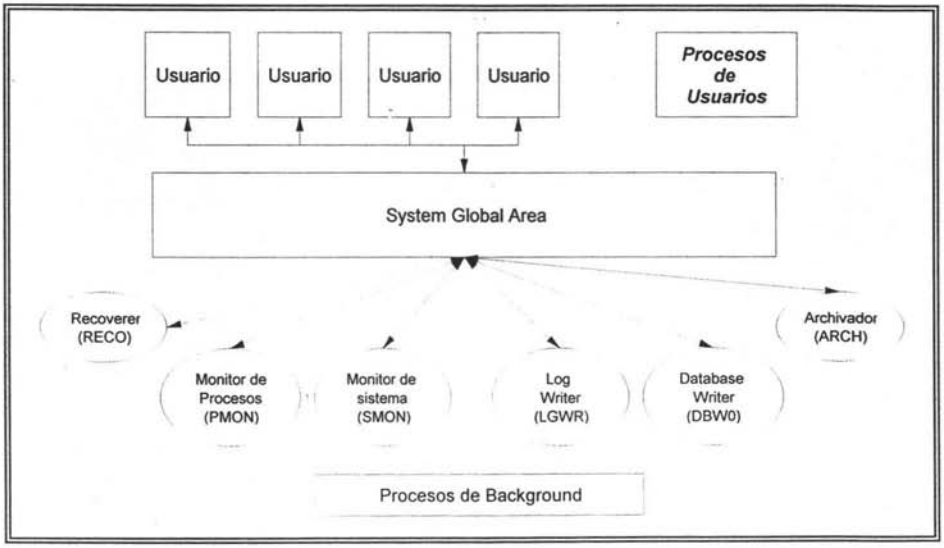


Figura 1.2 Una instancia Oracle.

Como se ejemplifica en la figura 1.2, los usuarios conectados simultáneamente a la base de datos, comparten la información almacenada en la SGA de la instancia a la que estén accesando.

1.2 Administrador de base de datos.

Un sistema manejador de base de datos Oracle no tiene límite para crecer en espacio físico y puede administrar a un número indefinido de usuarios (*ambos dependiendo del espacio en el disco duro y la memoria disponible*)[3]. En consecuencia, es necesario tener una persona o un grupo de ellas que se responsabilicen de la administración del sistema. El administrador de base de datos, o DBA es el encargado del buen funcionamiento.

En algunos sistemas de bases de datos hay varios DBA's con diferentes responsabilidades. Por ejemplo, algunos administradores pueden estar encargados de diferentes tareas, tales como la creación de usuarios, monitoreo de recursos del sistema y del mantenimiento de la seguridad.

En contraste con las responsabilidades de un DBA, un desarrollador de aplicaciones es responsable del diseño de objetos y desenvolvimiento de la base de datos, enfocado a mejorar el desempeño de las aplicaciones.

1.2.1 Responsabilidades del administrador de la base de datos.

Como se estableció en párrafos anteriores, el DBA tiene diferentes responsabilidades, eventualmente puede administrar también el equipo donde reside el DBMS. En la figura 1.3 se esquematizan de manera exclusiva las principales actividades de un DBA, concernientes a Oracle.

---

[3] **SMINE,** Harem. "*Oracle: Arquitectura, administración y optimización*" / traducción de Víctor Martín García. Madrid Díaz de Santos, 1992.

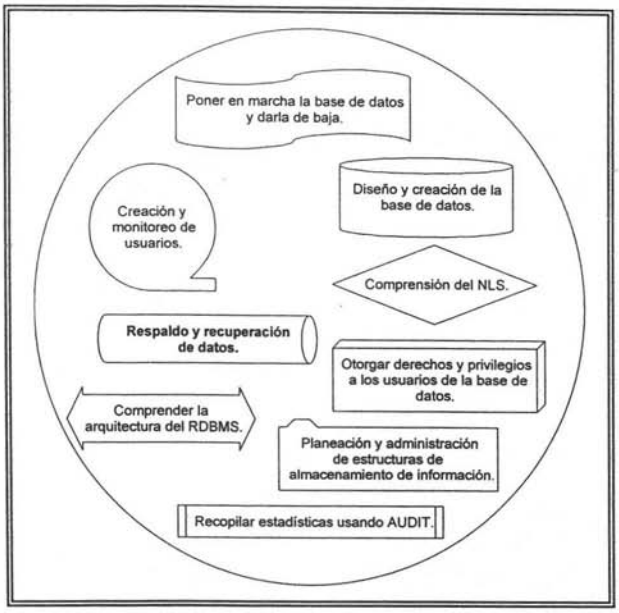


Figura 1.3 Responsabilidades de un Administrador de Base de Datos.

A. Diseño conceptual de la base de datos.

Ésta es quizá, la actividad más importante para un administrador de bases de datos. En ella se debe construir un esquema de la información que se usa en la empresa, independientemente de cualquier consideración física. A dicho proyecto se le denomina *esquema conceptual*. En su construcción, los diseñadores descubren la semántica (*significado*) de los datos de la compañía: encuentran entidades, atributos y relaciones.

El objetivo es comprender:

- La perspectiva que cada usuario tiene de los datos.
- La naturaleza de los datos, independientemente de su representación física.
- El uso de los datos a través de las áreas de aplicación.

El esquema conceptual se puede utilizar para que el diseñador transmita a la empresa lo que ha entendido sobre la información que maneja. Para ello, ambas partes deben estar familiarizadas con la notación utilizada en el esquema. La más popular es la notación del modelo entidad-relación.

El esquema conceptual[4] se construye utilizando la información que se encuentra en la especificación de los requisitos de usuario. El diseño conceptual es completamente independiente de los aspectos de implementación, como puede ser el DBMS que se vaya a usar, los programas de aplicación, los lenguajes de programación, el hardware disponible o cualquier otra consideración física. Durante todo el proceso de desarrollo del esquema conceptual éste se prueba y se valida con los requisitos de los usuarios. El esquema conceptual es una fuente de información para el diseño lógico de la base de datos.

B. Instalación y actualización del DBMS.

Como Administrador de la Base de Datos, se debe instalar el software del servidor Oracle, cualquier herramienta de tipo front–end y aplicaciones que accesen a la base de datos.

En algunas instalaciones de procesos distribuidos, la base de datos es controlada por una computadora central y las herramientas son ejecutadas en máquinas remotas, en este caso, el DBA debe instalar todos los controladores de red, para poder conectar las computadoras remotas a la base de datos.

[4] **CRONIN**, Daniel J. *"Mastering oracle :Featuring oracle's SQL standard"*. Indiana : Hayden, c1989.

C. Planeación de la base de datos.

El Administrador de la Base de Datos, debe planear:

- ❖ Las estructuras lógicas de almacenamiento de la base de datos.
- ❖ El diseño de la base de datos.

Es importante planear cómo las estructuras de almacenamiento lógico pueden afectar el desempeño de algunos procesos en la base de datos. Por ejemplo, se debe saber qué clase de información se almacenará y qué tipo de dispositivos de almacenamiento serán utilizados.

El diseño de la base de datos es uno de los puntos más importantes, contempla varias consideraciones que pudieran alterar el desempeño de la misma. Mencionando las más importantes:

- El desempeño de la computadora que se encuentra corriendo Oracle.
- La eficiencia de la base de datos durante las operaciones de acceso de datos.

D. Creación de usuarios (*esquemas*).

La creación de usuarios va íntimamente ligada a la seguridad dentro de la base de datos. Se puede controlar el acceso a una base de datos Oracle, creando, alterando, borrando y monitoreando usuarios[5].

[5] **KOCH,** George. "*Oracle: Manual de referencia*". Prólogo de Lawrence J. Ellison. Traducción. Osborne Mcgraw-Hill, c1999.

E. Control y monitoreo de usuarios a la base de datos.

Los usuarios en la base de datos son los encargados de almacenar, modificar y borrar la información. Por esto, se debe mantener una seguridad estrecha para la administración de los mismos. Dependiendo del tamaño de la base de datos y la cantidad de trabajo requerido para cada usuario, el administrador de la base de datos, debe ser el único con privilegios para crear, borrar y alterar usuarios, asimismo, el que otorgue los privilegios correctos para que el usuario desempeñe eficazmente su trabajo.

F. Mantenimiento del sistema de seguridad.

La seguridad debe ser considerada como un factor primordial para el funcionamiento óptimo del sistema Oracle y las diferentes aplicaciones que obtengan información. Algunos parámetros que ayudan a mantener un nivel de seguridad efectivo serían:

- El administrador de la base de datos debe tener privilegios del sistema operativo para crear y borrar archivos.
- Un usuario típico de la base de datos no deberá tener privilegios para crear o borrar archivos relacionados a la base de datos.
- Si el sistema operativo identifica los roles de la base de datos establecidos a cada usuario, el administrador de seguridad debe tener los privilegios necesarios para modificar las cuentas de los usuarios a nivel sistema operativo.

Se debe tener en cuenta que gran parte de las pérdidas de información en una base de datos, son ocasionadas por descuidos de usuarios.

G. Optimización del desempeño de la base de datos.

Oracle es un software sofisticado, que tiene la ventaja de poder ajustar su configuración; esta flexibilidad permite realizar pequeños ajustes, mejorando el desempeño de la base de datos. La afinación del sistema inicia en las fases de diseño y planeación de la base de datos y continúa a lo largo de la vida del sistema.

H. Respaldo y recuperación de la base de datos.

Este punto resulta de importancia subrayada para nuestra investigación. La planificación y prueba de procedimientos de respaldo y recuperación de datos de Oracle es un seguro con que se cuenta frente a posibles fallas del sistema operativo, del hardware, del propio software de Oracle y cualquier otro factor que implicara un daño severo dando por resultado la pérdida de archivos vitales de la base de datos. Cuanto mejor sea el plan, habrá mayor número de posibilidades disponibles para la recuperación. Como en simulacros de terremotos o incendios, un plan de respaldo y recuperación de datos necesitará disciplina y práctica.

Basándonos en esta premisa, podemos afirmar que la información de nuestra base de datos siempre tendrá disponibilidad y consistencia.

1.3 Arquitectura del manejador de bases de datos Oracle 9i.

Un administrador de base de datos debe entender la arquitectura del sistema que controla las citadas de bases de datos; en el caso que nos ocupa, debe comprender la estructura de Oracle.

Los componentes principales dentro de la arquitectura son:

- *Estructuras de memoria.*
- *Procesos.*
- *Archivos.*

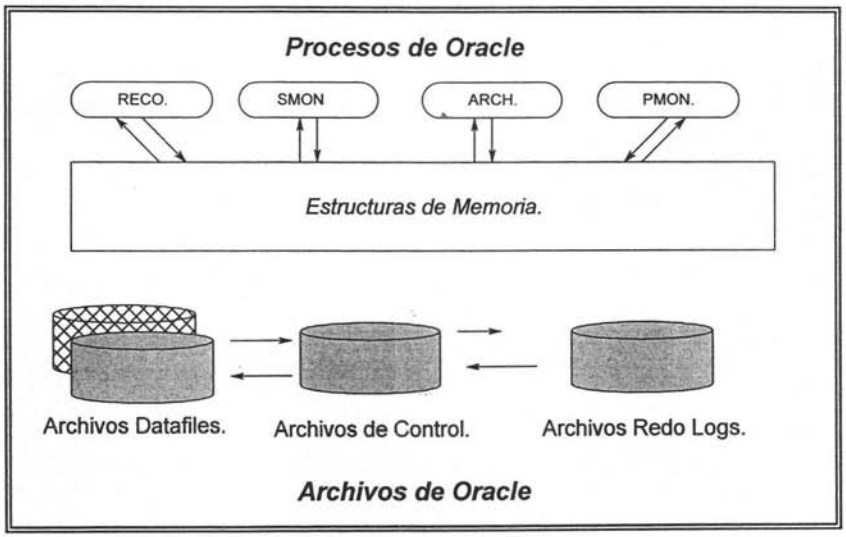


Figura 1.4 Componentes de la arquitectura de Oracle.

1.3.1 Estructuras de Memoria.

Un sistema Oracle utiliza estructuras de memoria y procesos para el acceso a la base de datos. *Todas las estructuras de memoria existen dentro de la memoria principal de la computadora en la que resida la base de datos denominada SGA*[6].

[6] **COREY,** Michael J. *"Oracle"* / Michael Abbey, Daniel J. Dechichio ; McGraw-Hill, c2003.

A. SGA.

La estructura de memoria principal dentro del DBMS Oracle es el Área Global del Sistema o SGA (*System Global Area* por sus siglas en Inglés) .

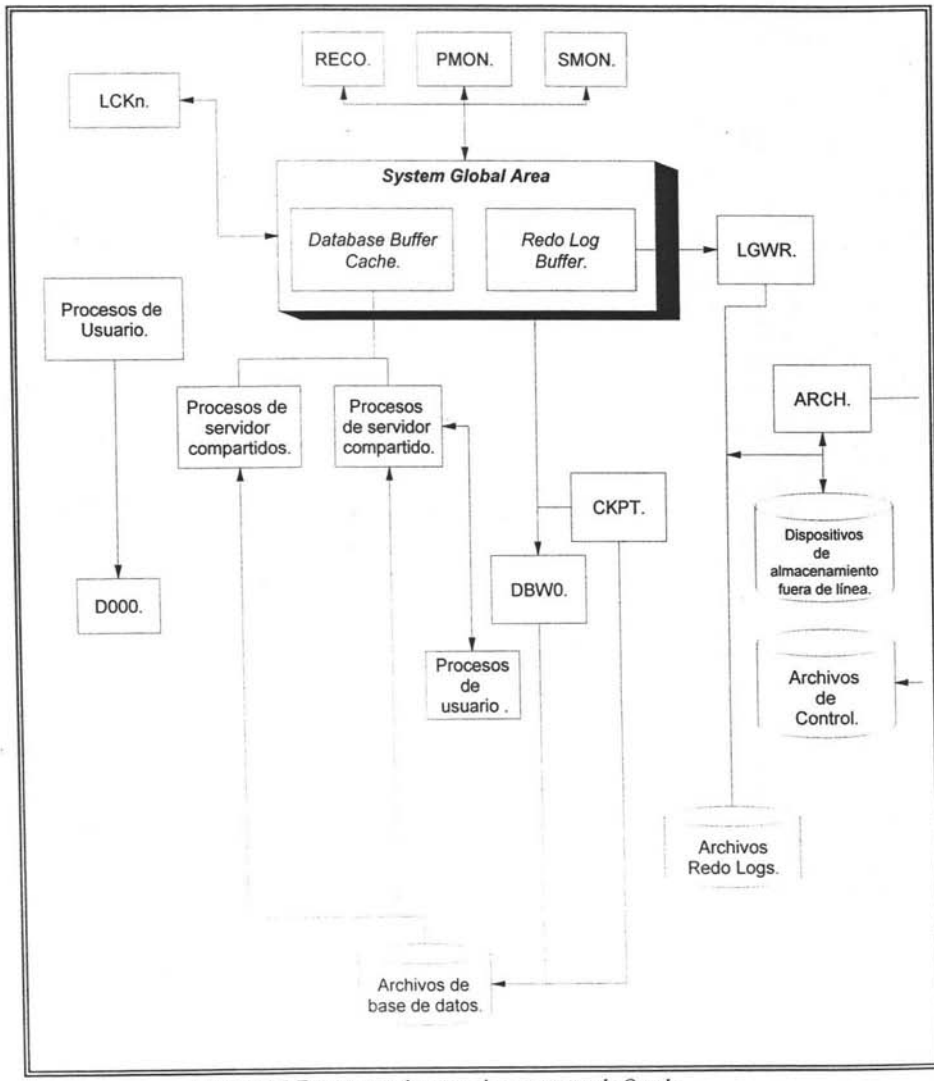


Figura 1.5 Estructuras de memoria y procesos de Oracle.

Está definida como una región de memoria compartida que contiene, además de los datos e información de una instancia Oracle, los parámetros de configuración y los datos que se estén extrayendo o manipulando en ese momento. Cada instancia Oracle tiene su propia SGA, administrando óptimamente los recursos del sistema.

La SGA está compuesta por tres principales elementos:

- *Database Buffer Cache.*
- *Redo Log Buffer.*
- *Shared Pool.*

B. Shared Pool.

La Shared Pool es una sección de la SGA que contiene estructuras de memoria compartida, tales como áreas específicas para SQL. Este tipo de segmentos de memoria, son requeridos exclusivamente para expresiones de SQL solicitadas a la base de datos. Un área compartida de SQL contiene por ejemplo, el plan de ejecución de la sentencia.

C. Database Buffer Cache.

Los buffers de una base de datos, se encargan de almacenar la información extraída más recientemente. Al juego de buffers de una base de datos se le conoce como Database Buffer Cache.

El Database Buffer Cache mantiene los bloques de datos más usados y más recientes en memoria. Esto es para disminuir la entrada y salida de datos de un disco duro, lo que propicia el mejoramiento del desempeño.

D. Redo Log Buffer.

El Redo Log Buffer de una SGA, genera un archivo de bitácora, que contiene todos los cambios hechos a la base de datos. Cualquier cambio producido en la base de datos, es reflejado en tiempo real en los archivos log, los cuales son indispensables en caso de ser necesaria una recuperación de información.

1.3.2 Procesos.

El manejador de la base de datos Oracle funciona mediante dos conjuntos de procesos que se encargan de diferentes tareas; dichos conjuntos se enlistan a continuación:

- Procesos de Background.

Oracle genera un conjunto de procesos de background (*se les asigna esta denominación porque no corren en un entorno en el que el usuario los pueda identificar*[7]) para cada instancia. Ellos desarrollan las funciones que de otra forma serían manejadas por múltiples programas por cada proceso generado por un usuario.

[7] **LOCKMAN,** David. "*Developing Personal Oracle8 applications*" Indianapolis, Indiana : Sams, c1997.

Los procesos de background se generan básicamente para cuatro tipos de tareas:

1. Procesos de servicio al usuario: Proveen la conexión entre los procesos de las aplicaciones independientes y las áreas de memoria de la base de datos Oracle.

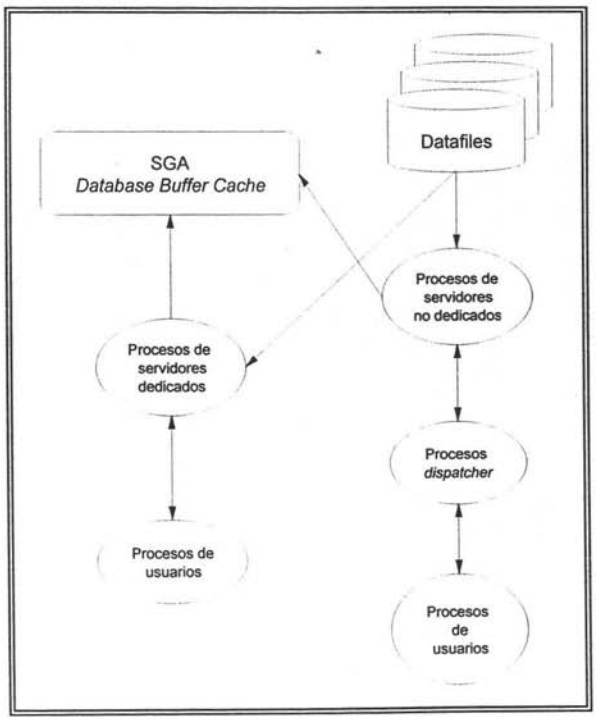


Figura 1.6 Procesos de servicios a usuarios.

Este tipo de procesos se encargan de conectar a usuarios con las áreas de memoria (*Figura 1.6*), diferenciando entre los procesos de servidores dedicados y servidores no dedicados.

2. Procesos de escritura de datos: Toman los datos almacenados en los buffers de la base de datos y los escriben en los datafiles.

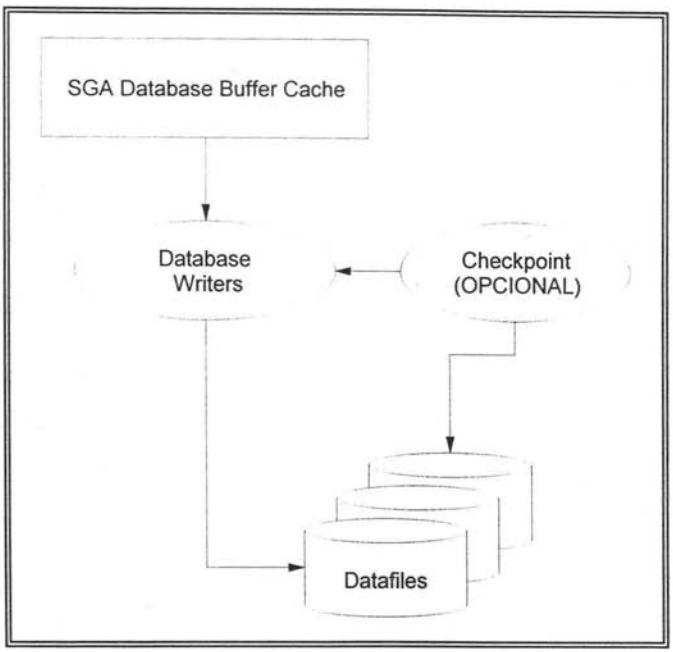


Figura 1.7 Proceso DBWR que escribe información en los archivos datafiles.

El funcionamiento en esta etapa está dividido en dos procesos: mientras que los *DBWR (Data Base Writers)* escriben la información que se encuentra temporalmente en la SGA en los datafiles, el proceso CKPT se encarga de especificar al proceso anterior en dónde debe escribir la información.

La figura 1.7 sintetiza el proceso de CKPT. En primer término, debe checar en cuál datafile se debe escribir la información para posteriormente especificarlo al proceso DBWR.

3. Procesos de escritura de archivos log: Toman los datos almacenados en el *log buffer* transfiriéndolos a los archivos redo log.

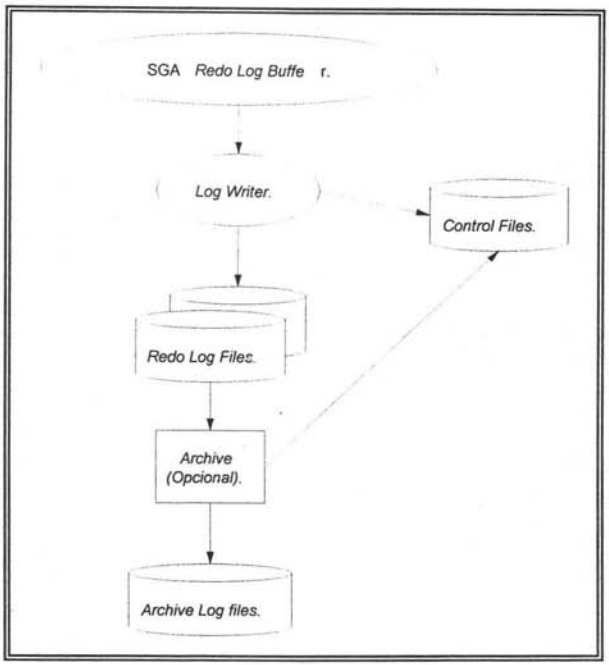


Figura 1.8 Procesos de escritura de archivos log.

Estos procesos son responsables de transferir las transacciones que han ocurrido desde los redo log buffer en la SGA a los archivos redo log .

En la figura 1.8 podemos apreciar la manera en que el proceso LGRW (*Log Writer* por sus siglas en inglés) obtiene los datos que estén almacenados en ese momento en la SGA, manteniendo una copia de todas las transacciones en los archivos redo log.

4. Procesos de monitoreo: Mantienen vigilado el funcionamiento correcto de los demás procesos.

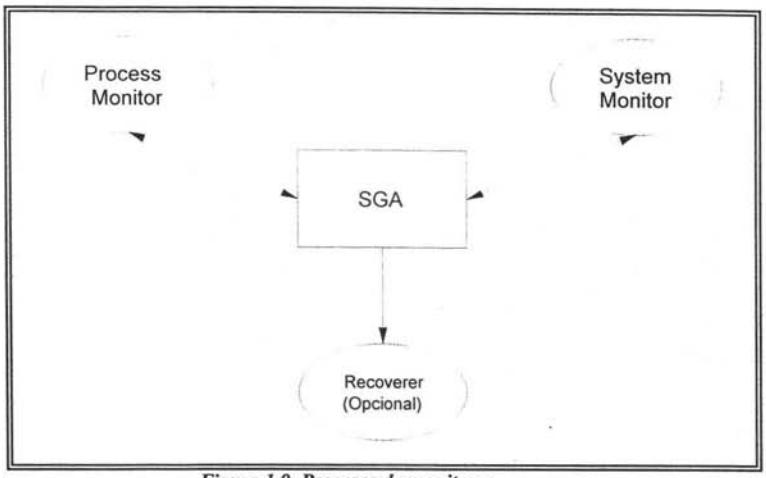


Figura 1.9 Procesos de monitoreo.

Los procesos de monitoreo tienen como función primordial el mantener el funcionamiento del sistema Oracle en óptimas condiciones, desde arrancar las bases de datos, hasta recuperar la información que pudiera llegar a perderse mediante una transacción remota.

Indefectiblemente, esta clase de procesos hacen referencia a la SGA (Figura 1.9). Su finalidad es mantener bajo control el funcionamiento de las diversas instancias de Oracle[8] .

[8] **WELLESLEY**, W.h. Inmon. "*Oracle design review guidelines*". Massachusetts: Qed information sciences, c1998.

Los procesos de monitoreo son los siguientes:

✓ *DBWn (Database Writer) [La letra n indica el identificador del proceso.]*

El proceso database writer (*DBW*) es responsable de transferir datos que han sido modificados dentro del área de memoria del buffer cache en los datafiles. La idea es garantizar la existencia de buffers libres en memoria, esperando a que un proceso de un usuario quiera escribir y modificar registros en la base de datos.

✓ *LGWR (Log Writer).*

Cada vez que Oracle inicia un commit[9], las transacciones son almacenadas en dos diferentes lugares a fin de proteger la información en caso de que los archivos físicos donde se guardan (*datafiles*) sufrieran algún daño.

El concepto general de este proceso es simple. Cada transacción hecha a la base de datos es conservada en el redo log buffer. Éste opera como estructura dinámica llamada cola: el primero que entra, el primero que sale. Así, la diversidad de transacciones es recopilada en estos archivos. Este proceso verifica cada tres segundos si existe un nuevo cambio para actualizarlo en los archivos redo log.

[9] **NOTA DEL AUTOR.** Se denomina *commit* al comando que permite al sistema aceptar todos los cambios realizados a la base de datos y almacenarlos.

✓ *CKPT (Checkpoint).*

En determinadas ocasiones, todos los buffers de la base de datos que mantienen todos los cambios en la SGA, son escritos por el DBWn en los datafiles. A este proceso se le conoce como checkpoint, el cual es responsable de señalar a DBWn los datafiles que tienen que ser actualizados.

✓ *SMON (System Monitor).*

Quizás es el proceso más importante dentro de los procesos de background. Se encarga de arrancar a la instancia o de recuperar información en caso de una falla. También se encarga de liberar espacio en los segmentos temporales de almacenamiento, que ya no están en uso y recuperar las transacciones detenidas durante una falla.

✓ *PMON (Process Monitor).*

El Process Monitor (*PMON*), es el encargado de la recuperación de los procesos cuando alguno falla. Es responsable de liberar los recursos de los diferentes procesos que estaban corriendo y limpiar el caché.

✓ *RECO (Recoverer).*

El recoverer es usado para solucionar transacciones que se encuentran pendientes a causa de la red o fallas en un sistema de bases de datos distribuidas. Cada determinado periodo de tiempo, el recoverer trata nuevamente de conectarse a la base de datos y automáticamente terminar el commit o el rollback, según sea el caso.

✓ *Dnnn (Dispatcher).*

Por lo menos un Dispatcher es generado para protocolo de comunicación en uso. Cada dispatcher es responsable de dirigir las distintas peticiones de datos de los diversos usuarios en dirección de los procesos disponibles a los procesos compartidos y regresarlos al usuario adecuado.

✓ *SNPn (Job Queue).*

Dentro de una configuración de bases de datos distribuidas, este tipo de proceso se encarga de actualizar las características de las diferentes tablas en la base de datos.

➢ Procesos de Usuarios.

Un proceso de usuario es creado y almacenado para ejecutar determinadas acciones en la base de datos[10]. Como puede ser una sentencia SQL o algun mecanismo interno de Oracle (Un trigger o un procedimiento almacenado)

[10] **GREENWALD**, Rick & **HOSKIN**, Robert: "*Mastering oracle power objetcs* ", Sebastopol California: O'Reilly, c1999.

1.3.3 Archivos.

Para la instalación de un software en una computadora, se copian una serie de archivos y documentos en un directorio o en varios subdirectorios bajo un directorio principal. El DBMS Oracle está compuesto de cientos, quizás miles de archivos dispersos en varios directorios, dependiendo del uso de cada uno.

- Oracle Optimal Flexible Arquitecture.

La ubicación de los archivos que componen el DBMS Oracle es un factor muy importante. El conocer dónde se localizan todos archivos permite una administración precisa y clara de las diferentes instancias.

Oracle maneja un concepto llamado Arquitectura Óptima y Flexible (*OFA por sus siglas en Inglés*), que permite que un software tan robusto como el Oracle pueda ser instalado fácilmente. Esta arquitectura es el estándar para los scripts de instalación en la mayoría de los sistemas operativos.

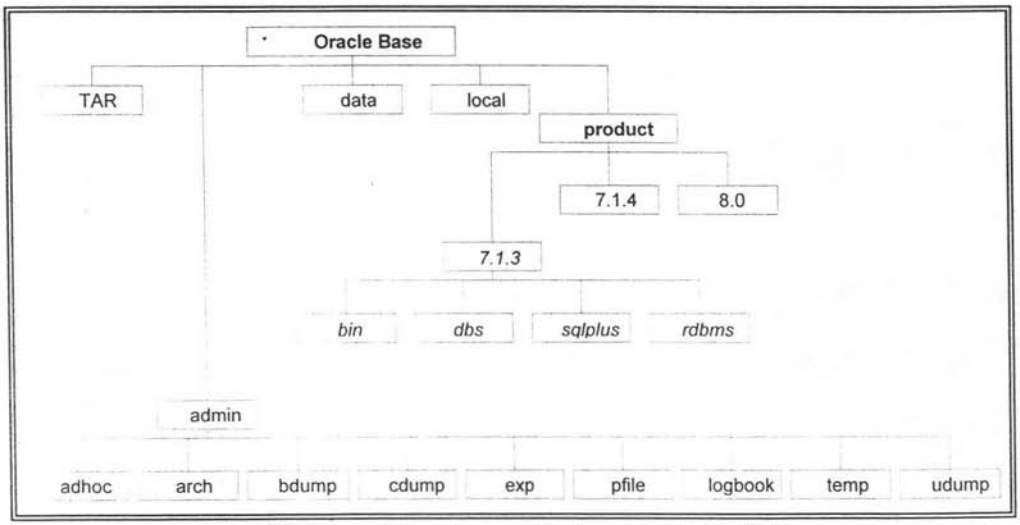


Figura 1.10 La arquitectura óptima y flexible para un sistema UNIX.

El punto base en la OFA es conocido como ORACLE_BASE, ahí se encuentran tres subdirectorios básicos. El subdirectorio "admin" almacena los archivos de bitácora y de otra información para el DBA. "Data" está diseñado para el almacenamiento de los datafiles y el subdirectorio product almacena las aplicaciones Oracle y los archivos de configuración de las diferentes instancias que se encuentran corriendo.

Como se muestra en la figura 1.10, se pueden tener diferentes versiones del manejador de bases de datos simultáneamente en un sistema. Esto resulta de especial utilidad cuando se está probando la nueva versión del DBMS, en tanto la base de producción continúa funcionando.

- Datafiles.

Los datafiles desarrollan la función más importante dentro del sistema Oracle, en tratándose del almacenamiento de datos en un formato recuperable. En ellos se deposita toda la información de la base de datos. Estos archivos no pueden ser leídos por utilerías del sistema operativo directamente, como podrían ser un *more* en UNIX o un *notepad* en Windows. La única forma de accesar a estos contenidos es vía SQL. De esta manera se puede controlar a dónde van a ser almacenados los datos, definiendo el datafile o el tablespace.

Un datafile está asociado exclusivamente a un tablespace, mientras que éste puede tener uno o más datafiles.

A manera de muestreo, la figura 1.11, establece la descripción obtenida de una vista física, la cual difiere sustancialmente de la vista lógica. Ambas contienen los mismos elementos, pero definidos en la primera por archivos y en la segunda por objetos.

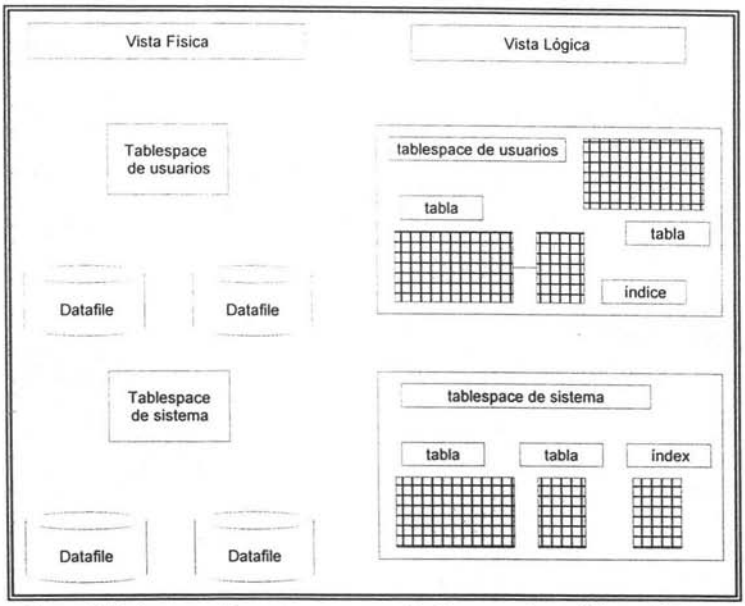


Figura 1.11 Comparación entre estructuras de almacenamiento físicas y lógicas.

- Tablespaces.

Se denominan tablespaces a las unidades lógicas en que se encuentra dividida una base de datos. Son usadas para agrupar estructuras físicas. Por ejemplo, un Tablespace agrupa objetos de una aplicación para simplificar la administración.

- Archivos Redo Log.

Cada base de datos Oracle tiene un juego de dos o más archivos redo logs. La función primordial de estos archivos es almacenar todos los cambios hechos a la información.

Si existiera alguna falla, se puede prevenir que ciertos datos fueran escritos de forma permanente en los datafiles; los cambios podrían obtenerse de los archivos redo logs y la información no representaría ninguna pérdida .

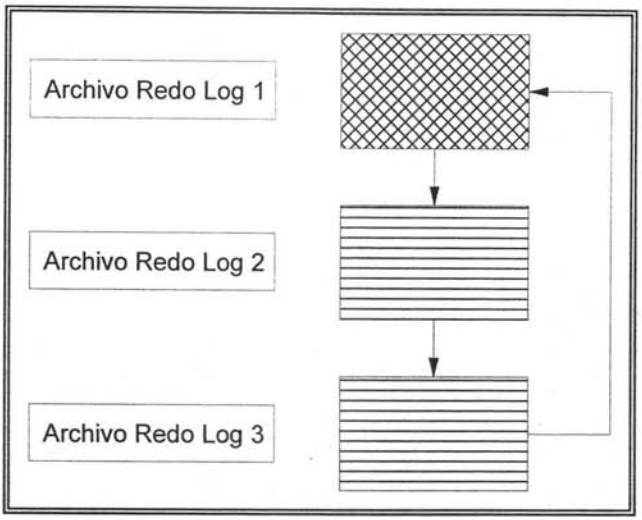


Figura 1.12 Funcionamiento cíclico de los archivos Redo Log.

Los archivos Redo Log funcionan de manera cíclica (figura 1.12). Una vez que el archivo en el que se está escribiendo la información ha alcanzado el tamaño predefinido, el sistema escribe las siguientes transacciones en el archivo subsecuente. Cuando concluye la escritura en el último archivo de la secuencia, empezará a generar nuevamente el primer archivo.

- Segmentos de Rollback.

El DBMS Oracle contiene uno o más segmentos de rollback, los cuales son porciones de la base de datos que registran los cambios efectuados en ella por las diversas aplicaciones y su función consiste en deshacer los movimientos realizados en la base, por si se requiriera volver a su primigenio estado.

Los segmentos de rollback almacenan los valores anteriores de la información que sufre constantes modificaciones y son liberados cuando se aplica commit.

- Archivos de control.

Cada base de datos Oracle tiene un archivo de control. Éste contiene la siguiente información:

- Nombre de la base de datos.
- Nombres y ubicaciones de los datafiles y archivos redo logs.

Como un archivo redo log, Oracle permite que el archivo de control sea multiplexado para protección del mismo. Cada vez que una instancia de Oracle es inicializada, su archivo de control es utilizado para identificar la base de datos y los archivos redo logs que deben ser abiertos para la operación de la propia base.

Oracle administra internamente estos archivos. Ningún usuario (*incluyendo el DBA*) podrá modificarlos manualmente o controlar su tamaño.

- Archivos de parámetros.

A pesar de que Oracle accesa a los archivos de control para determinar la configuración física de la base de datos, existen ciertos archivos de parámetros relacionados principalmente a mejorar el desempeño de la base de datos.

Para arrancar una instancia, Oracle debe leer un archivo de parámetros (*un archivo en formato texto, que contiene una lista de datos con los que Oracle pondrá en marcha la instancia*).

Existen dos archivos de parámetros principales:

- Init.ora : En este se encuentra el nombre de la instancia.
- Config.ora : Aquí se almacenan las ubicaciones de los archivos de control. También contiene el nombre de la base de datos y los parámetros del tamaño de los database block.

Mientras un archivo config.ora mantiene la configuración para varias instancias conservando parámetros en común, el archivo init[nombre de la instancia].ora administra parámetros independientes para cada una de éstas.

➢ Archivos de bitácoras y archivos trace.

Con el fin de ayudar a la administración de la base de datos, Oracle tiene algunas utilerías que registran los eventos más importantes dentro de las bitácoras del sistema. Este archivo puede estar en diferentes ubicaciones, sin embargo, el nombre no varía alert_[nombre de la instancia].log[11].

Entre la información que se encuentra almacenada en este archivo destacan:

- Acciones del DBA como son detener y arrancar la instancia o ciertos comandos que pueden modificar el estado de la base de datos (*create, modify o drop*).

[11] **SMINE,** Hatem. *"Oracle: arquitectura, administración y optimización"*. Traducción Víctor Martín García. Madrid: Díaz de Santos, 2000.

➢ Errores internos del DBMS.

Los archivos "trace" son generados por los procesos de background cuando existe un problema superior, o cuando se detecta algún conflicto en uno de ellos o en su caso, por la falta de los aludidos procesos. El contenido de estos archivos varía, dependiendo de la cantidad de información que el proceso genere. En este caso, la fecha, el proceso que detecta el problema y el número de error Oracle.

1.4 ¿Por qué es importante planear la recuperación de información?.

En grandes sistemas, el funcionamiento de las bases de datos de varios terabytes en un entorno cliente/servidor complejo, es una tarea desalentadora. Los componentes software y hardware deben cooperar de forma precisa para proporcionar la información al usuario final. Sin embargo, llegan a suceder errores.

De acuerdo con el documento publicado por la IEEE en relación a la seguridad en informática con clave FF8907, las fallas se clasifican en distintos tipos, que pueden ser agrupados en las siguientes categorías:

- Físicas.
- De diseño.
- De operación.
- De entorno.

Las fallas físicas son causadas generalmente por anomalías en el hardware, tales como problemas en el disco duro o una descompostura del CPU. Los fallos en el diseño son ocasionados por conflictos en el software. Cualquier error, tanto en el sistema operativo como en el software de la base de datos de la aplicación, contribuye a una falla en el diseño.

Por otra parte, los fallos de operación son causados por la intervención humana. Algunos ejemplos de problemas de operación son los errores atribuidos a la inexperiencia de los DBA's, equivocaciones de usuario, configuraciones inadecuadas del sistema o procedimientos inapropiados de backup. Finalmente, un fallo de entorno es un yerro debido a cuestiones del entorno exterior, como temblores, sobretensiones de energía eléctrica o condiciones anormales de temperatura.

Un DBA puede ejercer la mayor parte del control sobre los fallos de operación. Aunque un DBA no sea capaz de predecir anomalías físicas, de diseño o de entorno, debe estar preparado para los problemas que él mismo pueda provocar. El DBA debe planificar un procedimiento de backup sólido, probándolo periódicamente para actualizarlo según vaya creciendo la base de datos.

Además los DBA también se pueden preparar para hacer frente a los fallos, practicando métodos de recuperación simulados en sistemas de prueba[12].

Errar es humano, pero muchos de nuestros yerros se pueden minimizar si los prevenimos, preparándonos adecuadamente.

[12] ROLLAND, F. D. "*Relational database management with oracle*" Wokingham, England : Addison-Wesley, 2004.

➢ *Características de las bases de datos.*

La información almacenada dentro de las bases de datos tiene tres peculiaridades que deben considerarse en relación al uso que se vaya a destinar.

- Disponibilidad.

  Existen casos en el que el acceso a la información es constante. Aplicaciones críticas (*v.gr.* como las bancarias), están basadas en la disponibilidad de la base de datos durante veinticuatro horas por siete días a la semana. Por ejemplo, los cajeros automáticos de cualquier institución bancaria accesan de forma remota a diferentes bases de datos a efecto de identificar a un cliente y obtener su saldo, sin importar el día o la hora.

- Nivel de seguridad.

  Con el uso más frecuente de Internet, el acceso remoto a diferentes bases de datos ha crecido considerablemente. El accesar en línea a la biblioteca central de la UNAM o realizar alguna compra en un portal de comercio electrónico es de lo más normal. Sin embargo, existe el riesgo de que la información pudiera ser alterada.

  Cuando se envía información a través de una forma en un web site, ésta debería estar encriptada para su seguridad; sin embargo existen formas para desencriptar la información. A pesar de que existen diferentes soluciones para implementar de comercio electrónico, ninguna ofrece una seguridad del cien por ciento.

- Tamaño Físico.

  Algunas bases de datos crecen de una forma exponencial. El número de llamadas locales que registra una central telefónica por minuto se ve reflejado como un registro por cada intento. Obviamente esto repercute esencialmente en el tamaño físico de la base de datos.

  El mantenimiento del almacenamiento de la base de datos se vuelve una tarea compleja. Dispositivos como discos magnetoópticos, arreglos de discos duros redundantes conectados por fibra óptica o cintas de diversas capacidades, son útiles para el desempeño de la base, pero no sustentan una garantía.

Capítulo II Herramientas y estrategias de respaldo de bases de datos con el DBMS Oracle.

Con motivo de la presentación de la nueva versión del DBMS Oracle 10g, tuvo verificativo una conferencia, en donde el ponente, Larry Ellison CEO de Oracle Software, hizo la siguiente aseveración:

"Existen dos tipos de DBA en el mundo: los que han tenido una caída de su base de datos de producción y los DBA que la tendrán".

Como administrador de la base de datos, se debe garantizar que su información estará protegida y es recuperable. Oracle ofrece varias herramientas con las que se puede desarrollar una estrategia de respaldo ofreciendo la premisa de que la aludida información siempre será confiable y consistente.

Los requerimientos empresariales de mantener disponible la información las veinticuatro horas del día, siete días a la semana, ha favorecido el desarrollo de diversas técnicas para restaurar o recuperar los datos, sin que para ello sea óbice el uso de la base de datos, v.gr. facturación, comercio electrónico, CRM (Customer Relationship Management por sus siglas en Inglés), o ERP (Enterprise Resource Planning por sus siglas en Inglés). Entre dichas técnicas Oracle sugiere el mantenimiento de una base de datos en *stand by* y actualizarla como si fuera un espejo. Sin embargo, tal alternativa suele ser bastante costosa, ya que deben copiarse requerimientos específicos a donde vaya a residir esta base, como serían el almacenamiento, el software, el procesamiento, etcétera.

Por lo tanto, en este capítulo se proponen diversas formas de respaldar la información de la base de datos, de conformidad a una pluralidad de factores, por ejemplo, el tiempo, las necesidades de la organización, última transacción registrada, entre otros.

2.1 ¿Qué es un respaldo de una base de datos Oracle?

En términos generales, un respaldo de una base de datos es una copia exacta de la información, cuyo objetivo es su reconstrucción fidedigna, en el caso probable de que sobreviniera algún daño. Esta copia incluye archivos esenciales que constituyen la base de datos Oracle, tales como archivos de control, archivos logs y datafiles, las cuales fueron definidas en el Capítulo 1 punto 3 *"Arquitectura del manejador de bases de datos Oracle 8I"*. Si existiera un fallo en un dispositivo de almacenamiento ligado a la base de datos, el respaldo es la clave para restaurar la información.

La pérdida de ingresos que tienen distintas organizaciones empresariales, tales como bancos, compañías de servicio de mensajería especializada y aerolíneas, por mencionar sólo algunas, resulta cuantiosa cuando su base de producción no está disponible por cinco o diez minutos. Otro ejemplo podría ser si se perdiera un datafile como consecuencia de la falla en un disco duro sin que su recuperación fuera factible debido a la carencia de un respaldo. El DBA debe restaurar y recuperar la base de datos oportuna y eficazmente, para reanudar las operaciones, por ende, es necesario tener bien definidas las estrategias de respaldo y recuperación de la base de datos.

El DBA debe diseñar la estrategia de respaldo de acuerdo a las necesidades de la compañía; por ejemplo, si es aceptable la pérdida de información por algún fallo en un dispositivo de almacenamiento, quizás no sea necesaria la continua realización de respaldos. No obstante, cuando deba tenerse una base de datos de producción disponible las veinticuatro horas, siete días a la semana, el DBA debe respaldar la información constantemente. La frecuencia del respaldo, una vez seleccionado su tipo entre las diferentes opciones de respaldos, son en gran medida determinadas por las necesidades de las empresas.

2.2 Desarrollando una estrategia de respaldo de base de datos.

Antes de crear una base de datos Oracle, el DBA debe decidir cómo protegerla contra potenciales fallas de los diferentes medios de almacenamiento; si no se desarrolla previa a su creación, una estrategia de respaldo, no podrá desplegarse su recuperación parcial o total si existiera alguna falla en un disco duro que incidiera en el daño de un datafile, un archivo de control u otro elemento esencial del DBMS.

Sin importar la estrategia de respaldo implementada, es necesario tratar de seguir los siguientes principios, pues ayudarán a determinar cuándo es fundamental realizar un respaldo, o qué partes de la base de datos se deben respaldar. Tales principios son:

A. Tratar de mantener un "*grupo de redundancia*".

Bajo esta denominación se conoce el conjunto de archivos que son necesarios para la restauración de una base de datos Oracle (Datafiles, archivos de control y archivos redo log), los cuales se determinan con el adjetivo de redundantes debido a la reiteración de elementos específicos de diversas formas. El grupo de redundancia contiene:

- ✓ *El último respaldo de todos los archivos de la base de datos.*
- ✓ *Todos los archivos de redo log generados con posterioridad a que el último respaldo fue tomado.*
- ✓ *Una copia tanto de los archivos redo log, como de los archivos de control.*
- ✓ *Archivo tnsnames.ora y listener.ora.*

Es vital que este grupo de redundancia no esté almacenado en los discos duros en los que la base de datos resida.

B. Multiplexar archivos de control y los archivos redo log.

Los archivos de control y los archivos redo log son cruciales para las operaciones de respaldo y recuperación de base de datos. Se deben mantener por lo menos dos copias de estos conjuntos de archivos en diferentes ubicaciones multiplexadas por el DBMS o modificadas en espejo por el sistema operativo.

C. Desarrollar respaldos frecuentemente.

El desarrollo de respaldos es esencial para la recuperación de cualquier esquema. Se debe basar la frecuencia de los respaldos acorde a los cambios realizados a la base de datos, por ejemplo:

- ✓ Adición y borrado de tablas o registros.
- ✓ Actualizaciones de la información de la base de datos.
- ✓ Modificación de la configuración de la base de datos.

Si los usuarios generan un monto significativo de modificaciones en los datos, los respaldos tendrán que efectuarse con una frecuencia mayor a la requerida para base de datos únicamente de lectura, cuya actualización es periódica.

D. Desarrollo de respaldos cuando se realicen cambios estructurales.

El DBA es conocido también como un usuario que puede realizar cambios estructurales a la base de datos[13].

Si se realizaran ciertos cambios estructurales (por ejemplo: Crear o borrar un tablespace, renombrar un datafile de un tablespace ya existente o renombrar un grupo de archivos redo log), es necesario realizar un respaldo antes y después de los cambios efectuados.

E. Probar la estrategia de respaldo.

Se deben producir ensayos de las estrategias de respaldo de la base de datos antes y después de la puesta en marcha de un sistema de producción. De esta manera, se conocerán los tiempos de respuesta y se minimizarán los problemas que pudieran presentarse en una situación real.

2.3 Manejando estructuras de datos.

Quizá la estrategia de respaldo de información más sencilla es la que se detalla en este punto. El DBA debe prevenir de diferentes maneras una futura pérdida de información además de desarrollar una defensa acorde a la falla.

[13] **KOLSTE**, Bruce & **PETERSEN** David. "*Oracle power objects handbook*" Berkeley, California ; México City: Osborne Mc Graw-Hill.

El manejo inteligente de las estructuras de datos viene a significarse como un aspecto importante dentro de las estrategias de respaldo. ¿Cuáles serían las acciones a seguir por el DBA para prevenir la falla en una base de datos si el archivo de control se corrompiera? ¿Cómo puede solucionarse la pérdida de un conjunto de archivos redo log, si el disco duro en el que residían presenta una falla?. Además de los datafiles, los archivos más importantes para desarrollar una estrategia de respaldo son:

- ✓ Archivos de control.
- ✓ Archivos redo logs.
- ✓ Archivos redo log mientras la base de datos se encuentra en modo ARCHIVELOG.

Si estas estructuras no estuvieran disponibles para trabajar con el DBMS, el DBA no tendría la posibilidad de recuperar los datos que pudieran perderse. Se pueden generar estrategias de respaldo que ayudaran a prevenir situaciones en donde estén involucrados los archivos de control y los archivos redo log; por ejemplo:

- ✓ Mantener los archivos datafiles, los archivos redo log y archivos de control en diferentes ubicaciones (Discos duros, arreglos de discos, cintas), actualizándolos en tiempo real (Multiplexión). Esto con la finalidad de recuperar la base de datos ante cualquier eventualidad mientras el sistema se encuentra completamente disponible.
- ✓ Respaldar los archivos redo log en modo archive de una manera frecuente y en diversos dispositivos.

2.3.1 Manejando los archivos de control.

El archivo de control es un pequeño archivo binario que contiene un registro del esquema de la base de datos. Es una de las estructuras más importantes, ya que es necesario para poner en marcha la base de datos y mantener una operación exitosa. El DBMS actualiza el archivo de control continuamente durante el uso de la base de datos; éste debe estar disponible para poder actualizarse en cualquier momento en que el DBMS lo requiera. Si por alguna razón el archivo de control no estuviera accesible, la base de datos no podría montarse y su recuperación sería difícil.

Un archivo de control contiene información requerida para accesar a la base de datos en forma de una instancia. Esta información no puede modificarla cualquier usuario (incluyendo al DBA), sólo el DBMS tendrá acceso.

El archivo de control tiene varias propiedades que lo hacen crucial para procesos de respaldo y recuperación de la base de datos:

- ✓ Identifica el nombre de la base de datos.
- ✓ Guarda el registro de la ubicación de los datafiles y de los archivos redo logs.
- ✓ Almacena información del proceso de background checkpoint para poder sincronizar todos los archivos de la base de datos.
- ✓ Almacena información del Recovery Manager (RMAN). La información con la que se obtiene el catálogo de objetos del RMAN es extraída desde el archivo de control.

➢ Desplegando la información del archivo de control.

Cuando se maneja un archivo de esta trascendencia, es necesario disponer de toda su información relativa. Existen dos vistas con esta información.

Vista.	***Descripción.***
V$CONTROLFILE	Lista los nombres de los archivos de control.
V$DATABASE	Indica información del archivo de control; por ejemplo, cuándo fue creado, la fecha de su última actualización y si se trata de un respaldo o del archivo original.

Por ejemplo, con la siguiente consulta pueden obtenerse las ubicaciones de los archivos de control:

```
SQL> select name from V$controlfile;

NAME.
--------------------------------------------------------------------------------
/datafile1/oradata/bdweb/control01.ctl
/datafile1/oradata/bdweb/control02.ctl
/datafile1/oradata/bdweb/control03.ctl
```

Para desplegar el tipo de archivo de control, se corre la siguiente consulta:

```
SQL> SELECT controlfile_type FROM v$database;

CONTROL
-------
CURRENT
```

Para obtener la información de los archivos de control, datafiles y archivos redo log, debe aplicarse la siguiente consulta:

```
SQL> SELECT member FROM v$logfile
 2  UNION ALL
SELECT name FROM v$datafile
UNION ALL
SELECT name FROM v$controlfile; 3   4   5
MEMBER
--------------------------------------------------------------------------------
/datafile1/oradata/bdweb/redo01.log
/datafile1/oradata/bdweb/redo02.log
/datafile1/oradata/bdweb/redo03.log
/datafile1/oradata/bdweb/system01.dbf
/datafile1/oradata/bdweb/tools01.dbf
/datafile1/oradata/bdweb/rbs01.dbf
/datafile1/oradata/bdweb/temp01.dbf
/datafile1/oradata/bdweb/users01.dbf
/datafile1/oradata/bdweb/indx01.dbf
/datafile1/oradata/bdweb/drsys01.dbf
/oracle/app/oracle/product/8.1.6/dbs/prueba44
/oracle/app/oracle/product/8.1.6/dbs/prueba
/datafile1/oradata/bdweb/prueba2.dbf
/datafile1/oradata/bdweb/system02.dbf
/datafile1/oradata/bdweb/control01.ctl
/datafile1/oradata/bdweb/control02.ctl
/datafile1/oradata/bdweb/control03.ctl
```

- Respaldando el archivo de control después de efectuar cambios estructurales en la base de datos.

Cada vez que el DBA agrega, renombra o borra un datafile o un archivo redo log, el DBMS actualiza el archivo de control, reflejando los cambios que se hicieran a la base de datos.

Cuando se actualiza la estructura física de la base de datos, es necesario respaldar el archivo de control. De no efectuarlo, el respaldo más reciente no reflejará los cambios estructurales efectuados antes de una probable falla del sistema o de la corrupción del archivo de control, trayendo como consecuencia que el sistema no pueda ser restaurado de manera óptima.

- Multiplexión de los archivos de control.

Para tener un respaldo del archivo de control en tiempo real en diferentes locaciones, el DBMS permite multiplexarlo simultáneamente en cada ubicación; es decir, la producción de varios archivos de control idénticos que serán actualizados cada vez que suceda alguna modificación estructural de la base de datos.

El sistema operativo ofrece otra opción: se pueden configurar los archivos "*en espejo*"; es decir, este medio permite generar copias del archivo de control en dos o más discos duros iguales excluyendo el que se encuentre en uso. Es recomendable implementar esta solución, ya que es capaz de soportar fallas del hardware, en tanto que la del DBMS no está preparada para tales eventualidades. En la figura 2.1 se distingue gráficamente el funcionamiento de cada uno de los dos procesos de respaldo.

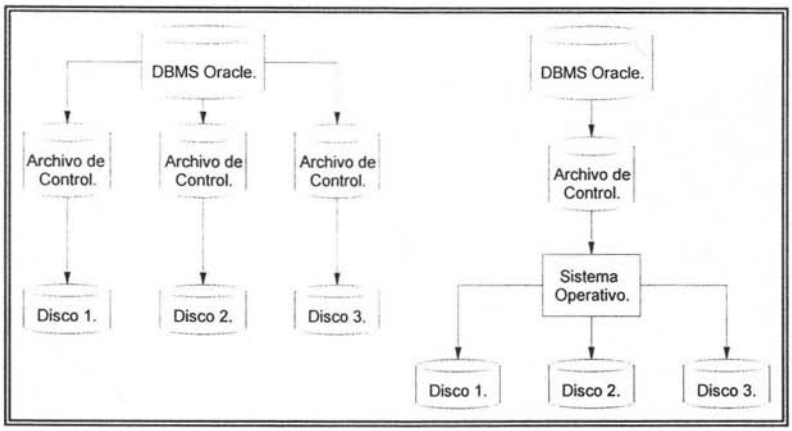


Figura 2.1 Comparación entre la multiplexión y el manejo en espejo de los archivos de control.

2.3.2 Administración de los archivos Online Redo Log.

Una de las estructuras esenciales para la recuperación de una base de datos es conocida como los archivos redo log, los cuales almacenan todos los cambios que se realicen en la misma. Cada una de las instancias que resida en la base de datos, tiene un grupo de archivos redo logs en previsión de que la instancia fallara.

Al instalar el DBMS Oracle, algunos de los parámetros que deben responderse son los relativos a la cantidad de archivos redo log que tendrá la instancia y cuál será su medida. En el capítulo 1 se estableció la forma en que los archivos redo log mantienen un funcionamiento cíclico (Figura 1.12), en consecuencia, si se está configurando una base de datos de producción, se requiere especificar que la cantidad de archivos y su medida sean lo suficientemente grandes para almacenar todos los cambios.

A. Desplegando información de los archivos redo log.

Existen dos vistas en el diccionario de datos que contienen información sobre los archivos redo log.

Nombre de la vista.	Descripción.
V$LOG	Identifica los archivos redo log, el número de miembros del grupo y cuáles archivos están en modo archive.
V$LOGFILE	Despliega el nombre y el status de los archivos redo log,

La manera de obtener la respuesta a cuáles son los grupos de archivos redo log que están en modo archive, es a través de la siguiente consulta:

```
SQL> SELECT group#, sequence#, status, archived FROM v$log;

    GROUP#  SEQUENCE  # STATUS        ARC
---------- ---------- ---------------- ---
         1      21529 CURRENT         NO
         2      21527 INACTIVE        NO
         3      21528 INACTIVE        NO
```

Como se puede apreciar, el archivo número 21529 es donde el DBMS está almacenando actualmente la información, en tanto que los redo log nombrados 21527 y 21528 se encuentran inactivos.

B. Multiplexión de los archivos redo log.

El DBMS Oracle ofrece la característica de multiplexar el conjunto de archivos redo log de cada instancia. En el capítulo anterior, se analizó el proceso LGWR que es el encargado de almacenar las transacciones en los archivos redo log. Cuando se ponen en modo de multiplexión, el proceso LGWR escribe la información de las transacciones en diferentes ubicaciones, eliminando con esto la posibilidad de una falla de recuperación debida a un error en el dispositivo de almacenamiento.

En comparación a los archivos de control, no es recomendable multiplexar los archivos redo log a través del sistema operativo, ya que podrían corromperse, en tanto que el proceso LGWR es transparente.

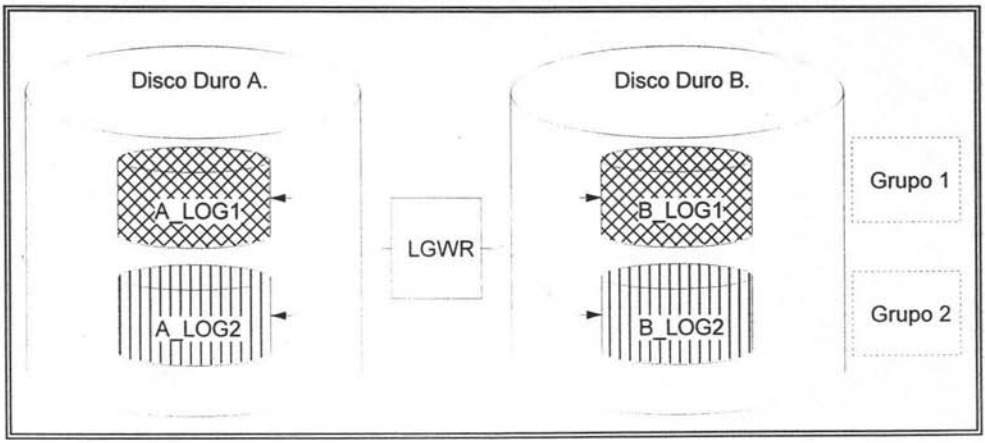


Figura2.2 El proceso LGWR escribiendo las transacciones de 2 diferentes grupos en varios discos duros.

Es altamente recomendable almacenar los conjuntos de archivos redo log en distintos discos duros o diversas particiones. En la figura 2.2 se muestra como se está administrando el proceso de multiplexion de los archivos redo log. Los archivos A_LOG1 y B_LOG1 pertenecen al mismo grupo, incluso son idénticos, pero están almacenados en diferentes ubicaciones y varían en el nombre.

C. Manejo de archivos redo log en modo archivelog.

Si se corre una base de datos en modo archivelog, el DBMS permite que se salven grupos de archivos redo log llenos de transacciones, a dichos archivos se les conoce como redo logs archivados (*archived redo logs*).

El modo archivelog es una operación para convertir archivos online redo log a archivos redo log archivados.

Los usos para el modo archivelog son:

- ✓ Recuperar una base de datos.
- ✓ Actualizar una base de datos en *stand-by*.

Si el modo archivelog está activado, el proceso LGWR no puede ser reutilizado y en consecuencia no se pueden sobreescribir los archivos redo log hasta que sean archivados. Por consiguiente, cada archivo redo log generado en archivelog, contiene una copia de los archivos online redo log creados desde la activación del modo archivelog. Por ello, cualquier cambio que se le hiciera a la base de datos será almacenado en los archivos. Cualquier eventualidad que aconteciera con su información, se puede solucionar usando los archivos redo log en modo archivelog.

D. Desplegando información de los archivos redo log en modo archivelog.

Existen algunas vistas en el diccionario de datos que permiten obtener información del modo archivelog.

En la siguiente tabla se especifica la información de cada una de las vistas concernientes al modo archivelog.

Vistas	***Descripción***
V$DATABASE	Identifica si la base de datos está o no en archive mode.
V$ARCHIVED_LOG	Despliega información de los archivos redo log en modo archive desde el archivo de control.
V$ARCHIVE_DEST	Describe las ubicaciones donde son almacenados los archivos redo log.
V$LOG	Despliega todos los grupos de archivos redo log de la base de datos. Indicando cuáles necesitan ser archivados.
V$LOG_HISTORY	Contiene la información histórica del proceso de archivelog.

Por ejemplo, en tratándose de conseguir información de los grupos de archivos redo log que necesitan ser archivados, se puede ejecutar el siguiente script:

```
SQL> SELECT group#, archived FROM sys.v$log;

    GROUP# ARC
---------- ---
         6 NO
         5 NO
         4 NO
         3 NO
         2 NO
         1 NO

6 rows selected.
```

Para saber el modo actual de archivado de una base de datos, se puede consultar a la tabla V$DATABASE:

```
SQL> SELECT log_mode FROM sys.v$database;

LOG_MODE
------------
NOARCHIVELOG
```

El comando svrmgrl, permite checar la información de los archivos redo log en modo archivelog.

```
SVRMGR> connect internal
Password:
Connected.
SVRMGR> archive log list;
Database log mode              Archive Mode
Automatic archival             Disabled
Archive destination            /oracle/app/oracle/product/8.1.6/dbs/arch
Oldest online log sequence     21527
Next log sequence to archive   21527
Current log sequence           21529
```

Con este comando es posible obtener la información necesaria de los parámetros de configuración del modo archivelog para la instancia en uso, entre ellos destacan los siguientes:

- ✓ Saber si la base de datos está funcionando actualmente en modo archivelog.
- ✓ Modo archivelog está en modo automático o normal.
- ✓ Destino específico de los archivos archive redo log.
- ✓ El archivo más viejo de la secuencia.
- ✓ El archivo actual que se está guardando.

E. Selección del modo archivelog.

Al implementar el modo archivelog deben considerarse algunos puntos con relación al desempeño de la base de datos y una futura recuperación de la información[14].

[14] **HOECHST**, Tim. **MELANDER**, Nicole, **CHABRIS**, Christopher. *"Guide to oracle"* . New York; México: McGraw-Hill, c1990.

F. Funcionamiento de una base de datos en modo noarchivelog.

Cuando una base de datos se está ejecutando en modo noarchivelog, el guardado de los archivos redo log es deshabilitado. El archivo de control indica que los grupos de archivos redo log que se encuentren llenos de transacciones no serán archivados.

El correr una base de datos en modo noarchive log tiene las siguientes consecuencias:

- ✓ Se puede restaurar mas no recuperar la información de la base de datos desde el último respaldo que se tenga.
- ✓ Se pueden ejecutar respaldos de la base de datos desde el sistema operativo únicamente cuando la base de datos no está en operación.

No obstante, la no utilización del modo archivelog es una manera de ahorrar recursos al sistema, lo que se ve reflejado en su desempeño.

G. Funcionamiento de una base de datos en modo archivelog.

Cuando una base de datos está en modo archivelog, Oracle requiere que los archivos redo log sean guardados. En el archivo de control se especifica el grupo de archivos redo log que no podrá usar el proceso LGWR hasta que sean archivados[15].

[15] CROOKS, Ted. "*Using Oracle*" Carmel, Indiana : Que. c1998.

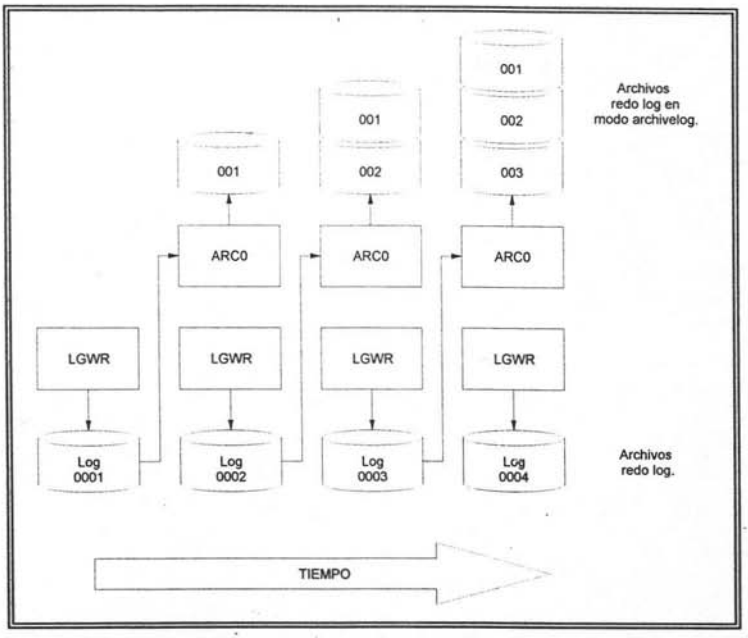


Figura 2.3 Archivos redo log usados en modo archivelog.

En la figura 2.3 se muestra el funcionamiento del modo alrchivelog. El proceso LGWR escribe un archivo redo log, el cual es archivado por el proceso ARCH. A cada nuevo archivo en formato archivelog le es asignado un número secuencial en el nombre, incrementándose en uno.

El modo archivelog presenta las siguientes ventajas:

- ✓ Un modo de respaldo en conjunción de los archivos redo logs y los redo logs en modo archived, que ofrece tanto la restauración como la recuperación de su información, garantizando que se pueden salvar todas las transacciones.
- ✓ Se pueden generar respaldos de tablespaces, mientras la base de datos está en operación.

- ✓ Se pueden recuperar respaldos de tablespace que estén offline.
- ✓ Se puede mantener una base de datos que esté en modo stand-by actualizada, con los archivos redo log en modo archivelog que genere la base de datos de producción.

Es necesario remarcar que cuando se está corriendo una base de datos en modo archivelog, es importante monitorear los recursos del sistema, sobretodo en cuanto a almacenamiento se refiere, ya que si una base de datos está realizando muchas transacciones, se generarán un número de archivos redo log en modo archive considerable, ocupando más espacio en disco duro y obviamente consumiendo recursos del sistema.

2.4 Diferentes tipos de respaldos en una base de datos.

El DBA es responsable de garantizar que la información de una base de datos esté respaldada y sea recuperable. Con el paso del tiempo, las bases de datos de alta disponibilidad tuvieron un cambio, el comercio electrónico u otras aplicaciones críticas tienen la necesidad de estar disponibles las veinticuatro horas, los siete días de la semana. Ello determinó un sentido específico en las estrategias de respaldo de los diferentes tipos de bases de datos.

Oracle proporciona una serie de opciones y procedimientos de respaldo que ayudan a proteger y restaurar las bases de datos. Si estas opciones se implementan correctamente, permiten realizar el proceso de copia de seguridad con efectividad.

2.4.1 Respaldos Físicos.

El término respaldo físico se refiere a la copia de los archivos que conforman tanto la arquitectura del DBMS como los propios de la base de datos. Este tipo de respaldos son los más frecuentemente ejecutados por los DBA. Para ello se requiere el conocimiento depurado de una estrategia de las señaladas en el punto "*2.3 Manejando estructuras de datos*" del cuerpo de esta investigación.

Los respaldos físicos se realizan desde el sistema operativo, sin embargo con esta premisa es necesario dividir los respaldos físicos en dos categorías:

- ✓ Respaldos físicos consistentes de la base de datos.

Los respaldos físicos consistentes de una base de datos son copias de sus archivos pero con la característica de que se encuentran sincronizados; para poder copiar estos archivos de manera simultánea, es necesario interrumpir momentáneamente la operación a la base de datos para evitar que el DBMS actualice algún archivo durante una transacción en proceso. Este tipo de respaldos también son conocidos como **respaldos en frío u offline**.

No es necesario respaldar el conjunto de archivos redo log en modo archivelog que haya generado el DBMS, ya que se están respaldando todos los archivos actualizados hasta ese momento. Este proceso es la única forma de respaldar bases de datos que no se encuentren en modo archivelog.

Algunos DBA realizan un respaldo en frío como parte del respaldo del sistema operativo. Esto significa que cuando el administrador del sistema realiza un respaldo de todo el sistema, también realiza un respaldo de los archivos del DBMS. El DBA debe asegurarse únicamente de que la base de datos se encuentre fuera de ejecución para realizar el respaldo desde el sistema operativo.

Existen herramientas nativas de Oracle que permiten realizar un respaldo en frío como "Enterprise Respaldo Utility". Sin embargo, la dirección constante a la que apunta el mercado es a tener la información disponible, por lo cual la herramienta para desarrollar este tipo de respaldos es el "Recovery Manager" conocido comúnmente como RMAN.

- ✓ Respaldos físicos inconsistentes de la base de datos.

Si el DBA se encuentra ante la circunstancia de que la base de datos debe de estar en operación continua durante las veinticuatro horas los siete días de la semana, es necesario realizar respaldos de una manera inconsistente. A estos respaldos también se les conoce como **respaldos en caliente u online.**

El respaldo de una base de datos es inconsistente porque ciertas estructuras del DBMS están siendo modificadas (datafiles principalmente) mientras que el respaldo está en progreso. Durante esta operación, la base de datos debe estar en modo archivelog para poder actualizar el respaldo de los cambios que no se vieran reflejados en el respaldo.

Decidir la conveniencia de ejecutar o no un respaldo inconsistente, depende de los requerimientos de disponibilidad de la información.

Los respaldos inconsistentes también son creados cuando la base de datos es respaldada después de una caída del DBMS o una falla al detener la operación de la base de datos. Este proceso es válido si se estuviera ejecutando el modo archivelog, porque con este conjunto de archivos, se podría formar un respaldo consistente.

- ✓ Desplegando la lista de archivos de un respaldo físico.

Antes de comenzar un respaldo físico, es necesario identificar los archivos que se copiarán. Ello determinará el tipo de proceso a ejecutar.

Sobre la vista V$DATAFILE se puede obtener un listado de los datafiles con la siguiente consulta:

```
SQL> SELECT name FROM v$datafile;

NAME
--------------------------------------------------------------------------------
/datafile1/oradata/bdweb/system01.dbf
/datafile1/oradata/bdweb/tools01.dbf
/datafile1/oradata/bdweb/rbs01.dbf
/datafile1/oradata/bdweb/temp01.dbf
/datafile1/oradata/bdweb/users01.dbf
/datafile1/oradata/bdweb/indx01.dbf
/datafile1/oradata/bdweb/drsys01.dbf
/oracle/app/oracle/product/8.1.6/dbs/prueba44
/oracle/app/oracle/product/8.1.6/dbs/prueba
/datafile1/oradata/bdweb/prueba2.dbf
/datafile1/oradata/bdweb/system02.dbf

11 rows selected.
```

Como puede notarse, el resultado obtenido es la ubicación y el nombre del archivo, pero si se quisiera saber a qué tablespace pertenecen, lo indicado es realizar un join con la vista V$TABLESPACE (Nota: TS# es un identificador del objeto).

```
SQL> SELECT t.name "Tablespace", f.name "Datafile"
  2  FROM v$tablespace t, v$datafile f
WHERE t.ts# = f.ts#
ORDER BY t.name;  3    4

Tablespace                     Datafile
------------------------       ------------------------------------------------
DRSYS                          /datafile1/oradata/bdweb/drsys01.dbf
```

No obstante, existe una vista llamada dba_data_files, que permite observar a qué tablespace pertenece cada datafile.

```
SQL> select file_name, tablespace_name from dba_data_files
  2  where tablespace_name like '%CALL_DAT%';

FILE_NAME                                      TABLESPACE_NAME
---------------------------------------------  ---------------------------------------
/home/oracle/billing/oradata/billing/call_dat01.dbf    CALL_DAT
/home/oracle/billing/oradata/billing/call_dat02.dbf    CALL_DAT
/home/oracle/billing/oradata/billing/call_dat03.dbf    CALL_DAT
/home/oracle/billing/oradata/billing/call_dat04.dbf    CALL_DAT
/home/oracle/billing/oradata/billing/call_dat05.dbf    CALL_DAT
```

✓ Desarrollando respaldos desde el sistema operativo.

Existen herramientas que permiten realizar un respaldo de la base de datos desde el DBMS, o bien con herramientas del sistema operativo. El instrumento que se escoja dependerá del sistema operativo en el que se esté trabajando.

Los respaldos desde el sistema operativo se dividen en tres categorías:

a) Respaldos completos de la base de datos.

Como se señaló con anterioridad, para desarrollar y obtener un respaldo consistente de todas las estructuras de la base de datos, se requiere suspender la ejecución de ésta. En tal caso, carece de relevancia el hecho de que la base de datos esté operando en modo archivelog, habida cuenta que con un respaldo de esta naturaleza, lo único que se produce es una restauración y no una recuperación.

Este respaldo completo es un buen punto de partida para poder recuperar una base de datos o implementar una base de datos espejo en stand-by, ya que si la base de datos estuviera en modo archivelog puede obtenerse hasta la última transacción almacenada en los archivos.

Por ejemplo, se pueden usar comandos como el "dd" o el "tar" en UNIX para generar respaldos de la base de datos. Estos scripts serían programados y ejecutados por el DBA o el administrador del sistema.

b) Respaldos de tablespaces y datafiles.

Cuando se desarrollan respaldos de un tablespace o un datafile desde el sistema operativo, es necesario tener la base de datos en modo archivelog. De no hacerlo así, no se podría restablecer la base de datos, ya que no se tendrían los archivos redo log, los cuales almacenan cualquier cambio efectuado en la misma.

c) Respaldos de archivos de control.

El archivo de control, como se mencionó en puntos anteriores, no puede ser modificado desde el sistema operativo por ningún usuario, en consecuencia deben realizarse ciertos procedimientos para su respaldo pero desde el DBMS. Existen dos formas de respaldar el archivo de control, en un archivo físico y mediante un archivo de tipo trace, ambos se analizan con más detalle en el Capítulo Cuarto.

2.4.2 Respaldos Lógicos.

El respaldo lógico o *export*, crea una copia lógica de los objetos de la base de datos y los almacena en un archivo binario. Esta herramienta provee un método simple para transferir objetos entre diversas bases de datos Oracle.

El export extrae las definiciones de los objetos y de las diferentes tablas desde una base de datos Oracle y las almacena en un archivo con formato binario en donde se reúnen toda clase de objetos.

Cuando se realiza una exportación de una base de datos Oracle, objetos (tales como tablas) son extraídos, seguidos de sus objetos relacionados (por ejemplo: índices, grants, etcétera), si existiera alguno, posteriormente se escribirá en un archivo de tipo "dmp" (Figura 2.4).

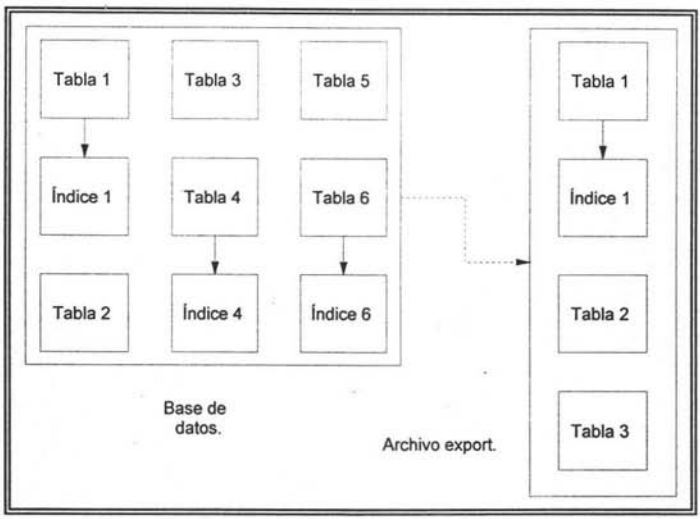


Figura 2.4 *Esquema de exportación.*

El mecanismo de respaldo del export proporciona una recuperación determinada hasta un punto específico en el tiempo y no se puede utilizar con redo log archivados. No existe la noción de importar una tabla y rehacerla utilizando los redo logs archivados. Estos son parte del respaldo físico que graban en disco información especifica sobre los cambios realizados en los bloques de datos. El archivo que genera el *export* es esencialmente un archivo donde se almacenan sentencias SQL que utiliza el *import* para generar información en la base de datos.

➢ Tipos de respaldos lógicos:

Existen tres tipos de estrategia de exportación, las cuales ofrecen alternativas en cuanto al espacio en almacenamiento y por consecuencia en tiempo:

✓ Exportaciones en aumento o creciente.

Una exportación en aumento, respalda únicamente las tablas que han cambiado desde la más reciente exportación completa, en aumento o acumulada. Una exportación en aumento copia la definición de las tablas y los datos que se encuentran en ella, no sólo los que se hayan modificado. Se desarrollan más exportaciones en aumento que exportaciones completas o acumuladas.

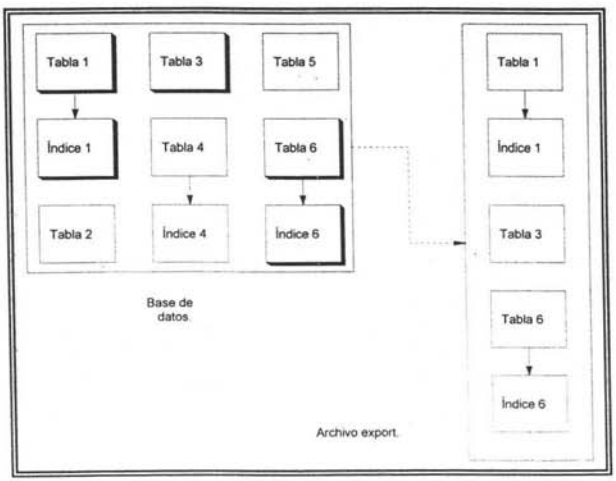


Figura 2.5 Ejemplo de exportación en aumento A.

Como se observa en la figura 2.5, a partir de una exportación completa (figura 2.4) los objetos que han sido modificados en la base de datos son los que se copian a un nuevo archivo dmp a través de una exportación en aumento.

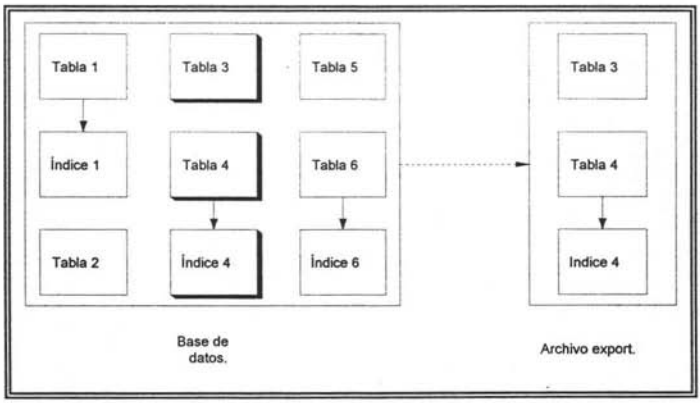


Figura 2.6 Ejemplo de exportación en aumento B.

En la figura 2.6 se ilustra una exportación en aumento a partir de la figura 2.5; si la tabla número 3 sufriera algún cambio, nuevamente sería exportada; en comparación, la tabla 4 no había sido copiada en la más reciente exportación.

- ✓ Exportaciones acumuladas.

  Como su nombre lo señala, una exportación acumulada[16] respalda las tablas que han sido modificadas desde la última exportación acumulada o completa.

16 **SAYLES,** Jonathan S. *"How to use oracle SQL *plus"* Wellesley, Massachusetts : Qed information sciences, 2001.

Una exportación acumulada comprime un número indefinido de exportaciones en aumento o crecientes de un solo archivo. En este caso no sería necesario almacenar los archivos de exportaciones en aumento ya que el respaldo acumulado los reemplaza.

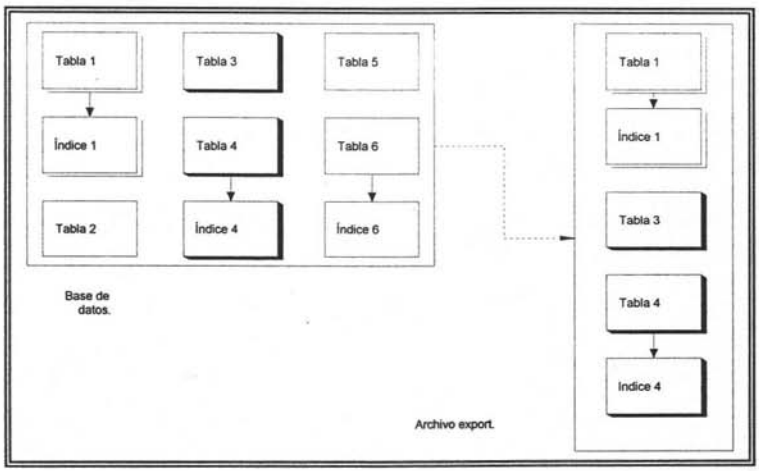


Figura 2.7 Ejemplo de exportación acumulada.

En la figura 2.7 se muestra cómo las tablas 1, 3 y 4 (con sus respectivos objetos) han sufrido diferentes modificaciones durante distintas exportaciones, es así como la exportación acumulada los respalda.

✓ Exportaciones completas.

Una exportación completa es la base de una exportación acumulada o en aumento. Es equivalente a exportar todos los objetos de la base de datos, con la excepción de que también copia las tablas respaldadas por una exportación en aumento o acumulada.

Existen dos formas de invocar a la herramienta export, de modo interactivo sería de desde el shell o prompt del sistema operativo, ejecutando el comando y posteriormente contestando a cada una de las preguntas que hace el sistema.

Por ejemplo, si se quisiera exportar de una manera interactiva una tabla, el proceso sería de la siguiente forma:

```
[oracle@billing oracle]$ exp
Export: Release 8.1.6.1.0 - Production on Mon Dec 2 12:38:58 2002
(c) Copyright 1999 Oracle Corporation.  All rights reserved.


Username: system
Password:

Connected to: Oracle8i Enterprise Edition Release 8.1.6.1.0 - Production
With the Partitioning option
JServer Release 8.1.6.0.0 - Production
Enter array fetch buffer size: 4096 >

Export file: expdat.dmp >
(1)E(ntire database), (2)U(sers), or (3)T(ables): (2)U > 3
Export table data (yes/no): yes >
Compress extents (yes/no): yes >
Export done in US7ASCII character set and US7ASCII NCHAR character set
About to export specified tables via Conventional Path ...
Table(T) or Partition(T:P) to be exported: (RETURN to quit) > scott.emp

Current user changed to SCOTT
. . exporting table                         EMP      15 rows exported
Table(T) or Partition(T:P) to be exported: (RETURN to quit) >

Export terminated successfully without warnings.
```

De una manera interactiva, la herramienta export permite especificar diferentes parámetros; por ejemplo, si se van a exportar los datos de las tablas, respaldo por usuario, por objeto o tabla, de una manera simple.

A nivel comando, el método de invocar el export se diferencia en lo que respecta a introducir todos los parámetros en la línea de comando.

A manera de muestreo:

```
[oracle@billing oracle]$ exp system/manager file=tesis.dmp tables=scott.emp
Export: Release 8.1.6.1.0 - Production on Mon Dec 2 12:49:44 2002
(c) Copyright 1999 Oracle Corporation.  All rights reserved.

Connected to: Oracle8i Enterprise Edition Release 8.1.6.1.0 - Production
With the Partitioning option
JServer Release 8.1.6.0.0 - Production
Export done in US7ASCII character set and US7ASCII NCHAR character set

About to export specified tables via Conventional Path ...
Current user changed to SCOTT
. . exporting table                         EMP      15 rows exported
Export terminated successfully without warnings.
```

Aunque también se pueden especificar estos parámetros mediante un archivo de parámetros[17], ejecutando la herramienta de la siguiente forma:

```
billing1:bd_aux> exp system/manager parfile=tesis.dmp
```

Este archivo está compuesto con los parámetros que se especifican en los siguientes puntos.

[17] PERRY, James T. LATEER Joseph g. "*Understanding Oracle*", San Francisco : Sybex, c1999.

➢ Características de los respaldos lógicos.

Un respaldo lógico es aquel que copia los datos de la base de datos y no registra la localización de los datos[18].

La herramienta *export* que ofrece Oracle, se puede utilizar para realizar respaldos lógicos de la base de datos, copiando los datos y las definiciones de la misma almacenándolas en un archivo binario del sistema operativo en formato interno de Oracle.

Este tipo de resguardos son comparables con los respaldos en caliente u online, ya que necesitan que la base de datos esté en ejecución. Esto se debe a que la herramienta del export ejecuta un proceso denominado snapshot, que obtiene un reporte instantáneo de todo lo que se va a copiar, verificando con ello la consistencia en las lecturas de las tablas.

Usualmente, el respaldo de exportación consume mayor tiempo que el que requerido para un respaldo físico. Si la exportación se realiza a disco, o si se cuenta con múltiples unidades de cintas y se exporta a ellas, se pueden realizar sesiones de exportación en paralelo para minimizar el tiempo de su completa exportación.

[18] **INMON,** William H. *"Using oracle to build decision support systems"* Wellesley : Qed information sciences, c1999.

A continuación se indican algunas de las ventajas de utilizar la herramienta *export*.

- ✓ Una de las principales ventajas de un respaldo lógico estriba en que se pueden detectar los bloques de datos corruptos al realizar una exportación lo que propicia que el procedimiento de exportación falle. Entonces, es necesario solucionar el fallo antes de intentar realizar un respaldo lógico de nuevo.
- ✓ La exportación proporciona un nivel adicional de protección frente a errores de usuario o fallos estructurales. Por ejemplo si un usuario eliminara una tabla accidentalmente, sería muy fácil restablecerla con la herramienta import, comparado con una restauración incompleta de un respaldo físico.
- ✓ La exportación ofrece una gran flexibilidad en la selección de los datos y definiciones que se requieren exportar.
- ✓ Se pueden realizar exportaciones completas, incrementables o acumulativas.
- ✓ Los archivos de exportación se pueden importar en cualquier base de datos en la máquina actual. Ese archivo se puede compartir desde la red e importar los datos desde una máquina remota.

Sin embargo, una de las desventajas de utilizar la herramienta *export*, es que el proceso podría resultar muy lento si se exportan grandes cantidades de datos. Es recomendable realizar una exportación completa una vez al mes si es posible, lo que coadyuvaría a mantener una alta disponibilidad de la base de datos en el caso de que se requiera si una recuperación en el ámbito de objetos.

Esta clase de respaldo debería realizarse en forma adicional a los respaldos físicos. Asimismo dependiendo del fallo, es el método de restauración que se debe utilizar.

➢ Parámetros de los respaldos lógicos.

En la siguiente tabla se muestra una lista de parámetros que se pueden utilizar para controlar una exportación:

Parámetro	Valor predefinido	Descripción
USERID	Indefinido	**El nombre y contraseña del usuario que realiza la exportación.**
BUFFER	Dependiente del sistema operativo	**El tamaño en bytes del buffer utilizado para buscar files de datos. Si se especifica cero, o si la tabla contiene datos de tipo long sólo se busca una fila a la vez.**
FILE	Expdat.dmp	**El nombre del archivo de salida binaria, creado por la herramienta export al nivel del sistema operativo.**
GRANTS	SÍ	**Indica si se exportan los permisos.**
INDEXES	SÍ	**Señala si se exportan los índices.**
ROWS	SÍ	**Establece si se exportan los registros de las tablas, si se configuran o sólo se exportan las tablas sin los datos.**
CONSTRAINTS	SÍ	**Advierte si se exportan las restricciones.**
COMPRESS	SÍ	**Informa si se comprimen los datos de las tablas en un extent de la exportación.**
FULL	No	**Observa si se exporta la información completa de la base de datos.**
OWNER	Usuario Actual	**Despliega una lista de los nombres de los usuarios cuyos objetos se van a exportar.**
TABLES	Indefinido	**Describe una lista de nombres de tablas a exportar.**
RECORDLENGHT	Dependiente del sistema operativo	**Señala la longitud de bytes del registro de archivo.**
INCTYPE	Indefinido	**Tipo de exportación incremental.**
RECORD	SÍ	**Indica si se realiza una exportación incremental.**
PARFILE	Indefinido	**Nombre del archivo de parámetros de la exportación.**

2.4.3 Herramienta RMAN (Recovery Manager).

La herramienta RMAN (Recovery Manager en Inglés), maneja las operaciones de respaldo y recuperación de bases de datos Oracle. RMAN usa la información acerca de la base de datos para automáticamente localizar los datos y dependiendo de la acción, restaurar o recuperar datafiles, archivos de control y archivos redo log.

El RMAN obtiene la información de cualquier archivo de control relacionado a la base de datos o desde un repositorio de información llamado catálogo de recuperación, el cual es mantenido por el RMAN:

➢ Características del Recovery Manager.

La herramienta RMAN es una interfase de línea de comando (CLI command line interface en Inglés), se ejecuta como si fuera un proceso más del DBMS Oracle con la función de respaldar, recobrar o restaurar la base de datos a la que esté conectado.

Las versiones del software de RMAN dirigen al servidor de procesos de Oracle. Este servidor se encarga de leer el datafile, el archivo de control o el archivo redo log que está siendo respaldado o viceversa creando los mismos objetos que se están restaurando.

RMAN desarrolla procedimientos de respaldo y recuperación de bases de datos y simplifica las tareas del administrador en este sentido. La herramienta RMAN ofrece:

- ✓ Configura operaciones de respaldo de bases de datos con periodicidad.
- ✓ Genera una bitácora con todas las acciones que generó RMAN.
- ✓ Con el catálogo de recuperación se pueden automatizar operaciones de respaldo y recuperación de bases de datos.

- ✓ RMAN puede realizar operaciones de restauración y respaldo en paralelo.
- ✓ Se pueden desarrollar respaldos de una base de datos, por datafile o por tablespace.
- ✓ Se pueden direccionar los respaldos a un disco duro local o en red, sin necesidad de utilizar cualquier administrador de medios.

Para realizar respaldos en un medio secuencial (un medio secuencial puede ser una cinta) utilizando RMAN, es necesario tener un software administrador de medios que trabaje con el software de Oracle.

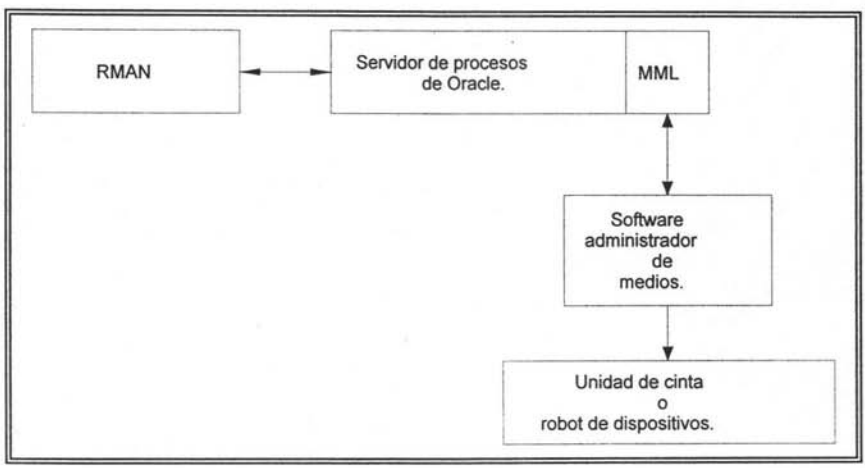


Figura 2.8 Arquitectura del software administrador de medios y el DBMS Oracle.

En la figura 2.4 se muestra la manera como se efectúa el empalme entre el DBMS y el software administrador de medios. Oracle llama a librería de mantenimiento de medios (MML Media Management Library en Inglés) para realizar la conexión con el software administrador de medios y el RMAN.

Esto con la finalidad de poder mandar los respaldos a diferentes dispositivos secuenciales (Cintas, robots, discos magneto–ópticos, entre otros).

2.5 Cuadro comparativo de características de métodos de respaldo.

Este cuadro comparativo entre los diversos medios de respaldo de bases de datos, está fundamentado en las diferentes características expuestas en este capítulo.

Característica	RMAN	Respaldo físico	Respaldo lógico
Respaldo de la base de datos, mientras no se encuentra en operación.	Sí lo soporta, pero requiere que la base de datos esté montada.	Sí lo soporta.	No lo soporta.
Base de datos en operación.	Se necesitan los comandos BEGIN/ END RESPALDO.	Se generan más archivos redo log cuando se usan los comandos BEGIN/ END RESPALDO.	Requieren segmentos de rollback para generar un respaldo consistente.
Respaldos en aumento. (Acumulativo).	Sí lo soporta. Respalda todos los bloques de datos que sean modificados.	No lo soporta.	Sí lo soporta pero no es realmente un respaldo en aumento. Ya que respalda sólo los bloques de datos de los cuales tiene conocimiento.
Detección de bloques corruptos.	Sí los detecta. Identifica los bloques de datos corruptos y los escribe en V$RESPALDO CORRUPTION.	No lo soporta.	Sí lo soporta. Identifica los bloques de datos corruptos en la bitácora de la herramienta export.
Respaldos automáticos de instancias.	Sí lo soporta. Establece el nombre y las ubicaciones de todos los archivos a ser respaldados.	No es soportado. Los archivos a respaldar deben ser seleccionados manualmente.	Sí lo soporta. Lo puede realizar por usuario, objeto o la base de datos completa.
Desarrollo de respaldos de catálogos de recuperación.	Sí lo soporta. Realiza un catálogo de recuperación.	No lo soporta.	No lo soporta.
Desarrollar respaldos a un dispositivo	Sí lo soporta. Necesita un software	Sí lo soporta. Es una administración	Sí lo soporta.

secuencial.	administrador de cintas.	manual por el software especial.	
Respaldos de archivos de parámetros y de archivos de passwords.	No lo soporta.	Sí lo soporta.	No lo soporta.
Sistema operativo en diferentes lenguajes.	Sí lo soporta.	Depende del sistema operativo.	Sí lo soporta.

2.5.1 ¿Cuál es el mejor método de respaldo?

El mejor método de respaldo es el que se adapte a las necesidades del negocio que maneja la base de datos. Deben ser considerados diversos factores dentro del proceso de respaldo de la base de datos y tomar en cuenta aspectos de la restauración de la información. Mientras que para una organización existen situaciones en las que el tiempo es factor fundamental y la recuperación de la información debe ser inmediata, en otros casos es suficiente una actualización semanal.

Por lo tanto, los procesos de respaldo y recuperación de bases de datos, trabajan en conjunto con las necesidades del negocio. Es así que la estrategia que se adoptará dependerá directamente de los requerimientos empresariales.

2.6 Consideraciones del hardware.

Aunque todas las bases de datos Oracle se componen de las mismas estructuras, las opciones de las que se dispongan dependerán de la plataforma del hardware y del sistema operativo que se utilice. En el aspecto del respaldo de bases de datos, Oracle permite implementar soluciones diversas dependiendo de las necesidades.

ESTA TESIS NO SALE
DE LA BIBLIOTECA

Existen diversas soluciones que simplifican los procesos de respaldo, además de dar un margen de error si existiera una eventualidad.

A. Servidores de bases de datos autónomos.

La configuración conceptual más sencilla de una base de datos consta de un servidor, que accede a una única base de datos en una máquina host (servidor), con un solo disco duro. Como se muestra en la figura 2.5 todos los archivos se almacenan en el único dispositivo.

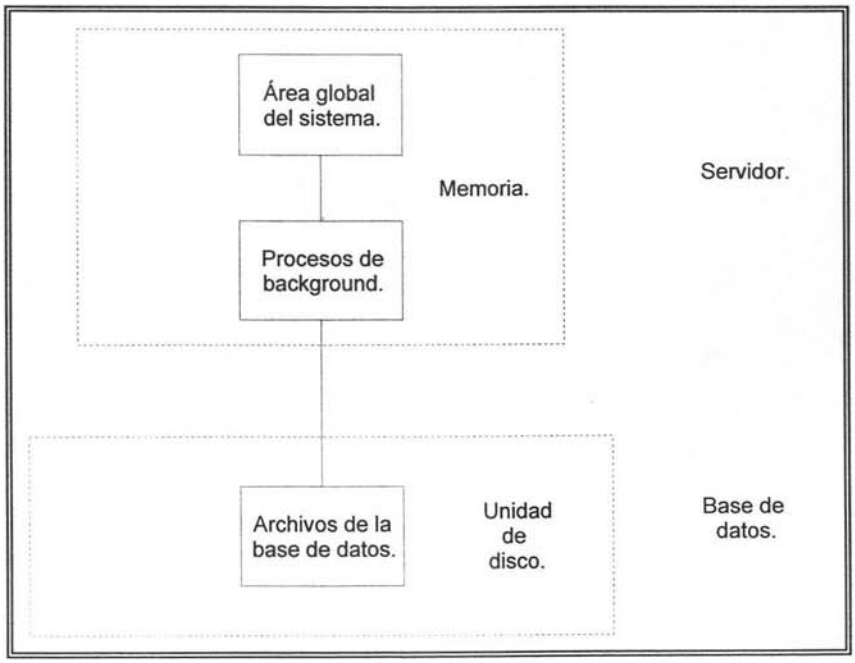


Figura 2.9 Configuración simple de una base de datos Oracle.

B. Arreglos Redundantes de discos independientes (RAID).

RAID (Redundant Array of Independent (o Inexpensive) Disks por sus siglas en Inglés) es una técnica que fue diseñada para proveer de velocidad y confiabilidad así como para incrementar la capacidad de almacenamiento de un sistema, usando múltiples discos duros en lugar de utilizar solamente uno.

Básicamente RAID toma varios discos duros y los utiliza como si estuviera utilizando uno de mayor capacidad, los beneficios que obtiene dependen del nivel de RAID que se esté implementando.

Existen dos tipos de arreglos de discos, los lógicos y los físicos. Mientras un arreglo físico puede ser dividido o agrupado en varios arreglos lógicos, éstos pueden ser divididos en dispositivos de almacenamiento administrados por el sistema operativo.

Es importante remarcar tres conceptos fundamentales en el funcionamiento de RAID:

- ✓ Réplica (Mirroring).

El tener una réplica dentro de un arreglo de discos significa tener una copia en tiempo real del mismo dato en diferentes ubicaciones. El sistema de arreglo de discos duros escribe la información en dos archivos idénticos. Ésta es una de las dos técnicas de redundancia que utiliza RAID para proteger probables pérdidas de información.

El beneficio estriba en que cuando un disco o un arreglo falle, la operatividad del sistema puede continuar sin necesidad de interrupciones.

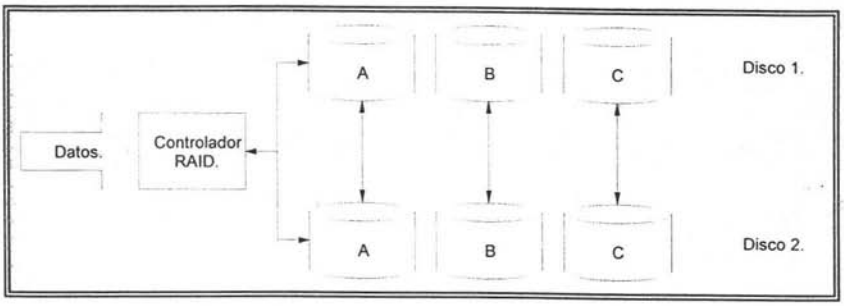


Figura 2.10 Réplica en un arreglo de discos RAID.

Como se puede observar en la figura 2.10, el controlador del RAID tiene la tarea de definir cómo se van a almacenar y procesar los datos a través del arreglo de disco. Existen tres archivos (A, B y C) que están duplicados en un segundo disco duro. Esto permite que en caso de una caída, tanto la recuperación como la puesta en marcha sea instantánea.

- ✓ Paridad.

Paridad es otra técnica de redundancia que implementa RAID. Este término es comúnmente usado en dispositivos de comunicaciones, como el caso de módems. También es usado en dispositivos de memoria. En RAID es muy similar este concepto.

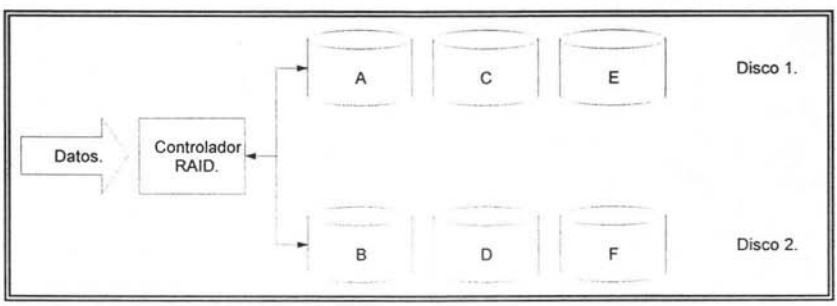
Dicho en otras palabras, si se tuviera un número indeterminado de elementos, se podría generar un elemento de paridad obteniendo un total de X+1 elementos. Aún si alguno o varios de estos X+1 elementos se perdieran, la información continuaría intacta, debido a que el elemento de paridad tomaría la posición y obviamente la información del elemento fallido. Para el funcionamiento de esta operación no se requiere una cantidad específica de bits.

El nivel de tolerancia de los discos duros no es tan alto como el de un sistema de réplica en uso, ya que la paridad no siempre se utiliza en un solo disco duro, sino que está distribuida en todo el arreglo.

No debe pasar desapercibido que esta configuración procesa más información, ya que en cada ocasión que se requiera de almacenamiento, es necesario que el controlador del RAID verifique en qué disco duro se debe depositar la información en específico.

✓ Stripping.

Stripping es una técnica que permite mejorar el desempeño del arreglo de discos, distribuyendo la información en todos los discos duros. El principio fundamental detrás del *stripping* es el paralelismo. A manera de ejemplo, si se llegara a formar un archivo de dimensiones considerables, del tamaño de varios terabytes, su lectura se dificultaría en gran medida, ocasionando una pérdida de tiempo importante, sin embargo si fuera fraccionado en varios discos y se pudiera leer simultáneamente el desempeño mejoraría ostensiblemente.



A mayor número de discos, la velocidad será incrementada. (Figura 2.10).

Figura 2.11 Ejemplo de Stripping.

C. Niveles de RAID.

Existen seis niveles básicos dentro de la configuración del RAID, los cuales manejan básicamente los tres conceptos anteriores.

- Nivel 0.

El nivel más sencillo, sólo requiere utilizar *stripping*. La redundancia es un parámetro que no está establecido en esta modalidad. No es recomendado para aplicaciones críticas.

- Nivel 1.

Este nivel es usualmente implementado como réplica (*mirroring*). Dos copias idénticas de los datos son almacenadas en sendos discos duros. Cuando un dispositivo falla, el otro continua con la operación del sistema. La recuperación de éste es asequible, ya que únicamente se copia la información del respaldo al dispositivo base.

- Nivel 2.

Esta configuración es la "*oveja negra*" de RAID, ya que no implementa ninguno de sus elementos estándar (Réplica, *Stripping* o Paridad). Utiliza algo similar a *stripping* a nivel de bit con el código Hamming ECC. Además es la única que varía en este aspecto. Los datos son divididos y almacenados en discos redundantes. Los bits redundantes son calculados por el código Hamming (código corrector de errores [ECC]).

Cada vez que surge una escritura en el arreglo de discos, los códigos de ECC son calculados y almacenados en discos especiales de ECC. Cuando los datos son leídos de nueva cuenta, también se leen los códigos para verificar si no existen errores en la lectura, Es decir, se trata de una especie de paridad pero con sus complicaciones. Este sistema de protección es usado frecuentemente en los dispositivos de memoria.

- Nivel 3.

Este nivel usa *stripping* a valor de byte con paridad dedicada. En otras palabras, los datos son divididos en el arreglo a nivel de byte con un disco duro manteniendo la información redundante. La idea detrás de este segmento es de dividir los datos, mejorando notablemente el desempeño y usando paridad dedicada la cual toma cuidado de la redundancia.

- Nivel 4.

Este apartado guarda semejanza con el anterior. Su diferencia radica en que utiliza *stripping* a nivel de bloque de datos y no de byte. La ventaja es que se puede definir el tamaño del stripping requerido para las aplicaciones.

- Nivel 5.

RAID 5 utiliza *stripping* a nivel de bloque y paridad distribuida. Esta sección trata de remover el cuello de botella de un dispositivo único de paridad. Con el uso de un algoritmo de paridad, este nivel escribe los datos a través de todos los dispositivos de almacenamiento así como los datos relativos a la paridad. Con esto se elimina el cuello de botella que se formaba cuando todos los datos de paridad se escribían en un solo dispositivo.

D. Servidores de bases de datos con arreglos de discos duros (RAID).

Cuando se disponen de varios discos duros, los archivos de la base de datos, pueden separarse en dispositivos distintos. Al efectuarse esta división, se optimiza el rendimiento de la base de datos, ya que se reduce el tiempo de acceso a la información. Durante su funcionamiento, lo habitual es que se requiera información de varios archivos para realizar una transacción o una consulta. Si los archivos no se distribuyen entre múltiples discos, el sistema tendría que leer varios archivos del mismo disco duro en forma concurrente.

Otra ventaja que ofrece el tener la base de datos almacenada en un arreglo de discos duros, es que se pueden implementar soluciones de multiplexión de archivos de control o de archivos redo log en modo archive. En la figura 2.6 se puede apreciar el funcionamiento de la multiplexión de los archivos de control con varios discos duros.

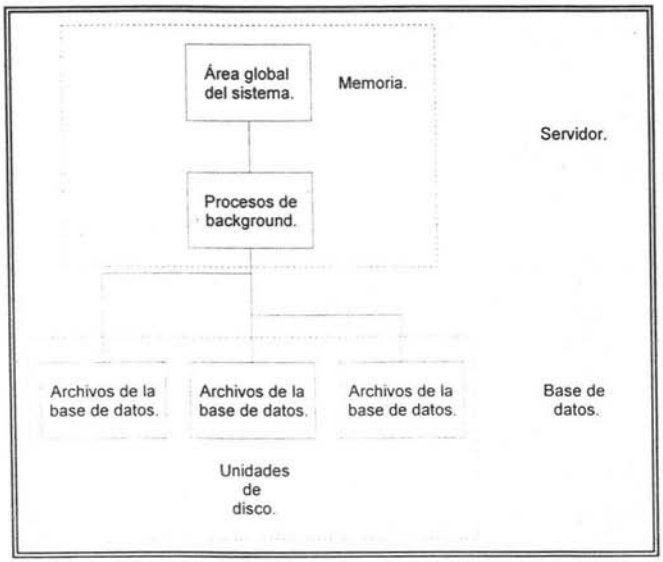


Figura 2.12 Base de datos almacenada en un arreglo de discos duros.

E. Fuentes de poder redundantes.

Cada servidor en el que resida una base de datos Oracle, puede sufrir una caída si existiera algún problema con el suministro eléctrico. Una característica que se puede implementar en el hardware es la adición de fuentes de poder redundantes.

Si alguna de estas fuentes de poder fallara, el servidor no va a caer dado que cada fuente es capaz de correr el servidor por sí solo indefinidamente, lo que permite quitar la fuente dañada e implementar una de respaldo recuperando la redundancia para con posterioridad dar servicio a la unidad que falló. Durante todo esto, los sitios en la red nunca van a estar abajo.

F. Componentes de hardware de reserva.

Como se expuso en el punto anterior, es necesario manejar un stock de componentes de hardware de reserva, en caso de posibles fallas de discos duros y fuentes de poder. Es necesario confrontar las especificaciones de cada uno de estos dispositivos para evitar incompatibilidades.

G. Robots y almacenamiento de información secuencial.

En los albores de la computación, la información se almacenaba en tarjetas de papel perforadas; tiempo después aparecieron cintas de papel que funcionaban bajo el mismo principio, codificaban la información a través de las perforaciones, pero la cinta ofrecía más capacidad de almacenamiento.

Poco después aparecieron las cintas magnéticas y su evolución fue tal que se llegó a manejar como el dispositivo más común para almacenar información (floopy).

Las cintas han evolucionado junto con los discos duros; nadie pretende usar una cinta para realizar consultas, pero sí para guardar información histórica, la cual, no es práctico guardarla en discos ya sean sencillos o arreglos de disco. La información que se almacena en discos es aquella sobre la cual se va a tener constante consulta, en tanto, la que no va a ser consultada con frecuencia, se reunirá en cintas, en este caso, los respaldos de las bases de datos. Actualmente las cintas son de alta velocidad y capacidad, v.gr. una cinta que puede almacenar 60 gigabytes y puede ser escrita o leída en menos de veinte minutos. Las cintas, al igual que los discos, se han agrupado en arreglos de cintas, a las cuales se les llama "robots". Estos robots "saben" lo que han almacenado y rotan las cintas, a manera de mantener la integridad de la información de un sistema.

H. ¿Por qué es importante tener una copia del respaldo de nuestra base de datos en una caja de seguridad?

La principal característica del site de cómputo donde resida el servidor de la base de datos, es que se encuentre a salvo de errores humanos. Factores como suministro eléctrico, temperatura y ubicación física deben ser primordiales. Sin embargo siempre existe la posibilidad de un imprevisto, en donde el site llegara a presentar daños irreversibles. En estos casos, se debe mantener un respaldo de la base de datos, de cualquier tipo en una caja de seguridad de un banco o una institución que se hiciera responsable por él. Es importante remarcar que sería el único medio para poder recuperar la información.

Capítulo III Técnicas de restauración y recuperación de una base de datos con el DBMS Oracle.

Debido al avance sostenido de la tecnología en materias de hardware y software, la relación de eventos que podrían ocasionar la caída de una base de datos se ha resumido a una sola dirección: El error humano está catalogado como el factor principal por el que una base de datos sufre averías.

Previo a la comprensión de los principios y estrategias de la recuperación, es preciso conocer y entender las estructuras de datos utilizadas en este proceso. El DBA debe dominar la arquitectura del DBMS así como su funcionamiento interno, a efecto de estar en aptitud de diagnosticar certeramente la falla y emplear el proceso de recuperación o restauración adecuado.

Asimismo, debe tener la pericia suficiente en la operación de las herramientas que Oracle ofrece para restaurar la información de la base de datos, como en la especie destacan import y RMAN (Recovery Manager por sus siglas en Inglés) esta última, que reúne una gran cantidad de opciones para una recuperación total o parcial de la base de datos.

De esta manera, el proceso de restauración se compone esencialmente de tres elementos:

- ✓ Se debe contar con un respaldo de la base de datos, soporte primordial del procedimiento de recuperación de la información.
- ✓ De conformidad a las características de la falla, un diagnóstico correcto resulta de vital importancia, sin soslayar que la base de datos no puede estar fuera de operación por un tiempo considerable.
- ✓ Finalmente, la realización oportuna del proceso de recuperación o restauración.

La etapa de restauración opera conjuntamente con la de respaldo, cuyas estrategias estarán siempre determinadas por las características especiales del negocio.

3.1 Conceptos del funcionamiento interno de la recuperación.

Las definiciones que se exponen a continuación contienen algunas estructuras de datos fundamentales empleadas usualmente en la recuperación.

3.1.1 Creación y estimación del redo.

En el capítulo 2 observamos la forma en que los archivos redo log almacenan los cambios realizados a la base de datos. Para concretar su exposición, se considera necesario precisar cuál es la información que guardan dichos archivos.

- Vector de cambio.

Un vector de cambio describe una modificación específica efectuada sobre un bloque de información único de la base de datos. Contiene, entre otras referencias, un número de versión, el código de operación de la transacción y la dirección del bloque de datos al que pertenece dicha modificación[19]. El número de versión se copia desde el referido bloque de datos. Mientras la restauración se lleva a cabo, Oracle lee el vector de cambio y aplica el proceso al bloque de datos adecuado. Cuando es administrado un vector de cambio al bloque de datos, se adiciona en uno el número de versión.

[19] **WEBB**, Kenneth & **LLAFRENIERE**, Lori *"1950- Oracle distributed systems: A c programmer's development guide"* Blue Ridge Summit, Pensylvania : Windcrest, c1999.

➢ Contenido y registro de los archivos redo log.

Un registro de redo es un grupo de vectores de cambio que describen un *cambio atómico* el cual consiste en dos o más modificaciones en un bloque físico; un ejemplo de ello sería la inserción de un registro en una tabla, puede envolver cambios en los bloques de diferentes índices y de la tabla misma, así como en los bloques de los segmentos de rollback.

Los múltiples registros de redo son generados por algunas transacciones y cada uno de ellos puede contener un conjunto de vectores de cambio. La restauración garantiza que éstos sean aplicados de manera integral o bien, ninguno de ellos, independientemente del tipo de fallo que se presente. Esto se debe a que la unidad de recuperación es una transacción y por ende, todos los cambios son aplicados o no unitariamente.

A continuación se reseña un ejemplo claro de la constitución de los vectores de cambio y registros de redo. Una sentencia SQL que actualiza la tabla scott.emp:

```
SQL> update emp set empno=1234 where empno=7369;
1 row updated.
SQL> commit;
Commit complete.
```

Cuando se ejecuta la sentencia update, la secuencia de operaciones es la siguiente:

1. Se generan los vectores de cambio del registro de redo.
2. Se salva el registro de redo en un buffer de redo log (El cual, finalmente se escribe en el archivo redo log en disco).
3. Se aplican los vectores de cambio a los bloques de datos.

En el ejemplo en cuestión, el registro de redo generado en el paso 1 contiene tres vectores de cambio:

1. La transacción debe escribir una entrada de undo en la tabla de transacción del segmento de rollback. En razón a que dicha tabla es también otro bloque de datos, la introducción de una entrada de undo modificaría este bloque y consecuentemente genera redo. Por ello, el primer vector de cambio de registro de redo contiene el cambio de la tabla de la transacción.

2. Posteriormente se debe almacenar el valor anterior de EMPNO(7369) en un bloque dentro del segmento de rollback. Esta es otra modificación a un bloque de la base de datos, y por consiguiente genera redo. Por tanto, el segundo vector de cambio redo para el bloque de undo.

3. El último cambio y el más obvio, se efectúa al bloque de datos donde se modifica el valor de EMPNO a 1234. Por tanto, el tercer vector de cambio es el redo para el bloque de datos.

El registro de redo para esta transacción contiene tres vectores de cambio:

- ✓ Cambio a la tabla de transacción del segmento de rollback.
- ✓ Cambio al segmento de rollback del bloque de datos.
- ✓ Cambio al bloque de segmento de datos perteneciente a la tabla emp.

Por supuesto, éste puede no ser el único registro de redo que se generó. Consideremos que si la tabla EMP tiene un índice sobre la columna EMPNO, lo viable es que también debe ser modificado el índice lo cual originará un segundo registro de redo (que a su vez contendrá varios vectores de cambio). De forma similar, si posterior a la transacción se ejecuta COMMIT, se creará un tercer registro de redo.

En esa tesitura, ante la pérdida de datos, la recuperación debe efectuarse *hacia adelante* dado que la unidad de recuperación es una transacción y como se señaló en líneas anteriores, se aplicarían todos los registros de redo para mantener la consistencia de la base de datos, o bien no se aplicaría ninguno de ellos[20].

- Cálculo de la cantidad de archivos redo log.

Para lograr una aproximación respecto a cuántos archivos redo log son necesarios en una instalación de Oracle, se pueden utilizar los procedimientos que se enlistan enseguida: El primero estima la cantidad de archivos redo que se generan en un día; el segundo obtiene el número de archivos redo generados por una transacción específica. Conociendo esta información y la tasa de transacciones, es posible calcular la cantidad de archivos redo log que es necesario especificar durante el proceso de instalación.

La orden ***archive log list***, proporciona la información consultando el número de la secuencia del archivo redo log, por ejemplo:

[20] SMINE, Hatem *ob.cit.*

```
SVRMGR> connect internal
Password:
Connected.
SVRMGR> archive log list
Database log mode              No Archive Mode
Automatic archival             Disabled
Archive destination            /home/oracle/OraHome1/dbs/arch
Oldest online log sequence     1036115
Current log sequence           1036117
```

Al ejecutar esta orden dos días consecutivos a la misma hora, y calculando la diferencia entre las cantidades de current log sequence, se obtiene una aproximación de cuántos archivos redo log se han creado en 24 horas.

Si multiplicamos este número por el tamaño del archivo redo log, se obtiene una estimación de la cantidad de archivos redo log esenciales para una instalación del DBMS Oracle.

Otro método para calcular la cantidad de archivos redo log generados por una transacción particular, surge de la multiplicación de este valor por la tasa de transacciones (el número de transacciones realizadas en 24 horas), para tener una apreciación de la cantidad de archivos redo log óptima para la instalación en bytes. Los pasos que a continuación se establecen, están dirigidos a lograr el cálculo de la cantidad de archivos redo log por transacción.

1. Ejecutar el siguiente conjunto de sentencias SQL antes de operar la transacción. Esto marcará el valor inicial de los archivos redo log (obtenido desde la vista V$SYSSTAT) antes de ejecutar la transacción en el paso 2.

```
Connected to:
Oracle8i Enterprise Edition Release 8.1.6.1.0 - Production
With the Partitioning option
JServer Release 8.1.6.0.0 - Production

SQL> column name format a40;
SQL> column redo_i new_value redo;
SQL> set termout off;

SQL> select value redo_i from v$sysstat where statistic#=71;

  REDO_I
----------
     10577
```

2. Ejecutar la transacción. En este momento se asume que éste es el único procedimiento que se está operando en la base de datos.

3. Ejecutar la siguiente sentencia SQL, la cual obtiene la diferencia entre el valor final del archivo redo log y el valor inicial que resultó del paso 1. Este número suministra la cantidad de archivos redo log creados (en bytes) cumpliendo la transacción del paso 2.

```
SQL> update scott.emp set empno=7369 where empno=1234;
1 row updated.

SQL> commit;
Commit complete.

SQL>  select (value-&redo) redo from v$sysstat where statistic#=71;
old   1:  select (value-&redo) redo from v$sysstat where statistic#=71
new   1:  select (value-    10577) redo from v$sysstat where statistic#=71

     REDO
----------
        4
```

Para este ejemplo, se utilizó un update sobre la tabla SCOTT.EMP, apreciándose que el cambio fue mínimo ya que únicamente se actualizan diez registros, y el cambio está dado en 4 bytes.

3.1.2 Número de cambio del sistema (SCN).

El número de cambio del sistema (SCN *System Change Number* por sus siglas en Inglés), es una estructura de datos fundamental que define a la base de datos en un instante preciso de tiempo.

Cuando se realiza un commit en una transacción, un SCN le es asignado a dicha transacción, cuyo objeto es identificarla de manera particular. Los SCN proporcionan el mecanismo de reloj interno de Oracle y pueden ser vistos como relojes lógicos, no obstante, no deben confundirse con el reloj del sistema.

Los SCN funcionan como si se tomaran fotografías instantáneas a la base de datos, elementos que son vitales para los procesos de recuperación de la base de datos. Es pertinente puntualizar que Oracle efectúa la restauración exclusivamente con base en los SCN.

A manera de muestreo, si la transacción realiza un insert y un commit, Oracle le asignará un SCN a esa transacción, por ejemplo 44. La siguiente transacción que recibe un commit diez minutos después recibe el SCN 45. Si la transacción consecutiva recibiera un SCN mayor a los otros dos (v.gr. 48), significaría que Oracle realizó un trabajo interno al cual asignó los SCN subsecuentes (46 y 47). Este procedimiento garantiza que cada uno de los SCN sea único, se incrementan con el tiempo y podrían en determinadas ocasiones no ser secuenciales.

Cabe señalar que la secuencia de los SCN jamás se reinicia hasta cero, a menos que se vuelva a crear la base de datos.

Los SCN se utilizan en tablas de transacciones, cabeceras de bloques, archivos de control, cabeceras de archivos de datos y registros redo log.

- SCN bajo y SCN alto.

Cada archivo redo log posee un número secuencial a efecto de identificarlo de forma única.

Cuando un archivo redo log se satura de registros, se cierra y se abre uno nuevo. El archivo redo log se marca con un SCN bajo, que es uno mayor que el SCN alto del archivo redo log anterior, y el valor del SCN alto actual se establece a infinito ya que Oracle no sabe cuántos SCN se almacenarán en el archivo redo log actual. Esta información puede ser consultada por medio de la vista V$LOG_HISTORY:

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

RECID       STAMP   THREAD# SEQUENCE# FIRST_CHANGE# FIRST_TIM NEXT_CHANGE#
---------- ---------- ---------- ---------- ------------- --------- ------------
-
   1036112 481400981       1    1036112    1352911064    23-DEC-02 1352911993
   1036113 481401240       1    1036113    1352911993    23-DEC-02 1352912885
   1036114 481401747       1    1036114    1352912885    23-DEC-02 1352913776
   1036115 481402769       1    1036115    1352913776    23-DEC-02 1352914705
   1036116 481404579       1    1036116    1352914705    23-DEC-02 1352915594
   1036117 481642187       1    1036117    1352915594    23-DEC-02 1352916462
   1036118 481643524       1    1036118    1352916462    26-DEC-02 1352917375
   1036119 481647106       1    1036119    1352917375    26-DEC-02 1352918273

65535 rows selected.
```

En este ejemplo se ha utilizado una base de datos de producción, el número 1036112 de secuencia de log corresponde al thread 1. El SCN más bajo obviamente fue 1 y el más alto es el último 1352917375. Marca 65535 registros en razón a que la numeración no es continua debido al funcionamiento interno de Oracle.

Si cualquiera de los cambios aludidos fuera necesario en el futuro para efectuar una recuperación, Oracle solicitaría que este archivo en particular sea aplicado para recuperar hacia adelante el respaldo de la base de datos.

- SCN normal offline.

Un SCN de tipo normal offline se refiere a un SCN que se mantiene en la tabla del diccionario de datos **ts$** para cada tablespace que se pone offline en la opción normal. Cuando esto acontece, un checkpoint se realiza en todos los archivos de datos pertenecientes al tablespace; en este instante es asignado el SCN offline normal. Éste es usado por Oracle para regresar al tablespace al estado online.

- Stop SCN.

En el archivo de control se registra un stop SCN relativo a cada archivo de datos. Cuando un archivo de datos está online y cualquier instancia tiene abierta la base de datos, el stop SCN detiene el archivo de datos correspondiente. Cuando se pone offline un tablespace, el stop SCN es registrado en el archivo de control para cada archivo de datos perteneciente al tablespace. Esto significa que no se generará redo para el archivo de datos una vez que se haya asignado el stop SCN.

El stop SCN se utiliza durante la recuperación del medio para asegurar que ésta finalizará cuando se alcance un valor de SCN igual al stop SCN del archivo de datos en el caso de estar recuperando un archivo de datos offline.

3.1.3 Threads de archivos redo logs.

Un archivo redo log contiene las modificaciones realizadas a la base de datos. Los registros de redo creados para reformar la información se almacenan en estos archivos de log.

Los archivos redo log son esenciales para la operación normal de la base de datos. Cada instancia de una base de datos Oracle posee, al menos dos grupos de archivos redo log; un grupo de redo log contiene uno o más archivos de log online, (Denominados miembros) que son idénticos y residen en unidades de disco diferente.

Un conjunto de archivos redo log se denomina "thread de archivos redo log". Cada instancia registra cambios en su propio conjunto de archivos redo log online o en su propio thread de redo. Si se tiene una base de datos de instancia única, Oracle crea el primer thread de archivos de redo log cuando se crea la base de datos.

Cada thread está identificado de forma única por un número de thread. Después de crear un thread de archivos redo log, se debe activar el thread utilizando la opción PUBLIC o PRIVATE. La opción PUBLIC indica que cualquier instancia puede hacer uso de este thread. Si se utiliza la opción PRIVATE, significa que un parámetro THREAD = "n", deberá ser incluido específicamente en el archivo INIT.ORA, donde "n" es el número de thread. El cual será asignado a una instancia en particular

Si hay múltiples threads disponibles en la base de datos, uno de ellos debe ser especificado al momento del montaje.

3.1.4 Conmutación de archivos redo log.

La conmutación de archivos redo log es el proceso por el cual el LGWR deja de escribir en el archivo actual y cambia al siguiente archivo de log online disponible. Cuando Oracle crea el redo, utiliza el buffer de redo log en memoria y los archivos redo log en disco. Se escribe en disco el buffer de redo log a los archivos redo log y el buffer del redo log se reutiliza para almacenar más redo.

El mismo principio aplica para los archivos redo log en disco. A medida que los archivos de log en disco se llenan, Oracle conmuta el siguiente archivo de log disponible mientras que el proceso de ARCH guarda el archivo lleno.

Un número de secuencia identifica a cada archivo log, Oracle necesita un mínimo de dos archivos de log en disco y un buffer de redo en memoria. El proceso LGWR escribe a un único archivo de log en disco en cada momento, pero el buffer de redo puede escrito por varios procesos de manera recurrente.

Una conmutación de archivos redo log puede ser ocasionada por una de las siguientes condiciones:

- ✓ Los procesos en primer plano ya no son capaces de asignar espacio en el buffer de redo log.
- ✓ El DBA ejecuta la orden **alter system switch logfile.**

3.1.5 Checkpoints.

Un checkpoint es un evento en la base de datos que escribe la información modificada de la memoria caché al disco duro, actualizando el archivo de control y los archivos de datos[21].

Después de un checkpoint, el redo de los archivos redo log deja de ser útil para una recuperación. Si el tamaño en disco del archivo redo log fuera ilimitado y no tuviera importancia el tiempo de recuperación, es probable que el proceso checkpoint no fuera necesario, ya que lo procedente sería aplicar los cambios a la base de datos desde los archivos redo log que estuvieran almacenados. Pero debido a la naturaleza circular del archivo redo log, es esencial asegurar su copia a un log archivado antes de asignar espacio en el archivo redo log y escribir el redo.

Previo a que se realice una modificación a un bloque de datos, el redo para tal cambio se ha introducido en el buffer de redo log, y antes de que el bloque de datos se escriba en el archivo de datos en disco, su redo ha sido escrito en el archivo redo log.

- Eventos que generan un checkpoint.

Los checkpoint se ocasionan automáticamente cuando ocurre un evento durante la operación normal de la base de datos. Pero pueden ser accionados normalmente ejecutando una orden desde el comando SVRMGR de Oracle:

[21] HOECHST, Tim *ob.cit.*

```
billing2:bd_prue> svrmgrl

Oracle Server Manager Release 2.3.3.0.0 - Production
Copyright (c) Oracle Corporation 1994, 1995. All rights reserved.

Oracle7 Server Release 7.3.3.0.0 - Production Release
With the distributed option
PL/SQL Release 2.3.3.0.0 - Production

SVRMGR> connect internal
Connected.
SVRMGR> alter system checkpoint local;
Statement processed.
```

Esta orden ejecutará explícitamente un checkpoint a una instancia desde la cual que se activa. Existen tres tipos de checkpoints.

- ✓ Checkpoint local (de thread).

En la especie, una instancia particular produce un checkpoint en todos los archivos de datos. Dicho en otras palabras, todos los buffers con datos en una instancia en específico son trasladados a los archivos de datos de la base de datos; por ejemplo, la sentencia **alter system checkpoint local** ejecuta un checkpoint local.

- ✓ Checkpoint global.

En la especie, todas las instancias ejecutan un checkpoint en todos los archivos de datos de la base de datos; v.gr. la sentencia **alter system checkpoint global** lo hará en toda la base de datos.

✓ Checkpoint de archivo.

Bajo este apartado, todas las instancias ejecutan un checkpoint en un subconjunto de los archivos de datos; a manera de ejemplo, la orden **alter tablespace SYSTEM begin backup** ejecuta un checkpoint global en todos los archivos de datos pertenecientes al tablespace system.

Los checkpoints locales son específicos de instancia y delimitados al thread local. Un checkpoint global puede realizarse en respuesta a una orden SQL o cuando se realiza un checkpoint de base de datos. Los checkpoints locales pueden iniciarse debido a la conmutación de archivo log, ejecución de una orden SQL o cuando alcanza el intervalo de checkpoint especificado por el parámetro LOG_CHECKPOINT_INTERVAL de INIT.ORA.

Los checkpoints locales y globales se realizan siempre en todos los archivos de datos de la base de datos.

Un checkpoint de archivo se realiza siempre en respuesta a una orden SQL, por ejemplo, operaciones de base de datos, entre otras la realización de un respaldo en caliente o poner offline un tablespace, requieren que se efectúe un checkpoint en todas las instancias, pero sólo en un tablespace específico.

Los checkpoints son parte integral del funcionamiento normal de la base de datos. Su frecuencia es controlable, pero no se debe soslayar que su ejecución puede desembocar en una operación de alto consumo de E/S, lo cual debe ser cuidadosamente ajustado.

Como los checkpoints pueden ser disparados por usuarios o bien por eventos de base de datos, además de que su procesamiento se realiza usualmente bajo actividad normal en la base de datos, podría suceder que haya múltiples checkpoints disparados de manera solapada. Para evitarlo, cada tipo de checkpoint conlleva un privilegio de *override o ignored* en el momento de su activación.

Cuando se dispara un checkpoint con override, el proceso anterior es reemplazado por el actual. Esto viene a significar que carece de relevancia el punto del procedimiento en que se encuentre el checkpoint anterior, Oracle iniciará otro ignorando al primero. El override sólo se puede realizar en checkpoints locales y globales.

En la tabla que se muestra a continuación, se describen los eventos de la base de datos, las sentencias SQL y los parámetros del archivo INIT.ORA que producen checkpoints:

Checkpoints generados por eventos en primer y segundo planos.	***Rápido / Lento.***	***Override.***	***G/L/A***
Alter system switch logfile	Lento	SI	L
Alter system checkpoint (Local o Global)	Rápido	SI	G/L
Alter tablespace <nombre> begin backup	Rápido	NA	A
Alter tablespace offline (normal, temporary)	Rápido	NA	A
Instance shutdown (normal or immediate)	Rápido	SI	L
Log file switch normal	Lento	SI	L
Log file switch stuck	Rápido	NA	L
Parámetro de INIT.ORA LOG_CHECKPOINT_TIMEOUT	Lento	NO	L
Parámetro de INIT.ORA LOG_CHECKPOINT_INTERVAL	Lento	NO	L

Los checkpoints global, local y de archivo se denotan por G= global, L= local y A= Archivo, N/A significa no aplicable.

En el caso de un checkpoint global, el trabajo realizado para procesar el checkpoint conlleva los pasos que se enlistan a continuación:

1. *Obtener o mantener la secuencia de estado de la instancia. El instance state enqueue* se adquiere durante transiciones de estado de la instancia. Oracle adquiere este encolamiento para asegurar que la base de datos se mantenga abierta mientras el proceso de checkpoint tenga verificativo.

2. *Capturar la información de checkpoint actual.* Por esta fase se crea una estructura cuya función es registrar información, incluyendo la hora del checkpoint actual, los threads activos en este momento, el thread actual que realiza el checkpoint y (lo más importante) la dirección en el archivo redo log, que será el punto de ruptura para recuperación.

3. *Identificar los buffers sucios.* Se hace consistir en la identificación de los buffers sucios. Esto se realiza mediante la exploración de los buffers en caché cíclicamente hasta que se haya detectado la totalidad de los buffers sucios. Si Oracle encuentra un buffer sucio dentro del rango de archivos sobre los que se efectúa el checkpoint, la cabecera del buffer será marcada *para ser direccionado a disco.* Durante este proceso, Oracle omite los buffers de segmento temporales y los no modificados (de sólo lectura), dado que no se genera redo para ellos. Una vez identificados los buffers sucios, se ordena al proceso DBWR que realice la escritura.

4. *Escribir los buffers sucios.* Este paso vuelve a disco todos los buffers sucios utilizando el proceso DBWR. Una vez que el DBWR escribe todos los buffers, establece un indicador de que ha terminado de salvar los buffers a disco. El proceso LGWR (o CKPT) comprueba continuamente hasta que reconoce que el proceso DBWR se ha completado.

5. *Actualizar los archivos de datos y los archivos de control.* Finalmente las cabeceras de archivos de datos y archivo de control son actualizados con la información registrada en el paso 2. El archivo de control contiene una estructura de checkpoint para cada thread activado. Cada cabecera de archivo de datos comprende también una estructura de checkpoint. La información en estas estructuras se actualiza como parte integrante de este paso.

En ambos casos, la información de checkpoint (capturada en el paso 2) no se actualiza en la cabecera del archivo. El primer caso se efectúa cuando el archivo de datos se encuentra en modo de respaldo en caliente. En esta situación, Oracle no sabe cuándo leerá la cabecera del archivo del respaldo del Sistema Operativo, y la copia de backup debe tener el checkpoint SCN cuando comience la copia. El segundo caso se da si el checkpoint SCN ya existe en el disco. Esto significa que los cambios realizados por el checkpoint ya existen en el disco y puede ocurrir si un checkpoint rápido correspondiente a un respaldo en caliente actualiza la cabecera del archivo cuando un checkpoint global está en proceso. Se reitera, Oracle captura el checkpoint SCN antes de que el proceso propiamente dicho comience, por lo que es posible que una orden como begin backup (que realiza un checkpoint rápido de tablespace) pueda anticiparse.

Oracle comprueba la consistencia de las cabeceras de los archivos de datos antes de actualizarlos. Una vez verificadas, las cabeceras de archivos de datos se actualizan para reflejar el checkpoint actual. Los archivos no confirmados y los archivos que causan errores durante la escritura de actualización, son ignorados. Un archivo necesitará una recuperación del medio si se sobreescriben los archivos de log, y, en este caso, el DBWR pone offline los archivos de datos.

Poner offline un archivo de datos es realizado siempre por el proceso DBWR. No se puede poner offline un archivo de datos si la base de datos está funcionando en modo NOARCHIVELOG o si el archivo de datos pertenece al tablespace SYSTEM. Si Oracle puede escribir todos los bloques sucios o si no es preciso escribir nada (porque los bloques de datos están en el futuro del redo, por lo que todos los cambios ya existen en el archivo de datos en disco), entonces no se ha hecho ningún daño.

Oracle mantiene un contador de checkpoints en las cabeceras de los archivos de datos. Se utiliza para verificar que se está utilizando la versión actual del archivo de datos durante la operación normal y para evitar la restauración de una versión errónea de un archivo de datos durante la recuperación. Este contador se incrementa incluso si los archivos de datos están en modo de backup en caliente. El contador de checkpoint para cada archivo de datos también se mantiene en el archivo de control para la entrada correspondiente del archivo de datos.

- Procesamiento de checkpoint en Oracle.

El algoritmo de checkpoint ha sido modificado en versiones del DBMS Oracle. A continuación se ofrece una breve descripción.

Los buffers sucios en la caché se concatenan en una nueva cola, denominada la cola de checkpoint. Cada cambio al buffer tiene un valor de redo asociado. La cola de checkpoint contiene buffers sucios registrados en orden de acuerdo a la posición en el archivo de log; esto es, los buffers en la cola de checkpoint se ordenan de acuerdo a su valor de redo inferior.

Hay que observar que dado que el buffer se enlaza en la cola en el orden en que haya sido utilizado, éste no se mueve si se realizan modificaciones adicionales al buffer antes de que sea escrito. En otras palabras, una vez que un buffer es enlazado en la cola de checkpoint, permanece en el mismo lugar hasta que se escribe.

El proceso DBWR escribe los buffers de la cola en orden ascendente de redo inferior, en respuesta a una solicitud de checkpoint. Cada solicitud de checkpoint individualiza un valor de redo. Una vez que DBWR escribe los buffers cuyo valor de redo es igual o mayor al valor de redo del checkpoint, se declara completo el checkpoint y registrado en el archivo de control y cabeceras de archivo.

Puesto que los buffers en la cola de checkpoint están ordenados por valor de redo inferior y DBWR escribe los buffers de checkpoint en orden de menor valor de redo, puede darse el caso de tener múltiples solicitudes activas de checkpoint. A medida que DBWR escribe los buffers, comprueba el valor de redo de aquéllos en la cabecera de la cola con el valor de redo del checkpoint. Todas las solicitudes de checkpoint cuyo valor de redo es inferior al valor de redo inferior del buffer a la cabecera de la cola de checkpoint pueden ser declarados corno completos. DBWR continúa escribiendo lotes de buffers de checkpoint mientras existan solicitudes activas de checkpoint pendientes.

El nuevo algoritmo es una mejora sobre el anterior en varios aspectos:

- ✓ DBWR siempre sabe exactamente qué buffers necesitan ser escritos para satisfacer una petición de checkpoint.
- ✓ Cada escritura de checkpoint asegura que se trata de un progreso hacia la finalización del primer checkpoint (el que tiene el menor valor de redo).

- ✓ Es posible distinguir entre múltiples solicitudes de checkpoint basándose en el valor de redo de checkpoint y completarlas en dicho orden.

➢ Checkpoint rápido y checkpoint lento.

La velocidad a la que se realiza el checkpoint es determinada realmente por el proceso DBWR y no por los procesos LGWR o CKPT, como podría pensarse[22]. El proceso LGWR (o CKPT) simplemente transmite al proceso DBWR cómo debe manejar las escrituras de los buffers marcados para ser escritos por el checkpoint. Una vez que el proceso DBWR es informado, comienza a explorar todas las cabeceras de los buffers en busca de buffers sucios que necesitan ser direccionados a disco. Una vez acabada la exploración, todos los buffers que se han leído en modo *lectura consistente* (esto es, bloques leídos en memoria con la sentencia SELECT) son ignorados junto con los buffers de segmento temporal, ya que no se ha generado redo para ellos. A continuación se explora el resto de los buffers, y si se encuentra un buffer sucio, será reservado para ser escrito. Si Oracle está haciendo un checkpoint *lento,* el proceso DBWR detiene el procesamiento de dicho checkpoint en el caso de que se surta alguna de las siguientes condiciones:

- ✓ Si se alcanza el tamaño del parámetro de **db_checkpoint_write_batch** (número de buffers).
- ✓ Cuando se han explorado más de 1.000 buffers y no se ha encontrado ningún buffer sucio que escribir a disco.

[22] KROHN, Mike. "*Using the oracle toolset*". Wokingham, England: Addison-Wesley, 2003.

La idea es liberar CPU, que de otra forma se estaría utilizando, afectando la respuesta en primer plano. Además, si se establece un valor muy alto para **db_check_point_write_batch,** la E/S perjudicará al primer plano. Sin embargo, si Oracle está haciendo un *checkpoint rápido,* el DBWR simplemente continúa explorando todos los buffers en la memoria caché. En este caso, se evitan aspectos como la sobrecarga del manejo y paso de mensajes o posibles cambios de contexto. Una vez iniciado, el proceso DBWR no tendrá otra función hasta que todos los buffers sucios hayan sido escritos en disco como parte del proceso de checkpoint rápido.

➢ Checkpoint de thread.

Se le denomina *checkpoint de thread* al procedimiento de un checkpoint de instancia. Cada thread ejecutará checkpoints independientemente de otros threads y a su vez se actualizará la información de checkpoint del archivo de control.

En el archivo de control se mantiene una *estructura de checkpoint* para cada thread. Esto significa que sólo está asegurada la escritura a disco de los buffers sucios de la instancia que está ejecutando el checkpoint. Oracle garantiza que todo el redo generado en este thread antes del checkpoint SCN ha sido aplicado a los archivos de datos activos, y que se han escrito los bloques a los archivos de datos en disco. Entre otras cosas, la estructura de checkpoint contiene la siguiente información:

- ✓ *El current SCN,* en el cual se verificó el checkpoint.
- ✓ El thread que hizo el checkpoint.
- ✓ La información de fecha (timestamp) en la que se registró el current SCN.

Cuando tiene lugar un checkpoint, Oracle registra el valor de SCN y la información de fecha de ese instante en el archivo de control. Oracle garantiza que todos los cambios realizados sobre la base de datos antes de este checkpoint SCN están en disco. Lo anterior comprueba que en el caso de una caída de la base de datos, la recuperación de fallos aplicará sólo los cambios a partir de ese SCN.

- Checkpoint de base de datos.

Cuando una base de datos tiene múltiples threads, existe una estructura de checkpoint para cada thread en el archivo de control. Una de estas estructuras se escribe también en las cabeceras del archivo de datos y se define como *estructura de checkpoint de base de datos o información de checkpoint de base de datos*. La estructura de checkpoint de thread que es seleccionada como estructura de checkpoint de base de datos es la que tiene el mínimo checkpoint SCN. Por ejemplo, si hay tres threads abiertos con valores de checkpoint SCN de thread de 300, 350 y 400, el checkpoint SCN de base de datos será igual a 300, puesto que éste es el mínimo valor de todos los checkpoint SCN de thread.

Oracle garantiza que todos los cambios que poseen un valor de SCN menor que el checkpoint SCN de base de datos han sido escritos a los archivos de base de datos en disco. En el caso de una base de datos de instancia única, el checkpoint de thread en el archivo de control es el mismo que el checkpoint de base de datos en los archivos de datos. Si no hay threads abiertos, el checkpoint de la base de datos será el más alto de todos los threads activados, ya que todos los cambios anteriores al checkpoint de base de datos están escritos en los archivos de datos online. El checkpoint de base de datos se emplea para actualizar las cabeceras de archivo cuando una instancia realiza un checkpoint de su thread.

➢ Checkpoint de archivo de datos.

Cada cabecera de archivo de datos contiene la información acerca del checkpoint. El SCN correspondiente al checkpoint asegura que todos los cambios previos a ese SCN están en disco. La información de checkpoint en todos los archivos de datos online se actualiza al momento de ejecutarse un checkpoint de archivo o un checkpoint global. La única excepción es cuando se encuentra en progreso un respaldo en caliente. Por ejemplo, si el checkpoint SCN de un archivo de datos es 500, entonces, cuando el archivo de datos se pasa a modo de backup en caliente, este valor no cambia hasta que se ejecute una orden **end hot backup.**

Puesto que no se actualiza el valor de SCN, se garantiza que el archivo de datos de backup tiene el mismo valor de checkpoint SCN de 500. Por tanto, si alguna vez se restaura este archivo de datos para hacer una recuperación del medio, ese proceso comienza desde el SCN 500. Como se vio anteriormente, el valor de checkpoint SCN se almacena también en el archivo de control para cada archivo de datos.

3.1.6 Histórico del log.

Se puede configurar el archivo de control para que contenga los registros históricos de cada archivo redo log que es utilizado por la base de datos. Cada registro en esta tabla proporciona información sobre un archivo redo log. A su vez, cada registro histórico contiene, entre otras cosas, el número de thread, número de secuencia de log, low SCN y high SCN. Esta información se puede obtener consultando la vista V$LOGHISTORY. El parámetro MAXLOGHISTORY se puede utilizar durante la creación de una base de datos para indicar cuánto histórico se desea almacenar en el archivo de control.

El objetivo de mantener esta información es reconstruir los nombres de los archivos de log almacenado a partir de los números de SCN y de thread. Puesto que el número de secuencia de log es parte de la información del checkpoint. En las bases de datos abiertas con instancia única no es precisa esta tabla de histórico de log para reconstruir los nombres de archivo de log durante la recuperación.

3.2 Métodos de recuperación.

Esta sección expone los métodos de recuperación utilizados por Oracle y en varias opciones disponibles para el DBA. Existen tres tipos básicos de recuperación. Re*cuperación de bloque online, recuperación de thread y recuperación del medio.* En los tres casos, el algoritmo que aplica los registros de redo contra un bloque individual es el mismo. Pero antes de abordar su estudio, es preciso comprender los conceptos de los mecanismos *de aplicación de redo, recuperación hacia adelante (roll forward) y recuperación hacia atrás (rollback),* y cómo determina Oracle la necesidad de hacer una recuperación de uno o varios archivos de datos.

3.2.1 Aplicación de redo.

Cuando se arranca una base de datos con la orden startup desde el SVRMGR, aquélla pasa a través de varios estados. En primer término, la base de datos se encuentra en estado *sin montar.* Ahí, Oracle lee el archivo INIT.ORA para determinar el tamaño de la SGA, posteriormente crea la SGA y arranca los procesos de background. En este momento, el DBA observa un mensaje en el terminal que dice «instance started».

A continuación, la instancia *monta* la base de datos. En este estado, se abre el archivo de control y aparece el mensaje de «database mounted». En esta fase, se pueden ejecutar órdenes como **recover database** o cualquier orden **alter database.** Asimismo se puede ejecutar la orden **alter session** para escribir información de trazas del archivo de control, cabeceras del archivo redo log, cabeceras de archivo de datos y bloques de datos a archivos de trazas.

En el tercer y último estado, la instancia *abre (open)* la base de datos mostrando en pantalla el mensaje «database opened» al usuario. Si la instancia está abriendo la base de datos por primera vez después de una caída, se requiere efectuar una recuperación de fallos. Hay dos pasos para una recuperación de fallos. El *primero es recuperar hacia adelante* la base de datos, donde todo el redo almacenado en los archivos redo log se aplicará a los archivos de base de datos y se abrirá un nuevo thread. Como parte del segundo paso (conocido como *recuperación de transacción)* se deshacen todas las transacciones no confirmadas.

Una pregunta frecuente es ¿cómo sabe Oracle cuándo debe aplicar recuperación a un archivo o archivos en particular? Se ha visto que cada archivo de datos, en su cabecera, posee un contador de checkpoint que se incrementó cada vez que Oracle efectúa un checkpoint en el archivo de datos. El archivo de control también tiene un contador de checkpoint para cada archivo de datos. También se ha visto que cada cabecera de archivo de datos contiene un SCN como parte de su estructura de checkpoint, que se denomina *start SCV.* Correspondiendo con cada archivo de datos, el archivo de control posee un *stop SCN.* Durante la operación normal de la base de datos, el stop SCN en el archivo de control toma el valor infinito. El start SCN en el archivo de datos que se incrementa cada vez que se realiza un checkpoint.

Cuando se detiene una base de datos con las opciones NORMAL o IMMEDIATE, el checkpoint que se ejecuta establecerá el valor del stop SCN igual al correspondiente start SCN en la cabecera de cada archivo de datos. En esa circunstancia, la próxima vez que se abra la base de datos, Oracle hará dos comprobaciones: La primera tiene el objetivo de verificar si el contador de checkpoint en la cabecera del archivo de datos coincide con su correspondiente contador de checkpoint en el archivo de control. Una vez comprobado lo anterior, se efectúa la segunda comprobación, ésta compara el valor de start SCN en la cabecera del archivo de datos con su correspondiente stop SCN en el archivo de control. Si el stop SCN es igual al start SCN, entonces no es necesaria la recuperación para dicho archivo. Esta confrontación se realiza para archivo de datos y después se abre la base de datos. Como parte de la apertura, se vuelve a poner el stop SCN a infinito.

En el caso de que la base de datos se pare utilizando la orden **shutdown abort** no se realiza un checkpoint y el stop SCN se queda con valor infinito cuando la base de datos se detiene. Durante el siguiente arranque, se vuelven a comprobar en primer lugar los contadores. Si éstos son iguales (no se reemplazaron los archivos de datos con una copia de respaldo), Oracle procederá a comparar el start SCN y el stop SCN. En dicha circunstancia, dado que el stop SCN puede valer infinito y el start SCN tiene algún valor, Oracle determinará que no son iguales y por ende, será preciso realizar una recuperación de thread. En este caso, se ejecutará una recuperación de fallos dado que se está iniciando la instancia después de una caída. Como parte de la recuperación de fallos, Oracle lee los archivos de log online y aplica los cambios a la base de datos como parte de la *recuperación hacia adelante, y* lee la tabla de transacción del segmento de rollback para llevar a cabo una recuperación de transacción (recuperación hacia atrás).

Después de parar la base de datos, si se reemplaza uno de los archivos de datos con una copia de respaldo, Oracle lo determinará como parte de la comprobación de contador de checkpoint y preguntará la conveniencia de aplicar una recuperación del medio. De la cabecera del archivo de datos, Oracle también conoce el número de secuencia de log del archivo redo log archivado donde comienza la recuperación. Oracle solicita que se aplique una recuperación del medio a partir de dicho número de secuencia.

- Recuperación hacia adelante (roll forward).

Cualquier tipo de recuperación (de thread o de medio) se realiza en dos partes. La primera es la recuperación hacia adelante. Dicha recuperación implica la aplicación secuencial de los registros de redo a los bloques de datos correspondientes. Oracle aplicará todos o ninguno de los cambios de un registro de redo atómico.

Este proceso se realiza de la siguiente manera: En primer lugar, se abre el archivo de log para cada thread que estuviera activo en el momento en que se asignó el SCN. Si el archivo de log está online (como en el caso de una recuperación de fallos), entonces se abre automáticamente. Si el log es un archivo de log archivado, entonces solicitará que el usuario introduzca el nombre del archivo de log.

El orden de aplicación de los registros de redo sin SCN no es preciso, sin embargo es suficiente para que la recuperación hacia atrás permita la consistencia de la base de datos. Si se requiriera el siguiente archivo de log en un thread, se utiliza una copia online disponible. En caso negativo, se escriben a disco los buffers de recuperación sucios, y se incrementan los checkpoint de los archivos de datos de tal suerte que no sea necesario volver a aplicar el redo (esto se conoce como un checkpoint de redo). Entonces se solicita el siguiente archivo de log.

Cabe subrayar que la aplicación de redo, en ocasiones puede necesitar salvaguardar y volver a aplicar el redo que se saltó anteriormente.

Un número de versión es asignado a cada bloque de datos. Cada modificación realizada al bloque de datos se registra en el archivo de log como un vector de cambio. Éste tendrá un número de versión determinado en una unidad mayor a la del bloque. Cuando se realiza la recuperación, por ejemplo, el cambio 11, debe aplicarse al bloque que tenga el número de versión 10. Después de aplicar el cambio, se incrementará en 1 y valdrá 11. Posteriormente se debe aplicar el cambio número 12 a dicho bloque, y así sucesivamente.

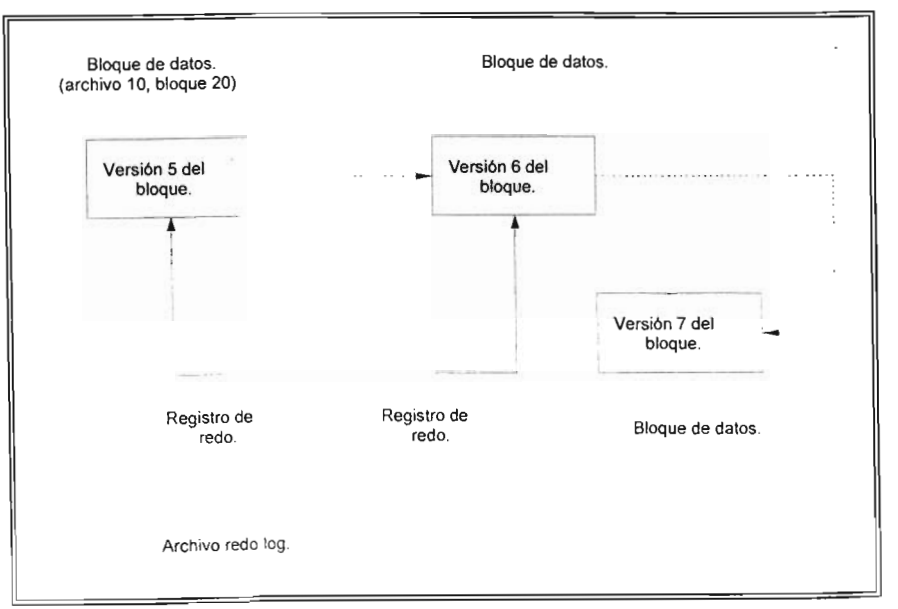


Figura 3.1 Recuperación hacia adelante de un bloque de datos.

La Figura 3.1 que antecede, muestra que los cambios 6 y 7 del archivo redo log se han aplicado a un bloque de datos, recuperándolo hacia adelante. En esta ilustración, se puede observar que en el archivo redo log existen dos registros de redo que pertenecen al número de bloque 20 del archivo 10. Si se considera que al inicio de la recuperación el bloque de datos en disco tiene una versión de 5. y, por tanto, se debe aplicar el cambio número 6 del archivo redo log, por ello, el primer registro de redo se aplicará al bloque 20. Como parte de la recuperación hacia adelante, una vez que se ha aplicado el cambio al bloque, se incrementó a 6 el número de versión del bloque. Ahora, el segundo registro de redo (correspondiente al archivo 10, bloque 20) en el archivo redo log tiene el cambio número 7, que debe ser aplicado al bloque de datos con número de versión 6. Esto modificará el número de versión del bloque a 7.

En esa situación, si existe otro registro de redo para este bloque de datos que tiene un número de cambio de 9, este registro de redo no puede aplicarse al bloque de datos, dado que su número de versión es 7. Ello significa que debe aplicarse el cambio 8 antes de poder aplicar el 9. Dicho en otras palabras, se deben aplicar secuencialmente todos los cambios al bloque. No obstante, en algún momento, cuando se debe aplicar el cambio 15 a un bloque y la versión de dicho bloque es, por ejemplo, 19, significa que el bloque se encuentra por delante del cambio de redo (esto es, el cambio ya se encuentra en el bloque de datos). En este caso, se salta el redo y se lee el siguiente registro de redo en el archivo redo log. Un bloque de datos puede estar en el futuro del redo, es decir, si se ha realizado una recuperación de base de datos, pero sólo se ha restaurado un archivo de datos y otros archivos son los actuales. Entonces se aplica redo sólo al fichero que realmente necesita la recuperación. Sin embargo, en razón a que durante la recuperación todo el redo es examinado, también se intentará aplicar redo a los archivos que no lo necesitan. Esto no requiere que Oracle lea el bloque de datos, pero comprueba el redo SCN con el checkpoint SCN de la cabecera del archivo.

Si el checkpoint SCN está por delante del redo SCN, entonces Oracle determina que el bloque de datos se encuentra en el redo futuro (lo que significa que el bloque ya tiene incorporado el cambio). Por tanto, se salta el registro de redo y se examina el siguiente registro del archivo redo log.

- Recuperación hacia atrás (roll back).

Una vez que se ha aplicado todo el redo (recuperación hacia adelante), la segunda parte del proceso de restauración es la recuperación hacia atrás. Este proceso es conocido como recuperación de transacción. Los segmentos de rollback son el mecanismo por el que Oracle deshace las transacciones no confirmadas. Puesto que los segmentos de rollback residen en archivos de datos y están protegidos por el mecanismo de redo, se debe aplicar todo el redo antes de que se pueda deshacer cualquier transacción.

Oracle encuentra las tablas de transacción consultando la tabla base del diccionario, undo$. Explora las tablas de transacción de los segmentos de rollback de las transacciones activas. Para cada transacción no confirmada, Oracle recorre la tabla undo$ y deshace todos los cambios. Es razonable ver que se genera undo, y, por tanto, se producirá el guardado de los archivos de log, en el caso de que se deshagan muchas o muy grandes transacciones. Esto se debe a que deshacer transacciones origina cambios en bloques de archivos de datos, lo cual repercute en la generación de redo.

El parámetro ROLLBACKSEGMENTS de archivo INIT.ORA no tiene efecto aquí. Se buscan todas las transacciones en todos los segmentos de rollback y se deshacen las transacciones no confirmadas. Una vez concluido, todos los segmentos de rollback adquiridos por la instancia estarán ONLINE, y todos los demás OFFLINE.

Los segmentos de rollback que contengan transacciones muertas que no pueden limpiarse se marcan como NEEDS RECOVERY. El segmento de rollback SYSTEM está siempre ONLINE para que funcione la base de datos y no debería incluirse en el parámetro ROLLBACK_SEGMENTS del archivo INIT.ORA. En este momento, la recuperación de transacción está completa y los usuarios pueden conectarse.

El segmento de rollback SYSTEM es único y especial, y ello conlleva repercusiones para la restitución. El undo generado por todas las transacciones que involucran undo$ (la tabla base del diccionario cuyo propietario es SYS) utiliza el segmento de rollback SYSTEM. Esto significa que las corrupciones del segmento de rollback SYSTEM son un problema muy serio.

- Recuperación de bloque de datos.

Oracle ejecuta automáticamente la recuperación al nivel de bloque mientras se lleva a cabo la operación normal de la base de datos y es transparente para el usuario. Cuando un proceso muere mientras cambia un buffer, Oracle reconstruye el buffer utilizando archivos redo log online para el thread actual y lo escribe en disco. La cabecera del buffer contiene información acerca del rango de registros de redo que le deben ser aplicados.

Cuando Oracle detecta un bloque corrupto en la caché, intenta eliminarlo del disco para su posterior recuperación mediante los archivos de log online. Comienza con el archivo de log online que contiene registros de redo de los que aún no se ha hecho checkpoint contra el archivo de datos que contiene el bloque. Esto es debido a que ningún buffer debería necesitar recuperación de un momento anterior al último checkpoint. Se exploran los archivos redo log en orden y se aplican los registros de redo del bloque. La recuperación se detiene al final del archivo redo log con el número de versión actual al momento en que comenzó la recuperación de bloque.

Si ocurre un error durante la recuperación, se marca el bloque como corrupto y se muestra un *error de corrupción de bloque*.

Si el proceso PMON está ejecutando la recuperación de bloque, Oracle no le permite emplear mucho tiempo en el recobro de un buffer. PMON progresa algo en la realización de la recuperación y entonces comprueba si hay algo más qué limpiar (como procesos finalizados anormalmente o transacciones de rollback). Para comprobar la cantidad de recuperación efectuada por PMON, Oracle limita la cantidad de redo que se aplica en una llamada a la recuperación de bloque. El máximo número de bloques de redo a aplicar en una llamada a la recuperación de bloque por PMON es una constante específica de puerto y los usuarios no tienen control sobre ella.

La recuperación al nivel de bloque es una operación normal ejecutada de manera automática por Oracle durante la operación normal de la base de datos y no implica ninguna acción por parte del DBA.

- Recuperación del medio.

Mientras que la recuperación de bloque se realiza automáticamente por la base de datos, la restauración del medio se produce en respuesta a una orden lanzada por el DBA[23]. Su función es restaurar cambios que se hayan perdido debido a que un archivo de datos pasó a estado offline sin hacer checkpoint. Por ejemplo, si se pone offline un tablespace mediante la opción IMMEDIATE, los archivos de datos pasarán a modo offline sin que Oracle haya ejecutado un checkpoint. La recuperación del medio puede aplicar tanto archivos de log online como archivado.

[23] LOCKMAN, David. *"Developing Personal Oracle8i applications"* Indianapolis, Indiana: Sams, c2003.

- Cuándo realizar una recuperación del medio.

Un backup de archivo de datos restaurado siempre necesita una recuperación del medio, incluso si ésta se puede llevar a cabo con los archivos de log online. Lo propio se hará para un archivo de datos que se puso offline sin un checkpoint previo. La base de datos no se puede abrirse si alguno de los archivos de datos online requiere de una recuperación del medio. En función del tipo de fallo ocurrido y del procedimiento de recuperación que se desea emplear, es posible la recuperación de la base de datos mientras una parte de la misma está abierta; pero si la base de datos está abierta, el archivo a recuperar debe estar offline.

- Operación de la recuperación del medio: recuperación de base de datos, tablespace y archivo de datos.

Cuando el contador de checkpoint en la cabecera del archivo de datos no concuerda con el relativo al archivo de control, Oracle lo detecta advirtiendo que es necesaria una recuperación del medio. Cuando se ejecuta la orden **recover** desde el SRVMGR, la recuperación comienza en el mínimo SCN de los archivos de datos que se están recuperando. Esto significa que Oracle comprueba el valor de SCN en la cabecera del archivo para todos los archivos de datos y elige el que tenga el valor de SCN más antiguo. Oracle da inicio a la aplicación de una recuperación del medio a este archivo empezando con este thread. El checkpoint SCN de cada archivo se salva para evitar aplicar redo anterior al checkpoint. Lo mismo acontece con el stop SCN más alto (registrado en el archivo de control) para saber si la recuperación debería detenerse antes de haber aplicado todo el redo.

Existen tres opciones que se pueden emplear para que la recuperación del medio tenga verificativo. La primera se denomina *recuperación de base de datos.* Esto viene a significar la posibilidad de restaurar todos (o algunos de) los archivos de datos a partir de la copia de respaldo y recuperar la base de datos completa. El segundo tipo es una *recuperación de tablespace.* Es factible activar una recuperación del medio sobre un tablespace específico mientras una porción de la base de datos está abierta y funcionando. Ello implica que se recuperarán todos los archivos de datos pertenecientes al tablespace. El tercer tipo de recuperación es la *recuperación de archivo de datos.* Aquí se puede recuperar un archivo de datos específico mientras está en uso el resto de la base de datos. Estas tres opciones utilizan los mismos criterios para determinar si se pueden recuperar o no los archivos.

Cuando un proceso recupera un archivo de datos, primero procede a bloquear el archivo de datos en modo EXCLUSIVE. Si el proceso no puede obstruir el archivo porque algún otro proceso tenga un bloqueo sobre él (ya sea porque el archivo esté online o porque otro proceso está haciendo recuperación del archivo de datos), entonces Oracle devuelve un error diciendo que el archivo de datos está en uso. Ello evita que dos sesiones de recuperación recobren el mismo archivo e impide la recuperación del medio de un archivo que está en uso.

Durante el proceso de recuperación del medio, el redo de todos los threads activos es aplicado. Oracle conserva una lista inicial de threads activos que debe recuperar. A medida que empieza a inspeccionar los archivos redo log, sabe si se han activado nuevos threads. De ser así, también aplicará recuperación a dichos threads. El último registro de redo en cada thread de redo es un registro de *end of thread,* que indica a Oracle que no hay más redo que aplicar para ese thread concreto. La recuperación de un thread particular se completa cuando se aplica este registro.

La recuperación del medio se completa una vez que se han recuperado todos los threads activos hasta el final de cada uno de ellos.

Mientras se aplica el redo, Oracle puede necesitar conmutar entre threads para recuperar hacia adelante los bloques suficientes para aplicar el siguiente elemento de redo. Oracle puede necesitar la aplicación del mismo log archivado múltiples veces en el caso de que contenga diversos bloques que hayan sido modificados por otros threads. Durante la recuperación del medio, se aplicará un redo de thread hasta que alcance el marcador de end_of_thread o hasta que se requiera aplicar un cambio de redo que se encuentra en el futuro de un bloque. Si un thread encuentra que tiene redo en el futuro de un bloque, la recuperación conmutará a otro thread. Ocasionalmente, se debería recuperar hacia delante el bloque lo suficiente como para aplicar este elemento de redo. El ejemplo en la Figura 3.2 ilustra el concepto de conmutación de threads.

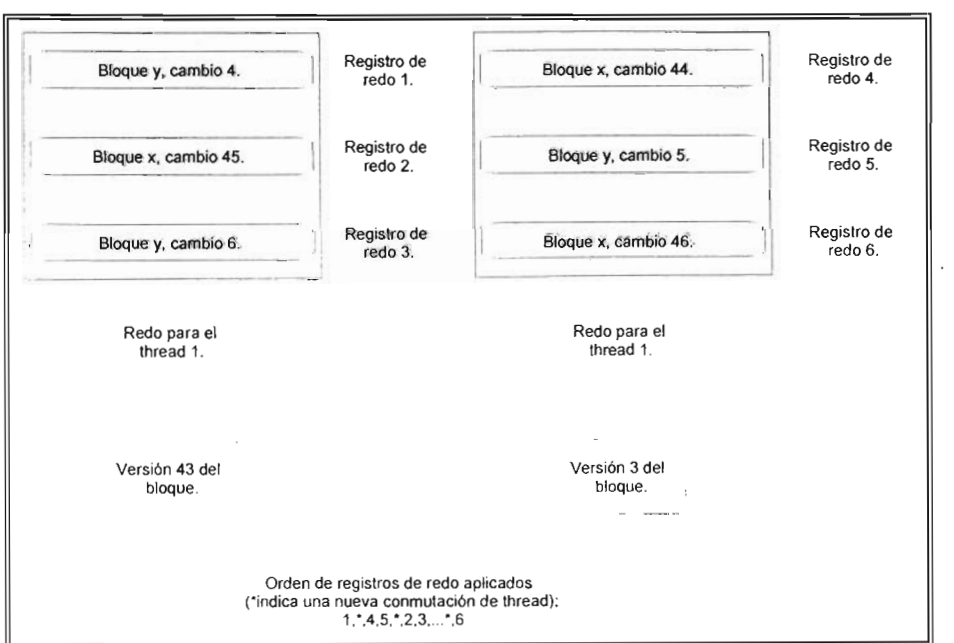


Figura 3.2 Proceso de Conmutación de thread.

La Figura 3.2 muestra dos bloques de base de datos y redo para dos threads. El registro de redo 1 del thread 1 contiene el cambio 4 de bloque de datos "y". Puesto que el bloque "y" tiene el número de versión 3 en disco, éste es el siguiente cambio que se debe aplicar. Sin embargo, el registro de redo 2 del thread 1 contiene el cambio 45 y éste no puede aplicarse al bloque "x", ya que el redo está en el futuro del bloque. En otras palabras, se debe aplicar el cambio 44 al bloque antes de poder aplicar el cambio 45; por tanto, Oracle debe conmutar threads y empezar a aplicar la recuperación comenzando desde el registro de redo 4 del thread 2. Obsérvese que después de aplicar el registro de redo 4, Oracle continúa con la aplicación de redo en este thread hasta que es forzado a conmutar de nuevo o hasta que complete la recuperación para este thread.

La Ilustración 3.2 muestra el orden en que se aplican los registros de redo en este ejemplo. El asterisco (*) indica que ha ocurrido una conmutación de thread.

- Requisitos previos para la utilización de una recuperación del medio.

Si ocurre un fallo del medio mientras la base de está operando en modo NOARCHIVELOG, podría provocar que no fuera posible lograr la recuperación completa mediante el empleo de respaldos físicos. Es decir, si se realizan backups offline semanalmente, debe estarse preparado para la contingencia de a perder, en el peor de los casos, una semana de datos si ocurre un fallo del medio. Lo anterior obedece a que los cambios hechos sobre la base de datos no son almacenados en los archivos de log guardado, ya que la base de datos está trabajando en modo NOARCHIVELOG, por ello, ejecutar la base en este modo sólo está indicado si existe la posibilidad de recobrar los datos, si fuera necesario, durante un fallo del tipo de archivos dañados y del tipo de fallo del medio ocurrido.

Se podría realizar una recuperación completa (lo que implica la restauración de archivo[s] de datos a partir de un respaldo, aplicando *todos* los cambios realizados desde el momento de la realización de dicho respaldo y recuperar hacia adelante la base de datos completamente) sin perder ningún dato.

Advertencia: Para la siguiente explicación, se asume que la base de datos está operando en modo ARCHIVELOG.

Como se observó con anterioridad, se pueden ejecutar tres tipos de órdenes de recuperación:

- Recover database.
- Recover tablespace.
- Recover datafile.

La orden de recuperación a emplear dependerá del tipo de fallo que se haya producido o de la circunstancia de querer mantener abierta la base de datos mientras la recuperación se efectúa. El término para esta última opción, (restauración de un archivo de datos o un tablespace mientras la base de datos está abierta), es *recuperación online.* Si por el contrario, la base de datos está cerrada mientras la recuperación se lleva a cabo, el término será *recuperación offline.* La siguiente tabla resume qué tipo de recuperación se puede realizar durante la restauración de la base de datos de un tablespace o de un archivo de datos.

Orden de recuperación.	**Base de datos online.**	**Base de datos offline.**
Recover database	No	Sí
Recover tablespace	Sí	No
Recover datafile	Sí	Sí

Cuando se ejecuta una orden **recover database,** la base de datos debe siempre de estar montada, *pero no abierta*[24]. Puesto que un tablespace es una entidad lógica, Oracle lo reconoce sólo cuando la base de datos está abierta; por consiguiente, cuando se emplea la orden recover tablespace, la base de datos tendrá que estar abierta, lo cual no será óbice para que el tablespace se recupere offline. (El tablespace SYSTEM no se puede recuperar utilizando la orden **recover tablespace,** ya que no se puede poner offline).

Para recuperar un archivo de datos, se puede usar la orden **recover datafile** sin que tenga relevancia que la base de datos esté abierta o cerrada, pero no deberá perderse de vista qué tipo de archivos se estén recuperando; es decir, si se están recuperando los archivos de datos de SYSTEM, la base de datos debe estar cerrada, puesto que la base de datos no puede trabajarse con los archivos de datos de SYSTEM offline. Si se intenta recobrar archivos pertenecientes a un tablespace de usuario, la base de datos puede estar abierta, pero los archivos a recuperación deben estar offline.

- Recuperación e implementación de base de datos.

Oracle proporciona al DBA una serie de opciones de recuperación del medio a nivel de base de datos. Con independencia del método a emplear, el concepto fundamental de la recuperación es muy simple: antes de abrir la base de datos, todos los archivos de datos deben estar recuperados hasta el lapso de tiempo exacto y no deben haber cambios en el futuro a partir de dicho punto; a manera de muestreo consultemos la figura 3.3 para pronta referencia:

[24] PEPIN, David. *"Oracle : DBA's guide"* Carmel, Indiana : Que, c2003

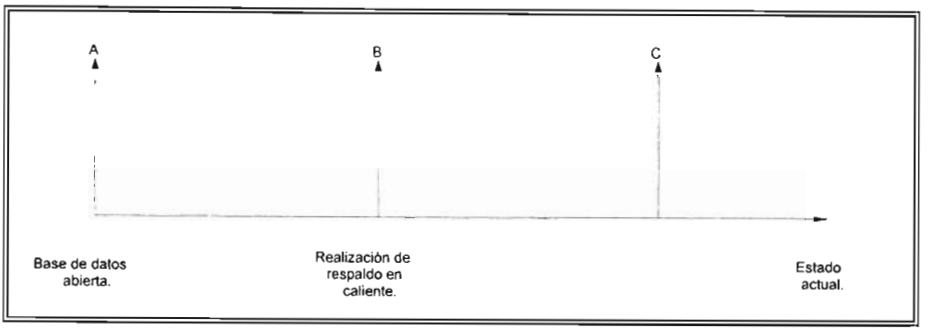


Figura 3.3 Proceso de Conmutación de thread.

La base de datos se encuentra abierta en el estado A, se realiza un backup en caliente en el estado B, y el estado actual de la base de datos es C.

En el supuesto caso que se presentara un fallo del medio que propiciara la pérdida de un archivo de datos vinculado a un tablespace de usuario, existen dos opciones de recuperación:

- ✓ Se pueden recuperar todos los archivos de datos del estado B y restaurarlos al estado C utilizando la orden **recover database.**

- ✓ El archivo de datos dañado puede ser recobrado a partir de un backup; para ello se requiere montar la base de datos, poner offline el archivo de datos y abrir la base de datos antes de su restauración con la orden **recover datafile.**

En cualquier caso, al finalizar la recuperación, todos los archivos deben estar en el estado C (o el estado actual, cualquiera que sea). No se debe iniciar la base de datos con un archivo de datos en estado B y el resto en estado C, ello podría ocasionar inconsistencia en la base de datos. Para evitar situaciones como esta, Oracle realiza el seguimiento de la consistencia de los archivos.

Es posible y hasta necesario en algunas situaciones radicales, iniciar la base de datos en un modo inconsistente. En tal circunstancia es preciso tomar una serie de precauciones, debiendo reconstruir la base de datos después de abrirla. La razón de que se presenten circunstancias de tal magnitud obedecen a que el DBA no ha definido un procedimiento de respaldo adecuado. Es poco probable que un DBA con un buen plan de respaldo y restauración tenga que enfrentar una situación como la descrita.

La sintaxis para utilizar la recuperación de base de datos es la siguiente:

```
RECOVER [AUTOMATIC] [FROM 'location'] [DATABASE]
[UNTIL CANCEL]
[UNTIL TIME date]
[UNTIL CHANGE integer]
[USING BACKUP CONTROLFILE)
```

Las palabras clave entre corchetes son opcionales. Si se utiliza la opción AUTOMATIC, la recuperación se realiza de manera automática, sin cuestionar al DBA los nombres de los archivos redo log durante la recuperación del medio.

Alternadamente, la orden **set autorecovery on I off** puede ser empleada desde el SVRMGR para activar o desactivar automáticamente la recuperación. No obstante, cuando a Oracle se le ordena que realice una recuperación automática, los archivos de log archivado deben encontrarse en la ubicación especificada por el parámetro LOQ_ARCHIVE_DEST del archivo INIT.ORA y el formato del nombre del archivo debería ser el mismo especificado en el parámetro LOG_ARCHIVE_FORMAT de INIT.ORA. Si no se desea que Oracle realice recuperación automática, se omitirá esta opción durante la utilización de la orden **recover.**

Esto provoca que Oracle pregunte el nombre del siguiente archivo de log archivado, debiendo especificarse. Otra disyuntiva es escribir la orden **alter database recover** para ejecutar una recuperación del medio de la base de datos.

La siguiente palabra reservada es FROM, que también es opcional. Se emplea en el caso de que la ubicación del archivo sea diferente a la especificada en el parámetro LOG_ARCHIVE_DEST del archivo INIT.ORA.

Si no se aplica la palabra reservada UNTIL, Oracle da por sentado que se está solicitando una recuperación de la base de datos completa; v.gr., la orden -

SVRMGR> **RECOVER DATABASE**

realiza una recuperación del medio en todos los archivos de datos que se encuentren online, de considerarse necesario. Si todas las instancias se han bajado de manera ordenada y no se restauran archivos de backup, este comando propiciará un error informando que no se requiere recuperación. El mismo resultado se obtendrá si alguna instancia tiene abierta la base de datos, ya que habrá bloqueos sobre los archivos de datos. La recuperación de base de datos se puede ejecutar sólo en el caso de que la base de datos esté cerrada pero montada.

- Recuperación completa frente a recuperación incompleta.

Frente a un fallo del medio, la recuperación de una base de datos (sin perder ninguno) se denomina recuperación completa. Se designa recuperación incompleta si posterior a la restauración se perdiera alguna información específica de la base de datos. Es recomendable la recuperación completa en el caso de que se encuentren disponibles todos los archivos redo log, archivos de datos de respaldo (para todas las bases de datos dañadas o perdidas) y un archivo de control actual y válido.

La recuperación incompleta debe utilizarse ante la imposibilidad de recuperar todos los datos por completo, v.gr., para recuperar la pérdida de un archivo redo log online o archivado o de archivos de control. Esta opción también es la adecuada para recuperar la base de datos a una circunstancia previa en el tiempo. Por ejemplo: si accidentalmente se destruye (drop) una tabla a las 10:00 A.M. y se requiere su recuperación, es posible restaurar los archivos de datos apropiados a partir de un backup, realizando la recuperación hasta un determinado punto en el tiempo, es decir, recuperación incompleta hasta un punto antes de las 10:00 A.M.

Para ello existen tres opciones: UNTIL CANCEL, UNTIL TIME y UNTIL CHANGE. Estos comandos agregados a la orden **recover database** permiten la recuperación basada en cancelación, en tiempo y en SCN, respectivamente[25].

Si se elige la opción UNTIL CANCEL, Oracle permite recuperar hacia adelante los archivos de log uno por uno. Cuando sea pertinente la detención de la recuperación, se ejecuta la orden cancel. Cabe precisar que en este modo no se aplican de forma automática los archivos de log online. Si la recuperación es de diversos threads de redo, puede haber archivos de log en otros threads que se hayan aplicado fraccionariamente al momento de cancelar la recuperación.

La opción UNTIL TIME permite al DBA recuperar hasta un determinado lapso de tiempo de un archivo redo log. Como ya lo señalamos esta opción funciona similarmente a la UNTIL CHANGE, salvo que se debe determinar una hora en lugar de un SCN.

[25] KOCH, George. "*Oracle : The Complete Reference*". Berkeley: México: Osborne McGraw-Hill, c1990.

Por último, la opción UNTIL CHANGE restaura la base de datos a un estado consistente. Oracle anota como referencia el SCN especificado, aplicándose todos los registros de redo con un SCN inferior a la referencia. Esta opción concluye la aplicación del redo asociado con ese SCN o superior. Así, la transacción correspondiente al SCN particular es deshecha. Si el DBA requiere recuperar hasta una transacción con un SCN específico, entonces añadirá una unidad al SCN especificado en este orden.

A continuación se enlistan algunos modelos de recuperación incompleta:

```
SVRMGR> recover database until cancel;
SVRMGR> recover database until time '1995-04-15:17:55:00';
SVRMGR> recover database until change integer.
SVRMGR> recover database until cancel using backup controlfile;
```

El primer modelo describe la recuperación hasta que la orden cancel es ejecutada. La segunda orden efectúa la restauración hasta un punto determinado en el tiempo, el cual se encuentra señalado entre apóstrofes. En este caso se aplican a la base de datos todos los cambios operados hasta las 5:55 P.M. del día 15 de abril de 1995. El tercero produce la recuperación hasta un SCN, especificado como un entero. La última orden es como la primera, salvo que en ella se emplea un archivo de control de backup para la recuperación.

- Opción RESETLOGS.

La opción RESETLOGS es necesaria cuando una base de datos es abierta con posterioridad a la ejecución de una de las acciones siguientes:

- Recuperación incompleta.
- Recuperación utilizando un archivo de control de backup.
- Recuperación con un archivo de control creado utilizando la orden **create controlfile** con la opción RESETLOGS.

Es procedente señalar que si se emplea esta opción para abrir la base de datos, Oracle desecha el redo que no se haya aplicado durante la recuperación, asegurándose que nunca más se vuelva a aplicar. A continuación inicializa la información del archivo de control respecto de los archivos de log y los threads de redo.

Mientras se lleva a cabo una recuperación completa de la base de datos, si se han aplicado totalmente los threads de redo a todos los archivos de datos online, se puede asegurar que la base de datos es consistente. Por otra parte cuando se lleva a cabo una recuperación incompleta, surge la posibilidad de restaurar un archivo desde un backup que no sea suficientemente antiguo. Éste es en definitiva el camino a seguir si el archivo tiene un checkpoint diferente al resto de archivos. Por esta razón, antes de abrir la base de datos con la opción RESETLOGS, se debe comprobar que la totalidad de los archivos de datos se hayan recuperado hasta el instante preciso de tiempo para estar en posibilidad de asegurar la consistencia de la base de datos.

En el archivo de control se mantienen un resetiogs SCN y un *contador* para identificar de forma exclusiva cada una de las ejecuciones de una apertura de base de datos con la opción RESETLOGS. Los valores se escriben en la cabecera de cada archivo de datos y archivo redo log. La aplicación de un archivo redo log durante una recuperación es imposible si su número de secuencia de log no coincide con exactitud al esperado por Oracle. No se puede recuperar un archivo de datos de un backup realizado antes de abrir la base de datos con la opción RESETLOGS. Lo anterior es con el objeto de asegurar que no vuelvan a la base de datos los cambios descartados al inicializar los archivos de log. Por tanto, se debe tomar muy en cuenta la realización de un respaldo de base de datos (online u offline) inmediatamente después de abrir la base de datos con la opción RESETLOGS. Lo anterior no es óbice para que los tablespaces de sólo lectura y cualquier tablespace que estuviera offline con la opción NORMAL se puedan poner online aun después de abrir la base de datos con la opción RESETLOGS.

3.2.2 Recuperación con la herramienta import.

El concepto básico que sustenta la herramienta import es muy simple. Su función es insertar los objetos que son extraídos desde una base de datos Oracle por medio de la herramienta export dentro de otra base de datos. El archivo que genera el export (archivo de tipo dump .DMP), sólo puede ser leído por la herramienta import.

La herramienta import lee las definiciones de los objetos y de las tablas de datos que la herramienta export extrajo de una base de datos Oracle y las almacena en un archivo con formato binario.

Una de las aplicaciones más frecuentemente empleadas del import consiste en la extracción de grandes cantidades de información copiando el archivo DMP, para con posterioridad bajarlo a otra base de datos alojada en una computadora que no comparta los mismos recursos de la red, a través de cualquier medio magnético u otro dispositivo de almacenamiento.

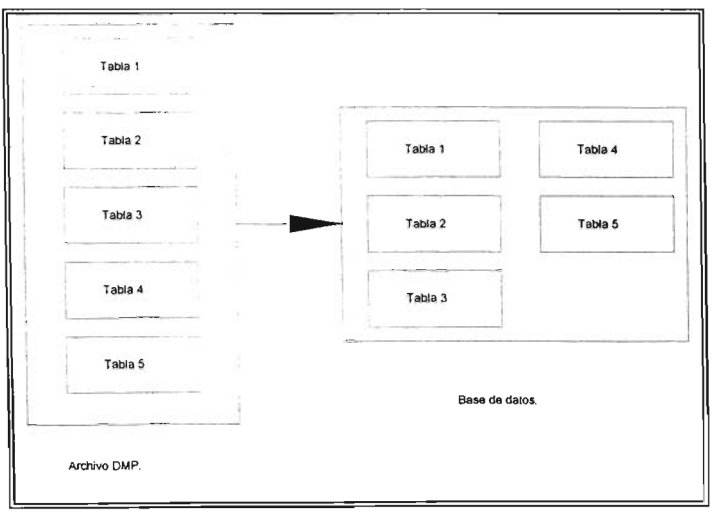


Figura 3.4 Importación de un archivo DMP.

En la figura 3.4 se describe la importación de los objetos. Éstos son importados leyéndose desde el archivo que genera la herramienta export. El archivo DMP contiene objetos en el siguiente orden:

1. Definiciones de las tablas.
2. Datos de las tablas.
3. Índices de las tablas.
4. Contraints, triggers y procedimientos almacenados.

Esta secuencia previene también que los datos sean rechazados debido al orden en que las tablas son importadas. De la misma forma evita que los triggers vayan a dispararse varias veces durante la importación (una cuando es originalmente insertado y otra durante la importación).

A manera de ejemplo, si la tabla SCOTT.EMP tiene un constraint de integral referencial sobre la tabla SCOTT.DEPT y la tabla EMP es importada primero, serían rechazados todos los registros que hicieran referencia a los departamentos, si estuvieran los triggers activados. Cuando los datos son importados en tablas que ya existían, es probable que existan registros que sean rechazados por la integral referencial.

- Opciones de importación.

La herramienta import provee tres diferentes modos de importación. Los objetos que son importados dependen de la forma en que se efectuó dicha importación así como del modo de exportación que haya sido usado. Todos los usuarios tienen dos tipos de importación. Un usuario con el role IMP_FULL_DATABASE tiene tres opciones:

Tabla	Este modo permite importar tablas en particular en vez de todas las tablas que le pertenezcan.
Usuario	Este modo faculta la importación de todos los objetos pertenecientes al usuario (tablas, datos, permisos e índices entre otros).
Full Database	Sólo los usuarios con el role IMP_FULL_DATABASE pueden importar en este modo, el cual importa la base de datos completa a través de un archivo .DMP generado por la herramienta export.

- Utilizando la herramienta import.

Se puede llamar a la herramienta import de tres maneras.

1. Utilizando el siguiente comando:

imp username/password PARFILE = filename

El archivo parfile funciona de una manera similar al usado por la herramienta export. Es un archivo que contiene los parámetros que típicamente se usan.

2. Utilizando el comando

imp username /password <parameters>

Remplazando <parameters> con los parámetros que el usuario necesitara utilizar.

Los parámetros que utiliza el export son:

✓ **ANALIZE:**

Default :Y

Se especifica cuando la herramienta import utiliza sentencias de SQL ANALIZE encontradas en el archivo export.

✓ **BUFFER:**

Default : Depende del sistema operativo.

El tamaño del buffer es el número de bytes que utiliza Oracle para transferir los registros desde el archivo DMP a la base de datos.

✓ **CHARSET:**

Default : Ninguno.

Determina el juego de caracteres que usará el import para extraer los datos.

✓ **COMMIT:**

Default :N

Especifica si dará la instrucción commit después insertar cada objeto.

✓ **DESTROY:**

Default : N

Declara que los datos existentes en la base de datos provenientes del archivo DMP serán sobrescritos.

✓ **FEEDBACK:**

Default : 0

Enumera una especie de barra de progreso. Por ejemplo si se define FEEDBACK =10, por cada punto en la barra de progreso se entiende que Oracle insertó diez registros.

✓ **FILE:**

Default :expdat.dmp

Diferencia el nombre del archivo origen de los datos.

✓ **FROMUSER:**

Default : Ninguno

Define el nombre del usuario dueño de los objetos a exportar. Es una opción muy útil cuando se realizan migraciones de bases de datos.

✓ **FULL:**

Default : N

Señala si es una importación completa desde un archivo DMP.

✓ **GRANTS:**

Default : Y

Observa si los grants serán importados o no.

✓ **HELP:**

Default :N

Muestra la ayuda acerca de la herramienta import.

✓ **IGNORE:**

Default :N

Define si mostrará o no los errores en tanto corre el proceso de import.

✓ **INCTYPE:**

Default : No definido.

Puntualiza el tipo de importación incremental que se utilizará. Existen dos modos:

SYSTEM	Importa las versiones más recientes de los objetos del DBMS.
RESTORE	Importa los objetos de todos los usuarios de la base de datos

✓ **INDEXES:**

Default :Y.

Advierte si se copiarán los índices o no.

✓ **INDEXFILE:**

Default : Ninguno.

Describe el datafile donde se crearán los índices de la base de datos.

✓ **LOG:**

Default : None.

Nombre del archivo de bitácora donde almacenará todos los errores probables que llegara a presentar la importación.

✓ **ROWS:**

Default : Y.

Señala si se importarán los registros de las diferentes tablas.

✓ **SHOW:**

Default Y.

Este parámetro tiene la función de mostrar el contenido del archivo DMP. Las sentencias SQL se mostrarán en el orden que serán ejecutadas por la herramienta import.

✓ **TABLES:**

Default : Ninguno.

Establece el nombre de las tablas a exportar. Si se escribe un asterisco (*) se entiende que se deben importar todas las tablas.

✓ **TOUSER:**

Default : Ninguno.

Enlista los nombres de usuario de los que su esquema será importado.

✓ **USERID:**

Default :No está definido.

Determina el login y el password de un usuario que está realizando una exportación.

3. Introduciendo el comando

imp username/password

Con esto se comenzará una sesión interactiva con la herramienta import. La herramienta no pregunta todos los parámetros que se especificaron anteriormente pero los más importantes sí.

3.3 Oracle Recovery Manager (RMAN).

RMAN es una herramienta de Oracle que se utiliza para respaldar, restaurar y recuperar bases de datos. RMAN inicia en el servidor donde reside la base de datos procesos de respaldo y restauración de la propia base.

RMAN tiene un lenguaje intérprete de comandos (CLI), y puede ser ejecutado en modo interactivo o modo batch. RMAN provee de las siguientes categorías de comandos:

- ✓ Respaldo, restauración copia y recuperación de bases de datos.
- ✓ Comandos para el mantenimiento y recuperación de la base de datos.
- ✓ Almacenamiento de comandos de mantenimiento.
- ✓ Reporte y lista de comandos.

En el capítulo 2 se expuso el funcionamiento del RMAN en el modo de respaldo. En este apartado se describirá su operación en modo de respaldo y restauración.

3.3.1 Catálogo de Recuperación.

Un catálogo de recuperación es un repositorio de información que es usado por el RMAN para determinar cómo puede ejecutar peticiones de respaldo y recuperación de base de datos.

El catálogo de recuperación contiene la información que se enlista a continuación:

- ✓ Copias de los datafiles.
- ✓ Copias de los archivos redo log online y de los redo log en modo archived.
- ✓ Sentencias SQL pertenecientes al proceso de restauración, conocidas como scripts almacenados.

Oracle recomienda que se use RMAN en caso de emplear más de veinte datafiles. Por ello el catálogo de recuperación requiere sincronizarse con el archivo de control para que la recuperación se realice de manera óptima en el caso de que se presentara una caída.

No obstante, en el caso de que no existiera un catálogo de recuperación, RMAN es capaz de trabajar con el archivo de control en su lugar. El inconveniente sería que no todas las configuraciones estarían disponibles, v.gr.:

- ✓ Ejecución de procesos almacenados.
- ✓ Restauración y recuperación si el archivo de control está dañado.
- ✓ Recuperación de un tablespace en modo point_in_time.

En razón de lo anterior, se recomienda que los archivos de parámetros estén en modo de multiplexión previendo la posibilidad de una falla.

- Generación de reportes.

Es pertinente contar con un registro de cuáles fueron los archivos que se respaldaron y cuáles son los necesarios para una recuperación de la base de datos. Por ello existe el modo de reporte el cual desarrolla un informe detallado del catálogo de recuperación.

El modo de reporte del RMAN despliega información del siguiente tipo:

- ✓ Los archivos que necesitan ser respaldados.
- ✓ Que archivos no han sido respaldados recientemente.
- ✓ Los archivos de respaldo que necesitan ser borrados.

De la misma forma en como se generan reportes, RMAN es capaz de listar los procesos que puede desarrollar dependiendo del catálogo de recuperación, v.gr.:

- ✓ Define con qué datafiles puede trabajar.
- ✓ Lista los redo logs y redo log en modo archived almacenados.
- ✓ Define el número de respaldos en línea.

Así, RMAN puede determinar de manera inmediata cuál es la falla en la base de datos y automáticamente realizar el proceso correcto.

Capítulo IV Escenarios prácticos de fallas.

Una vez que acontece la caída de una base de datos, se requiere una recuperación a nivel operacional dentro de los 30 minutos subsecuentes a la falla, la cual pudo haber sido ocasionada por un corte eléctrico, un error humano o un desperfecto en el hardware, entre otros. Aun con este margen, existen aplicaciones que necesitan ser reestablecidas en menos de tres minutos; esto debido a los costos ocasionados por la pérdida de información y tiempo en el que el sistema no estuvo disponible.

El mantener la disponibilidad y consistencia de una base de datos es más difícil cuando se efectúa un gran volumen de transacciones y las aplicaciones tienen un cierto grado de complejidad. Las fallas técnicas, como la falta de suministro eléctrico o algún problema en el hardware dependerán de la previsión con que se cuente. La disponibilidad de la base de datos estará estrechamente vinculada a la capacidad del DBA para restaurarla en el menor tiempo posible.

Una base de datos disponible y consistente debe su eficiencia a una metodología operacional que comprenda las siguientes características estratégicas:

- **Prevención**

  La disminución de riesgos que probablemente dañen la base de datos dependerá directamente de la confianza que se tenga en la infraestructura operacional de la compañía. Una empresa debe tener la certeza de funcionalidad tanto en el de hardware como en el software; un buen sistema de seguridad, procedimientos automatizados, cambios justificados en las rutinas, planeación en los procesos y pruebas de los sistemas. Una estructura confiable es formada a través de una buena comprensión de las necesidades del negocio.

- **Detección**

  Las posibles fallas y sus duraciones son reducidas al límite mediante el empleo de herramientas de monitoreo capaces de detectar problemas y en consecuencia, iniciar automáticamente procesos de reparación. Estas herramientas deben detectar, alertar, reportar y responder a las anomalías del sistema de forma instantánea, minimizando los tiempos de caídas del sistema.

- **Reparación**

  La duración de una caída del sistema es reducida al mínimo cuando los procesos de reparación están completamente automatizados y son iniciados a su vez por los procesos de detección de fallas. Las técnicas de rescate requieren de una buena comprensión a los planes de recuperación y restauración, éstos deben estar y entendidos. Herramientas robustas de respaldo y recuperación son esenciales para desarrollar esta tarea.

4.1 Desarrollo de un plan de respaldo y recuperación.

Incorporar estas estrategias dentro de la infraestructura operacional de cualquier negocio hacen necesario tener un plan de recuperación en sitio, un plan contra fallas y un plan de recuperación contra desastres.

➢ Niveles de Servicio.

Para poder desarrollar un plan de respaldo y recuperación es necesario entender las prioridades del negocio y consecuentemente los niveles de servicio. Se requiere determinar si la disponibilidad del negocio debe mantenerse arriba durante 7 días por 24 horas por 365 días y evaluar el costo de una caída del sistema o en su defecto cuánto cuesta por minuto. Algunas compañías aseveran que su sistema debe mantenerse arriba en un esquema de 7 días por 24 horas, pero su misión crítica es exclusivamente de las 9:00 a las 17:00 horas, de lunes a viernes; por ende, las prioridades del trabajo deben ser especificadas.

En la tabla que se expone a continuación, se muestran ejemplos de disponibilidad en porcentaje, los tiempos promedio de sus caídas (en minutos y segundos) y los requerimientos del sistema basándose en los estándares que sugiere Oracle en ambientes de producción con un promedio de 5000 peticiones por minuto.

Disponibilidad.	***Tiempo de caída en segundos.***	***Tiempo de caída en minutos.***	***Requerimientos.***
99.00%	5256	87.6	Redundancia en la base de datos y en el hardware, múltiples conexiones de red, respaldos en caliente y un plan de recuperación ensayado.
99.25%	3942	65.7	Procedimientos probados de respaldo y recuperación de la base de datos (Menos de una hora) y monitoreo eficiente.

Disponibilidad.	Tiempo de caída en segundos.	Tiempo de caída en minutos.	Requerimientos.
99.50%	2628	43.8	Automatización de procesos de respaldo y recuperación, copias redundantes de la base de datos, modo archive activado.
99.75%	1314	21.9	Plan recuperación contra desastres (menos de 30 minutos)
99.90%	526	8.76	Redundancia remota.
99.99%	53	0.88	Automatización completa de detección y corrección de anomalías.
99.999%	5	0.08	Redundancia total dentro de la infraestructura de la base de datos, que incluya el cliente, aplicaciones, sistema y hardware.

- Estimando requerimientos del negocio.

Cuando los niveles de servicio y tolerancia de fallas han sido determinados y comprendidos, deben definirse los siguientes parámetros:

- ✓ ¿Cuánta información está dispuesto el negocio a perder, a causa de una caída del sistema?

Las compañías no esperan tener pérdidas en su información. No obstante, esto no puede efectuarse para todos los casos de contingencia. A manera de muestreo, durante una caída de una base de datos, todas las transacciones se perderán a partir de que no sean almacenadas en el archivo redo log.

Para responder esta pregunta, es necesario considerar el peor de los escenarios ¿Qué pasaría si por un desastre natural se perdiera el site de cómputo? en ese momento se evalúa si realmente existe la necesidad indefectible de no poder perder información o si puede considerarse aceptable cierto margen de pérdida.

La respuesta a esta situación evidenciará cuál será la inversión en un plano de recuperación. Esto afectará la estrategia de recuperación y la arquitectura de las aplicaciones.

- ✓ ¿Cuánto sería el tiempo admisible de recuperación del sistema en situaciones inesperadas?

El tiempo de recuperación del medio (MTTR "*Mean Time To Recover*" por sus siglas en Inglés), es el tiempo que se toma para recuperar y restaurar las operaciones end-to-end. Este periodo empieza desde que el software de detección es el primero en responder al problema, hasta que los clientes pueden regresar a sus actividades.

En este ejemplo se muestran todos los niveles que comprende una recuperación exitosa.

- Efectiva detección de fallas (Monitoreo y reporte de alertas).
- Buena definición de planes de contingencia (Automatización y ensayo de procedimientos).
- Iniciación y conclusión de planes de contingencia (Procesos automatizados).
- Restauración y recuperación rápida de la base de datos (Buenos planes de respaldo y recuperación).
- Abrir la base de datos eficazmente (Automatización y mejoramiento del funcionamiento).
- Desempeño óptimo de segmentos de rollback (Definir el tiempo que tomará el deshacer cada transacción).
- Automatización contra desastres (Definir dentro de la aplicación una probable base de datos alterna, en el caso de que no pudiera conectarse a la original).

Esta característica definirá la frecuencia de los respaldos de la base de datos. No obstante, mientras más complicada sea la técnica de respaldo y recuperación de la base de datos el costo se incrementará. En la figura 4.1 se muestra la relación entre disponibilidad y costo en una base de datos.

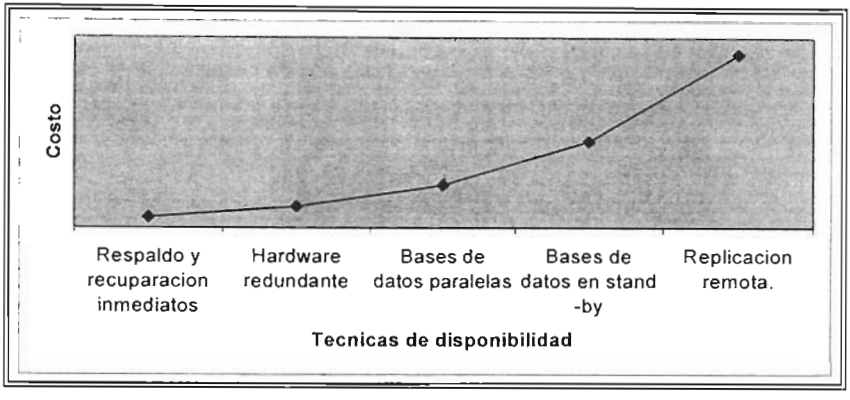


Figura 4.1 Costo de las técnicas de disponibilidad.

- ✓ ¿Cuál es el costo de la caída del sistema y de la pérdida de información?

El costo que tiene una caída del sistema se deriva de diferentes factores entre los que destaca la caída del negocio, productividad y la porción del mercado que ataca la compañía. Para cuantificar el costo del tiempo en el que la base de datos esté fuera de producción, es necesario definir cuánto puede invertirse en un plan contra desastres. Si se perdiera una cantidad muy grande de dinero por una caída del sistema por hora,, se requeriría contar con un plan de recuperación contra desastres que comprenda la consistencia de todo el hardware y software del sistema ya sea local como remotamente, técnicas de respaldo y recuperación eficientes y configuración de replicación para cualquier eventualidad.

- ✓ ¿Pueden las aplicaciones del sistema recuperarse de una restauración por objeto?

Una recuperación por objeto ofrece enormes ventajas, y si se planea adecuadamente, es capaz de recuperar el sistema de casi cualquier eventualidad. En lugar de desarrollar una recuperación de un tablespace o la totalidad de la base de datos cuando se elimina una tabla, realizando una importación de una base de datos en stand-by o bien cargarla desde cualquier otro medio, puede reducir el tiempo y el costo de la caída de manera considerable. Sin embargo esta opción no es válida en caso de desastres ya que se requeriría la opción de sincronizar los elementos del objeto; por ejemplo, si se necesitara la recuperación de algún datafile y ante la recuperación de todo el tablespace, sería necesario sincronizar todos los elementos para el funcionamiento óptimo de la base de datos.

- Prevención, detección y reparación.

Las recuperaciones óptimas son caracterizadas por prácticas y ensayos de los planes de recuperación, mecanismos eficientes de detección y automatización de la recuperación.

Las técnicas preventivas son esenciales; por ejemplo, precauciones de seguridad ya sean permisos para generar cualquier proceso en la base de datos y los respaldos con cierta periodicidad.. Sin una metodología de prevención, la probabilidad y la frecuencia de una caída del sistema se incrementa dramáticamente. Cuando se quiere tener una base de datos disponible y consistente siempre, debe tenerse un sistema administrado de una forma correcta, eficiente y con un sistema de seguridad.

Con estas características, la posibilidad de enfrentar una caída se reduce a comparación de un sistema mal configurado.

Existen cuatro puntos dirigidos a mantener el funcionamiento de la base de datos:

Mantenimiento.

El monitoreo de alertas y reportes acerca del hardware, de la red, de las aplicaciones y de la base de datos constituyen acciones que requieren realizarse regularmente. Las anomalías deben reportarse en tiempo real y mantener un récord histórico de cuál fue la acción con la que se respondió al problema.

Servicio.

Está intrínsecamente relacionado con la habilidad de operar efectivamente en áreas tales como: respaldo, reparación, actualización y reestructuración. Poniendo a correr procesos en lugar de analizar problemas cuando éstos suceden agilizará la estrategia de restauración.

Eficiencia.

La valoración de su trabajo es muy importante para cualquier usuario, esto implica características específicas como la cantidad, el tipo, la velocidad de las actividades del usuario.

Seguridad.

Uno de los factores de mayor trascendencia es la seguridad, la cual exige definir políticas de privacidad, de auditoría y accesos de control. La información

de la compañía es confidencial y por lo tanto, deben prevenirse actos que pudieran modificarla.

La mayoría de las fallas que pudieran tirar el sistema de base de datos son evitables, siempre y cuando las estrategias de respaldo estén implementadas para su detección y reparación. Esta característica requiere un sistema de mantenimiento con la habilidad de monitorear y reactivar el sistema, la base de datos y algunas excepciones de la red. Esto es esencial para llegar a conocer los requerimientos del MTTR.

4.1 Estimando la infraestructura contra de desastres.

Una evaluación de la infraestructura es necesaria para revelar posibles debilidades y resolverlas a tiempo. Con la consistencia apropiada en la arquitectura del sistema, las estrategias de recuperación, prevención y detección jamás serán utilizadas.

La figura 4.2, muestra las características principales de la detección, prevención y reparación. Se enfatizan los elementos particulares de cada proceso; no obstante debe subrayarse que la parte de prevención es la de mayor trascendencia ya que es la que soporta toda la infraestructura.

Lo ideal es desarrollar mecanismos que sean implementados para reducir la posibilidad de fallas; sin embargo si llegaran a ocurrir, la recuperación debe realizarse de forma automática.

Puede protegerse la base de datos de muy variadas formas; existen técnicas que no requieren una factura muy costosa o de un staff especializado.

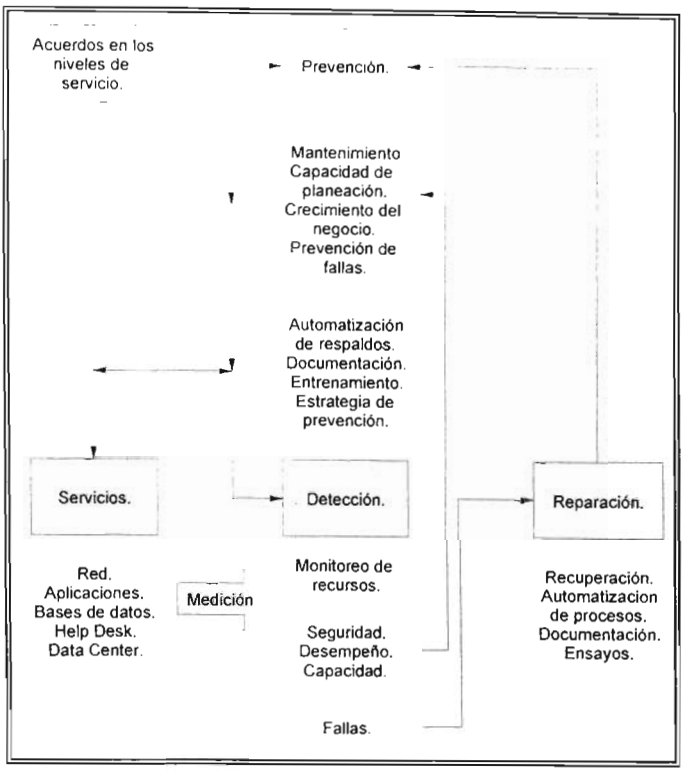


Figura 4.2 Prevención, detección y reparación.

- Habilitando el modo archive log.

El poner en marcha el modo archivelog implica que los archivos redo antiguos, que contienen todos los cambios de la base de datos pueden ser almacenados en una ubicación diferente a los archivos redo log y por ende, ser utilizados para realizar una recuperación de un datafile. En la mayoría de los escenarios, se requiere que la opción de archivelog esté habilitada.

Deben tenerse los archivos redo log y los archivos redo log en modo archivelog en diversas ubicaciones, esto con la finalidad de mejorar el desempeño del almacenamiento, mientras se escriben los archivos archivelogs, se puede estar rescribiendo alguno de los archivos redo log en otra ubicación. Esto es con la finalidad de mantener varios conjuntos de datos en multiplexión o en espejo y así evitar una posible caída del sistema.

Para poner en marcha el modo archive log es necesario utilizar los siguientes pasos: Oracle requiere de por lo menos dos conjuntos de archivos redo logs para almacenar los posibles cambios que sufriera la base de datos, en tanto que un conjunto es almacenado en disco, al otro conjunto se le insertan todos los cambios que se estén efectuando en tiempo real.

Es necesario modificar el archivo de parámetros que se encuentre en la en la siguiente ubicación $ORACLE_BASE/admin./pfile con nombre INIT"nombre de la instancia".ora. en el se encuentran las siguientes líneas que deben especificarse:

log_archive_dest = /u01/oradata/PROD/archivelogs/

En este debe especificarse la ubicación donde serán almacenados los archivos redo log en modo archive.

log_archive_format = arch_PROD_%S.arc

Este define el formato del nombre de los archivos redo log en modo archive; en este caso, maneja el nombre de la instancia "PROD" y los caracteres %S identifican un número dentro de una secuencia.

log_archive_start = true

Por último, este parámetro automatiza el proceso de archivelog, y en consecuencia no es necesario inicializarlo desde el comando *svrmgrl*. Sin embargo, puede realizarse este proceso de una manera manual. Después de haber realizado los pasos anteriores se utiliza el comando svrmgr.

SVRMGR> connect internal

Una vez que está conectado es necesario dar de baja la instancia que se pondrá en archive mode.

SVRMGR> shutdown immediate

El siguiente paso es iniciar la instancia; sin embargo es imprescindible no abrirla.

SVRMGR> startup mount

Esta es la instrucción más importante habida cuenta que se habilita el modo archivelog en la instancia de trabajo

SVRMGR> alter database archivelog;

Una vez realizado este proceso es necesario abrir la instancia.

SVRMGR> alter database open;
SVRMGR> archive log list

El último comando permite observar el estado del modo archive, el cual está en funcionamiento y la actual secuencia de los archivos redo log; si se requiere apreciar el funcionamiento del modo archive se puede suministrar la sentencia "alter system switch logfile;" para ver la forma en como se generan los nuevos archivos.

SVRMGR> connect internal
Password:
Connected.
SVRMGR> archive log list;
Database log mode Archive Mode
Automatic archival Enabled
Archive destination /archivelog
Oldest online log sequence 21538
Next log sequence to archive 21540
Current log sequence 21540
SVRMGR>

Asimismo la siguiente consulta SQL también revela si la instancia está en modo archivelog o no.

SVRMGR> select * from v$database;

Para deshabilitar el modo archivelog debe realizarse un procedimiento similar al que se efectuó para ponerlo en funcionamiento, pero se debe utilizar la instrucción "**alter database noarchivelog**". El parámetro *log_archive_start* en el archivo init.ora debería estar en un status FALSE permanentemente, evitando así el funcionamiento indeseado de este proceso.

➢ Reubicación de Datafiles

Un parámetro que es importante resaltar es el indicado para el manejo de los archivos de datos o datafiles, los cuales pueden ser movidos a otra ubicación o bien, cambiar su nombre de archivo (restaurándolo por ejemplo de un respaldo de una cinta si llegara a existir un fallo en el disco duro en el que reside el datafile) después de dar de baja la instancia y utilizando la sentencia "**alter database rename 'UBICACIÓN Y NOMBRE DEL ARCHIVO'**", para indicar al RDBMS Oracle que se ha cambiado la ubicación de dicho archivo. A manera de muestreo, se pueden analizar las siguientes sentencias, donde se observa el movimiento de un datafile de un disco con nombre u03 a otro con nombre u04 :

```
SVRMGR> connect internal
SVRMGR> shutdown immediate
SVRMGR> !mv /u03/oradata/PROD/devlPROD_1.dbf
/u04/oradata/PROD
SVRMGR> startup mount
SVRMGR> alter database rename file
'/u03/oradata/PROD/devlPROD_1.dbf'
2> to '/u04/oradata/PROD/devlPROD_1.dbf';
SVRMGR> alter database open;
SVRMGR> select * from v$datafile;
```

Puede apreciarse que el procedimiento es bastante simple: en primer lugar se da de baja la instancia, se modifica la ubicación del archivo "**devlPROD_1.dbf**" y se especifica al RDBMS Oracle la nueva ubicación antes de reiniciar la instancia.

Es posible desarrollar un cambio similar sin dar de baja la instancia modificando el tablespace relacionado al datafile que se quiera reubicar.

```
SVRMGR> connect internal
SVRMGR> alter tablespace development offline;
SVRMGR> !mv /u03/oradata/PROD/devlPROD_1.dbf
/u04/oradata/PROD
SVRMGR> alter database rename file
'/u03/oradata/PROD/devlPROD_1.dbf'
2> to '/u04/oradata/PROD/devlPROD_1.dbf';
SVRMGR> alter tablespace development online;
SVRMGR> select * from v$datafile;
```

En este ejercicio, se alteró el status del tablespace "*development*" poniéndolo también offline; esto con la finalidad de que todos los datafiles pertenecientes al enunciado tablespace pudieran moverse en un caso específico. En la especie, es útil esta operación cuando se requiera tener parte de la base de datos funcionando aunque la restante esté recibiendo mantenimiento.

Finalmente, otro caso que pudiera presentarse es la necesidad de cambiar de ubicación todos los datafiles de una instancia de un disco duro a otro. Esto se realiza mediante un proceso en el que se respalda el archivo de control generando un script en el que se pueden especificar las nuevas ubicaciones.

Este archivo es generado en el subdirectorio en el que se descargan los archivos con terminación "trc"; esta ubicación puede obtenerse mediante los siguientes comandos:

```
SQL> select value from v$parameter where name =
'user_dump_dest';
SVRMGR> show parameter user_dump
```

Ambas sentencias obtienen el mismo resultado:

```
SVRMGR> show parameter user_dump
NAME                    TYPE        VALUE
------------------------------------------------------------
user_dump_dest          string      /oracle/app/oracle/admin/bdweb
```

El siguiente paso se traduce en la realización del backup del archivo de control, para lo cual se efectúa el siguiente procedimiento:

```
SVRMGR> connect internal
SVRMGR> alter database backup controlfile to trace;
SVRMGR> show parameter user_dump
SVRMGR> shutdown immediate
$ mv /u03/oradata/PROD/*PROD*.dbf /u04/oradata/PROD
$ cd /u00/oracle/admin/PROD/udump
```

La segunda instrucción **alter database backup controlfile to trace;** realiza un script para generar un archivo de control; un ejemplo de este archivo denominado s000_611.trc, se enlista a continuación:

```
[oracle@billing3 bdump]$ more s000_611.trc
/oracle/app/oracle/admin/bdweb/bdump/s000_611.trc
Oracle8i Enterprise Edition Release 8.1.6.1.0 - Production
With the Partitioning option
JServer Release 8.1.6.0.0 - Production
ORACLE_HOME = /oracle/app/oracle/product/8.1.6
System name:    Linux
Node name:      billing3
Release.        2.2.14-5.0
Version:        #1 Tue Mar 7 20:53:41 EST 2000
Machine:        i686
Instance name: bdweb
Redo thread mounted by this instance: 1
Oracle process number: 12
Unix process pid: 611, image: oracle@billing3 (S000)
```

```
*** SESSION ID:(12.27038) 2003-05-28 16:52:08.359
*** 2003-05-28 16:52:08.359
# The following commands will create a new control file and use it
# to open the database.
# Data used by the recovery manager will be lost. Additional logs may
# be required for media recovery of offline data files. Use this
# only if the current version of all online logs are available.
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "BDWEB" NORESETLOGS
ARCHIVELOG
   MAXLOGFILES 32
   MAXLOGMEMBERS 2
   MAXDATAFILES 254
   MAXINSTANCES 8
   MAXLOGHISTORY 17016
LOGFILE
  GROUP 1 '/datafile1/oradata/bdweb/redo01.log' SIZE 500K,
  GROUP 2 '/datafile1/oradata/bdweb/redo02.log' SIZE 500K,
  GROUP 3 '/datafile1/oradata/bdweb/redo03.log' SIZE 500K
DATAFILE
  '/datafile1/oradata/bdweb/system01.dbf',
  '/datafile1/oradata/bdweb/tools01.dbf',
  '/datafile1/oradata/bdweb/rbs01.dbf',
  '/datafile1/oradata/bdweb/temp01.dbf',
  '/datafile1/oradata/bdweb/users01.dbf',
  '/datafile1/oradata/bdweb/indx01.dbf',
  '/datafile1/oradata/bdweb/drsys01.dbf',
  '/oracle/app/oracle/product/8.1.6/dbs/prueba44',
  '/oracle/app/oracle/product/8.1.6/dbs/prueba',
  '/datafile1/oradata/bdweb/prueba2.dbf',
  '/datafile1/oradata/bdweb/system02.dbf'
CHARACTER SET US7ASCII
;
# Recovery is required if any of the datafiles are restored backups,
# or if the last shutdown was not normal or immediate.
RECOVER DATABASE
# All logs need archiving and a log switch is needed.
ALTER SYSTEM ARCHIVE LOG ALL;
# Database can now be opened normally.
ALTER DATABASE OPEN;
# No tempfile entries found to add.
#
[oracle@billing3 bdump]$
```

Las primeras líneas explican la versión del manejador de la base de datos que se está utilizando, el nombre del servidor y el nombre de la instancia, entre otros detalles. De su análisis puede apreciarse que tiene la estructura de un programa en el que va a actualizar los archivos de control de la instancia; es necesario poner el carácter de comentario dentro de las primeras líneas hasta encontrar la sentencia **STARTUP NOMOUNT**. En este programa se encuentra una sección que hace referencia a los datafiles, no debe modificarse el orden pero sí es posible modificar la ubicación de los datafiles que se quieran reubicar. Una vez que se hayan reubicado los archivos, se borra la línea de **RECOVER DATABASE**, lo cual propiciará que el script quede de la siguiente manera:

```
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "BDWEB" NORESETLOGS
ARCHIVELOG
   MAXLOGFILES 32
   MAXLOGMEMBERS 2
   MAXDATAFILES 254
   MAXINSTANCES 8
   MAXLOGHISTORY 17016
LOGFILE
  GROUP 1 '/datafile1/oradata/bdweb/redo01.log' SIZE 500K,
  GROUP 2 '/datafile1/oradata/bdweb/redo02.log' SIZE 500K,
  GROUP 3 '/datafile1/oradata/bdweb/redo03.log' SIZE 500K
DATAFILE
  '/datafile2/oradata/bdweb/system01.dbf',
  '/datafile2/oradata/bdweb/tools01.dbf',
  '/datafile2/oradata/bdweb/rbs01.dbf',
  '/datafile2/oradata/bdweb/temp01.dbf',
  '/datafile2/oradata/bdweb/users01.dbf',
  '/datafile2/oradata/bdweb/indx01.dbf',
  '/datafile2/oradata/bdweb/drsys01.dbf',
  '/oracle2/app/oracle/product/8.1.6/dbs/prueba44',
  '/oracle2/app/oracle/product/8.1.6/dbs/prueba',
  '/datafile2/oradata/bdweb/prueba2.dbf',
  '/datafile2/oradata/bdweb/system02.dbf'
 CHARACTER SET US7ASCII;
 ALTER SYSTEM ARCHIVE LOG ALL;
 ALTER DATABASE OPEN;
```

Para ejecutar el script se realizan los siguientes comandos:

SVRMGR> connect internal
SVRMGR> @s000_611.trc

Con el carácter @ se ejecuta el script y el resultado es la nueva ubicación de todos los datafiles. Puede checarse el éxito de la ejecución con la sentencia:

SVRMGR> select * from v$datafile;

```
SQL> select name from v$datafile;
NAME
----------------------------------------------------------------
/datafile1/oradata/bdweb/system01.dbf
/datafile1/oradata/bdweb/tools01.dbf
/datafile1/oradata/bdweb/rbs01.dbf
/datafile1/oradata/bdweb/temp01.dbf
/datafile1/oradata/bdweb/users01.dbf
/datafile1/oradata/bdweb/indx01.dbf
/datafile1/oradata/bdweb/drsys01.dbf
/oracle/app/oracle/product/8.1.6/dbs/prueba44
/oracle/app/oracle/product/8.1.6/dbs/prueba
/datafile1/oradata/bdweb/prueba2.dbf
/datafile1/oradata/bdweb/system02.dbf

11 rows selected.
```

La nueva ubicación de los datafiles es observable mediante una vista del diccionario de datos.

➢ Agregando otro archivo redo log al conjunto.

En ciertos casos puede ser útil disponer de un archivo redo log extra asociado a un grupo existente en una ubicación diversa a la original, ya que podrán seguirse almacenando los cambios que llegaran a afectar la base de datos durante la ocurrencia de un desperfecto en un disco duro, bajo el esquema de espejo. Para lograrlo, se necesita especificar el nombre del nuevo archivo redo log, la ubicación y el grupo al que pertenece; su tamaño será idéntico a los otros miembros del grupo. A continuación se muestra la sintaxis para la creación de un elemento extra al conjunto de archivos redo log:

```
SQL> alter database add logfile member
2> '/archivelog/redo04.log' to group 1;
```

Cabe mencionar que una vez que se ha generado el archivo redo log en un determinado grupo, éste tendrá el status de **INVALID**, lo cual se debe a que el conjunto del archivo redo log en particular no ha estado en uso; para suprimir esa bandera, basta con efectuar la sentencia "*alter system switch logfile*". Para poder checar el status de los archivos redo log, puede utilizarse la siguiente instrucción:

```
SVRMGR> select * from v$logfile;
GROUP#    STATUS  MEMBER
--------- ------- ------------------------------------------------------
        1         /datafile1/oradata/bdweb/redo01.log
        2         /datafile1/oradata/bdweb/redo02.log
        3         /datafile1/oradata/bdweb/redo03.log
        1         /archivelog/redo04.log

4 rows selected.
```

A fin de agregar más miembros a cada uno de los grupos de archivos redo log es necesario regenerar los archivos de control. El procedimiento es similar al que se realizó para modificar la ubicación de todos los archivos datafiles; se genera un respaldo de un archivo de control y se modifica el parámetro maxlogmembers dependiendo el número de miembros de cada grupo; por ejemplo, el siguiente script tiene definidos cinco miembros por cada grupo de archivos redo log:

```
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "BDWEB" NORESETLOGS
ARCHIVELOG
   MAXLOGFILES 32
   MAXLOGMEMBERS 5
   MAXDATAFILES 254
   MAXINSTANCES 8
   MAXLOGHISTORY 17016
LOGFILE
  GROUP 1 '/datafile1/oradata/bdweb/redo01.log' SIZE 500K,
  GROUP 2 '/datafile1/oradata/bdweb/redo02.log' SIZE 500K,
  GROUP 3 '/datafile1/oradata/bdweb/redo03.log' SIZE 500K
DATAFILE
  '/datafile2/oradata/bdweb/system01.dbf',
  '/datafile2/oradata/bdweb/tools01.dbf',
  '/datafile2/oradata/bdweb/rbs01.dbf',
  '/datafile2/oradata/bdweb/temp01.dbf',
  '/datafile2/oradata/bdweb/users01.dbf',
  '/datafile2/oradata/bdweb/indx01.dbf',
  '/datafile2/oradata/bdweb/drsys01.dbf',
  '/oracle2/app/oracle/product/8.1.6/dbs/prueba44',
  '/oracle2/app/oracle/product/8.1.6/dbs/prueba',
  '/datafile2/oradata/bdweb/prueba2.dbf',
  '/datafile2/oradata/bdweb/system02.dbf'
CHARACTER SET US7ASCII;
ALTER SYSTEM ARCHIVE LOG ALL.
        ALTER DATABASE OPEN;
```

De esta manera pueden definirse cuántos miembros tendrá cada conjunto de archivos redo log.

➢ Agregando un nuevo conjunto de archivos redo log.

Puede agregarse un nuevo conjunto de archivos redo logs a los ya existentes, con la finalidad de evitar futuras fallas en la base de datos en tanto esté en funcionamiento el proceso de almacenamiento de las transacciones en los archivos redo logs. El proceso para efectuar la adición de un grupo se realiza de la siguiente manera:

```
SQL> alter database add logfile group 4
  2> ('/u00/oradata/PROD/logPROD_4a.rdo',
  3> '/u01/oradata/PROD/logPROD_4b.rdo') size 500K;
SQL> select * from v$logfile;
```

De lo anterior se observa que se genera un nuevo grupo (4), el cual tendrá dos archivos y cada uno medirá 500k. Con la consulta **select * from v$logfile;** pueden apreciarse todos los grupos de archivos redo log que mantiene la instancia actual.

➢ Multiplexión de archivos de control.

Otro proceso auxiliar en los métodos de prevención de caídas del sistema consiste en mantener varios archivos de control y que la instancia refleje en ellos cualquier cambio físico que ocurra. El proceso es algo complejo, dado que previamente debe darse de baja la instancia, copiar en diferentes ubicaciones el archivo de control y modificar el archivo de parámetros donde se especifica la ubicación de los archivos de control; todo ello con el objeto de que una vez que se restablezca el servicio de la instancia, ocupe los cambios efectuados en la base.

El proceso se efectuaría de la siguiente manera:

```
SVRMGR> connect internal
SVRMGR> shutdown immediate
```

Aquí se da de baja la instancia.

```
$ cp -p /u03/oradata/PROD/ctrlPROD_1.ctl /u01/oradata/PROD/ctrlPROD_2.ctl
$ cp -p /u03/oradata/PROD/ctrlPROD_1.ctl /u00/oradata/PROD/ctrlPROD_3.ctl
```

En un sistema UNIX, se puede utilizar el comando cp para copiar el archivo de control en diferentes ubicaciones.

```
$ vi $ORACLE_HOME/dbs/initPROD.ora
control_files = (/u03/oradata/PROD/ctrlPROD_1.ctl,
        /u01/oradata/PROD/ctrlPROD_2.ctl,
        /u00/oradata/PROD/ctrlPROD_3.ctl)
```

Es necesario agregar en el archivo de parámetros los nuevos archivos de control que se copiaron, separándolos por comas y especificando claramente su ubicación.

```
SVRMGR> connect internal
SVRMGR> startup
SVRMGR> select * from v$controlfile;
```

Una vez realizados los pasos anteriores puede levantarse la instancia.

➢ Aplicando la técnica de espejo para los archivos archive log.

Básicamente se trata de una técnica en la que es posible programar una tarea (un proceso batch en MSDOS o un cron en UNIX) en la que se copien los archivos archived redo logs cada determinado tiempo a diferentes ubicaciones, con la finalidad de obtener un respaldo en tiempo real, ante la eventualidad de una falla en el dispositivo de almacenamiento. Esto ayudaría en el caso de que se perdieran los últimos archived redo logs del dispositivo de almacenamiento y no se pudieran recuperar.

4.2 Respaldo de bases de datos Oracle desde el sistema operativo UNIX.

La Facultad de Estudios Superiores Acatlán tiene sus bases de datos en servidores Sun Solaris 5.7; por ello, se dará un enfoque particular bajo este esquema, habida cuenta que el objetivo se que persigue en este trabajo es dotar de un apoyo a la comunidad estudiantil de esta casa de estudios.

Los respaldos desde el sistema operativo UNIX pueden ser usados como una alternativa a diversas herramientas de Oracle entre las que destacan el "Enterprise Backup Utility" o el "Recovery Manager". Una de las razones por las que se usa este tipo de herramientas es la incompatibilidad del software o del hardware con las herramientas antes mencionadas.

➢ Respaldo de bases de datos con la herramienta cpio de UNIX.

El comando cpio es una herramienta que permite concatenar archivos y directorios, compactándolos para evitar que ocupen demasiado espacio en disco y que este procedimiento sea compatible con unidades de almacenamiento secuencial, como en el caso de las cintas.

No obstante, debe hacerse notar que es necesario tener la instancia fuera de operación para que ninguno de los archivos importantes de la base de datos (archivos de control, archivos datafiles y archivos de parámetros) pueda corromperse. Las opciones básicas del cpio son las siguientes :

✓ **-i**

Lee un archivo que ya existe, creado con la opción –o, a menos que tenga la opción –t, que extraerá toda la información del archivo de acuerdo con los patrones de búsqueda.

✓ **-o**

Escribe un nuevo archivo con la entrada estándar, usando una lista de archivos; un ejemplo podría ser utilizando el propio comando ls :

ls . | cpio -o >arch

En este ejemplo puede apreciarse la forma en que con la ayuda del "pipe" se puede definir como:

✓ **-p**

Copia todo el árbol de directorios especificados.

Existen algunas otras opciones en las que se requiere especificar el tamaño del bloque de datos que se utilizará (-C) esta opción es obligatoria para el trabajo de dispositivos secuenciales.

En un caso práctico podría especificarse que copiara desde la variable $ORACLE_HOME y $ORACLE_BASE con la opción -p, o realizar un trace de un archivo de control y extraer todas las ubicaciones de los archivos datafiles para copiarlos.

Esta herramienta es bastante similar en las diversas versiones de UNIX, lo cual le aporta la capacidad de emigrar datos de diferentes plataformas que no estén conectadas a la red corporativa.

Un ejemplo de ello podría visualizarse en la obtención de todas las ubicaciones, a través del diccionario de datos, mediante la vista v$datafiles:

```
SQL>header off
SQL>spool datafiles.txt
SQL> select name from v$datafile;

/home/oracle/billing/oradata/billing/system01.dbf
/home/oracle/billing/oradata/billing/tools01.dbf
/home/oracle/billing/oradata/billing/rbs01.dbf
/home/oracle/billing/oradata/billing/temp01.dbf
/home/oracle/billing/oradata/billing/call_dat04.dbf
/home/oracle/billing/oradata/billing/univ_dat03.dbf
/home/oracle/billing/oradata/billing/call_dat05.dbf
/home/oracle/billing/oradata/billing/idx_prod04.dbf
/home/oracle/billing/oradata/billing/call_ndx06.dbf

SQL> spool off
```

Y una vez obtenido tal archivo, puede programarse un script bajo esta leyenda:

```
for I in `cat datafiles.txt`
do
cpio –o –C 40960 $I >> respaldo
done
```

El resultado de lo anterior, permite guardar todos los archivos datafiles que se encuentren en la instancia en la que se esté trabajando. Es posible que existan varias instancias en una misma base de datos; por ello es necesario supervisar los datos contenidos en cada una para tener un respaldo integral; es recomendable checar el archivo "tnsnames.ora" ubicado en $ORACLE_HOME/network mediante el que se podrán examinar las instancias que residen en esa base de datos.

4.3 Respaldo de bases de datos Oracle.

Tal vez la tarea de mayor trascendencia del DBA, sea mantener la información disponible y consistente en todo momento y para todos los usuarios, lo cual implica contar con procedimientos de respaldo y recuperación de la base de datos capaces de actuar contra cualquier falla. En esta sección se expondrán las técnicas de respaldo de bases de datos, los tipos de respaldo y algunos métodos que podrían auxiliar al desempeño de los procesos antes mencionados.

4.3.1 *¿Archivos importantes a respaldar?*

Los tipos de archivos que se enlistan a continuación son vitales para el funcionamiento normal de una base de datos Oracle:

- ✓ Datafiles.
- ✓ Archivos de control.
- ✓ Archivos Redo Log.
- ✓ Archivos Redo Log en modo Archived. (Si está activado)
- ✓ Archivos de parámetros.
- ✓ Archivo de password (en caso de usarse)

La manera usual de obtener los nombres de los archivos de control, de los datafiles y de los archivos redo log es a través de una consulta al diccionario de datos:

```
SQLWKS> select name from v$datafile;
NAME
--------------------------------------------------------------
--
/datafile1/oradata/bdweb/system01.dbf
/datafile1/oradata/bdweb/tools01.dbf
/datafile1/oradata/bdweb/rbs01.dbf
/datafile1/oradata/bdweb/temp01.dbf
/datafile1/oradata/bdweb/users01.dbf
/datafile1/oradata/bdweb/indx01.dbf
/datafile1/oradata/bdweb/drsys01.dbf
/oracle/app/oracle/product/8.1.6/dbs/prueba44
/oracle/app/oracle/product/8.1.6/dbs/prueba
/datafile1/oradata/bdweb/prueba2.dbf
/datafile1/oradata/bdweb/system02.dbf
11 rows selected.
```

```
SQLWKS> select name from v$controlfile;
NAME
--------------------------------------------------------------
---
/datafile1/oradata/bdweb/control01.ctl
/datafile1/oradata/bdweb/control02.ctl
/datafile1/oradata/bdweb/control03.ctl
3 rows selected.
SQLWKS> select member from v$logfile;
MEMBER
--------------------------------------------------------------
---
/datafile1/oradata/bdweb/redo01.log
/datafile1/oradata/bdweb/redo02.log
/datafile1/oradata/bdweb/redo03.log
/archivelog/redo04.log
4 rows selected.
```

También se pueden obtener los tablespaces con sus respectivos datafiles mediante la siguiente consulta:

```
SQLWKS> select tablespace_name,file_name from dba_data_files
order by tablespace_name;
TABLESPACE_NAME                FILE_NAME
--------------------------------------------------------------
---
DRSYS
/datafile1/oradata/bdweb/drsys01.dbf
DRSYS
/oracle/app/oracle/product/8.1.6/dbs/prueba44
INDX
/datafile1/oradata/bdweb/indx01.dbf
RBS                        /datafile1/oradata/bdweb/rbs01.dbf
SYSTEM
/datafile1/oradata/bdweb/system01.dbf
SYSTEM
/datafile1/oradata/bdweb/system02.dbf
TEMP
/datafile1/oradata/bdweb/temp01.dbf
TOOLS
/datafile1/oradata/bdweb/tools01.dbf
USERS
/datafile1/oradata/bdweb/users01.dbf
USERS
/oracle/app/oracle/product/8.1.6/dbs/prueba
USERS
/datafile1/oradata/bdweb/prueba2.dbf
11 rows selected.
```

La lista de los archivos archive logs no puede obtenerse por medio de una consulta SQL, sin embargo, se pueden lograr las ubicaciones en donde se almacenan tales archivos, el formato de los nombres y si está activo el modo archive log cons a través de las siguientes sentencias:

```
SQLWKS> select value from v$parameter where name =
'log_archive_dest';
VALUE
--------------------------------------------------------------------
-----------
/archivelog/
1 row selected.
SQLWKS> select value from v$parameter where name =
'log_archive_format';
VALUE
--------------------------------------------------------------------
----------
arch_BDWEB_%t_%s.arc
1 row selected.
SQLWKS> select value from v$parameter where name =
'log_archive_start';
VALUE
--------------------------------------------------------------------
-----------
TRUE
1 row selected.
```

Los parámetros pueden ser obtenidos usando el siguiente SQL script:

```
SVRMGR> show parameter archive
SVRMGR> archive log list ----
```

Para obtener la lista de archivos redo log files de los últimos cinco días (la fecha es cuando el archivo fue generado y no cuando fue copiado al subdirectorio raíz de los archivos redo log) puede generarse un script para la vista v$log_history o a la tabla v$archived.

```
SQL> select archive_name from v$log_history
      2> where trunc(to_date(substr(time,1,8),'MM/DD/YY'))
>=
      3> trunc(sysdate)-5;
```

El nombre del archive de parámetros frecuentemente se encuentra en la siguiente ubicación: $ORACLE_HOME/dbs/initPROD.ora (Debe sustituirse el Oracle SID en el PATH).

Ante la contingencia de no lograrse el acceso a la base de datos o bien al archivo INIT.ora, podrían conseguirse todas las ubicaciones de los archivos importantes a respaldar si se ha usado el estándar que sugiere el manejador de bases de datos, utilizando el nombre de la instancia en cada uno de los nombres de estos archivos (datafiles, controlfiles, redo log files, and archive redo log files), en el sistema operativo UNIX puede utilizarse el siguiente comando a fin de obtener las enunciadas ubicaciones:

$ find / -name '*PROD*' ! -type d 2>/dummy

- Respaldos en Frío

Los respaldos en frío son realizados cuando la base de datos no se encuentra en estado operativo. Después de haber ejecutado el comando "shutdown immediatte", usualmente es cuando se obtiene la más reciente versión de todos los datafiles, archivos de control y redo log.

Durante el tiempo en que se está ejecutando el respaldo, la base de datos no estará en ejecución; los archivos se copiarán a una cinta o a algún otro dispositivo de almacenamiento (posiblemente éstos serán comprimidos). Este tipo de respaldo puede ejecutarse sin importar si la base de datos está configurada en modo archived o no.

- Respaldos en Caliente

Se hacen mientras la base de datos sigue dando servicio y la información está siendo actualizada. Esta opción debe ser utilizada solamente en sistemas cuyo funcionamiento se requiere 24-horas-por-día/ 7 días a la semana. Aún en el caso de que se contara con un espacio de media hora libre en el que la base de datos pudiera estar abajo, un respaldo en frío estaría contemplado como la mejor opción.

Nunca se deben respaldar los archivos redo log mediante el procedimiento de respaldo en caliente. Restaurar un respaldo de un archivo redo log que se encuentra dañado causaría la corrupción durante la recuperación de base de datos.

Hemos señalado que los archivos redo log almacenan cualquier cambio que realice la base de datos, esto hace a los redo logs, los archivos más vulnerables del esquema de respaldo en caliente. No obstante, es indispensable su respaldo, habida cuenta que ellos podrían almacenar las últimas actualizaciones a los datafiles pendientes por respaldar, dándole cierta cobertura en esta área.

No puede realizarse un resguardo de los archivos de control en línea, sino que deberá guardarse una copia especial de ellos[26]. Para realizar respaldos calientes, deberán resguardarse solamente los datafiles de un tablespace a la vez. Para reducir el tiempo que el tablespace está en modo de respaldo, es mejor copiar sus datafiles a un directorio de disco de respaldo[27]. Cuando se han hecho todas las copias, (se puede entonces copiar ese directorio de disco de respaldo para grabar mientras que la base de datos está funcionando normalmente). Se deben utilizar los comandos "begin backup" y "end backup", en la forma en que se muestra abajo; pues de no observarse lo anterior los respaldos serán corrompidos e inútiles.

[26] HOECHST, Tim. "*Guide to oracle*" New York; México: McGraw-Hill, c2004.

[27] ÁVALOS, Juan Jesús. "*Oracle for dummies*". UNAM Facultad de Contaduría y Administración, 2005 México.

Es necesario observar que los datos de los tablespaces restantes siguen disponibles para los demás usuarios, y cualquier transacción se puede realizar todavía en esa sección de la base. Para cada tablespace, se deberán ejecutar los pasos siguientes (el tablespace development es sólo un ejemplo – es necesario cambiar el nombre del tablespace y los nombres de los datafile apropiados para cada respaldo del tablespace):

SQL> alter tablespace development begin backup;
SQL> !cp -p /u03/oradata/PROD/devlPROD_*.dbf /u03/oradata/prod1

El siguiente paso consiste en copiar los datafiles de los tablespace a un directorio de disco de respaldo (/u03/oradata/prod1-- por ejemplo), con convenciones de nombramiento estándares.

SQL> alter tablespace development end backup;

Es necesario forzar un "log switch" de un redo log a fin de contener todas las transacciones en un solo archivo.

SQL> alter system switch logfile;

Otra opción consistiría en utilizar el comando **svrmgr "archived log next"**, el cual esperará hasta que el proceso de archive esté completo para el redo log actual, antes de regresar el control al DBA.

$ ls /u01/oradata/PROD/archivelogs/*.arc >logfile.lst

Genera la lista de los archivos del archivelog que se copiarán (no es correcto copiar el directorio completo debido a que puede existir un archivo del archivelog en medio de algún proceso).

$ sleep 10

Espera por algunos segundos para permitir al proceso archivelog terminar de escribir el archivo en turno.

$ cat logfile.lst |
> sed "s/\(.*\/\)\([^\/].*\)/cp -p \1\2 \/u03\/oradata\/prod1\/\2/" >logfile.shl
$ sh logfile.shl

Los comandos antes mencionados copian los archivos del archivelog a su directorio de disco de respaldo.

Es bueno generar un archivo de control de respaldo (binario y textual) después de que se hayan copiado el resto de los archivos:

SQL> alter database backup controlfile to
2> '/u03/oradata/prod1/controlfile.ctl';
SQL> alter database backup controlfile to trace;
$ ls -ltr /u00/oracle/admin/PROD/udump

La sentencia que se expresa a continuación tiene como objetivo encontrar el nombre del justo textual del archivo de control creado (el último en el listado del comando ls, tal como ora_16060.trc).

$ cp -p /u00/oracle/admin/PROD/udump/ora_16060.trc /u03/oradata/prod1

Ahora, deberá copiarse ese archivo de control al directorio de disco de respaldo.

Después de que todos los datafiles y archivos redo log se han copiado al directorio de disco de respaldo, una buena sugerencia es copiarlos a un dispositivo de almacenamiento no fijo como pudiera ser una cinta. Si se deseara comprimir esos archivos primero con una utilidad que utilizara el formato zip tal como gzip de GNU, este proceso deberá efectuarse antes del comando que copiara los archivos a la cinta (cpio), según lo demostrado abajo.

```
$ cd /u03/oradata/prod1
$ gzip -vN1 *
$ find /u03/oradata/prod1 | cpio -ovC64 >/dev/rmt0
```

Los respaldos calientes parciales, en los cuales los tablespaces se sostienen en las horas separadas, posiblemente a través de varios días para el sistema entero de tablespaces, son posibles, pero no es un procedimiento recomendado.

4.3.2 Otros procesos nocturnos

Además de los respaldos, pueden generarse procesos que prevengan cualquier pérdida de datos y para tener en cuenta la detección preactiva del problema y corrección, tal como se describe a continuación:

1. Crear y guardar las copias actuales de los archivos de control.
2. Realizar las exportaciones completas de la base de datos para las restauraciones rápidas de tablas críticas y comprobar la integridad de los datafile.
3. Generar las definiciones para todas las tablas y los índices.
4. Recopilar la estadística sobre los usos de los datafiles y el espacio en disco usado por los índices como monitoreo preactivo.

A continuación deberán respaldarse los archivos de control para cada una de las instancias de la base de datos, ello en prevención de que se corrompiera el archivo de control y se tuviera que reconstruir. Para cada una de las instancias de la base de datos (por ejemplo SEED, TRNG, PPRD, y PROD), es indispensable realizar el cambio de la instancia en uso (utilizando el siguiente comando "oraenv" e integre el nombre del la instancia, o utilice "export ORACLE_SID=SEED", de cambiar a la instancia SEED en Unix), y ejecutar el siguiente script en SQLplus o svrmgr para crear el archivo de control en el directorio user_dump_dest especificado en el archivo de init.ora:

SQL> alter database backup controlfile to trace;

Debe crearse un archivo export de la base de datos. Esto también detectará una probable corrupción del bloque en los datos de cualquier tabla, puesto que se realiza una exploración completa de la tabla (FULL SCAN). Las corrupciones del bloque no se detectan en copias físicas de éste durante los respaldos de cinta, por ello no podría conocerse la propagación de malos bloques a todos sus respaldos hasta que un usuario intente tener acceso a datos en ese bloque específico. Durante la exportación, si se encuentra un mal bloque de los datos, la exportación fallará.

**$ echo systempassword | exp system file=/u03/oradata/prod1.dmp **
> full=y log=/u03/oradata/prod1.dmp.log >/dummy 2>/dummy

Se considera importante la creación de un archivo índice relacionado a la exportación de las tablas y sus definiciones. Este archivo también puede ser útil para separar las tablas en diferentes tablespaces dependiendo del producto instalado.

```
$ echo systempassword | imp system file=/u03/oradata/prod1.dmp \
> indexfile=/u03/oradata/prod1.idx full=y \
> log=/u03/oradata/prod1.idx.log >/dummy 2>/dummy
```

La recopilación de la estadística sobre uso del espacio en disco permite saber cuándo reasignar a las tablas más extensas con la finalidad de evitar problemas relacionados a la falta de espacio libre, o para agregar datafiles a los tablespaces cuando estén a punto de llenarse. (alter tablespace.<>.add datafile.<>.), o para reconstruir los índices que tienen actualmente espacio inusitado o perdido debido a las transacciones sobre sus tablas asociadas.
Aquí se muestran algunos SQL scripts que resultan de utilidad para resolver potenciales problemas potenciales del espacio para las tablas y los tablespaces y para establecer si un índice tiene exceso de de espacio perdido y si debe ser reconstruido en su caso:

```
SQL> select segment_name,segment_type,extents,max_extents
2> from sys.dba_segments where extents + 5 > max_extents
3> or extents > 50 order by segment_name,segment_type desc;
SQL> col "Free" format 9,999,999,999
SQL> col "Tot Size" format 9,999,999,999
SQL> select tablespace_name,sum(bytes) "Tot Size"
2> from dba_data_files group by tablespace_name;
SQL> select tablespace_name,sum(bytes) "Free"
2> from dba_free_space group by tablespace_name;
SQL> validate index posnctl.nhrfinc_key_index;
SQL> select name,del_lf_rows_len from index_stats;
```

- Respaldo en cinta.

El mejor método para almacenar periódicamente los archivos redo log a una cinta es enviarlos al disco duro de respaldo y tener cronjob que se encargue de copiar dichos archivos a la cinta. Nunca deben copiarse directamente los archivos al dispositivo magnético de almacenamiento, ya que tal procedimiento requeriría un bobinador dedicado exclusivamente para los archivelogs, y además el respaldo necesitaría terminar antes de iniciar un nuevo ciclo (de otra manera, la base de datos detendrá su operación. Si se contara con un lector de cinta dedicado exclusivamente a los archivelogs, podrían ser respaldados con más con frecuencia que un procedimiento de respaldo normal, aproximadamente cada 10 minutos. En ese caso, deberán respaldarse solamente los archivos nuevos que no han sido colocados aún en la cinta.

4.4 Escenarios de recuperación de desastres.

Las técnicas de recuperación de Bases de Datos no son útiles a menos que se tenga pleno conocimiento de su empleo para restaurar la base de datos incluido el cambio más reciente realizado a la información. Oracle ofrece estas ventajas, en tanto se tenga el conocimiento de cómo utilizarlas. En esta sección serán descritas las capacidades de la recuperación del Oracle y cómo emplearlas para la recuperación general de datafiles, de los tablespaces, o de la base de datos en su totalidad, y para los panoramas específicos de la recuperación del desastre, que son variaciones en los temas de los procedimientos de recuperación generales.

4.4.1 Puntos a checar antes de iniciar una recuperación.

Los mensajes exhibidos durante arranque de la base de datos indicarán generalmente qué clase de problemas está experimentando. Después de que se obtenerse un mensaje de error durante su arranque inicial, muy probablemente se necesitará ejecutar una sentencia de shutdown antes de proceder con la recuperación de la base de datos. Los archivos de alerta se encontrarán en el directorio bdump de la base de datos (puede encontrarse esta ubicación usando "select value from v$parameter where name = 'background_dump_dest';" en SQLplus o "show parameter background_dump" desde el svrmgr), de lo cual se obtendrá algo como lo que se señala a continuación /u00/oracle/admin/PROD/bdump/alert_PROD.log.

Los procesos del Oracle pueden monitorearse desde shell de UNIX con el siguiente comando "ps - ef| grep ora ". Se obtendrán los siguientes procesos corriendo de la base de datos PROD: ora_smon_PROD, del ora_pmon_PROD, ora_dbwr_PROD, ora_lgwr_PROD, ora_arch_PROD, y de otros procesos que tendrán la siguiente nomenclatura ora_xxxx_PROD .

- ¿Qué es necesario restaurar?

Solamente deberá restaurarse el datafile que está dañado en el caso de que se esté recuperando hasta el tiempo actual. Todos los datafiles deberán ser restaurados si se está intentando recuperar hasta un momento anterior al actual, pero ello deberá suceder una vez que dichas datafiles hayan sido respaldados. Puesto que el DBMS Oracle requiere que todos los datafiles estén sincronizados con el mismo SCN (System Change Number), la recuperación de un solo datafile a un momento reciente tener un SCN diferente al resto de datafiles, que el DBMS no aceptaría.

En primer término, tienen que restaurarse todos los archive logs que sean necesarios en la recuperación (para el periodo de tiempo entre el respaldo y el momento de la caída), si no se encuentran en línea actualmente disponible o si se han sido corrompidos esos archivos en línea del archivelog.

Los los archivos de control no deben ser restaurados a menos que la totalidad de las copias se hayan perdido.

Tampoco deben ser restaurados los archivos redo log mediante un respaldo en caliente (De hecho, los archivos redo log solamente deben ser respaldados en frío). Restaurar un respaldo con un archivo que estaba recién generado causaría la corrupción durante la recuperación de base de datos. Si se borran o se corrompen los archivos redo log, sólo es necesario remover el o los archivos dañados y dar reset al sistema que los genera, lo cual se produce bajo las siguientes instrucciones:.

SVRMGR> connect internal
SVRMGR> startup mount
SVRMGR> alter database open resetlogs;

4.4.2 Panorama general de la recuperación de desastres.

La primera acción a realizar frente a una caída de la base de datos es la creación de un respaldo de los datafiles, archivos de control, archivos redo log y de los archivos con los parámetros de inicialización (init.ora) para la instancia que presente el problema, y copiarlos a un directorio ajeno al Oracle_Home de la respectiva instancia. No es aconsejable tratar de inicializar esta instancia sin haber realizado previamente este respaldo.

De otra forma puede experimentarse una situación aún peor. Si existen problemas durante el arranque de la instancia, la base de datos exhibirá un mensaje de error sobre el primer conflicto, que generalmente proporcionará una idea sobre qué tipo de panorama de la recuperación se necesitará aplicar. En la mayoría de los casos, se requerirá dar de baja la instancia después del primer intento de arranque para comenzar con la recuperación.

Existen tres opciones primarias de la recuperación para los escenarios de fallas de dispositivos de almacenamiento, las cuales se enlistan a continuación:

1) Recuperación de la base de datos
2) Recuperación de un Datafile
3) Recuperación de un Tablespace

Los restantes panoramas de recuperación son variaciones de los tres casos expuestos. Los pasos de la recuperación desde el punto de vista del escenario en específico que se trate, serán analizados más adelante, no obstante, una recuperación genérica puede definirse de la siguiente forma:

1. La base de datos es dada de baja o en su defecto deberá ponerse en estado fuera de línea el datafile o el tablespace dañado.
2. El datafile afectado se restaura desde el último respaldo con ayuda de los archivos redo log que fueron generados posteriormente a éste.
3. Se utilizan los comandos de restauración
4. Los archivos son puestos nuevamente en línea.

➢ Opción básica de recuperación de bases de datos Oracle.

La opción recover database se utiliza para recuperar todos los datafiles que requieran de este proceso, denominando a esta recuperación completa, o hasta un punto en el pasado antes de la falta, llamándolo en esta ocasión recuperación incompleta, la cual solamente está disponible bajo esta opción.

La recuperación de la base de datos se realiza en estado de "mount"; en este caso, la base de datos debe ser dada de baja antes de iniciar el proceso. Todos los datafiles deben estar en línea para ser recuperados; en contraste a las otras dos opciones donde se encuentran fuera de línea para la mayoría de los escenarios. Los pasos básicos son los siguientes:

1. Deberá obtenerse una copia de los datafiles dañados a partir del último respaldo que se tenga antes de la falla.
2. Montar la base de datos.
3. Los datafiles deberán ser puestos en línea.
4. Iniciar la recuperación.
5. Abrir la base de datos.

Este proceso puede obtenerse mediante los siguientes comandos.

$ cp -p /u03/oradata/prod1/devlPROD_1.dbf /u03/oradata/PROD

Obtenga una copia del datafile dañado mediante su respaldo, para este ejemplo el nombre del datafile será devlPROD_1.dbf.

SVRMGR> connect internal
SVRMGR> startup mount
SVRMGR> select * from v$datafile;

Muestra el estado de todos los datafiles, en la especie, cuáles están en línea y cuáles están fuera de linea.

SVRMGR> alter database datafile '/u03/oradata/PROD/devlPROD_1.dbf' online;

El status del datafile que se está remplazando de fuera de línea a en línea deberá ser cambiado.

SVRMGR> set autorecovery on

Con esta instrucción puede definirse que no sea necesario especificar los archivos redo que serán utilizados por la recuperación. La ubicación donde Oracle buscará los archivos redo log será log_archive_dest (debe estar habilitado el modo archive).

SVRMGR> recover database;

Recuperará todos los datafiles hasta el momento actual.

SVRMGR> alter database open;

Abrirá la base de datos a todos los usuarios.

La opción "Recover Database" puede también ser utilizada para realizar una recuperación incompleta hasta un cierto punto en el tiempo antes de que ocurriera la falla o antes de la última vez que se dio de baja la base de datos. Para hacer esto, todos los datafiles deben de ser restaurados desde el último respaldo que se tenga antes de iniciar la recuperación (no son necesarios los archivos redo log o archivos de control), y todos los datafiles se recuperan de nuevo al mismo punto en tiempo - no se pueden realizar recuperaciones incompletas parciales.

Existen tres opciones de recuperación incompleta disponibles:

- **Basada en tiempo.**
- **Basada en cancelación.**
- **Basada en cambio.**

La recuperación basada en tiempo recobra los datafiles hasta una fecha y una hora definidas por el usuario. La recuperación basada en cancelación implica una auto recuperación mediante archivos redo log y archived redo log hasta que se escriba la palabra "cancel" en el prompt; en consecuencia el modo "autorecovery on" no es apropiado para esta recuperación. La restauración basada en cambios aplicará los archivos redo log y los archived redo logs, pero no incluirá un SCN (System Change Number) especificado por el usuario. Los pasos básicos son similares a una recuperación completa de la base de datos, a excepción de que serán necesarios los siguientes comandos de recuperación dependiendo del escenario:

SVRMGR> recover automatic database until time '1998-07-24:15:45:00';

La opción "automatic" es similar al "set autorecovery on", aquí puede realizarse la recuperación hasta un segundo antes de la falla.

SVRMGR> recover database until cancel;

Iniciará la recuperación utilizando todos los archivos de restauración hasta que el DBA escriba cancel en el prompt.

SVRMGR> recover automatic database until change 378;

El SCN se podrá consultar en la tabla v$log_history en los campos low_change# y high_change# para cada uno de los archived logs, y en la tabla v$log para los archivos redo log en la columna first_change# (Se podrán aplicar cualesquiera de los archivos redo log a excepción del actual).

Después de una recuperación incompleta, debe abrirse la base de datos con la opción "resetlogs", según se muestra abajo, y una vez dada de baja la base de datos debe realizarse de forma inmediata un nuevo respaldo debido a que el último ya no será útil dado que el sistema de archivos redo log fue re-iniciado.

SVRMGR> alter database open resetlogs;

- Opción básica para la recuperación de un datafile..

Este tipo de restauración se utiliza para recuperar un archivo en específico hasta un punto en el tiempo antes de la falla, sincronizándolo con los otros datafiles. Este proceso se realiza también desde el estado de montaje (después de haber dado de baja la instancia), con el datafile en línea o fuera de ella. Un datafile dañado debe ponerse fuera de línea antes de arrancar la instancia. Debido a que el tablespace "SYSTEM" no puede ponerse fuera de línea, esta técnica no puede utilizarse con la base de datos arriba y con los datafiles en línea.

Los pasos básicos para la recuperación desde el estado de MONTAJE son:

1. Recupere el datafile dañado desde un respaldo anterior.
2. Inicie la base de datos hasta el punto de montaje y repare el datafile dañado.
3. Ponga en línea el datafile reparado.
4. Abrir la base de datos.

Los pasos antes expuestos se muestran de la siguiente manera.

$ cp -p /u03/oradata/prod1/devlPROD_1.dbf /u03/oradata/PROD
SVRMGR> connect internal
SVRMGR> startup mount
SVRMGR> set autorecovery on
SVRMGR> recover datafile '/u03/oradata/PROD/devlPROD_1.dbf';
SVRMGR> select * from v$datafile;
SVRMGR> alter database datafile '/u03/oradata/PROD/devlPROD_1.dbf' online;
SVRMGR> alter database open;

Los pasos básicos para recuperar un datafile cuando la base de datos está en operación (a excepción de uno dentro del tablespace SYSTEM) son:

1. Poner fuera de línea el datafile.
2. Obtener una copia en buen estado del datafile dañado desde otro respaldo.
3. Restaurar el datafile dentro desde la instancia.
4. Poner en línea a este datafile.

En razón a la sentencia anteriormente señalada:

```
SVRMGR> connect internal
SVRMGR> alter database datafile '/u03/oradata/PROD/devlPROD_1.dbf'
offline;
SVRMGR> !cp -p /u03/oradata/prod1/devlPROD_1.dbf /u03/oradata/PROD
SVRMGR> recover automatic datafile '/u03/oradata/PROD/devlPROD_1.dbf';
SVRMGR> alter database datafile '/u03/oradata/PROD/devlPROD_1.dbf'
online;
```

Para realizar una recuperación de un datafile cuando está en operación la base de datos (a excepción de uno dentro del tablespace SYSTEM) puede también realizarse con una variante: dar de alta la base de datos e inicie su operación mientras se tiene el datafile en reparación.

Los pasos básicos son:

1. Monte la base de datos.
2. Ponga fuera de línea el datafile.
3. Abra la base de datos e inicie la operación.
4. Recupere una copia en buen estado del datafile.
5. Restaure el datafile dentro de la instancia.
6. Ponga en línea el datafile.

Según se muestra abajo.

```
SVRMGR> connect internal
SVRMGR> startup mount
SVRMGR> alter database datafile '/u03/oradata/PROD/devlPROD_1.dbf'
offline;
SVRMGR> alter database open;
```

La base de datos está disponible para todos los usuarios a excepción de del datafile afectado.

SVRMGR> !cp -p /u03/oradata/prod1/devlPROD_1.dbf /u03/oradata/PROD
SVRMGR> recover automatic datafile '/u03/oradata/PROD/devlPROD_1.dbf';
SVRMGR> alter database datafile '/u03/oradata/PROD/devlPROD_1.dbf' online;

Se puede acceder a la información de tablas que se encuentren en un tablespace el cual tenga un datafile fuera de línea siempre y cuando los datos se encuentren en los demás datafiles del tablespace y el header de la tabla no esté en el datafile dañado.

➢ Opción básica de recuperación de un tablespace.

La opción de recuperación de un tablespace se utiliza para recuperar todos los datafiles que necesiten de este proceso en un tablespace hasta el punto en el tiempo antes de la falla, sincronizándolos con los otros datafiles. La recuperación se realiza con la base datos en operación y después de poner al tablespace fuera de línea (que provoca que todos sus datafiles se pongan fuera de línea). En razón a que el tablespace del sistema no puede ponerse fuera de línea, este procedimiento no puede utilizarse para ese caso en particular. Los pasos básicos en tanto la base de datos esté abierta son:

1. Poner fuera de línea el tablespace.
2. Restaurar los datafiles afectados desde un respaldo.
3. Recuperar el tablespace afectado.
4. Poner en línea el datafile reparado.

Este procedimiento se obtiene con la siguiente sentencia:

SVRMGR> connect internal
SVRMGR> alter tablespace development offline immediate;

Con lo anterior se obtendrá el inicio de una transacción de rollback para todas las operaciones relacionadas al tablespace que presenta el problema.

SVRMGR> !cp -p /u03/oradata/prod1/devlPROD* /u03/oradata/PROD
SVRMGR> recover automatic tablespace development;
SVRMGR> alter tablespace development online;

Realizar una recuperación de un tablespace después de poner en marcha a la base de datos podría servir para tener en operación el sistema y efectuar la recuperación de forma paralela. Podría efectuarse mediante los siguientes pasos:

1. Montar la base de datos.
2. Poner fuera de línea los datafiles afectados.
3. Iniciar la base de datos.
4. Poner fuera de línea el tablespace.
5. Restaurar los datafiles desde un respaldo.
6. Recuperar el tablespace y
7. Ponerlo en línea.

Como se expone abajo.

SVRMGR> connect internal
SVRMGR> startup mount
SVRMGR> alter database datafile '/u03/oradata/PROD/devlPROD_1.dbf' offline;
SVRMGR> alter database open;
SVRMGR> alter tablespace development offline;

SVRMGR> !cp -p /u03/oradata/prod1/devlPROD* /u03/oradata/PROD
SVRMGR> recover automatic tablespace development;
SVRMGR> alter tablespace development online;

Si un error de escritura ocurriera en uno de los datafiles cuando se ponga fuera de línea el datafile, puede utilizarse la siguiente instrucción "alter tablespace tsname offline temporary;", entonces se podría correr una recuperación para este datafile, pero si el problema fuera en todos los datafiles de un tablespace sería necesario emplear la instrucción "alter tablespace tsname offline immediate;". Ello podría correr la opción para el tablespace completo.

4.5 Escenarios prácticos de desastres.

Los siguientes puntos describen de una manera sencilla diferentes problemas que llegan a presentar las instancias Oracle, deberán definirse los síntomas y acciones inmediatas a tomar, para lograr la disponibilidad de la base de datos.

a) Espacio en disco no disponible en el que se almacenan los archivos redo log en modo archived.

Síntomas: Si el disco que contiene el directorio en el que se están escribiendo los archivos redolog en modo archived se saturara, Oracle detendrá todas las transacciones hasta en tanto el proceso "archiver" pueda continuar escribiendo en ese directorio. Mediante la ejecución de un comando para averiguar cuánto espacio disponible existe en disco (a manera de muestreo, en UNIX podría tratarse de un "df - k") se puede determinar el problema. Las sesiones de los usuarios actuales se detendrán, y los usuarios que intenten entrar obtendrán el siguiente mensaje: "ERROR: ORA-00257: archiver error.

Connect internal only, until freed", ello en razón a que el proceso del archiver todavía está esperando tener espacio para archivar sus registros. Para verificar esta conclusión pueden seguirse los pasos que se enlistan a continuación:

SVRMGR> connect internal
SVRMGR> select value from v$parameter where name = 'background_dump_dest';

Muestra el path en el que se están almacenando los archivos.

SVRMGR> !tail -200 /u00/oracle/admin/PROD/bdump/alert_PROD.log

Muestra "ORA-00272: error writing archive log", expresando que el grupo de archivos redo log no puede ser escrito.

SVRMGR> !ls -ltr /u00/oracle/admin/PROD/bdump/arch*

Obtiene el último archivo trace que fue generado en el directorio bdump, v.gr. arch_13106.trc (debe ser el último listado).

SVRMGR> !cat /u00/oracle/admin/PROD/bdump/arch_13106.trc

También muestra "ORA-00272: error writing archive log".

Acción: Se necesitará liberar espacio en ese volumen del disco para que Oracle pueda continuar operando, removiendo archivos de ese volumen, o, como una mejor opción, borrando archivos viejos que pudieran tenerse respaldados en una cinta. No se deben eliminar archivos redolog que no hayan sido respaldados, puesto que no se podrían recuperar transacciones de esos archivos si la base de datos fallara.

El siguiente script elimina los viejos archivos dependiendo de un número de días establecido. Si se respalda la base de datos cada noche, un día equivale al número más pequeño que se debe incorporar, o tres si no se realiza ningún procedimiento de respaldo los lunes. Se debe acceder al sistema como el usuario nativo de Oracle del sistema operativo o en su defecto como administrador del mismo.

```
# File: remove_old_logs.shl
echo "Es necesario correr este script como administrador o como el usuario del
sistema operativo del DBA,"
echo "Removiendo los archivos más viejos  a X días."
echo "Inserte el numero de días a borrar: \c"
read DAYS_KP; export DAYS_KP
find /u01/oradata/PROD/archivelogs -name '*.arc' -mtime +$DAYS_KP -exec rm
{} \;
   echo "Resultados después de liberar espacio:"
   du -k
   df -k
```

b) Pérdida de un archivo de control

Síntomas: No podría haber síntomas hasta que se intentara dar de alta o de baja la instancia de Oracle. En un caso en que los archivos de control fueran borrados mientras la base de datos estuviera fuera de operación, se mostraría un error "ORA-00210: cannot open control file '/u03/oradata/PROD/ctrl_PROD_01.ctl' o, si los archivos de control fueran sobrescritos, se conseguiría" ORA-00201: control file versión incompatible with ORACLE versión" junto con sus nombres. En el arranque para ambos casos, se conseguiría "ORA-00205: error in identifying control file '/u03/oradata/PROD/ctrl_PROD_01.ctl ', junto con "ORA-07366: sfifi: invalid file, file does not have valid header block" si está sobrescrito.

Acción: Si se tiene un archivo de control actualizado se puede ejecutar esta instrucción para regenerarlo "alter database backup controlfile to trace" en el subdirectorio bdump, es necesario editar el header y ejecutar el archivo.

SVRMGR> connect internal
SVRMGR> shutdown abort
SVRMGR> !ls -ltr /u00/oracle/admin/PROD/udump/*.trc

Obtiene el último archivo trace con el que se puede generar nuevamente un archivo de control, tal como ora_31494.trc.

SVRMGR> !vi /u00/oracle/admin/PROD/udump/ora_31494.trc

Este archivo deberá ser editado y eliminarse las líneas que se encuentren antes del STARTUP NOMOUNT line.

- **SVRMGR> @/u00/oracle/admin/PROD/udump/ora_31494.trc**

Si se carece de un archivo de control, es necesario restaurar todos los datafiles y los archivos de control pero no los archivos redo log, desde el último respaldo que se tenga disponible y realizar la recuperación a través de la sentencia "using backup controlfile":

SVRMGR> connect internal
SVRMGR> shutdown abort

En este punto se restaurarán todos los datafiles y los archivos de control desde el último backup utilizando los archivos redo log en modo archive hasta cierto punto en el tiempo pero no pueden utilizarse los archivos redo log.

SVRMGR> connect internal
SVRMGR> startup mount
SVRMGR> recover automatic database using backup controlfile;
SVRMGR> alter database open resetlogs;

Cierre la base de datos y haga de manera inmediata un respaldo, habida cuenta que el viejo respaldo ya no será útil una vez que se utilice la opción RESETLOGS.

c) Pérdida del datafile Temp.

Síntomas: Durante los ordenamientos largos (select distinct, order by, group by, union) que no se pueden poner en memoria , éste fallará con el error "ORA-01157: cannot identify data file 3 - file not found "si la falla sucedió a la mitad del ordenamiento, o"ORA-01116: error in opening database file 3" si la falla se produjo antes de que el ordenamiento iniciara, junto con el nombre del archivoORA-01110: data file 3: '/u03/oradata/PROD/tempPROD_1.dbf' ". No se establecerá nada relacionado a este problema en el archivo de alert.log, y por ello no se generará ningún archivo trace.

Acción: Poner fuera de línea el datafile, una vez realizado, deberá eliminarse el tablespace y volverlo a regenerar. Este proceso puede producirse con la base de datos en operación desde la utilería SQLplus o desde el comando svrmgrl en versiones de Oracle posteriores.

SQL> alter database datafile '/u03/oradata/PROD/tempPROD_1.dbf'
2> offline;
Si la base de datos sólo está montada, puede ejecutarse el comando svrmgr para abrirla y ponerla en ejecución.

SQL> select file_name,bytes/1024 kbytes,bytes/1024/1024 mbytes

2> from dba_data_files where file_name like '%temp%';

Muestra el tamaño del datafile TEMP en Kb y Mb.

SQL> select * from dba_tablespaces where tablespace_name = 'TEMP';

Muestra los parámetros del tablespace TEMP

SQL> drop tablespace temp;
SQL> create tablespace temp
2> datafile '/u03/oradata/PROD/tempPROD_1.dbf' size 80M
3> default storage (initial 262144 next 262144 minextents 1
4> maxextents 249 pctincrease 0);

Elimine y reconstruya el tablespace con los parámetros originales del tamaño del datafile, a menos que sea necesario ponerlo en alguna otra ubicación en la que fuera necesario cambiar los parámetros.

d) Datafile Temp fuera de línea.

Síntomas: Similar al punto anterior, a excepción del mensaje "ORA-00376: file 3 cannot be read at this time ", junto con el nombre del archivo.
Acción: Puede ponerse fuera de línea el datafile, eliminarlo y volverlo a regenerar, o puede hacerse una recuperación datafile con la base de datos abierta y poner en línea al datafile.

SVRMGR> recover automatic datafile '/u03/oradata/PROD/tempPROD_1.dbf';
SVRMGR> alter database datafile '/u03/oradata/PROD/tempPROD_1.dbf'
2> online;

e) Pérdida de un grupo de archivos redo log (los cuales han sido almacenados en modo archived)

Síntomas: La base de datos falla cuando se intenta realizar un acceso al grupo de archivos redo log que no se encuentra o que está dañado. Los usuarios que estén haciendo transacciones en la base serán expulsados de la instancia mostrándoles el error "ORA-01092: ORACLE instance terminated. Disconnection forced", y los usuarios que intenten entrar se les mostrará el mensaje "ERROR: ORA-03114: not connected to ORACLE"y "ERROR: ORA-00472: PMON process terminated with error". No se establecerá ningún cambio en el archivo de alert.log; sin embargo, habrá archivos que serán generados en el directorio dump por los procesos pmon, lgwr y dbwr. Estos archivos deben ser verificados desde el sistema operativo debido a que la base de datos no está arriba.

$ grep background_dump_dest /u00/oracle/product/v723/dbs/initPROD.ora

Muestra la ruta del directorio dump.

$ cd /u00/oracle/admin/PROD/bdump
$ ls -ltr *.trc

Obtiene los últimos archivos trace generados por los procesos lgwr y dbwr.

$ cat pmon_13612.trc

Muestra "ORA-00470: LGWR process terminated with error" (also in dbwr).

$ cat lgwr_32306.trc

Muestra "ORA-00313: open failed for members of log group 3 of thread 1" and messages containing the missing log file names like "ORA-00312: online log 3 thread 1: '/u03/oradata/PROD/logPROD_3b.dbf'".

Acción: Se necesitará arrancar la base de datos y ponerla únicamente hasta el punto de montaje, eliminar al grupo que falta del archivos redo log (y eliminar a todos los miembros del grupo si es que no han sido eliminados previamente), y agregar a un grupo de archivos redo log nuevo, una vez que se haya realizado este proceso la base de datos puede ponerse en operación.

SVRMGR> connect internal
SVRMGR> startup

Muestra "ORA-01081: cannot start already-running ORACLE - shut it down first".

SVRMGR> startup force

O bien podría iniciarse a través de un "shutdown abort" seguido de un "startup". Mostrará "Database mounted." Y el error "ORA-00313: open failed for members of log group 3 of thread 1" con un adicional mensaje acerca de los archivos de bitácora que fueron generados.

SVRMGR> select bytes/1024 from v$log where group# = 3;

Muestra el tamaño en Kbytes de los archivos faltantes.

SVRMGR> alter database drop logfile group 3;
SVRMGR> alter database add logfile group 3
2> ('/u03/oradata/PROD/logPROD_3.dbf',
3> '/u03/oradata/PROD/logPROD_3b.dbf') size 500K;

Creará un nuevo grupo de archivos redo log , con el tamaño antes definido.

SVRMGR> alter database open;

Si sólo un miembro del grupo de archivos redo log fallara, otro miembro tomará el control y permitirá que el DBMS siga funcionando sin mayor problema evidente. No obstante alert_PROD.log mostrará errores entre otros "ORA-00313: open failed for members of log group 3 of thread 1" y "ORA-00312: online log 3 thread 1: '/u03/oradata/PROD/logPROD_3.dbf'" para cada uno de los cambios de grupo de archivos redo log que se efectúen. En este caso, elimine y reconstruya al miembro:

SVRMGR> select * from v$log;

Si el estado de este "Active" para el grupo es necesario realizar un "alter system switch logfile;"

SVRMGR> alter database drop logfile member
2> '/u03/oradata/PROD/logPROD_3.dbf';
SVRMGR> alter database add logfile member
2> '/u03/oradata/PROD/logPROD_3.dbf' to group 3;

f) Pérdida del grupo actual de los archivos redo log (los cuales necesitan ser salvados en modo archived)

Pérdida del grupo actual de los archivos redo log (los cuales necesitan ser salvados en modo archived)

Síntomas: El comportamiento es similar al caso donde no existe suficiente espacio en disco duro para seguir almacenando los archivos redo log en modo archived; sin embargo, haciendo un "df - k" (en el caso de un sistema Unix) demuestra que hay espacio suficiente para seguir escribiendo los archivos archivelogs. Las sesiones de los usuarios actuales estarán congeladas, y a los usuarios que intenten entrar se les mostrará el siguiente mensaje "ERROR: ORA-00257: archiver error. Connect internal only, until freed", puesto que el proceso que escribe los archivos redo log en modo archived a disco todavía está esperando para archivar este grupo de archivos redo log debido a que Oracle entra en un ciclo colisionando todos los archivos redo log que se encuentren hasta que regrese a verificar al último, en este caso el que continúa abierto. Para comprobar esta conclusión es necesario realizar el siguiente procedimiento.

SVRMGR> connect internal
SVRMGR> select value from v$parameter where name = 'background_dump_dest';

Muestra la ruta del directorio dump.

SVRMGR> !tail -200 /u00/oracle/admin/PROD/bdump/alert_PROD.log

Mostrará el siguiente error "ORA-00286: No members available, or no member contains valid data", indicando que los miembros pertenecientes al grupo de archivos redo log están corruptos o no se encuentran en la ubicación donde Oracle los está buscando, también puede observarse el siguiente mensaje "ORACLE Instance PROD - Can not allocate log, archival required", "Thread 1 cannot allocate new log, sequence 21", y "All online logs needed archiving".

SVRMGR> !ls -ltr /u00/oracle/admin/PROD/bdump/arch*

Obtiene el ultimo archivo de bitácora que fue generado tal como podría ser arch_22882.trc en el directorio dump.

SVRMGR> !cat /u00/oracle/admin/PROD/bdump/arch_22882.trc

También mostrará "ORA-00286: No members available, or no member contains valid data".

SVRMGR> shutdown abort
SVRMGR> startup

Mostrará "ORA-00313: open failed for members of log group 2 of thread 1". En su defecto si se ha tratado de eliminar y regenerar el grupo de archivos redo log desde un punto de montaje se podría obtener el siguiente mensaje:

SVRMGR> alter database drop logfile group 2;

En consecuencia se observará "ORA-00350: log 2 of thread 1 needs to be archived".

Acción: Esto requiere una recuperación incompleta hasta un momento antes en que Oracle quiso utilizar este grupo de archivos redo log., puesto que el Oracle no puede continuar trabajando sin generar un archivo de este tipo de forma correcta, y puesto que no se puede eliminar y reconstruir el grupo de archivos redo log mientras el proceso archiver tiene marcado este grupo de archivos. Todos los cambios pendientes se perderán. Una recuperación incompleta basada en tiempo se expone de la siguiente manera:

SVRMGR> connect internal
SVRMGR> shutdown abort

En este momento todos los datafiles deberán ser restaurados desde el último backup, los archivos redo log que sean necesarios y los archivos redo log, pero no los archivos de control.

SVRMGR> startup mount
SVRMGR> select group#, sequence#, bytes, first_change#, first_time, status from v$log;

Es necesario que se obtengan los cambios y sus respectivas fechas realizados por cada uno de los grupos de archivos redo log para poder generar la recuperación a partir del momento previo a la falla. Y estos valores se incluirán en la siguiente instrucción.

SVRMGR> recover automatic database until time '1998-11-16:23:52:04';
SVRMGR> alter database open resetlogs;

Inmediatamente después cierre la operación de la base de datos y realice un respaldo ya que el antiguo no funcionará debido a la opción RESETLOGS que se aplicó[28].

[28] PEPIN, David. *Ob.cit.*

4.7 Recovery Manager (Respaldo y recuperación de bases de datos)

Para ejemplificar la forma de respaldar y recuperar bases de datos mediante el uso de RMAN a continuación se definirán dos instancias a las que se denominarán PROD y RCAT.

Es importante establecer si un catálogo de recuperación será utilizado o si en su defecto, se empleará el archivo de control de la base de datos que será respaldada. Por lo regular lo recomendable es utilizar un catálogo. En grandes sistemas el rendimiento con un catálogo es bastante bueno. Además ante la eventualidad de perder el archivo de control, la recuperación requeriría de arduo trabajo. La estrategia consiste en generar el catálogo dentro de una instancia ajena a la que se requiere respaldar, exclusivamente para procesos de respaldo y recuperación.

4.7.1 Creación de usuario y catálogo de recuperación.

Como primer paso deberá generarse un tablespace que se vendrá a significar en el catálogo del RMAN definido bajo el nombre de CATALOG

```
svrmgrl> create tablespace CATALOG
datafile '/u01/oracle/rcat/cat1rcat.dbf
size 20M;
```

Se generará el usuario RMAN con el tablespace previamente creado, como su Tablespace primario.

svrmgrl> create user rman identified by rman
temporary tablespace TEMP
default tablespace CATALOG quota unlimited on
CATALOG;

Y por último se otorgará el grant de recovery_catalog_owner al usuario RMAN.

svrmgrl> grant recovery_catalog_owner to rman;

En este momento el usuario RMAN está listo para generar el catálogo de recuperación; ahora, debido que RMAN es una herramienta relativamente nueva, se mostrará cómo crear un catálogo desde una base de datos 8 o una 8i en adelante (cabe destacar que RMAN no puede ser soportado en bases de datos 7 o inferiores).

El procedimiento para generar el catálogo desde una base de datos 8 será:

svrmgrl> connect rman/rman@rcat

svrmgrl> @?/rdbms/admin/catrman.SQL

Correr este script tomará algunos minutos. Una vez que concluya el catálogo, contendrá toda la información necesaria para lograr la recuperación. No obstante, la creación del catálogo es diferente desde una base de datos 9i:

En el caso del sistema operativa Unix, es necesario definir la variable de ambiente con la instancia Oracle con la que se trabajará.

% set ORACLE_SID=RCAT

% rman catalog rman/rman

Se accesa al sistema de RMAN con el tablespace definido anteriormente como objetivo.

RMAN> create catalog;

Esto generará el esquema del catálogo de recuperación en el tablespace por default. Ya que se tiene un catálogo de recuperación, es necesario obtener la información del archivo de control de la base de datos con la que se va a trabajar. Se puede checar la vista v$controlfile_record_section para determinar la información que se encuentra almacenada en el archivo de control.

El siguiente paso a realizar implicará conectar al RMAN con la base de datos que se requiere respaldar.

% rman TARGET / RCVCAT rman/rman@rcat

Esta es la cadena de conexión de RMAN, donde se registran dos accesos, el primero, para acceder al sistema de RMAN con la base de datos que se pretende restaurar y el segundo, a la instancia donde se encuentra el catálogo de recuperación.

Recovery Manager: Release 8.0.5.2.0 - Production
RMAN-06005: connected to target database: PROD
RMAN-06008: connected to recovery catalog database

En seguida deberá registrarse la información en el catálogo de recuperación, una vez que toda la información haya sido capturada.

RMAN> register database;

Al completar el catálogo es necesario verificar que se encuentre completo, para realizar este proceso, es necesario ejecutar las siguientes sentencias.

RMAN> list incarnation of database;
RMAN-03022: compiling command: list
RMAN-06240: List of Database Incarnations
RMAN-06241: DB Key Inc Key DB Name DB ID CUR Reset SCN Reset Time
RMAN-06242: ------- ------- -------- ---------- ----- --------- ----------
RMAN-06243: 1 2 PROD 3351020544 YES 1 10-AUG-04

Con ello estará confirmado que el proceso de generación del catálogo ha sido exitoso.

4.7.2 Realizando respaldos de bases de datos con RMAN

➢ Copia de un archivo de control usando RMAN.

Cuando RMAN necesita sincronizarse con el archivo de control actual, generará una copia fiel temporal de éste, al que denominará "port-specific". Es necesario utilizar el comando "set snapshot controlfile name " para cambiar el nombre del archivo temporal que se generó.

Por ejemplo:

set snapshot controlfile name to '/oracle/dba/prod/temp_prod.ctl';

➢ Respaldos en frío con RMAN

Una vez configurado el ambiente, se pueden desarrollar respaldos de la base de datos. Este primer ejemplo muestra cómo realizar un respaldo de base de datos en frío utilizando RMAN.

% set ORACLE_SID=PROD

En primer término, se define la instancia con la que se trabajará, en segundo lugar se baja la instancia Oracle y se deja nuevamente en el estado mount.

svrmgrl> connect internal
svrmgrl> shutdown immediate;
svrmgrl> startup mount

Ahora desde RMAN se corre el script con los parámetros previamente expuestos:

```
% rman target / rcvcat rman/rman@rcat
run {
allocate channel c1 type disk format
'/backup/u00/oracle/prod/df_%d_%p_%c';
backup (database);
SQL 'alter database open';
release channel c1;
}
```

A fin de verificar que el proceso de respaldo se haya completado con éxito es necesario inspeccionar de nueva cuenta el catálogo.

```
RMAN> list backupset of database;
RMAN-03022: compiling command: list
RMAN-03025: performing implicit partial resync of recovery catalog
RMAN-03023: executing command: partial resync
RMAN-08003: starting partial resync of recovery catalog
RMAN-08005: partial resync complete
RMAN-06230: List of Datafile Backups
RMAN-06231: Key File Type LV Completion_time Ckp SCN Ckp Time
RMAN-06232: ------- ---- ------ ---- ---------- ---------- ----------
RMAN-06233: 1386 5 Full 21-OCT-99 468068 21-OCT-99
```

En este caso sí fue exitoso.

➢ Respaldos en caliente

Para generar un respaldo en caliente, es necesario que la base de datos se encuentre en modo archive. Para verificar si el modo archive está activado es necesario ejecutar los siguientes comandos.

set ORACLE_SID=PROD
svrmgrl> connect internal
svrmgrl> archive log list
Database log mode Archive Mode
Automatic archival Enabled
Archive destination '/u01/oracle/prod/archive'
Oldest online log sequence 151
Next log sequence to archive 152
Current log sequence 152

Como puede observarse, el modo archived está activo. Además pueden advertirse parámetros como el archivo actual, el directorio donde se almacenan los archivos, entre otros.

Ahora se realizará el respaldo en caliente de la base de datos y de los archivos redo log generados.

```
% set ORACLE_SID=PROD
% rman target / rcvcat rman/rman@rcat
RMAN> run {
allocate channel c1 type disk;
format '/backup/u00/oracle/prod/df_%d_%s_%p';
backup (database include current controlfile);
release channel c1;
allocate channel c1 type disk format
'/backup/u00/oracle/prod/al_%d_%s_%p';
backup (archivelog all delete input);
release channel c1;
}
```

- Respaldando archivos redo log en modo archive

Otra alternativa consiste en realizar el respaldo de los archivos redo log bajo el modo archived usando el RMAN y borrándolos simultáneamente para evitar que se sature el directorio de trabajo de este servicio. El script para lograrlo se enlista a continuación:

```
Run {
set command id 'RMAN';
allocate channel t1 type 'sbt_tape';
allocate channel t2 type 'sbt_tape';
backup
filesperset 20
format 'al_%t_%s_%p'

archivelog all
delete input;
release channel t1;
}
```

Como nota importante, es necesario remarcar que puede monitorearse el avance de cualquier proceso que esté ejecutando RMAN con el siguiente script:

```
select sid, serial#, context,
round(sofar/totalwork*100,2) "% Complete",
substr(to_char(sysdate,'yymmdd hh24:mi:ss'),1,15) "Tiempo Actual"
from v$session_longops
where compnam = 'dbms_backup_restore'; -- para 9i
where substr(opname,1,4)='RMAN'; -- para 10g
```

Esto producirá una salida como la que se muestra a continuación.

```
SID SERIAL# CONTEXT % Complete Tiempo Actual
---------- ---------- ---------- --------------- -----------------
12 56 980408 14:21:07
```

Si el proceso no tiene un incremento cada dos minutos es necesario verificar los archivos de alertas y las bitácoras porque probablemente exista algun inconveniente. Para descartar problemas relacionados con la instancia puede correrse el siguiente query:

select * from v$session_wait where wait_time = 0;

En consecuencia, se mostrará si existen sesiones de Oracle obstruidas esperando una llamada sbt , que son las que usa RMAN.

Existen varios estados en los que la instancia se puede encontrar para iniciar una recuperación.

1. Si se intenta recuperar un archivo de control, debe iniciarse la instancia con "startup nomount"
2. Si el caso es el de los datafiles del tablespace system, se iniciará como "startup mount"
3. Sin embargo, si se requiere restaurar un tablespace o un datafile, la base de datos puede estar en operación, pero los objetos afectados deben estar fuera de línea.

➢ Recuperación de archivos de control.

Este es un ejemplo de un script que restaura el archivo de control previamente respaldado.

```
% set ORACLE_SID=PROD
svrmgrl> connect internal
svrmgrl> shutdown abort;
svrmgrl> startup nomount;
```

Se pone la base de datos en un estado "nomount" y se inicia paralelamente el RMAN para restaurar la base de datos.

```
rman target / rcvcat rman/rman@rcat
run {
allocate channel t1 type 'sbt_tape';
allocate channel d1 type disk;
restore controlfile to 'd:\target\ctl1targ.ora';
replicate controlfile from 'd:\target\ctl1targ.ora';
restore database;
SQL "alter database mount";
recover database;
SQL "alter database open";
}
```

- Recuperación de tablespace de sistema.

Para restaurar el tablespace de sistema se utiliza el siguiente script, bajo la misma lógica de poner fuera de línea la base y paralelamente ejecutar el script de RMAN:

```
% set ORACLE_SID=PROD
svrmgrl> connect internal
svrmgrl> shutdown immediate;
svrmgrl> startup mount;
```

```
rman target / rcvcat rman/rman@rcat
run {
allocate channel d1 type disk;
restore tablespace SYSTEM;
recover database;
SQL "alter database open";
}
```

- Sincronización del catálogo de recuperación.

Como se ha expuesto, el elemento más importante del RMAN es el catálogo de recuperación, que consiste en un repositorio de todos los objetos de la base de datos. No obstante, es importante realizar una sincronización del catálogo con frecuencia para evitar pérdida de información o de estructura. Existen dos tipos de sincronización, en una sincronización parcial el catálogo será actualizado dependiendo únicamente del archivo de control, pero no actualizará ningún otro objeto (datafiles, tablespaces, segmentos de rollback, etc.). En contraste, la restauración completa verificará cada uno de los objetos para actualizarlos.

Para sincronizar el archivo de control se puede correr el siguiente script:

```
set ORACLE_SID=target_db

rman target / rcvcat rman/rman@rcat
rman> resync catalog;
RMAN-03022: compiling command: resync
RMAN-03023: executing command: resync
RMAN-08002: starting full resync of recovery catalog
RMAN-08004: full resync complete
```

La sincronización puede ser realizada por un proceso que se ejecute a determinada hora durante la noche (En Unix podría ser un cronjob) y con esto se evitaría tener alguna discrepancia en la información.

Conclusiones

El manejo oportuno y eficaz de la información viene a significarse en el elemento más importante dentro de cualquier compañía u organización. Ese poder brinda la factibilidad de definir estrategias, decisiones y soluciones a una infinidad de problemas. No obstante lo avanzado de la tecnología, la posibilidad de que sobrevenga una falla siempre se encuentra presente. Es por ello que deviene en impostergable la necesidad de contar con una copia confiable de dicha información, ante cualquier situación anormal capaz de detener la consulta o actualización de la base de datos, con el consabido impacto perjudicial a la negociación.

A lo largo del presente estudio, se han expuesto las diferentes técnicas de respaldo y de recuperación que deben tenerse en consideración durante el empleo del DBMS Oracle, software de presencia actual en todos los rubros laborales de influencia significativa en los sectores público y privado. Es crucial remarcar en este punto, la importancia subrayada de establecer qué tipo de estrategia deberá seguirse para mantener *siempre* la información disponible y consistente a partir del parámetro fijado por las necesidades del servicio; ya se trate del número de peticiones que reciba la base de datos por minuto, o si se tiene un horario de operación improrrogable, o bien sea un sistema de operación crítica interrumpida durante las veinticuatro horas del día.

Las técnicas expuestas a lo largo de esta tesis no alcanzarían su objetivo sin mantener una estrategia capaz de lograr que la pérdida de información sea realmente mínima y que el tiempo de la caída no se prolongue más allá de un límite aceptable para la organización, durante el cual no se ocasione un impacto negativo.

Ahora bien, a pesar de tener una efectiva estrategia de respaldo de bases de datos, es importante no soslayar aquellos escenarios de desastres básicos que pudieran llegar a presentarse. Por ello, es indispensable la realización de simulacros de desastres, lo cual conlleva a evaluar las acciones necesarias ante las probables eventualidades.

La actual tendencia que sigue Oracle para el respaldo y recuperación de sus bases de datos es la orientada a objetos: cualquier elemento que exista en ella, desde una tabla, una secuencia o un trigger hasta un datafile o un archivo de control serán manejados de la misma forma. En esa tesitura, la utilería RMAN será en un futuro no muy lejano, el estándar para este tipo de actividades que implicará una base de datos capaz de almacenar todos los objetos que sea necesario respaldar.

Éste es un concepto con el que Oracle ha dejado entrever que en sus bases de datos pueden almacenarse cualesquiera tipos de información, inclusive los objetos de otras bases de datos. En ese orden de ideas, las arquitecturas consistentes en mantener bases de datos en *stand by* o espejos dejan de tener funcionalidad habida cuenta que ahora es más fácil recuperar un objeto desde un simple *select,* a realizar una importación desde una instancia ajena u obtener cualquier *update* desde algún medio de almacenamiento.

El primer plan de estudios de MAC del cual orgullosamente soy egresado, impartía materias que en su momento fueron muy útiles para el desarrollo de este trabajo entre las que destacaron *"Estructura de datos"*, "*Arquitectura de computadoras*", "*Bases de Datos*", y "*Sistemas Operativos*". El plan actual fue enriquecido con temas de gran trascendencia como "*Ingeniería de Software*", *Redes de cómputo*" y "*Bases de Datos Distribuidas*", tópicos para los que se estima que este documento podría aportar referencias convenientes para los alumnos de la carrera de Matemáticas Aplicadas y Computación así como para los operadores de la base de datos del Centro de Cómputo de la FES Acatlán.

Entre los razonamientos que de manera natural surgieron como corolarios del presente trabajo se encuentran los siguientes:

- Derivado de la formación de la carrera, surge para los alumnos la necesidad de enfocar los problemas a través de una solución matemática mediante la optimización de recursos u otros medios, utilizando la computadora como una herramienta potencial de trabajo.
- La sistematización de cualquier tarea es uno de los diversos retos a los que se enfrenta un egresado de la carrera de Matemáticas Aplicadas y Computación; durante el desarrollo de este proceso, el universitario se percata gradualmente de los beneficios adicionales que se están obteniendo y se descubren –y admiten– las razones de impartir materias cuya naturaleza no es absolutamente matemática.
- Resulta por demás interesante observar y comprender cómo se ven involucradas las matemáticas de forma aplicada en la arquitectura y el funcionamiento de la computadora.
- Finalmente, como en todos los ámbitos del ejercicio profesional, es altamente satisfactorio enfrentarse a un problema surgido en el ambiente laboral y contar con los elementos aportados por nuestra Alma Máter para resolverlo con efectividad.

Índice por Temas.

Bibliografía

Bisland, Ralph B.
Database management : Developing application systems using
Oracle / Ralph B. Bisland Jr.
Englewood cliffs, New Jersey : Prentice Hall, c1989.
QA76.9D3 P455

Perry, James T.
Understanding Oracle / James T. Perry, Joseph G. Lateer
San Francisco : Sybex, c1989.
QA76.9D3 P455

Thro, Ellen
The Database Dictionary : Dbase, r: Base, Oracle, ingres,
db2, foxbase, clipper, SQL / Ellen thro
San Marcos, California : Microtrend, c1990
QA76.9D3 T47

Perry, James T.
Understanding Oracle / James t. Perry, Joseph G. Lateer
San Francisco : Sybex, c1989.
QA76.9.O73 P47

Rolland, F. D.
Relational Database Management with Oracle / F. D. Rolland
Wokingham, England : Addison-wesley, 1989
QA76.9D3 R66

Jaramillo López, Judith.
Técnicas para la recuperación de información y búsqueda de texto en bases de datos relacionales
México, UNAM, 2001

Sayles, Jonathan S.
How to use oracle SQL *plus / Jonathan s. sayles
Wellesley, Massachusetts : Qed information sciences, 1991
QA76.73S67 S37

Cronin, Daniel J.
Mastering Oracle : Featuring oracle's SQL standard / Daniel
Carmel, Indiana : Hayden, c1989
QA76.9O73 C76

Koch, George
Oracle : The complete reference / George koch ; [foreword by Lawrence J. Ellison]
Berkeley; México: Osborne McGraw-Hill, c1990
QA76.9D3 K63

Crooks, Ted
Using oracle / Ted crooks
Carmel, indiana : Que, c1991
QA76.99D3 C76

Pepin, David
Oracle : Programmer's guide / David pepin
Carmel, indiana : Que, c1989
QA76.9D3 P435

Rodgers, Ulka
Oracle : A data base developer's guide / Ulka Rodgers
Englewood cliffs, New Jersey : Yourdon, c1991
QA76.9D26 R63

Hoechst, Tim
Guide to oracle / Tim hoechst, Nicole Melander, Christopher Chabris
New York ; México : McGraw-Hill, c1990
QA76.9D3 H61

Inmon, William H.
Using oracle to build decision support systems / W. h. inmon
Wellesley : Qed information sciences, c1990
QA76.9D3 I48

Webb, Kenneth 1950-
Oracle distributed systems : A c programmer's development guide / Kenneth Webb and Lori Lafreniere
Blue Ridge Summit, Pensylvania : Windcrest, c1991
QA76.73S67 W43

Krohn, Mike
Using the oracle toolset / Mike krohn
Wokingham, England : Addison-Wesley, 1992c1993
QA76.76A65 K76

Sturner, Gunther
Oracle 7 : A user's and developer's guide, including release 7.1 / Gunther Sturner ; tr. by George Staw

New York : Van Nostrand Reinhold, 1995
QA76.9D3 S79713

Smine, Hatem
Oracle : arquitectura, administración y optimización /
Hatem Smine ; tr. Victor Martín Garcia
Madrid : Díaz de Santos, 1992
QA76.9D36 S5618

Kolste, Bruce
Oracle Power Objects Handbook / Bruce Kolste and David Petersen
Berkeley, California ; México City : Osborne Mc Graw-Hill,1999
QA76.9D36 K65

Lockman, David
Developing Personal Oracle7 applications / David Lockman
Indianapolis, Indiana : Sams, c1995.
QA76.9D3 L63 1995

Greenwald, Rick
Mastering oracle power objetcs / Rik Greenwald and Robert Hoskin
Sebastopol, California : O'Reilly, c1997
QA76.9D3 G726

Oracle RDBMS errors messages and codes manual : version 8.0
Redwood, City, California : Oracle, 1999
QA76.9D367 O734

Oracle RDBMS utilities user's guide : version 8.0
Redwood, City, California : Oracle, 1992
QA76.9D367 O738

Oracle RDBMS database administrator's guide : version 7.0
Redwood, City, California : Oracle, 1992
QA76.9D367 O733

Corey, Michael J.
Oracle / Michael J. Corey, Michael Abbey, Daniel J. Dechichio ; tr. Jose Pieltain Alvarez Arenas
Madrid ; México : McGraw-Hill, c1995
QA76.9D3 C6718

Burleson, Donald Keith
High-performance oracle database applications / Donald K. Burleson

Scottdale, Arizona : Coriolis, c1996
QA76.9D3 B86

Smine, Hatem
Oracle : arquitectura, administración y optimización /
Hatem Smine ; tr. Víctor Martín García
Madrid : Díaz de Santos, 1992
QA76.9D36 S5618

Kolste, Bruce
Oracle power objects handbook / Bruce Kolste and David
Petersen
Berkeley, California ; México City : Osborne Mc Graw-Hill,
QA76.9D36 K65

Lockman, David
Developing Personal Oracle7 applications / David Lockman
Indianapolis, Indiana : Sams, c1995.
QA76.9D3 L63 1995

Greenwald, Rick
Mastering oracle power objetcs / Rik Greenwald and Robert
Hoskin
Sebastopol, California : O'Reilly, c1997
QA76.9D3 G726

Documentos Electrónicos

Oracle Corporation [en línea]:Physical Backup and Recovery: An Insider's Perspective. 2003 [fecha de consulta: 4 Abril 2003].Disponible en:
< *http://www.metalink.oracle.com/metalink/plSQL/ml2_gui.startup*>.

Oracle Corporation [en línea]:Archiver Best Practices. 2003 [fecha de consulta: 5 Septiembre 2002].Disponible en:
< *http://www.metalink.oracle.com/metalink/plSQL/ml2_gui.startup*>.

Oracle Corporation [en línea]:Official Backup Policy -- Certification, RMAN, EBU, Third-Party Software. 2004 [fecha de consulta: 22 Octubre 2002].Disponible en:
< *http://www.metalink.oracle.com/metalink/plSQL/ml2_gui.startup*>.

Oracle Corporation [en línea]:How to Recover from a Lost Datafile with Different Scenarios. 2000 [fecha de consulta: 22 Octubre 2002].Disponible en:
< *http://www.metalink.oracle.com/metalink/plSQL/ml2_gui.startup*>.

Oracle Corporation [en línea]:Recover database after disk loss. 2003 [fecha de consulta: 22 enero 2003].Disponible en:
< *http://www.metalink.oracle.com/metalink/plSQL/ml2_gui.startup*>.

Oracle Corporation [en línea]:Sample Hot Backup Script for Unix. 2001 [fecha de consulta: 27 enero 2003].Disponible en:
< *http://www.metalink.oracle.com/metalink/plSQL/ml2_gui.startup*>.

Oracle Corporation [en línea]:Script To Identify Files Needed For Hot Backups. 2002 [fecha de consulta: 27 enero 2003].Disponible en:
< *http://www.metalink.oracle.com/metalink/plSQL/ml2_gui.startup*>.