Universidad Autónoma de Querétaro

Facultad de Informática

Auditoría de Bases de Datos Oracle

TESINA

Que para obtener el título de

LICENCIADO EN INFORMÁTICA

Presenta

Maricela Mejía Rivera

Dirigida por: ISC. Jabel Reséndiz González

Santiago de Querétaro, Qro. Septiembre 2002.

CARTA DE ACEPTACIÓN

Por este medio, se otorga constancia de aceptación de tesina para obtener el título de Licenciado en Informática, que presenta la pasante **MARICELA MEJÍA RIVERA** con el tema denominado "*Auditoria de Base de Datos Oracle*".

Este trabajo fue desarrollado como una investigación derivada del curso de titulación **"ADMINISTRACIÓN DE BASE DE DATOS",** dando cumplimiento a uno de los requisitos contemplados en el artículo 34 del reglamento de titulación vigente, en lo referente a la opción de titulación por realización y aprobación de cursos de actualización.

Se extiende la presente para los fines legales a que haya lugar y para su inclusión en todos los ejemplares impresos de la tesina, a los diecisiete días del mes de mayo del dos mil dos.

ATENTAMENTE

ING. JABEL RESÉNDIZ GONZÁLEZ
PROFR. CURSO DE TITULACIÓN

INDICE

1. INTRODUCCIÓN

En la actualidad la mayor parte de la información de una organización se concentra en Bases de Datos, esto es por la facilidad que representa este formato para el manejo de la misma, dado que las Bases de Datos representan un modo de organización de la información muy sencillo y además homogéneo y óptimo; los encargados de los sistemas en las empresas han optado por esta opción de manejo de la información.

Las bases de datos, por su estructura, son una herramienta excelente para la organización de la información en sistemas, se compone de TABLAS, RELACIONES, ATRIBUTOS e INDICES; que permiten conservar la integridad de la información por módulos, lo que significa que ante un posible error, la localización y solución del mismo será más fácil.

En la actualidad, ORACLE es uno de los manejadores de datos mayormente demandados debido a su capacidad de soporte de sistemas robustos, lo cual no disminuye su eficiencia.

Sabemos que no basta con un buen sistema de Bases de Datos para que todo se lleve a cabo correctamente, como por arte de magia. El trabajo de un buen Administrador es el complemento para un excelente trabajo.

El administrador de Bases de Datos será el encargado del diseño, control y mantenimiento de la misma, por lo que debe tener un conocimiento previo y preciso de cada una de las actividades que realizan los empleados y cuya información será depositada en la Base de Datos. Por lo anterior, se requiere que el administrador de Bases de Datos esté constantemente en relación con ellos. La información con la que se alimente la Base de Datos debe provenir de cada uno de los departamentos de la organización de manera precisa, pues esto, se reflejará en las salidas (reportes) del sistema de Bases de Datos.

Para el caso de ORACLE, el administrador debe también tener conocimientos relacionados con: Diccionario de datos caché, árbol semántico de las sentencias SQL, las sentencias SQL y PL/SQL. Ya que, será responsable de la instalación y actualización de Oracle y sus productos asociados, ajuste de la Base de Datos para obtener el rendimiento óptimo de la misma, estrategias de seguridad y recuperación.

Para un mayor entendimiento del tema, en este trabajo hablaremos desde la composición de una Base de Datos hasta su administración y mantenimiento y, desde luego, de la Auditoría, que es el tópico principal que nos atañe, como una herramienta de control de las Bases de Datos Oracle.

También veremos algunos ejemplos que nos aclaren el funcionamiento de los comandos de auditoría de este sistema de Bases de Datos.

La primordial tarea de este trabajo, como ya lo mencioné, es el estudio de la Auditoría de las Bases de Datos Oracle, ya que me parece importante y de mucha utilidad que en la actualidad

se cuente con sistemas tan potentes para el manejo de la información, pero que de nada servirían sin una adecuada administración, control y mantenimiento. En el caso de ORACLE se nos permite manejar el modulo de auditoría del diccionario de datos que además de detectar posibles fallas en la información, nos ayuda a prevenir y corregir sin la necesidad de deshabilitar el sistema por mucho tiempo, en ocasiones, nisiquiera hay que salir de éste o hacer que los usuarios suspendan sus tareas.

Una constante auditoría sobre el Sistema de Bases de Datos, nos permitirá contar siempre con información íntegra, esto es de suma importancia para las organizaciones ya que de ésta dependen todas sus labores y sobre todo la toma de decisiones.

Por lo anterior, me pareció de suma importancia resaltar mediante este trabajo las capacidades del sistema manejador de Bases de Datos Oracle y combinarlas con la aplicación de un debido control mediante la Auditoría, que nos permita emitir a las partes interesadas de una organización la información necesaria, siempre íntegra y siempre confiable.

1.1. ADMINISTRACION DE BASES DE DATOS ORACLE

1.1.1. ANTECEDENTES HISTORICOS

En sus comienzos, Oracle era principalmente una empresa de bases de datos relacionales, las cuales eran una nueva forma de pensar sobre como deberían estructurarse y almacenarse los datos; la clave de este nuevo pensamiento consiste en entender las relaciones existentes entre los datos y en estructurar la base de Información para que refleje dichas relaciones. El objetivo de una base de datos relacional consiste en construir una estructura en la cual las modificaciones requeridas no la afecten a ella, sino únicamente a los datos, es decir, se minimicen las modificaciones a las aplicaciones, se termine con la redundancia de los datos y se garantice la sincronización de los cambios hechos a los mismos. "Estos cambios solo deben afectar una tabla, y no varios archivos como frecuentemente sucedía cuando se manejaba el enfoque tradicional".

1.1.2. RESPONSABILIDADES DE UN ADMINISTRADOR DE BASES DE DATOS (DBA)

El DBA contribuye con su trabajo al funcionamiento eficaz de todos los sistemas que se ejecutan con la base de datos Oracle; además ofrece asistencia técnica a quienes interactúan con la Base de Datos y se espera que tenga soltura en todos los aspectos técnicos que surjan con el software de Oracle. Dentro de sus responsabilidades están:

- Instalación y actualización del Oracle y de todos sus productos asociados
- Asignación de recursos para la utilización de Oracle: memoria, espacio en disco, perfiles de usuario etc.
- Ajuste de la base de datos para conseguir el rendimiento optimo.
- Enlace con el servicio mundial de asistencia al cliente de Oracle (Oracle Wordwide Support) para resolver problemas técnicos que requieran la intervención de Oracle.
- Estrategias de copia de seguridad y recuperación.
- Colaboración con el personal de administración del sistema y desarrolladores de aplicaciones.

El administrado de BD oracle debe tener un conocimiento suficiente de los siguientes términos: (obviamente su manejo y administración)

Shared Pool

Es una porción de la SGA, creada al subir la Instancia Oracle; La cual contiene:

- El Diccionario de Datos Cache.
- Las sentencias SQL y PL/SQL.
- El árbol semántico de cada sentencia SQL.
- El plan de ejecución de cada sentencia SQL.
- Estas son guardadas para que las sentencias sean requeridas en más de una ocasión o por más de un usuario.

Database Buffer Cache
Es un área de la SGA que guarda copias de los bloques de datos más recientemente leídos del disco, esto se hace para un mejor desempeño pues si los datos son de nuevo requeridos por un usuario, su acceso es más rápido.
Los bloques pueden contener datos modificados que no son permanentemente escritos a disco y los cuales maneja Oracle de una manera consistente para atender la concurrencia de los usuarios conectados a la base de datos, dichos usuarios comparten el acceso a esta área.

Redo Log Buffer
Es un buffer en el cual se registran secuencialmente todos los cambios hechos a los datos (sentencias DML, commits, rollbacks).
Es usado para reconstruir los cambios hechos a la Base de Datos y a los Segmentos de Rollback cuando ocurre una falla y se necesita hacer recuperación de datos.
Su uso se puede omitir con la opción UNRECOVERABLE en sentencias create table, create index y en sql*loader.

Diccionario de Datos Cache
Es una colección de tablas y vistas que contienen información referente a la base de datos, sus estructuras y sus usuarios. Esta información incluye:

- los nombres de todas las tablas y vistas de la Base de Datos.
- los nombres y los tipos de datos de las columnas de las tablas.
- los privilegios de todos los usuarios.

Memoria Oracle (SGA)
Su tamaño está determinado por los parámetros:
Shared_Pool_Size= Tamaño en bytes del área para SQL compartidos y sentencias PL/SQL.
Db_Block_Size = Tamaño en bytes de un solo bloque de datos.
Db_Block_Buffers = Numero de Buffers a localizar en memoria.
Log_Buffer = Numero de bytes localizados para para los Redo Log Buffer.

SGA = Shared_Pool_Size + (Db_Block_Size * Db_Block_Buffers) +
Log_Buffer.

1.1.3. COMPOSICION DE UNA BASE DE DATOS ORACLE

DATAFILES
Archivos físicos que contienen toda la información de la base de datos; en ellos están estructuras tales como tablas e índices.

REDO LOG FILES
Archivos físicos que almacenan el registro de todos los cambios hechos a la base de datos, son utilizados principalmente para procesos de recuperación y almacena la información proveniente de los Redo Log Buffers.

SEGMENTOS DE ROLLBACK

Son una parte de la base de datos, la cual Oracle utiliza para una actividad que lleva cabo y que consiste en poder restablecer los datos al estado en que estaban antes de que un usuario empezara a modificarlos.

En estos segmentos se almacena una imagen de como eran los datos antes de realizar una transacción para mantener la consistencia de los mismos en operaciones no grabadas y que requieran de un proceso de anulación.

ALERT FILE

Archivo en el cual se registran cronológicamente:

- Los mensajes y errores producidos por Oracle.
- Operaciones administrativas como sentencias DDL,
- STARTUP, SHUTDOWN, ARCHIVE LOG y RECOVER.
- Los parámetros suministrados al subir la instancia.

Oracle usa este archivo para facilitar la labor de administración en el momento de solucionar problemas.

La ubicación de este archivo esta determinada por el parámetro BACKGROUND_DUMP_DEST (del archivo de parámetros).

Procesos Background

- Database Writer (DBWR)
- Log Writer (LGWR)
- Checkpoint (CKPT)
- System Monitor (SMON)
- Process Monitor (PMON)
- Archiver (ARCH)
- Recoverer (RECO)
- Lock (LCKn)
- Snapshot Refresh (Snnn)
- Shared Server (Snnn)
- Dispatcher (Dnnn)
- Parallel Query (Pnnn)

SMON System Monitor

Recuperaciones automáticas de la instancia.

Libera el espacio de segmentos temporales en memoria o en disco (sorts y join de tablas (tablespace temp)).

Efectúa el trabajo de defragmentación en los datafiles (hace contiguas las áreas de espacio libre).

LGWR Log Writer

Escribe las transacciones que se encuentran en los Redo Log a Disco cuando ocurre un commit, cuando se llena la tercera parte de los Redo Log. Esta operación permite que Oracle pueda recuperarse frente a varios tipos de fallos y únicamente existe uno por instancia.

DBWR Database Writer
Es un proceso obligatorio que maneja el Database Buffer Cache para que los procesos de servidor siempre encuentren buffers libres, dicho de otra manera escribe los bloques de datos modificados (en memoria) en los archivos de la base de datos (datafiles) utilizando el algoritmo LRU (menos recientemente utilizados).
Es uno de los dos únicos procesos que tienen permitido escribir en los archivos de datos que componen la base de datos Oracle. En ciertos sistemas operativos se pueden tener varios escritores de bases de datos por motivos de rendimiento.

1.1.4. PROCESOS OPCIONALES

ARCH Archiver
Es un proceso opcional, encargado de copiar el contenido de los archivos de REDO LOG a cinta o a disco para hacer recuperaciones en caso de fallas. Únicamente es necesario cuando la base de datos se encuentra en modo ARCHIVELOG.

CKPT Checkpoint
Asegura que todos los datos modificados en memoria (database buffers) sean escritos a disco. Oracle produce un punto de comprobación al conmutar entre los distintos registros que hay en memoria para que las transacciones sean consistentes entre los diferentes usuarios, además escribe en disco toda la información que los usuarios han modificado en memoria y notifica al archivo de control el registro de la transacción.

LCKn Lock
Es un proceso opcional, configurado para manejar los bloqueos entre bases de datos Oracle cuando estas se encuentran en distintas computadoras y compartiendo el mismo conjunto de discos (es decir en modo servidor en paralelo).

RECO Recoverer
Este proceso solo se observa cuando la base de datos ejecuta la opción distribuida de Oracle. La transacción distribuida es una en la que dos o más emplazamientos de datos debe mantenerse sincronizados, Por ejemplo cuando se tiene una copia de los datos en diferentes ciudades y por fallas en una línea telefónica se pierde una transacción en la mitad de su actualización. El proceso recuperador entonces resuelve las transacciones que hayan quedado inconsistentes en las dos ciudades.

1.2. CREACION DE UNA BASE DE DATOS

1.2.1. OPCIONES AL CREAR UNA BASE DE DATOS ORACLE

1.2.1.1. ARCHIVO DE PARAMETROS
Es un archivo de texto que contiene una lista de los parámetros de configuración de la instancia (Memoria y procesos Background utilizados por Oracle).
Oracle para poder subir la instancia, debe leer el archivo de parámetros initSID.ora, en donde SID es el nombre de la base de datos; estos parámetros son determinados por el administrador de la base de datos al crearla o antes de subir una instancia, y con ellos se pueden determinar aspectos como el tamaño de la memoria asignada a Oracle, el tamaño de cada bloque en el cual se almacenarán datos etc.

1.2.1.2. MANIPULACION DE LA BASE DE DATOS

Startup nomount pfile=initprueba.ora (sube la instancia de la base de datos prueba).
Alter database prueba open (Permite a todos los usuarios accesar la base de datos).
Alter database prueba mount (monta la base de datos para mantenimiento).
Alter database mount exclusive (es el default y solo permite a la actual instancia accesar la base de datos).
Estando conectado a la base de datos como usuario Internal se puede alterar el estado de la base de datos asi:
Alter system enable restricted session (Para futuras conexiones solo permite conectar usuarios que posean ese privilegio).
Alter system disable restricted session (Permite que todos los usuarios se conecten a la base de datos).

1.2.1.3. BAJANDO LA INSTANCIA ORACLE

Conectarse a la base de datos como internal o como un usuario con privilegios suficientes para bajar la instancia (puede ser desde una utilidad como Svrmgrl, OEM o Sql*dba).

Shutdown (si no hay usuarios conectados baja la instancia y cierra la base de datos, de lo contrario su función consiste en no dejar conectar ningún usuario y esperar a que los que estén conectados salgan o se maten sus tareas.
Existen dos opciones que varían esta opción: BAJANDO LA INSTANCIA ORACLE
Shutdown Immediate: Las sentencias que están siendo procesadas por los usuarios no son terminadas completamente, aquellas transacciones que no han sido grabadas (con commit) son reversadas y el servidor Oracle no espera a que los usuarios actualmente conectados a la base de datos se desconecten, sino que cierra y desmonta la base de datos y baja la instancia.

Shutdown Abort: El servidor Oracle no reversa las transacciones que no han sido grabadas y no espera que los usuarios se desconecten de la base de datos, tampoco cierra ni desmonta la base de datos, su trabajo consiste en bajar la instancia (procesos Oracle y memoria). Cuando se requiera volverla a subirla, Oracle por consistencia en sus procesos procede a hacer una recuperación automática de la instancia.

1.3. OBJETOS DE LA BASE DE DATOS

Oracle utiliza para su funcionamiento muchas estructuras con las cuales un administrador de base de datos debe familiarizarse, ellas se denominan objetos y cada uno de ellos tiene una función específica o trabajo que realizar, y de su buen funcionamiento depende el óptimo desempeño de la Base de Datos. En los ejemplos presentados posteriormente para la creación, borrado o manipulación de registros o estructuras se debe saber que Oracle maneja Esquemas; un esquema es una forma de referirse a un Objeto que ha sido creado por otro usuario y al cual puedo tener o no los privilegios para manipularlo.
Al necesitar manipular un registro de una tabla o un objeto creado por otro usuario y no tener los privilegios necesarios, debo anteponer el nombre del usuario creador del objeto seguido de un punto y del nombre del objeto "en la instrucción SQL". Ej, para seleccionar todos los registros de la tabla cliente debería escribir "SELECT * from VENTAS.cliente" en donde VENTAS es un usuario creado en la base de datos y el cual es el propietario de la tabla cliente.

1.3.1. TABLA: Es la unidad básica de almacenamiento en un sistema de bases de datos relacionales, en ellas son almacenados los datos de los usuarios y los datos del sistema Oracle; Cada tabla se compone de varias columnas las cuales cuentan con un tipo de datos asociado. La información sobre la estructura de todas las tablas se encuentra en el diccionario de datos y Oracle la utiliza para su funcionamiento.

Creación de una tabla:

```
CREATE TABLE cliente (
k_cliente NUMBER(3) NOT NULL,
n_cliente VARCHAR2(40) NOT NULL,
r_vendedor DATE,
PCTFREE 10
PCTUSED 65
STORAGE (
INITIAL 4M
NEXT 3M
PCTINCREASE 0
MINEXTENTS 2
MAXEXTENTS 20))
TABLESPACE USERS;
```

Modificación de una tabla

```
1. ALTER TABLE cliente
ADD ( d_dirección VARCHAR2(25),
n_teléfono NUMBER(10))
2. ALTER TABLE cliente
MODIFY ( n_cliente VARCHAR2(50))
3. ALTER TABLE cliente
MODIFY ( d_dirección VARCHAR2(50) NOT NULL)
4. ALTER TABLE cliente
STORAGE (PCTINCREASE 100
MAXEXTENTS 50)
```

1. Adicionando los campos d_dirección y n_teléfono a la tabla
2. Ampliando el tamaño del campo n_cliente
3. Agregando un constraint que impide la entrada de valores nulos en el campo d_dirección
4. Alterando los parámetros de almacenamiento de la tabla.

Borrando una tabla

```
1. DROP TABLE cliente
2. DROP TABLE cliente CASCADE CONSTRAINTS
3. TRUNCATE TABLE cliente
```

1. Borra la tabla y su estructura si no existen constraints de integridad referencial
2. Borra la tabla y los CONSTRAINTS de integridad referencial hacia la tabla.
3. Borra los datos de la tabla si no existen constraints de integridad referencial.

1.3.2. VISTA : Es una consulta SQL de una o varias tablas, la cual se encuentra almacenada en la base de datos y cuyos resultados se devuelven al usuario igual que los de una consulta a una tabla. A diferencia de una tabla, una vista no contiene datos sino únicamente una consulta SQL. Son útiles en seguridad, pues se pueden crear por ejemplo para restringir el acceso a ciertos campos de una tabla (es decir a los usuarios no se les da acceso a toda la tabla, sino a los campos contenidos en la vista), también se utilizan para facilitar a los usuarios los (join) complejos entre tablas y la escritura de nombres largos o difíciles de las tablas o de sus columnas en las consultas creadas por ellos.

Creación de una Vista

```
1. CREATE VIEW Vista_Cliente as select k_cliente, n_descripción
from cliente.
2. CREATE O REPLACE VIEW Vista_Cliente as select k_cliente,
n_descripción from cliente.
```

INDICE: Así como el índice de un libro ayuda a accesar su contenido de una manera más ágil, un índice de una tabla le ayuda a la base de datos a recuperar información con mayor velocidad. Un índice es una copia en miniatura de una tabla con información sobre la(s) columna(s) que forman parte del índice, y no sobre todas las columnas de la tabla. De esta manera le proporcionan a Oracle un veloz acceso a los datos pues no necesitan subir a memoria los registros completos de la tabla sino únicamente la(s) columna(s) indexada(s) para proceder a realizar la búsqueda requerida, ello conlleva a que en la memoria se puedan cargar muchas mas columnas de registros deseados en lugar de registros completos no deseados.

Creando un Indice

```
1.     CREATE     UNIQUE     INDEX     ind_ven_r_vendedor     ON
CLIENTE(R_VENDEDOR)
TABLESPACE users
STORAGE (INITIAL 200K NEXT 100K PCTINCREASE 75%).
2.     CREATE     BITMAP     INDEX     ind_ven_r_vendedor     ON
CLIENTE(R_VENDEDOR).
```

1. UNIQUE especifica que el valor de la columna en la tabla a ser indexada es UNICO.
2. BITMAP es un tipo de índice utilizado para tablas con millones de registros o en columnas con baja cardinalidad, es decir, aquellas en las cuales el numero de valores distintos es pequeño por ejemplo el sexo "F/M" o el estado civil.

Si no se especifica tablespace, Oracle crea el índice en el tablespace default del usuario que crea el índice. Se aconseja crear el índice en un tablespace diferente al de datos y de ser posible en diferente disco para un mejor desempeño de la base de datos.

1.3.3. SINONIMO : Es un nombre alternativo que se crea para un objeto de la base de datos; es normalmente utilizado para las tablas y las vistas de Oracle. Los sinónimos se crean normalmente para ocultar el propietario, la ubicación o el nombre real de una tabla (así otros usuarios la pueden accesar sin importar quien la haya creado o en donde se encuentre), también es utilizado para proporcionar a los usuarios nombres de objetos menos complicados que los reales.

Creación de un sinónimo

```
1. CREATE PUBLIC SYNONYM cliente FOR ventas.cliente.
2. CREATE SYNONYM cliente FOR ventas.cliente@BASE2
```

1. Crea un sinónimo llamado cliente para todos los demás usuarios de la base de datos, aunque ellos solo podrán hacer actualizaciones al mismo cuando se les otorguen privilegios tales como select, update, insert y delete.
2. Crea un sinónimo llamado cliente de un objeto que se encuentra en otra base de datos llamada BASE2.

1.3.4. ROLE: Es un objeto creado para simplificar el manejo de los privilegios en la Base de Datos cuando existen muchas tablas y muchos usuarios que las accesan. Consisten en agrupar una serie de privilegios en un objeto llamado rol, para que posteriormente este objeto sea otorgado diferentes usuarios o a otros roles. La racionalización se da debido a que hay usuarios que necesitan los mismos privilegios que otros y bastaría con asignar tales privilegios al mismo rol y este a su vez a cada usuario en lugar de tener que asignar individualmente los privilegios por usuario); de la misma manera, para eliminar un privilegio a estos usuarios solo necesitaría eliminarla del role y automáticamente lo perderían.

Algunas propiedades de los roles son:
* Una vez creados no tienen dueño.
* Pueden ser asignados a algún usuario de la base de datos o a otro rol.
* Pueden ser habilitados o deshabilitados por un usuario que tenga permisos.
* Pueden requerir autorización (password) para ser habilitados en determinada aplicación.
* Deben ser creados con cierto criterio de empresa, por aplicaciones, por cargos.
* Pueden ser habilitados desde SQL*PLUS, PL/SQL, lenguajes de tercera generación.

Creación de un rol

```
1. CREATE ROLE rol_ventas.
2. CREATE ROLE rol_ventas identified by xxxxxx.
```

Crea un rol llamado rol_ventas.
Crea un rol llamado rol_ventas y el usuario debe digitar el password xxxxxx para habilitarlo.

1.3.5. GRANTS (PRIVILEGIOS): Son otorgados por los dueños de los objetos y permiten a otros usuarios trabajar con sus datos. Algunos son:
* Select: permite que otros usuarios pueden examinar el contenido de tablas o vistas que no fueron creadas por ellos.
* insert permite a quien lo posee la creación de registros en las tablas de otros usuarios.
* update permite que otros usuarios puedan modificar o cambiar datos en tablas que no son de su propiedad.
* delete permite que otros usuarios puedan eliminar registros en tablas que no hayan sido creadas por ellos.

Asignación de Privilegios

```
1. GRANT select, insert, update ON cliente TO rol_ventas.
2. GRANT rol_ventas TO Maricela
3. GRANT rol_ventas TO Maricela WITH ADMIN OPTION
```

1. Asigna privilegios de consulta inserción y actualización a rol_ventas.
2. Otorga todos los privilegios dados a rol_ventas a un usuario de la base de datos llamado Maricela.
3. Hace los mismo que el anterior pero adicionalmente permite que el usuario Maricela pueda conceder privilegios de rol ventas a otros usuarios de manera transitiva.

1. Creación Usuarios

El Superusuario de Oracle es system

```
sql> connect system/password
sql> create user Mary identified by Mary99
solo se puede conectar a al bd
sql> connect Mary/Mary99
sql> connected
sql> connect system/password
sql> Grant dba to Mary (maxima permiso)
Grant resource to Mary (puede crear tabla paces, tablas)
sql>
```

2. Guía de Auditoría para el Administrador de Oracle

2.1.Auditoría de la Base de Datos vía sistema operativo

El diccionario de los datos de cada base de datos tiene una tabla llamada SYS.AUD$, normalmente se refiere a los medio de auditoría de base de datos.

La Auditoría de la base de datos vía sistema operativo puede guardar todos los archivos de la auditoría generados como el resultado de declaración, privilegio, u objeto auditado.

Su sistema operativo puede o no puede apoyarse de la base de datos que audita al proceso de auditoría de sistema operativo. Si esta opción está disponible, considere las ventajas y desventajas de usar la base de datos o sistema operativo que audita el proceso para guardar los archivos de auditoría de base de datos.

Ventajas del uso de la Auditoria de Bases de Datos

- Se pueden observar porciones de algún proceso de Auditoría predefinido o pertenecientes al diccionario de datos.
- Se pueden utilizar las herramienta de Oracle (Oracle Reports) para generar informes de la auditoría.
- Alternamente, se puede llevar a cabo el proceso de auditoría vía Sistema Operativo y evaluar los archivos fuente de múltiples aplicaciones, incluso Oracle.

Por consiguiente, la actividad del sistema examinador podría ser más eficaz porque todos los archivos de la auditoría están en un mismo lugar.

El sistema operativo también puede contener un proceso de la auditoría que guarda los archivos de la auditoría generado por el sistema operativo que audita fácilmente. Sin embargo, esta facilidad se debe a que Oracle trabaja sobre un sistema operativo dependiente

2.2.Guardando la Información Auditada

Aunque auditar es relativamente barato, se debe limitar el número de eventos auditados tanto como sea posible. Esto minimizará el impacto de la actuación en la ejecución de declaraciones que se auditan, y minimiza el tamaño del proceso de la auditoría.

Use las siguientes guías generales al inventar una estrategia de auditoría:

Evalúe su propósito de auditar.
Después de que usted tiene una comprensión clara de las razones por las cuales considera que se debe de auditar, usted puede crear una estrategia apropiada para auditar y puede evitar el auditar innecesariamente.

Por ejemplo, suponga que usted está auditando para investigar una actividad sospechosa de la base de datos.
Esta información no es bastante específica. ¿Qué tipo de actividad de la base de datos es sospechosa?

El propósito real de la auditoría podría ser que una persona no autorizada para modificar las tablas de la base de datos realizó acciones arbitrarias.
Este tipo de acción se consideraría acreedora a una auditoría.

Audite inteligentemente

Audite el número mínimo de declaraciones, usuarios u objetos necesarios para conseguir la información requerida. Esto significa desechar la información innecesaria o insignificante para la auditoría y darle valor a aquella verdaderamente relevante en la misión dentro del espacio consumido por el tablespace del SISTEMA. Equilibre su necesidad de recoger la información con la seguridad suficiente, con habilidad y procurando guardarla y procesarla adecuadamente.

Por ejemplo, si usted está auditando para recoger la información sobre la actividad de la base de datos, determine qué tipos de actividades usted está rastreando exactamente, intervenga sólo las actividades de interés, y sólo utilice la cantidad de tiempo necesario para recoger la información que usted desea. No intervenga los objetos si usted sólo está interesado en la información de I/O lógica de cada sesión.

Auditando las actividades sospechosas de la Base de datos.

Cuando usted audita para supervisar actividades sospechosas de la base de datos, use las pautas siguientes:

· Audite de manera general, posteiormente específicamente.
Al empezar a auditar las actividades sospechosas de la base de datos, es común que no mucha información esté disponible a usuarios específicos asignados u objetos del esquema. Por consiguiente, deben ponerse las opciones de la auditoría generalmente al principio. Una vez que la información de la auditoría preliminar se guarda y se analiza, con ello podemos pasar a opciones más específicas. Este proceso debe continuar hasta que exista suficiente evidencia para hacer las conclusiones concretas sobre el origen de la actividad sospechosa de la base de datos.

· Protega el proceso de la auditoría.
Al auditar la actividad sospechosa de la base de datos, es necesario poner mucho cuidado (proteger) al proceso de la auditoría misma para evitar que la información de la auditoría no pueda agregarse, pueda cambiarse, o pueda anularse; a menos que deba intervenirse.

Auditando la Base de datos en Actividad Normal

Cuando su propósito por auditar es recoger la información histórica sobre las actividades de la base de datos particulares, use las pautas siguientes:

· Audite sólo acciones pertinentes.
Evitar desordenar la información significante con los archivos de la auditoría inútiles y reducir la cantidad de administración del proceso de auditoría, sólo intervenga las actividades objetivo de base de datos.

· Respalde los archivos de la auditoría y deseche los del proceso de la auditoría.
Después de que usted ha coleccionado la información requerida, archive la información de interés para la auditoría. Una vez realizado este paso deseche la información depositada en el sistema para evitar modificaciones en la BD.

2.3.Creación de vistas para la Auditoría

Creando y Anulando Vistas para le proceso de Auditoría de las Base de datos

Esta sección describe cómo crear y anular las vistas para la auditoría de la base de datos, e incluye los temas siguientes:

· Creación de las Vistas para Proceso de Auditoría
· Anulación de las Vistas para Proceso de Auditoría

El proceso de auditoría de base de datos (SYS.AUD $) se utiliza para una sola tabla que exista en el diccionario de datos de cada base de datos de Oracle.
Para que la información de la auditoría a esa tabla sea significante se utilizan varias vistas predefinidas.
Estas vista se crean de manera exclusiva para la auditoría, y pueden ser eliminadas una vez que la auditoría termina o bien, si se decide no tomar en cuenta.

Creando las Vistas para el Proceso de Auditoría

Si usted decide usar este mecanismo para auditar, cree las vistas de auditoría conectando como SYS y ejecutando el script CATAUDIT.SQL. Este script crea las vistas siguientes:

· STMT_AUDIT_OPTION_MAP
· AUDIT_ACTIONS
· ALL_DEF_AUDIT_OPTS
· DBA_STMT_AUDIT_OPTS
· USER_OBJ_AUDIT_OPTS, DBA_OBJ_AUDIT_OPTS,
· USER_AUDIT_TRAIL, DBA_AUDIT_TRAIL,
· USER_AUDIT_SESSION, DBA_AUDIT_SESSION,
· USER_AUDIT_STATEMENT, DBA_AUDIT_STATEMENT,
· USER_AUDIT_OBJECT, DBA_AUDIT_OBJECT,
· DBA_AUDIT_EXISTS
· USER_AUDIT_SESSION, DBA_AUDIT_SESSION,
· USER_TAB_AUDIT_OPTS

2.4.Borrando las vistas de la Auditoría

Si usted a finalizado la auditoría y no necesita las vistas de proceso de auditoría, anúlelas conectándose a la base de datos como SYS y ejecutando el script CATNOAUD.SQL. El nombre y situación del script CATNOAUD.SQL residen en el sistema operativo dependiente.

2.5.Manejo de la información de la Auditoría

Esta sección describe varios aspectos para manejar la información de proceso de auditoría, e incluye los temas siguientes:

- Eventos auditados por default
- Selección de opciones de Auditoría
- Habilitando y deshabilitando la BD Auditada
- Controlando el crecimiento y tamaño del Proceso de Auditoría
- Protección del Proceso de Auditoría

Dependiendo de los eventos a auditar y las opciones seleccionada para auditar, los archivos de auditoría pueden contener diferente información. En dicha información se incluye siempre cada registro generado por el proceso de auditoría, por lo que puede ser significativa para alguna acción de la auditoría en particular, y puede ser:

- El nombre del usuario
- Identificación de sesión
- Identificador de la terminal o central de trabajo
- Nombre del objeto accesado
- Actividad realizada o que intentó realizar
- Código de la actividad realizada
- Fecha y tiempo

Algunos archivos generados por el sistema operativo en el proceso de Auditoría contienen codificaciones no legibles (encriptadas). Estos pueden descifrarse como sigue:

El Código de la acción

Esto describela acción que se realizó o intentó. La tabla AUDIT_ACTIONS del diccionario de datos contiene una lista de estos códigos y sus descripciones.

Los privilegios Utilizados

Esto describe quién y con qué privilegio del sistema realiza la acción. La tabla de SYSTEM_PRIVILEGE_MAP lista todos estos códigos, y sus descripciones.

El Código de ejecución

Esto describe el resultado de la actividad realizada o intentada. Las actividades exitosas devuelven un valor de cero, mientras las actividades infructuosas devuelven el código de error de Oracle que describe por qué la actividad era infructuosa.

2.6.Eventos auditados por default

Sin importar si la BD está habilitada o no. El servidor de Oracle siempre ejecutará ciertas acciones sobre ella, en conjunto con le sistema operativo. Estos eventos incluyen a lo siguiente:

Tabla 2.6.1 Acciones de un servidor sobre una Base de Datos Oracle

instancia startup	Uno de los registros que genera la auditoría es el detalle del usuario del OS que empieza la acción, el identificador de la terminal, la fecha y tiempo, y no importa si la base de datos auditada se encuentra habilitada o se desactivó. Esto se ejecuta en el OS mediante la auditoría, aún cuando el resultado de la auditoría de la base de datos no está disponible hasta después de que el startup se ha completado con éxito. Una vez que se obtuvo la información necesaria para la auditoría se debe reiniciar o volver al estado anterior la BD para evitar que el administrador realice acciones no apropiadas.
instancia shutdown	Este registro de auditoría genera los detalles del usuario en el SO cuando cierra su sesión de trabajo, tales como: el identificador de la terminal de trabajo, la fecha y la hora.
Conexión a la BD con privilegios de administrador.	Este registro de auditoría genera los detalles del usuario en el SO cuando se ha conectado a Oracle como SYSOPER o SYSDBA (operador o administrador). Proporciona los privilegios que tiene y la responsabilidad con los usuarios.

2.7.Selección de Opciones de Auditoría

Dependiendo de las opciones seleccionada para auditar, los archivos de la auditoría pueden contener tipos diferentes de información. Sin embargo, todas las opciones auditadas generan la información siguiente:

- el usuario que ejecutó la acción auditada
- el código de acción (un número) eso indica la acción auditada ejecutada por el usuario
- el objeto u objetos que hacen referencia a la acción auditada
- la fecha y tiempo en que la acción auditada fue ejecutada

Oracle le permite poner las opciones de la auditoría a tres niveles:

Tabla 2.7.1 Opciones de Auditoría por niveles

Declaración	Las auditorías basadas en el tipo de declaración se enfocan a los SQL de declaración como: CREATE, TRUNCATE, y DROP TABLE, o sea su declaración.
Privilegio	El uso de privilegios para auditoría en un sistema en particular, se lleva a cabo mediante la creación de la tabla de privilegios.
Objeto	En este caso, la auditoría se enfoca a los objetos (tablas) de la BD en particular y de manera interrelacionada.

Auditando Las Conexiones y Desconexiones

Cuando un usuario se conecta o inicia su sesión de trabajo, no se genera un registro único por cada tipo en particular de acción emitida o realizada; más bien, genera un registro por cada sesión creada. Dicho registro contiene fecha y hora de entrada (conexión) y salida (desconexión) de la sesión. La información acumula el tipo de conexión, privilegios, desconexión, tiempo de la sesión, procesos I/O lógicos y físicos, además de un identificador único del registro de la sesión que sirve de parámetro para la auditoría.

Las Opciones de Auditoría de privilegio

Esta opción se dedica a auditar exactamente los privilegios correspondientes al sistema. Por ejemplo: para ejecutar un DELETE ANY TABLE el privilegio debe corresponder a un DELETE TABLE.
Para habilitar esta opción se usaría una declaración como la siguiente:

```
AUDIT DELETE ANY TABLE
      BY ACCESS
      WHENEVER NOT SUCCESSFUL;
```

Las Opciones de Auditoría de objeto

Mediante el diccionario de datos podemos revisar las opciones validas dentro del esquema de los objetos disponibles.
La siguiente tabla lista las declaraciones SQL auditadas para cada objeto de la base de datos.

Tabla 2.7.2 Opciones de Auditoría por objeto

Objeto	Tabla
ALTER	ALTER object (table or sequence)
AUDIT	AUDIT (object)
COMMENT	COMMENT object (table or view)
DELETE	DELETE FROM object (table or view)
EXECUTE	EXECUTE object (procedure)
GRANT	GRANT privilege ON object
INDEX	CREATE INDEX ON object (tables only)
INSERT	INSERT INTO object (table, view, or procedure)
LOCK	LOCK object (table or view)
RENAME	RENAME object (table, view, or procedure)
SELECT	SELECT . . .FROM object (table, view, snapshot)
UPDATE	UPDATE object (table or view)

Habilitando Las Opciones de Auditoría

El comando AUDIT de SQL habilita la acción de auditoría para las opciones de privilegios y a través de objetos.
Auditar bajo las declaraciones de los privilegios puede incluir al usuario para cuya opción se especifica una lista de los usuarios en la que se definen sus privilegios y limitaciones. El comando AUDIT de SQL es el encargado de llevar a cabo esta opción. El usuario no solamente puede tener privilegios como tal o para algunas acciones especificas del sistema, por lo que nos podemos apoyar con el comando AUDIT SYSTEM de privilegios.

Cualquiera que sea la opción de auditoría que se elija, se deben tomar a consideración las siguientes condiciones:

- Obtener información necesaria / obtener información suficiente
- Decidir el punto de auditoría: acceso / sesión

Cada nueva sesión dentro de la Base de Datos será auditada y se generará un archivo de registro dentro del diccionario de datos. Este registro, adicional a los datos auditados predefinidos, contiene la duración de la sesión en la base de datos. Si se deciden cambiar los parámetros de auditoría, las sesiones que le anteceden seguirán utilizando la opción de auditoría anterior o la creada cuando se inicio la sesión.

ADVERTENCIA:

Para generar el reporte de auditoría una vez que se desea revisar la información y volver a ejecutar el proceso de auditoría o concluirla. Oracle genera estos archivos, mismos que pueden extraerse mediante el comando AUDIT_TRAIL más el parámetro del archivo de la Base de Datos.

Habilitando la Auditoría por la declaración de Privilegios

La primera parte consiste en auditar todas las conexiones o accesos a la Base de Datos, ya sean exitosas o no, sin tomar en cuanta al usuario, se podría llamar por SESION (la cual contendrá un valor único). O bien, se puede especificar el usuario, para que sea auditado cada vez que accese a la BD.

AUDITORÍA DE LA SESIÓN;

Para habilitar la auditoría puede basarse en el ejemplo siguiente:

```
AUDIT SESSION
BY scott, lori;
```

Para revisar si el resultado fue exitoso, podemos apoyarnos del comando DELETE ANY TABLE, con la declaración siguiente:

```
AUDIT DELETE ANY TABLE;
```

Para revisar los accesos infructuosos podemos utilizar los comandos SELECT, INSERT Y DELETE en todas las tablas y ejecutando el procedimiento para revisión de Privilegios del Sistema para cada uno de los usuarios que accesó a la BD sin éxito. Podemos basarnos en la siguiente declaración:

```
AUDIT SELECT TABLE, INSERT TABLE, DELETE TABLE,
   EXECUTE PROCEDURE
      BY ACCESS
      WHENEVER NOT SUCCESSFUL;
```

El comando AUDIT SYSTEM será necesario cuando se desee evaluar los privilegios de los usuarios. Normalmente el administrado de la BD es el responsable de los privilegios otorgados, por lo que sobre éste recae cualquier ambigüedad detectada en el sistema de privilegios de los usuarios de la BD.

Habilitando la Auditoría por Objeto

Para auditar las acciones DELETE a la tabla en una sesión del empleado Scott, se realiza la declaración siguientes:

```
AUDIT DELETE ON scott.emp;
```

Para auditar las acciones exitosas SELECT, INSERT y DELETE sobre la tabla DEPT en los accesos del usuario Jward, se lleva a cabo la siguiente declaración:

```
ON jward.dept
     BY ACCESS
     WHENEVER SUCCESSFUL;
```

En la auditoría de un objeto predefinido en una acción no exitosa en un SELECT y para una sesión default se ejecuta la siguiente declaración:

```
AUDIT SELECT
     ON DEFAULT
     WHENEVER NOT SUCCESSFUL;
```

En ausencia del auditor de sistemas, el administrador será el único autorizado para llevar a cabo esta acción como medida de seguridad y control y cuidando meticulosamente que la información se mantenga integra en todo momento.

Deshabilitando las Opciones de Auditoría

El comando NOAUDIT es el encargado de desactivar las opciones de auditoría en Oracle.
Se utiliza para reestablecer la BD a su estado anterior a la Auditoría. Nuevamente es preciso aclarar que los autorizados para ejecutar son el auditor de sistemas y, en su defecto, el administrado de la BD.
Se puede utilizar el comando NOAUDIT acompañado de la cláusula WHENEVER para especificar el resultado tanto de los casos exitosos como los que no lo son.
Los comandos BY SESSION/BY ACCESS no es necesario desactivarlos manualmente, ya que al ejecutar el comando NOAUDIT, se deshabilitan automáticamente.

Deshabilitando la Auditoría de la Declaración y Privilegios

Las declaraciones siguientes desactivan las opciones de auditoría correspondientes:

```
NOAUDIT session;
NOAUDIT session BY scott, lori;
NOAUDIT DELETE ANY TABLE;
NOAUDIT SELECT TABLE, INSERT TABLE, DELETE TABLE,
    EXECUTE PROCEDURE;
```

Las declaraciones siguientes deshabilitan las declaraciones sobre el sistema y opciones de auditoría de privilegios:

```
NOAUDIT ALL;
NOAUDIT ALL PRIVILEGES;
```

Para ejecutar las opciones anteriores se debe tener el privilegio de AUDIT SYSTEM (auditor del sistema).

Deshabilitando La Auditoría de Un Objeto

Las declaraciones siguientes desactivan la auditoría para las opciones correspondientes:

```
NOAUDIT DELETE
   ON emp;
NOAUDIT SELECT, INSERT, DELETE
   ON jward.dept;
```

Además, para desactivar todas las opciones de auditoria del objeto tabla EMP, se ejecuta la aación siguiente:

```
NOAUDIT ALL
   ON emp;
```

Deshabilitando la opción de auditoría a Objetos por default

Para desactivar las opciones de auditoría a un objeto predeterminado, se ejecuta la siguiente acción:

```
NOAUDIT ALL
   ON DEFAULT;
```

NOTA: No es recomendable que otra persona diferente al auditor o al administrador ejecute dichas acciones, es importante, desde el momento en que se otorgan los privilegios definir quién y porqué tendrá el de SYSTEM_AUDIT, para evitar que usuarios comunes realicen cambios dentro de la BD.

2.8.Habilitando y deshabilitando la Auditoría en la Base de Datos

El usuario de base de datos autorizado bajo los privilegios correspondientes, puede llevar a cabo esta acción. Oracle solamente guardará los archivos del proceso de auditoría cuando se encuentre activada la BD. El administrador de la BD generalmente será el responsable.

La base de datos auditada se habilita y deshabilita con el comando AUDIT_TRAIL más el parámetro del archivo de resultado y la bd. El parámetro puede tomar cualquiera de los valores siguientes:

Tabla 2.8.1 Habilitar y deshabilitar una Base de Datos

BD	Habilita la BD que se audita y dirige los archivos de auditoría a la BD que interviene en el proceso.
OS	Habilita la BD que se audita y dirige los archivos de auditoría al SO que interviene en el proceso.
NONE	Deshabilita la auditoría (Este valor por default.)

Después de haber revisado los archivos se reinicia la BD para deshabilitar o habilitar el proceso de auditoría en esta.

2.9.Control del tamaño y crecimiento del proceso de Auditoría

Si el proceso de auditoría ha saturado por completo y no puede insertarse un solo archivo más, ya no se podrán ejecutar más acciones de auditoría con éxito hasta que el proceso sea purgado. En este momento se debe enviar un mensaje de advertencia a los usuarios para poder llevar a cabo el proceso de depuración. Por consiguiente el administrador debe estar al pendiente del tamaño y crecimiento del proceso de auditoría.
Una vez que se habilita el proceso de auditoría y se empiezan a generar los archivos, éste puede crecer debido a diferentes factores:

- Número de opciones de auditoría que se ejecutan
- Frecuencia de ejecución de acciones de auditoría

Para controlar el crecimiento se pueden utilizar lo siguientes métodos:

- Habilitar y deshabilitar la BD después de cierto grupo de actividades ejecutadas. Cuando se habilita se generan los archivo de resultados, pero si se deshabilita y se purga la bd, no se quedan residentes en el sistema.
- Se debe ser sumamente selectivo con las opciones de auditoría a ejecutar, ya que si se está obteniendo información innecesaria, ésta también se estará guardando el los archivos de manera inútil.
- Mantener hermetismo sobre las actividades de auditoría realizadas:

  El administrador de la BD debe tener control sobre los objetos auditables y los privilegios que se conceden a los usuarios.

  Aun cuando todos los usuarios de la BD tenga privilegios para crear una sesión de auditoría, el responsable de la seguridad será el administrador de la BD, que fue quien concedió esos privilegios.

En ambos casos el administrador de la BD es quien controla por completo la seguridad.

El tamaño máximo del proceso de auditoría de la BD, es predeterminado durante la creación de la base de datos (SYS.AUD$TABLE). Por default se alojan bloques de 10K por cada acción.
Dicha acción no puede trasladarse a otro tablespace, pues como ya se dijo, está predefinido en el sistema. Sin embargo, se pueden modificar los parámetros de almacenamiento para los bloques de información.

Depurando la información almacenada por el proceso de Auditoría

Una vez realizada la auditoría, se deshabilita la BD por un lapso de tiempo, el administrador puede reiniciar y eliminar los archivos que se generaron durante el proceso de auditoría y liberar espacio para redireccionar la auditoría o simplemente para que la base de datos continúe con su desempeño normal.
Para anular estos archivo se utiliza la siguiente declaración:

```
DELETE FROM sys.aud$;
```

Por ejemplo, si se desea anular los archivos generados durante la auditoría a la tabla EMP, se puede realizar lo siguiente:

```
DELETE FROM sys.aud$
     WHERE obj$name='EMP';
```

Si se desea respaldar la información generada por el proceso de auditoría para propósitos históricos, el administrador puede copiar los archivos a una tabla de la BD (por ejemplo: "INSERT INTO table SELECT ... FROM sys.aud$ ..."), o exportar los archivos a un directorio del sistema operativo.

Solamente el usuario SYS, tiene el privilegio para ejecutar DELETE ANY TABLE, por lo que seguramente tendrá también el privilegio para ejecutar DELETE SYS.AUD$ y anular los archivos de la BD que intervinieron en el proceso de auditoría.

NOTA:
Si el espacio destinado para la información de la auditoría se encuentra lleno, pueden verse intervenidas las conexiones y los usuarios comunes pueden tener problemas al conectarse a la BD. En este caso el administrador de la BD debe ingresar a ésta como SYS y realizar lo pertinente para liberar espacio.

Reduciendo el Tamaño del Proceso de Auditoría

Una vez finalizado el proceso de auditoría y ya que se han eliminado los archivos de información, el espacio lógico aún existe. Puede darse al caso también que segmentos del espacio asignado no se hayan utilizado, éstos pueden reducirse como sigue:

1. Si se desea optimizar la información actual del proceso de auditoría, se puede copiar a otra tabla en la BD o exportarla utilizando la herramienta EXPORT.
2. Conectarse a la BD con privilegios de administrador.
3. Truncar el SYS.AUD$ utilizando el comando TRUNCATE.
4. Recargar nuevamente los archivos del proceso de auditoría liberados en el paso 1.

2.10. Protección de la Base de Datos mediante el proceso de Auditoría

Al realizar la auditoría sobre actividades sospechosas de la BD, se debe proteger la integridad de los archivos del proceso de auditoría para garantizar la exactitud de la información recabada.
Para proteger el proceso de auditoría es importante que solamente el administrador y el auditor de sistemas ejecuten las acciones de auditoría sobre la BD.
Para auditar posibles modificaciones de datos durante el proceso de auditoría se puede realizar lo siguiente:

```
AUDIT INSERT, UPDATE, DELETE
    ON sys.aud$
    BY ACCESS;
```

Los archivos de auditoría generados como resultado de ésta podrán solo anularse por el administrador o el auditor usando el comando SYS.AUD$.

2.11. Observando la Información de la Base de Datos durante la Auditoría

Los siguientes son ejemplos de cómo se puede examinar e interpretar la información generada durante el proceso de auditoria, e incluye lo siguiente:

- Lista activa de declaraciones en las opciones de auditoría
- Lista activa de privilegios en las opciones de auditoría
- Lista activa de los objetos específicos en las opciones de auditoría
- Lista de objetos predefinidos en las opciones de auditoría
- Lista de archivos de la auditoría
- Lista de archivos para la opción AUDIT SESSION

Para auditar actividades sospechosas en una BD se puede revisar lo siguiente:

- Passwords, tablespace, privilegios de usuarios.
- Número de bloqueos a usuarios con claves exclusivas
- Registros alterados, porqué y por quién

Si por ejemplo, el administrador ha detectado ambigüedades sospechosas en los usuarios JWARD y SWILLIAMS en determinadas acciones, puede realizar lo siguiente:

```
AUDIT ALTER, INDEX, RENAME ON DEFAULT
    BY SESSION;
CREATE TABLE scott.emp . . . ;
CREATE VIEW scott.employee AS SELECT * FROM scott.emp;
AUDIT SESSION BY jward, swilliams;
AUDIT ALTER USER;
AUDIT LOCK TABLE
    BY ACCESS
    WHENEVER SUCCESSFUL;
AUDIT DELETE ON scott.emp
    BY ACCESS
    WHENEVER SUCCESSFUL;
```

Las declaraciones siguientes surgen como consecuencia para el usuario JWARD:

```
ALTER USER tsmith QUOTA 0 ON users;
DROP USER djones;
```

Las declaraciones siguientes surgen como consecuencia para el usuario SWILLIAMS:

```
LOCK TABLE scott.emp IN EXCLUSIVE MODE;
DELETE FROM scott.emp WHERE mgr = 7698;
ALTER TABLE scott.emp ALLOCATE EXTENT (SIZE 100K);
CREATE INDEX scott.ename_index ON scott.emp (ename);
CREATE PROCEDURE scott.fire_employee (empid NUMBER) AS
  BEGIN
    DELETE FROM scott.emp WHERE empno = empid;
  END;
/

EXECUTE scott.fire_employee(7902);
```

A continuación se muestran algunas de las listas usando vistas del proceso de auditoría y el diccionario de datos:

2.11.1. Lista Activa para la declaración de las Opciones de Auditoría

El siguiente query regresa el resultado en la opción de auditoría para las declaraciones fijas:

```
SELECT * FROM sys.dba_stmt_audit_opts;
```

USER_NAME	AUDIT_OPTION	SUCCESS	FAILURE
JWARD	SESSION	BY SESSION	BY SESSION
SWILLIAMS	SESSION	BY SESSION	BY SESSION
	LOCK TABLE	BY ACCESS	NOT SET

2.11.2. Lista Activa para las Opciones de Privilegios en la Auditoría

El siguiente query regresa el resultado en la opción de auditoría para privilegios que son fijos:

```
SELECT * FROM sys.dba_priv_audit_opts;
```

USER_NAME	AUDIT_OPTION	SUCCESS	FAILURE
ALTER USER	BY SESSION	BY SESSION	

2.11.3. Lista Activa de Opciones para la Auditoría de Objetos Específicos

El siguiente query regresa el resultado en la opción de auditoría para cualquier objeto contenido en el esquema Scott:

```
SELECT * FROM sys.dba_obj_audit_opts
    WHERE owner = 'SCOTT' AND object_name LIKE 'EMP%';
```

```
OWNER OBJECT_NAME  OBJECT_TY LT AUD  COM DEL GRA  IND INS LOC ...
----- ------------------- ------------------ ------------ ------- ----- ------- ------ ---- ---
SCOTT EMP          TABLE      S/S -/-  -/-  A/- -/- S/S    -/-  -/- ..
SCOTT EMPLOYEE     VIEW       -/- -/-  -/-  A/- -/- S/S    -/-  -/- ...
```

La vista devuelve la información referente a las opciones de auditoría para el objeto especificado. La información de la vista se interpreta como sigue:

- El carácter "- " indica que la opción de la auditoría no es fija.
- El carácter que "S" indica que la opción de la auditoría es fija por el comando SESSION.
- El carácter "A" indica que la opción de la auditoría es fija, por el comando ACCESS.
- Cada opción de la auditoría tiene dos posibles opciones WHENEVER SUCCESSFUL y WHENEVER NOT SUCCESSFUL, (siempre exitoso o no exitoso)

2.11.4. Lista de Opciones de Auditoría para Objetos por Default

El siguiente query devuelve el resultado en la opción de auditoría para un objeto predefinido.

```
SELECT * FROM all_def_audit_opts;
```

```
ALT  AUD  COM  DEL  GRA  IND  INS LOC REN SEL UPD REF EXE
---  ---  ---  ---  ---  ---  --- --- --- --- --- --- ---
S/S  -/-  -/-  -/-  -/-  S/S  -/- -/- S/S -/- -/- -/- -/-
```

La vista devuelve la información referente a USER_OBJ_AUDIT_OPTS y DBA_OBJ_AUDIT_OPTS

2.11.5. Lista de Archivos de Auditoría

El siguiente query devuelve los archivos generados en la auditoría de un objeto:

```
SELECT * FROM sys.dba_audit_object;
```

2.11.6. Lista de Archivos para la Opción AUDIT SESSION

El siguiente query devuelve los archivos generados en la auditoría en una declaración AUDIT SESSION:

```
SELECT username, logoff_time, logoff_lread, logoff_pread,
   logoff_lwrite, logoff_dlock
   FROM sys.dba_audit_session;
```

USERNAME	LOGOFF_TI	LOGOFF_LRE	LOGOFF_PRE	LOGOFF_LWR	LOGOFF_DLO
JWARD	02-AUG-91	53	2	24	0
SWILLIAMS	02-AUG-91	3337	256	630	0

2.11.7. Auditando A través de los Triggers de la Base de Datos

Se pueden utilizar los triggers de la Base de Datos para complementar la auditoría.

Al decidirse a utilizar los triggers para la actividad de auditoría se deben considerar las siguientes observaciones:

- Los triggers pueden auditar las declaraciones DML utilizadas en las tablas.
- Toda la información de la auditoría de la BD Oracle se guarda automáticamente.
- Una manera más fácil de obtener información auditable de la BD Oracle se obtiene a través de los triggers.

El trigger siguiente audita las modificaciones a la tabla EMP en una BD por renglón. Requiere de un "reason code" para que guarde las variaciones. El trigger se muestra a continuación:

```
CREATE TRIGGER audit_employee
AFTER INSERT OR DELETE OR UPDATE ON emp
FOR EACH ROW
BEGIN
/* AUDITPACKAGE is a package with a public package
   variable REASON. REASON could be set by the
   application by a command such as EXECUTE
   AUDITPACKAGE.SET_REASON(reason_string). Note that a
   package variable has state for the duration of a
   session and that each session has a separate copy of
```

```
   all package variables. */
IF auditpackage.reason IS NULL THEN
   raise_application_error(-20201,'Must  specify  reason  with
',  'AUDITPACKAGE.SET_REASON(reason_string)');
END IF;
INSERT INTO audit_employee VALUES
    (:old.ssn, :old.name, :old.job_classification, :old.sal,
    :new.ssn, :new.name, :new.job_classification, :new.sal,
    auditpackage.reason, user, sysdate );
END;
```

3. Seguridad de las Bases de Datos Oracle

La información dicen que es poder, y como las BD son un almacén de información también almacenan poder, por lo que han sido objeto de intentos de acceso no autorizados desde su nacimiento. Por eso, las BD se han dotado de unos mecanismos que hacen posible la gestión de la seguridad en el acceso a la información que almacenan.

3.1.Posibilidades

Oracle pone al alcance del DBA varios niveles de seguridad:

- Seguridad de cuentas para la validación de usuarios.
- Seguridad en el acceso a los objetos de la base de datos.
- Seguridad a nivel de sistema para la gestión de privilegios globales.

3.1.1. Seguridad de Cuentas

Para acceder a los datos en una BD Oracle, se debe tener acceso a una cuenta en esa BD. Cada cuenta debe tener una palabra clave o password asociada. Una cuenta en una BD puede estar ligada con una cuenta de sistema operativo. Los passwords son fijados cuando se crea un usuario y pueden ser alterados por el DBA o por el usuario mismo. La BD almacena una versión encriptada del password en una tabla del diccionario llamada dba_users. Si la cuenta en la BD está asociada a una cuenta del sistema operativo puede evitarse la comprobación del password, dándose por válida la comprobación de la identidad del usuario realizada por el SO.

3.1.2. Seguridad de Objetos

El acceso a los objetos de la BD se realiza vía privilegios. Estos permiten que determinados comandos sean utilizados contra determinados objetos de la BD. Esto se especifica con el comando GRANT. Los privilegios se pueden agrupar formando lo que se conoce por roles. La utilización de los roles simplifica la administración de los privilegios cuando tenemos muchos usuarios. Los roles pueden ser protegidos con passwords, y pueden activarse y desactivarse dinámicamente, con lo que constituyen una capa más de seguridad en el sistema.

3.1.3. Roles del Sistema

Los roles se pueden utilizar para gestionar los comandos de sistema disponibles para los usuarios. Estos incluyen comandos como CREATE TABLE o SELECT ANY TABLE. Todos los usuarios que quieran acceder a la BD deben tener el rol CONNECT; aquellos que necesiten crear segmentos necesitaran el rol RESOURCE. Un usuario con el rol DBA tiene derecho para ver y manejar todos los datos de la BD. En Oracle CONNECT, RESOURCE y DBA son roles de sistema. Las acciones contra cada tipo de objeto son autorizadas por privilegios separados. Así, un usuario puede tener concedido el privilegio CREATE TABLE, pero no el ALTER TABLE.

3.2.Implementación de Seguridad

No se podrá acceder a la BD a menos que se acceda primero al servidor en el que la BD está ejecutándose. El primer paso en la seguridad de la BD es asegurar la plataforma en la que reside. Una vez que esto ha sido conseguido, se debe considerar la seguridad del sistema operativo. Oracle utiliza una serie de ficheros a los que los usuario no tienen porque acceder de manera directa. Por ejemplo, los ficheros de datos o los de redo log son escritos y leídos sólo por los procesos Oracle. Así, sólo los DBAs que han creado estos ficheros necesitan acceder directamente a ellos a nivel del sistema operativo.

3.2.1. Creación de Usuarios

El objetivo de la creación de usuarios es establecer una cuenta segura y útil, que tenga los privilegios adecuados y los valores por defecto apropiados. En Oracle se puede especificar todo lo necesario para abrir una cuenta con el comando CREATE USER. Los parámetros que se le pueden pasar son:

Tabla 3.2.1.1 Creación de Usuarios

Parámetro	Significado
Username	Nombre del Usuario (Esquema)
Password	Palabra clave de la cuenta. Puede ser asociada directamente a una cuenta del sistema operativo.
Default Tablespace	Espacio de tablas por defecto en el que los objetos de este usuario serán creados. Esto no da al usuario derechos de crear objetos.
Temporary Tablespace	El espacio de tablas en el que se almacenarán los segmentos temporales de las ordenaciones.
Quota	Espacio máximo que puede ocupar en un espacio de tablas.
Profile	Asigna un perfil al usuario. Los perfiles se utilizan para restringir el uso de recursos como el tiempo de CPU.

A continuación se puede ver un ejemplo de uso del comando CREATE USER en el que se crea una cuenta para el usuario Pérez:

```
SVRMGR> create user perez
     2> identified by zerep
     3> default tablespace users
     4> temporary tablespace temp;
```

Si no se especifica un perfil, se aplica el perfil por defecto de la BD, que se llama DEFAULT y tiene asignados todos los límites a UNLIMITED.

Si no se especifica una cuota el usuario no puede crear objetos.

3.2.2. Eliminación de Usuarios

Los usuarios pueden ser eliminados de la BD utilizando el comando DROP USER. Este comando tiene un único parámetro, CASCADE, el cual permite borrar todos los objetos del usuario antes de eliminar el usuario.

A continuación un ejemplo en el que eliminamos al usuario Pérez:

```
SVRMGR> drop user perez cascade;
```

Si a continuación se crea otro usuario con el mismo nombre no hereda los objetos del anterior usuario con ese nombre. La razón estriba en que Oracle asigna a cada cuenta un número además del nombre, y utiliza ese número para determinar el propietario de todos los objetos que crea esa cuenta, y no utiliza el nombre sino para la comunicación con los usuarios. De este modo al crear un nuevo usuario, aunque sea con el mismo nombre, no puede heredar los objetos que antes eran de otro usuario con el mismo nombre.

3.2.3. Privilegios del Sistema

Los roles de sistema se utilizan para distribuir la disponibilidad de los comandos del sistema utilizados para gestionar la BD. Los privilegios más comunes están en la siguiente tabla. En ella se distinguen entre privilegios de manejo de objetos y de gestión de la BD. La palabra clave ANY significa que ese usuario tiene el privilegio para todos los esquemas en la BD. Hay que hacer notar que ANY y PUBLIC no son sinónimos.

Tabla 3.2.3.1 Comandos del Sistema

Privilegio	*Capacidades*
Manejo de Objetos	...
CREATE ANY INDEX	Crear cualquier índice.
CREATE [PUBLIC] SYNONYM	Crear sinónimos [públicos].
CREATE [ANY] TABLE	Crear tablas. El usuario debe tener cuota en el espacio de tablas, o ha de tener asignado el privilegio UNLIMITED TABLESPACE.
CREATE [ANY] VIEW	Crear vistas.
ALTER ANY INDEX	Alterar cualquier índice.
ALTER ANY TABLE	Alterar cualquier tabla
DROP ANY INDEX	Borrar cualquier índice.
DROP ANY SYNONYM	Borrar cualquier sinónimo.
DROP PUBLIC SYNONYM	Borrar sinónimos públicos.
DROP ANY VIEW	Borrar cualquier vista.
DROP ANY TABLE	Borrar cualquier tabla.
SELECT ANY TABLE	Efectuar selecciones de cualquier tabla o vista.
INSERT ANY TABLE	Insertar en cualquier tabla o vista.
DELETE ANY TABLE	Borrar filas de cualquier tabla o vista, y también truncar.
ALTER SESSION	Alterar los parámetros de la sesión.
CREATE SESSION	Conectarse a la BD.
Gestión de la BD	...
CREATE PROFILE	Crear perfiles de usuario.
CREATE ROLE	Crear roles.
CREATE ROLLBACK SEGMENT	Creación de segmentos de rollback.
CREATE TABLESPACE	Crear espacios de tablas.
CREATE USER	Crear usuarios.
ALTER PROFILE	Alterar perfiles existentes.
ALTER ANY ROLE	Alterar cualquier rol.

ALTER ROLLBACK SEGMENT	Alterar segmentos de rollback.
ALTER TABLESPACE	Alterar espacios de tablas.
ALTER USER	Alterar usuarios.
DROP PROFILE	Borrar un perfil existente.
DROP ANY ROLE	Borrar cualquier rol.
DROP ROLLBACK SEGMENT	Borrar un segmento de rollback existente.
DROP TABLESPACE	Borrar un espacio de tablas.
DROP USER	Borrar un usuario. Añadir CASCADE si el usuario posee objetos.
ALTER DATABASE	Permite una sentencia ALTER DATABASE.
GRANT ANY PRIVILEGE	Otorgar cualquiera de estos privilegios.
GRANT ANY ROLE	Otorgar cualquier rol a un usario.
UNLIMITED TABLESPACE	Puede usar una cantidad de almacenamiento ilimitada.
DROP PROFILE	Borrar un perfil existente.

Los privilegios se pueden agrupar en roles, para así satisfacer a distintos tipos de usuarios. En la instalación se crea un rol llamado OSOPER que sirve para los operarios de la máquina donde está la BD y permite realizar copias de seguridad en frió y en caliente. Los privilegios de OSOPER son STARTUP, SHUTDOWN, ALTER DATABASE OPEN/MOUNT, ALTER DATABASE BACKUP, ARCHIVE LOG, RECOVER y RESTRICTED SESSION.

Se pueden crear nuevos roles. Por ejemplo, podemos crear un rol llamado creadorCuentas que sólo pueda crear usuarios y no pueda realizar ninguna otra operación de DBA. Las sentencias que permiten hacer esto son las siguientes:

```
SVRMGR> create role creadorCuentas;
Statement processed.
SVRMGR> grant create session, create user to creadorCuentas;
Statement processed.
```

Oracle incluye otros tres roles de sistema: CONNECT, RESOURCE y DBA, cuyos privilegios son:

Tabla 3.2.3.2 Comandos del Sistema

Rol	*Privilegios*
CONNECT	alter session, create session, create cluster, create table, create view, create synonym, create sequence, create database link
RESOURCE	create cluster, create table, create procedure, create sequence, create trigger
DBA	todos los privilegios de sistema con la opción with admin option

3.2.4. Perfiles del Usuario

Los perfiles se utilizan para limitar la cantidad de recursos del sistema y de la BD disponibles para un usuario. Si no se definen perfiles para un usuario se utiliza el perfil por defecto, que especifica recursos ilimitados.

Los recursos que pueden ser limitados vía perfil son los siguientes:

Tabla 3.2.4.1 Recursos que pueden ser limitados

Recurso	*Descripción*
SESSIONES_PER_USER	El número de sesiones concurrentes que un usuario puede tener en una instancia.
CPU_PER_SESSION	El tiempo de CPU, en centenas de segundos, que una sesión puede utilizar.
CONNECT_TIME	El número de minutos que una sesión puede permanecer activa.
IDLE_TIME	El número de minutos que una sesión puede permanecer sin que sea utilizada de manera activa.
LOGICAL_READS_PER_SESSION	El número de bloques de datos que se pueden leer en una sesión.
LOGICAL_READS_PER_CALL	El número de bloques de datos que se pueden leer en una operación.
PRIVATE_SGA	La cantidad de espacio privado que una sesión puede reservar en la zona de SQL compartido de la SGA.
COMPOSITE_LIMIT	El número de total de recursos por sesión, en unidades de servicio. Esto resulta de un calculo ponderado de CPU_PER_SESSION, CONNECT_TIME, LOGICAL_READS_PER_SESSION y PRIVATE_SGA, cuyos pesos se pueden variar con el comando ALTER RESOURCE COST.

Los perfiles se pueden crear vía el comando CREATE PROFILE, y se pueden modificar con la sentencia ALTER PROFILE.

En general, el perfil por defecto debe ser adecuado para los usuarios normales; los usuarios con requerimientos especiales deberían tener perfiles especiales.

3.2.5. Cuentas BD sobre Cuentas SO

Los usuarios pueden entrar en la BD una vez que han dado un nombre de usuario y una palabra de paso. Sin embargo, es posible aprovecharse del Sistema Operativo para obtener un nivel adicional de autentificación.

La cuenta de la BD y del SO pueden relacionarse de manera que la de la BD se diferencie sólo en un prefijo de la cuenta del SO. Este prefijo se fija en el parámetro OS_AUTHENTIC_PREFIX del fichero init.ora. Por defecto esta puesto a "OPS$", pero puede tomar cualquier forma, incluso ser la cadena nula, "", con lo que no existe prefijo que diferencie las cuentas en la BD y en el SO.

Por ejemplo, consideremos la cuenta del SO llamada alu10. La correspondiente cuenta en la BD sería OPS$ALU10. Cuando el usuario alu10 está en activo en el SO, puede acceder a su cuenta OPS$ALU10 sin especificar el password, como se muestra a continuación:

```
$ sqlplus /
```

En este caso el carácter "/" toma el lugar de la combinación del nombre de usuario y del password que debería aparecer.

Las cuentas pueden ser creadas con passwords aunque no vaya a ser utilizado. Ya que de este modo será posible acceder a la cuenta de OPS$ALU10 desde otra cuenta diferente a la alu10 del SO. La sentencia de creación de la cuenta OPS$ALU10 puede ser la siguiente:

```
SVRMGR> create user ops$alu10
     2> identified by 01ula
     3> default tablespace users
     4> temporary tablespace temp;
```

Así, la forma de conectarse como OPS$ALU10 desde una cuenta del SO diferente es:

```
$ sqlplus ops$alu10/01ula
```

Existen dos formas de evitar este problema. La primera es creando el usuario BD sin especificar un password, utilizando la claúsula IDENTIFIED EXTERNALLY. De esta manera

evitamos la necesidad de expecificar un password para la cuenta, obligando a la conexión entre las cuentas de la BD y del SO:

```
SVRMGR> create user ops$alu10
     2> identified externally
     3> default tablespace users
     4> temporary tablespace temp;
```

La otra manera de evitar el problema es crear una cuenta con un password imposible, aspecto este que se explicará más adelante.

3.2.6. Protección por passwords

Los passwords puede proteger tanto cuentas como roles. Los passwords se fijan a la hora de la creación de ambos y se pueden modificar con los comandos ALTER USER y ALTER ROLE, respectivamente.

No es necesario asignar un password a un rol, pero si tiene uno debe ser especificado por el usuario cuando se asigna ese rol.

3.2.7. Gestionando Privilegios

Los privilegios dan acceso a los usuarios a los datos que no poseen. Los roles con grupos de privilegios que facilitan la administración de los privilegios. Pero los privilegios se pueden manejar de manera explícita en algunas circunstancias.

Los privilegios se crean vía el comando GRANT y son registrados en el diccionario de datos.

Los privilegios que pueden otorgarse sobre objetos son los siguientes:

Tabla 3.2.7.1 Privilegios que pueden otorgarse sobre objetos

Privilegio	*Capacidades Otorgadas*
SELECT	Puede consultar a un objeto.
INSERT	Puede insertar filas en una tabla o vista. Puede especificarse las columnas donde se permite insertar dentro de la tabla o vista.
UPDATE	Puede actualizar filas en una tabla o vista. Puede especificarse las columnas donde se permite actualizar dentro de la tabla o vista.
DELETE	Puede borrar filas dentro de la tabla o vista.
ALTER	Puede alterar la tabla.
INDEX	Puede crear índices de una tabla.
REFERENCES	Puede crear claves ajenas que hagan referencia a esta tabla.
EXECUTE	Puede ejecutar un procedimiento, paquete o función.

Haciendo un privilegio PUBLIC lo hace disponible a todos los usuarios de la BD.

Aunque los privilegios se puedan otorgar individualmente, no resulta razonable basar la gestión de los privilegios en su asignación individual. La gestión de los privilegios se facilita con la utilización de los roles. A continuación se puede ver como se crean dos roles, el ALUMNOS que permite establecer una sesión, y el rol INSERTA_PEREZ que permite insertar y seleccionar en la tabla emp de perez:

```
SVRMGR> create role alumnos;
Statement processed.
SVRMGR> grant create session to alumnos;
Statement processed.
SVRMGR> create role inserta_perez;
Statement processed.
SVRMGR> grant select, insert on perez.emp to inserta_perez;
Statement processed.
```

Se pueden asignar roles a roles:

SVRMGR> grant usuarios to inserta_perez;
Los roles pueden asignarse a los usuarios. Así, podemos asignar el rol INSERTA_PEREZ al usuario alu20: SVRMGR> grant inserta_perez to alu20;

Los roles se pueden denegar con el comando REVOKE.

3.2.8. Lista de Privilegios Otorgados

La información de los privilegios otorgados se almacena en el diccionario de datos. Estos datos son accesibles a través de las siguientes vistas del diccionario de datos:

Tabla 3.2.8.1 Privilegios que se almacenan en el DD

Vista	*Contenidos*
DBA_ROLES	Nombres de los roles y su estado del *password*.
DBA_ROLES_PRIVS	Usuarios a los que han sido otorgados roles.
DBA_SYS_PRIVS	Usuarios a los que han sido otorgados privilegios del sistema.
DBA_TAB_PRIVS	Usuarios a los que han sido otorgados privilegios sobre objetos.
DBA_COL_PRIVS	Usuarios a los que han sido otorgados privilegios sobre columnas de tablas.
ROLE_ROLE_PRIVS	Roles que han sido otorgados a otros roles.
ROLE_SYS_PRIVS	Privilegios de sistema que han sido otorgados a roles.
ROLE_TAB_PRIVS	Privilegios de tabla que han sido otorgados a roles.

3.3.Encriptación de passwords y Trucos

Conocer el modo en que se encriptan y se tratan los passwords puede posiblitar al DBA la realización de ciertas tareas que de otro modo le resultarían imposibles. Esto incluye el establecer passwords imposibles y la habilidad de convertirse en otro usuario.

3.3.1. Almacenamiento de passwords

Cuando se especifica un password para un usuario o rol, la BD almacena la versión encriptada del mismo en el diccionario de datos. El mismo password para diferentes usuarios genera diferentes versiones encriptadas. Éstas están compuestas por una cadena de 16 caractéres alfanuméricos (con las letras en mayúsculas).

El proceso de validación de los passwords es sencillo, ya que cuando un usuario introduce su password es encriptado y comparado lo almacenado en el diccionario de datos. Si son iguales el password es correcto; incorrecto en otro caso.

Se puede echar un vistazo a los passwords mirando en la tabla DBA_USERS:

```
SQL> select username, password from dba_users;
```

3.3.2. Passwords Imposibles

¿Qué puede pasar si en vez de especificar el password especificamos su versión encriptada, y esa versión encriptada no sigue las normas de generación de passwords encriptados? El resultado es que tendremos una cuenta a la que nunca se podrá entrar, ya que ningún password generará la versión encriptada que está almacenada en el diccionario de datos.

Esto se puede hacer utilizando una forma no comentada del comando CREATE USER como se indica a continuación:

```
SVRMGR> create user perez identified by values
'123AB456CD789EF0';
```

Una buena manera de crear una versión encriptada que no coincida con ningún password es poner menos de 16 caractéres, de esta manera tendremos un password imposible.

Los efectos de este comando se ven cuando el usuario perez intenta acceder introduciendo su password, desde otra cuenta por ejemplo. Como la versión encriptada del password no va a coincidir con la almacenada, se obliga al usuario perez a entrar desde su cuenta del SO, proceso que no comprueba el password de la BD. De este modo se impide que un usuario pueda suplantar a otro desde una cuenta de SO no apropiada.

3.3.3. Convertirse en otro usuario

Como se puede fijar la versión encriptada de un password, es posible tomar una cuenta temporalmente, cambiando su password original, para restaurarlo a continuación. Esto permite que el DBA se convierta temporalmente en otro usuario.

Esto se puede hacer siguiendo los siguientes pasos:

1. Consultar la tabla DBA_USERS para conseguir la versión encriptada del password actual del usuario que vamos a utilizar.
2. Generar el comando alter user que permita restaurar el password original, guardándolo en un fichero para su posterior ejecución.
3. Cambiar el password de la cuenta y acceder a ella.
4. Cuando el trabajo como el otro usuario haya acabado, ejecutar el comando alter user creado antes para restaurar el valor original del password.
5. Esto se puede ver en el siguiente script:

```
REM *
REM * Este script genera los comandos necesarios para
convertirse
REM * temporalmente en otro usuario.
REM *
REM * Solo para DBAs
REM *

set pagesize 0
set verify off
set feedback off
set echo off

REM *
REM * Preguntar por el usuario a manipular.
REM *

accept user prompt 'Usuario? '
set termout off

REM *
REM * Crear el fichero llamado reset.sql
REM *

spool reset.sql

REM *
REM * seleccionar el password encriptado de DBA_USERS.
REM *
```

```
select 'alter user &&user
identified by values '||''''||password||''''||';'
from dba_users where username= upper('&&user');

prompt ! rm reset.sql
prompt exit

REM *
REM * Cerrar el fichero llamado reset.sql
REM *

spool off

REM *
REM * Cambiar el password a user
REM *

set termout on
accept clave prompt 'Nueva Clave para &&user? '

REM *
REM * Cambiar la clave para el usuario
REM *

alter user &&user identified by &&clave;

connect &&user/&&clave;

prompt ***
prompt *** Ahora es &&user
prompt *** No olvide ejecutar reset.sql al final
prompt ***
```

La ejecución de este script produce otro script llamado reset.sql con el siguiente contenido:

```
alter user pepe identified by values '742A081355525D4E';
! rm reset.sql
exit
```

El script reset.sql se ejecutará con la orden

```
$ sqlplus system/manager @reset.sql
```

3.4.Auditoría de Seguridad

El SGBD Oracle tienen la capacidad de auditar todas las acciones que tienen lugar en la BD. Se pueden auditar tres tipos de acciones:

1. intentos de entrada en cuentas de la BD.
2. accesos a los objetos de la BD.
3. acciones sobre la BD.

La BD registra todos los intentos de acción, tanto los exitosos como los infructuosos, aunque es un parámetro configurable.

Para habilitar la capacidad de auditoría, se debe fijar el parámetro AUDIT_TRAIL en el fichero init.ora. Los registros de auditoría se almacenan en la tabla SYS.AUD$ o bien su gestión se deja al SO. Cuando se decide utilizar la tabla SYS.AUD$ esta debe revisarse periódicamente, por si hiciera falta truncarla debido a que su aumento de tamaño puede causar problemas de espacio en el tablespace SYSTEM. Los valores del parámetro AUDIT_TRAIL son los que se exponen en la siguiente tabla:

Tabla 3.4.1 Valores del parámetro AUDIT_TRAIL

Valor	*Descripción*
NONE	Deshabilita la auditoría
BD	Habilita la auditoría, escribiendo en la tabla SYS.AUD$.
OS	Habilita la auditoría, dejando al SO su gestión.

3.4.1. Auditando Conexiones

Todo intento de conexión con la BD será registrado. El comando para iniciar la auditoría es

```
SVRMGR> audit session;
```

Para determinar si se deben registrar sólo los éxitos, o sólo los fracasos se pueden utilizar los siguientes comandos:

```
SVRMGR> audit session whenever successful;
SVRMGR> audit session whenever not successful;
```

Si los registros de auditoría se almacenan en la tabla SYS.AUD$, entonces pueden verse a través de la vista DBA_AUDIT_SESSION.

```
select
   os_username,                    /* nombre de usuario SO */
   username,                       /* nombre de usuario BD */
   terminal,
   decode(returncode,'0','Conectado',
                  '1005','Solo username, sin password',
                  '1017','Password incorrecto',
                   returncode),  /* comprobacion de error */
   to_char(timestamp,'DD-MON-YY   HH24:MI:SS'),   /*   hora   de
entrada */
   to_char(logoff_time,'DD-MON-YY   HH24:MI:SS')   /*   hora   de
salida */
from dba_audit_session;
```

Para deshabilitar la auditoria de las conexiones basta con ejecutar la siguiente sentencia:

```
SVRMGR> noaudit session;
```

3.4.2. Auditando Acciones

Se puede auditar cualquier acción que afecte a cualquier objeto de la BD. Para facilitar la gestión, las acciones a auditar se encuentran agrupadas según los grupos que se muestran en la siguiente tabla:

Tabla 3.4.2.1 Acciones Auditables

Grupo	*Comandos Auditados*
CLUSTER	Todas las sentencias que afecten a *clusters*.
DATABASE LINK	Todas las sentencias que afecten a enlaces de BD.
EXISTS	Todas las sentencias que fallen porque ya existe un objeto en la BD.
INDEX	Todas las sentencias que afecten a índices.
NOT EXISTS	Todas las sentencias que fallen porque un determinado objeto no existe.
PROCEDURE	Todas las sentencias que afecten a procedimientos.
PROFILE	Todas las sentencias que afecten a perfiles.
PUBLIC DATABASE LINK	Todas las sentencias que afecten a enlaces públicos de BD.
PUBLIC SINONYM	Todas las sentencias que afecten a sinónimos públicos.
ROLE	Todas las sentencias que afecten a roles.
ROLLBACK SEGMENT	Todas las sentencias que afecten a segmentos de *rollback*.

SEQUENCE	Todas las sentencias que afecten a secuencias.
SESSION	Todas las sentencias de acceso a la BD.
SYNONYM	Todas las sentencias que afecten a sinónimos.
SYSTEM AUDIT	Todas las sentencias AUDIT y NOAUDIT.
SYSTEM GRANT	Todas las sentencias afecten a privilegios.
TABLE	Todas las sentencias que afecten a tablas.
TABLESPACE	Todas las sentencias que afecten a espacios de tablas.
TRIGGER	Todas las sentencias que afecten a disparadores.
USER	Todas las sentencias que afecten a las cuentas de usuarios.
VIEW	Todas las sentencias que afecten a vistas.

Por ejemplo, para auditar todas acciones que tienen que ver con las tablas sirve el siguiente comando:

```
SVRMGR> audit table;
```

Y para deshabilitar la auditoría se utilizará el siguiente comando:

```
SVRMGR> noaudit table;
```

También se puede afinar un poco más en la auditoría fijando un usuario concreto al que seguir la pista:

```
SVRMGR> audit table by perez;
```

Cada acción auditada recibe un código numérico al que se puede acceder a través de la vista AUDIT_ACTIONS. Una vez que conocemos el código de la acción, podemos utilizarlo para determinar como dicha acción ha afectado a un objeto, consultado la vista DBA_AUDIT_OBJECT.

3.4.3. Auditando Objetos

Además de la auditoría de acciones sobre los objetos, se puede seguir el rastro a las operaciones de manipulación de tablas: SELECT, INSERT, UPDATE y DELETE. Estas auditorías se pueden hacer por sesión o por acceso.

Un ejemplo de sentencias de auditorías sobre objetos se puede ver en el siguiente grupo de sentencias:

```
SVRMGR> audit insert on perez.emp;
SVRMGR> audit all on perez.emp by session;
SVRMGR> audit delete on perez.emp by access;
```

Los registros de auditoría se pueden ver en la misma vista DBA_AUDIT_OBJECT anteriormente mencionada.

3.4.4. Protegiendo los Registros de Auditoría

Los registros de la tabla SYS.AUD$ pueden ser objeto de intentos de acceso para ser eliminados ya que pueden reflejar acciones no autorizadas en la BD. Así, resulta interesante reflejar ese tipo de acciones. Esto se consigue con el siguiente comando:

```
SVRMGR> audit all on sys.aud$ by access;
```

De este modo cualquier acción contra la tabla SYS.AUD$ quedará registrado. Además, las acciones contra la tabla SYS.AUD$ sólo pueden ser borradas por los usuarios que puedan conectarse como INTERNAL.

4. Backup y Recuperación Oracle

Para conseguir un funcionamiento seguro de la BD y una pronta recuperación ante fallos se necesita planear una estrategia de copias de seguridad, backup, y de recuperación, recovery, ya que de nada sirve pensar que estamos a salvo de tales circunstancias, y que eso no me puede pasar a mí. Y el primer paso a dar es definir las características fundamentales de la implantación, porque mal vamos a conseguir unos objetivos si se desconocen o están indefinidos. El segundo paso es establecer unos planes de copias de seguridad y recuperación que nos permitan asegurar los objetivos.

4.1.Introducción al Backup y a la Recuperación

Planear y comprobar los procedimientos de backup del sistema es la única garantía que existe contra fallos del sistema, del SO, del software o cualquier otro tipo de circunstancias.

Las causas de error en una sistema de BD pueden agruparse en las siguientes categorías:

Físicas
son causadas por fallos del hardware, como por ejemplo del disco o de la CPU.
De Diseño
son agujeros en el software, ya sea en el SO o en el SGBD.
De Funcionamiento
son causadas por la intervención humana, debidos a fallos del DBA, configuraciones inapropiadas o mal planteamiento de los procedimientos de backup.

Del entorno
como por ejemplo desastres naturales, fallos de corriente, temperatura excesiva.
De entre todas estas posibilidades, el DBA sólo puede influir y prever los errores de funcionamiento, ya que el resto habitualmente no está dentro de sus responsabilidades y capacidades.

Dada la complejidad de los sistemas actuales y las necesidades cada vez más críticas en la disponibilidad de los sistemas, donde una BD caída puede causar pérdidas millonarias, puede ser interesante considerar los mecanismos de protección hardware y de redundancia que la tecnología nos proporciona:

UPS o fuentes de corriente ininterrumpida, espejado de disco, o tecnología RAID, Componentes duplicados, Sistemas redundantes.

Una de las más importantes decisiones que un DBA debe tomar es decidir si arrancar la BD en modo ARCHIVELOG o no. Esta decisión tiene sus ventajas e inconvenientes:

Ventajas:
Aunque se pierdan los ficheros de datos, siempre se puede recuperar la BD con una copia antigua de los ficheros de datos y los ficheros de redo log archivados.
Es posible realizar backups en caliente.
Inconvenientes:
Se necesitará más espacio en disco.
El trabajo del DBA se incrementa al tener que determinar el destino del archivado de los redo log.

4.1.1. Presentación del Backup

Los backups se pueden clasificar en físicos y lógicos. Los físicos se realizan cuando se copian los ficheros que soportan la BD. Entre estos se encuentran los backups del SO, los backups en frío y los backups en caliente.

Los backups lógicos sólo extraen los datos de las tablas utilizando comandos SQL y se realizan con la utilidad export/import.

Backups del SO
Este tipo de backup es el más sencillo de ejecutar, aunque consume mucho tiempo y hace inaccesible al sistema mientras se lleva a cabo. Aprovecha el backup del SO para almacenar también todos los ficheros de la BD. Los pasos de este tipo de backup son los siguientes:

1. Parar la BD y el SO
2. Arrancar en modo superusuario.
3. Realizar copia de todos los ficheros del sistema de ficheros
4. Arrancar el sistema en modo normal y luego la BD.

Backups de la BD en Frio
Los backups en frio implican parar la BD en modo normal y copiar todos los ficheros sobre los que se asienta. Antes de parar la BD hay que parar también todos las aplicaciones que estén trabajando con la BD. Una vez realizada la copia de los ficheros, la BD se puede volver a arrancar.

Backups de la BD en Caliente
El backup en caliente se realiza mientras la BD está abierta y funcionando en modo ARCHIVELOG. Habrá que tener cuidado de realizarlo cuando la carga de la BD sea pequeña. Este tipo de backup consiste en copiar todos los ficheros correspondientes a un tablespace determinado, los ficheros redo log archivados y los ficheros de control. Esto para cada tablespace de la BD.
Backups Lógicos con Export/Import

Estas utilidades permiten al DBA hacer copias de determinados objetos de la BD, así como restaurarlos o moverlos de una BD a otra. Estas herramientas utilizan comandos del SQL para obtener el contenido de los objetos y escribirlos en/leerlos de ficheros
Una vez que se ha planeado una estrategia de backup y se ha probado, conviene automatizarla para facilitar así su cumplimiento.

4.1.2. Presentación de la Recuperación

Oracle proporciona diferentes modos de recuperar un fallo en la BD, y es importante que el DBA conozca como funciona cada uno de ellos para determinar cuándo ha de ser utilizado.

Una de las mayores responsabilidades del DBA consiste en tener la BD a punto, y prepararla ante la posibilidad de que se produzca un fallo. Así, ante un fallo el DBA podrá recuperar la BD en el menor tiempo posible. Los procesos de recuperación dependen del tipo de error y de las estructuras afectadas.

Así, los tipos de error que se pueden producir son:

Errores de Usuario
Como por ejemplo un usuario borrando una fila o eliminando una tabla. Estos errores se solucionan importando una tabla de una copia lógica anterior. Si no se dispone de la copia lógica, se puede recuperar la BD en una instancia auxiliar, exportar la tabla en cuestión de la instancia auxiliar e importarla en la instancia operativa.

Fallos de Sentencias
Se definen como la imposibilidad del SGBD Oracle de ejecutar alguna sentencia SQL. Un ejemplo de esto se produce cuando se intenta una selección de una tabla que no existe. Estos fallos se recuperan automáticamente mediante un rollback de la transacción que contenía la sentencia fallida. El usuario necesitará volver a ejecutar otra vez la transacción cuando se haya solucionado la causa del problema.

Fallos de Procesos
Es una terminación anormal de un proceso. Si el proceso era un proceso de usuario, del servidor o de una aplicación el PMON efectuará la recuperación del proceso. Si el proceso era alguno de los de background, la instancia debe de ser parada y arrancada de nuevo, proceso durante el cual se recupera la caída efectuando un roll forward y un rollback de las transacciones no confirmadas.

Fallos de la Red
Algunas veces los fallos en la red producen fallos de proceso, que son tratados por el PMON. Si en el error de red se ve envuelta una transacción distribuida, una vez que se reestablece la conexión, el proceso RECO resuelve los conflictos automáticamente.

Fallos de Instancia
Pueden deberse a fallos físicos o de diseño del software que hacen que algún proceso background caiga y la instancia con él. La recuperación es automática cuando se levanta la BD, tomándose más o menos tiempo en la recuperación.

Fallos del Sistema
Son los fallos más peligrosos, no sólo porque se pueden perder datos, sino porque se tarda más tiempo en recuperar que los otros fallos. Además se depende mucho de la experiencia del DBA para levantar la BD rápidamente y sin pédida (o casi) de datos.
Existen tres tipos de recuperación en Oracle: a nivel de bloque, de thread y física.

Recuperación de bloques
Es el mecanismo de recuperación más simple, y se realiza automáticamente. Se produce cuando un proceso muere justo cuando está cambiando un bloque, y se utilizan los registros redo log en línea para reconstruir el bloque y escribirlo en disco.

Recuperación de threads
Se realiza automáticamente cuando Oracle descubre que una instancia muere dejando abierto un thread, entonces se restauran los bloques de datos modificados que estaban en el cache de la instancia muerta, y cerrando el thread que estaba abierto. La recuperación se efectúa automáticamente cuando la BD se levanta.

Recuperación física
Se realiza como respuesta a un comando RECOVER. Se utiliza para convertir los ficheros de backup en actuales, o para restaurar los cambios que fueron perdidos cuando un fichero de datos fue puesto offline sin un checkpoint, aplicando los fichero redo log archivados y en línea.

4.2.Principios de Backup

Un backup válido es una copia de la información sobre la BD necesaria para reconstruir la BD a partir de un estado no utilizable de la misma. Normalmente, si la estrategia de backup se basa en la copia de los ficheros de datos y en el archivado de los ficheros redo log, se han de tener copias de los ficheros de datos, de los ficheros de control, de los ficheros redo log activos y también de los archivados. Si se pierde uno de los ficheros redo log archivados se

dice que se tiene un agujero en la secuencia de ficheros. Esto invalida el backup, pero permite a la BD ser llevada hasta el principio del agujero realizando una recuperación incompleta.

4.2.1. Diseño de BD y Reglas Básicas de Backup

Antes de nada, es muy importante entender ciertas reglas que determinan la situación de los ficheros y otras consideraciones que afectarán al esquema de backup:

Es recomendable archivar los ficheros redo log en disco, y luego copiarlos a cinta, pero siempre en un disco diferente del que soporta los ficheros de datos y de redo log activos.
Los ficheros copias no deben estar en el mismo dispositivo que los originales. No siempre hay que pasar las copias a cinta, ya que si se dejan en disco se acelera la recuperación. Además, si se copian las copias a cinta y se mantienen en el disco, se puede sobrevivir a diversos fallos de dispositivo.
Se deberían mantener diferentes copias de los ficheros de control, colocadas en diferentes discos con diferentes controladores.
Los ficheros redo log en línea deben estar multiplexados, con un mínimo de 2 miembros por grupo, residiendo cada miembro en un disco distinto.
Siempre que la estructura de la BD cambie debido a la inclusión, borrado o renombrado de un fichero de datos o de redo log, se debe copiar el fichero de control, ya que almacenan la estructura de la BD. Además, cada fichero añadido también debe ser copiado. El fichero de control puede ser copiado mientras la BD está abierta con el siguiente comando:

```
SVRMGR> alter database backup controlfile to 'destino';
```

Teniendo en cuenta las reglas anteriores, los siguientes puntos pueden considerarse un ejemplo de estrategia de backup:

Activar el modo ARCHIVELOG.
Realizar un backup al menos una vez a la semana si la BD se puede parar. En otro caso, realizar backups en caliente cada día.
Copiar todos los ficheros redo log archivados cada cuatro horas. El tamaño y el número de ellos dependerá de la tasa de transacciones.
Efectuar un export de la BD semanalmente en modo RESTRICT.

4.2.2. Backups Físicos

Los backups físicos son aquellos que copian físicamente los ficheros de la BD. Existen dos opciones: en frío y en caliente. Se dice que el backup es en frio cuando los ficheros se copian con la BD esta parada. En caliente es cuando se copian los ficheros con la BD abierta y funcionando.

Backup en Frío
El primer paso es parar la BD con el comando shutdown normal. Si la BD se tiene que parar con inmediate o abort debe rearrancarse con el modo RESTRICT y vuelta a parar en modo normal. Después se copian los ficheros de datos, los de redo log y los de control, además de los redo log archivados y aún no copiados.

Una buena idea es automatizar todo este proceso con los scripts correspondientes, de modo que no nos olvidemos de copiar ningún fichero.

Como este tipo de backup es una copia de los ficheros de la BD, si estos contienen algún tipo de corrupción, la traspasaremos a la copia de seguridad sin detectarla. Por esto es importante comprobar las copias de seguridad.

Backup en Caliente
Si la implantación de BD requiere disponibilidad de la misma 24h. al día, 7 días a la semana no se pueden realizar backups en frió. Para efectuar un backup en caliente debemos trabajar con la BD en modo ARCHIVELOG. El procedimiento de backup en caliente es bastante parecido al frio. Existen dos comandos adicionales: begin backup antes de comenzar y end backup al finalizar el backup. Por ejemplo, antes y después de efectuar un backup del tablespace users se deberían ejecutar las sentencias:

```
SVRMGR> alter tablespace users begin backup;
SVRMGR> alter tablespace users end backup;
```

Así como el backup en frió permitía realizar una copia de toda la BD al tiempo, en los backups en caliente la unidad de tratamiento es el tablespace. El backup en caliente consiste en la copia de los ficheros de datos (por tablespaces), el actual fichero de control y todos los ficheros redo log archivados creados durante el periodo de backup. También se necesitarán todos los ficheros redo log archivados después del backup en caliente para conseguir una recuperación total.

4.2.3. Backups Lógicos

Este tipo de backups copian el contenido de la BD pero sin almacenar la posición física de los datos. Se realizan con la herramienta export que copia los datos y la definición de la BD en un fichero en un formato interno de Oracle.

Para realizar un export la BD debe estar abierta. Export asegura la consistencia en la tabla, aunque no entre tablas. Si se requiere consistencia entre todas las tablas de la BD entonces no se debe realizar ninguna transacción durante el proceso de export. Esto se puede conseguir si se abre la BD en modo RESTRICT.

Entre las ventajas de efectuar un export están las siguientes:

- Se puede detectar la corrupción en los bloques de datos, ya que el proceso de export fallará.
- Protege de fallos de usuario, por ejemplo si se borra una fila o toda una tabla por error es fácil recuperarla por medio de un import.
- Se puede determinar los datos a exportar con gran flexibilidad.
- Se pueden realizar exports completos, incrementales y acumulativos.
- Los backups relizados con export son portables y sirven como formato de intercambio de datos entre BDs y entre máquinas.

Una de las desventajas de realizar backups lógicos con export es que son mucho más lentos que los backups físicos.

Tabla 4.2.3.1 Parámetros de Export

Parámetro	*Defecto*	*Descripción*
USERID	indefinido	el username/password del usuario que efectúa el *export*.
BUFFER	dependiente del SO	El tamaño en bytes del buffer utilizado.
FILE	expdat.dmp	el nombre del fichero destino.
GRANTS	Yes	indica si se exportan también los derechos.
INDEXES	Yes	indica si se exportan también los índices.
ROWS	Yes	indica si se exportan también las filas de las tablas, o sólo las definiciones de las tablas.
CONSTRAINTS	Yes	indica si se exportan también las restricciones.
COMPRESS	Yes	indica si se exporta en modo comprimido.
FULL	No	indica si se exporta la BD entera.
OWNER	usuario actual	una lista de usuarios cuyos objetos se quieren exportar.
TABLES	indefinido	la lista de tablas a exportar.
RECORDLENGTH	dependiente del SO	la longitud en bytes del registro del fichero.
INCTYPE	indefinido	el tipo de *export* incremental.
RECORD	Yes	indica si se anota el *export* incremental en las tablas SYS.INCVID y en SYS.INCEXP.
PARFILE	indefinido	el fichero de parámetros.

Modos de Export

Existen tres modos de realizar una exportación de datos:

Modo Tabla

Exporta las definiciones de tabla, los datos, los derechos del propietario, los índices del propietario, las restricciones de la tabla y los disparadores asociados a la tabla.

Modo Usuario

Exporta todo lo del modo de Tabla más los clusters, enlaces de BD, vistas, sinónimos privados, secuencias, procedimientos, etc. del usuario.

Modo BD Entera

Además de todo lo del modo Usuario, exporta los roles, todos los sinónimos, los privilegios del sistema, las definiciones de los tablespaces, las cuotas en los tablespaces, las definiciones de los segmentos de rollback, las opciones de auditoría del sistema, todos los disparadores y los perfiles.

El modo BD entera puede ser dividido en tres casos: Completo, Acumulativo e Incremental. Estos dos últimos se toman menos tiempo que el completo, y permiten exportar sólo los cámbios en los datos y en las definiciones.

Completo

Exporta todas las tablas de la BD e inicializa la información sobre la exportación incremental de cada tabla. Después de una exportación completa, no se necesitan los ficheros de exportaciones acumulativas e incrementales de la BD anteriores.

```
$ exp userid=system/manager full=y inctype=complete
constraints=Y
file=full_export_filename
```

Acumulativo

Exporta solo las tablas que han sido modificadas o creadas desde la última exportación Acumulativa o Completa, y registra los detalles de exportación para cada tabla exportada. Después de una exportación acumulativa, no se necesitan los ficheros de exportaciones incrementales de la BD anteriores.

```
$ exp userid=system/manager full=y inctype=cumulative
constraints=Y
file=cumulative_export_filename
```

Incremental

Exporta todas las tablas modificadas o creadas desde la última exportación Incremental, Acumulativa o Completa, y registra los detalles de exportación para cada tabla exportada. Son interesantes en entornos en los que muchas tablas permanecen estáticas por periodos largos de tiempo, mientras que otras varían y necesitan ser copiadas. Este tipo de exportación es útil cuando hay que recuperar rápidamente una tabla borrada por accidente.

```
$ exp userid=system/manager full=y inctype=incremental
constraints=Y
file=incremental_export_filename
```

La política de exportación puede ser la siguiente: realizar una exportación completa el día 1 (por ejemplo el domingo), y luego realizar exportaciones incrementales el resto de la semana. De este modo de lunes a sábado sólo se exportarán aquellas tablas exportadas, ahorrando tiempo en el proceso.

4.3.Principios de Recuperación

Para entender los principios de la recuperación, se necesita entender las estructuras de datos subyacentes utilizadas en la recuperación.

4.3.1. Definiciones y Conceptos

Los ficheros redo log contienen los cambios realizados sobre la BD. Conviene presentar algunos conceptos relacionados con ellos.

Vector de Cambio
describe un cambio simple en un bloque de datos de la BD. Entre otros datos, contiene el número de versión, el código de la transacción, y la dirección del bloque afectado.
Registro Redo log
es un conjunto de vectores de cambio que describen un cambio atómico sobre la BD. La transacción es también la unidad de recuperación.
Evolución de Redo log por día
se puede calcular ejecutando el comando archive log list en dos días consecutivos y calculando la diferencia del número de secuencia de los ficheros redo log, multiplicado por el tamaño de un fichero redo log:

```
SVRMGR> archive log list;
```

```
Database log mode              No Archive Mode
Automatic archival             Disabled
Archive destination            /opt/app/oracle/admin/demo/arch/arch.log
Oldest online log sequence     3
Current log sequence           5
```

System Change Number, SCN
es un dato que define la versión confirmada de la BD en este instante de tiempo. Cuando una transacción es confirmada, se le asigna un SCN que la identifica unívocamente. Los ficheros redo log son marcados con dos SCN. Cuando se abre un nuevo fichero redo log se le marca con un SCN, low SCN, que es uno mas que el SCN mayor del anterior fichero redo log; y su high SCN es puesto a infinito. Los SCN también se asocian al fichero de control, ya que cuando se para una BD, un tablespace o fichero de datos, se almacena para cada fichero de datos su stop SCN en el fichero de control.

Cambio de redo log
es el proceso mediante el cual se deja de utilizar un fichero redo log y el LGWR combia al siguiente fichero redo log disponible. Se puede hacer con el comando alter system switch logfile.

Checkpoints
son activados automáticamente durante el funcionamiento normal de la instancia, pero pueden ser activados manualmente con el comando alter system checkpoint local o alter system checkpoint global dependiendo si nos referimos a la instancia en la que estamos, o si queremos que afecte a todas las instancias activas, respectivamente. Cada checkpoint lleva implicito un SCN, y Oracle asegura que todos los cámbios con un SCN menor que el del checkpoint dado han sido escritos en el disco.

4.3.2. Métodos de Recuperación

Existen varios métodos de recuperación, pero todos ellos se basan en la aplicación de los registros de redo log.

Aplicación de Redo Log
Cuando una BD se arranca con el comando startup, la BD pasa por los estados nomount, mount y open. En este tercer estado, se verifica que se pueden abrir todos los ficheros de log y de datos. Si la BD se arranca por primera vez después de una caida, se necesitará efectuar una recuperación que consiste en dos pasos: avanzar la BD hacia adelante aplicando los registros redo log, deshacer las transacciones no confirmadas.

Cada fichero de datos tiene en su cabecera el último checkpoint efectuado, así como el fichero de control también lleva esa cuenta. El checkpoint lleva incluido el SCN. Este es conocido como SCN de inicio de fichero. Asociado a cada fichero de datos el fichero de control tiene el SCN de final, puesto inicialmente a infinito. El SCN de inicio se incrementa con cada checkpoint.

Cuando la BD se para en modo normal o inmediato iguala el SCN de parada para cada fichero de datos al SCN almacenado en cada fichero de datos. Cuando se abre otra vez la BD se realizan dos comprobaciones. La primera es mirar si el contador de checkpoints en la cabecera de los ficheros de datos coincide con el correspondiente del fichero de control. Si es así, se compara el SCN de inicio de cada fichero de datos con el SCN de final almacenado en el fichero de control. Si son iguales no se necesita recuperación en este fichero de datos. Como parte de la apertura se pone a infinito el SCN de final para ese fichero de datos.

Si la BD se paró con en modo abort no se ejecutó el checkpoint y el SCN de fin para los fichero de datos está a infinito. Así, durante la BD se abre, y suponiendo que el contador de checkpoints coincide, se comparan los SCN de inicio y de final, y como el último es infinito se efectura una recuperación aplicando los cambios almacenados en los ficheros redo log en línea para avanzar la BD, y los registros de roll back de los segmentos de roll back para deshacer las transacciones no confirmadas.

Si después de parar la BD se reemplaza un fichero de datos por su copia de seguridad, al arrancar la BD Oracle detecta que el contador de checkpoints del fichero de datos no coincide

con el almacenado en el fichero de control. Así, se tendrá que echar mano a los ficheros redo log archivados, empezando por aquel cuyo número de secuencia aparece en la cabecera del fichero de datos.

4.3.3. Recuperación Física

La utilización de una copia de backup de ficheros de datos siempre necesita de una recuperación física. También es así cuando un fichero de datos se pone offline sin un checkpoint.

Oracle detecta que se necesita una recuperación física cuando el contador de checkpoints de la cabecera del fichero de datos no coincide con el correspondiente contador de checkpoints del fichero de control. Entonces se hace necesario el comando recover. La recuperación comienza en el SCN menor de los ficheros de datos en recuperación, aplicando los registros de redo log a partir de él, y parando en el SCN de final mayor de todos los ficheros de datos.

Existen tres opciones para realizar una recuperacion física. La primera es una recuperación de BD donde se restaura la BD entera. La segunda es una recuperación de tablespace donde, mientras una parte de la BD está abierta, se puede recuperar un tablespace determinado. Esto significa que serán recuperados todos los ficheros de datos asociados al tablespace. El tercer tipo es la recuperación de un fichero de datos específico mientras el resto de la BD está abierta.

Requisitos para Utilizar Recuperación Física

La primera condición que se ha de poner para poder recuperar físicamente una BD es que ésta se esté utilizando en modo ARCHIVELOG. De otro modo, una recuperación completa puede que no sea posible. Si trabajamos con la BD en modo NOARCHIVELOG, y se hace una copia semanal de los ficheros de la BD, se debería estar preparado para perder, en el peor de los casos, el trabajo de la última semana si sucede un fallo. Ya que los ficheros de redo log contendrían un agujero y no se podía avanzar la BD hasta el instante anterior al fallo. En este caso el único medio para reconstruir la BD es hacerlo desde un export completo, recreando el esquema de la BD e importando todos los datos.

Recuperación de la BD

La BD debe estar montada pero no abierta. El comando de recuperación es el siguiente:

```
RECOVER [AUTOMATIC] [FROM 'localizacion'] [BD]
   [UNTIL CANCEL]
   [UNTIL TIME fecha]
   [UNTIL CHANGE entero]
[USING BACKUP CONTROLFILE]
```

Las opciones entre corchetes son opcionales:

AUTOMATIC hace que la recuperación se haga automáticamente sin preguntar al DBA por el nombre de los ficheros redo log. También se puede utilizar para este cometido el comando set autorecovery on/off. Los ficheros redo log deben estar en la localización fijada en

LOG_ARCHIVE_DEST y el formato del nombre de los ficheros debe ser el fijado en LOG_ARCHIVE_FORMAT.
FROM se utiliza para determinar el lugar donde están los ficheros redo log, si es distinto del fijado en LOG_ARCHIVE_DEST.
UNTIL sirve para indicar que se desea realizar una recuperación incompleta, lo que implica perder datos. Solo se dará cuando se han perdido redo log archivados o el fichero de control. Cuando se ha realizado una recuperación incompleta la BD debe ser abierta con el comando alter database open resetlogs, lo que produce que los redo log no aplicados no se apliquen nunca y se inicialice la secuencia de redo log en el fichero de control. Existen tres opciones para parar la recuperación:
UNTIL CANCEL permite recuperar un redo log cada vez, parando cuando se teclea CANCEL.
UNTIL TIME permite recuperar hasta un instante dado dentro de un fichero de redo log
UNTIL CHANGE permite recuperar hasta un SCN dado.
USING BACKUP CONTROLFILE utiliza una copia de seguridad del fichero de control para gobernar la recuperación.
Recuperación de un tablespace

La BD debe estar abierta, pero con el tablespace a recuperar offline. El comando de recuperación es el siguiente:

```
RECOVER [AUTOMATIC] [FROM 'localizacion']
   TABLESPACE nombre_tablespace [, nombre_tablespace]
```

Recuperación de un Fichero de Datos
La BD debe estar abierta o cerrada, dependiendo del fichero a recuperar. Si el fichero a recuperar es de un tablespace de usuario la BD puede estar abierta, pero con el fichero a recuperar offline. Si el fichero es del tablespace SYSTEM la BD debe estar cerrada, ya que no puede estar abierta con los ficheros del SYSTEM offline. El comando de recuperación es el siguiente:

```
RECOVER [AUTOMATIC] [FROM 'localizacion']
   DATAFILE nombre_fichero [, nombre_fichero]
```

Creando un Fichero de Control
Si el fichero de control ha resultado dañado y se ha perdido se puede utilizar una copia de seguridad del mismo o crear uno nuevo. El comando de creación de un nuevo fichero de control es CREATE CONTROLFILE. Este comando se puede ejecutar sólo con la BD en estado nomount. La ejecución del comando produce un nuevo fichero de control y el montaje automático de la BD.

Un comando interesante que ayuda a mantener los ficheros de control a salvo es el siguiente:

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

que produce un script que puede ser utilizado para generar un nuevo fichero de control y recuperar la BD, en caso necesario. El fichero de traza generado es el siguiente:

```
Dump file /opt/app/oracle/admin/demo/udump/demo_ora_515.trc
Oracle7 Server Release 7.3.2.3.0 - Production Release
With the distributed, replication and Spatial Data options
PL/SQL Release 2.3.2.3.0 - Production
ORACLE_HOME = /opt/app/oracle/product/7.3.2
System name:    SunOS
Node name:      cartan
Release:        5.5
Version:        Generic
Machine:        sun4m
Instance name: demo
Redo thread mounted by this instance: 1
Oracle process number: 7
Unix process pid: 515, image: oracledemo

Fri May 15 11:41:19 1998
Fri May 15 11:41:19 1998
*** SESSION ID:(6.2035) 1998.05.15.11.41.19.000
# The following commands will create a new control file and use
it
# to open the database.
# No data other than log history 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   "DEMO"   NORESETLOGS
NOARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 2
    MAXDATAFILES 30
    MAXINSTANCES 1
    MAXLOGHISTORY 100
LOGFILE
  GROUP 1 '/export/home/oradata/demo/redodemo01.log'  SIZE 2M,
  GROUP 2 '/export/home/oradata/demo/redodemo02.log'  SIZE 2M,
  GROUP 3 '/export/home/oradata/demo/redodemo03.log'  SIZE 2M
DATAFILE
  '/export/home/oradata/demo/system01.dbf',
  '/export/home/oradata/demo/rbs01.dbf',
  '/export/home/oradata/demo/rbs02.dbf',
  '/export/home/oradata/demo/rbs03.dbf',
```

```
  '/export/home/oradata/demo/temp01.dbf',
  '/export/home/oradata/demo/tools01.dbf',
  '/export/home/oradata/demo/users01.dbf'
;
# Recovery is required if any of the datafiles are restored
backups,
# or if the last shutdown was not normal or immediate.
RECOVER DATABASE
# Database can now be opened normally.
ALTER DATABASE OPEN;
```

4.3.4. Recuperación Lógica

Oracle dispone de la herramienta import para restaurar los datos de una BD a partir de los ficheros resultados de un export. Import lee los datos de los ficheros de exportación y ejecuta las sentencias que almacenan creando las tablas y llenándolas de datos.

Tabla 4.3.4.1 Parámetros del Import

Parámetro	*Defecto*	*Descripción*
USERID	indefinido	el username/password del usuario que efectúa el *import*.
BUFFER	dependiente del SO	El tamaño en bytes del buffer utilizado.
FILE	expdat.dmp	el nombre del fichero de exportación a importar.
SHOW	No	indica si se muestran los contenidos del fichero de exportación, sin importar ningún dato.
IGNORE	Yes	indica si ignorar los errores producidos al importar un objeto que ya existe en la BD.
GRANTS	Yes	indica si se importan también los derechos.
INDEXES	Yes	indica si se importan también los índices.
ROWS	Yes	indica si se importan también las filas de las tablas.
FULL	No	indica si se importan el fichero entero.
FROMUSER	Indefinido	una lista de los usuarios cuyos objetos se han exportado.
TOUSER	Indefinido	una lista de los usuarios a cuyo nombre se importan los objetos.
TABLES	indefinido	la lista de tablas a importar.
RECORDLENGTH	dependiente del SO	la longitud en bytes del registro del fichero.

INCTYPE	indefinido	el tipo de *import* incremental (SYSTEM o RESTORE).
COMMIT	No	indica si se efectúa un commit después de importar cada fila. Por defecto, *import* efectúa un commit después de cargar cada tabla.
PARFILE	indefinido	el fichero de parámetros.

Para importar un export incremental se puede efectuar la siguiente secuencia de pasos:

Utilizar la copia más reciente del import para restaurar las definiciones del sistema:

```
$ imp userid=sys/passwd inctype=system full=Y
file=export_filename
```

Poner los segmentos de rollback online.
Importar el fichero de exportación completa más reciente:

```
$ imp userid=sys/passwd inctype=restore full=Y file=filename
```

5. REFERENCIAS BIBLIOGRAFICAS Y ELECTRONICAS

REFERENCIAS BIBLIOGRAFICAS

Loney, Kevin. *Oracle8 Manual del Administrador,* 3ª. Edición, España, Mc Graw Hill, 2000.

Abbey, Michael; Abramson, Ian; Corey, Michael J. *Oracle 8i Guia de Aprendizaje,*3ª. Edición, España, Mc Graw Hill, 2000.

Velpuri Rama. *Oracle8 Manual de Backup y Recuperación de Datos,* 2ª. Edición, México, Mc Graw Hill, 1999.

Theriault, Marlene; Newman Aarón. *Oracle Security Handbook,* 2ª. Edición, U.S.A., Mc Graw Hill, 2001.

Scherer, Douglas. *Oracle8i Tips & Techniques,* 2ª. Edición, U.S.A., Mc Graw Hill, 2000.

REFERENCIAS ELECTRONICAS

Oracle, http://www.oracle.com/es/ 12/01/2002
Oracle, http://www.oracle.com/dk/ 12/01/2002
UAM, http://software.uaemex.mx/documentos/oracle/IFS.htm 16/12/2001
OracleProd,http://www.add.es/productos/AREA1/ORACLE/HTML/oacle8i_2.html 14/12/2001
Oracle, http://sqltech.cl/doc/oracle8i/addendum.817/a85455/toc.htm 12/01/2002
Todo Oracle, http://sqltech.cl/doc/oracle8i/em.817/a85248/toc.htm 9/01/2002
Oracleinf, http://sqltech.cl/doc/oracle8i/ois.817/a65435/toc.htm 16/12/2001
Oracleinf, http://sqltech.cl/doc/oracle8i/ois.817/a65435/using_ad.htm#1012573 16/12/2001
Oracleinf, http://sqltech.cl/doc/oracle8i/ois.817/a65435/security.htm#1012573 4/01/2002
Oracleinf, http://sqltech.cl/doc/oracle8i/addendum.817/a85455/toc.htm 4/01/2002
Oracleinf, http://sqltech.cl/doc/oracle8i/server.817/a76956/audit.htm#1528 4/01/2002