

WWW.ACLODU.DO



**ACADEMIA
CÓDIGO
LIBRE**

LIBROS PRACTICAS MANUALES SCRIPTS VIDEOS

Manual GNU Certified Oracle Administrator



GCOA
Ing. José Paredes

www.codigolibre.org

GNU Certified Oracle Administrator GCOA

Academia Código Libre
Mayo 2013.-

Documentación realizada por:

Ing. José Paredes
GNU/Instructor DBA

Valentín Jiménez
GNU/Instructor DBA

Meidy Alvarez
GNU/Instructor DBA

Capítulos

Capítulo 1- Instalación y Manejo Del S.O GNU/Linux.....	7
Capítulo 2- Comandos GNU/Unix.....	30
Capítulo 3- Instalación de Oracle 11gR2.....	45
Capítulo 4- Estructuras de Almacenamiento.....	67
Capítulo 5- Introducción SQL.....	73
Capítulo 6- Diccionario de Datos.....	79
Capítulo 7- Arranque y Parada.....	84
Capítulo 8- Procesos Background.....	93
Capítulo 9- Fichero de Control.....	97
Capítulo 10- Los Datafiles.....	99
Capítulo 11- TablesPaces.....	101
Capítulo 12- Redo Log.....	107
Capítulo 13- Modos de Operación de la BD.....	112
Capítulo 14- Usuarios, roles, privilegios y perfiles.....	115
Capítulo 15- Configuración del Entorno de Red de Oracle.....	127
Capítulo 16- Oracle Flashback.....	137
Capítulo 17- Copias de Seguridad y Recuperación.....	143
Capítulo 18- Copia logica export/import.....	145
Capítulo 19- SQL*LOADER.....	153
Capítulo 20- Automatización de Oracle en GNU/Linux.....	155
Capítulo 21- Instalación de Clientes.....	163
Capítulo 22- Consejo para un GNU/Dba.....	169
Capítulo 23- LVM (volúmenes lógicos).....	175
Capítulo 24- ASM (Automatic Storage Management).....	181
Capítulo 25- Rman Oracle Recovery Manager.....	191
Capítulo 26- Herramientas GUI para Oracle.....	243

Copyright

Esta documentación es regida por los reglamentos de la GNU Free Documentation License y la GNU General Public License v3, siéntase libre de copiar, modificar, aprender, distribuir y redistribuir esta documentación; haga uso de su derecho como si fuere su deber.

Prefacio

En estas páginas se plasma de manera informativa los pasos para convertirse en un administrador de base de datos, pero usando Oracle Database; la finalidad de la documentación no es aprender la tecnología privativa de Oracle, la finalidad es aprovechar los conceptos y aplicarlos a este motor de datos y otros motores libres.

Esta es la primera etapa de varias en este camino hacia la carrera profesional de la administración de base de datos y análisis de estructuras de almacenamiento de registros de datos.

Instalación y Manejo Del S.O GNU/Linux

Objetivos: En este capítulos tratamos de conocer sobre GNU/Linux.

Historia

GNU/Linux es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado Linux con el sistema operativo GNU. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL (Licencia Pública General de GNU, en inglés: General Public License) y otra serie de licencias libres.

A pesar de que Linux es, en sentido estricto, el sistema operativo, parte fundamental de la interacción entre el núcleo y el usuario (o los programas de aplicación) se maneja usualmente con las herramientas del proyecto GNU y con entornos de escritorio basados en GNOME, que también forma parte del proyecto GNU aunque tuvo un origen independiente. Sin embargo, una parte significativa de la comunidad, así como muchos medios generales y especializados, prefieren utilizar el término Linux para referirse a la unión de ambos proyectos. Para más información consulte la sección "Denominación GNU/Linux" o el artículo "Controversia por la denominación GNU/Linux".

A las variantes de esta unión de programas y tecnologías, a las que se les adicionan diversos programas de aplicación de propósitos específicos o generales se las denomina distribuciones. Su objetivo consiste en ofrecer ediciones que cumplan con las necesidades de un determinado grupo de usuarios. Algunas de ellas son especialmente conocidas por su uso en servidores y super computadoras. donde tiene la cuota más importante del mercado. Según un informe de IDC, GNU/Linux es utilizado por el 78% de los principales 500 servidores del mundo, otro informe le da una cuota de mercado de 89% en los 500 mayores super computadores. Con menor cuota de mercado el sistema GNU/Linux también es usado en el segmento de las computadoras de escritorio, portátiles, computadoras de bolsillo, teléfonos móviles, sistemas embebidos, videoconsolas y otros dispositivos.

Licencias

Una licencia de software es un contrato entre el licenciatario (autor/titular de los derechos de explotación/distribuidor) y el licenciatario del programa informático (usuario consumidor /usuario profesional o empresa), para utilizar el software cumpliendo una serie de términos y condiciones establecidas dentro de sus cláusulas.

Las licencias de software pueden establecer entre otras cosas: la cesión de determinados derechos del propietario al usuario final sobre una o varias copias del programa informático, los límites en la responsabilidad por fallos, el plazo de cesión de los derechos, el ámbito geográfico de validez del contrato e incluso pueden establecer determinados compromisos del usuario final hacia el propietario, tales como la no cesión del programa a terceros o la no reinstalación del programa en equipos distintos al que se instaló originalmente.

GNU General Public License

La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente sus siglas del inglés GPL, es una licencia creada por la Free Software Foundation en 1989 (la primera versión, escrita por Richard Stallman), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Existen varias licencias "hermanas" de la GPL, como la licencia de documentación libre de GNU (GFDL), la Open Audio License, para trabajos musicales, etcétera, y otras menos restrictivas, como la MPL, o la LGPL (Lesser General Public License, antes Library General Public License), que permiten el enlace dinámico de aplicaciones libres a aplicaciones no libres.

Validez legal

La licencia GPL, al ser un documento que cede ciertos derechos al usuario, asume la forma de un contrato, por lo que usualmente se la denomina contrato de licencia o acuerdo de licencia. En los países de tradición anglosajona existe una distinción doctrinal entre licencias y contratos, pero esto no ocurre en los países de tradición civil o continental. Como contrato, la GPL debe cumplir los requisitos legales de formación contractual en cada jurisdicción.

La licencia ha sido reconocida, entre otros, por juzgados en Alemania, particularmente en el caso de una sentencia en un tribunal de Múnich, lo que indica positivamente su validez en jurisdicciones de derecho civil.

Distribuciones.

Al ser GNU/Linux un SO de fuente abierta, básicamente esto provee la libertad de poder personalizar el sistema a tu comodidad, esto ha permitido la generación de una diversidad de SO GNU/Linux cada una específica para un público y/o propósito en particular. La elección de una distribución viene condicionada por muchos factores. Hay muchas distribuciones actualmente y cada poco aparecen nuevas, entre las mas conocidas tenemos:

- **Debian** es una distribución totalmente libre desarrollada por un grupo muy numeroso de colaboradores en el más puro espíritu de Linux. Se basa en paquetes .deb
- **RedHat** y **CentOS** son actualmente las distribuciones más difundidas. El sistema de paquetes 'RPM' es muy bueno y utilizado como base por un gran número de distribuciones.
- **SuSE** Es una distribución comercial alemana que ha tenido un crecimiento espectacular. Ofrece un entorno muy amigable que facilita mucho la instalación. Seguramente es la más fácil de instalar y de mantener. Capacidad de auto detección de Hardware. Muy recomendable para cualquiera que no tenga muchos conocimientos de Linux.
- **Mandriva GNU/Linux** (fusión de la distribución francesa Mandrake Linux y la brasileña Conectiva Linux) es una distribución de Linux aparecida en 1998, enfocada a principiantes o usuarios medios.
- **Slackware** es una distribución totalmente libre y muy sencilla en el sentido de que está poco elaborada. Fue creada por Patrick Volkerding. Fue una de las primeras y tuvo su época de gran auge pero actualmente ha cedido protagonismo. Su sistema de paquetes se basa en los famosos tarballs (paquetes TAR.GZ), No dispone de un buen sistema de actualización.

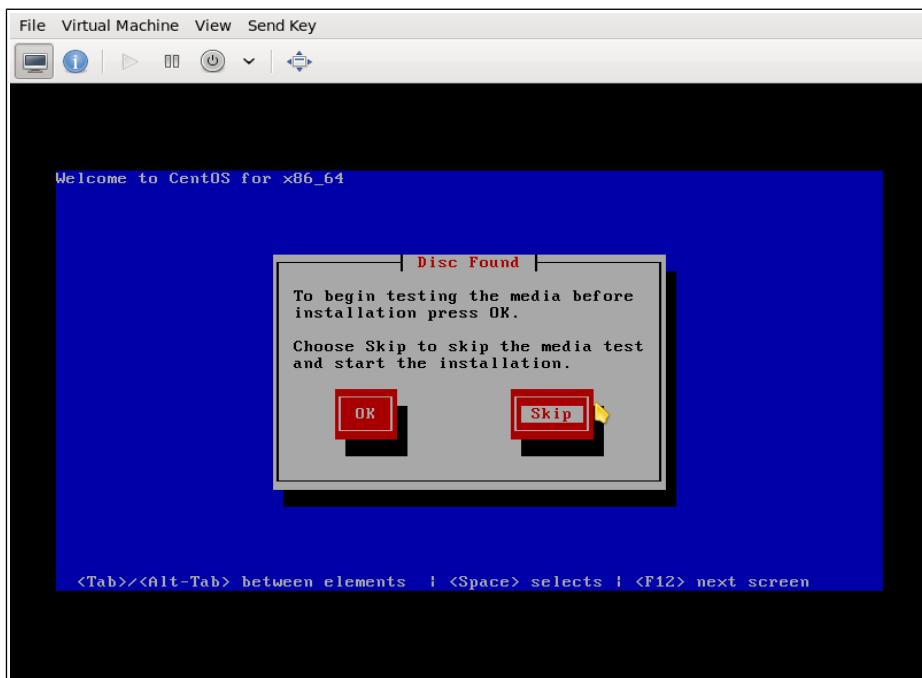
Particiones

Se recomienda estudiar muy bien el esquema de particionamiento de su instalación, por lo general, de manera no obligatoria pero recomendada bajo el concepto LVM:

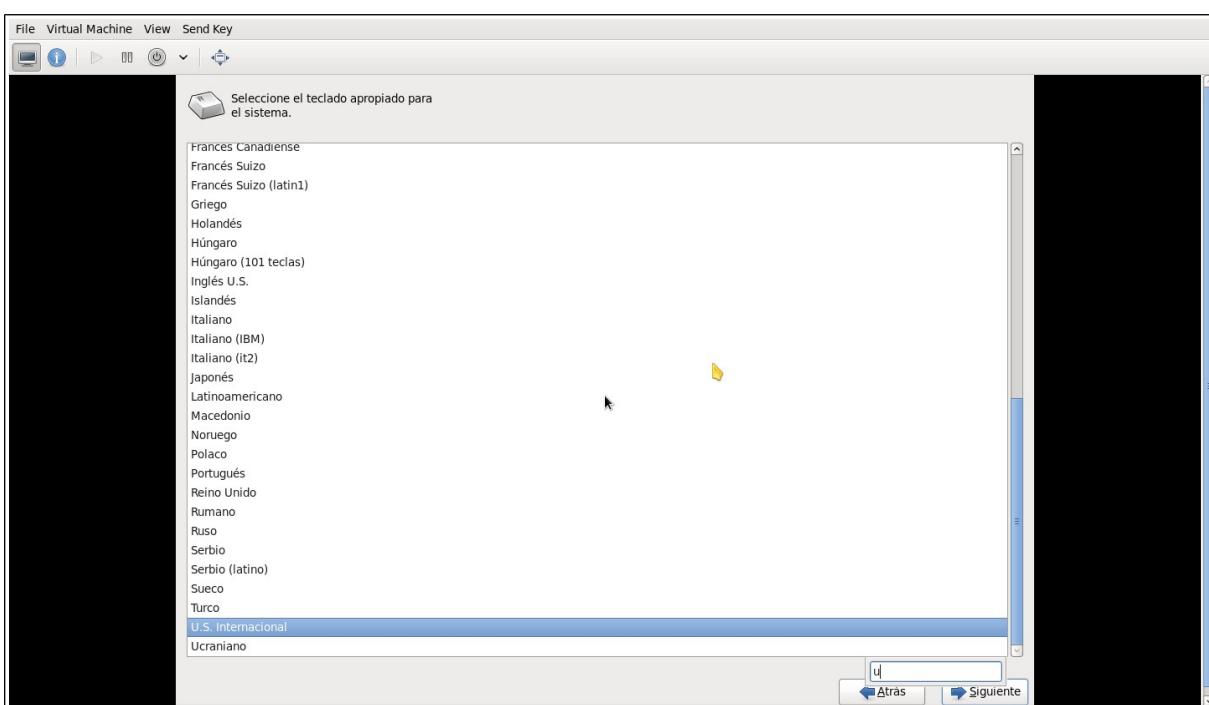
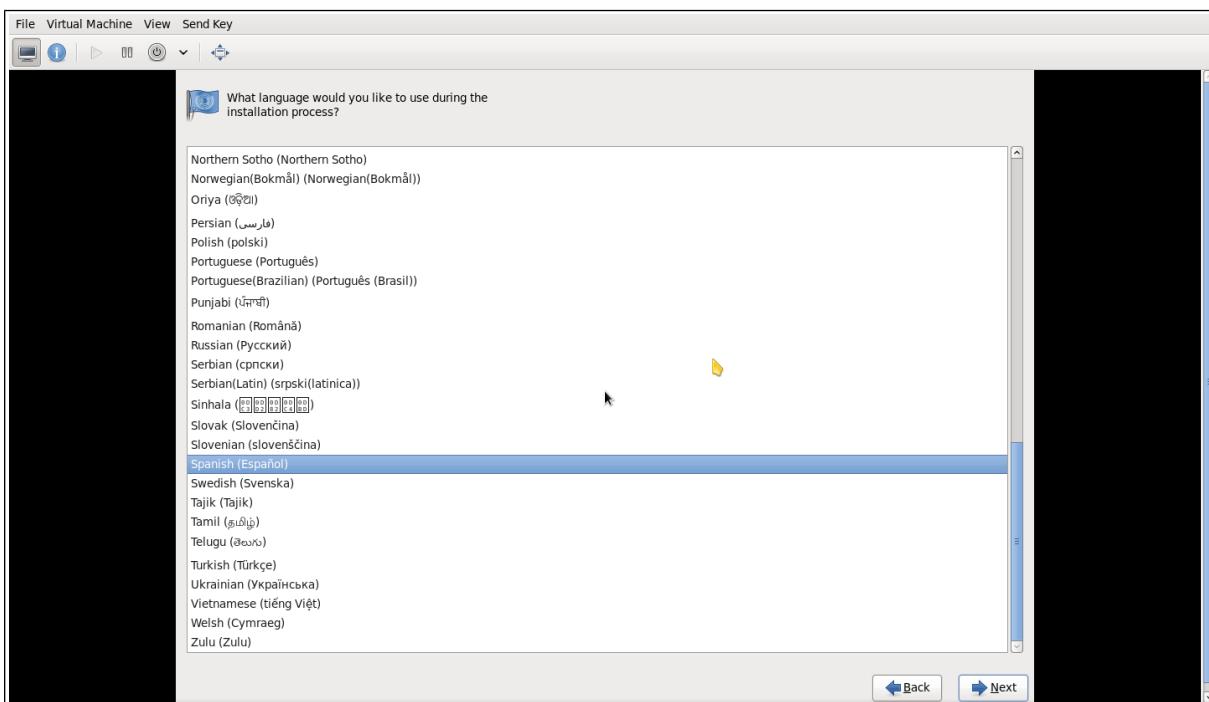
Esquemas de ejemplo con un disco de 500 GB y 4GB Ram.

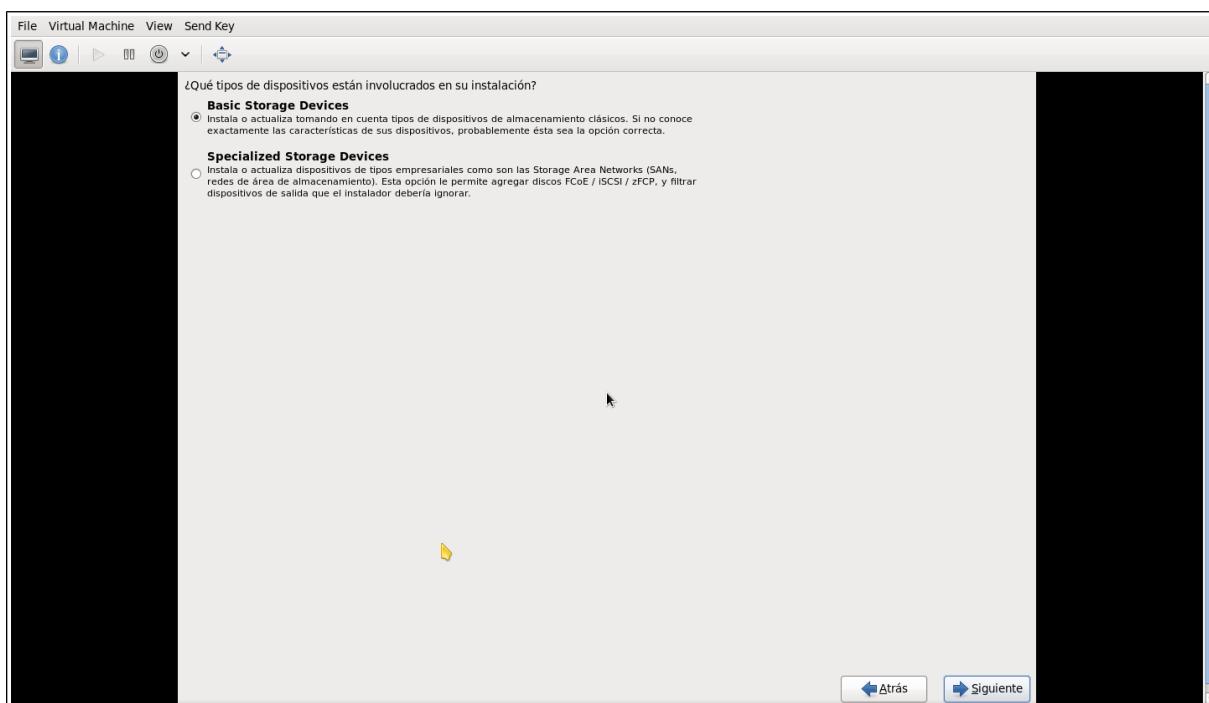
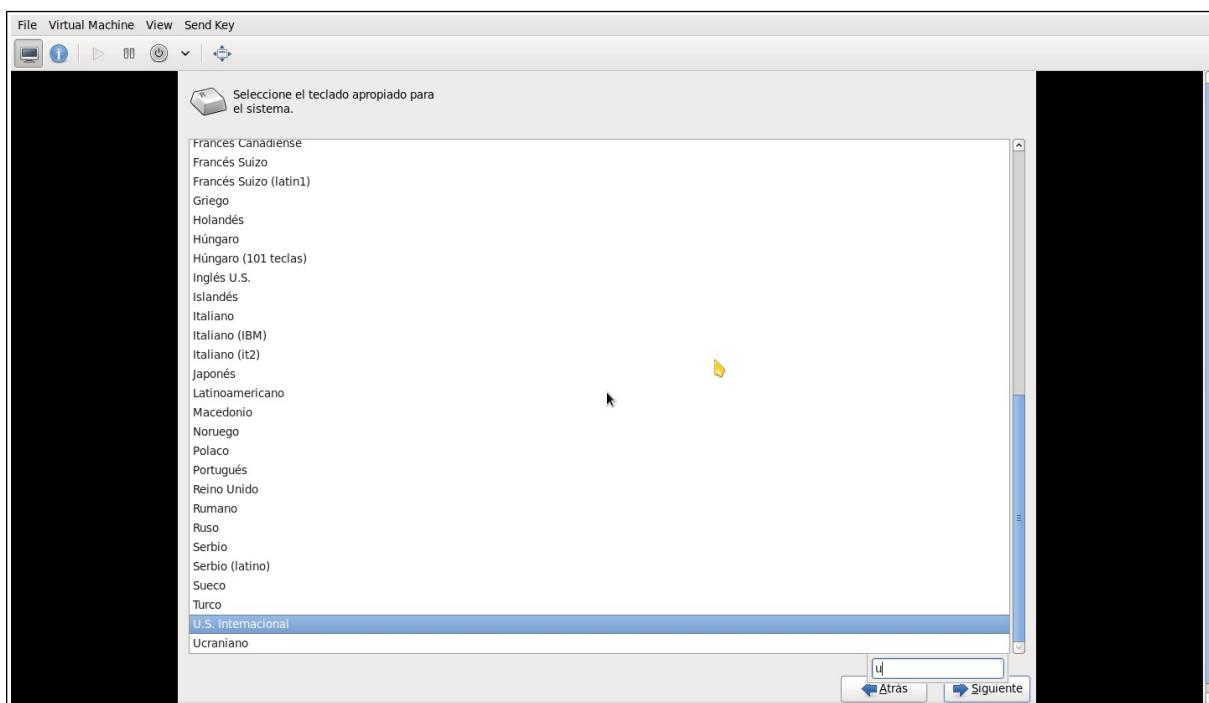
/boot	(500 MB)
/	(10% o 15%)
PV	Volúmenes físicos (PV)
VG	Grupos de volúmenes (VG)
VL	Volúmenes lógicos (LV)
swap	Debe asignarse el doble del tamaño del RAM físico. (20% o 30%)
/var	(5% o 10%)
/usr	(20%)
/home	(20%)
/opt	El restante del disco

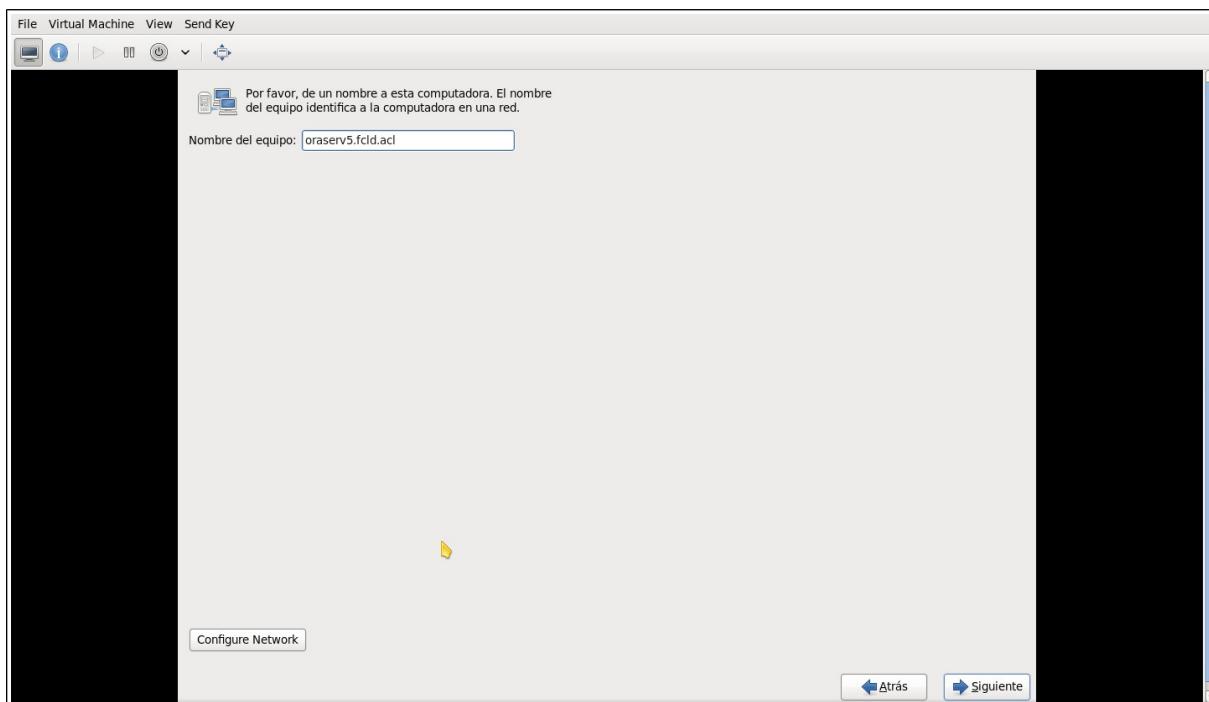
Pasos para la instalación GNU/Linux:

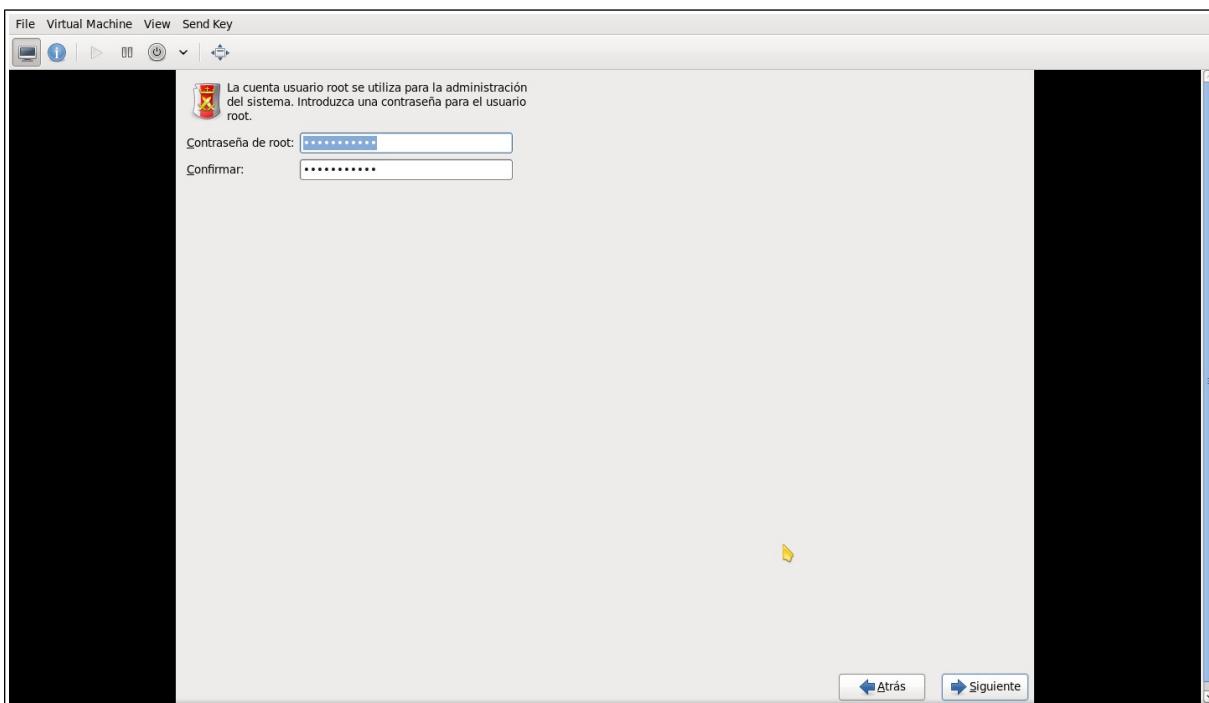
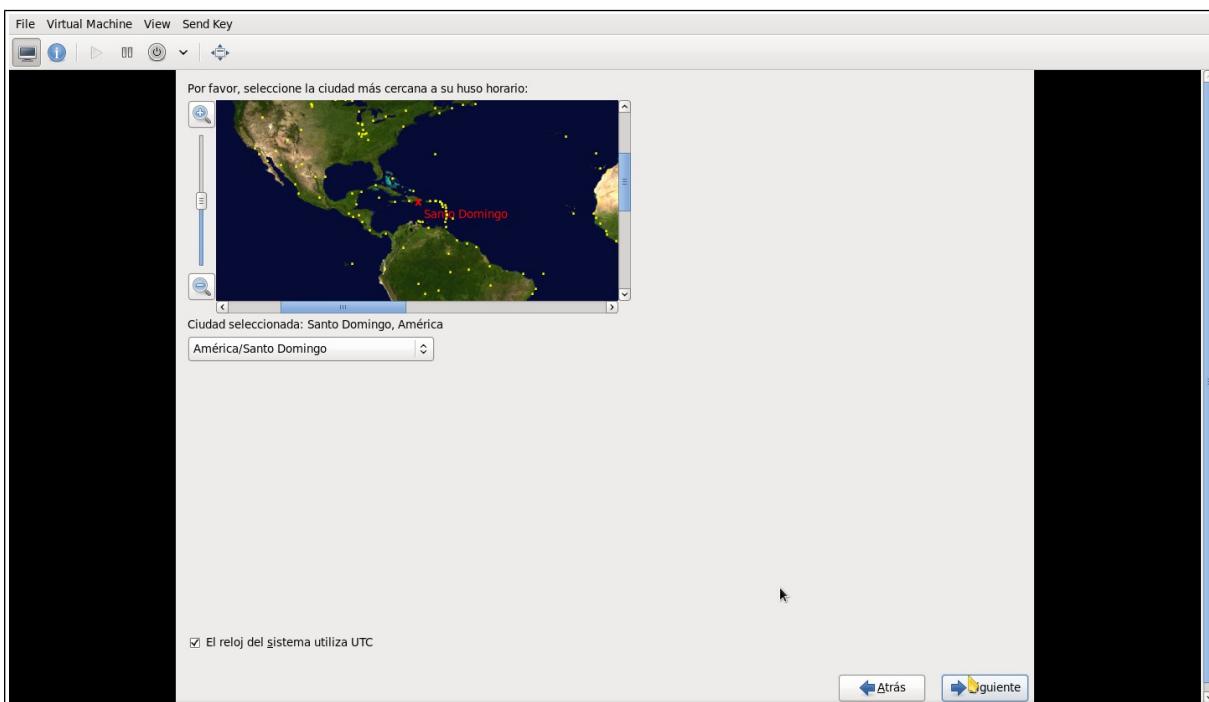


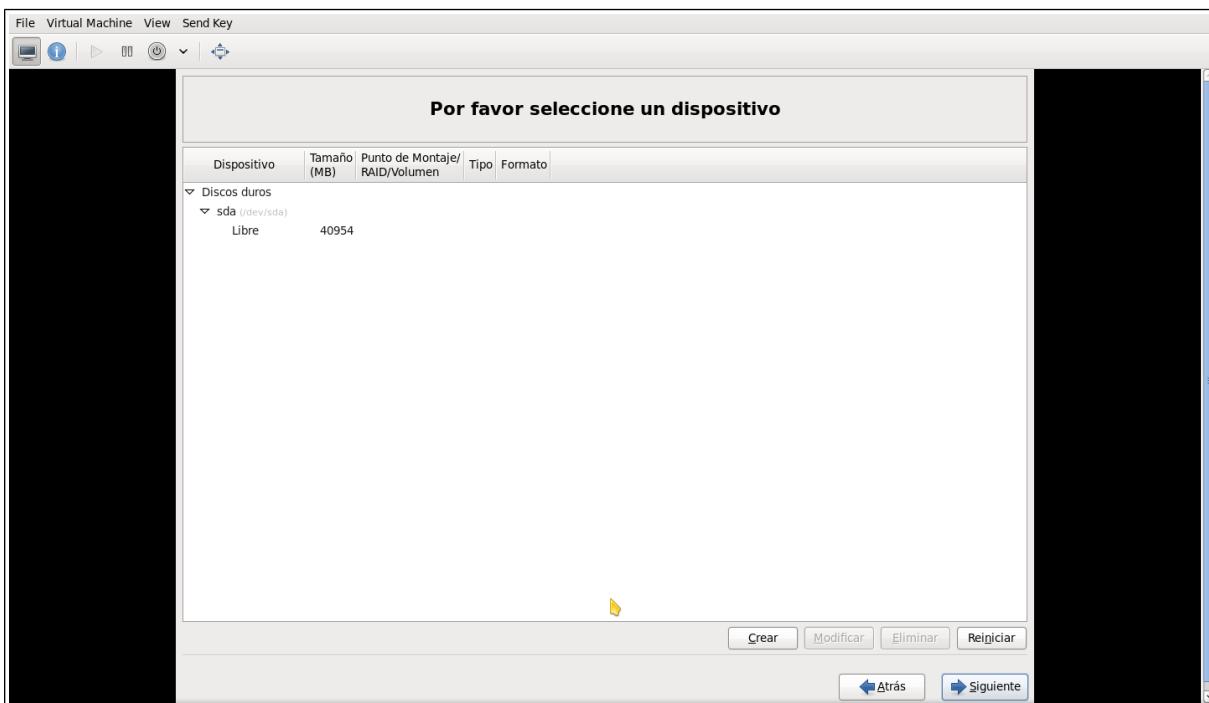
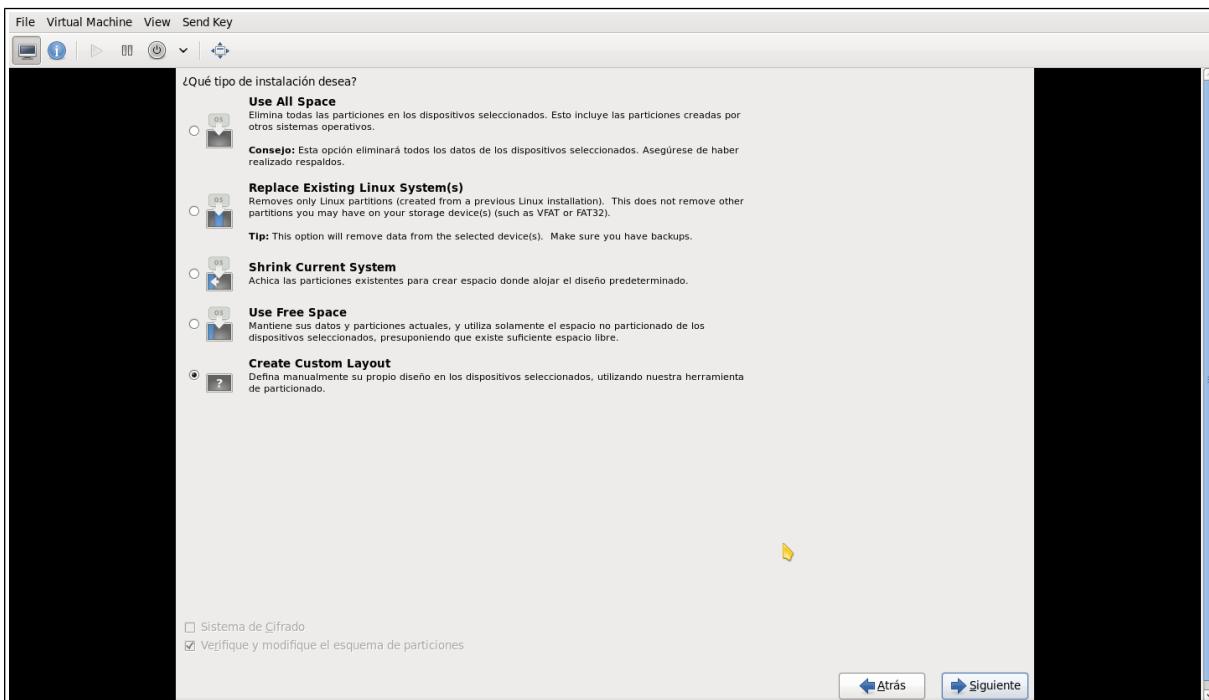


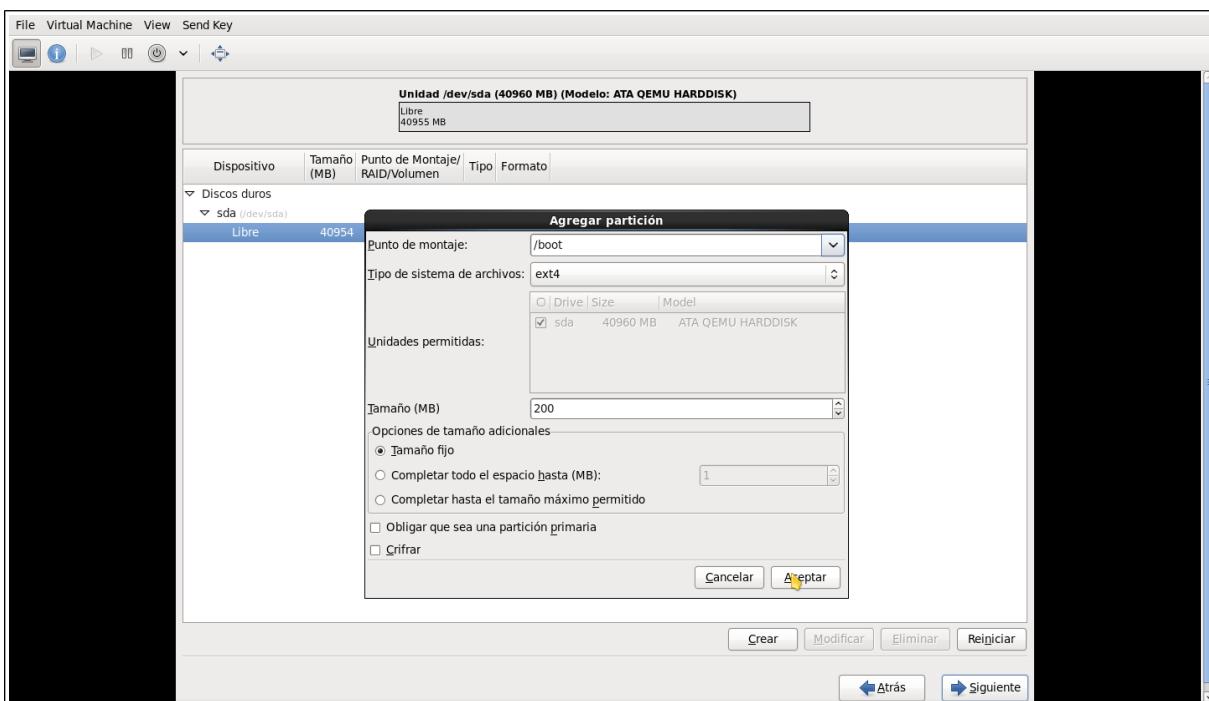
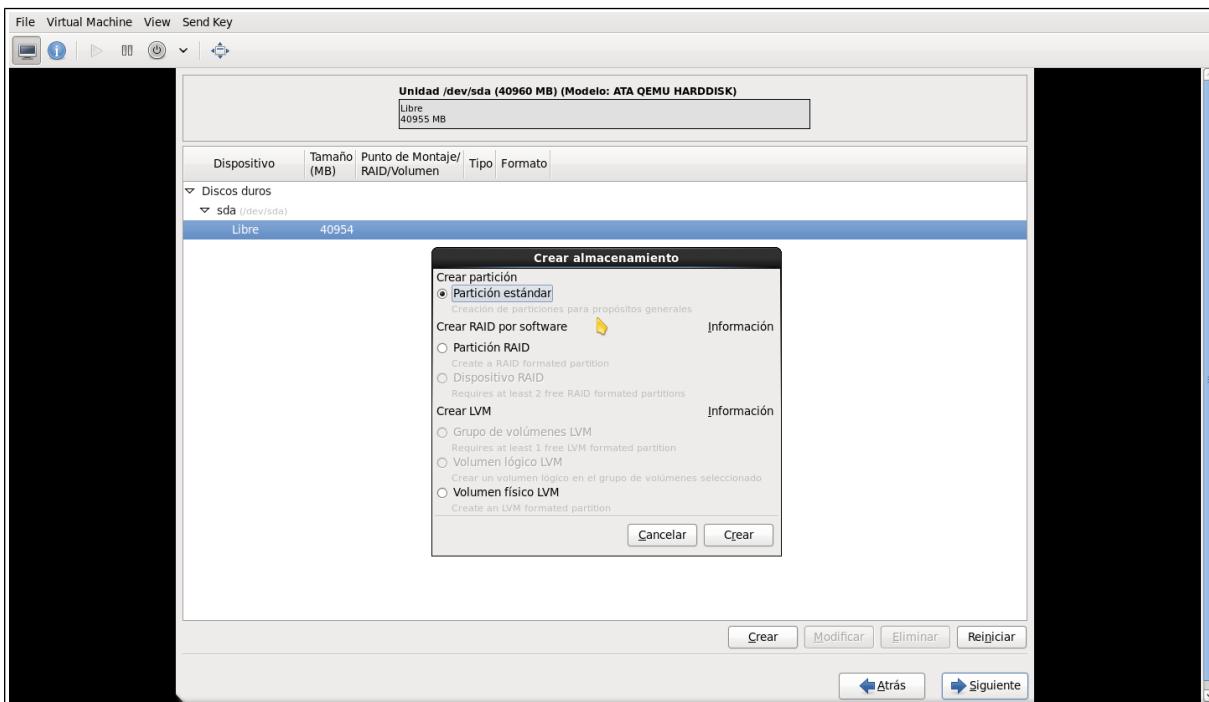


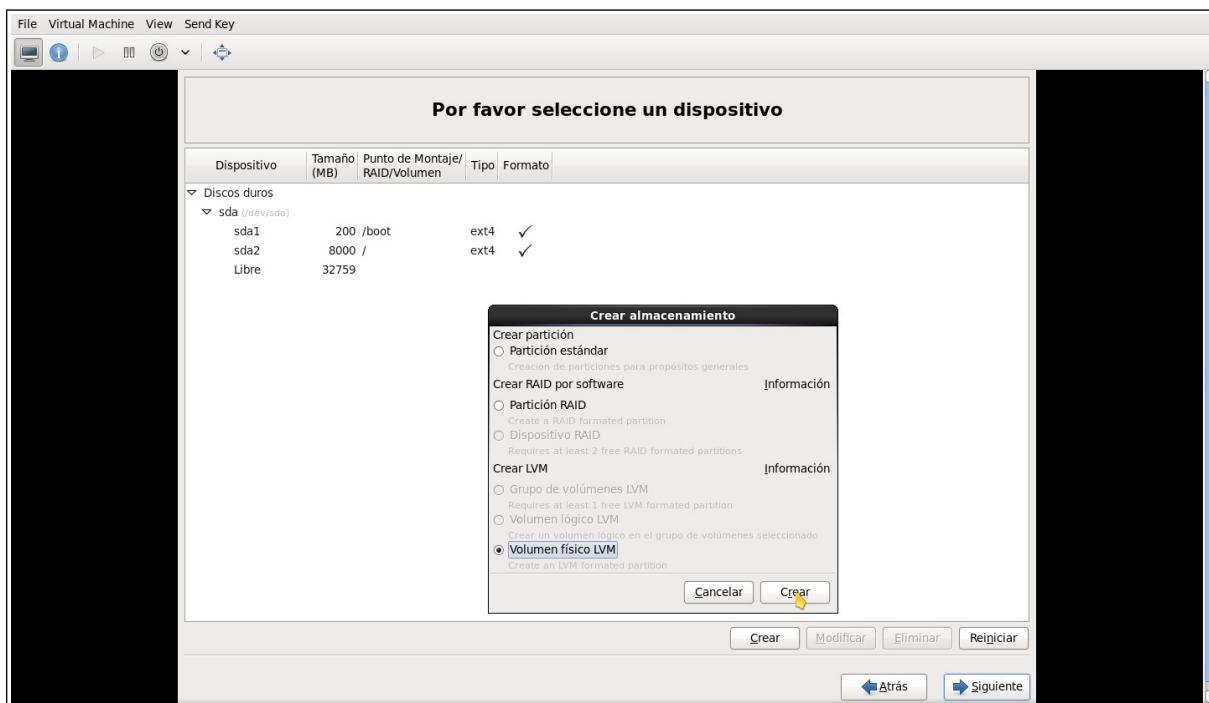
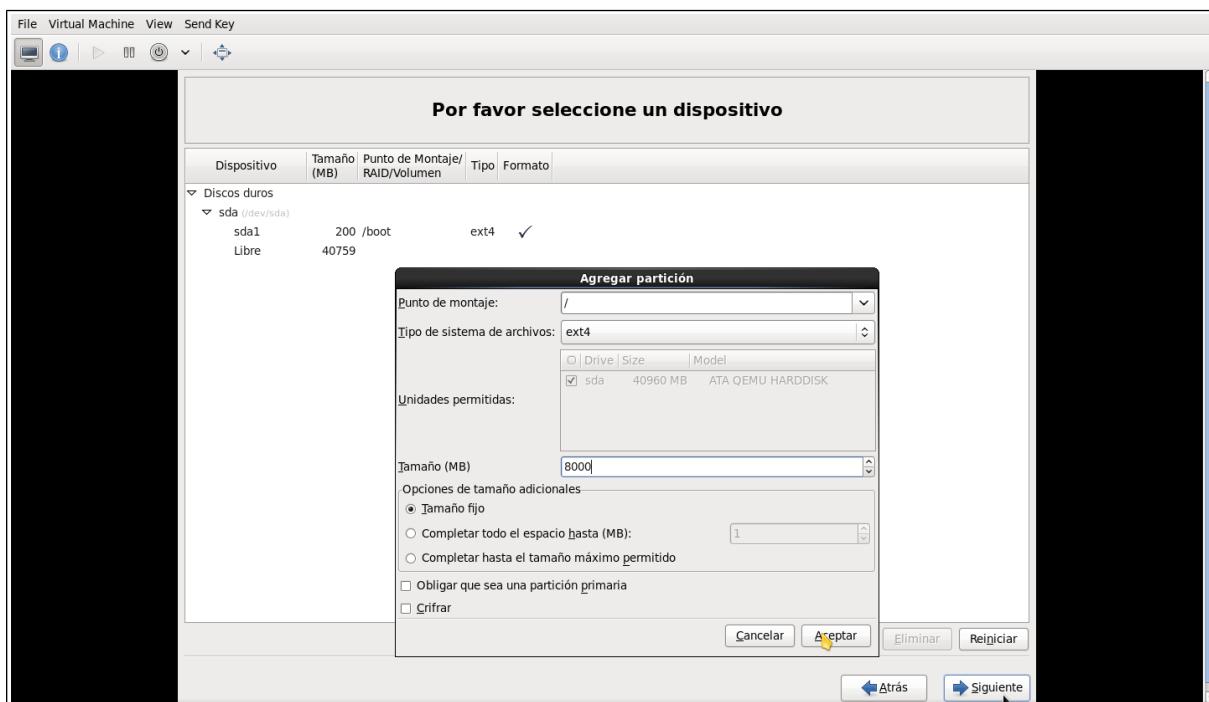


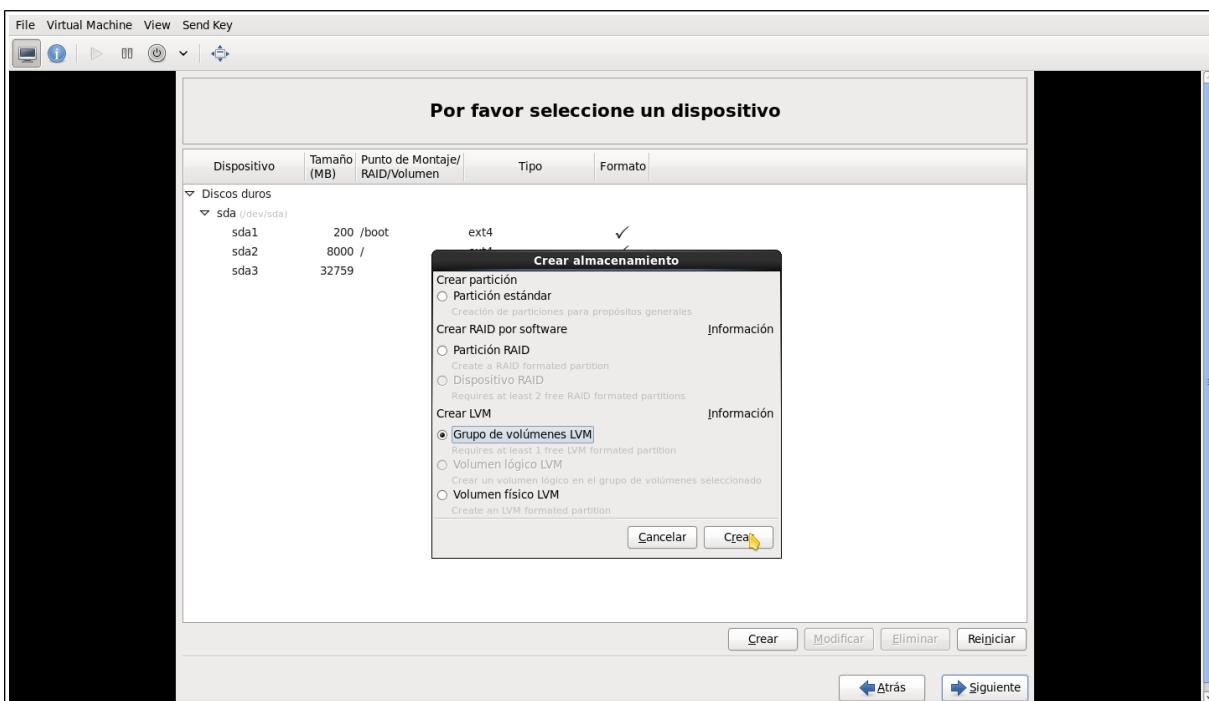
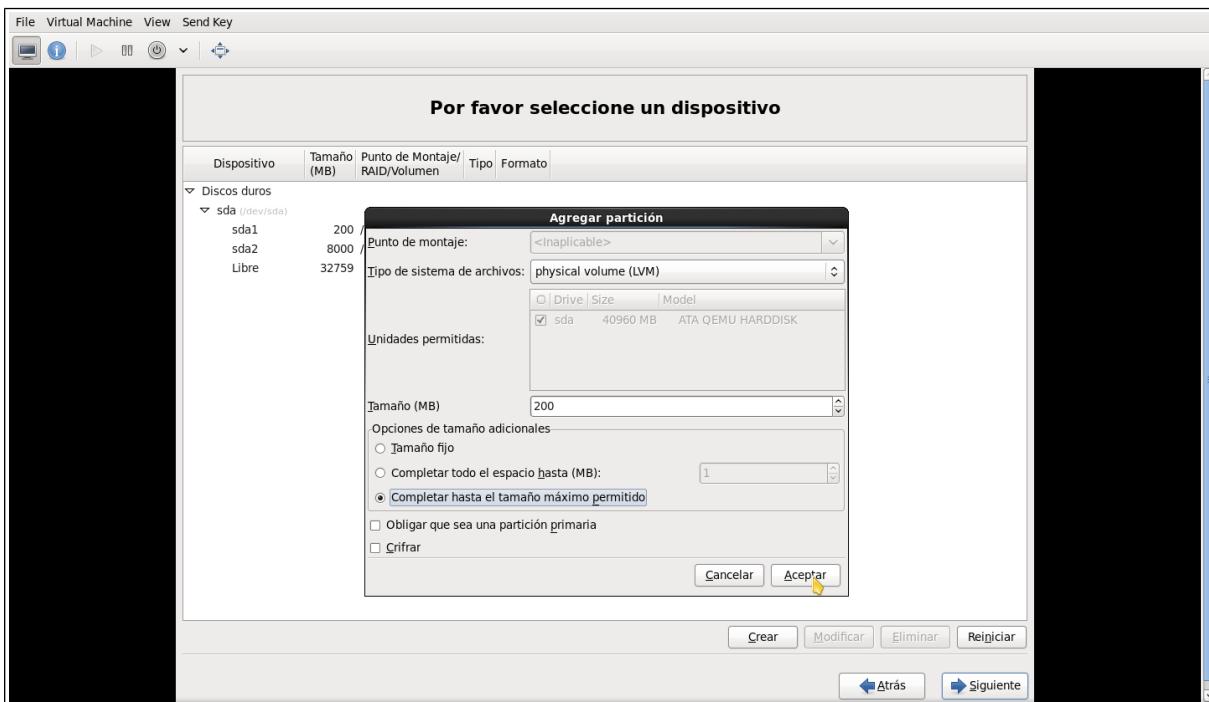


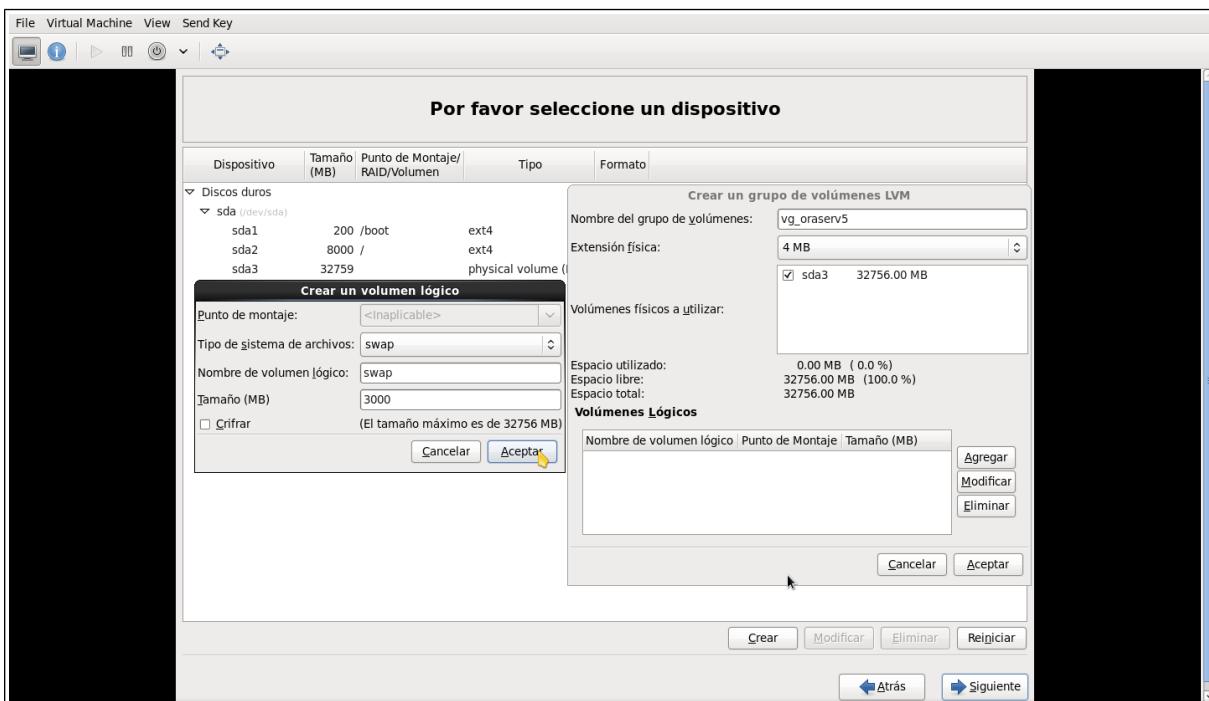
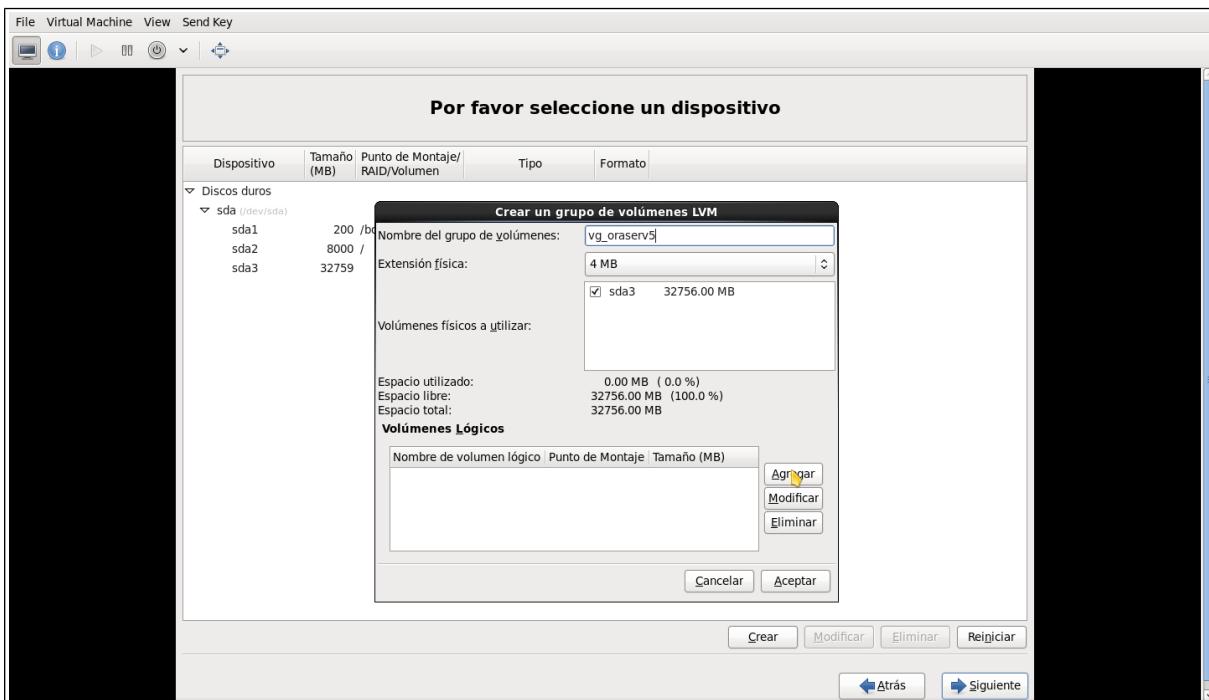


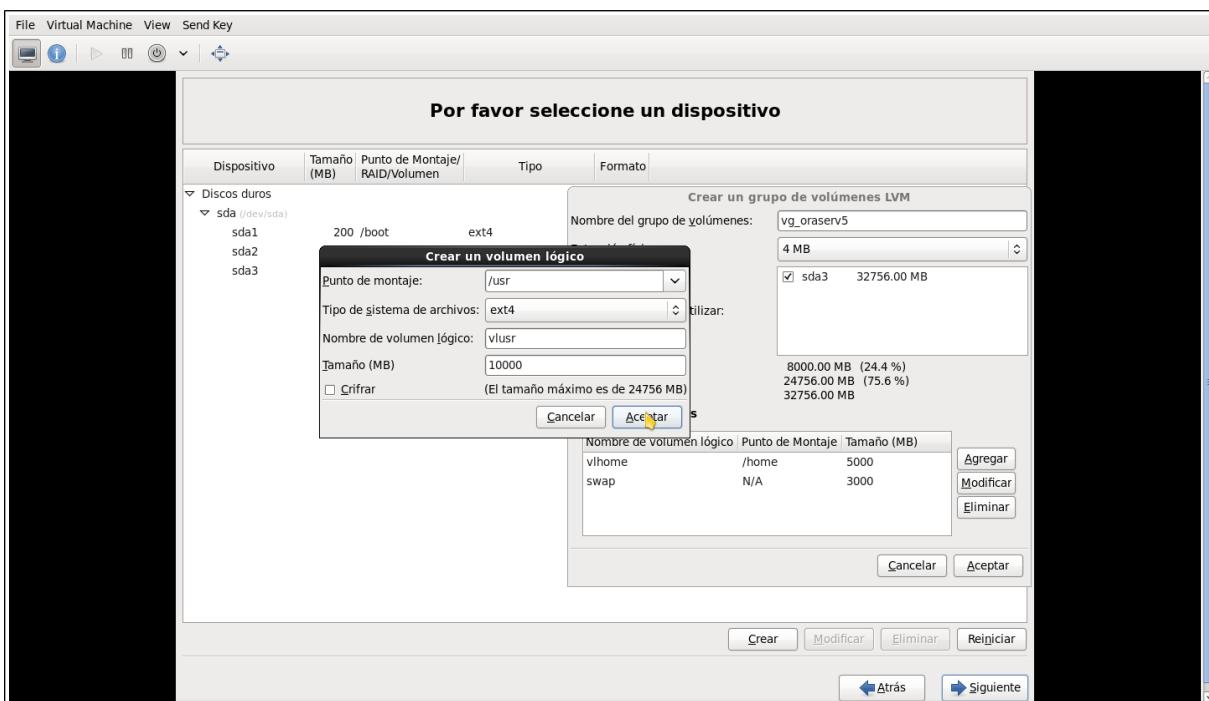
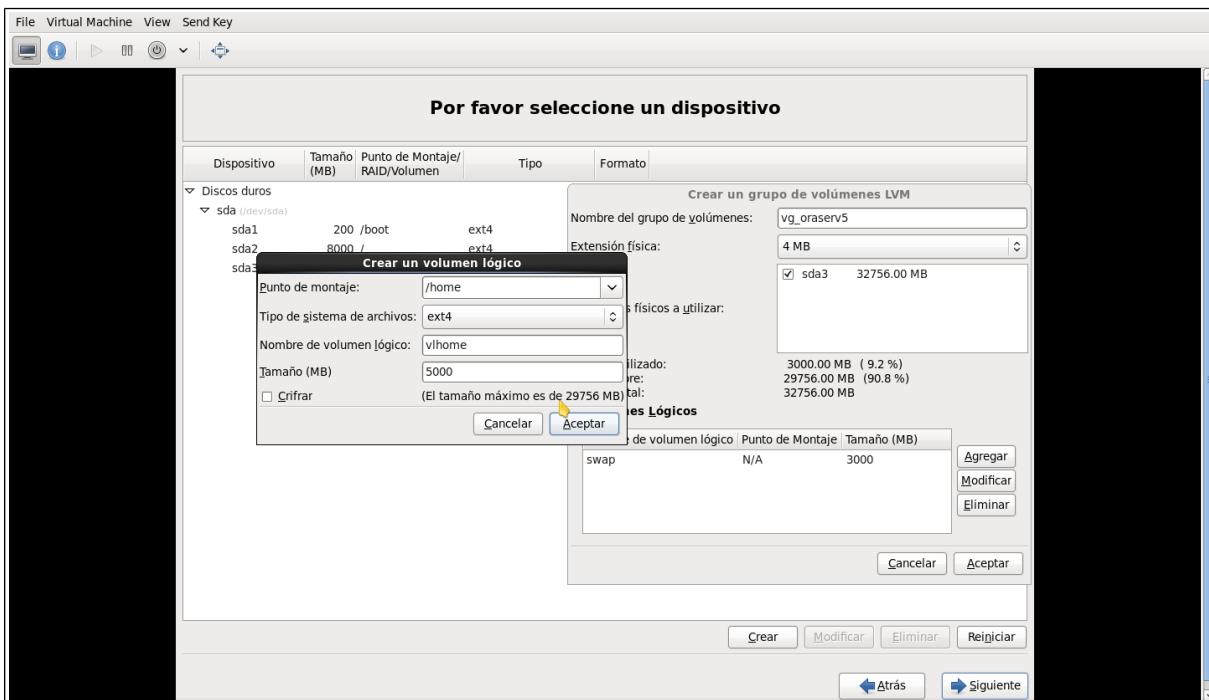


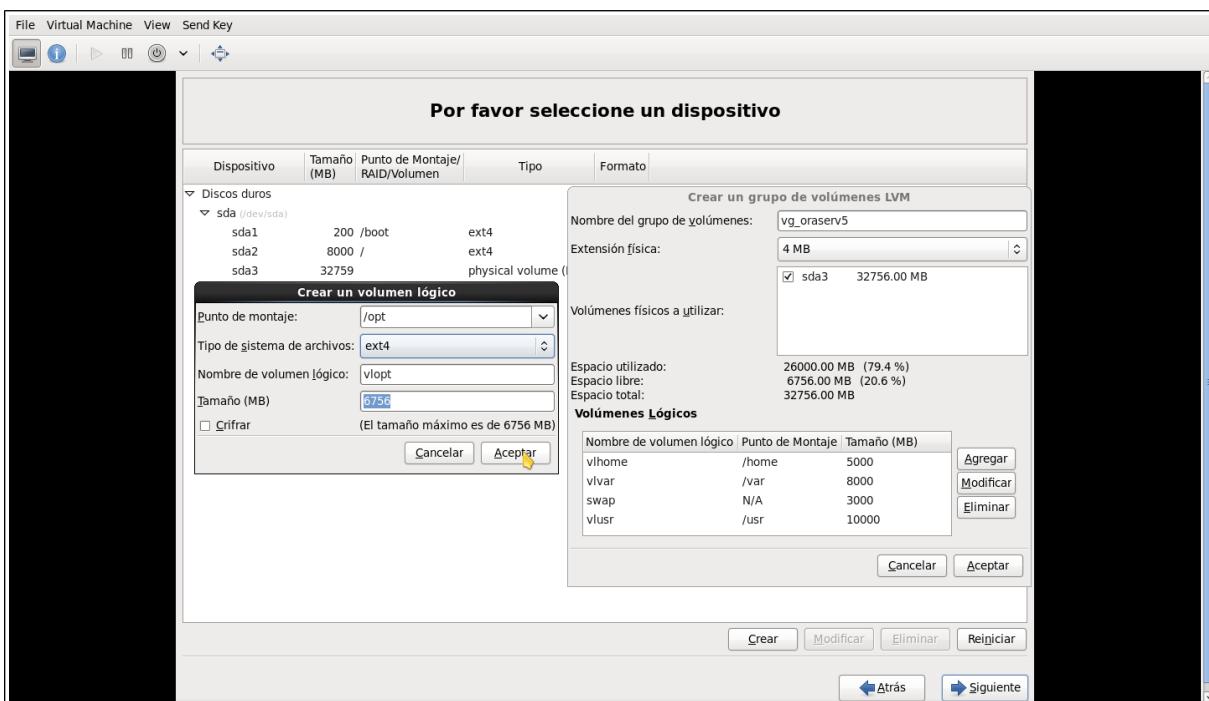
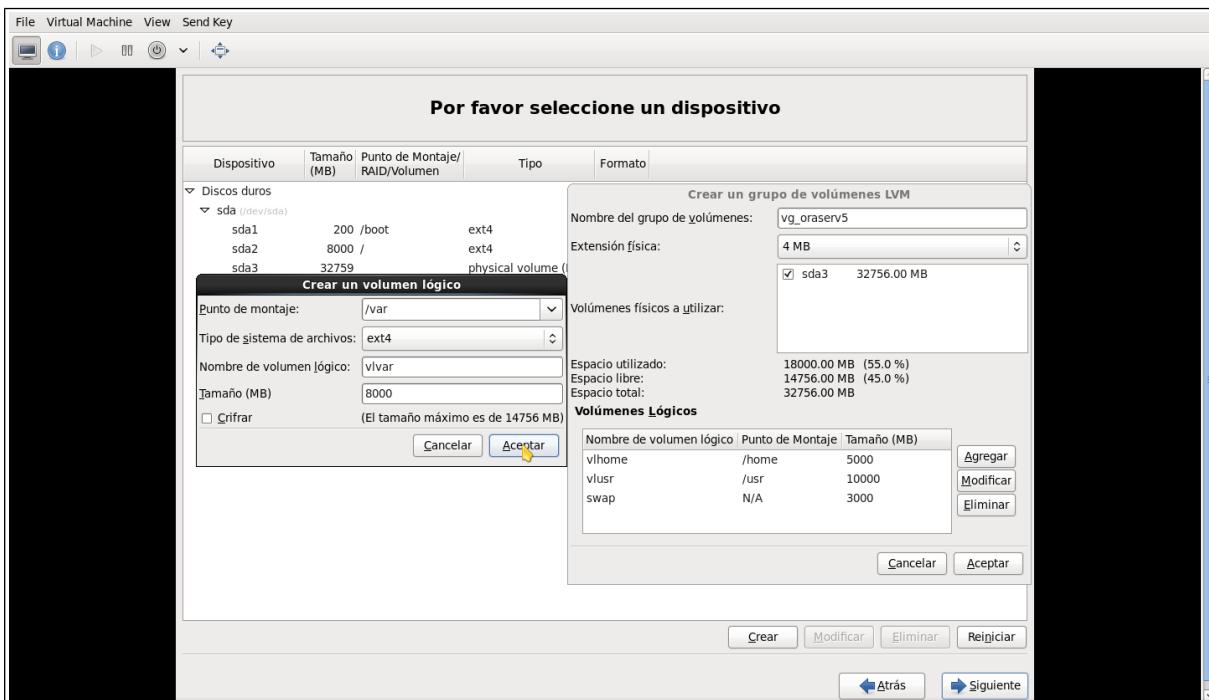


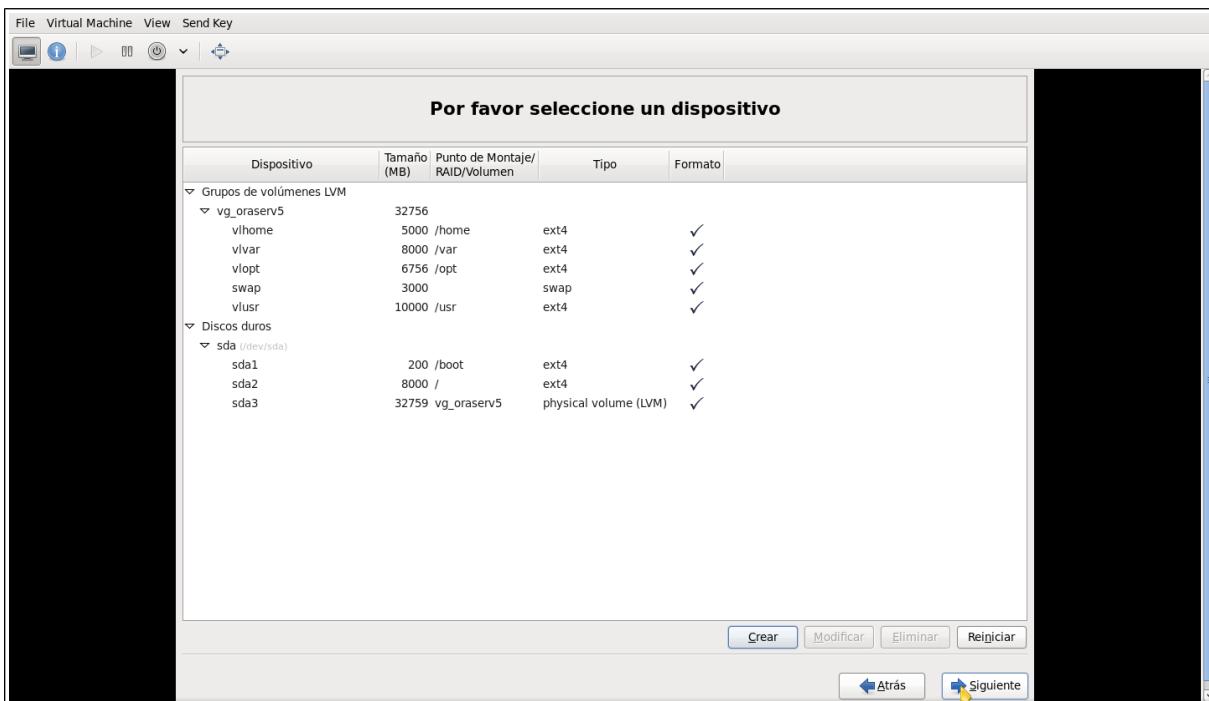
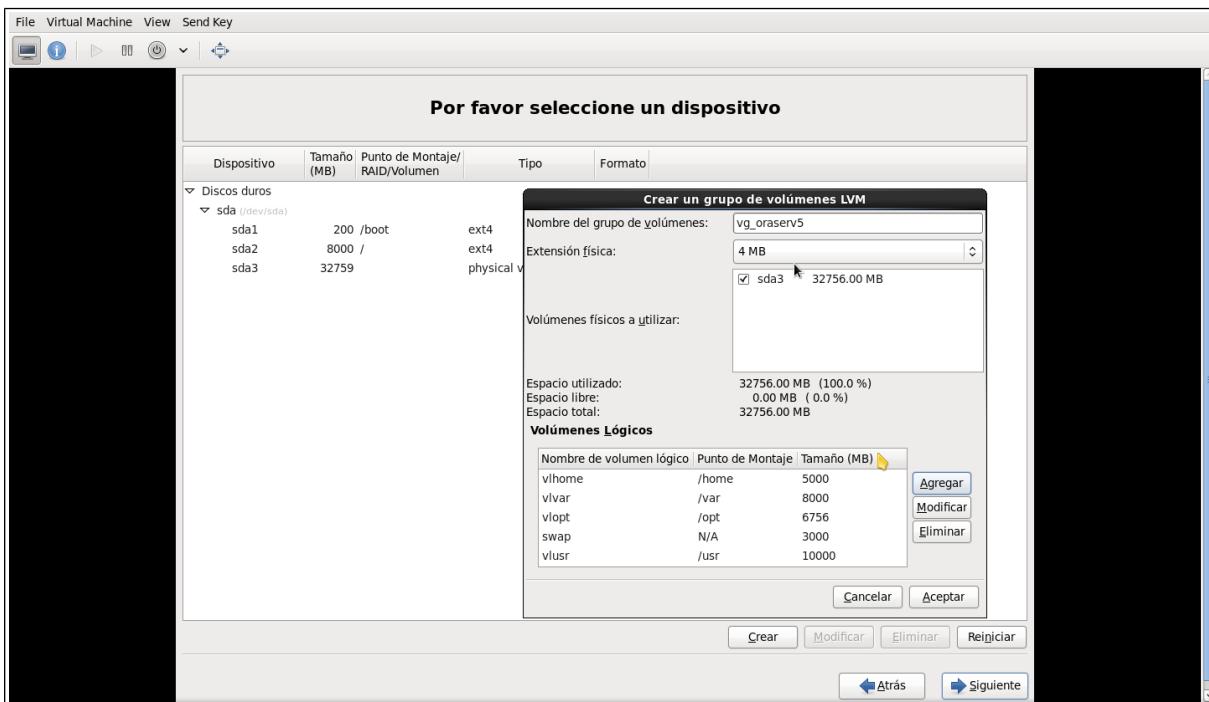


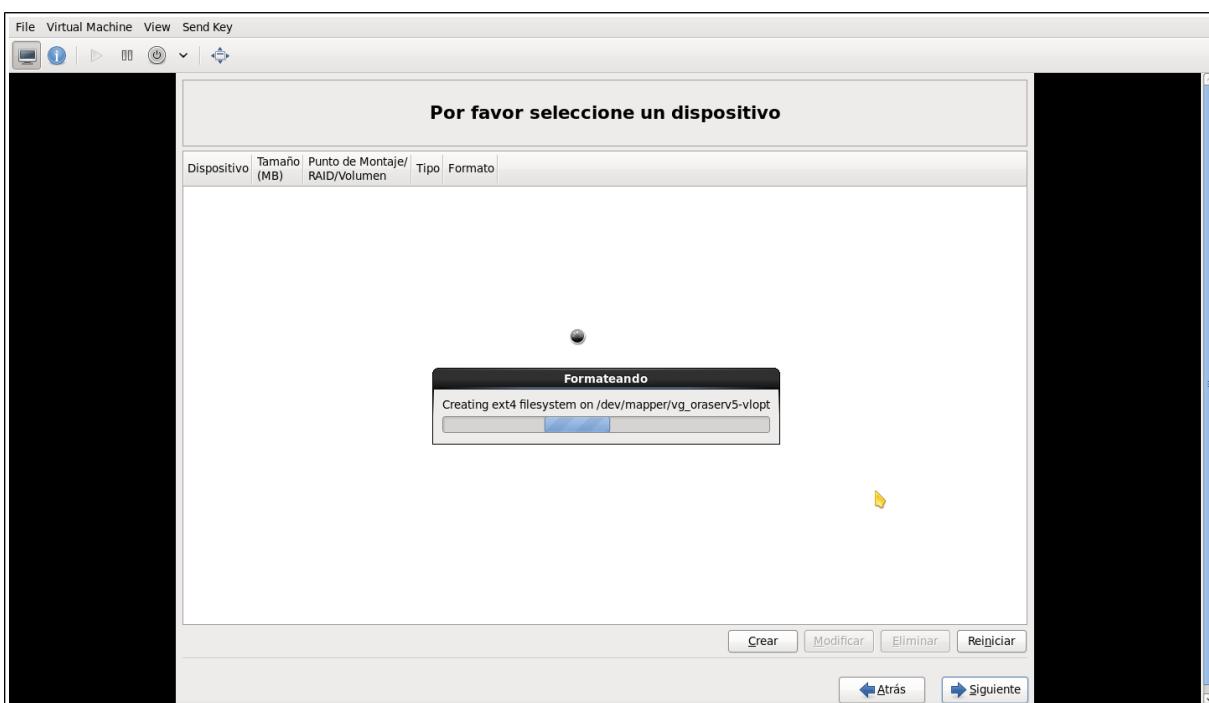
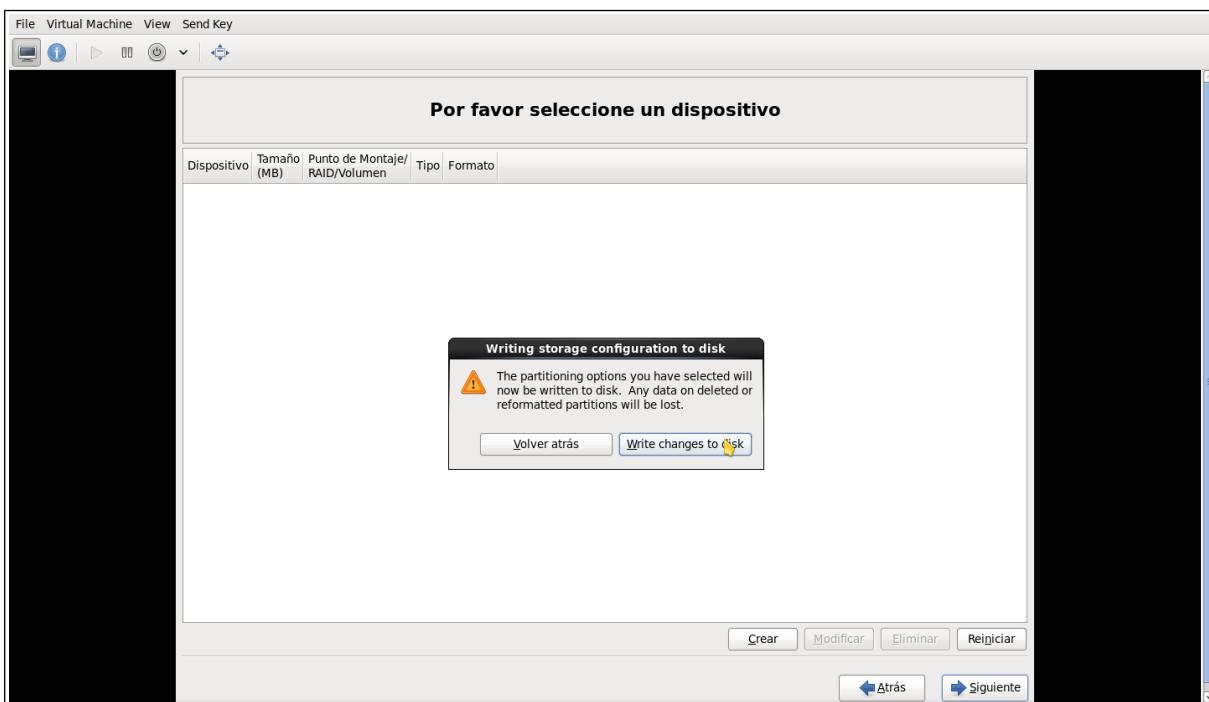


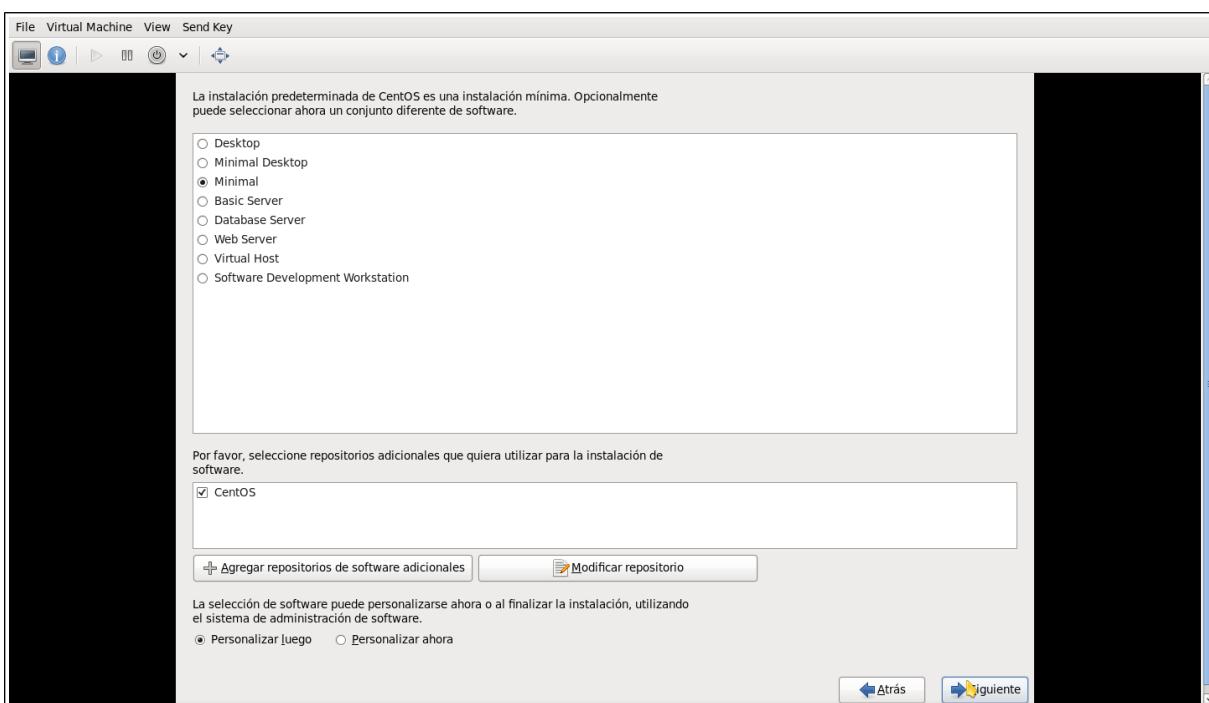
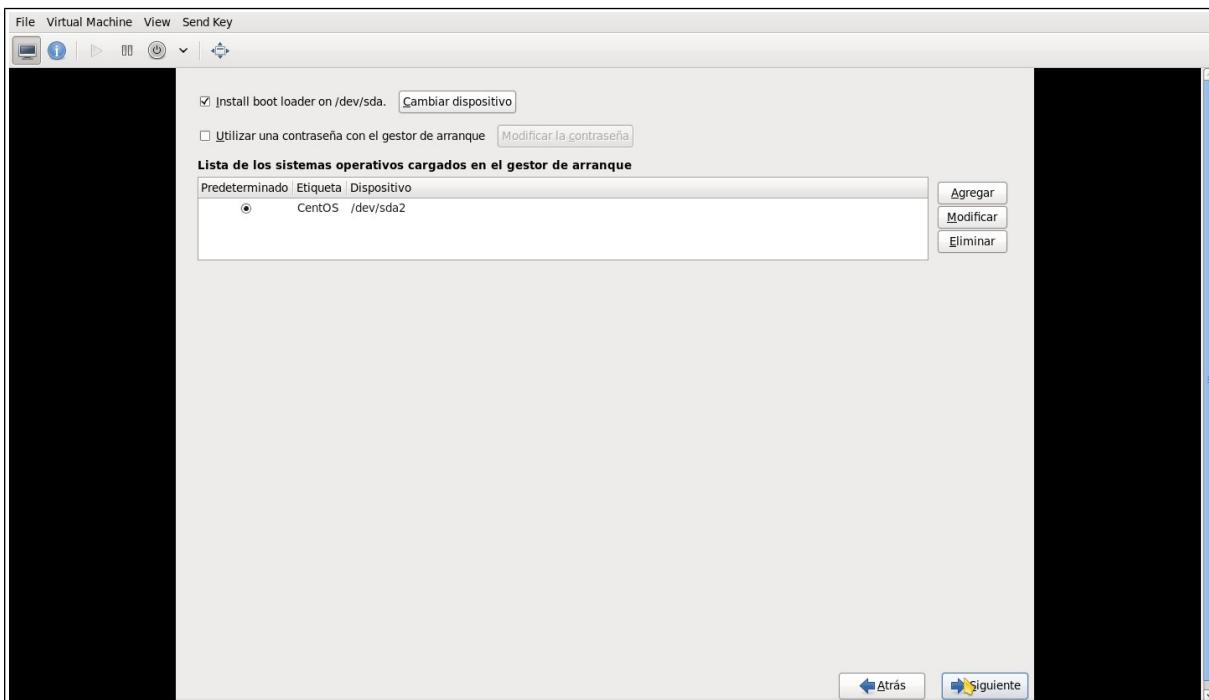




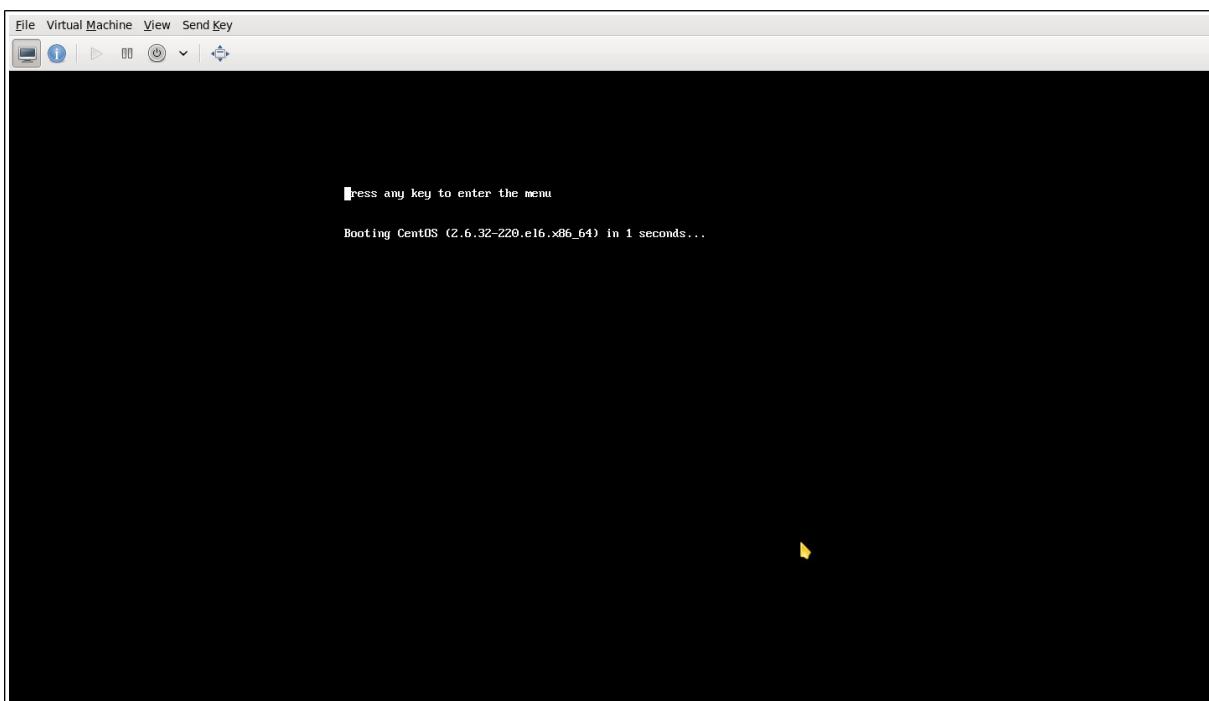
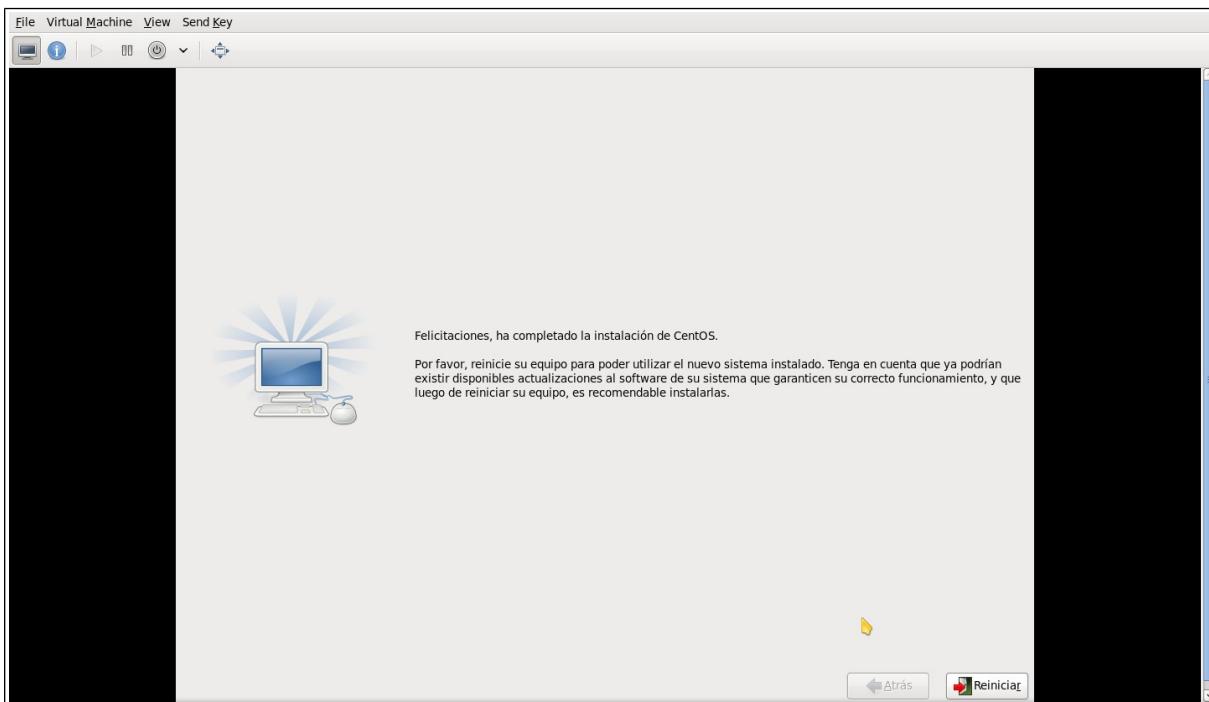


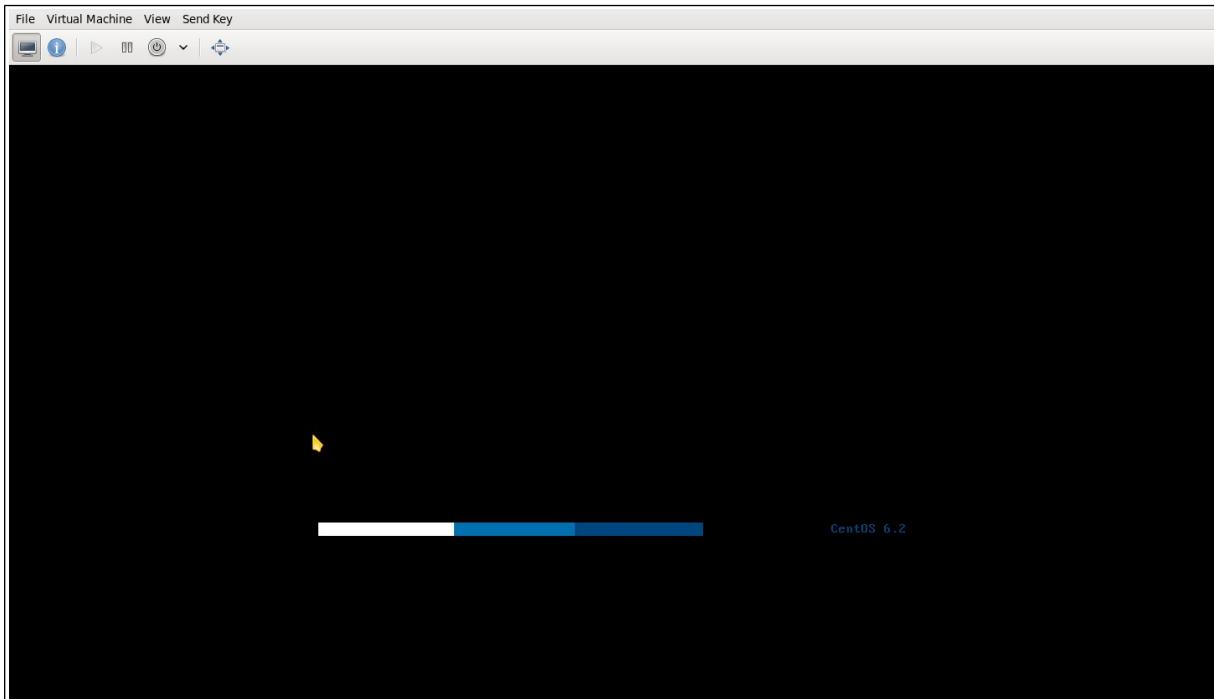












Todas con el Sistema de Archivo Ext4

```
[root@oraserv5 ~]# df -HT
Filesystem  Type      Size  Used  Avail Use% Mounted on
/dev/sda2   ext4      16G   519M  15G   4% /
/dev/sda1   ext4      508M   69M  414M  15% /boot
/dev/mapper/vg_oraserv5-vlhome ext4    135G  1.5G  126G  2% /home
/dev/mapper/vg_oraserv5-vlopt  ext4    207G  37G  160G  19% /opt
/dev/mapper/vg_oraserv5-vlusr  ext4     11G   3.9G  6.0G  40% /usr
/dev/mapper/vg_oraserv5-vlvar  ext4    113G  6.4G  101G  6% /var
```

```
[root@oraserv5 ~]#lvscan
ACTIVE            '/dev/vg_oraserv5/vlhome' [126.95 GiB] inherit
ACTIVE            '/dev/vg_oraserv5/vlvar' [106.87 GiB] inherit
ACTIVE            '/dev/vg_oraserv5/vlswap' [11.72 GiB] inherit
ACTIVE            '/dev/vg_oraserv5/vlopt' [195.31 GiB] inherit
ACTIVE            '/dev/vg_oraserv5/vlusr' [9.77 GiB] inherit
```

Comandos GNU/Unix

Interprete de comandos (SHELL).

Es la interfaz primaria entre un usuario sentado ante su terminal y el sistema operativo.

Es un proceso que muestra un indicador de comandos (\$, %, o #) y aguarda que el usuario digite un comando

```
[jp@oraserv5 ~]$ (Esto Se le llama Prompt)
```

Antes de la Arroba esta el nombre del Usuario, Luego de la Arroba esta el Nombre de nuestro PC.

FICHERO- Cualquier información salvada en disco identificada con un nombre.

DIRECTORIO- Colección de ficheros identificada con un nombre.

PATH- Identificación de un camino para alcanzar un fichero indicando el conjunto de directorios por lo que se atraviesa.

Ejecución de comandos.

Ejecutando comandos en el Prompt.

```
[jp@oraserv5 ~]$ Comando [Opción] [Archivo]
```

Listo el contenido del directorio donde estoy.

```
[jp@oraserv5 ~]$ ls
```

Listo el contenido del directorio /dev en formato largo

```
[jp@oraserv5 ~]$ ls -l /dev
```

Los comando no se escriben en mayúscula.

```
[jp@oraserv5 ~]$ LS
```

bash: LS: command not found

Si el shell no devuelve error es que todo esta funcionando correctamente.

Ls : El comando para mostrar los ficheros y/o directorios.

```
[jp@oraserv5 ~]$ ls
```

Mkdir : Comando que nos permite crear directorios. Un directorio, es un archivo el cual tiene la propiedad de poder almacenar, ficheros y directorios.

```
[jp@oraserv5 ~]$ mkdir oracled
```

Cd : Comando que nos permite movernos a través del árbol de directorios.

```
[jp@oraserv5 ~]$ cd oracle
```

Cp : Copiar archivos y/o directorios.

```
[jp@oraserv5 ~]$ cp /etc/group .
```

Mv : Comando con el cual puedo borrar archivos y/o directorios, también permite renombrarlos.

```
[jp@oraserv5 ~]$ mv group grupos.txt
```

Pwd : El comando pwd indica el camino absoluto del directorio en el cual nos encontramos actualmente.

```
[jp@oraserv5 ~]$ pwd
```

File : El comando file determina con cierto grado de precisión el tipo de un fichero que se le pasa como argumento

```
[jp@oraserv5 ~]$ file grupos.txt
```

Du : Comando que permite conocer el tamaño de un directorio y su respectiva jerarquía de directorios

```
[jp@oraserv5 ~]$ du -sch /home/oracle
```

Cat : Muestra un archivo en la pantalla.

```
[jp@oraserv5 ~]$ cat grupos.txt
```

Tail : visualiza las ultimas líneas.

```
[jp@oraserv5 ~]$ tail grupos.txt
```

Head : visualiza las primeras líneas.

```
[jp@oraserv5 ~]$ head grupos.txt
```

Less : Muestra un archivo en la pantalla pagina por pagina

```
[jp@oraserv5 ~]$ less grupos.txt
```

Rm : Borrar archivos y/o directorios. Este es un comando que debemos utilizar con mucho cuidado, ya que si borramos algunos archivos por equivocación, Sera imposible recuperarlos.

```
[jp@oraserv5 ~]$ rm grupos.txt
```

Df : Se emplea para conocer información acerca de las particiones y dispositivos montados actualmente en el sistema.

```
[jp@oraserv5 ~]$ df
```

Free : Se emplea para conocer información acerca de las memorias

```
[jp@oraserv5 ~]$ free
```

Cuenta Root

En Linux existen diferentes tipos de cuentas de usuario, sin embargo existe una que es la de administración general del sistema, esta se denominada root o superusuario, es decir, es capaz de crear otros usuarios y cambiar la configuración del sistema en general. Básicamente el superusuario puede activar o desactivar cualquier tipo de servicios, cuentas de usuarios, configuraciones del sistema, entre otros.

Comandos su - y sudo

(su) es una herramienta que nos permite cambiarnos de usuario, eventualmente se usa para cambiar al usuario administrador o cuando necesitamos 'Disfrazarnos de otro usuario', las opciones mas usuales del comando su son:

su usuario. Si no se especifica un usuario, se esta cambiando al superusuario root, este comando de este modo solo es útil para disfrazarnos de otros usuarios, pues al no cargar las variables de ambiente de ese usuario es poco lo que podemos hacer.

```
[jp@oraserv5 ~]$ su oracle
```

su - usuario. Con la opción del guion (-) añadida en el comando su, carga las variables de ambientes de dicho usuario, pudiéndose así usar todos los aspectos del perfil de dicho usuario.

```
[jp@oraserv5 ~]$ su - oracle
```

su -c usuario. Nos permite ejecutar un comando como otro usuario. Si no se le especifica usuario asumirá que es root.

```
[jp@oraserv5 ~]$ su -c 'lsnrctl status'
```

```
[jp@oraserv5 ~]$ su -c 'ifconfig'
```

El programa sudo (de las siglas superuser - o substitute user - do) es una utilidad de los sistemas tipo Unix, que permite a los usuarios ejecutar programas con los privilegios de seguridad de otro usuario (normalmente el usuario root) al igual que el comando su, de manera segura. Se diferencia de su, porque se maneja a través de su archivo de configuración /etc/sudoers, y no cambia de usuario, sino mas bien obtiene los privilegios del usuario en cuestión. Se instala por defecto en /usr/bin.

useradd : Crear usuarios.

```
[root@oraserv5 ~]# adduser poracle
```

passwd : Colocar contraseña a los usuarios.

```
[root@oraserv5 ~]# passwd oracle
```

userdel : Eliminar usuario.

```
[root@oraserv5 ~]# userdel poracle
```

userdel : Eliminar usuario.

```
[root@oraserv5 ~]# userdel poracle
```

groupadd : Creación de grupo.

```
[root@oraserv5 ~]# groupadd goracle
```

groupdel : Eliminar grupo.

```
[root@oraserv5 ~]# groupdel goracle
```

gpasswd : Asignación de usuario existentes a grupos existentes.

```
[root@oraserv5 ~]# gpasswd -a poracle goracle
```

chmod : asigna permisos a ficheros o carpetas

```
[root@oraserv5 oracled]# chmod 421 grupos.txt
```

Stat: Para ver los permisos en archivos y carpetas o ls -l.

```
[root@oraserv5 oracled]# stat grupos.txt
```

R: significa permiso para leer

W: significa permiso para escribir

X: significa permiso para ejecutar

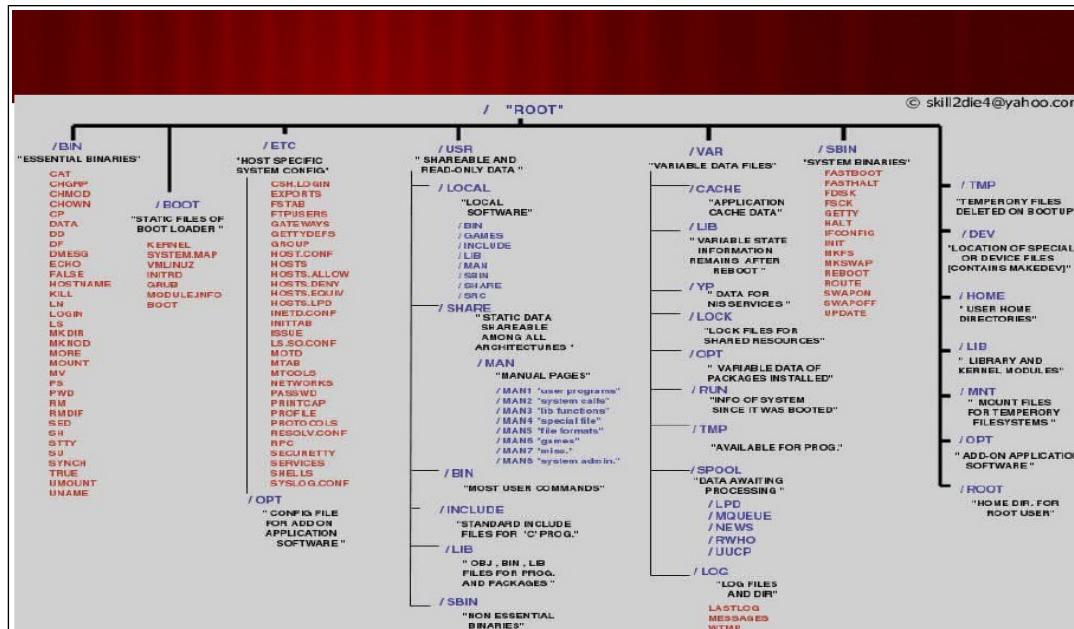
Practicas:

Debe adquirir en www.acl.edu.do la practica **Comandos-Para un Dba GCOA** la cual le permitirá a usted avanzar con conocimiento del S.O que debe tener un Dba Senior.

Sistema de archivos

La estructura del sistema de archivo de GNU/Linux es jerárquica, partiendo de su archivo principal (/), este tipo de estructura, similar a un árbol, nos indica donde podemos encontrar nuestros archivos.

/	Directorio raíz, donde todo empieza bin Binarios de comandos esenciales
boot	Archivos estáticos de gestor de arranque (boot loader)
dev	Archivos de dispositivos
etc	Configuración del sistema local-máquina
home	Directorios home de los usuarios
lib	Librerías compartidas
lost+found	Directorio para almacenar archivos a recuperar
mnt	Punto de montaje de particiones temporales
opt	Para colocar software que no fue incluida en el sistema operativo
root	Directorio hogar del usuario root
sbin	Binarios del sistema esenciales
tmp	Archivos temporales
usr	Segunda jerarquía mayor
var	Información variable, logs del sistema, Servicio
proc	Sistema de archivo virtual (información del kernel)



El editor VI

Vi es un editor de texto muy poderoso, es el editor por defecto de todas las distribuciones GNU/Linux. Vi es un editor con dos modos: edición y comandos. En el modo de edición el texto que ingrese será agregado al texto, en modo de comandos las teclas que oprima pueden representar algún comando de vi. Para cambiar de un modo de comando a inserta suficiente con que presione la tecla i.

Cuando comience a editar un texto estará en modo insertar para salir de este modo suficiente con teclear ESC de esta forma podemos pasar las opciones para manejar el texto. Los comandos básicos para manejar vi son:

:q	Para salir sin hacer cambios, en el caso que hayamos escrito algo que no queremos guardar o tratemos de editar un archivo que no nos pertenece agregamos el símbolo "!" Para forzar la salida. Ej: :q!
:w	Para guardar sin salir.
:wq	Para guardar y salir.
:x	Para guardar y salir.
:\$	Para ir a la ultima linea.
:0	Para ir a la primera linea.
:o	Para insertar una nueva linea donde se encuentra el cursor.
:set number	Para que nos numere la linea donde se encuentra el cursor.
:set nonumber	Para quitar la numeración
/	Para buscar una palabra o frase.
:nohl	Para quitar el sombreado de búsqueda
:sh	Nos permite salir del editor guardando en buffer con lo que estábamos trabajando. Para regresar a vi tecleamos Ctrl+D.
%s/viejo/nuevo/.	Nos sirve para buscar una palabra y sustituirla la sintaxis es
yy	Para copiar. yy copia la linea en donde se encuentra el cursor y le podemos pasar cuantas lineas queremos que copie a partir de donde esta el cursor en adelante. Ej: 7yy copiara desde la linea del cursor hasta siete lineas.
dd	Para cortar. dd corta la linea en donde se encuentra el cursor y le podemos pasar cuantas lineas queremos que corte a partir de donde esta el cursor en adelante. Ej: 3dd cortara desde la linea del cursor hasta tres lineas.
pp	Pegara los datos previamente copiados con yy o cortados con dd . El mismo principio que yy y dd , le podemos especificar, pasando un numero, que cantidad de lineas pegaremos.
:u	Para deshacer el ultimo cambio. Deshace de forma recurrente.
:redo	Para rehacer el ultimo cambio, al igual que :u va rehaciendo de forma recurrente.

Emacs

El EMACS original significa, Editor MACroS para el TECO, TECO era un editor de texto del laboratorio IA. Fue escrito en 1975 por Richard M. Stallman junto con Guy Steele. Se han lanzado muchas versiones de EMACS hasta el momento, pero actualmente una de las usadas comúnmente es GNU Emacs, iniciado por Richard M. Stallman en 1984, siendo esta a su vez el primer programa de GNU.

Comandos mas comunes en relación con las ventanas

Teclado	Mandato	Acción
C-x 0	delete-window	Borra la ventana activa
C-x 1	delete-other-windows	Borra todas las ventanas menos la activa
C-x 2	split-window-vertically	Divide horizontalmente la ventana activa en dos ventanas
C-x 3	split-window-horizontally	Divide verticalmente la ventana activa en dos ventanas
C-x o	other-window	Lleva el cursor a la próxima ventana

Comandos básicos de emacs

- Para salvar sin salir
CTRL-X S
Para Salvar el mismo fichero pero con otro nombre, es decir conservando la estructura del fichero original (similar a salvar como)
CTRL-X CTRL-W
- Para salir del archivo, si el archivo fue modificado preguntara si se desea guardar los cambios
CTRL-X CTRL-C
- Para llamar un archivo existente desde emacs, el archivo debe existir
CTRL-X I
- Para buscar un archivo en emacs, si no lo encuentra lo crea
CTRL-X F
- Para cancelar un mandato (Keyboard keys)
CTRL-G
- Para deshacer de forma recurrente
CTRL-X U
- Para abrir un directorio y de esta manera abrir, modificar, borrar ficheros contenido en dicho fichero.
CTRL-X D
- Ir al inicio del fichero
CTRL-INICIO (tecla home)
- Ir al final del fichero
CTRL-FIN (tecla end)
- Ir al inicio de la linea donde se encuentra el cursor
CTRL-DERECHA (teclas de direccionamiento)
- Ir al final de la linea donde se encuentra el cursor
CTRL-IZQUIERDA (teclas direccionamiento)

➤ Copiar y pegar

Para copiar datos el procedimiento es el siguiente:

CTRL-ESPACIO

En la zona de eco pondrá mark set movámonos al final del bloque o la linea para seleccionar el texto en cuestión

CTRL-IZQUIERDA

Hasta aquí hemos seleccionado el texto que vamos a usar ya sea para copiar o cortar. Para copiar el texto previamente seleccionado

ALT-W

Para cortar el texto previamente seleccionado

CTRL-W

Para pegar nos posicionamos usando las teclas direccionamiento donde deseamos pegar el texto.

CTRL-Y

➤ Para buscar texto

CTRL-S

➤ Terminar la búsqueda

CTRL-R

➤ Reemplazar texto

ALT-SHIFT-5

Te preguntara por el texto a reemplazar, le especificamos la búsqueda, por ejemplo la letra a y luego te preguntara por la cadena de reemplazo, por ejemplo b, luego te preguntara por todas las ocurrencias que deseas reemplazar, una por una le contestas si o no.

Para conectar desde Emacs a una Base Datos

ALT+X luego **sql-oracle** (Nos pedirá Usuario+Password+Database)

➤ Luego para activas el Modo Sql en otro buffer es ALT+X luego **sql-mode**, reconocerá toda la sintaxis SQL.

```

emacs@oraser5.fcl.d.acl
File Edit Options Buffers Tools SQL Help
File Edit Options Buffers Tools SQL Help
;; This buffer is for notes you don't want to save, and for Lisp evaluation.
;; If you want to create a file, visit that file with C-x C-f,
;; then enter the text in that file's own buffer.

select * from cat;
show user;

U:*** *scratch* All L8 (SQL[ansi])-----
SEQUENCE
TABLE_TYPE
-----+
KETS_AUTOTASK_STATUS TABLE
KETS_CLIENT_CONFIG TABLE
KETS_CLIENT_TASKS TABLE
KOTS_BOTS TABLE
KOTADS TABLE
KOTADXS TABLE
KOTMDS TABLE
KOTTBS TABLE
KOTTBXS TABLE
KOTTDS TABLE
4839 rows selected.

SQL> SQL> USER is "SYS"
SQL> SQL> -----
U:*** *SQL* Bot L6176 (SQL[oracle]):run-----
Sent string to buffer *SQL*.

```

Tipo de Paquetes en GNU/Unix

yum es un paquete administrador de software(software package manager). Es una muy útil herramienta para instalar, actualizar y remover paquetes junto con sus dependencias en distribuciones Linux basados en RPM (redhat, fedora, centos, etc.). Automáticamente determina las dependencias necesarias y lo que debe realizarse para instalar paquetes. Esta enfocada al uso práctico y rápido de las funciones más comunes de YUM a través de ejemplos.

YUM = yellowdog updater modified, es su nombre completo, originado en la distro YellowDog, que sobre la base de rpm creó este paquete administrador de software. Actualmente mantenido por la Universidad de Duke.

En este directorio se hallan los archivos de los repositorios, /etc/yum/repos.d/ Debemos agregar repositorios, es la segunda opción después de una instalación GNU/Linux para BD.

- ✓ **Lista los repositorios que se tengan de yum.**

```
[root@oraserv5 ~]# yum repolist
```

- ✓ **Lista de todos los paquetes disponibles para instalación, actualización o ya instalados.**

```
[root@oraserv5 ~]# yum list
```

- ✓ **Muestra solo los paquetes disponibles o ya instalados de mysql.**

```
[root@oraserv5 ~]# yum list |grep mysql
```

- ✓ **Lista de todos los paquetes instalados en el sistema.**

```
[root@oraserv5 ~]# yum installed
```

- ✓ **Lista de todos los paquetes disponibles para ser instalados.**

```
[root@oraserv5 ~]# yum available
```

- ✓ **Busca el 'paquete' en la base de datos de paquetes para instalar. 'paquete' puede ser una palabra parcial del paquete a buscar.**

```
[[root@oraserv5 ~]# yum search lvm
```

- ✓ **Muestra información resumida de los paquetes disponibles a actualizarse.**

```
[root@oraserv5 ~]# yum info system-config-lvm
```

- ✓ Instala la última versión del paquete indicado.

```
[root@oraserv5 ~]# yum install system-config-lvm
```

- ✓ Remueve el paquete indicado.

```
[root@oraserv5 ~]# yum remove system-config-lvm
```

- ✓ Descarga un paquete Rpm con yum.

```
[root@oraserv5 ~]# yumdownloader bash-completion
```

RPM

RPM (RPM Package Manager, anteriormente conocido como Red Hat PackageManager, es un sistema de gestión de paquetes de equipamiento lógico para GNU/Linux y que está considerado en la Base Estándar para Linux (Linux Standard Base o LSB), proyecto cuyo objetivo es desarrollar y promover estándares para mejorar la compatibilidad entre las distribuciones de GNU/Linux para permitir a las aplicaciones ser utilizadas en cualquier distribución.

RPM fue originalmente desarrollado por Red Hat, Inc. para su distribución de GNU/Linux y ha sido llevado hacia otra distribuciones de GNU/Linux y otros sistemas operativos.

RPM utiliza una base de datos que se almacena dentro del directorio /var/lib/rpm, la cual contiene toda la meta-information de todos los paquetes que son instalados en el sistema y que es utilizada para dar seguimiento a todos los componentes que son instalados. Ésto permite instalar y desinstalar limpiamente todo tipo de aplicaciones, programas, bibliotecas compartidas, etc. y gestionar sus dependencias.

El mandato rpm viene instalado de modo predeterminado en CentOS, Fedora, Red Hat Enterprise Linux , SuSE Linux Enterprise, openSuSE, Mandriva y las distribuciones derivadas de éstas.

- ✓ Lista los paquetes instalados en el sistema.

```
[root@oraserv5 ~]# rpm -qa
```

- ✓ Si se quiere localizar un paquete o paquetes en particular.

```
[root@oraserv5 ~]# rpm -qa|grep x11
```

✓ **Información sobre el paquete instalado.**

```
[root@oraserv5 ~]# rpm -qi system-config-lvm
```

✓ **Te informa a que paquete pertenece el comando “ls” o cualquier otra aplicación.**

```
[root@oraserv5 ~]# rpm -qf /bin/ls
```

✓ **Instalar un paquete RPM.**

```
[root@oraserv5 ~]# rpm -ivh bash-completion
```

✓ **Actualizar un paquete RPM y si no esta lo instala.**

```
[root@oraserv5 ~]# rpm -Uvh bash-completion
```

✓ **Lista de archivos del paquete rpm.**

```
[root@oraserv5 ~]# rpm -ql bash-completion
```

✓ **Desinstalar un paquete RPM.**

```
[root@oraserv5 ~]# rpm -e bash-completion
```

✓ **Consulta de un paquete RPM.**

```
[root@oraserv5 ~]# rpm -q bash-completion
```

Tar.Gz

Los paquetes tar.gz son creados en forma no binaria, para ser compilado de acuerdo a un parámetro de configuración, esto sucede cuando la aplicación no esta disponible en alguna distribución ya porque sea recientemente creado por una comunidad emergente.

Comando tar

Este comando es importante para manejar los paquetes .tar, ya que con este se pueden empaquetar directorio para luego ser comprimidos, se le llaman “tarballs”, se usan para ahorrar espacio o para hacer copias de seguridad (backups)

✓ **Empaquetar**

```
[oracle@oraserv5 ~]$ tar -cvf BD.tar /opt/app/oracle/oradata
```

✓ **Desempaquetar**

```
[oracle@oraserv5 ~]$ tar -xvf BD.tar -C opt/app/oracle/oradata
```

Se puede además comprimir a vuelo mientras se empaqueta (son dos procesos distintos). Lo hacemos de la manera siguiente.

Comprimir en tar.gz

```
[oracle@oraserv5 ~]$ tar -cvzf BD.tar.gz /opt/app/oracle/oradata
```

✓ **Descomprimir un tar.gz**

```
[oracle@oraserv5 ~]$ tar -xvzf BD.tar.gz -C /opt/app/oracle/oradata
```

✓ **Comprimir un tar.bz2**

```
[oracle@oraserv5 ~]$ tar -cvjf BD.tar.bz2 /opt/app/oracle/oradata
```

✓ **Descomprimir un tar.bz2**

```
[oracle@oraserv5 ~]$ tar -xvjf BD.tar.bz2 gzip
```

gzip es una abreviatura de GNU ZIP, un software libre GNU que reemplaza al programa compress de UNIX. gzip fue creado por Jean-loup Gailly y Mark Adler. Apareció el 31 de octubre de 1992 (versión 0.1). La versión 1.0 apareció en febrero de 1993.

gzip se basa en el algoritmo Deflate, que es una combinación del LZ77 y la codificación Huffman. Deflate se desarrolló como respuesta a las patentes que cubrieron LZW y otros algoritmos de compresión y limitaba el uso del compress. No se debe confundir gzip con ZIP, el cual no es compatible.

gzip SÓLO COMPRIIME ARCHIVOS, pero no los empaqueta. Debido a esto a menudo se usa junto con alguna herramienta para empaquetar (regularmente tar).

✓ **Para comprimir**

```
[oracle@oraserv5 admin]$ gzip tnsnames.ora
```

✓ Ver el archivo comprimido

```
[oracle@oraserv5 admin] ls
```

Nota También podemos ver información sobre el nivel de compresión de un archivo ejecutando el comando con la opción “-l”.

gunzip

Esta es la herramienta que permite descomprimir en GNU/Linux y UNIX, al igual que su contrario, gzip, solo descomprime archivos, este concepto se puede aclarar mejor viendo la sección del comando “tar” en el punto anterior.

```
[oracle@oraserv5 admin]$ gunzip tnsnames.ora.gz
```

Nota: Es bueno aclarar que cuando se comprime o descomprime con estas herramientas, no se hace copia del archivo, es decir, literalmente el archivo se convierte de normal a comprimido y viceversa.

Compilando desde el fuente

Una vez extraído el contenido del paquete tar.gz, se le pasan parámetros para configurar en entorno de acuerdo a un prefijo o variable definida, luego se procede a compilar el paquete, nótese que existen diferentes compiladores (gcc, g++, jre, entre otros), estos paquetes vienen con su guía documentada de como instalar, una vez se haya terminado la compilación, se procede a su instalación en el sistema

No todos los archivos .tar.gz2 o tarballs son archivos de fuentes para compilar, sino que puede ser que descarguemos un tarball y que contenga ya el programa que queremos ejecutable (archivos .sh, .run, .bin, etc.). Para saber si debemos compilar tendremos que revisar el archivo README y en caso de que éste no dé instrucciones de compilación ya podemos sospechar que no hay que compilar. Si aún así no estamos seguros podemos verificar que dentro haya un fichero ejecutable llamado configure. Si no existe ya no hay duda que no hay que compilar.

Es muy recomendable (si no obligatorio) que nos leamos el archivo de ayuda de instalación, que puede llamarse README o INSTALL, ya que como hemos comentado más arriba, aunque el proceso de compilación de un programa es muy similar entre los distintos programas, en este archivo figura el proceso de compilación específico para el programa.

✓ Descomprimir el archivo con las fuentes del programa

```
[root@oraserv5 ~]# bzip2 -cd nmap-5.61TEST4.tar.bz2 | tar xvf -
```

Nota: ./ Se emplea para ejecutar un archivo ejecutable que se halla en la misma carpeta donde nos encontramos en ese momento.

El fichero configure genera la configuración idónea para nuestro server y le dice a GCC cómo debe compilar las fuentes para que funcionen en nuestra máquina.

- ✓ Seguidamente, ejecutaremos el fichero ejecutable configure que viene dentro de la carpeta nuestro programa a compilar. Bastará con teclear la siguiente orden desde la consola de comandos:

```
[root@oraserv5 ~]# ./configure
```

- ✓ Generar el paquete binario

```
[root@oraserv5 ~]# make all
```

- ✓ Instalar el paquete binario

```
[root@oraserv5 ~]# make install
```

- ✓ Borrar la Carpeta

```
[root@oraserv5 ~]# make clean
```

Configuraciones necesarias luego de instalación minimal.

Instalamos el servidor X

```
[root@oraserv5 ~]# yum -y groupinstall "X Window System"
```

Para una instalación remota en un servidor que no necesita una interfaz gráfica, instala los siguientes paquetes y las dependencias para hacer el reenvío por X11:

```
[root@oraserv5 ~]# yum -y install xorg-x11-xauth xorg-x11-apps dejavu*
```

Asegúrese de que el nombre es del server este /etc/hosts, de lo contrario el oyente TNS no se iniciará.

```
192.168.100.178 oraserv5 oraserv5.acl
```

Paramos y desactivamos los iptables:

```
[root@oraserv5 ~]# service iptables stop
```

```
[root@oraserv5 ~]# chkconfig iptables off
```

Deshabilitamos el SeLinux:

```
[root@oraserv5 ~]# vi /etc/selinux/config  
SELINUX=disabled
```

Reiniciamos nuestro server:

```
[root@oraserv5 ~]# reboot
```

Ya podemos ir al cliente y conectarnos remotamente al server e iniciar la operación en el mismo.

Copiamos el scripts ACL-Install-Oracle-1.0.3-2012.sh:

```
[root@oraserv5 scripts]# scp FCLD-Install-Oracle-1.0.2.sh root@192.168.122.5:
```

Instalación de Oracle 11gR2

Objetivos: En este modulo perseguimos que el estudiante aprenda que es oracle, como instalarlo, conozca ademas a identificar su Hardware y determine si su equipo cumple con los requerimientos para la instalación de Oracle 11gR2 así como la preparación del sistema para dicha instalación.

Oracle es un sistema gestor de bases de datos que implementa el modelo relacional y una versión del modelo relacional orientado a objeto. Es un sistema multiplataforma y puede trabajar en diferentes modelos de ejecución (Cliente-Servidor, Centralizado, Procesamiento Distribuidos, entre otros). Las limitaciones de Oracle están determinadas por la plataforma en la cual se ejecuta. Oracle esta orientado a medianas y grandes demandas por lo que dispone de todas las características que se requieren de un Sistema Gestor de Bases de Datos:

- **Mecanismos de seguridad.** Lenguaje de control que permite manejar derechos de consulta, modificación y creación de datos a los usuarios.
- **Copias de Seguridad y Recuperación** Permite realizar dichas copias y recuperaciones con el servidor parado o funcionando (Copias en calientes), permitiendo un servicio ininterrumpido.
- **Herramientas de desarrollo.** Generadores de informe, formularios, etc.

Elementos del servidor Oracle

El servidor Oracle está formado por dos elementos:

- **La instancia de la base de datos.** Consta de datos (llamados estructuras de memoria) y de procesos en memoria (procesos background) necesarios para dar servicio a los usuarios de la base de datos. Puede haber más de una instancia si se distribuye la base de datos en más de una máquina. Cada instancia abre una y sólo una base de datos.
- **Ficheros en disco.** Representan la base de datos en sí. Consta de:
 - La estructura física se corresponde a los ficheros del sistema operativo: de datos (datafiles), de redo log y de control (control files).
 - La estructura lógica está formada por los tablespace y los objetos de un esquema de BD (tablas, vistas, índices, etc.).

Definiciones

Tablespace

Un espacio de tablas es una división lógica de una base de datos. Los espacios de tabla permiten agrupar usuarios o aplicaciones para mejor mantenimiento y desempeño. Cada base de datos debe contener por lo menos el espacio de tablas System. Cada espacio de tabla se guarda en disco en uno o mas archivos denominados archivos de datos (datafile). Cada archivo de datos pertenece a un y solo un espacio de tablas.

Redo logs

Son registros de todas las transacciones. Se usan para recuperar las transacciones de una base de datos en caso de falla

Control files

Los archivos de control mantienen la estructura física de la base de datos. Registran información de control acerca todos los archivos de la base de datos, mantienen consistencia interna y guían las operaciones de recuperativo

Trace files

Los archivos de rastreo contienen información de eventos significantes encontrados por los procesos de fondos (hablaremos de estos mas adelante).

Alert log

Los registros de alerta guardan los comandos y sus resultados de los principales eventos en la base de datos.

La Instancia

Como mencionamos ya anteriormente, la integran los proceso de background y la SGA. Estos procesos y la estructura de memoria son compartidos por los usuarios, esta estructura mantiene los datos consultados mas recientemente.

La estructura de memoria mejora el desempeño de la base de datos porque disminuye la cantidad de operaciones de entrada/salida sobre los archivos de datos.

Definición de Variables de Entorno

Hay muchas variables de entorno de Oracle y las que se mencionan son muy importantes para una instalación correcta y para el uso de una base de datos Oracle. No es necesario configurar ninguna de ellas, pero si las configura antes de la instalación puede evitar problemas en el futuro.

- ORACLE_BASE: Especifica la base de la estructura de directorios de Oracle para OFA.
- Su uso es opcional, pero puede facilitar futuras instalaciones y actualizaciones. Se trata de una ruta de acceso de directorio similar a la que aparece en este ejemplo:
➤ /opt/app/oracle
- ORACLE_HOME: Especifica el directorio que contiene el software de Oracle. Se trata de una ruta de acceso de directorio similar a la que aparece en este ejemplo:

\$ORACLE_BASE/product/11.2.1/db_1

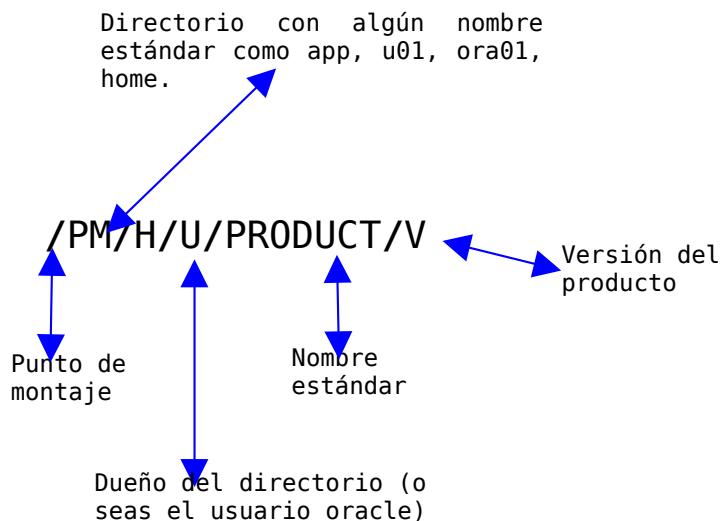
- ORACLE_SID: Nombre de la instancia inicial (por defecto, oraserv5). Es una cadena de números y letras que debe comenzar por una letra. Oracle Corporation sugiere que se utilice un máximo de ocho caracteres para los identificadores del sistema.
- NLS_LANG: Especifica los valores iniciales de soporte de idioma nacional (NLS) para una sesión en formato idioma_teritorio.juego de caracteres.

Arquitectura OFA

Oracle Flexible Architecture, la arquitectura flexible de oracle es un estructura de directorios que propone oracle para la fácil localización de cualquier fichero del servidor. OFA propone los puntos de montaje de los directorios, ademas de una nomenclatura que permite organizar dichos ficheros:

- Nombramiento de archivos de Linux y puntos de montaje (Strings+ numbers).
- Nombramiento de directorios.
- Nombramiento de archivos de base de datos.
- Localizaciones de los archivos.

Un ejemplo de ruta de localización del software Oracle que cumple con OFA seria:



Oracle también recomienda que el software se mantenga separado de los datos. En resumen estos serian los directorios y su localización:

```

ORACLE_BASE = /opt/app/oracle
ORACLE_HOME = $ORACLE_BASE/product/11.2.0 ejecutables = $ORACLE_HOME/bin
initSID.ora, orapwSID = $ORACLE_HOME/dbs
admin= $ORACLE_BASE/admin/SID
pfile = $ORACLE_BASE/
bdump = $ORACLE_BASE/admin/SID/bdump (BDUMP)
Ficheros de la base de datos = /opt/oradata/SID

```

Requerimientos de Hardware y Software:

Los requerimientos que Oracle recomienda para su instalación:

Requerimientos del Sistema	Mínimo	Recomendado
Memoria RAM	512 MB	2 GB
Almacenamiento SWAP	1 GB	4 GB
Espacio en Disco en /opt	1.5 GB Software 4 GB Base de Datos	5GB Software 20 GB Base de Datos

Revisamos cuanto tenemos de disponibilidad en recursos de gestión de memoria:

- **Revisando la memoria RAM:** grep MemTotal /proc/meminfo
- **Revisando la memoria SWAP:** grep SwapTotal /proc/meminfo
- **Revisando la memoria RAM y SWAP:** free -tm

Revisamos cuanto tenemos de disponibilidad en recursos de almacenamiento:

- **Revisando la disponibilidad del disco:** df -h

Instalar los paquetes para 10gR2:

```
[jp@oraserv5 ~]$ yum -y libaio.i686 libaio libXp.i686 libaio-devel.i686 libaio-devel gcc sysstat libstdc++-devel compat-libstdc++-33.i686 libstdc++.i686 libXt.i686 libstdc++.x86_64 glibc.i686 gcc-c++ libXp glibc glibc-common libXt.i686 libXtst.i686 compat-libstdc++-33 glibc-headers glibc-devel.i686 glibc-devel
```

Instalar los paquetes para 11gR2:

```
[jp@oraserv5 ~]$ yum -y unixODBC.i686 libaio.i686 ksh libaio libgcc.i686 elfutils-libelf-devel libaio-devel.i686 libaio-devel gcc unixODBC unixODBC-devel sysstat elfutils-devel libstdc++-devel compat-libstdc++-33.i686 unixODBC-devel.i686 elfutils-libelf.i686 libstdc++.i386 libstdc++.i686 libstdc++.x86_64 gcc-c++ glibc glibc.i686 glibc-common glibc-headers glibc-devel.i686 glibc-devel compat-libstdc++-33 compat-libcap1
```

Descripción de parámetros del Kernel a modificar:

- **SEMMSL = 100:** cantidad máxima de conjuntos de semáforos
- **SEMMSN = 256:** cantidad máxima de semáforos, sólo para la instalación inicial
- **SEMMSN = sum(processes) + max(processes) + 10*count(BBDD):** processes es el parámetro del init de cada BD
- **SEMOPM = 100:** numero máximo de operaciones por "semop call"
- **SEMOMS = 100:** valor mínimo recomendado, sólo para la instalación inicial
- **SHMMAX = 2147483648:** tamaño máximo del segmento de memoria compartida y por tanto de la SGA, 2 GB para kernel SMP. Se recomienda la mitad de la RAM.
- **SHMMIN = 1:** tamaño mínimo de un segmento de memoria compartida.
- **SHMMNI = 4096:** tamaño máximo de segmentos de memoria compartida.
- **SHMSEG = 4096:** tamaño máximo de segmentos de memoria por proceso.
- **SHMVMX = 32767:** valor máximo de un semáforo.

Modificación de parámetros del Kernel:

1. Agregar los siguiente valores para los parámetro del archivo /etc/sysctl.conf

```
kernel.shmmmax = 536870912
kernel.shmmni = 4096
kernel.shmall = 2097152
# semaphores: semmsl, semmns, semopm, semmni
kernel.sem = 250 32000 100 128
fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default=262144
net.core.rmem_max=262144
net.core.wmem_default=262144
net.core.wmem_max=262144
```

2. Ejecutamos el siguiente comando para que surjan efectos los cambios:

```
[root@oraserv5 ~]# sysctl -p
```

- Por defecto, hay un límite por usuario, de 1024 descriptores fichero, y 2047 procesos.

3. Editemos el fichero /etc/security/limits.conf para ajustar estos valores. Añade las siguientes líneas a /etc/security/limits.conf :

```
* soft nproc 2047
* hard nproc 16384
* soft nofile 1024
* hard nofile 65536
```

- Las dos últimas líneas imponen un límite inicial de 1024, pero permiten que un usuario aumente el límite a 65536 usando el comando ulimit -n 65536. Las dos primeras líneas limitan el número de procesos.

4. Añade la siguiente línea a /etc/pam.d/login si no existe :

```
session required /lib/security/pam_limits.so
```

pam_limits.so es el módulo PAM que procesa la configuración del fichero /etc/security/limits.conf que cambiamos antes.

5. Deshabilitar la SeLinux /etc/selinux/config:

```
SELINUX=disabled
```

6. Apagamos los IpTables y los desactivamos:

```
[root@oraserv5 ~]# service iptables stop
```

```
[root@oraserv5 ~]# chkconfig iptables off
```

Cuenta Oracle

Grupos y usuarios que deben existir en el sistema :

- Grupo oinstall: Oracle Inventory group
- Grupo dba: Database administrator
- Usuario oracle: Usuario propietario del software

Estos son los comandos para la creación de los usuarios y grupos requeridos:

```
[root@oraserv5 ~]# groupadd oinstall
```

```
[root@oraserv5 ~]# groupadd dba
```

```
[root@oraserv5 ~]# useradd -g oinstall -G dba oracle
```

```
[root@oraserv5 ~]# passwd oracle
```

Creación de Directorios

Siguiendo con las recomendaciones de OFA(tratamos este tema mas adelante), la estructura a seguir de los directorios que usariámos para la instalación de Oracle seria:

```
[root@oraserv5 ~]# mkdir -p /opt/app/oracle/product/11.2.0/db_1
```

Asignamos como dueño del los directorios que creamos a el usuario Oracle.

```
[root@oraserv5 ~]# chown -R oracle.oinstall /opt/app
```

Puesto que el instalador solo se ejecuta en sistemas operativos certificados (RedHat o United Linux), para arrancar el instalador disfrazamos nuestro sistema, para que busque la versión del Linux que estamos usando en su archivo de configuración guardado en /etc y le cambiamos la versión del sistema por uno certificado por Oracle:

```
[oracle@oraserv5 ~]$ vim /etc/redhat-release  
redhat-4
```

Editamos el archivo .bashrc

Variables del usuario Oracle

```
# J.P
export ORACLE_HOSTNAME=oraserv5.fcl.dacl
export ORACLE_BASE=/opt/app/oracle
export ORACLE_HOME=/opt/app/oracle/product/11.2.0/db_1
export ORACLE_SID=oraserv5
export ORACLE_UNQNAME=oraserv5
export PATH=$ORACLE_HOME/bin:$PATH:
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
export NLS_DATE_FORMAT='dd-mm-yyyy hh24:mi:ss'
DISTRIB_RELEASE=5
alias sqlplus='rlwrap sqlplus'

## export JAVA_HOME JDK/JRE ##
export JAVA_HOME="/usr/java/latest"

## export JAVA_HOME JDK ##
export JAVA_HOME="/usr/java/jdk1.7.0_11"

## export JAVA_HOME JRE ##
export JAVA_HOME="/usr/java/jre1.7.0_11"
```

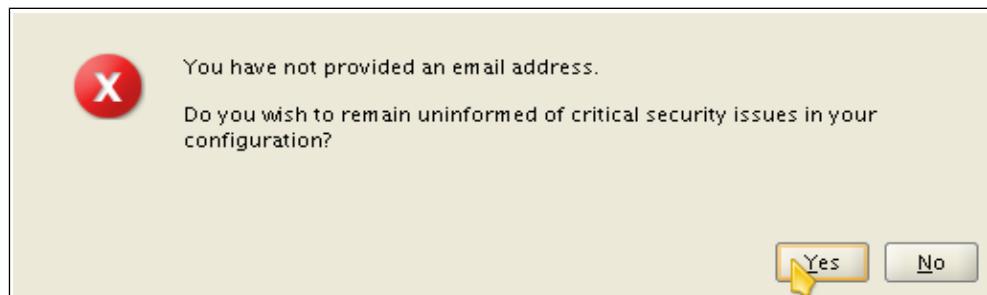
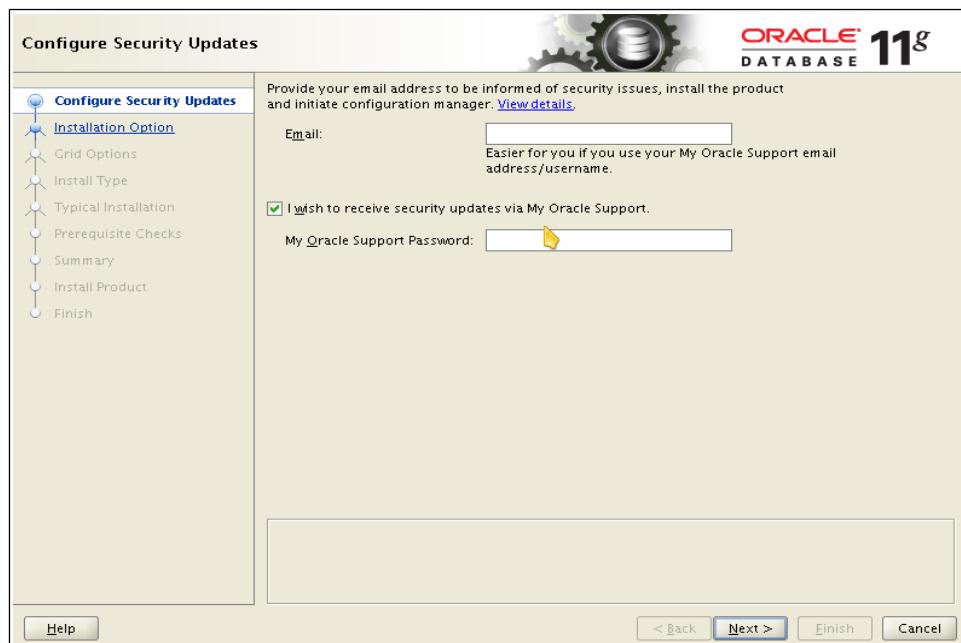
Luego se ejecuta el instalador:

Debemos estar logueado con el usuario Oracle.

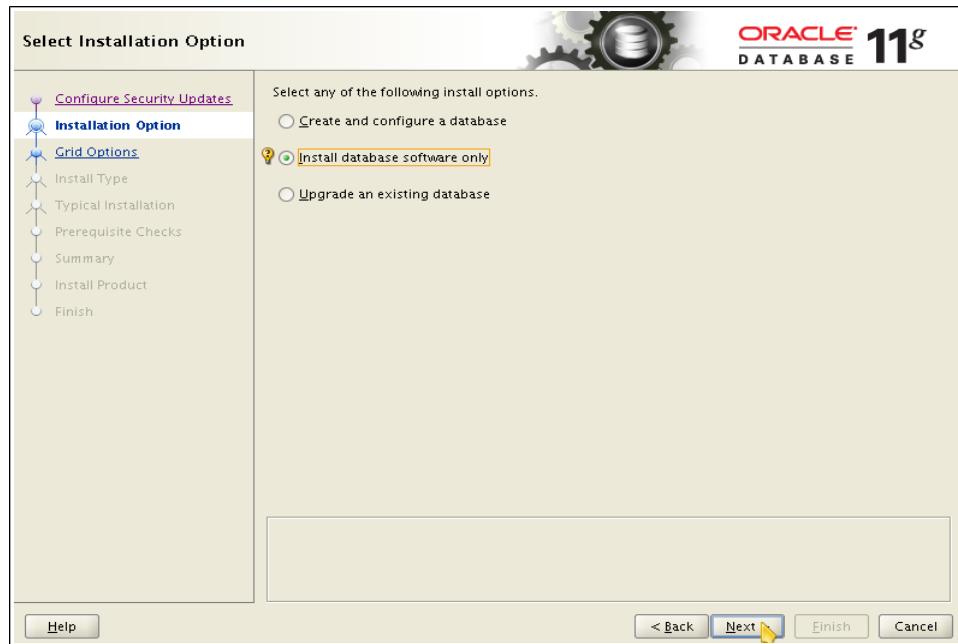
```
[oracle@oraserv5 database]$ ./runInstaller
```

Pasos para la instalación del software:

Una vez que empieza el instalador, en mi caso no seleccione la configuración con MOS (My Oracle Support), de la misma manera deje en blanco el correo electrónico



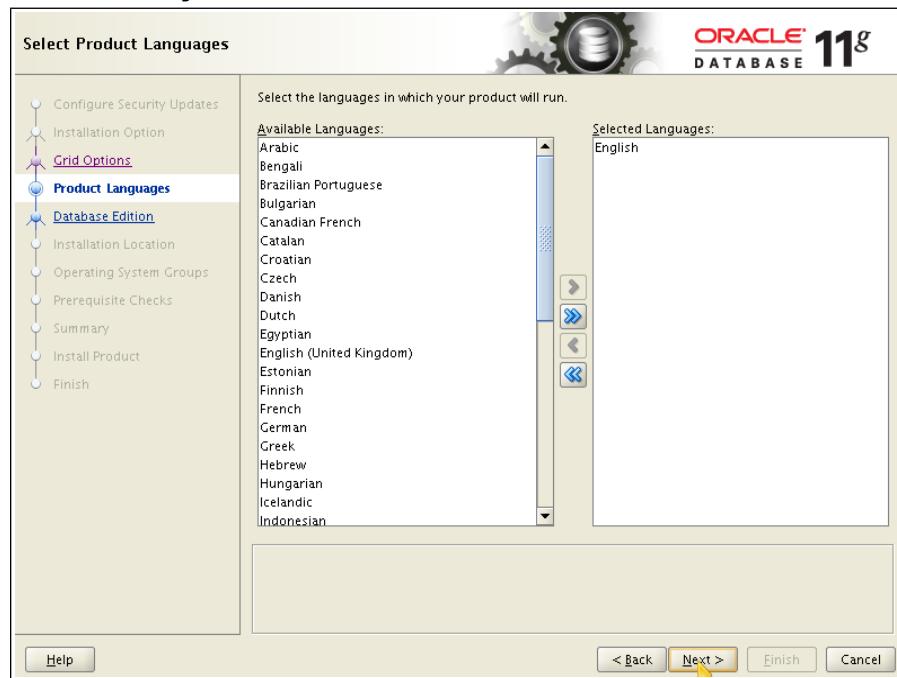
Ya como uno de los objetivos de esta entrada es la creación de una base de datos vía un aplicación DBCA, en la siguiente sección vamos a decirle al instalador que únicamente queremos instalar el Software.



Para continuar seleccione un tipo de instalación de una sola instancia.



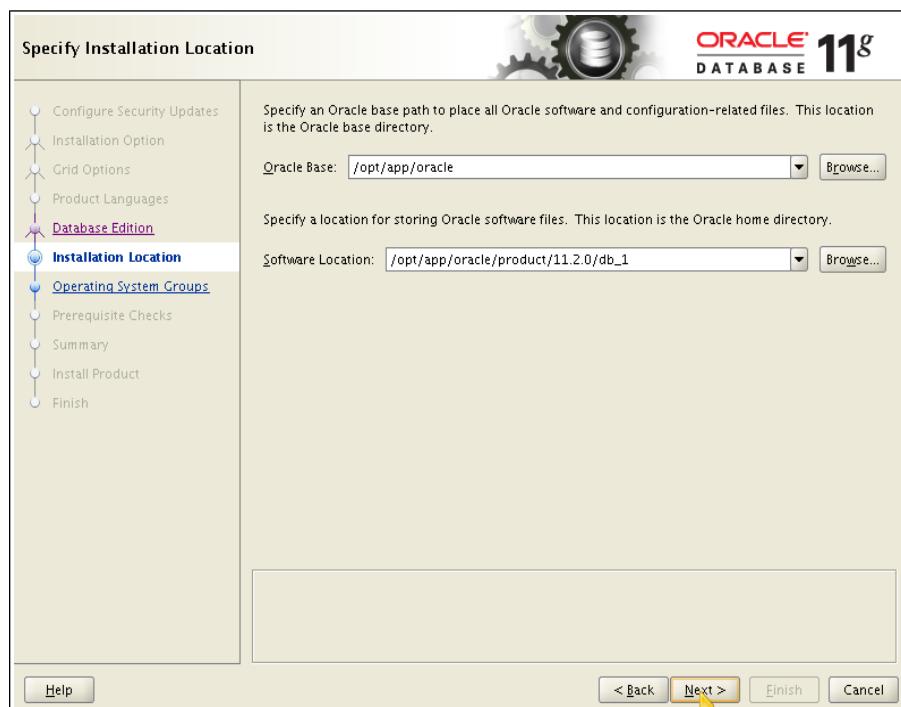
Ahora seleccionamos el idioma en que vamos a correr, en mi caso seleccione inglés, este se los dejo a su conveniencia.



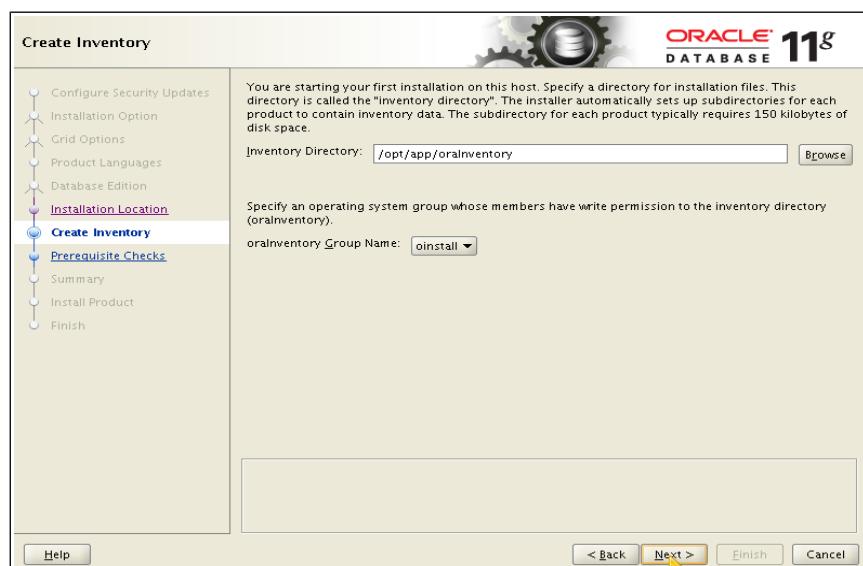
El tipo de instalación que lleve a cabo fue el de Enterprise Edition, esta es una selección que es de acuerdo a sus necesidades y licenciamiento que tienen.



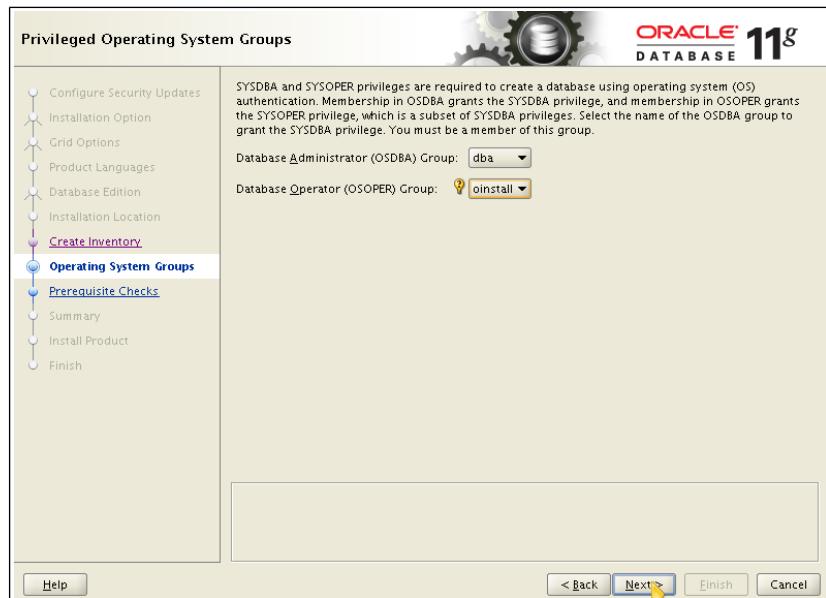
En esta parte el instalador verifica lo que es el ORACLE_BASE y el ORACLE_HOME, como estas variables ya las habíamos definido, no hay realmente nada que hacer mas que continuar.



Crear inventario, esta interfaz se dejan todos los valores como vienen y se presiona el botón siguiente.

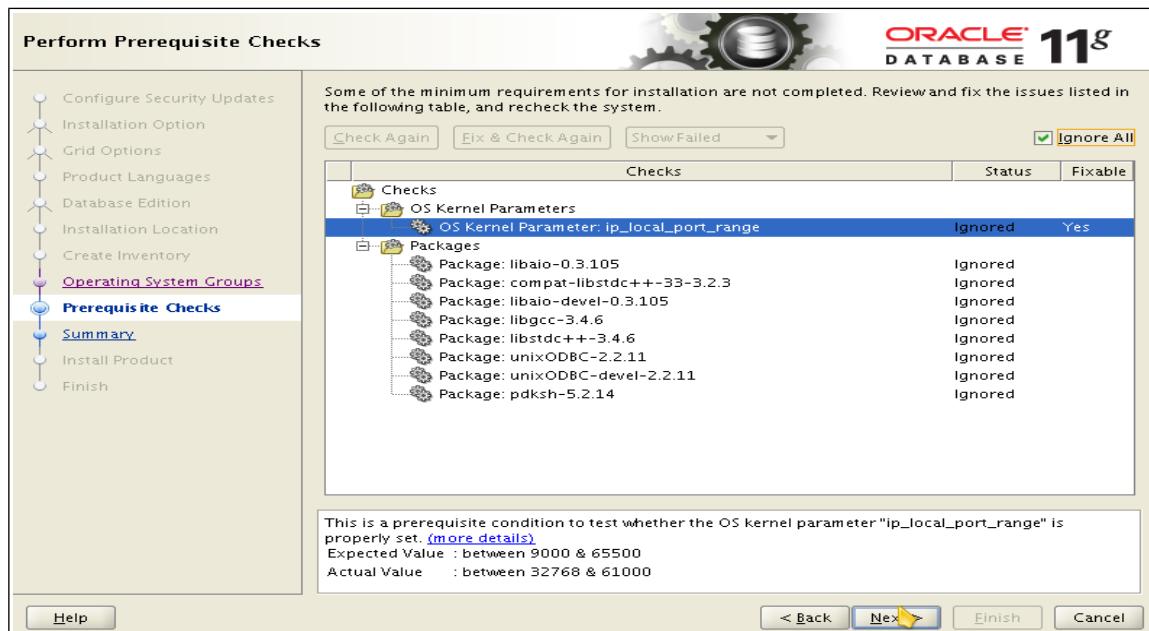


En la siguiente parte asignamos los grupos que van a tener permisos para nuestros binarios, en mi caso seleccione al mismo grupo dba, pero puede ser el grupo oinstall que creamos en los pre-requisitos.

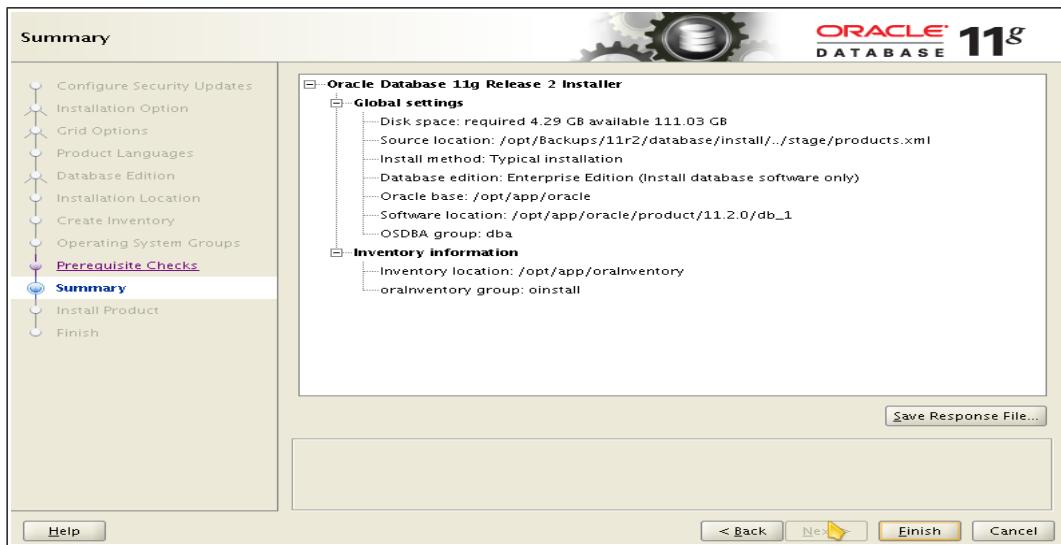


Si ya llegaste a esta parte del instalador y no has verificado los pre-requisitos, el instalador te va a hacer mención de cuales son los pre-requisitos que no se cumplen para proseguir con la instalación.

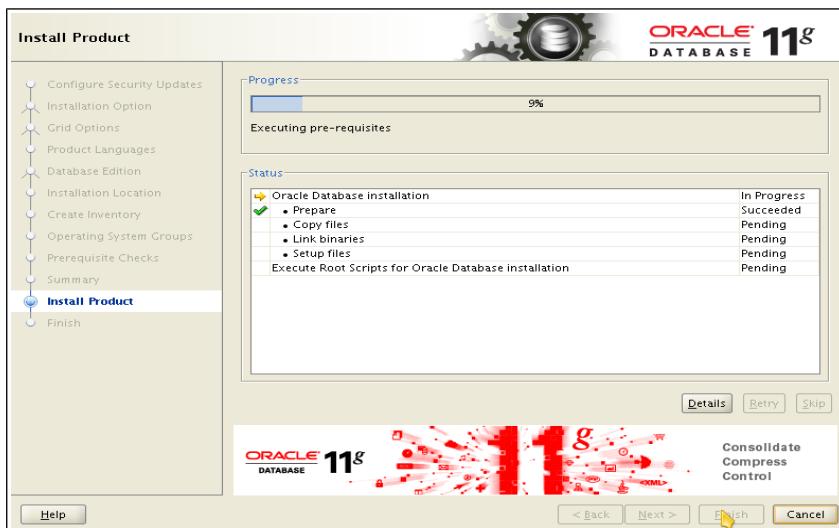
11gR2 tiene una funcionalidad de crear un script llamado runfixup.sh, que tiene que correr con el usuario root, para corregir todos los pre-requisitos necesarios que no tengan que ver con una modificación directa con un tamaño de disco o instalar paquetes, como por ejemplo los parámetros de Kernel necesarios.



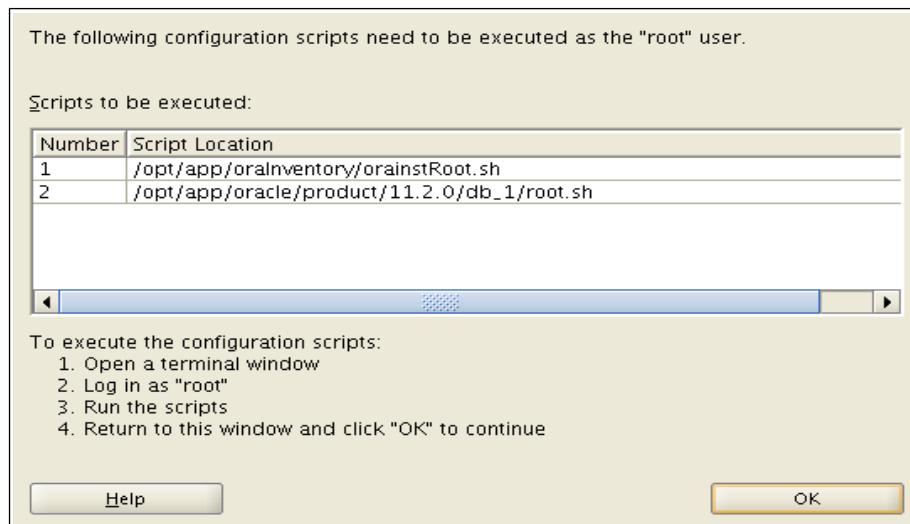
Una vez que arreglamos todos los pre-requisitos necesarios, le damos siguiente y nos lleva a una pantalla de resumen de instalación, en donde podemos verificar si queremos hacer algún cambio a nuestra instalación, si estamos de acuerdo le damos siguiente e inicia la instalación. En caso de ignoramos los mismo. Resumen, Se presenta el resumen de la instalación, en este punto se presiona el botón terminar.



Producto, En esta interfaz comienza el proceso de instalación del motor de base de datos. Despues de un alto porcentaje de avance se despliega una ventana indicando que la creación de la base de datos ha terminado, en la cual se debe presionar el botón aceptar. Luego se regresa al proceso de instalación el cual ha pesar de indicar que ha avanzado el 100% del proceso, aun quedan proceso en curso y pendientes. Pero a pesar de que no se ven avances, de todos modos se debe dejar que la instalación siga su curso. Despues aparece un mensaje indicando dos archivos que deben ser ejecutados, en esta ventana se debe presionar el botón Aceptar.



Por ultimo, tenemos que correr los scripts que te menciona el instalador, estos se corren con el usuario root.



Scripts orainstRoot.sh, Cambia los privilegios y grupos

```
File Edit View Search Terminal Help
[root@oraserv5 ~]# /opt/app/oraInventory/orainstRoot.sh
Changing permissions of /opt/app/oraInventory.
Adding read,write permissions for group.
Removing read,write,execute permissions for world.

Changing groupname of /opt/app/oraInventory to oinstall.
The execution of the script is complete.
[root@oraserv5 ~]#
```

root.sh script realiza muchas cosas, a saber: Cambia o establece correctamente las variables de entorno copia de algunos archivos en bin /usr/ local /, los archivos se dbhome, oraenv, etc coraenv creación de /etc/oratab archivo o la adición de la base de datos y la entrada en home de SID en /etc/oratab.

```
File Edit View Search Terminal Help
[root@oraserv5 ~]# /opt/app/oracle/product/11.2.0/db_1/root.sh
Running Oracle 11g root.sh script...

The following environment variables are set as:
ORACLE_OWNER= oracle
ORACLE_HOME= /opt/app/oracle/product/11.2.0/db_1

Enter the full pathname of the local bin directory: [/usr/local/bin]:
Copying dbhome to /usr/local/bin ...
Copying oraenv to /usr/local/bin ...
Copying coraenv to /usr/local/bin ...

Creating /etc/oratab file...
Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
Finished product-specific root actions.
[root@oraserv5 ~]#
```

DBCA (DataBase Configuration Assistant)

En Unix/Linux se ejecuta con "dbca". Se trata de un asistente gráfico que permite:

Crear una BD. A partir de plantillas predefinidas, que contienen:

- Sólo la estructura. Se pueden cambiar todos los parámetros del init.
 - También con ficheros: no se puede cambiar nada.
- Añadir opciones a una BD existente (Java, Intermedia, Spatial, OLAP, etc).
- Borrar una BD.
- Gestionar plantillas de BD: crear una plantilla a partir de otra, o a partir de sólo la estructura de una BD, o también incluyendo los datos de la misma.

Creación de Una Base Datos

Existen dos modos de crear una base de datos en Oracle. Usando el asistente Creación de una base de datos, el Oracle Database Configuration y el otro modo es por comandos:

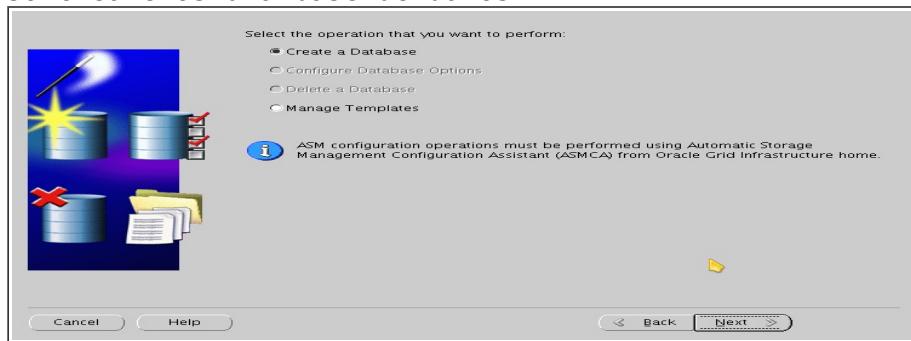
Creación de una Base de Datos con DBCA

Lo primero que veremos sera la pantalla de bienvenida. Presionar el botón "siguiente":

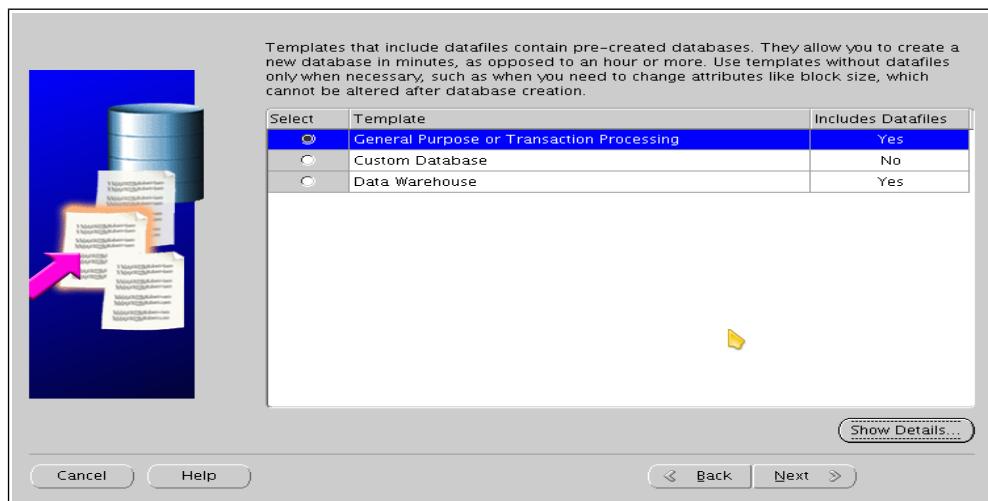
Seleccionamos la opción a realizar:

- Crear base de datos.
- Configurar opciones de base de datos
- Suprimir base de datos
- Gestionar plantilla
- Configurar la gestión automática de almacenamiento

En este caso crearemos una base de datos.

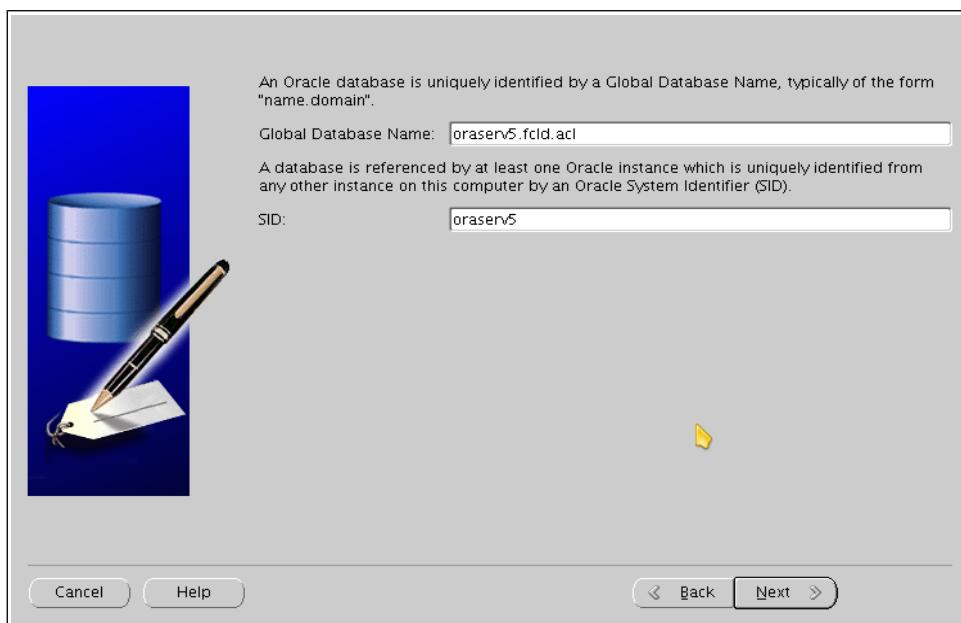


Seleccionamos una plantilla para crear la base de datos, en nuestro caso una de propósito general.

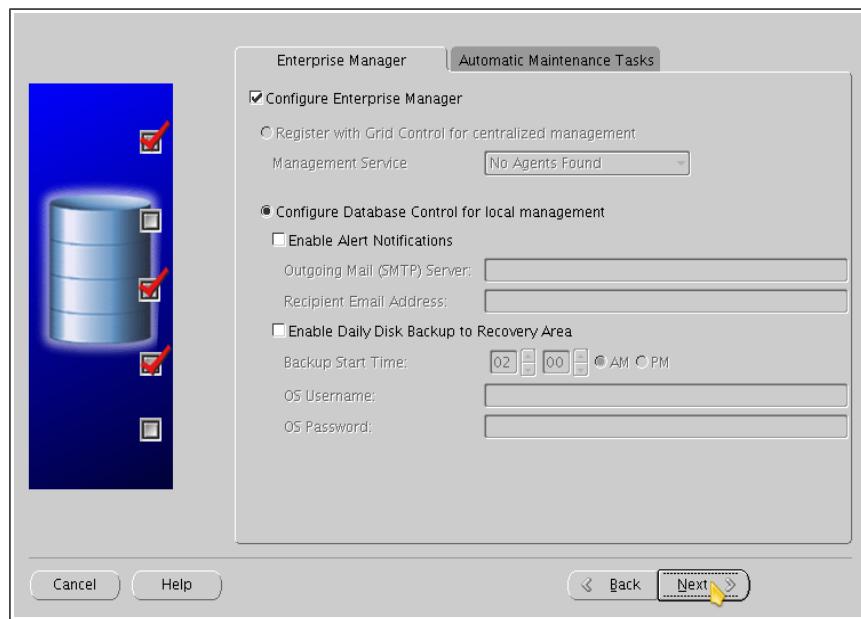


Elegimos el nombre global de la base de dato y del sid. Oracle recomienda que se una nombre calificado(fqdn), nosotros usamos el nombre del equipo como nombre global y la primera parte del nombre como nombre del sid.

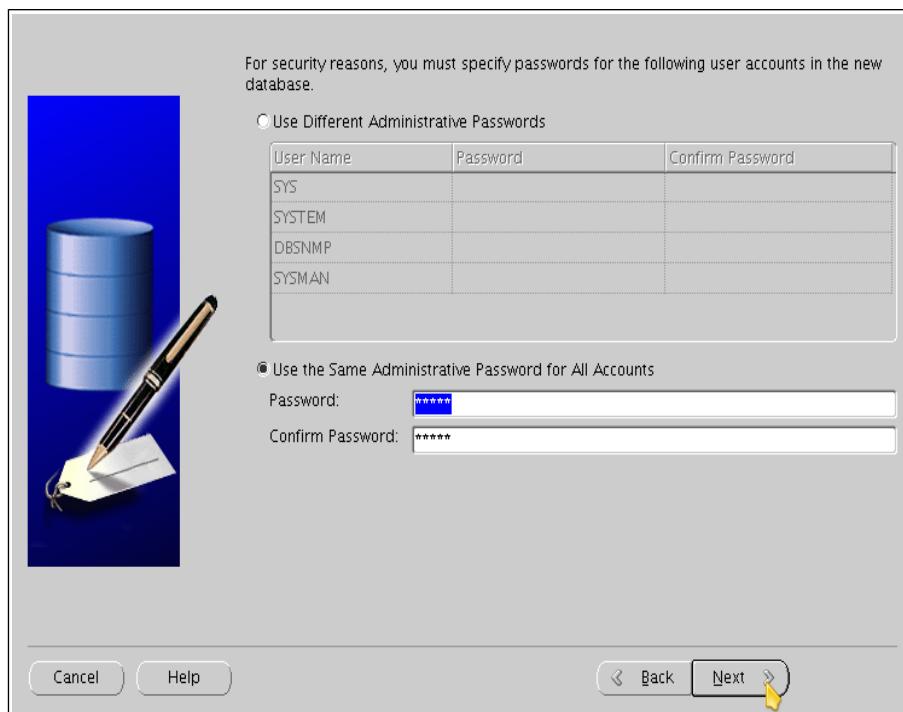
- oraserv5.fcl.dacl
- oraserv5



Lo siguiente es configurar la base de datos con Enterprise Manager.

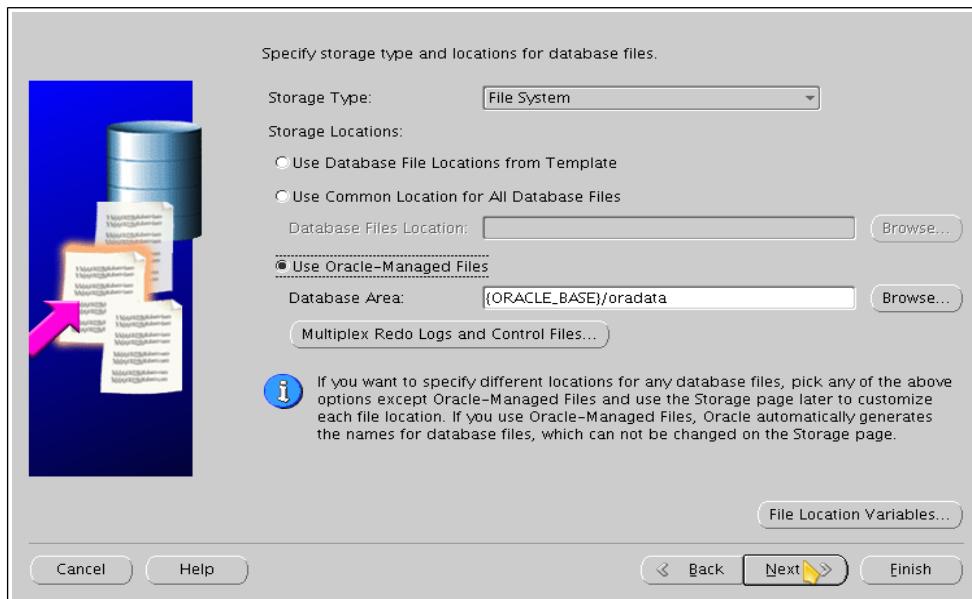


Elegimos las contraseña para los usuarios que crea oracle.



Seleccionamos el mecanismo de almacenamiento de la base de dato. Por ahora solo file system.

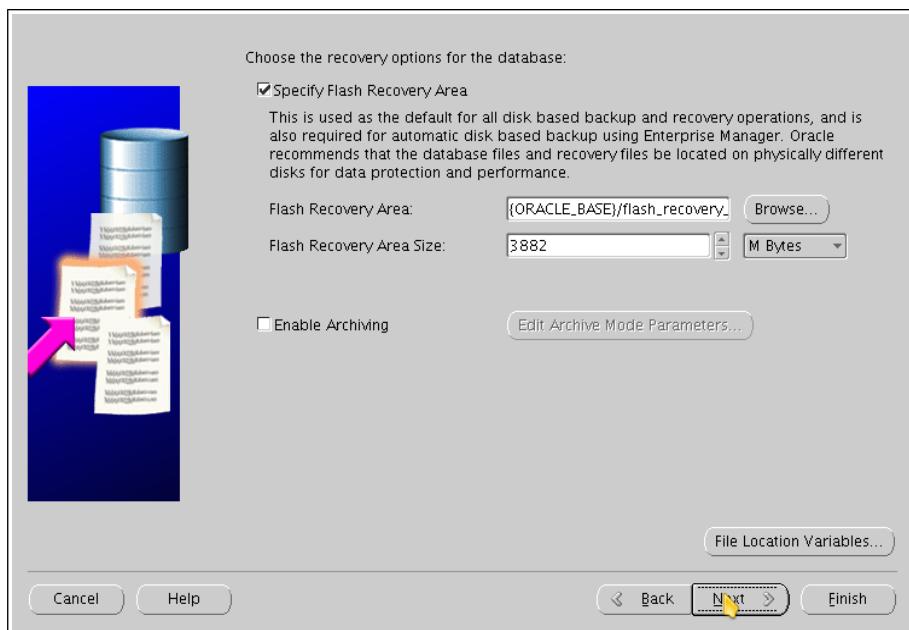
Seleccionamos el OMF.



Seleccionamos las ubicaciones en la que crearemos los archivos de la base de dato. Por ahora elegimos la opción de usar ubicaciones de base de datos de plantilla.

Elegimos las opciones de recuperación para la base de datos.

- {ORACLE_BASE}/flash_recovery_area

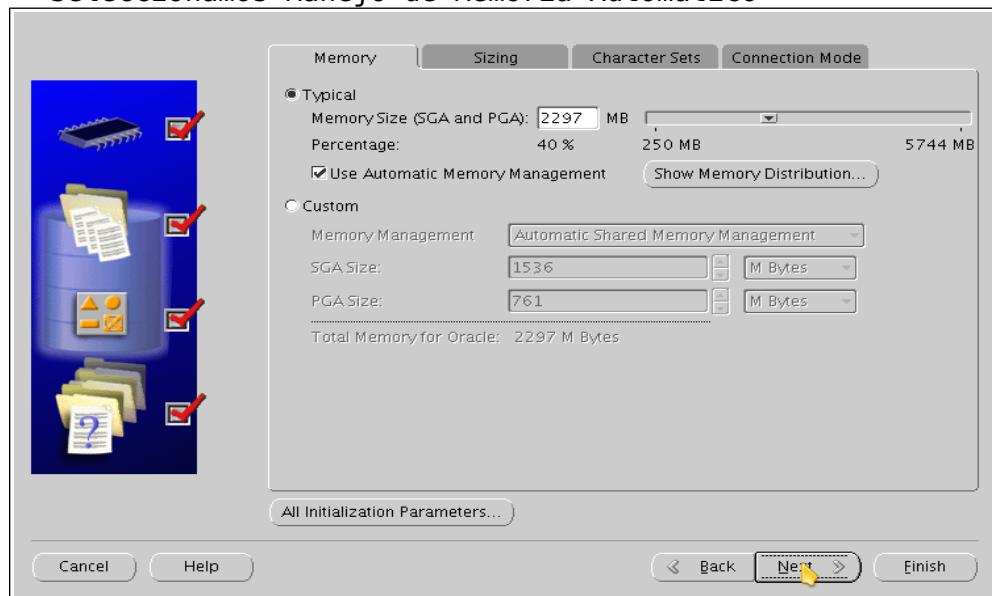


Instalamos los esquema de ejemplos. Nos servirá para las practicas.

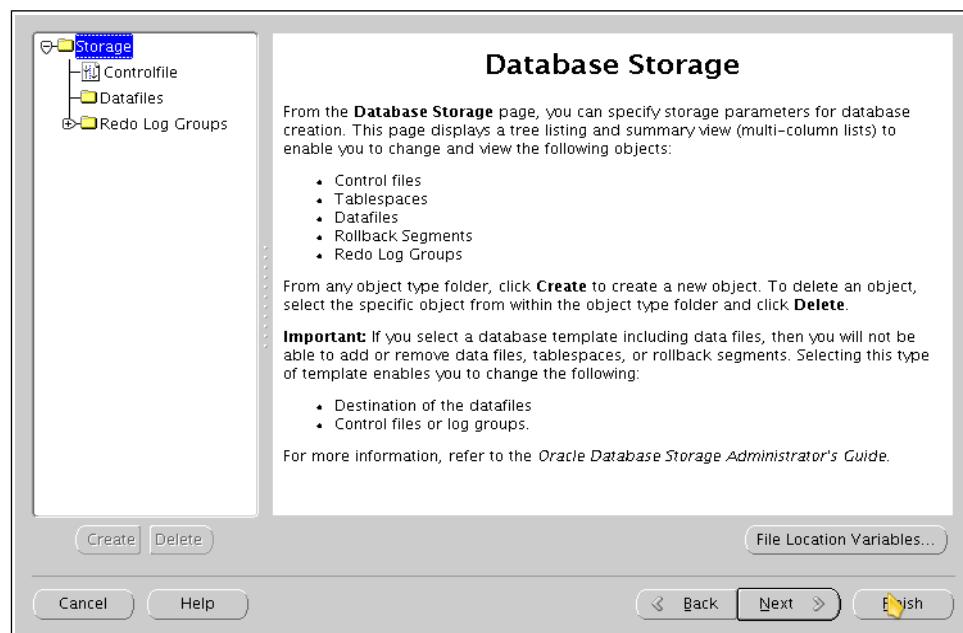


La siguiente pantalla nos pedirá datos como la memoria, tamaño, juego de caracteres y modo de conexión Los valores por defecto nos servirán.

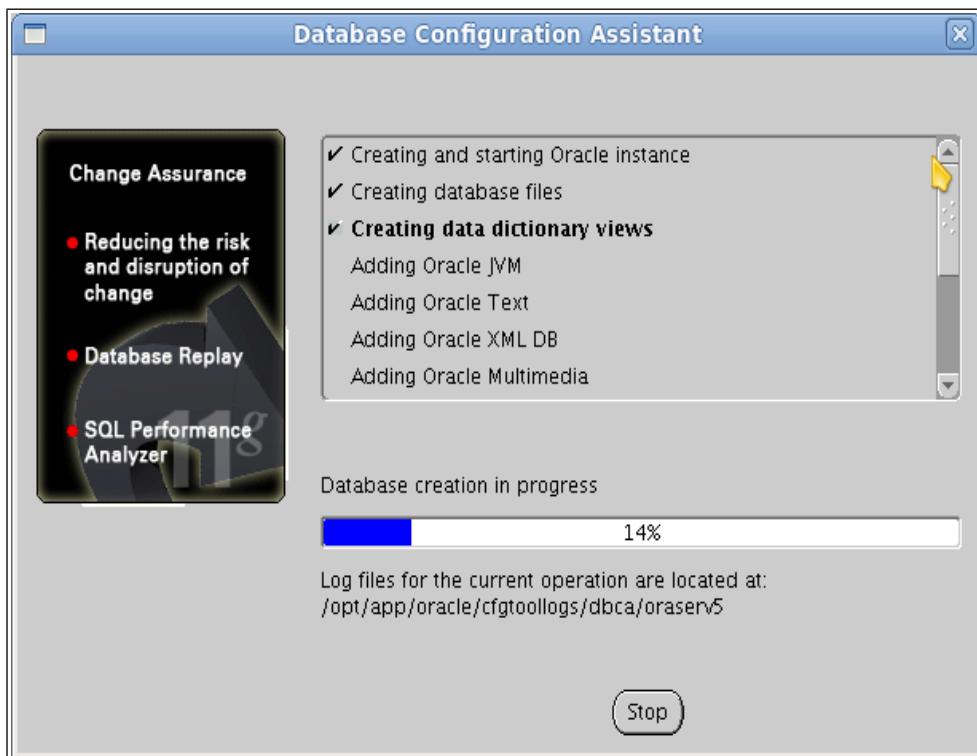
- Seleccionamos Manejo de Memoria Automático



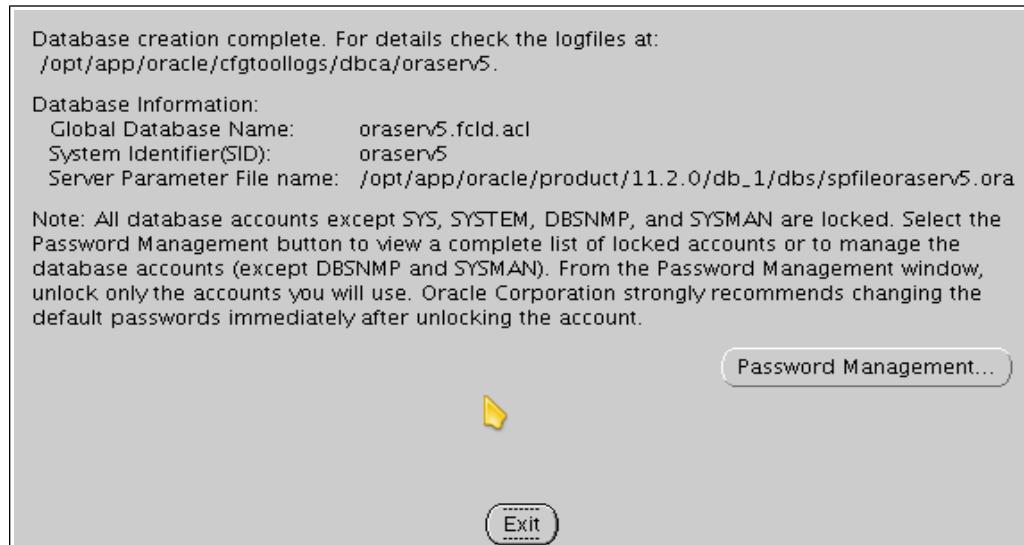
Por ultimo veremos el resumen de los detalles de la base de datos a crear.



Inicio de la Instalación:



Resumen Final:



Enterprise Manager :)

Bienvenido x Facebook x codigolibre x traductor x requisitos x cPanel X x cPanel x Oracle Ent... x Gestion di... x

ORACLE Enterprise Manager 11g Database Control Conectado como SYS

Instancia de Base de Datos: oraserv5.acl

General

- Estatus: Activo
- Activo desde: 26-ene-2012 22H56' GMT-04:00
- Nombre de la Instancia: oraserv5
- Versión: 11.2.0.1.0
- Estado de Escucha: Activo
- Listener: LISTENER, localhost

CPU del Host

Sesiones Activas

Tiempo de Respuesta SQL

Resumen de Diagnósticos

Resumen de Espacio

Alta Disponibilidad

Alertas

Violaciones de Política

Actividad del Trabajo

Enlaces Relacionados

Estructuras de Almacenamiento

Objetivos: En esta capítulo explicaremos el concepto de almacenar los objetos de la Bases Datos físicos y lógicos y ubicación de los mismos.

Estructuras de Almacenamiento

Una base de datos se divide en unidades lógicas de almacenamiento denominadas tablespaces. Cada tablespace se compone de muchos bloques lógicos de datos de Oracle. El parámetro DB_BLOCK_SIZE especifica el tamaño que tiene un bloque lógico. El rango de tamaño de un bloque lógico es de 2 KB a 32 KB. El tamaño por defecto es 8 KB. Un número determinado de bloques lógicos contiguos constituye una extensión. Un juego de extensiones que se asignan según una estructura lógica concreta constituye un segmento. Un bloque de datos de Oracle representa la unidad más pequeña de E/S lógica.

Almacenamiento de Datos de Tabla

Al crear una tabla, se crea un segmento para contener los datos. Un tablespace contiene una recopilación de segmentos. Desde el punto de vista lógico, una tabla contiene filas de valores de columna. En última instancia, una fila se almacena en una base de datos en forma de una parte de una fila. Se denomina parte de una fila porque en determinadas circunstancias puede que la fila entera no se almacene en un solo lugar. Esta situación se produce cuando una fila insertada es demasiado grande para caber en un solo bloque o cuando una actualización hace que el espacio actual de una fila existente se quede pequeño.

Estructuras de Bases de Datos Físicas y Lógicas

Una base de datos Oracle es una recopilación de datos que se trata como una unidad. El objetivo general de una base de datos es almacenar y recuperar la información relacionada. La base de datos tiene estructuras lógicas y físicas.

Tablespaces

Una base de datos se divide en unidades lógicas de almacenamiento denominadas tablespaces, que agrupan estructuras lógicas relacionadas. Por ejemplo, los tablespaces suelen agrupar todos los objetos de una aplicación para simplificar algunas operaciones administrativas. Puede tener un tablespace para los datos de la aplicación y otro adicional para los índices de la aplicación.

Bases de Datos, Tablespaces y Archivos de Datos

La relación entre las bases de datos, los tablespaces y los archivos de datos se ilustra en la diapositiva. Cada base de datos está dividida de manera lógica en uno o más tablespaces. Se crean explícitamente uno o más archivos de datos para cada tablespace con el fin de almacenar físicamente los datos de todas las estructuras lógicas de un tablespace. Si se trata de un tablespace TEMPORARY, en lugar de un archivo de datos, el tablespace tendrá un archivo temporal.

La Base de Datos desde la perspectiva lógica.

Base Datos .

Es un conjunto de datos que tienen un representan una información captada del mundo real, con ellos se puede realizar diversos procesos.

Esquemas

Un esquema es una recopilación de objetos de base de datos propiedad de un usuario de la base de datos. Los objetos de esquema son estructuras lógicas que hacen referencia directa a datos de la base de datos. Los objetos de esquema incluyen estructuras como, por ejemplo, tablas, vistas, secuencias, procedimientos almacenados, sinónimos, índices, clusters y enlaces de base de datos. En general, los objetos de esquema incluyen todo lo que la aplicación cree en la base de datos.

Extensiones

Al nivel siguiente del espacio de la base de datos lógica se denomina extensión. Una extensión es un número específico de bloques de datos contiguos (obtenidos en una única asignación) que se utilizan para almacenar un tipo determinado de información.

Segmentos

Al nivel de almacenamiento de la base de datos lógica por encima de una extensión se denomina segmento. Un segmento es un juego de extensiones asignadas para una determinada estructura lógica. Por ejemplo, los diferentes tipos de segmentos incluyen:

- **Segmentos de datos:** Cada tabla no de cluster y no organizada por índices tiene un segmento de datos. Todos los datos de la tabla se almacenan en las extensiones de su segmento de datos. Para una tabla particionada, cada partición tiene un segmento de datos. Cada cluster tiene un segmento de datos. Los datos de cada tabla del cluster se almacenan en el segmento de datos del cluster.
- **Segmentos de índice:** Cada índice tiene un segmento de índice que almacena todos sus datos. Para un índice particionado, cada partición tiene un segmento de índice.

- **Segmentos de deshacer:** El administrador de la base de datos crea un tablespace UNDO para almacenar temporalmente información de deshacer. La información de un segmento de deshacer se utiliza para generar información de base de datos de lectura consistente y, durante la recuperación de la base de datos, para realizar una operación de rollback de las transacciones sin confirmar para los usuarios.
- **Segmentos temporales:** La base de datos Oracle crea segmentos temporales cuando una sentencia SQL necesita un área de trabajo temporal para terminar la ejecución. Cuando la sentencia termina la ejecución, las extensiones del segmento temporal vuelven a la instancia para un uso futuro. Especifique un tablespace temporal por defecto para cada usuario o un tablespace temporal por defecto que se utilice en toda la base de datos.

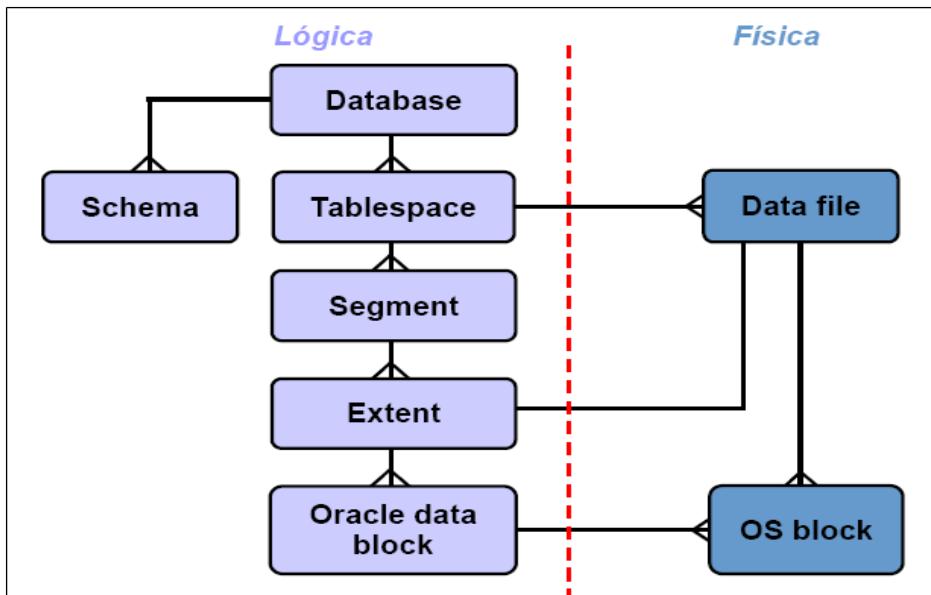
La base de datos Oracle asigna el espacio dinámicamente. Cuando las extensiones existentes de un segmento están completas, se agregan extensiones adicionales. Debido a que las extensiones se asignan según sea necesario, las extensiones de un segmento pueden o no ser contiguas en el disco.

Bloques de Datos

Al nivel más detallado de granularidad, los datos de una base de datos Oracle se almacenan en bloques de datos. Un bloque de datos corresponde a un número concreto de bytes de espacio de la base de datos física en el disco. Se especifica un tamaño del bloque de datos para cada tablespace cuando se crea. Una base de datos utiliza y asigna espacio libre de la base de datos en bloques de datos Oracle.

Un bloque es la unidad mínima de almacenamiento de información de Oracle. A los bloques también se les conoce como "bloques de datos", "bloques lógicos" o "bloques oracle". Cada uno de estos bloques está formado por un número determinado de bloques del sistema operativo. A la hora de crear una nueva base de datos se debe indicar cuántos bloques de sistema operativo formarán un bloque de datos o bloque oracle. Es muy importante decidir bien este valor de antemano ya que una vez creada la base de datos ya no se puede modificar más que en migraciones a versiones más actuales del producto.

Un bloque de datos es la mínima unidad de Lectura / Escritura en una base de datos Oracle, es decir, Oracle no lee y escribe en bloques del sistema operativo sino que lo hace en unidades lógicas que son los bloques de datos y que varían de una base de datos a otra en la misma máquina ya que es un valor que se debe indicar en la creación de cada base de datos Oracle.



Cuentas Predefinidas: SYS y SYSTEM

La cuenta SYS:

- Tiene el rol DBA otorgado.
- Tiene todos los privilegios con ADMIN OPTION.
- Es necesaria para el inicio, el cierre y para algunos comandos de mantenimiento.
- Es propietaria del diccionario de datos.
- Es propietaria del repositorio de carga de trabajo automática (AWR).

La cuenta SYSTEM tiene otorgado el rol DBA.

Estas cuentas no se utilizan para operaciones rutinarias.

Las cuentas SYS y SYSTEM tienen otorgado por defecto el rol de administrador de base de datos (DBA).

La cuenta SYS tiene además todos los privilegios con la opción ADMIN OPTION y es propietaria del diccionario de datos. Para conectarse a la cuenta SYS es necesario utilizar la cláusula AS SYSDBA. Cualquier usuario al que se le otorgue el privilegio SYSDBA puede conectarse a la cuenta SYS mediante la cláusula AS SYSDBA. Sólo los usuarios con "privilegios", a los que se les otorga el privilegio SYSDBA o SYSOPER, pueden iniciar y cerrar la instancia de base de datos.

A la cuenta SYSTEM se le otorga el rol DBA por defecto, pero no el privilegio SYSDBA. Práctica recomendada: Debido a la aplicación del principio de privilegio más bajo, estas cuentas no se utilizan para operaciones rutinarias. Los usuarios que necesiten privilegios DBA tienen cuentas separadas a las que se les otorgan los privilegios necesarios. Por ejemplo, Jim tiene una cuenta de privilegio bajo denominada jim y una cuenta con privilegios denominada jim_dba. Este método permite aplicar el principio de privilegio más bajo, elimina la necesidad de compartir cuentas y permite auditar acciones individuales.

Las cuentas SYS y SYSTEM son cuentas necesarias en la base de datos. No se pueden eliminar.

Como acceder a la parte lógica desde SqlPlus

Entrar y ejecutar comandos

El prompt por defecto será SQL>. Si queremos cambiar el prompt, utilizaremos la instrucción SET SQLPROMPT. Un comando puede ser introducido en múltiples líneas. Siempre tendremos en el buffer de Oracle el último comando ejecutado. El buffer puede ser editado o guardado en un fichero. Se puede finalizar un comando de diferentes formas:

- Con un punto y coma (;). El comando está completo y debe ejecutarse.
- Barra en una línea vacía (/). Se ejecuta el comando del buffer, También se utiliza para ejecutar un PL/SQL.
- Línea en blanco. El comando es almacenado en el buffer.

Los siguientes comandos también se pueden incluir en un archivo de comandos:

CONNECT: para conectarse como otro usuario.

HELP: para obtener ayuda en línea.

EXIT: para dejar SQL*Plus y salir al Sistema Operativo.

DESCRIBE o **DESC:** para obtener información sobre la estructura de una tabla.

HOST o **!:** para ejecutar un comando del Sistema Operativo.

Parámetros

SET LINESIZE: pone el número máximo de caracteres por línea. Por defecto vale 80 y el máximo es 999.

SET PAGESIZE: pone el número de filas de la salida antes de empezar una nueva página. Por defecto es 25. Incluye el título y las líneas de pausa.

SET HEADING [ON | OFF]: Activa/desactiva la utilización de encabezados de columnas. Por defecto está activado.

SET NULL texto: Indica la cadena de caracteres que hay que colocar en sustitución de los valores NULL. Por defecto es "".

SET ECHO [ON | OFF]: Activa/desactiva la visualización de los comandos que SQL*Plus ejecuta según van siendo tratados. Por defecto está desactivada.

SET FEEDBACK [n | ON | OFF]: Muestra el número de registros recuperados en cada consulta cuando se recuperan n o más registros. ON se pueden considerar como n=1, y OFF como n=0.

SET VERIFY [ON | OFF]: Controla la salida de confirmación para los valores de las variables de sustitución. Por defecto está activado.

Modo de accesar como sys:

```
[oracle@oraserv5 ~]$ sqlplus sys/kikla as sysdba
```

Desbloquear de cuenta hr y scott para realizar los próximos ejercicios:

```
SQL> alter user hr account unlock;
```

```
SQL> alter user hr identified by oracle;
```

```
SQL> alter user scott account unlock;
```

```
SQL> alter user scott identified by oracle;
```

Modo de accesar como hr:

```
SQL> conn hr/oracle
```

Registros de historial (Logs)

Es el fichero de registro (logs) de la BD y la primera referencia para el DBA en el trabajo del “día a día” de la administración de la misma.

- Por defecto está en \$ORACLE_HOME/rdbms/log; o en el directorio que indique el parámetro BACKGROUND_DUMP_DEST del init. Si usamos OFA, una ubicación típica es \$ORACLE_BASE/admin/\$ORACLE_SID/bdump.
- Recoge tanto información de estado como errores:
 - ✓ Arranque y parada
 - ✓ Parámetros del init sin valores por defecto,
 - ✓ Arranque de los procesos background,
 - ✓ Cambio de fichero redolog (log switch),
 - ✓ Creación de tablespaces y segmentos de rollback,
 - ✓ Comandos alter (alter database, alter tablespace, etc),
 - ✓ Errores del tipo ORA-600 y otros tipos de errores que indican falta de espacio en los sistemas /opt/app/oracle/product/11.2.0/db_1/network/admin/sqlnet.oraz (llenado de tablas, índices, tablespaces, etc).

Introducción SQL

Objetivos: En esta capítulo explicamos una breve introducción al estándar SQL, con la finalidad que pueda discernir entre los diversos componentes de SQL, su manipulación y comprensión, para poder entender los scripts que manejaremos mas adelante en los posteriores capítulos

SQL es un lenguaje de consulta estructurado surgido de un proyecto de investigación de IBM para el acceso de las bases de datos relacionadas en los años 70s. Actualmente es el estándar de los lenguajes de base de datos y la mayoría de los sistemas de bases de datos lo soportan.

Componentes del SQL

SQL permite realizar consultas a la base de datos, pero además provee funciones de definición, control y gestión de la base de datos. Las sentencias SQL se clasifican, según su finalidad, en varios componentes o sub-lenguajes:

- **DDL (data definition language)**, lenguaje de definición de datos, incluye órdenes para definir, modificar o borrar las tablas en las que se almacenan los datos y de las relaciones entre estas.
- **DCL (data control language)**, lenguaje de control de datos, contiene elementos útiles para trabajar en un entorno multiusuario, en el que es importante la protección de los datos, la seguridad de las tablas y el establecimiento de restricciones de acceso.
- **DML (data manipulation language)**, lenguaje de manipulación de datos, nos permite recuperar los datos almacenados en la base de datos y también incluye órdenes permitir al usuario actualizar, insertar o modificar datos existentes en la tabla.

Lenguaje de Descripción de Datos (DDL):

- **CREATE:** usado para crear nuevas tablas, índices, secuencias y otros objetos. La sintaxis es la siguiente:

```
SQL> CREATE TABLE fcld (
    id_estu number,
    nombre varchar2(10));
```

- **INSERT:** Usados para cargar lotes de datos en la base de datos en una única operación

```
SQL> INSERT INTO fcld (id_estu, nombre) VALUES (12, 'Perpinan');
```

```
SQL> INSERT INTO scott.dept (deptno, dname) VALUES (45, 'jaguar');
```

- **DROP:** empleado para eliminar tablas, indices, secuencias y otros objetos. Su sintaxis es:

```
SQL> DROP TABLE nombre_tabla;
```

- **ALTER:** usado para modificar la tabla agregando campos o cambiando la definición de estos, también sirve para la cambiar la definición de algunos objetos:

```
SQL> ALTER TABLE fcld ADD (edad number);
```

```
SQL> ALTER TABLE fcld ADD CONSTRAINT a UNIQUE (id_estu);
```

Lenguaje de Manipulación de Datos (DML):

- **SELECT:** Usado para consultar registros de una base de datos que satisfagan un criterio determinado, la clausula From en el select es obligatorio.

```
SQL> SELECT col1, col2 FROM fcld;
```

```
SQL> SELECT ename, sal, hiredate FROM emp;
```

- **UPDATE:** Utilizado para modificar valores existente en una tabla.

```
SQL> UPDATE fcld SET (id_estu, nombre) = (15, 'Paredes');
UPDATE emp
      SET (job, deptno) = (select job, deptno from empno = 7499)
WHERE empno = 7698;
```

- **DELETE:** Usado para eliminar registros de una tabla.

```
SQL> DELETE from fcld where nombre='Paredes';
```

```
SQL> DELETE FROM emp
      WHERE deptno=20;
```

Clausulas

- **FROM:** Utilizada para especificar la tabla de la cual se van a extraer los datos.

```
SQL> SELECT * FROM dept;
```

- **WHERE:** usado para especificar las condiciones que debe reunir los registros que se van a seleccionar.

```
SQL> SELECT dname FROM dept WHERE deptno = 30;
```

- **GROUP BY:** Utilizada para separar los registros seleccionados en grupos específicos
- **HAVING:** Utilizada para expresar la condición que debe satisfacer cada grupo.
- **ORDER BY:** Utilizado para ordenar los registros seleccionados de acuerdo con un orden en específico.

Operadores Lógicos

- **AND:** es el conectivo y, evaluá dos condiciones y devuelve un valor de verdad solo si ambos son ciertos.

```
SQL> SELECT first_name, salary  
      FROM employees  
     WHERE last_name = 'Smith' AND salary > 7500;
```

- **OR:** es el conector lógico O, evaluá dos condiciones y devuelve un valor de verdad si al menos uno de los dos es cierto.

```
SQL> SELECT first_name, salary  
      FROM employees  
     WHERE first_name = 'Kelly' OR last_name = 'Smith';
```

- **NOT:** Negación lógica, devuelve el valor contrario de la expresión.

```
SQL> SELECT first_name, department_id  
      FROM employees  
     WHERE NOT ( department_id >= 30 );
```

Operadores de Comparación:

- > mayor que...

```
SQL> SELECT first_name || ' ' || last_name, commission_pct
  FROM employees
 WHERE commission_pct > 0.35;
```

- < menor que...

```
SQL> SELECT first_name || ' ' || last_name, commission_pct
  FROM employees
 WHERE commission_pct < 0.15;
```

- <>, != diferente de, distinto a...

```
SQL> SELECT first_name || ' ' || last_name, commission_pct
  FROM employees
 WHERE commission_pct <> 0.35;
```

- BETWEEN. Usado para especificar un intervalo de valores.

```
SQL> SELECT first_name, last_name
  FROM employees
 WHERE salary BETWEEN 5000 AND 6000;
```

- LIKE. Usado en la comparación de un modelo.

```
SQL> SELECT first_name, last_name
  FROM employees
 WHERE first_name LIKE 'Su%';
```

- IN. usado para definir que el valor a comparar se encuentre dentro de un grupo especificado en in.

```
SQL> SELECT first_name, last_name, department_id
  FROM employees
 WHERE department_id IN (10, 20, 90);
```

Funciones de Agregados

- AVG. Usado para calcular el promedio de los valores de un campo determinado.

```
SQL> SELECT AVG(sal)
  FROM emp;
```

- **COUNT.** Usado para devolver el numero de registro de selección.

```
SQL> SELECT COUNT(*)  
      FROM emp;
```

- **SUM.** Usado para devolver la sumatoria de todos los valores de un campo determinado.

```
SQL> SELECT SUM(sal)  
      FROM emp;
```

- **MAX.** Usado para devolver el valor mas alto de un campo especifico.

```
SQL> SELECT MAX(sal)  
      FROM emp;
```

- **MIN.** Usado para devolver el menor valor de un campo especifico.

```
SQL> SELECT MIN(sal)  
      FROM emp;
```

Diccionario de Datos

Objetivos: Aprender a utilizar las herramientas provistas para el análisis de los elementos de entrada y salida de datos, así como la clasificación de los diferentes objetos existentes en la base de datos.

¿Qué es el Diccionario de Datos?

El diccionario contiene datos sobre los objetos almacenados en la base de datos y otra información concerniente al sistema de gestión de la base de datos en si. El DD está compuesto por un conjunto de tablas y vistas asociadas donde se almacena toda la información sobre los objetos que componen la BD, así como la estructura lógica y física de la misma.

El Diccionario de Datos incluye dos tipos de objetos: tablas base y vistas.

- Las tablas base se crean automáticamente cuando creamos la BD con el comando `CREATE DATABASE;` y son las que realmente contienen la información del Diccionario de Datos.
- Las vistas se crean al lanzar el script `catalog.sql;` y permiten acceder a la información de las tablas del Diccionario de Datos (que está codificada).

El Diccionario de Datos contiene información sobre: la definición de todos los objetos la BD (tablas, vistas, índices, sinónimos, secuencias, procedimiento, funciones, paquetes, triggers, etc), el espacio ocupado por cada objeto, condiciones de integridad, usuarios, privilegios, roles, así como algunos segmentos de auditoria del sistema. Las columnas de todas las tablas del diccionario están descritas en la tabla `DICT_COLUMNS`. El resto de tablas del diccionario de Oracle se dividen en cuatro categorías: `USER`, `ALL`, `DBA` y `V$`.

- Las vistas cuyo nombre comienza por `ALL_` , pueden ser consultadas por todos los usuarios y ofrecen información sobre todos los objetos del sistema.

ALL_TABLES, ALLSEQUENCES, ALL_VIEWS

- Las vistas cuyo nombre comienza por `USER_` , ofrecen información para listar los objetos propios.

USER_TABLES, USERSEQUENCES, ...

Las vistas cuyo nombre comienza por **DBA_**, sólo son accesibles para tareas de administración.

```
DBA_TABLESPACES, DBA_DATA_FILES, DBA_CONSTRAINTS, DBA_VIEWS
```

- Cuando se entiende toda la información del diccionario, se entiende el sistema de gestión de bases de datos. El sistema no tiene otra información: todo lo que se refleja en el diccionario. También se puede ver la estructura de una tabla del diccionario:

```
SQL> DESCRIBE ALL_TABLES;
```

```
SQL> DESCRIBE ALL_INDEXES;
```

```
SQL> DESCRIBE ALLSEQUENCES;
```

Las vistas que comienzan con V\$ son las denominadas tablas dinámicas, se crean al arrancar la instancia y residen en memoria. Cuando cerramos la instancia desaparece su contenido.

```
V$instance, V$DATABASE, V$PROCESS, V$PARAMETER
```

Consultas Útiles en el Diccionario de Datos

Listar las tablas, índices y secuencias definidas por el usuario scott:

```
SQL> SELECT TABLE_NAME  
      FROM ALL_TABLES  
     WHERE OWNER = 'SCOTT';
```

Para los índices:

```
SQL> SELECT TABLE_NAME, INDEX_NAME  
      FROM ALL_INDEXES  
     WHERE OWNER = 'SCOTT';
```

Para las secuencias:

```
SQL> SELECT SEQUENCE_NAME  
      FROM ALL_SEQUENCES  
     WHERE SEQUENCE_OWNER = 'SCOTT';
```

Tabla **USER_CATALOG**: Contiene información sobre las tablas y vistas definidas por un usuario. El esquema de esta tabla es:

```
USER_CATALOG(TABLE_NAME, TABLE_TYPE);
```

- También podemos referirnos a ella usando su sinónimo público CAT. La siguiente instrucción SQL muestra todas las tablas y vistas definidas por el usuario actual:

```
SQL> SELECT * FROM CAT;
```

- Tabla USER_OBJECTS: Contiene información sobre todos los objetos definidos por el usuario actual. Además de la información accesible a través de USER_CAT, usando USER_OBJECTS tendremos acceso a las vistas, funciones procedimientos, índices, sinónimos, disparadores, etc. El esquema de la tabla es el siguiente:

USER_OBJECTS	(OBJECT_NAME, OBJECT_ID, OBJECT_TYPE, CREATED, LAST_DDL_TIME, TIMESTAMP, STATUS)
--------------	---

Donde:

- CREATED indica cuándo fue creado el objeto.
- TIMESTAMP es lo mismo que CREATED pero en formato STRING.
- LAST_DDL_TIME indica el último acceso por una instrucción DDL.
- STATUS indica si el objeto es válido o no.

Tabla USER_TABLES: Si queremos obtener más información sobre las tablas que su nombre no basta con usar USER_CAT. Hay que usar USER_TABLES o su sinónimo TABS. Podemos obtener su esquema utilizando DESCRIBE USER_TABLES;

Tabla USER_TAB_COLUMNS: Almacena información detallada sobre las columnas de las tablas. Su sinónimo público es COLS.

Tabla USER_VIEWS: Almacena información sobre las vistas definidas por un usuario. El esquema de esta tabla es:

USER_VIEWS(VIEW_NAME, TEXT_LENGTH, TEXT)
--

Donde TEXT es el texto de la vista y TEXT_LENGTH la longitud del mismo.

Tabla USER_CONSTRAINTS: Almacena información sobre las restricciones definidas por el usuario sobre las tablas.

SQL> SELECT TABLE_NAME, CONSTRAINT_TYPE, CONSTRAINT_NAME FROM USER_CONSTRAINTS;;

Donde el tipo de restricción es:

- C para indicar NOT NULL
- P para indicar PRIMARY KEY
- R para indicar REFERENCES

Los nombres de las restricciones son los proporcionados por el usuario o por el sistema cuando aquél no dio ninguno.

- Tabla ALL_CONS_COLUMNS: Almacena información sobre las columnas a las que se refiere una restricción de clave primaria o de integridad referencial.

```
SQL> SELECT COLUMN_NAME  
      FROM ALL_CONS_COLUMNS  
     WHERE CONSTRAINT_NAME = 'SYS_C003454';
```

Da como salida las columnas que forman la clave primaria Tabla USER_TRIGGERS: Almacena información sobre los disparadores definidos sobre las tablas de usuario.

```
SQL> SELECT TRIGGER_TYPE, TABLE_NAME, TRIGGERING_EVENT  
      FROM USER_TRIGGERS  
     WHERE TRIGGER_NAME = 'Actualiza Estadísticas';
```

- Para saber donde se encuentran físicamente nuestros archivos de control.

```
SQL> SELECT * FROM V$CONTROLFILE;
```

Para ver el nombre de la instancia y su estado.

```
SQL> SELECT INSTANCE_NAME, DATABASE_STATUS  
      FROM V$INSTANCE;
```

Para obtener información acerca de los procesos iniciados.

```
SQL> SELECT * FROM V$PROCESS;
```

Para conocer el nombre de la base de datos actual;

```
SQL> SELECT NAME FROM V$DATABASE;
```

VISTAS DEL DD

V\$MEMORY_DYNAMIC_COMPONENTS
V\$MEMORY_TARGET_ADVICE
V\$MEMORY_RESIZE_OPS
V\$DIAG_INFO
V\$DATABASE (Base de datos).
V\$INSTANCE (Instancia).
V\$SGA (SGA).
V\$SGAINFO (Gestión dinámica de la SGA).
V\$SGASTAT (SGA detallada).
V\$BUFFER_POOL (Buffers en la caché de datos)
V\$SQLAREA (Sentencias SQL).
V\$PROCESS (Procesos).
V\$BGPROCESS (Procesos background).
V\$DATAFILE (Ficheros de datos de la BD).
V\$CONTROLFILE (Ficheros de control de la BD).
V\$LOGFILE (Ficheros redo log de la BD).
DBA_TABLESPACES (Tablespaces de la BD).
DBA_SEGMENTS (Segmentos que hay en los tablespaces).
DBA_EXTENTS (Extensiones que componen los segmentos).
DBA_USERS: Todos los usuarios de la BD.
DBA_TS_QUOTAS: Cuotas de tablespace de los usuarios.
DBA_SYS_PRIVS: Privilegios de sistema concedidos a usuarios o roles.
DBA_TAB_PRIVS: Privilegios sobre objetos concedidos a usuarios o roles.
DBA_ROLES: Todos los roles de la BD
DBA_ROLE_PRIVS: Roles concedidos a usuarios.
ROLE_ROLE_PRIVS: Roles concedidos a otros roles.
DBA_TABLESPACES: descripción de los tablespaces.
DBA_PROFILES: muestra los perfiles y sus límites.
DBA_DATA_FILES: información a cerca de los ficheros de archivos de la base de datos .
DBA_INDEXES: descripción de todos los índices de la base de datos.
DBA_IND_COLUMNS: COLUMNS comprising INDEXEs on all TABLEs and CLUSTERS .

Arranque y Parada

Objetivos: En este capítulo trataremos sobre la estructura de inicio y detención de los servicios provistos por la base de datos y sus instancias en si, así como las tareas administrativas realizadas fuera de linea (offline).

FICHEROS DE PARÁMETROS DE INICIALIZACIÓN

Para arrancar la instancia, el servidor Oracle tiene que leer el fichero de parámetros de inicialización (spfile o init), cuya ubicación predeterminada es \$ORACLE_HOME/dbs.

El fichero de parámetros de inicialización puede ser de dos tipos: Hay determinadas operaciones que requieren parar la BD; como la actualización de algunos parámetros del init.ora; o hacer una copia física de la BD (copia en frío).

init: se trata de un fichero de texto, editable, cuyo nombre sigue el patrón:

`init$ORACLE_SID.ora.`

spfile: es un fichero binario, no editable pero visualizable, cuyo nombre sigue el patrón:

`spfile$ORACLE_SID.ora.`

Se crea, a partir de un init, con:

```
CREATE SPFILE [=‘nombre’] FROM PFILE [=‘nombre’];
```

- Nota1. Si se omiten los nombres, toma los valores por defecto.
- Nota2. La BD no podrá abrir el nuevo spfile hasta el siguiente arranque.
- Nota3. Se puede crear un init a partir de un spfile, invirtiendo la sintaxis.

Los parámetros del spfile se modifican con:

```
ALTER SYSTEM SET parámetro = valor [SCOPE = MEMORY | SPFILE | BOTH]
```

Nota. Si sólo queremos modificar el parámetro en el spfile, indicaremos SPFILE. Para hacer el cambio solo en memoria, especificar MEMORY.

➤ **Hay dos tipos de parámetros:**

- Explícitos: los que se indican en el fichero de parámetros.
- Implícitos: el resto, que tomarán un valor por defecto.
 - La forma de indicar valor a los parámetros es `parametro=valor`.
 - El símbolo # indica el comienzo de un comentario, pudiendo estar al principio o en medio de la línea.
 - En el `init.ora`, el parámetro `ifile` permite incluir otros ficheros con parámetros.
 - Una lista de valores se indicará entre paréntesis, separando los valores por comas.
 - Para indicar un valor de tipo cadena de caracteres hay que encerrarlo entre comillas simples.
 - Si usamos OFA, la ubicación típica para el `init.ora` es `$ORACLE_BASE/admin/$ORACLE_SID/pfile`. Después creamos un enlace en `$ORACLE_HOME/dbs` (ubicación por defecto).

El `init.ora`:

Nota. Es un fichero de texto que hay que mantener manualmente con un editor ASCII (vim, emacs, etc).

Arrancar la Base Datos

El arranque de una base de datos ORACLE requiere tres etapas:

1. Arrancar la instancia
2. Montar la base de datos
3. Abrir la base de datos

Arrancar la base de datos

En esta parte del arranque se generan los procesos background.

Se crea la SGA. Sus dimensiones se basan en el fichero de inicialización "init.ora".

```
SQL> connect sys as sysdba
```

```
SQL> startup nomount
```

Montar la base de datos

En esta parte del proceso de arranque se produce la conexión al/los archivo/s de control.

En este estado se puede:

- Cambiar el modo de archivado de la B.D.
- Renombrado de archivos de Redo Log o del asociado al tablespace SYSTEM
- Crear, modificar o suprimir nuevos Redo Log o grupos de Redo Log

Partiendo del anterior estado (nomount), montamos la base de datos de la siguiente forma:

```
SQL> alter database mount;
```

En caso de que queramos iniciar la base de datos en este estado bastaría con hacer lo siguiente:

```
SQL> connect sys as sysdba
```

```
SQL> startup mount
```

Abrir base de datos

En esta parte de proceso abren todos los ficheros asociados a los tablespaces y los ficheros de Redo Log.

La B.D. está accesible para todos los usuarios

Si es necesaria una recuperación (por un fallo de luz o CPU), se produce en este momento.

Partiendo del anterior estando (mount), abrimos la base de datos de la siguiente forma:

```
SQL> alter database open;
```

En caso de que queramos iniciar la base de datos en este estado bastaría con hacer lo siguiente:

```
SQL> connect sys as sysdba
```

```
SQL> startup
```

Más alternativas para el arranque de base de datos.

Arranque solo para usuarios con el privilegio RESTRICTED SESSION.

```
SQL> startup restrict
```

Arranque forzada

```
SQL> startup force
```

- Arranque con un fichero de parámetros distinto al habitual o localizado en una situación diferente a donde se encuentra por defecto.

```
SQL> startup pfile=/oracle/database/init2.ora
```

Parada base de datos

La parada de una B.D. Oracle se realiza mediante el comando SHUTDOWN desde SQL*DBA después de haber establecido una conexión como SYS AS SYSDBA.

Existen tres tipos de shutdown:

```
SQL> shutdown normal
```

```
SQL> shutdown immediate
```

```
SQL> shutdown abort
```

Shutdown normal

- Espera a que los usuarios conectados actualmente finalicen TODAS las operaciones.

Evita nuevas conexiones. Los usuarios que intentan conectarse reciben el mensaje "Shutdown in progress".

Cierra y desmonta la B.D. Cierra la SGA para los procesos background. No necesita recuperación al arrancar la base de datos.

```
SQL> connect sys as sysdba connected SQLPLUS> shutdown normal
```

Shutdown immediate

- Espera a que las transacciones actuales se completen.

Evita nuevas transacciones y nuevas conexiones. Los usuarios que intentan conectarse o los que ya están conectados al intentar realizar una nueva transacción reciben el mensaje "Shutdown in progress".

El proceso PMON finaliza las sesiones no activas y realiza ROLLBACK de aquellas transacciones que no estén validadas.

Cierra y desmonta la B.D. Cierra la SGA para los procesos background. No necesita recuperación al arrancar la base de datos.

```
SQL> connect sys as sysdba
```

```
SQL> shutdown immediate
```

Shutdown abort

Parada drástica, no espera a que los usuarios conectados actualmente finalicen sus transacciones. El usuario conectado recibe el mensaje "No logged on". No se realiza ROLLBACK de las transacciones pendientes.

El proceso PMON finaliza las sesiones no activas y realiza ROLLBACK de aquellas transacciones que no estén validadas.

SI necesita recuperación al arrancar la base de datos.

```
SQL> connect sys as sysdba
```

```
SQL> shutdown abort
```

Archivos Esenciales para iniciar la Base Datos

```
Init.ora
```

```
Spfileacl.ora
```

Cuando arrancamos una BDO, pasa por varios estados hasta que finalmente queda accesible a los usuarios: nomount, mount y open.

- En el primer estado (nomount) se arranca la instancia: lectura del fichero de parámetros, creación de la SGA, arranque de los procesos background y apertura del fichero alert\$ORACLE_SID.log.

Nota: el fichero de parámetros se busca en \$ORACLE_HOME/dbs, comenzando por spfile\$ORACLE_SID.ora. Si no lo encuentra, sigue con spfile.ora, y finalmente init\$ORACLE_SID.ora.

- Seguidamente la BD se monta (mount) abriendo el fichero de control y obteniendo de él los nombres de los ficheros que la componen: datafiles y redo log.
- Finalmente se abre la BD (open), procediendo a la apertura de los ficheros de datos (datafiles) y los ficheros redo log. El servidor

oracle comprueba la consistencia de la base de datos, y si es necesario el proceso SMON inicia la recuperación de la instancia.

COMANDO STARTUP

Arranca la instancia y abre la BD. Permite parar el proceso de arranque de la BD en cualquiera de sus fases (NOMOUNT, MOUNT).

STARTUP (abre la base de datos con el fichero de parámetros por defecto).

STARTUP PFILE: /opt/app/oracle/product/11.2.0/db_1/dbs/oraserv5init.ora

STARTUP NOMOUNT: para crear la base de datos.

STARTUP MOUNT: para renombrar datafiles, activar ARCHIVELOG o para hacer una recuperación completa de la BD.

STARTUP RESTRICT: sólo permite la conexión hacia la base de datos a usuarios con el privilegio RESTRICTED SESION.

STARTUP FORCE: hace SHUTDOWN ABORT y arranca la BD.

Viendo el proceso de arranque:

```
[oracle@oraserv5 ~]$ sqlplus sys/kikla as sysdba@oraserv5/oraserv5
```

```
SQL> startup nomount;
```

En esta parte del proceso de arranque se produce la conexión a los archivos de control.

En este estado se pueden realizar las siguientes tareas:

- Cambiar el modo de archivado de la B.D.
- Renombrar los archivos de Redo Log o del asociado al tablespace SYSTEM.
- Crear, modificar o suprimir nuevos Redo Log o grupos de Redo Log.

```
SQL> alter database mount;
```

```
SQL> alter database open database opened;
```

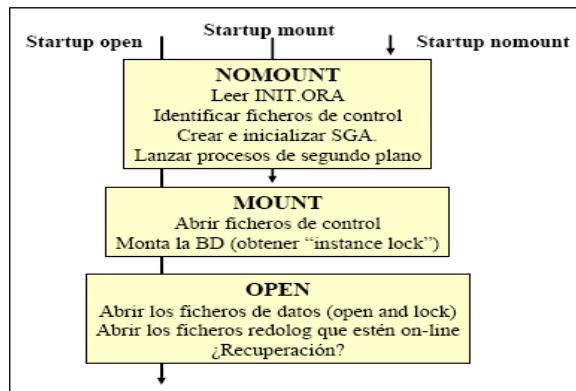
- En esta parte de proceso abren todos los ficheros asociados a los tablespaces y los ficheros de Redo Log. La B.D. está accesible para todos los usuarios

- En el primer estado (`nomount`) se arranca la instancia: lectura del fichero de parámetros, creación de la SGA, arranque de los procesos background y apertura del fichero `alert$ORACLE_SID.log`.

**Nota: el fichero de parámetros se busca en `$ORACLE_HOME/dbs`, comenzando por `spfile$ORACLE_SID.ora`. Si no lo encuentra, sigue con `spfile.ora`, y finalmente `init$ORACLE_SID.ora`.

Seguidamente la BD se monta (`mount`) abriendo el fichero de control y obteniendo de él los nombres de los ficheros que la componen: datafiles y redo log.

Finalmente se abre la BD (`open`), procediendo a la apertura de los ficheros de datos (datafiles) y los ficheros redo log. El servidor Oracle comprueba la consistencia de la base de datos, y si es necesario el proceso SMON inicia la recuperación de la instancia.



Parada de la Base Datos

Hay determinadas operaciones que requieren parar la BD; como la actualización de algunos parámetros del `init.ora`; o hacer una copia física de la BD (copia en frío). La BD se para con el comando `SHUTDOWN`, impidiendo cualquier conexión posterior.

`SHUTDOWN [NORMAL]`, espera a que terminen todas las transacciones en curso y todas las sesiones, fuerza un checkpoint, además de cerrar todos los ficheros y destruir (parar) la instancia.

`SHUTDOWN TRANSACTIONAL`, sólo espera a que terminen las transacciones en curso, fuerza un checkpoint, cierra los ficheros y destruye (para) la instancia.

`SHUTDOWN IMMEDIATE`, hace rollback de todas las transacciones en curso y cierra todas las sesiones; cierra y desmonta la BD, además de forzar un checkpoint, cerrar ficheros y parar la instancia (como los anteriores).

SHUTDOWN ABORT, cierra la instancia (destruye procesos background y SGA) sin esperar a desmontar ni cerrar la BD (como en una “caída”, ni hace checkpoint ni cierra ficheros). Requiere recovery de la instancia al arrancar (lo hace automáticamente el proceso SMON).

INSTANCIA

La integran los procesos “background” y la SGA . Abre una y sólo una BDO, y permite acceder a ella. Nota: con Oracle Real Application Cluster (RAC), más de una instancia usarán la misma BD.

En la máquina donde reside el servidor Oracle, la variable ORACLE_SID identifica a la instancia con la que estamos trabajando.

Para poder conectarnos a una BDO, necesitamos una serie de variables en el entorno del usuario del S.O. desde el que realizaremos la conexión. En el caso de unix/linux:

- ORACLE_HOME. Localización del sw Oracle a utilizar.
- ORACLE_SID. BD, asociada al ORACLE_HOME, a la que vamos a conectarnos. Tiene sentido sólo en un SBD.
- PATH=\$PATH:\$ORACLE_HOME/bin. Programas Oracle
- LD_LIBRARY_PATH=\$ORACLE_HOME/lib. Localización de las librerías compartidas (Linux/Unix).
- NLS_LANG=spanish_spain. Idioma del cliente (opcional).

Proceso de usuario: la ejecución de la aplicación que permite al usuario iniciar la conexión; por ejemplo, sql*plus.

Proceso servidor: se crea en el SBD cuando el usuario se conecta a la BD, y es el que realmente interactúa con la BD.

Una conexión de un proceso de usuario al SBD es una sesión en la BD (puede haber varias del mismo usuario). Se inicia cuando el usuario se valida contra la BD y termina al desconectarse.

Desconectar una sesión con “ALTER SYSTEM DISCONNECT SESSION 'sid, serial#' [POST_TRANSACTION] [IMMEDIATE];”. Destruye el servidor dedicado (o el circuito virtual si MTS).

Program Global Area (PGA) y PGA_AGGREGATE_TARGET (>=9i) .

Es una zona de memoria, fuera de la SGA, reservada para cada proceso servidor (y tb los background) del SBD.

Se crea para cada nuevo proceso servidor (o un proceso background); y se libera cuando el proceso termina.

En un entorno de servidores dedicados (dedicated server) contiene: sort area, información de sesión (privilegios de usuario y estadísticas de sesión), estado de los cursos (etapa del procesamiento de cada sentencia SQL que está usando actualmente la sesión), pila (stack space).

Con servidores compartidos (MTS), parte de estas estructuras se guardan en la SGA. Si se activa la Large Pool se almacenan en ella, si no se quedan en la Shared Pool.

PGA_AGGREGATE_TARGET (>=9i). Valor mínimo 10M, y default=20%SGA.. Vista V\$PGASTAT. Tamaño = en OLTP $\text{RAM} \times 0.80 \times 0.20$ (en DSS $\text{RAM} \times 0.80 \times 0.50$). Activar: si es distinto de cero (junto a WORKAREA_SIZE_POLICY=AUTO). Habilita el uso de una zona de memoria compartida para las PGA, evitando la necesidad de asignar parámetros como SORT_AREA_SIZE o HASH_AREA_SIZE.

Procesos Background

Objetivos: Aprender cuales son los procesos que son ejecutados en fondo por el sistema de base de datos Oracle, cuando este pone en ejecución una instancia de datos, y hacer uso de las herramientas del sistema operativo para la optimizar de los recursos que estos procesos puedan requerir en un momento dado.

Estructura de procesos

Procesos background: disponibles cuando se arranca una instancia Oracle. Son los siguientes:

- DBWR, PMON, SMON, LGWR y CKPT. opcionalmente podemos tener: ARCH, RECO, Dispatchers (Dnnn), Shared Servers (Snnn), etc.
- En 10g aparecen: PSP0 (Process Spawner), MMAN (Memory Manager), MMON (Memory Monitor), MMNL (Memory Monitor Light).
- En 11g tenemos: VKTM (Virtual Keeper of TiMe), FBDA (Flashback Data Archiver), DIAG (escribe el nuevo Automatic Diagnostic Repository), DIA0, SMC0/Wnnn (Space Management Coordinator), DBRM (Data Base Resource Manager)
- En 11g el parámetro DIAGNOSTIC_DEST (BACKGROUND_DUMP_DEST en 10g), directorio donde se guardan los ficheros de traza de los procesos background.

Procesos background (DBWR)

DBWR. Escribe los bloques de datos (y rollback) de la SGA (data buffer cache) en los ficheros de datos. Esto lo hace de forma asíncrona, cuando:

- Sucede un checkpoint.
- El número de buffers modificados alcanza un umbral.
- No quedan buffers libres.
- Ocurre un timeout.
- Ponemos un tablespace offline.
- Dejamos un tablespace en modo readonly.
- Borramos o “truncamos” una tabla.
- ALTER TABLESPACE nombrfsp BEGIN BACKUP.

Nota. Un checkpoint sucede cuando:

- El fichero redo log se llena al 90%.
- Se alcanza log_checkpoint_interval (bloques del SO).
- Se llega a log_checkpoint_timeout (en segundos).

- ✓ Nombre del proceso: DBW0 a DBW9 y DBWa DBWj (máximo 20).
- ✓ DB_WRITER_PROCESSES. Número de procesos arrancados (valor por defecto = 1 o CPU_COUNT/8, el mayor de ambos).

Procesos background (LGWR, SMON)

LGWR. Realiza escrituras secuenciales del contenido de la redo log buffer cache en los ficheros redo log. ¿Cuando?

- Se hace commit.
- La redo log buffer cache se llena 1/3.
- Hay 1Mb de cambios en la redo log buffer cache.
- Como mucho, cada 3 segundos.
- Siempre antes que escriba el DBWR.

SMON. Recupera la instancia, si es necesario, cuando ésta arranca: aplica los cambios registrados en los redo log (roll forward), abre la base de datos dejándola accesible a los usuarios, y hace rollback de las transacciones que no terminaron.

También se activa periódicamente, agrupando extensiones libres contiguas en extensiones de mayor tamaño (sólo para tablespaces con "default storage" cuyo pctincrease > 0).

Además libera el espacio ocupado por segmentos temporales durante el procesamiento de sentencias SQL.

Procesos background (PMON, CKPT, ARCH y nuevos 10g)

PMON. Se activa periódicamente, recuperando los recursos después de que un proceso falle: hace rollback de las transacciones que el usuario tenía en curso, libera bloqueos a nivel de tabla/fila y otros recursos reservados por el usuario, y vuelve a arrancar dispatchers "muertos" (dead dispatchers).

CKPT. Avisa al DBWR cuando sucede un checkpoint y actualiza las cabeceras de los ficheros de datos y de control (el DBWR volcará los buffers actualizados a los ficheros de datos). Si los checkpoints suceden muy frecuentemente puede haber contención en disco. Si tardan mucho se alargará el proceso de recovery. Como mucho sucederá un checkpoint al llenarse el redolog.

ARCH. Proceso opcional. Archiva automáticamente los redo log online si se activa el modo ARCHIVELOG; asegurando que se registran todos los cambios hechos en la base de datos. Archiva el redo log que se ha llenado, cuando sucede un "log switch".

MMAN (>=10g). Memory Manager. Gestiona Automatic Shared Memory Management.

MMON (>=10g). Memory monitor. Genera snapshots del AWR (abre procesos esclavos M000).

MMNL (>=10g). Memory Monitor Light. Captura frecuentemente “session history” (V\$ACTIVE_SESSION_HISTORY) y calcula métricas.

PSP0 (>=10G). Process spawner. Crea y gestiona otros procesos Oracle.

Procesos background nuevos 11g

VKTM (Virtual Keeper of TiMe): es el responsable de hacer el seguimiento del tiempo y se usa cada vez q hay q contar el tiempo.

- FBDA (Flashback Data Archiver): va archivando los datos históricos de las tablas que hay en el Flashback, almacenando una imagen previa de las filas y tb los metadatos de esas filas.
- DIAG y DIA0 (Diagnosability y Diagnosability Process 0, no hay más): se corresponden al nuevo ADR (Automatic Diagnostic Repository), el primero genera los volcados (logs, trazas, coreumps, etc), y el segundo gestiona la detección de bloqueos y la resolución de deadlocks.
- SMC0/Wnnn (Space Management Coordinator): el primero se encarga de la coordinación y ejecución de las tareas de gestión del espacio, para lo que necesitará procesos Wnnn.
- DBRM (Data Base Resource Manager): para configurar la gestión de recursos y su planificación .

MEMORIA: SGA Y PGA

Oracle recomienda asignar un máximo del 80% de la RAM de la máquina para el Servidor Oracle.

- A su vez, de dicho 80%, recomienda asignar un máximo del 80% para la SGA (quedaría como mínimo un 20% para PGAs).
- Con Oracle 11g se configura la memoria del Servidor (SGA+PGA) con un sólo parámetro llamado MEMORY_TARGET.

Automatic Memory Management (AMM) >= 11g

En Linux, memory_target y memory_max_size están limitados por el valor de /dev/shm, para cada una de las instancias que haya en el servidor: si /dev/shm vale 2G, entonces ninguno de ellos pueden valer más de 2Gb para cada instancia del servidor.

- Tb hay q configurar el Buffer de Redo con log_buffer (igual q en 10g y versiones anteriores).
- V\$MEMORY_DYNAMIC_COMPONENTS: tamaños actuales.
- V\$MEMORY_TARGET_ADVICE: recomendaciones de MEMORY_TARGET.
- V\$MEMORY_RESIZE_OPS: operaciones de AMM.

Comandos para manejar procesos GNU/Linux

Comando ps:

Es el que permite informar sobre el estado de los procesos. ps esta basado en el sistema de archivos /proc, es decir, lee directamente la información de los archivos que se encuentran en este directorio. Tiene una gran cantidad de opciones, incluso estas opciones varían dependiendo del estilo en que se use el comando. Entre las opciones del comando ps las mas comunes son.

Comando kill:

Literalmente quiere decir matar, sirve no solo para matar o terminar procesos sino principalmente para enviar señales (signals) a los procesos. La señal por default (cuando no se indica ninguna es terminar o matar el proceso), y la sintaxis es kill PID, siendo PID el número de ID del proceso.

Comando killall:

Funciona de manera similar a kill, pero con la diferencia de en vez de indicar un PID se indica el nombre del programa, lo que afectará a todos los procesos que tengan ese nombre.

Comando jobs:

Si por ejemplo, se tiene acceso a una única consola o terminal, y se tienen que ejecutar varios comandos que se ejecutarán por largo tiempo, se pueden entonces mandarlos a segundo plano o background con el objeto de liberar la terminal y continuar trabajando, esto lo hacemos poniendo el símbolo & justo después del comando a ejecutar, esta acción generara un jobs, es decir un proceso que esta corriendo en segundo plano. Cada job tiene un numero asociado, asignado por el shell, que luego servirá para subir dicho comando a primer plano posteriormente. Para manejar los jobs podemos usar los comandos fg(foreground) y bg(background) para mandar a segundo plano un proceso (bg) y subirlo a primer plano.

Fichero de Control

Objetivos: En este capítulo tratamos tópicos mas profundos sobre la estructura interna de la base de datos de Oracle, específicamente mas sobre los ficheros de control, los cuales son los que regulan el inicio de la instancia de la base de datos.

Contenido del fichero de control

Se trata de un fichero binario, sin el cual no es posible arrancar la BD. Por ello es conveniente mantener varias copias del mismo, en diferentes discos.

- Se lee al montar la BD.
- Su tamaño es fijo, y depende de los parámetros indicados al crear la BD con como por ejemplo MAXLOGFILES y MAXDATAFILES.
- El fichero de control contiene información como: nombre de la BD, fecha de creación de la BD, nombres de los tablespaces, nombre y localización de los ficheros de datos y de redo, número de secuencia del redo log en curso, información de checkpoint, información del archivado de los redo log, información de backup.

Multiplexar fichero de control

Para añadir una copia del fichero de control, se deben realizar los pasos siguientes:

- Se para la BD con SHUTDOWN NORMAL.
- Se hace una copia física del fichero de control, a nivel del sistema operativo. En Unix con el comando cp.
- Se incluye la nueva copia del fichero de control en el init.ora (o spfile); en el parámetro CONTROL_FILES.
- Arrancar la BD con STARTUP.

Se recomienda sacar una copia de seguridad del fichero de control cada vez que cambie la estructura física de la BD:

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

De esta forma se generan, en un fichero de traza, las sentencias sql necesarias para volver a crear el fichero de control:

```
SQL> ALTER DATABASE BACKUP CONTROLFILE  
TO '/opt/app/oracle/oradata/oraserv5/ora_control01.bak';
```

Esto realiza una copia binaria y aislada del fichero.

En la vista V\$CONTROLFILE tenemos la lista de todos los ficheros de control de la BD. En la vista V\$CONTROLFILE_RECORD_SECTION veremos las diferentes secciones y su estado de uso.

Backup del fichero de control

```
STARTUP NOMOUNT  
CREATE CONTROLFILE REUSE DATABASE "ACL" NORESETLOGS  
NOARCHIVELOG  
MAXLOGFILES 32  
MAXLOGMEMBERS 2  
MAXDATAFILES 1000  
MAXINSTANCES 1  
MAXLOGHISTORY 292  
LOGFILE  
GROUP 1 '/opt/app/oracle/oradata/oraserv5/redo01.log' SIZE 10M,  
GROUP 2 '/opt/app/oracle/oradata/oraserv5/redo02.log' SIZE 10M,  
GROUP 3 '/opt/app/oracle/oradata/oraserv5/redo03.log' SIZE 10M  
DATAFILE  
'/opt/app/oracle/oradata/oraserv5/system01.dbf',  
'/opt/app/oracle/oradata/oraserv5/undo_rbs01.dbf',  
'/opt/app/oracle/oradata/oraserv5/sysaux01.dbf',  
'/opt/app/oracle/oradata/oraserv5/users01.dbf'  
CHARACTER SET WE8ISO8859P15;  
ALTER DATABASE OPEN;
```

Luego sacamos una copia del fichero de control:

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

```
SQL> !ls lt $ORACLE_BASE/admin/$ORACLE_SID/udump|head 2
```

```
SQL> !more $ORACLE_BASE/admin/$ORACLE_SID/udump/oraserv5_ora_26373.trc
```

```
SQL> alter database backup controlfile to '/opt/app/oracle/oradata/control_copia1.ctl';
```

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

Los Datafiles

Objetivos: En este capítulo trataremos sobre todos el concepto los datafiles y resolver problema.

Un datafile es la representación física de un tablespace.

Son los "ficheros de datos" donde se almacena la información físicamente.

Un datafile puede tener cualquier nombre y extensión (siempre dentro de las limitaciones del sistema operativo), y puede estar localizado en cualquier directorio del disco duro, aunque su localización típica suele ser \$ORACLE_HOME/Database.

Un datafile tiene un tamaño predefinido en su creación (por ejemplo 100Mb) y este puede ser alterado en cualquier momento.

Cuando creamos un datafile, este ocupará tanto espacio en disco como hayamos indicado en su creación, aunque internamente esté vacío. Oracle hace esto para reservar espacio continuo en disco y evitar así la fragmentación. Conforme se vayan creando objetos en ese tablespace, se irá ocupando el espacio que creó inicialmente.

Un datafile está asociado a un solo tablespace y, a su vez, un tablespace está asociado a uno o varios datafiles. Es decir, la relación lógica entre tablespaces y datafiles es de 1-N, maestro-detalle.

Los datafiles tienen una propiedad llamada AUTOEXTEND, que se si está activa, se encarga de que el datafile crezca automáticamente (según un tamaño indicado) cada vez que se necesite espacio y no exista. Al igual que los tablespaces, los datafiles también puede estar en línea o fuera de ella.

La tarea de añadir un nuevo datafile a un tablespace existente es una labor habitual de la administración de las bases de datos en Oracle. ¿Cuando es necesario añadir un datafile? obviamente cuando el o los datafiles que componen el tablespace en cuestión no se pueden extender más, bien porque hemos llegado a la máxima capacidad del disco donde este ubicado o por que hemos llegado el máximo permitido por Oracle.

Lo primero que haremos es ver el estado de nuestro tablespace, por ejemplo tenemos el tablespace DATOS de la base de datos PREUBA01 que esta por encima del 90% de ocupación y necesitamos ampliarlo, desde sqlplus tecleamos los siguiente:

```
SQL> SELECT FILE_NAME, BYTES  
      FROM DBA_DATA_FILES  
     WHERE TABLESPACE_NAME = 'DATOS';
```

Para ver todos los datafiles de una base de datos:

```
SQL> select file_name, file_id, relative_fno  
from dba_data_files;
```

Crear y Modificar Datafiles

Cuando creamos un tablespaces normal o temporal debemos especificar aunque sea un datafile, se puede crear mas de 1 datafile:

```
SQL> ALTER DATABASE DATAFILE '<data_file_name | data_file_number>'  
RESIZE <n> K|M|G|T|P|E;
```

Mover un datafile para una nueva ruta.

```
SQL> ALTER DATABASE RENAME FILE  
'<old_full_path>' TO '<new_full_path>';
```

Para modificar un datafile y añadirle extensión automática se utiliza la siguiente instrucción:

```
SQL> ALTER DATABASE DATAFILE '<data_file_name' | data_file_number>  
AUTOEXTEND <OFF | ON [NEXT SIZE <n>K|M|G|T|P|E  
MAXSIZE <UNLIMITED | <n>K|M|G|T|P|E>;
```

Datafiles Online y Offline.

```
SQL> ALTER DATABASE DATAFILE '<data_file_name' | data_file_number>  
<ONLINE | OFFLINE [FOR DROP]>;
```

Agregar un Datafile a un Tablespaces.

```
SQL> ALTER TABLESPACE <tablespace_name>  
ADD DATAFILE '<path_and_file_name>' SIZE <n>K|M|G|T|P|E;
```

Eliminar un Datafile.

```
SQL> ALTER DATABASE DATAFILE '<file_name_or_file_number>'  
[offline] DROP;
```

TABLESPACES

Objetivos: En este capítulo trataremos sobre todos el concepto lógico de los diferentes tablespace y su importancia.

Tablespaces y Archivos de Datos

Las bases de datos, tablespaces y archivos de datos están estrechamente relacionados pero presentan importantes diferencias:

Una base de datos Oracle consta de una o más unidades lógicas de almacenamiento denominadas tablespaces, que en conjunto almacenan todos los datos de la base de datos.

Cada tablespace de una base de datos Oracle se compone de uno o más archivos denominados archivos de datos, que son estructuras físicas que se ajustan al sistema operativo en el que se está ejecutando el software de Oracle.

Los datos de una base de datos se almacenan en conjunto en los archivos de datos que constituyen cada tablespace de la base de datos.

Una única base de datos puede tener hasta 65.534 archivos de datos.

Desde Oracle 10g, una BD podría tener sólo los tablespaces SYSTEM y SYSAUX (>=10g). Oracle recomienda crear tablespaces adicionales para los datos, índices, rollback y segmentos temporales.

Según la estructura física, una BD está compuesta por: el fichero de control, ficheros redolog y ficheros de datos. Y la estructura lógica la componen tablespaces, segmentos, extensiones y bloques.

Cada tablespace consiste en uno o más ficheros del s.o. Llamados ficheros de datos (un fichero pertenece a un solo tablespace):

- Un tablespace sólo puede pertenecer a una BD a la vez.
- Puede tener cero o más segmentos (un segmento sólo pertenece a un tablespace).
- Exceptuando el SYSTEM, o aquellos que contengan segmentos de rollback activos, un tablespace se puede poner offline, con la BD funcionando.
- Un tablespace se puede poner en modo read-only o read-write. Tipos de tablespaces: permanent (datos: system, sysaux, aplicaciones), undo (rollback) y temporary (sort).
- Oracle >=10g permite crear “bigfile tablespaces”, de hasta 8EB (millones de terabytes).

TABLESPACES SYSTEM Y SYSAUX

Los tablespaces SYSTEM y SYSAUX son los únicos que, como mínimo, se crean con la BD (create database).

El tablespace SYSTEM (No debe contener datos de aplicaciones):

- Contiene el DD, incluidos procedimientos almacenados, funciones, triggers y paquetes.
- También alberga al segmento de rollback system .

El tablespace SYSAUX (>=10g) permite que en el tablespace SYSTEM sólo esté el DD, aglutinando las utilidades del sistema (Repositorio OEM, Intermedia, Spatial, OLAP, RMAN, XML DB, etc).

- ¿Qué hay en el tablespace SYSAUX y cuánto ocupa?

```
SQL> select occupant_name, space_usage_kbytes from v$sysaux_occupants;
```

- ¿Se puede mover el contenido de SYSAUX a otro tablespace?

```
SQL> select occupant_name, move_procedure, move_procedure_desc
  from v$sysaux_occupants;
```

Respecto al resto de tablespaces (no SYSTEM), se recomienda separar los ficheros de redo, luego datos de índices, después rollback y segmentos temporales. También es bueno separar datos estáticos y dinámicos.
Sentencia básica para crear un TableSpace:

```
SQL> CREATE [<BIGFILE | SMALLFILE>] TABLESPACE <tablespace_name>
  DATAFILE '<path_and_file_name>'
  SIZE <integer><K | M | G | T | P | E> [REUSE] AUTOEXTEND <OFF | ON>
  BLOCKSIZE <bytes>
  [<LOGGING | NOLOGGING | FILESYSTEM_LIKE_LOGGING>]
  [FORCE LOGGING]
  [ENCRYPTION USING '<encryption_algorithm>']
  IDENTIFIED BY <password> [NO] SALT
  [DEFAULT <COMPRESS [FOR <ALL | DIRECT_LOAD> OPERATIONS] | NOCOMPRESS>]
  [<ONLINE | OFFLINE>]
  EXTENT MANAGEMENT LOCAL <AUTOALLOCATE | UNIFORM SIZE <extent_size>>
  [SEGMENT SPACE MANAGEMENT <AUTO | MANUAL>]
  [FLASHBACK <ON | OFF>];
```

Creando Tablespace temporal.

```
SQL> CREATE TEMPORARY TABLESPACE <tablespace_name>
  TEMPFILE '<path_and_file_name>'
  SIZE <integer><K | M | G | T>
  AUTOEXTEND <ON | OFF>
  TABLESPACE GROUP <group_name>
  EXTENT MANAGEMENT LOCAL UNIFORM SIZE <extent_size>;
```

Tablespace Por defecto.

```
SQL> ALTER DATABASE DEFAULT TABLESPACE <tablespace_name>;
```

Agregando Datafile.

```
SQL> ALTER TABLESPACE ADD DATAFILE <file_name>,
<integer><K | M | G | T>
[REUSE]
<autoextend> <ON | OFF>
NEXT <integer><K | M | G | T>
MAXSIZE <integer><K | M | G | T | UNLIMITED>;
```

La sintaxis para renombrar un tablespace es la siguiente:

```
SQL> ALTER TABLESPACE <tablespace_name> RENAME TO
<new_tablespace_name>;
```

La sintaxis para poner Online/Offline a un tablespace es la siguiente:

```
SQL> ALTER TABLESPACE <tablespace_name> OFFLINE;
```

```
SQL> ALTER TABLESPACE <tablespace_name> ONLINE;
```

Poner a un tablespace en modo read only, causará que el tablespace no sea afectado por sentencias DML. Su sintaxis es la siguiente:

```
SQL> ALTER TABLESPACE nombreTablespace
[ READ ONLY | READ WRITE ]
```

Un tablespace puede ser cambiado de tamaño ya sea que se le agreguen datafiles o se le ajusten los datafiles que ya le pertenecen. Su sintaxis es la siguiente:

```
SQL> ALTER TABLESPACE DATAFILE nombreDatafile
RESIZE n[ M | G | T ]
```

Para dar de baja un tablespace, es necesario utilizar la instrucción DROP TABLESPACE. Su sintaxis es la siguiente:

```
SQL> DROP TABLESPACE nombreTablespace [ INCLUDING CONTENTS [ AND DATAFILES ] ]
```

FORMAS DE GESTIONAR EL ESPACIO DE UN TABLESPACE

Tablespaces manejados localmente (Oracle los recomienda >=8i):

Las extensiones libres se registran en un bitmap, de forma que cada bit corresponde a un bloque. El valor de cada bit indica si el bloque correspondiente está libre o usado. Existe un bitmap de este tipo en cada fichero del tablespace. Cada vez que una extensión se reserva o se libera, se modifica el bitmap correspondiente.

Ventajas:

- Reducción de la contención en las tablas del DD.
- No se genera rollback al reservar/liberar espacio (pues no se actualiza el DD).
- No es necesario hacer “coalesce”.

No tienen el mismo sentido INITIAL_EXTENT y NEXT_EXTENT y no se usan MIN_EXTENTS, MAXEXTENTS y PCTINCREASE del STORAGE al crear una tabla. Tampoco tiene sentido DEFAULT STORAGE del tablespace.

El tablespace system se puede “manejar localmente”, desde Oracle9i (en 8i no). Si el SYSTEM es “local”, NO se pueden crear tablespaces “por diccionario”. Desde Oracle9i, por defecto, los tablespaces se crean como “locales” (si el parámetro compatible>=9.0), excepto el SYSTEM.

Tablespaces gestionados a través del DD (a extinguir):

Es el método por defecto en Oracle8i. Las extensiones libres quedan registradas en tablas del DD. Cada vez que una extensión se libera o se reserva, las tablas correspondientes del DD deben ser actualizadas.

Permite definir STORAGE flexible a los segmentos (los “locales” NO).

TEMP: El tablespace temporal se utiliza cuando se ejecuta una sentencia SQL que necesita la creación de segmentos temporales (como grandes ordenaciones o la creación de un índice). De la misma forma que a cada usuario se le asigna un tablespace por defecto para almacenar objetos de datos creados, también se le asigna un tablespace temporal. La práctica recomendada es definir un tablespace temporal por defecto para la base de datos que se asignará a cada usuario recién creado, a menos que se especifique lo contrario. En la base de datos pre-configurada, el tablespace TEMP se especifica como tablespace temporal por defecto. Esto significa que si no se especifica ningún tablespace temporal al crear la cuenta de usuario, la base de datos Oracle asigna este tablespace al usuario.

UNDOTBS1: Éste es el tablespace de deshacer que utiliza el servidor de bases de datos para almacenar información de deshacer. Si una base de datos utiliza Gestión Automática de Deshacer (AUM), debe tener exactamente un tablespace de deshacer activo en cualquier momento. Este tablespace se genera durante la creación de la base de datos.

USERS: Este tablespace se utiliza para almacenar objetos y datos de usuarios permanentes. En la base de datos pre-configurada, el tablespace USERS es el tablespace por defecto para todos los objetos creados por usuarios que no son del sistema. Para los usuarios SYS y SYSTEM (usuarios del sistema), el tablespace permanente por defecto sigue siendo SYSTEM.

EXAMPLE: Este tablespace contiene los esquemas de ejemplo que se pueden instalar al crear la base de datos. Los esquemas de ejemplo proporcionan una plataforma común para los ejemplos. La documentación y los cursos de Oracle contienen ejemplos basados en los esquemas de ejemplo.

Nota: Para simplificar la administración es normal tener un tablespace sólo para índices.

Borrado de Tablespaces

Puede borrar un tablespace y su contenido (los segmentos incluidos en el tablespace) de la base de datos si el tablespace y su contenido ya no son necesarios. Debe tener el privilegio del sistema DROP TABLESPACE para borrar un tablespace.

Cuando borra un tablespace, se eliminan los punteros al archivo del archivo de control de la base de datos asociada. opcionalmente puede indicar al servidor de Oracle que suprima los archivos del sistema operativo (archivos de datos) que constituyen el tablespace borrado. Si no indica al servidor de Oracle que suprima los archivos de datos al mismo tiempo que el tablespace, deberá utilizar posteriormente los comandos adecuados del sistema operativo si desea suprimirlos.

No puede borrar un tablespace que contenga segmentos activos. Por ejemplo, si se está utilizando actualmente una tabla del tablespace o el tablespace contiene datos de deshacer necesarios para realizar un rollback de transacciones sin confirmar, no puede borrar el tablespace. El tablespace puede estar online u offline, pero es mejor que se ponga offline antes de borrarlo.

Visualización de Información de Tablespaces

Haga clic en View para ver información sobre el tablespace seleccionado. En la página View Tablespace, también puede hacer clic en Edit para modificar el tablespace.

También se puede obtener información de tablespaces y archivos de datos consultando lo siguiente:

- **Información de tablespaces:**

- DBA_TABLESPACES
- V\$TABLESPACE

- **Información de archivos de datos:**

- DBA_DATA_FILES
- V\$DATAFILE
- V\$_BACKUP_DATAFILE
- V\$_DATAFILE
- V\$_DATAFILE_COPY
- V\$_DATAFILE_HEADER

- **Información de archivos temporales:**

- DBA_TEMP_FILES
- V\$TEMPFILE
- USER_TABLESPACES

Uso de espacio de tablespace (advertencia 85%, crítico 97%)

Redo Log

Objetivos: Aprender el concepto de los redo log y su gran importancia para un DBA.

El Online Redo log, es una estructura física que consiste de mínimo de dos archivos, estos a su vez pueden estar multiplexados en dos o mas copias idénticas, que a estos se le conocen como miembros de un grupo de Redo log. Como mencionamos, el Online Redo Log consiste de mínimo dos archivos, esto permite que Oracle escriba en un archivo de Online Redo Log mientras el otro se archiva (Cuando mencionamos archivar, es si la base de datos se encuentra en modo ARCHIVELOG).

En los Online Redo logs se almacenan registros de Redo, los cuales están conformados por vectores de cambio (change vectors), cada uno de estos vectores describe los cambios a un bloque de datos.

Todos los registros de tipo redo tiene metadata relevante al cambio, incluyendo:

- SCN y la estampa de tiempo del cambio
- El ID de la transacción que ha generado el cambio
- SCN y la estampa de tiempo cuando la transacción fue cometida (si es que fue cometida)
- Tipo de operación que efectuó el cambio
- Nombre y tipo del segmento de dato modificado

Los Online Redo Log son usados únicamente en el proceso de la recuperación de la base de datos.

Básicamente, lo que hay que entender como principio, es que cuando algún DML (insert, update o delete) o un DDL (alter, create, drop) sucede en nuestra base de datos, Oracle registra los cambios en memoria, en un buffer llamado Redo Log Buffer, que con este buffer hay un proceso asociado llamado LGWR.

El proceso LGWR de lo que se encarga es de escribir de la estructura de memoria (Redo) Log Buffer a los Online Redo Logs, y muy importante es saber cuáles son las circunstancias que hacen que el LGWR escriba al Online Redo Log:

- Cuando un usuario hace un commit a la transacción.
- Cuando sucede un cambio (log switch) de archivo de Redo Log.
- Cuando han pasado tres segundos desde la última escritura del LGWR hacia el Online Redo Log.

- Cuando el Redo Log Buffer esta 1/3 lleno o contiene mas de 1Mb de datos en el buffer.
- Cuando el proceso DBWn necesita escribir datos del Database Buffer Cache hacia disco.

El proceso LGWR escribe a los archivos de Online Redo Log de manera circular, cuando el LGWR escribe en el ultimo archivo de Online Redo Log disponible, el LGWR se regresa a escribir al primer archivo de Online Redo Log.

Ahora que ya vimos que es , y como mencionábamos arriba, los Online Redo Logs, son únicamente usados en el proceso de recuperación de la base de datos.

En el proceso de recuperación, se presenta tanto lo que es aplicar cambios cometidos no reflejados en los datafiles, a esto se le conoce como Roll Forward, y remover los cambios aplicados no cometidos de los datafiles, a esto se le conoce como Roll Back.

Suena un poco confuso, pero realmente no lo es, lo único que hay que saber es que cuando se realiza un commit, Oracle añade un Marcador de Commit en el redo log buffer, así es como Oracle sabe que datos son cometidos y cuáles no.

Aquí un pequeño algoritmo de como es el proceso de recovery, este lo tome del blog de Arup Nanda, no me lo acredito, solamente lo estoy traduciendo:

Leer las entradas de tipo Redo Log, empezando con el mas antiguo

Verificar el numero SCN del Cambio

Buscar el Marcador de Commit.

Si el marcador es encontrado, entonces los datos han sido cometidos.

Si es encontrado, entonces buscar los cambios en los datafiles (vía el numero SCN)

 ¿Cambios están reflejados en los datafiles?

 Si si, entonces brinca

 Si no, aplicar los cambios a los datafiles (Roll Forward)

Si no es encontrado, entonces los datos están sin cometer, buscar los cambios en los datafiles

 ¿Cambios están reflejados en los datafiles?

 Si no, entonces brinca

 Si si, entonces hacer un update a los datafiles con los datos antes del cambio (Roll Back)

Para ver la información que tiene los redo logs, puedes hacer una sesión de logminer, que eso lo veremos en otra entrada, pero por el momento te enseño un ejemplo de la información que puedes ver.

En una sesión con el el usuario HR, voy a crear una tabla llamada BLAH, y voy ver la información de la transacción, una vez que veo esta información voy a darle commit para finalizar la transacción.

```
SQL> create table fcldlog( name varchar2(100), num number);
SQL> insert into "HR"."fcldlog"("NAME","NUM") values
('Texto Nada Mas Probar Que Inserto','60671');
```

```
SQL> select dbms_transaction.local_transaction_id from dual;
LOCAL_TRANSACTION_ID
-----
6.14.1208
```

```
SQL> commit;
```

Los ficheros redo log se usan de manera circular: cuando uno se llena, el LGWR comienza a escribir en el siguiente grupo (“log switch”), hasta volver al primero. Cuando ocurre un “log switch”, también sucede un “checkpoint”; y se actualiza el fichero de control. Podemos forzar un log switch o un checkpoint explícitamente con:

- ALTER SYSTEM SWITCH LOGFILE;
- ALTER SYSTEM CHECKPOINT;

El LGWR escribe al hacer commit, o cada 3 segundos, o si el buffer redolog se llena 1/3, y antes de que el DBWR vuelque los cambios de los buffers de datos a los ficheros de la BD.

Archivos redo log en línea: Permiten la recuperación de instancias de la base de datos. Si la base de datos falla y no pierde archivos de datos, la instancia puede recuperar la base de datos con la información de dichos archivos.

Procesos en Segundo Plano y Recuperación: Archivos Redo Log y Escritor de Log.

Los archivos redo log registran cambios en la base de datos como resultado de las transacciones y las acciones internas del servidor de Oracle. (Una transacción es una unidad de trabajo lógica, que consta de una o más sentencias SQL que ejecuta un usuario.) Los archivos redo log protegen la base de datos de pérdidas de integridad debidas a fallos del sistema provocados por interrupciones de alimentación, fallos de disco, etc. Los archivos redo log se deben multiplexar para garantizar que la información almacenada en ellos no se pierda en caso de un fallo de disco.

El redo log se compone de grupos de archivos redo log. Un grupo está formado por un archivo redo log y sus copias multiplexadas. Cada copia idéntica se considera miembro de dicho grupo y cada grupo se identifica mediante un número. El proceso de escritor de log (LGWR) escribe registros de redo desde el buffer de redo log a todos los miembros de un grupo de redo logs hasta que se llena el archivo o se solicita una operación de cambio de log. A continuación, se cambia y se escribe en los archivos del siguiente grupo. Los grupos de redo logs se utilizan de forma circular.

Práctica recomendada: Si es posible, los archivos redo log multiplexados deben residir en discos distintos.

En el entorno de Oracle RDBMS, los registros de rehacer comprenden archivos en un formato propietario que registra un historial de todos los cambios realizados en la base de datos. Cuando algo se cambia en un fichero de datos, Oracle realiza el registro de rehacer.

¿Qué hacen los redo log files?

Los Ficheros de redo log registran cambios a la base de datos como resultado de transacciones o acciones internas del servidor Oracle.

¿Para qué sirven los redo log file?

Protegen la base de datos de la pérdida de integridad en casos de fallos causados por suministro eléctrico, errores en discos duros.

¿Cómo funcionan los redo log files?

Trabajan de manera cíclica. Si un archivo redo log online se llena LGWR pasará al siguiente grupo de log en el cual se produce una operación de punto de control (check point), la información es almacenada en el archivo de control (control file).

Multiplexación en línea Archivos Redo Log

Es recomendable que los archivos de redo log sean multiplexados para asegurar que la información almacenada en ellos no se pierda en caso de un fallo en disco. cada copia idéntica es miembro de un grupo, y cada grupo es identificado por un número.

Grupos de Archivos redo log

Se llama grupo de archivos a un juego de copias idénticas de archivos redo log online. Son necesarios al menos 2 grupos para que una base de datos funcione correctamente. Los primero conjuntos de miembros y grupos de archivos redo log se crean junto a la creación de la base de datos. ¿Cómo obtenemos la información de los grupos de redo log de nuestra BD?

Consultando estas dos vistas, tenemos los datos que nos pueden interesar de los redo log files, ubicación, estado, tamaño.

```
SQL> SELECT *FROM V$LOGFILE;
```

```
SQL> SELECT *FROM V$LOG;
```

Para agregar un redo log aun grupo.

```
SQL> ALTER DATABASE ADD LOGFILE
  ('<log_member_path_and_name>',
   '<log_member_path_and_name>')
  SIZE <integer> <K|M>;
```

Para Eliminar un miembro de un grupo de redo log.

```
SQL> ALTER DATABASE ADD LOGFILE MEMBER '<log_member_path_and_name>'
  TO GROUP <group_number>;
```

MODOS DE OPERACION DE LA BD

Objetivos: En este capítulo trataremos sobre todos los diferentes modo en que puede operar la BD y usted debe elegir el mejor para su empresa.

Modo NOARCHIVELOG.

El archivado de los “redo log” está deshabilitado. Cuando un grupo de “redo” pasa a estar inactivo tras un “log switch”, está disponible para ser nuevamente usado por el LGWR.

Este modo protege a la bd de fallos en la instancia pero no de fallos en los soportes (“media failure”). Sólo los cambios recientes en la bd, almacenados en el “redo” en línea pueden recuperarse. Si ocurre un fallo en disco, la bd sólo puede recuperarse hasta el momento en que se realizó la copia más reciente.

Para la recuperación sólo pueden emplearse copias completas y coherentes realizadas con la bd cerrada consistentemente.

Pérdida de un Archivo de Datos en Modo NOARCHIVELOG

La pérdida de cualquier archivo de datos de una base de datos en modo NOARCHIVELOG necesita una restauración completa de la base de datos, incluidos los archivos de control y todos los archivos de datos.

Con la base de datos en modo NOARCHIVELOG, la recuperación sólo es posible hasta el momento en que se realizó la última copia de seguridad. Por este motivo, los usuarios tienen que volver a introducir todos los cambios realizados desde esa copia de seguridad.

Modo ARCHIVELOG.

El archivado de los “redo log” está habilitado. Un grupo de “redo” no puede reutilizarse por LGWR hasta ser archivado tras el “log switch”.

Se garantiza que todas las transacciones validadas pueden recuperarse en caso de fallo en el sistema o disco. Además pueden emplearse copias realizadas con la bd abierta y en uso normal.

Nota: durante este tema se asume que la base de datos está en modo noarchivelog.

Para poner la base de datos en modo ARCHIVELOG, realice los siguientes pasos:

1. Active la casilla de control del modo ARCHIVELOG.
2. Haga clic en Apply. La base de datos sólo se puede definir en el modo ARCHIVELOG desde el estado MOUNT.
3. Haga clic en Yes cuando se le pregunte si desea reiniciar la base de datos.
4. Cree una copia de seguridad de la base de datos.
Las bases de datos en modo ARCHIVELOG tienen acceso a todas las opciones de copia de seguridad y de recuperación.

Al poner la base de datos en el modo ARCHIVELOG los redo logs no se sobrescriben hasta que no se han archivado. El siguiente comando SQL se utiliza para poner la base de datos en modo ARCHIVELOG:

```
SQL> ALTER DATABASE ARCHIVELOG;
```

Este comando se puede emitir sólo mientras la base de datos tiene el estado MOUNT, por lo que la instancia se debe reiniciar para terminar este último paso. Se le pedirá que indique las credenciales del sistema operativo y de la base de datos durante el reinicio de la base de datos.

Las credenciales de base de datos deben ser las de un usuario con privilegios SYSDBA. Una vez reiniciada la instancia, se activarán los cambios realizados en los procesos de archivado, formato de log y destinos de log.

Con la base de datos en modo NOARCHIVELOG (modo por defecto), la recuperación sólo es posible hasta el momento en que se realizó la última copia de seguridad. Todas las transacciones realizadas después se perderán.

En el modo ARCHIVELOG, la recuperación es posible hasta el momento en que se realizó la última confirmación. La mayoría de bases de datos de producción se ejecutan en modo ARCHIVELOG.

****Nota:** Se puede ver el estado del archivado con el comando "archive log list" del sqlplus.

Nota: Realice una copia de seguridad de la base de datos después de haber cambiado al modo ARCHIVELOG porque sólo podrá recuperar la base de datos de la última copia de seguridad realizada en ese modo.

Usuarios, roles, privilegios y perfiles

Objetivos: En este capítulo trataremos sobre todos los engranajes de seguridad de la base de datos, específicamente en materia de autenticación; el estudiante podrá hacer manejos de los controles de autenticación de usuarios.

Una **cuenta de usuario** de base de datos es un medio de organizar la propiedad y el acceso a objetos de base de datos.

Una **contraseña** es una autenticación por parte de la base de datos Oracle.

Un **privilegio** es un derecho para ejecutar un tipo concreto de sentencia SQL o para acceder a un objeto de otro usuario.

Un **rol** es un grupo con nombre de privilegios relacionados que se otorgan a los usuarios o a otros roles.

Los **profiles** imponen un juego con nombre de límites de recursos en cuanto al uso de la base de datos y de los recursos de la instancia.

La **cuota** es un espacio asignado en un tablespace determinado. Es uno de los métodos mediante el cual puede controlar el uso de recursos por parte de los usuarios.

Cada base de datos tiene una lista válida de usuarios a los que permite la conexión y cuyo nombre previamente se ha definido. Para cada usuario debe haberse creado una cuenta de usuario y especificado un nombre de usuario, método de autenticación, espacios de almacenamiento por defecto y temporal, otros espacios de almacenamiento y cuotas, perfil de usuario. Acceder a la BD supone que un usuario ejecute una aplicación y se conecte a la instancia usando un nombre valido. Tras la autenticación, puede autorizarse, o no, el acceso a determinados elementos y la ejecución de ciertas acciones. Las formas más comunes de autenticar a un usuario son bien por base de datos, bien por sistema operativo (autenticación externa).

CUENTAS DE USUARIO PREDEFINIDAS.

Diversas cuentas de usuario se crean durante la instalación. Así toda BD incluye las cuentas de administración SYS, SYSTEM, SYSMAN y DBSNMP (cuentas privilegiadas y necesarias para tareas de administración como el arranque y la parada de bd, gestión de memoria y almacenamiento, creación y gestión de usuarios, monitorización y gestión de la BD -cuenta DBSNMP del agente Database Control ...).

Las contraseñas de estas cuentas se asignan al crear la BD con el Database.

Configuration Assistant (DBCA). Estas cuentas NO deben borrarse.

También se crean determinadas cuentas "internas", internal accounts, para ciertas utilidades o componentes de la BD como, por ejemplo, Oracle Application Express. Estas cuentas inicialmente están bloqueadas y su clave ha expirado. NO deben borrarse y tampoco usarse para conectar a la BD.

CUENTAS DE USUARIO PREDEFINIDAS. SYS y SYSTEM.

Ambas se crean con la contraseña indicada durante la instalación y se les asigna el rol DBA.

SYS

Posee el privilegio SYSDBA lo que permite realizar tareas de administración. Puede realizar todas las tareas de administración. Las tablas y vistas del diccionario de datos se almacenan en el esquema SYS (dichas tablas se gestionan por la BD y no deben ser modificadas por ningún usuario). No deben crearse tablas en el esquema SYS.

SYSTEM

Permite realizar todas las tareas de administración excepto "Backup y recovery" y actualización de la BD, "Database upgrade". No obstante, Oracle recomienda crear cuentas específicas para administración y monitorización de la BD.

Práctica recomendada: Debido a la aplicación del principio de privilegio más bajo, estas cuentas no se utilizan para operaciones rutinarias. Los usuarios que necesiten privilegios DBA tienen cuentas separadas a las que se les otorgan los privilegios necesarios. Por ejemplo, Jim tiene una cuenta de privilegio bajo denominada jp y una cuenta con privilegios denominada jp_dba. Este método permite aplicar el principio de privilegio más bajo, elimina la necesidad de compartir cuentas y permite auditar acciones individuales.

Las cuentas SYS y SYSTEM son cuentas necesarias en la base de datos. No se pueden eliminar.

Autenticación de Usuarios

La autenticación significa verificar la identidad de alguien (un usuario, dispositivo u otra entidad) que desea utilizar datos, recursos o aplicaciones. La validación de dicha identidad establece una relación de confianza para una mayor interacción. La autenticación también permite establecer responsabilidades al posibilitar el enlace de acceso y acciones con identidades concretas. Tras la autenticación, los procesos de autorización pueden permitir o limitar los niveles de acceso y acción permitidos para dicha entidad.

Al crear un usuario, debe decidir la técnica de autenticación que se va a utilizar y que se podrá modificar posteriormente.

Password: También denominada autenticación por la base de datos Oracle. Cree cada usuario con una contraseña asociada que se debe proporcionar cuando el usuario intente establecer una conexión. Al configurar una contraseña, puede establecer que venza inmediatamente, lo que obliga al usuario a cambiar la contraseña después de la primera conexión. Si decide utilizar el vencimiento de contraseñas de usuario, asegúrese de que los usuarios pueden cambiar la contraseña. Algunas aplicaciones no tienen esta función.

Las contraseñas siempre se cifran de forma automática y transparente durante las conexiones de red (cliente/servidor y servidor/servidor) mediante un algoritmo modificado Data Encryption Standard (DES) antes de enviarlas por la red.

External: También se denomina autenticación por el sistema operativo. Los usuarios se pueden conectar a la base de datos Oracle sin especificar un nombre de usuario o contraseña. Con la autenticación externa, la base de datos confía en el servicio de autenticación de red o del sistema operativo subyacente para restringir el acceso a cuentas de base de datos. No se utiliza ninguna contraseña de base de datos para este tipo de conexión. Si el servicio de red o del sistema operativo lo permite, éste podrá autenticar usuarios. Si es así, defina el parámetro de inicialización OS_AUTHENT_PREFIX y utilice este prefijo en los nombres de usuario Oracle.

El parámetro OS_AUTHENT_PREFIX define un prefijo que la base de datos Oracle agrega al principio del nombre de cuenta de sistema operativo de cada usuario. El valor por defecto de este parámetro es OPS\$ para la compatibilidad con versiones anteriores del software de Oracle. La base de datos Oracle compara el nombre de usuario con prefijo con los nombres de usuario Oracle de la base de datos cuando un usuario intenta conectarse. Por ejemplo, suponga que OS_AUTHENT_PREFIX se ha definido de la siguiente forma:

OS_AUTHENT_PREFIX=OPS\$

Si un usuario con una cuenta de sistema operativo denominada tsmith se tiene que conectar a una base de datos Oracle y lo va a autenticar el sistema operativo, la base de datos Oracle comprueba si hay un usuario de base de datos OPS\$tsmith correspondiente y, si es así, permite al usuario conectarse. Todas las referencias a un usuario autenticado por el sistema operativo deben incluir el prefijo, como se ve en OPS\$tsmith.

Nota: El texto del parámetro de inicialización OS_AUTHENT_PREFIX es sensible a mayúsculas/minúsculas en algunos sistemas operativos. Consulte la documentación de Oracle específica para el sistema operativo si desea más información sobre este parámetro de inicialización.

Global: Mediante la opción Oracle Advanced Security, la autenticación global (que es una autenticación potente) permite identificar usuarios mediante biométrica, certificados X509, dispositivos de elemento y Oracle Internet Directory. Para obtener más información sobre métodos de autenticación avanzados, consulte el curso Oracle Enterprise Identity Management.

Autenticación de Administradores

Seguridad del Sistema Operativo: En UNIX y Linux, por defecto, los DBA pertenecen al grupo del sistema operativo oinstall, que posee los privilegios necesarios para crear y suprimir archivos de base de datos.

Seguridad del Administrador: Las conexiones SYSDBA y SYSOPER se autorizan únicamente después de la verificación con el archivo de contraseñas o con los privilegios y los permisos del sistema operativo. Si se utiliza la autenticación del sistema operativo, la base de datos no utiliza el nombre de usuario y contraseña proporcionados. La autenticación del sistema operativo se utiliza si no existe archivo de contraseñas, si el nombre de usuario o la contraseña proporcionados no están en ese archivo o si no se proporciona ningún nombre de usuario y contraseña.

No obstante, si la autenticación se produce mediante el archivo de contraseñas, la conexión se registra con el nombre de usuario. Si la autenticación se produce a través del sistema operativo, entonces es una conexión CONNECT / que no registra el usuario concreto.

Nota: La autenticación del sistema operativo tiene prioridad sobre la autenticación del archivo de contraseñas. En concreto, si es miembro del grupo OSDBA u OSOPER del sistema operativo y se conecta como SYSDBA o SYSOPER, lo hará con los privilegios administrativos asociados independientemente del nombre de usuario y contraseña que especifique.

Para conectar como SYS:

```
SQL> CONNECT SYS AS SYSDBA
```

```
SQL> CONNECT / AS SYSDBA
```

****Nota:** hay que pertenecer al grupo dba (Unix/Linux) o crear un "fichero de autenticación" en el SBD.

Usuarios y su autenticación

Cada base de datos tiene una lista valida de usuarios. Para acceder a la misma un usuario debe ejecutar un aplicación y conectarse a la instancia usando un nombre valido previamente definido.

Las formas más comunes de autenticar a un usuario son:

- Por base de datos.
- Por sistema operativo (autenticación externa).

Ahora conectaremos con la base de datos (el usuario Oracle que se conecta siempre es el SYS):

```
SQL> CONNECT usuario/clave AS SYSDBA
```

Creación, modificación y eliminación de usuarios

Necesario el privilegio de sistema CREATE USER. Normalmente sólo lo tiene el usuario administrador. El creador del usuario puede indicar cuota sobre espacios de almacenamiento aunque él no las posea. No es posible la conexión del usuario creado hasta que posea el privilegio de sistema CREATE SESSION.

```
SQL> CREATE USER <usuario>
IDENTIFIED BY <contraseña>/EXTERNALLY
DEFAULT TABLESPACE <espacio>
TEMPORARY TABLESPACE <espacio>
QUOTA <xx>K/UNLIMITED ON <espacio>
PROFILE <perfil>
PASSWORD EXPIRE
ACCOUNT LOCK/UNLOCK;
```

- **Nombre de usuario:** Debe ser único respecto a otros nombres de usuario y roles. Cada usuario tiene asociado un esquema y dentro del mismo cada objeto debe tener un único nombre.
- **Identificación:** Un usuario autenticado de forma externa se ha de crear con la cláusula “IDENTIFIED EXTERNALLY”.
- **DEFAULT TABLESPACE:** Indica aquel espacio de almacenamiento donde se crearan los objetos del esquema del usuario cuando al hacerlo no se indica ninguno en particular. Si no se indica es el espacio **SYSTEM** (iError!).
- **TEMPORARY TABLESPACE:** Indica el espacio de almacenamiento donde se almacenan los segmentos temporales requeridos por el Usuario. No debe indicarse cuota y el espacio temporal por defecto es el **SYSTEM** (iError!).
- **QUOTA:** Indica la cantidad de espacio reservada en un determinado espacio de almacenamiento para el usuario. Por defecto un usuario no tiene cuota en ningún espacio de almacenamiento. Indicando **UNLIMITED**, es ilimitado el espacio a usar. Puede revocarse el acceso a un espacio de almacenamiento asignando cuota cero sobre el mismo. Los objetos ya creados permanecen pero no pueden crecer ni crearse ninguno más.

- **PROFILE:** Indica el perfil a asignar al usuario.
- **PASSWORD EXPIRE:** Fuerza al usuario a cambiar la clave antes de conectarse a la base de datos.
- **ACCOUNT:** “**ACCOUNT LOCK**”, bloquea la cuenta de usuario e inhabilita el acceso. “**ACCOUNT UNLOCK**”, desbloquea la cuenta de usuario y permite al acceso.

```
SQL> Create user PERPINAN
identified by fcld
default tablespace DATOS
temporary tablespace TEMP
quota 10 M on DATOS
quota 10 M on TEMP
account unlock;
Grant create session to PERPINAN;
```

Modificación de usuario

Los usuarios pueden cambiar sus propias claves, sin embargo para cambiar cualquier otro parámetro es necesario el privilegio “**ALTER USER**”.

```
SQL> ALTER USER <usuario>
IDENTIFIED BY <contraseña>/EXTERNALLY
DEFAULT TABLESPACE <espacio>
TEMPORARY TABLESPACE <espacio>
QUOTA <xx>K/UNLIMITED ON <espacio>
DEFALT ROLE <role>/ALL/ALL EXCEPT
<role>/NONE
PROFILE <perfil>
PASSWORD EXPIRE
ACCOUNT LOCK/UNLOCK;
```

DEFAULT ROLE: Indica los roles otorgados por defecto al usuario en la conexión. Se refiere a roles otorgados de forma directa al usuario. Oracle activa los roles sin necesidad de especificar sus contraseñas.

Eliminación de usuarios

Al borrar un usuario el esquema asociado, con todos sus objeto, desaparecen. Una posible solución para que permanezca el usuario y los objetos pero impedir la conexión es revocar el privilegio “**CREATE SESSION**”.

No es posible eliminar un usuario que permanezca conectado a la base de datos. Debe esperarse a que concluya o forzar su terminación (**ALTER SYSTEM KILL SESSION**). Es necesario tener el privilegio de sistemas “**DROP USER**”. Es conveniente estudiar las implicaciones que sobre otros esquemas tiene el borrado del usuario y de su esquema.

```
SQL> DROP USER <usuario>;
```

```
SQL> DROP USER <usuario> CASCADE;
```

Perfiles

Un perfil está constituido por un conjunto de límites de recursos de la base de datos. Diferentes perfiles pueden ser asignados a diferentes usuarios.

Crear perfiles

Es necesario el privilegio de sistema **CREATE PROFILE**. Existe un perfil por defecto denominado **DEFAULT**. Inicialmente todos los recursos designados en él tienen valor **UNLIMITED**, por lo que es conveniente modificarlo. Un usuario al que no se le asigna perfil posee el perfil **DEFAULT**, aquellos recursos para los que en el perfil asignado no se ha definido un valor, o se ha indicado **DEFAULT**, toman el valor designado en el perfil por defecto.

```
SQL> CREATE PROFILE <nombre_perfil>
      LIMIT <parámetros> <valor>/UNLIMITED/DEFAULT;
```

Descripción de los parámetros de recursos:

- **SESSIONS_PER_USER**. Número de sesiones concurrentes.
- **CONNECT_TIME**. Tiempo total para una sesión (minutos).
- **IDLE_TIME**. Tiempo de inactividad continua en una sesión (minutos).
- **LOGICAL_READS_PER_CALL**. Número de bloques de datos para una llamada de una SQL.
- **PRIVATE_SGA**. Cantidad de espacio, en bytes, para uso privado reservado en la “shared pool” de la **SGA** (se emplea K o M para indicar kilobyte o megabytes). Solo en “Shared Server”.
- **FAILED_LOGIN_ATTEMPTS**. Número de intentos fallidos de conexión antes del bloqueo.
- **PASSWORD_LIFE_TIME**. Número de días en que la clave es válida para autenticación. Se indica un valor para **PASSWORD_GRACE_TIME**, la clave expira si no se cambia en este periodo. Si no se indica valor para **PASSWORD_GRACE_TIME**, por defecto **UNLIMITED**, se genera un aviso pero el usuario puede seguir conectándose.
- **PASSWORD_GRACE_TIME**. Periodo de gracia donde se permite la conexión pero se notifica la necesidad de cambiarla.

- **PASSWORD_REUSE_TIME** . Número de días en los cuales la contraseña no puede reutilizarse.
- **PASSWORD_REUSE_MAX** . Número de cambios de clave necesarios antes de poder reutilizar la clave actual.
- **PASSWORD_LOCK_TIME**. Número de días que la cuenta estará bloqueada después de un cierto número de fallos de conexión indicado.

```
SQL> CREATE PROFILE FCLD LIMIT  
FAILED_LOGIN_ATTEMPTS 5  
PASSWORD_LIFE_TIME 60  
PASSWORD_REUSE_TIME 60  
PASSWORD_REUSE_MAX 5  
PASSWORD_VERIFY_FUNCTION verify_function  
PASSWORD_LOCK_TIME 1/24  
PASSWORD_GRACE_TIME 10;
```

Modificar un perfil

Es necesario poseer el privilegio de sistema “**ALTER PROFILE**”. Los valores modificados no afectan a las sesiones en curso.

```
SQL> ALTER PROFILE <perfil>  
LIMIT <parámetros> <valor>/UNLIMITED/DEFAULT;
```

Asignación de perfil

Los perfiles no pueden asignarse a roles ni a otros perfiles, solo a usuarios. Se puede realizar durante la creación del usuario (**CREATE USER**) o posteriormente (**ALTER USER**). Un usuario sólo puede tener un perfil asignado a la vez. Las asignaciones de perfiles no afectan a las sesiones activas.

Borrar un perfil

Debe poseerse el privilegio de sistema **DROP PROFILE**. El perfil **DEFAULT** no puede borrarse. Para eliminar un perfil asignado a un usuario debe usarse la opción **CASCADE**. Si se borra un perfil asociado a un usuario, a este se le asigna de forma automática el perfil **DEFAULT**. El borrado de un perfil no afecta a las sesiones en curso.

```
SQL> DROP PROFILE <perfil>;
```

```
SQL> DROP PROFILE <perfil> CASCADE;
```

Privilegios

Son los derecho a ejecutar un tipo determinado de sentencia SQL o a acceder a un objeto de otro usuario. Pueden asignarse a usuarios o, preferiblemente, a roles. Es importante no excederse en la concesión de privilegios.

Se distinguen dos tipos:

- **De sistema:** Permite realizar determinadas acciones en la base de datos (Por ejemplo, crear espacios de almacenamiento, crear usuarios, ...) o en cualquier esquema.
- **Sobre objetos:** Permite a un usuario acceder y manipular o ejecutar objetos concretos (tablas, vistas, secuencias, procedimientos, funciones o paquetes).

Otorgar privilegios

Para que un usuario pueda otorgar un privilegio de sistema bien debe haberse otorgado con **ADMIN OPTION**, permite a aquel a quien se le concede el privilegio poder otorgarlo o haber sido concedido el privilegio **GRANT ANY PRIVILEGE**.

```
SQL> GRANT <privilegio>/ALL_PRIVILEGES TO <usuario>/<rol>/PUBLIC;
```

```
SQL> GRANT <privilegio>/ALL_PRIVILEGES TO <usuario>/<rol>/PUBLIC  
      WITH ADMIN OPTION;
```

Al especificar **ALL PRIVILEGES** se otorgan todos los privilegios de sistema. La cláusula **PUBLIC** otorga el privilegio a todos los usuarios.

Quitar privilegios

```
SQL> REVOKE <privilegio>/ALL_PRIVILEGES FROM <usuario>/<rol>/PUBLIC;
```

Cualquier usuario con la opción **ADMIN OPTION** sobre un privilegio puede revocarlo. Quien lo hace no tiene porque ser el usuario que originalmente lo otorgó. Al retirar ciertos privilegios determinados objetos pueden quedar inconsistentes (procedimientos o vistas consultadas merced al privilegio **SELECT ANY TABLE**).

En el caso de **ADMIN OPTION** no hay un efecto en cascada cuando se retira un privilegio referente a operaciones **DDL** (por ej. **CREATE TABLE**); si lo hay cuando se revoca un privilegio referente a operaciones **DML** (por ejemplo **SELECT ANY TABLE**). Si se retira un privilegio de sistema de **PUBLIC**, pero existen usuarios a los que se ha otorgado aquel directamente o a través de roles, estos siguen pudiéndolo usar.

ROLES

Es un grupo de privilegios, de sistema o sobre objetos, a los que se les da un nombre y pueden ser asignados a otros usuarios y roles.

Características de los roles:

- Pueden otorgarse a cualquier usuario o rol, pero no a si mismo y tampoco de forma circular.
- Pueden tener contraseña.
- Su nombre es único en la bd, distinto a cualquier otro nombre de usuario o rol.
- No pertenecen a ningún esquema.
- Simplifican el manejo de privilegios. Los permisos pueden asignarse a un rol y este a los diferentes usuarios.
- Manejo de privilegios dinámico. Si se modifican los privilegios asociados al rol, todos los usuarios que lo posean los adquieran de forma inmediata.

Roles predefinidos:

Oracle proporciona roles predefinidos como ayuda a la administración de base de datos, entre los que se encuentran:

- **CONNECT.** Incluye sólo el privilegio **CREATE SESSION**.
- **RESOURCE.** Incluye **CREATE CLUSTER**, **CREATE INDEXTYPE**, **CREATE OPERATOR**, **CREATE PROCEDURE**, **CREATE SEQUENCE**, **CREATE TABLE**, **CREATE TRIGGER** y **CREATE TYPE**.
- **DBA.** Todo privilegio de sistema **WITH ADMIN OPTION**.
- **EXP_FULL_DATABASE.** Privilegios para realizar exportaciones completas e incrementales de la base de datos.
- **IMP_FULL_DATABASE.** Idem para importaciones completas.
- **DELETE_CATALOG_ROLE.** Privilegio de borrado en la tabla de auditoria de sistema (**AUD\$**).
- **EXECUTE_CATALOG_ROLE.** Privilegio de ejecución sobre objetos en el diccionario de datos.
- **SELECT_CATALOG_ROLE.** Privilegio de consulta sobre objetos del diccionario de datos.

Es recomendable crear roles específicos en cada bd y asignarles los permisos necesarios, evitando el uso de roles predefinidos.

Creación de Roles

Debe poseerse el privilegio **CREATE ROLE**. El nombre debe ser diferente a cualquier nombre de rol o usuario existente.

```
SQL> CREATE ROLE <rol> IDENTIFIED BY <contraseña>;
```

```
SQL> CREATE ROLE <rol> NOT IDENTIFIED/<>;
```

Antes verificamos quienes tienen privilegios para crear roles:

```
SQL> select * from dba_sys_privs where privilege ='CREATE ROLE';
```

```
SQL> create role fcld;
```

```
SQL> grant create session to fcld;
```

Ahora asignamos a un usuario al rol creado:

```
SQL> grant fcld to aperpian;
```

La cláusula **IDENTIFIED BY** indica como debe ser autorizado antes de usarse por un usuario al que se la ha otorgado.

Modificación de roles.

Un rol solo puede modificarse para cambiar su método de autenticación. Debe poseerse el privilegio de sistema **ALTER ANY ROLE** o haber sido otorgado el rol con la opción **ADMIN**. No se ven afectadas las sesiones en las que el rol está ya activo.

```
SQL> ALTER ROLE <rol> NOT IDENTIFIED/ IDENTIFIED BY <contraseña>;
```

```
SQL> GRANT <rol> TO <usuario>/<rol>/PUBLIC;
```

```
SQL> GRANT <rol> TO <usuario>/<rol>/PUBLIC WITH ADMIN OPTION;
```

Para que un usuario pueda otorgar un rol debe habersele concedido con **ADMIN OPTION**, poseer el privilegio **GRANT ANY ROLE**, o haberlo creado. El usuario que crea el rol implícitamente lo tiene asignado con **ADMIN OPTION**.

Roles por defecto.

Un rol por defecto es aquel que automáticamente se activa al conectarse. Con la sentencia **ALTER USER** se limitan los roles por defecto asignados a un usuario. La cláusula puede sólo indicar roles otorgados directamente al usuario con una sentencia **GRANT**.

```
SQL> ALTER USER <usuario> DEFAULT ROLE <rol1>,...<roln>/ALL  
[EXCEPT rol1 [,role2]... ] / NONE;
```

Revocar un rol.

Puede hacerlo cualquier usuario con la opción **ADMIN OPTION** para un rol, también aquellos usuarios con el privilegio **GRANT ANY ROLE** (pueden revocar cualquier rol), con **PUBLIC** se des asigna el rol de todos los usuarios.

```
SQL> REVOKE <rol1>, ...<roln> FROM <usuario>|<rol>|PUBLIC, ...
```

Eliminación de roles.

Debe poseerse el privilegio **DROP ANY ROLE** o haber ido concedido el rol con **ADMIN OPTION**.

```
SQL> DROP ROLE <rol>;
```

Al borrar un rol se des asigna de todos los usuarios y roles, y se elimina de la base de datos. Las sesiones en las que el rol está activo no se ven afectadas, pero ninguna otra lo podrá usar.

Configuración del Entorno de Red de Oracle

Objetivos: Aprenderá el servicio de red para oracle, lo necesario que debe hacer un dba para realizar varias conexiones.

Servicios de Red de Oracle

Los servicios de Red de Oracle activan conexiones de red de una aplicación de cliente o de capa media al servidor de Oracle. Una vez que se ha establecido una sesión de red, la Red de Oracle actúa como servicio de mensajería de datos tanto para la aplicación cliente como para el servidor de bases de datos. Es responsable de establecer y mantener la conexión entre la aplicación cliente y el servidor de bases de datos, así como de intercambiar mensajes entre ambos. La Red de Oracle, o lo que simule la Red de Oracle, como Java Database Connectivity (JDBC), está ubicada en cada computadora que necesite hablar con el servidor de bases de datos.

En la computadora cliente, la Red de Oracle es un componente en segundo plano para conectar aplicaciones a la base de datos.

En el servidor de bases de datos, la Red de Oracle incluye un proceso activo denominado listener. El listener de Red de Oracle es responsable de coordinar conexiones entre la base de datos y las aplicaciones externas.

El uso más común de los servicios de Red de Oracle es permitir conexiones de base de datos entrantes. Puede configurar servicios de red adicionales para permitir el acceso a bibliotecas de código externo (EXTPROC) y conectar la instancia de Oracle a orígenes de datos que no sean de Oracle, como Sybase, Informix, DB2 y SQL Server, mediante los Servicios Heterogéneos de Oracle.

Listener de Red de Oracle

El listener de Red de Oracle es el gateway a la instancia de Oracle para todas las conexiones de usuario no locales. Un único listener puede ocuparse de varias instancias de la base de datos y de miles de conexiones del cliente.

Una de las maneras de acceder al listener es a través de Enterprise Manager. Puede controlar la configuración del listener real así como los parámetros generales, como la protección por contraseña y las ubicaciones de archivos log.

Los administradores avanzados también pueden configurar los servicios de Red de Oracle editando manualmente los archivos de configuración con un editor de texto del sistema operativo estándar, como vi o gedit, si es necesario.

Establecimiento de Conexiones de Red

Para que una aplicación se conecte a un servicio a través de un listener de Red de Oracle, la aplicación debe tener información sobre dicho servicio, incluida la dirección o el host donde reside el listener, el protocolo que acepta el listener y el puerto que controla el listener. Una vez localizado el listener, la parte final de información que necesita la aplicación es el nombre del servicio al que desea conectarse.

El proceso de determinar esta información de conexión se denomina “resolución de nombres”.

Una vez que termina la resolución de nombres de Red de Oracle, se transfiere una solicitud de conexión de la aplicación de usuario o de capa media (denominada proceso de usuario en adelante) al listener de Red de Oracle. El listener recibe un paquete CONNECT y comprueba si dicho paquete CONNECT está solicitando un nombre de servicio de Red de Oracle válido. Si no se solicita el nombre del servicio (como en el caso de una solicitud tnsping), el listener reconoce la solicitud de conexión y no hace nada. Si se solicita un nombre de servicio no válido, el listener transmite un código de error al proceso de usuario.

“Oracle Net Services” proporciona soluciones de conectividad para entornos distribuidos y heterogéneos.

Está integrado por:

- Oracle Net.
- Listener.
- Oracle Connection Manager.
- Herramientas de comunicaciones.
 - » Oracle Net Configuration Assistant.
 - » Oracle Net Manager.
 - » Oracle Enterprise Manager.

ORACLE NET.

Es el sw que permite establecer y mantener una sesión de red desde una aplicación cliente a un servidor de bd; así como del intercambio de mensajes entre ambos. Es capaz de hacerlo al estar localizado en cada ordenador de la red.

- Existen distintas tipologías de conexión:
 - Conexiones cliente/servidor (C/S).
 - Conexiones web cliente.
- Conexiones web cliente sin uso de servidor de aplicaciones (Application Server).

ORACLE NET. CLIENTE/SERVIDOR.

Oracle Net reside en el cliente y en el servidor. Se sitúa en la capa más alta de protocolos de red, que indica como las aplicaciones acceden a la red y como los datos se dividen en paquetes para su transmisión. Oracle Net interactúa con el protocolo TCP/IP -con o sin SSL- pero también puede trabajar con el protocolo SDP (redes "Infiniband").

- Oracle Net se compone a su vez de "Oracle Net foundation layer" (establece y mantiene la conexión) y "Oracle protocol support" (mapea a protocolos estándar desde la tecnología Oracle Net).
- Las aplicaciones cliente Java acceden a la bd Oracle empleando además JDBC (Java Database Connectivity Driver). Oracle ofrece dos manejadores "JDBC OCI" -clientes con instalación local Oracle- y "JDBC Thin Driver" -clientes sin instalación local Oracle, por ejemplo "applets"-.

OCI: Oracle Call Interface. SW usado por el cliente para interactuar con el servidor de bd (interfaz entre la aplicación cliente y el lenguaje SQL que comprende el servidor).

TTC: Two-Task Common.

OPI: Oracle Program Interface.

La conexión de aplicaciones web cliente (con o sin Web Application Server) a una bd es similar al caso C/S.

ORACLE NET. WEB y APP. SERVER.

El protocolo HTTP, "Hypertext Transport Protocol" permite comunicar entre los navegadores web y los servidores de aplicaciones, estos gestionan datos, controlan el acceso a los mismos y sirven las peticiones de los navegadores.

Un servidor de aplicaciones puede soportar aplicaciones Java y "servlets" también.

Es posible acceder desde los clientes web directamente a la bd, sin servidor de aplicaciones, como al usar " Java applet". Además la bd puede aceptar conexiones con protocolos como HTTP, FTP o WebDAV (conexiones a Oracle XML DB).

ORACLE NET. CONNECTION POOLING.

Si se ejecutan múltiples aplicaciones web interactivas, algunas de ellas pueden, en un momento dado, estar inactivas. "Connection pooling" permite a la bd interrumpir a nivel físico una sesión inactiva, la sesión lógica permanece abierta, y restablecer la conexión física cuando haya alguna petición.

LISNER.

En la bd recibe la conexión inicial. Está configurado con una dirección de protocolo, aquellos clientes configurados con esta dirección pueden enviar peticiones al "listener". Una vez establecida la conexión, el cliente y la bd comunican directamente entre ellos.

ORACLE CONNECTION MANAGER.

Permite que múltiples sesiones cliente sean multiplexadas a través de una sola conexión a una bd.

Oracle Net Configuration Assistant.

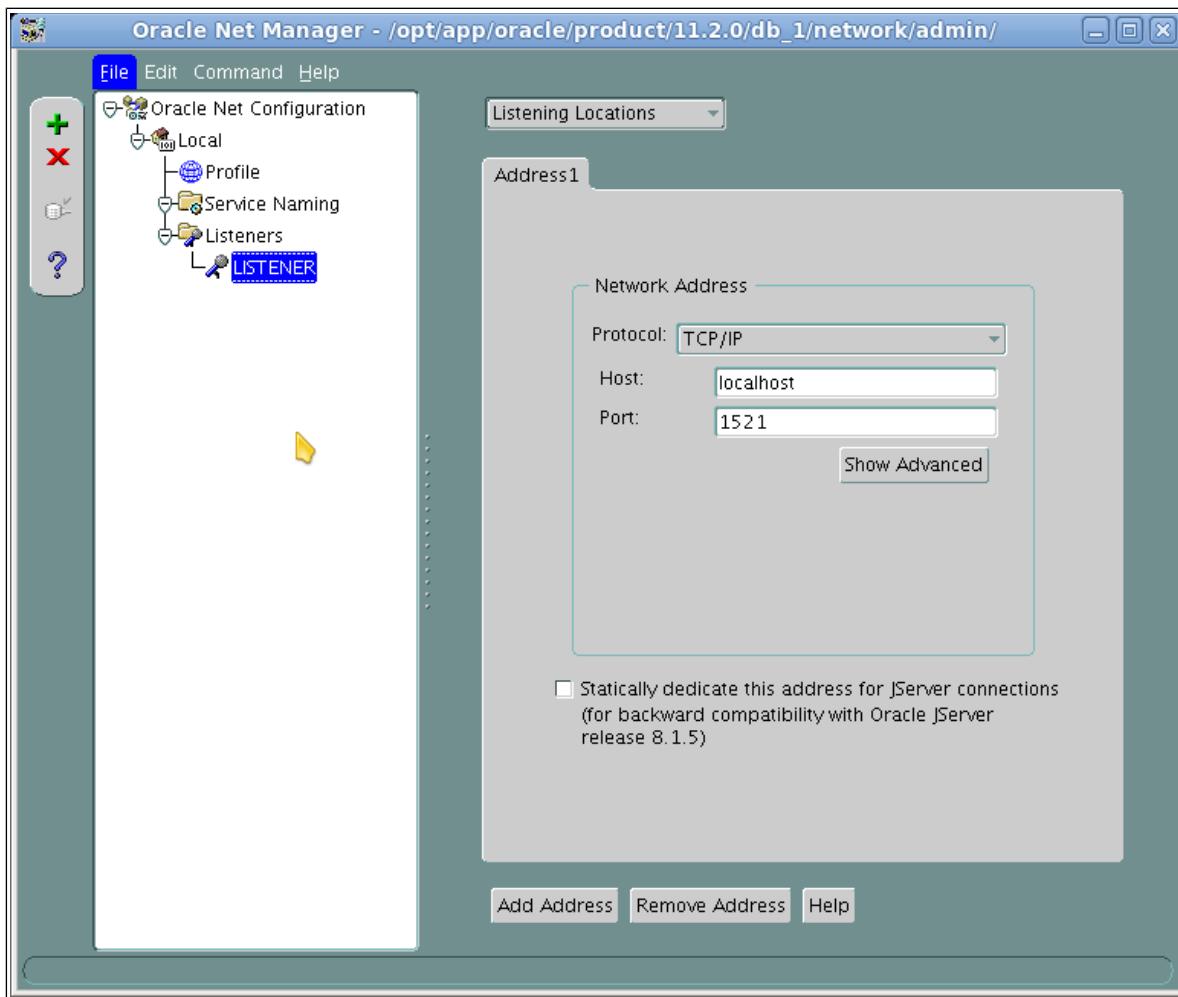
- Permite configurar los componentes básicos de red después de realizar la instalación de bd.
- Oracle Universal Installer lanza esta herramienta durante la instalación del sistema gestor de bd.
- En UNIX debe ejecutarse \$ORACLE_HOME/bin/netca



Oracle Net Manager.

- Proporciona un entorno integrado de configuración y gestión de Oracle Net Services.
- En UNIX debe ejecutarse \$ORACLE_HOME/bin/netmgr

En Oracle 11g, muchas funcionalidades de Net Manager están integradas en Oracle Enterprise Manager.



Una bd es un “servicio” desde el punto de vista de clientes.

El servicio está identificado por un nombre de servicio que viene determinado por el parámetro de inicialización SERVICE_NAMES. Por defecto es el nombre “global database”, compuesto por el nombre de bd (parámetro DB_NAME) y el nombre de dominio (DB_DOMAIN). Los clientes usan el nombre de servicio para identificar la bd a la que acceden.

La bd puede tener uno o más servicios asociados; así puede ser identificada de distintas formas por clientes diferentes, y un administrador puede limitar recursos del sistema.

La información necesaria para una conexión se almacena en un repositorio (local o centralizado -LDAP-).

Una base de datos tiene al menos una instancia integrada por un área de memoria llamada System Global Area (SGA) y por procesos Oracle.

El nombre de la instancia viene indicado por el parámetro de inicialización INSTANCE_NAME. Por defecto se refiere al Oracle System Identifier (SID) de la instancia.

REGISTRO DE SERVICIO

“Service registration” es una utilidad por la que el proceso PMON automáticamente proporciona información al “listener”, de forma que listener.ora no precisa ser configurado con información estática tal como:

- Nombres de servicio para cada instancia de la bd.
- Nombres de instancia de la bd.
- Manejadores de servicio (“dispatcher” disponible para cada instancia. o servidor dedicado)
- Dispatcher, instancia e información “node load” que permite al “listener” determinar qué “dispatcher” puede manejar adecuadamente una petición cliente. Si todos los “dispatchers” están bloqueados, el “listener” puede habilitar un servidor dedicado para la conexión.
- Para asegurar que el registro de servicio funciona correctamente el fichero de parámetros de inicialización debería contener los parámetros:
SERVICE_NAMES, para el nombre de servicio de la bd
INSTANCE_NAME, para el nombre de la instancia
- El valor del parámetro SERVICE_NAMES se considera antes que el “global database name” (compuesto por DB_NAME y DB_DOMAIN) y el valor de INSTANCE_NAME se considera antes que el SID indicado durante la instalación o creación de la bd.

Los eventos relacionados con el registro de servicio se almacenan en el fichero listener.log:

- service_register. Recepción de información de instancia por el “listener”.
- service_update. Recepción de información actualizada de una instancia por parte del “listener”.

– service_died. El “listener” ha perdido su conexión con el PMON. Se descarta cualquier información de registro para la instancia, los clientes no podrán conectarse a la instancia hasta un nuevo registro del PMON.

```
[jp@oraserv5 ~]$ cat $ORACLE_HOME/network/admin/tnsnames.ora
# tnsnames.ora Network Configuration File:
#/opt/app/oracle/product/11.2.0/db_1/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.
ORASERV5 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = oraserv5)
    )
  )
```

Los manejadores de servicio en el servidor pueden ser un “dispatcher” (modo servidor compartido) o un proceso servidor dedicado.

Puede suceder que un cliente desee usar un manejador de servicio particular, en este caso debe indicarse en el conector:

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=host)(PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=nombre_servicio)
    (SERVER=shared/dedicated)))
```

Si no se indica el valor para el parámetro SERVER se asume por defecto la configuración “shared server”. Si no hay “dispatchers” disponibles, se usarán servidores dedicados.

TIPOS DE configuración

Existen dos modelos de configuración de Oracle Net:

- Localizado o no centralizado: la información se almacena en un fichero tnsnames.ora local a cada máquina en la red.
- Centralizado: la información se guarda en un directorio central (LDAP).
 - En la exposición se considera un modelo de configuración no centralizado, donde la información sobre direcciones de red se almacena en el fichero tnsnames.ora de cada sistema.

MODELO LOCAL.

Existen los siguientes ficheros de configuración:

- cman.ora ... Residente en la máquina donde se ejecuta Oracle Connection Manager.

- listener.ora ... En el servidor de bd. Con información sobre protocolo , dirección de escucha, servicios para los que se escucha y parámetros de control.
- sqlnet.ora ... En el cliente y en el servidor bd. Puede contener ruta de conexiones, parámetros de control de acceso a bd, parámetros de Oracle Advanced Security, ...
- tnsnames.ora ... En los clientes. Contiene “net service names” -nombre de servicio asociado a un descriptor de conexión-. Usado en los mapeos.
 - Los ficheros de configuración residen generalmente en "\$ORACLE_HOME/network/admin", aunque pueden hacerlo en otros lugares.
 - El orden de búsqueda para sqlnet.ora es:

- El directorio indicado por la variable TNS_ADMIN
- El directorio \$ORACLE_HOME/network/admin

El orden de búsqueda tnsnames.ora es: para cman.ora, listener.ora y tnsnames.ora es:

- El directorio indicado por la variable TNS_ADMIN
- En UNIX, el directorio global de configuración (“global configuration directory”).
- El directorio \$ORACLE_HOME/network/admin

TNSNAMES.ORA

- Contiene nombres de servicio de red, usados al establecer una conexión, de red mapeados a descriptores de conexión. Dichos descriptores contienen la localización de un “listener” a través de una dirección de protocolo y el nombre de servicio de la bd al que conecta.
- Por defecto, tnsnames.ora reside en \$ORACLE_HOME/network/admin aunque también puede estar en el directorio indicado por la variable de entorno TNS_ADMIN o el “global configuration directory”. Su sintaxis básica es:

```
ORASERV5 =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
)
(CONNECT_DATA =
(SERVICE_NAME = localhost)
)
)
```

LIS TENER.ORA

El “listener” es un proceso que corre en la bd. Recibe peticiones de conexión de los clientes y gestiona el tráfico de dichas peticiones a la bd. Su configuración se almacena en un fichero llamado listener.ora y sus mensajes se generan en el fichero listener.log

Dado que todos los parámetros de configuración tienen valores por defecto, es posible arrancar y usar el “listener” sin configurar. El “listener” se llama por defecto LISTENER, no soporta servicios hasta el arranque de la bd, y escucha atendiendo a los siguientes valores:
(ADDRESS=(PROTOCOL=tcp)(HOST=host_name)(PORT=1521))

El puerto por defecto para conexión al “listener” es el 1521 (aunque en futuras versiones puede cambiar al 2483, TCP/IP, y 2484, TCP/IP con SSL). Puertos inferiores a 1024 están reservados para su uso por procesos privilegiados.

Cada listener está configurado con una o varias direcciones de protocolo, los clientes configurados con una de estas direcciones pueden enviarle peticiones. Una vez recibida la petición, el listener elige un manejador de servicio apropiado para servir la petición y redirige la misma hacia él.

Si el listener no se está ejecutando cuando arranca la instancia, PMON no puede registrar la información de servicio. PMON intenta conectar periódicamente, no obstante, pueden pasar hasta 60 segundos antes de que PMON registre la información tras el arranque del listener. Para iniciar el registro de información inmediatamente tras el arranque del listener debe emplearse la sentencia SQL:

ALTER SYSTEM REGISTER:

Por defecto, PMON registra información del servicio referente al puerto TCP/IP número 1521. Para poder usar un puerto distinto debe configurarse el parámetro de inicialización LOCAL_LISTENER

En un entorno “shared server” puede emplearse el atributo LISTENER del parámetro DISPATCHERS en el fichero de inicialización. Como LOCAL_LISTENER y el atributo LISTENER permiten a PMON registrar información del “dispatcher” con el listener, no es preciso especificar ambos si los valores del listener son los mismos.

Por ejemplo, para configurar el listener en el puerto 1421:

```
LOCAL_LISTENER=listener1  
DISPATCHERS="(PROTOCOL=tcp)(LISTENER=listener1)"
```

En el tnsnames.ora:

```
listener1=(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST= host)(PORT=1421)))
```

El registro dinámico de servicios configuración del fichero listener.ora no requiere ninguna.

El “listener” emplea la información dinámica adquirida mediante el registro dinámico antes de usar la información configurada estáticamente en el fichero listener.ora.

Durante la instalación de la bd, Oracle Universal Installer ejecuta el Oracle Net Configuration Assistant; este permite configurar el “listener” con un nombre y una dirección concreta.

También se crea una entrada para escucha del protocolo IPC que se configura de forma automática y sirve a procedimientos externos (escritos en un 3GL que pueden ser llamados desde el código PL/SQL -sólo C-).

CONTROL DEL “LISTENER”.

La utilidad para control del “listener” (Listener Control Utility) permite administrar este componente. La sintaxis es:

```
[jp@oraserv5 ~]$ LSNRCTL <sentencia> [nombre_listener]
```

Si no se indica nombre, hace referencia al “listener” de nombre “listener”. Para arrancar el “listener” lsnr:

```
[jp@oraserv5 ~]$ LSNRCTL START lsnr
```

También pueden ejecutarse sentencias en el “prompt” LSNRCTL (se obtiene tecleando lsnrctl sin argumentos).

```
[jp@oraserv5 ~]$ lsnrctl LSNRCTL> START lsnr
```

```
[jp@oraserv5 ~]$ lsnrctl start
```

```
[jp@oraserv5 ~]$ nmap localhost
```

Oracle Flashback

Objetivos: Aprenderá activar y recuperar objetos desde el FlashBack.

Área de Recuperación de Flash

Tecnología de Flashback: Ventajas

La arquitectura de la base de datos Oracle 11g aprovecha las ventajas tecnológicas únicas en el área de recuperación de bases de datos por la pérdida de datos debido a errores humanos. La tecnología de flashback ofrece un juego de funciones nuevas para visualizar y rebobinar los datos hacia delante y hacia atrás en el tiempo.

Esta tecnología revoluciona la recuperación ya que simplemente actúa en los datos cambiados. El tiempo que se tarda en recuperarse del error equivale ahora al que se tarda en cometerlo. Cuando sea aplicable, la tecnología de flashback ofrece ventajas importantes sobre la recuperación del medio físico en términos de facilidad de uso, disponibilidad y tiempo de restauración.

El área de recuperación de flash es un espacio separado en el disco para contener archive logs, copias de seguridad y logs de flashback.

Si ha configurado los archive logs para que se escriban en esta ubicación (con el indicador USE_DB_RECOVERY_AREA en una de las ubicaciones), es importante controlar este espacio para asegurarse de que no alcanza la capacidad.

Si la instancia no puede crear un archive log debido a falta de espacio, se detiene hasta que el administrador corrija la situación.

Al hacer clic en Recovery Settings en la página de propiedades Maintenance se accede a los valores de Flash Recovery Area. En esta página puede:

- Especificar la ubicación del área de recuperación de flash
- Especificar el tamaño del área de recuperación de flash (Oracle recomienda que sea al menos dos veces el tamaño de la base de datos para que pueda contener una copia de seguridad y varios archive logs)
- Verificar qué cantidad del área de recuperación de flash se ha consumido

Es posible restaurar una tabla a un estado anterior a aquel donde tuvieron lugar cambios, usando el espacio de "undo", mediante la sentencia FLASHBACK TABLE. Se proporciona así una forma rápida de recuperar una tabla modificada o borrada y se evita recurrir a métodos complejos.

La función de flashback de base de batos es más rápida que la recuperación point-in-time tradicional que utiliza archivos restaurados y archivos redo log. A medida que una base de datos aumenta de tamaño, la cantidad de tiempo necesaria para restaurar todos los archivos de datos con el fin de llevar a cabo una recuperación point-in-time tradicional resulta prohibitiva. Con el flashback de base de datos, el tiempo para recuperar una base de datos es ahora proporcional al número de cambios que es necesario deshacer (y no al tamaño de la base de datos) porque no tiene que restaurar los archivos de datos.

El flashback de base de datos se implementa mediante un tipo de archivo log denominado log de flashback de base de datos. La base de datos Oracle registra periódicamente "imágenes anteriores" de bloques de datos en los logs de flashback de base de datos. Las imágenes de bloque se pueden reutilizar para hacer retroceder con rapidez los cambios en los archivos de datos hasta cualquier momento en que se hayan capturado logs de flashback justo antes de la hora deseada. Entonces, se aplican los cambios a partir de los archivos redo log para llenar el hueco. Los logs de flashback de base de datos se crean y gestionan de forma automática en el área de recuperación de flash.

Características:

- La operación se realiza en línea.
- Se mantienen automáticamente todos los atributos de la tabla (índices, disparadores, "constraints") necesarios.
- Se mantienen las restricciones de integridad.
- No se pierden los datos originales, pudiendo volver al principio (no puede hacerse un "rollback" de la sentencia, pero puede realizarse una nueva recuperación a un momento justo anterior al actual -es conveniente anotar previamente el SCN actual-).
- Durante la recuperación se adquiere un bloqueo DML que impide cualquier operación sobre la tabla. Esta se realiza en una transacción única, independientemente del número de tablas que comprenda la recuperación. Requisitos:
 - Debe poseerse el privilegio FLASHBACK ANY TABLE o tener el privilegio FLASHBACK sobre la tabla.
 - Deben poseerse los privilegios SELECT, INSERT, DELETE, y ALTER en la tabla.
 - La información de "undo" debe ser suficiente como para recuperar hasta el punto indicado en la operación.

- Debe habilitarse en la tabla la característica “row movement” mediante la sentencia:

```
SQL> ALTER TABLE <nombre_tabla> ENABLE ROW MOVEMENT;
```

Al borrar una tabla, la bd no borra inmediatamente el espacio ocupado por la misma. La bd renombra la tabla y la coloca junto con sus objetos asociados (“constraints”, tablas anidadas y semejantes) en lo que denomina “recycle bin”, de forma que si fue borrada por error es posible recuperarla usando la sentencia FLASHBACK TABLE (característica “flashback drop”).

El “recycle bin” es una tabla de diccionario con información sobre los objetos borrados y sus asociados, los cuales no se eliminan hasta indicarlo explícitamente o hasta haber necesidades de espacio (se eliminan en orden “first-in first-out” - FIFO-). Cada usuario, a menos que tenga privilegio SYSDBA, sólo puede acceder a sus propios objetos en esta tabla:

```
SQL> SELECT * FROM RECYCLEBIN;
```

El espacio libre en un espacio de almacenamiento puede consultarse en la vista DBA_FREE_SPACE.

recuperación DE TABLAS. “RECYCLE BIN”.

Al borrar un espacio de almacenamiento incluyendo su contenido, los objetos en el mismo no se almacenan en el “recycle bin” y la BD elimina cualquier referencia a ellos. Igualmente sucede cuando se borra un usuario, “cluster” o “type”.

Este modo de funcionamiento, habilitado por defecto, puede habilitarse o deshabilitarse mediante el parámetro de inicialización RECYCLEBIN, o, dinámicamente, mediante las sentencias:

```
SQL> ALTER SESSION/SYSTEM SET recyclebin = OFF;
```

```
SQL> ALTER SESSION/SYSTEM SET recyclebin = ON;
```

Deshabilitar “recycle bin” no afecta a los objetos que ya se hallan en el mismo.

También se obvia en ciertas ocasiones el “recycle bin”:

En la sentencia DROP TABLE ... PURGE (ej. SQL> drop table emp purge;).

En la sentencia DROP TABLESPACE ... INCLUDING CONTENTS. Cada espacio de almacenamiento tiene su propio “recycle bin” por lo que borrar el espacio implica eliminar este y cualquier objeto en el mismo.

Al emplear DROP USER...CASCADE. Se elimina el usuario y todos sus objetos (incluidos aquellos en el “recycle bin”).

A los objetos almacenados en el “recycle bin” se les asigna un nombre generado por el sistema de la forma BIN\$unique_id\$version , donde :

- unique_id es un identificador único para el objeto compuesto por 26 caracteres, y
- versión es un número de versión asignado por la bd.

Es posible obtener información mediante las vistas:

- USER_RECYCLEBIN. Permite a un usuario ver los objetos de su propiedad que han sido borrados, tiene un sinónimo: RECYCLEBIN.
- DBA_RECYCLEBIN. Da información a los administradores sobre todos los objetos borrados. O mediante la sentencia SQL*Plus SHOW RECYCLEBIN.

Mediante la sentencia PURGE puede borrarse explícitamente el contenido del “recycle bin” y liberar el espacio asociado (son necesarios los mismos privilegios requeridos para borrar el elemento):

```
SQL> PURGE TABLE/INDEX <nombre>;
```

Nota: Al eliminar una tabla puede usarse el nombre asignado en el “recycle bin” o su nombre original.

```
SQL> PURGE TABLESPACE <nombre_tbsp> [USER <nombre_usuario>];
```

Nota: Se eliminan todos los objetos de “nombre_usuario” para el esp.almacenamiento “nombre_tbsp”.

```
SQL> PURGE RECYCLEBIN/DBA RECYCLEBIN;
```

Nota: Si se tiene el privilegio SYSDBA puede eliminarse todo el contenido del “recycle bin” indicando DBA_RECYCLEBIN.

recuperación DE TABLAS. “FLASHBACK TO BEFORE DROP”.

Con el flashback de tabla, puede devolver una o más tablas a un punto en el tiempo específico sin restaurar ninguna copia de seguridad.

Los datos se recuperan de un tablespace de deshacer para realizar una operación de flashback de tabla.

El privilegio FLASHBACK TABLE es necesario para realizar un flashback en una tabla.

El movimiento de filas debe estar activado en la tabla en la que se vaya a realizar la operación de flashback.

La sentencia FLASHBACK TABLE... TO BEFORE DROP permite recuperar objetos del “recycle bin”. Su uso requiere tener los mismos privilegios que se necesitan para borrarla:

```
SQL> FLASHBACK TABLE <nombre_tabla> TO BEFORE DROP RENAME  
TO <nombre_tabla>;
```

Puede indicarse bien el nombre de la tabla en el “recycle bin”, bien su nombre original. La cláusula “RENAME TO” permite renombrar la tabla.

Usar el nombre del “recycle bin” es útil si se ha borrado la tabla varias veces. En este caso puede consultarse el “recycle bin” y hacer “flashback” con el nombre correspondiente a la versión que quiera recuperarse:

```
SQL> SELECT object_name, original_name, createtime  
FROM recyclebin;
```

Los objetos que dependen de las tablas recuperadas, como los índices, no recuperan su nombre original sino que mantienen el generado por el “recycle bin”. Estos objetos deben renombrarse de forma manual si bien habría que averiguar cuáles son antes de restaurar la tabla.

Tendrá que renombrarlo y el nombre original debe ir entre dobles comillas. Resumen:

- RECYCLE BIN (>=10g). Contenedor donde Oracle guarda las tablas borradas (a no ser que se borre con la opción PURGE).
- DBA_RECYCLEBIN. Ver todas las tablas borradas. En USER_RECYCLEBIN o RECYLEBIN, sólo las mías. Tb con SHOW RECYLEBIN (desde sql*plus).
- Se puede desactivar con parámetro del init “recyclebin = off” (por defecto “on”). Tb con “ALTER SYSTEM | SESSION ...”.
- Recuperar tabla borrada:
FLASHBACK TABLE tablaBorrada TO BEFORE DROP;

- Borrar definitivamente:
PURGE TABLE tabla; (tb PURGE INDEX)
- Sólo van a RECYCLEBIN los índices de las tablas borradas (no con DROP INDEX)
 - PURGE RECYCLEBIN | DBA_RECYCLEBIN;
 - PURGE TABLESPACE nomtsp [USER usuario];
- Borrar directamente sin pasar por RECYCLEBIN:
 - DROP TABLE tabla PURGE;

Copias de seguridad y recuperación

Objetivos: Aprender el manejo de las herramientas de respaldo de datos, así como la importación y extracción de datos de diferentes fuentes, el estudiante podrá realizar tareas de mantenimiento sobre la copia de seguridad del sistema de base de datos.

Pueden distinguirse dos tipos de copias:

Copias de seguridad físicas:

Se realiza la copia de los ficheros que constituyen la base de datos. Se distinguen copias de seguridad en línea ("en caliente") y fuera de línea ("en frío"). Se pueden hacer manualmente (comando "tar" de s.o.) o mediante la utilidad RMAN ("recovery manager") de Oracle.

Copias de seguridad lógicas. Implican la lectura de un conjunto de registros de base de datos y su escritura en un fichero especial.
Se emplean utilidades propias de la base de datos: export/import -obsoletas- y data pump export/import -recomendadas-.

Las copias de seguridad lógicas se emplean para:

- Transferir objetos de datos entre bases de datos Oracle (independientemente de la plataformas en que residan).
- Proporcionar copia de seguridad lógica para objetos de la base de datos, un espacio de almacenamiento o la base de datos al completo.
- Migración entre versiones de base de datos (¡Peligro!). Una buena estrategia de copias de seguridad incluirá ambos tipos de copias, físicas y lógicas, pues son complementarias.

COPIA FISICA.

Realizar una copia física implica copiar los sistemas de ficheros asociados a la base de datos y aquellos donde se ha instalado la misma.

La copia de los ficheros de base de datos y de administración se realiza de forma diaria.

La copia del "software" de base de datos se realiza con frecuencia semanal, por ejemplo, dado su menor nivel de actualización.

Dado que se asume que la base de datos está en modo noarchivelog, sólo es posible realizar copias "en frío" de la misma. Por tanto, previamente a la copia física se efectuará una parada de la base de datos, de forma que su contenido sea íntegro y coherente.

COPIA FISICA. SISTEMAS DE FICHEROS.

Los sistemas de ficheros a copiar diariamente son:

Sistemas de ficheros de administración de la bd, bajo \$ORACLE_BASE/admin, y ficheros del ADR. Incluirá la copia del fichero de parámetros de inicialización "init.ora" y del fichero de parámetros "spfile" en su caso.

Sistemas de ficheros asociados a la base de datos que contendrán los ficheros de datos (.dbf), de control (.ctl) y de "redo log" (.log) de la base de datos (aquellos de la forma, por ejemplo, /uxx/oradata/<nombre_bd>).

Sistemas de ficheros asociados a la exportación de base de datos.

Sistema de ficheros /etc

Los sistemas de ficheros a copiar semanalmente son:

Sistemas de ficheros que integran la estructura O.F.A. Bajo \$ORACLE_BASE (/opt/app/oracle).

Todos los sistemas de ficheros que integran la copia diaria de la base de datos.

COPIA FISICA: TAR (UNIX)

La copia se realizará tras:

Haber hecho la copia lógica, exportación, correspondiente de la base de datos.

Haber cerrado cualquier comunicación con la base de datos (proceso "listener").

Cerrar la base de datos (shutdown) de un modo consistente.

El proceso debe ser automatizado y el cese de actividad de la base de datos afectar lo menos posible al servicio.

Puede usarse la sentencia TAR de sistema operativo (copia los sistemas de ficheros indicados a un dispositivo físico)

Copia logica export/import

Objetivos: Ya usted como DBA podrá realizar migración de BD a nivel lógica?.

La utilidad **export** de Oracle permite crear un archivo histórico con la información de objetos de la base de datos o bien de la base de datos completa. Esta utilidad permite almacenar tanto las definiciones de las tablas como los datos que contengan. También permite el intercambio de información entre máquinas, bases de datos o versiones del servidor de Oracle. Es muy útil para proporcionar protección contra las perdidas de datos de los usuarios.

Existen tres niveles de realizar la exportación

- **Tabla:** exporta las tablas especificadas en el esquema de usuario. Solamente los usuarios con privilegios pueden exportar tablas de otros esquemas
- **Usuario:** exporta todos los objetos del esquema del usuario
- **Base de Datos:** exporta todos los objetos de la base de datos excepto los del esquema **SYS**. Solo los usuarios con privilegios pueden realizar este tipo de exportación.

El comando **exp** crea un fichero que únicamente es posible restaurar con la utilidad **import**. Este comando debe ejecutarse desde la consola, no dentro del **sqlplus**.

La sintaxis del comando **export** es:

```
[oracle@oraserv5 ~]$ exp keyword = valor
```

```
[oracle@oraserv5 ~]$ exp keyword = (valor1,valor2,...)
```

y los principales parámetros son:

- **USERID** Usuario/password del esquema a exportar
- **FILE** Fichero de salida
- **ROWS** Incluye las fila de la tabla en la exportación
- **TABLES** Tablas que deben exportarse
- **HELP** Muestra los parámetros de export en modo interactivo
- **LOG** Fichero para mensajes de información y error

```
[oracle@oraserv5 ~]$ exp hr/kikla file=hr.dmp full=yes log=hr.log
```

La utilidad **import** permite la recuperación de información de la base de datos a partir de un fichero exportado mediante la utilidad **export**.

Dependiendo del modo en que se haya realizado el **export**, podemos realizar diferentes importaciones. (definiciones de tablas, definiciones más datos, esquemas completos,...)

La sintaxis del comando **import** es:

```
[oracle@oraserv5 ~]$ imp keyword = valor
```

```
[oracle@oraserv5 ~]$ imp keyword = (valor1,valor2,...)
```

Los parámetros que utiliza son los mismos que la utilidad **export**.

La secuencia del proceso de importación es al siguiente:

- Se crean las tablas.
- Se definen los índices
- Se importan los datos.
- Los índices son cargados
- Se importan los triggers y se habilitan las restricciones de integridad de las nuevas tablas.

Hay que destacar que el orden en el que se importan las tablas es importante. Por ejemplo, si una tabla con una clave ajena, tiene un restricción referencial con una clave primaria de otra tabla, y la tabla con la clave ajena es importada primero, todas las filas que hagan referencia a la clave primaria no serán importadas.

```
[oracle@oraserv5 ~]$ imp antonio/kikla file=hr.dmp full=yes
```

El contenido del fichero de parámetros será:

```
[oracle@oraserv5 ~]$ vi parfile.txt
buffer=102400
file=/export/CURSO01/tablas_scott05.dmp
log=/export/CURSO01/tablas_scott05.log
tables=(EMP, DEPT)
compress=y
```

Introducción a Oracle Data Pump

¿Qué es?

Son utilidades integradas en la Base de Datos que permiten cargas y descargas de información a gran velocidad. Todo el proceso de exportación/importación se lleva a cabo en el servidor.

No son compatibles con el export/import originales, aunque estas últimas hay necesidad de utilizarlas en los siguientes casos:

- Al importar ficheros creados con la utilidad "export". Por ejemplo, aquellos procedentes de bases de datos con versiones anteriores a la 10g.
- Si se desea exportar ficheros que serán importados con la utilidad "import". Por ejemplo, si se van a transferir datos a bases con una versión anterior a la 10g.

¿De qué está formado?

Consta de los siguientes componentes:

- Clientes expdp e impdp. Usan los procedimientos proporcionados por el paquete DBMS_DATAPUMP.
- Paquete DBMS_DATAPUMP. Conocido como Data Pump API, permite crear y monitorizar los trabajos implicados.
- Paquete DBMS_METADATA. Conocido como Metadata API, proporciona la definición de los objetos de la bd a Data Pump

¿Cuándo es útil exportar / importar datos en Oracle?

Se suelen utilizar para hacer cosas como:

- Migraciones de base de datos, tablas o usuarios.
- Copia de seguridad de base de datos tablas concretas o esquemas de usuarios.
- Implementar algún dato concreto de una base de datos en otra base de datos.

¿Por qué usar Oracle Data Pump?

- Gestiona infinitamente mejor el tema del "mapeo" de esquemas y tablespaces (una de las pesadillas del antiguo export/import).
- Permite compresión de los datos "al vuelo", ya no es necesario exportar y comprimir vía "pipe" o una vez exportado (sólo Enterprise Edition).

- Nos da un control extraordinario de que exportar y cómo hacerlo, podemos filtrar que objetos queremos y que objetos no (incluso a nivel de filas para tablas concretas).
- Si se para por problemas, en ciertos casos (por ejemplo si se ha quedado sin espacio en el tablespace) no aborta, se pone en pausa y permite solucionar el problema y continuar.
- Permite paralelizar y cambiar el paralelismo “al vuelo”, aumentando o disminuyendo el número de procesos de carga/extracción (solo Enterprise Edition).
- Permite cargar datos “al vuelo” de otra BBDD mediante DBLink, esto es, sin pasar por fichero en disco (incompatible con campos de tipo LONG y RAW).
- Permite exportar datos de un SCN o timestamp anterior al actual, lo que permite exportar la BBDD como estaba en un momento anterior del tiempo.
- Según Oracle es MUCHO más rápida que sus predecesoras, citando textualmente:

LOCALIZACION DE FICHEROS.

Es necesario crear objetos tipo directorio (asocia un nombre a un directorio o sistema de ficheros) por un usuario administrador o con el privilegio CREATE ANY DIRECTORY. Al exportar/importar se indica mediante el parámetro DIRECTORY.

```
SQL> CREATE DIRECTORY bk_datapump AS '/opt/Backups/oraserv5 ';
```

Por defecto se asigna el valor para DATA_PUMP_DIR:

Tras la creación de un directorio, su creador debe conceder el permiso de lectura (READ) o escritura (WRITE) a otros usuarios.

```
SQL> GRANT READ, WRITE ON DIRECTORY <directorio> TO <usuario>;
```

Nota: Implica tener acceso a través de bd pero no el poder acceder a través de s.o.

El orden de precedencia usado para determinar la localización es:

- 1.- Objeto directorio formando parte de la especificación de un nombre de fichero (DUMPFILE=<localización>:<fichero>.dmp).
- 2.- Objeto directorio indicado en el parámetro DIRECTORY.
- 3.- Valor de la variable de entorno DATA_PUMP_DIR (definida en el entorno del cliente):

```
SQL> CREATE DIRECTORY DUMP_FICHERO AS '/DATAPUMP/FICHEROS';
```

```
# EXPORT DATA_PUMP_DIR=<VARIABLE_FICHERO_VOLCADO>
```

- 4.- Si es un usuario privilegiado se emplea el valor por defecto en bd del directorio DATA_PUMP_DIR (se crea automáticamente en la creación de la BD o al actualizar el diccionario de datos). No confundir con la variable de entorno.

Para saber el valor asignado a la variable DATA_PUMP_DIR puede usarse la sentencia:

```
SQL> SELECT directory_name, directory_path FROM dbaDirectories  
WHERE directory_name='DATA_PUMP_DIR';
```

Si no se trata de un usuario privilegiado, debe haberse concedido previamente acceso al directorio DATA_PUMP_DIR

DATA PUMP EXPORT.

Se emplea la utilidad expdp, indicando las características de la exportación en la línea de comandos o mediante un fichero de parámetros. Los ficheros resultantes de una exportación sólo pueden importarse con D.P. Import.

Cualquier usuario puede exportar objetos de su propiedad o su esquema al completo.

Los usuarios no privilegiados deben tener permiso WRITE en el objeto directorio y deben indicar éste en el parámetro DIRECTORY o junto al nombre de fichero de volcado.

Modo en línea: expdp HELP=y

Modo interactivo: expdp directory=<localización>

Modo fichero de parámetros: expdp PARFILE=<fichero_parámetros>

Nota: Si no se indica ningún otro parámetro se usan los valores por defecto en la exportación: expdat.dmp y export.log -ficheros-, exportación a nivel de esquema.

DATA PUMP IMPORT.

Se emplea la utilidad impdp, indicando las características de la importación en la línea de comandos o mediante un fichero de parámetros. Sólo pueden usarse en la importación ficheros creados con D.P. Export.

Los usuarios deben tener permiso READ en el objeto directorio donde resida el fichero de volcado y de escritura donde se cree el fichero de registro (log) y los SQL.

Modo en línea: impdp HELP=y

Modo interactivo: impdp directory=<localización>

Modo fichero de parámetros: impdp PARFILE=<fichero_parámetros>

Nota: Si no se indica ningún otro parámetro se usan los valores por defecto en la importación: expdat.dmp e import.log -ficheros-, importación total del fichero.

Parámetros de expdp/impdp:

Algunos de los parámetros básicos son estos:

- CONTENT =<ALL | DATA_ONLY | METADATA_ONLY>
Indica contenido a exportar: datos, metadatos o ambos.
- DIRECTORY =<DATA_PUMP_DIR>
Localización para los ficheros de volcado y registro.
- DUMPFILE = <directorío_objeto:>nombre_fichero
Fichero de volcado. Por defecto expdat.dmp
- ESTIMATE = <BLOCKS | STATISTICS>
Método de estimación de espacio en disco necesario para cada tabla.
- ESTIMATE_ONLY =<N | Y>
Permite estimar el espacio consumido en una exportación sin realizarla.
- EXCLUDE = tipo_objeto <:cláusula>
Filtrar metadatos exportados indicando objetos y tipos excluidos. Por Página 5defecto ninguno. Es excluyente con el parámetro INCLUDE.
- FILESIZE
Tamaño máximo para ficheros de exportación (por defecto ilimitado).
- FULL = <N | Y>
Realizar una exportación de la bd al completo.

- **JOB_NAME**
Especificar un nombre para el trabajo, “job”, de exportación.
- **LOGFILE = <directorio_objeto:>nombre_fichero**
Fichero de registro, por defecto export.log
- **STATUS = nn**
Frecuencia con que muestra el estado del trabajo junto a una descripción de la operación actual (por defecto es cero y no muestra información hasta acabar la exportación de cada objeto).

DATA PUMP.

INFORMACIÓN.

Puede obtenerse información en las siguientes vistas:

–DBA_DATAPUMP_JOBS y USER_DATAPUMP_JOBS. Identifican trabajos Data Pump activos.

–DBA_DATAPUMP_SESSIONS. Identifica sesiones de usuario asociadas a un trabajo.

–V\$SESSION_LONGOPS. Información sobre el progreso del trabajo indicando una estimación de la tasa de transferencia. Columnas relevantes son USERNAME –propietario del trabajo-, OPNAME –nombre de trabajo-, TARGET_DESC –operación-, SOFAR -megabytes (MB) transferidos-, TOTALWORK –número estimado de megabytes (MB)-, UNITS -MB- y MESSAGE.

Resumen de comandos.

Exportar una tabla:

```
[oracle@oraser5 ~]$ expdp usuario/pass dumpfile=Nombre_Fichero.dmp  
tables=usuario.tabla directory=directorio donde se guarda.
```

Exportar esquema de usuario:

```
[oracle@oraser5 ~]$ expdp usuario/pass dumpfile=Nombre_Fichero.dmp  
schemas:usuario directory=directorio donde se guarda.
```

Exportar una base de datos.

```
[oracle@oraser5 ~]$ expdp usuario/pass dumpfile=Nombre_Fichero.dmp  
logfile=nombre.log full=y directory=directorio.
```

Importar una tabla:

```
[oracle@oraserv5 ~]$ impdp usuario/pass dumpfile=Nombre_Fichero_a_recuperar.dmp  
tables=usuario.tabla directory=directorio donde está el fichero.
```

Importar esquema de usuario:

```
[oracle@oraserv5 ~]$ impdp system/root dumpfile=Nombre_Fichero_a_recuperar.dmp  
schemas=nombre_usuario directory=directorio donde está el fichero.
```

Importar una base de datos:

```
[oracle@oraserv5 ~]$ impdp usuario/pass dumpfile=Nombre_Fichero_a_recuperar.dmp  
logfile=nombre.log full=y directory=directorio.
```

Exportar tablespaces

```
[oracle@oraserv5 ~]$ expdp system/password DUMPFILE=nombrefichero.dmp  
DIRECTORY = directorio TRANSPORT_TABLESPACES=nombre_tablespaces TRANSPORT_FULL_CHECK=Y
```

Importar tablespaces

```
[oracle@oraserv5 ~]$ impdp system/password DUMPFILE= nombrefichero.dmp  
DIRECTORY= directorio TRANSPORT_DATAFILES= Ruta de los ficheros de datos.
```

SQL*LOADER

Objetivos: Aprender el manejo de carga de datos de forma masiva optimizando tiempo.

SQL*Loader carga datos de archivos externos en tablas de una base de datos Oracle. Dispone de un potente sistema de análisis de datos que limita muy poco el formato de los datos en el archivo de datos. Los archivos que utiliza SQL*Loader son los siguientes:

Archivos de datos de entrada: SQL*Loader lee datos de uno o más archivos (o equivalentes de archivos del sistema operativo) especificados en el archivo de control. Desde la perspectiva de SQL*Loader, los datos del archivo de datos se organizan en registros. Un archivo de datos concreto puede tener un formato de registro fijo, variable o de flujo. El formato de registro se puede especificar en el archivo de control con el parámetro INFILE. Si no se especifica ningún formato de registro, el valor por defecto es el formato de registro de flujo.

Archivo de control: El archivo de control es un archivo de texto escrito con un lenguaje que SQL*Loader comprende. El archivo de control indica a SQL*Loader dónde encontrar los datos, cómo analizarlos e interpretarlos, dónde insertarlos, etc. Aunque no está definido de forma precisa, se puede decir que un archivo de control tiene tres secciones.

- La primera sección contiene amplia información de sesiones, por ejemplo:
 - opciones globales, como el nombre del archivo de datos de entrada y los registros que se van a saltar
 - Cláusulas INFILE para especificar dónde localizar los datos de entrada
 - Datos que se van a cargar
- La segunda sección se compone de uno o más bloques INTO TABLE. Cada uno de estos bloques contiene información sobre la tabla (como el nombre de tabla y las columnas de la misma) en la que se van a cargar los datos.
- La tercera sección es opcional y, si está presente, contiene datos de entrada.

Archivo log: Cuando SQL*Loader inicia la ejecución, crea un archivo log. Si no puede crear un archivo log, la ejecución termina. El archivo log contiene un resumen detallado de la carga, que incluye una descripción de cualquier error producido durante la carga.

Archivo de errores: El archivo de errores contiene registros que fueron rechazados, por SQL*Loader o por la base de datos Oracle. SQL*Loader rechaza los registros de archivos de datos cuando el formato de entrada no es válido. Una vez que SQL*Loader acepta un registro de archivo de datos para su procesamiento, éste se envía a la base de datos Oracle para su inserción en una tabla como una fila. Si la base de datos Oracle determina que la fila es válida, ésta se inserta en la tabla. Si se determina que la fila no es válida, se rechaza el registro y SQL*Loader lo pone en el archivo de errores.

Archivo de desechos: Este archivo se crea sólo cuando es necesario y si se ha especificado que se active uno. El archivo de desechos contiene registros que se han filtrado de la carga porque no coinciden con ningún criterio de selección de registro especificado en el archivo de control. Desde la línea de sentencias debe ejecutarse:

```
[oracle@oraserv5 ~]$ sqlldr userid=<usuario>/<contraseña>
control=<nombre_fichero_control> log=<nombre_fichero_log>
```

Creamos una tabla de prueba:

```
SQL> CREATE TABLE fcld (
  id_estu integer,
  nombre varchar2(30),
  apellido  varchar2(30));
```

Los datos que queremos introducir están en el fichero estudiantes.txt que tiene solo las tres filas que aparecen bajo los guiones:

```
[oracle@oraserv5 ~]$ vim estudiantes.txt
```

El fichero de control que usamos es fcld.ctl que es el siguiente:

```
[oracle@oraserv5 ~]$ vim fcld.ctl
load data
infile 'estudiantes.txt'
into table fcld
fields terminated by "\t"
( id_estu, nombre, apellido )
```

Cargamos los datos a la tabla fcld.

```
[oracle@oraserv5 ~]$ sqlldr hr/oracle control=fcld.ctl
```

Validamos que la data esta en la tabla.

```
SQL> select * from fcld;
```

🐧 Automatización de Oracle en GNU/Linux🐧

Objetivos: Aprender automatizar procesos y colocar inicio de oracle al encender nuestro server, colocando los backup en procesos programados aprovechando herramientas de GNU/Linux.

CRONTAB

En el sistema operativo Unix, cron es un administrador regular de procesos en segundo plano (demonio) que ejecuta procesos o guiones a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el fichero crontab. El nombre cron viene del griego chronos que significa "tiempo".

Para modificar el crontab de un usuario ejecutamos:

```
[oracle@oraserv5 ~]$ crontab -u usuario -e
```

Esto abrirá el archivo crontab de usuario con nuestro editor por defecto (variable EDITOR), en este archivo existen 6 campos a los cuales le asignamos los siguientes valores:

Campo	Valor
1-)minuto	0-59
2-)hora	0-23
3-)día del mes	1-31
4-)mes	1-12
5-)día de la semana (ej.: sábado)	0-7 (0 y 7 son domingo)
6-)comando	

Ejemplos:

Crear un archivo comprimido de los logs a las 10:24 PM (22:24) todos los martes durante el mes de diciembre:

```
24 22 * 12 2 tar -cjf /backup/logs.tar.bz2 /var/logs
```

Lista las tareas programada del usuario actual.

```
[oracle@oraserv5 ~]$ crontab -l
```

Lista las tareas programada del usuario especificado como root.

```
[root@deivy ~]# crontab -u deivy -l
```

Luego de haber visto el resultado del ultimo cron elimina todas la tarea programada anteriormente.

```
[root@deivy ~]# crontab -r <elimina las tarea programada por el usuario>
```

Verifica que no tenga mas tareas en tu usuario.

```
[root@deivy ~]# crontab -l
```

Nota: Logs de cron se encuentran en /var/log/cron .

Programando copia de Seguridad con cron

Inicio de Backup programado en GNU/Linux en oraserv5:

Scripts: expdp_final.sh

```
#!/bin/bash
export ORACLE_HOME=/opt/app/oracle/product/11.2.0/db_1
export ORACLE_SID=oraserv5
export PATH=$PATH:$ORACLE_HOME/bin
DATE=`date +%Y-%m-%d:%T`
expdp hr/kikla DIRECTORY=acldump DUMPFILE=fullhr.$DATE.dmp
LOGFILE=fullhr.$DATE.log
exit 0
```

Programando Backup:

```
[oracle@oraserv5 ~]$ crontab -e
*/2 * * * * /bin/sh /opt/Backups/scripts/expdp_final.sh
```

Verificando el mismo:

```
[root@oraserv5 log]# tail -f cron
```

Ready:).

Runlevels

Runlevel - nivel de ejecución (al iniciar o arrancar el sistema operativo).

El kernel ejecuta init al arrancar. Este programa, ahora como proceso, cargará los subprocessos necesarios para la puesta en marcha del sistema. Cuando init ha terminado de cargarse vacía el subdirectorio /tmp y lanza getty que se encarga de permitir hacer login en el sistema a los usuarios.

Los niveles de ejecución (Runlevel), definidos en el archivo /etc/inittab (en Ubuntu es /etc/init/rc-sysinit.conf) determinan los servicios que tendremos disponibles en cada uno de ellos. Es una forma de tener diferentes modos de trabajo, cada uno de ellos con características bien definidas, en función del tipo de tarea a que estén orientados.

Existen ocho niveles de ejecución, que están numerados del cero al seis, más otro denominado con la letra "S" (tiene un alias con la letra "s", que realmente es igual a el no 1). Los niveles de ejecución de ejecución que manejaremos y una descripción de para qué están definidos se puede ver en la siguiente tabla:

Nivel de ejecución	Modo
0	Detener el sistema
1	Mono usuario, sin soporte de red, Mantenimiento
2	Multiusuario, sin soporte de red, Poco sistemas lo usan
3	Modo multiusuario completo
4	Sin uso. Recomendado para pruebas
5	Modo X11; Multiusuario completo en entorno gráfico
6	Reiniciar el sistema
s/S	Nivel Usuario único

Cada uno de estos niveles de ejecución tiene un directorio donde se encuentran los servicios o demonios que inician cuando son ejecutados. Un nivel de ejecución es un mecanismo para proporcionar distintas configuraciones del equipo y se implementa mediante el uso de árboles de scripts de inicio, almacenados en directorios llamados /etc/rc1.d, /etc/rc2.d, etc. Esta característica es implementada típicamente de forma muy simple, por lo que usted debe modelar sus entradas en el archivo basándose en otras del mismo, o bien consultar la documentación de su sistema para obtener más información.

Init

Init es el proceso padre de todos los demás procesos, si ejecutamos pstree -np | less podremos ver que el primer procesos que esta arriba y con el PID de 1 es init seguido luego de todo un árbol de procesos que se desprende de el, pero init también es un comando que nos podemos cambiarnos de los runlevels ya antes mencionados, ejemplo:

Para cambiar al runlevel 3 con el comando init lo hacemos de la siguiente manera:

```
[root@oraserv5 ~]# su -c "init 3"
```

Podemos verificar en que runlevel estamos actualmente con:

```
[oracle@oraserv5 ~]$ runlevel
```

Pudiendo luego de esto volver a subir a nuestro runlevel por defecto (generalmente el 5) con

```
[oracle@oraserv5 ~]$ su -c "init 5"
```

Comando service

Con el comando service podemos iniciar, parar, reiniciar y ver el estados de determinados demonios, ejemplos:

Para ver iniciar el demonio de ssh podemos hacerlo de la siguiente forma:

```
[oracle@oraserv5 ~]$ su -c 'service sshd start'
```

Para detenerlo podemos hacer:

```
[oracle@oraserv5 ~]$ su -c 'service sshd stop'
```

Para ver su estatus podemos hacer:

```
[oracle@oraserv5 ~]$ su -c 'service sshd status'
```

Y por ultimo para reiniciarlo podemos hacer:

```
[oracle@oraserv5 ~]$ su -c 'reboot'
```

chkconfig

El comando chkconfig se usar para podemos indicarle a los demonios en que runlevels deben correr y en cuales no.

Hay herramientas que hacen esto con ventanas ncurses y puede resultar mas fácil, una de ellas es “**ntsysv**”. Te permite hacer esto de una forma “gráfica” en el shell, aunque no puedes elegir los runlevels que deseas, solamente activar y desactivar.

Ejemplos:

Para ver una lista de todos los demonios y de cuales runlevels tienen habilitados podemos ejecutar:

```
[oracle@oraserv5 ~]$ chkconfig --list
```

Donde cada demonio es una linea y cada demonio tiene sus runlevels enumerados del 0 a 6 y cada numero dice al lado si esta **on** (encendido, si va a correr en ese runlevel) o si esta **off** (apagado, si no va a correr en ese runlevel)

Ejemplo, si queremos hacer que ssh corra en los runlevels 2, 3, 4, 5 (chkconfig hace que los demonios corran por defecto en estos runlevels) podemos hacerlo de la siguiente manera.

Cambiando los servicio de runlevel.

```
[root@oraserv5 ~]# chkconfig [--list] [--type type][name]
```

```
[root@oraserv5 ~]# chkconfig --add name
```

```
[root@oraserv5 ~]# chkconfig --del name
```

```
[root@oraserv5 ~]# chkconfig --override name
```

```
[root@oraserv5 ~]# chkconfig [--level levels] [--type type] name <on|off|reset|resetpriorities>
```

```
[root@oraserv5 ~]# chkconfig [--level levels] [--type type] name
```

Iniciando Automáticamente la Instancia y Procesos de Oracle

Necesitamos que nuestra instancia de oracle arranque automáticamente cuando se reinicia o arranca el equipo.

Los siguientes pasos deben de funcionar tanto para oracle 10g como para 11g.

La ubicación de algunos archivos podría diferir dependiendo de la instalación (esencialmente el init.d o los runlevels).

Debemos hacer lo siguiente en como ROOT:

1) Editar el archivo el archivo /etc/oratab

En dicho archivo tendremos que poner algo similar a esto =

```
oraserv5:/opt/app/oracle/product/11.2.0/db_1:Y
```

Su sintaxis es:

```
[instancia]:[$ORACLE_HOME]:[Y o N]
```

La N indica que no queremos que se arranque automáticamente. La Y indica que queremos que se arranque automáticamente.

Trabajando con los scripts dbstart y ddshut

Si no esta creado el archivo /etc/init.d/oracle como ROOT (desde consola podemos ejecutar cat > /etc/init.d/oracleserv5) lo creamos e introducimos este script, si lo estuviera revisaremos su configuración:

```
#!/bin/bash
### BEGIN INIT INFO
# Provides:          oracleserv5
# Required-Start:
# Required-Stop:
# Default-Start:    3 4 5
# Default-Stop:     0 1 2 6
# Short-Description: Startup script for Oracle Databases
# Description:       Oracle Database Auto-Start Script
### END INIT INFO
export ORACLE_HOME=/opt/app/oracle/product/11.2.0/db_1
export ORACLE_SID=oraserv5
export PATH=$ORACLE_HOME/bin:$PATH:
        case "$1" in
start)
        echo -n "Starting Oracle: "
        su oracle -c "$ORACLE_HOME/bin/dbstart"
        su oracle -c "$ORACLE_HOME/bin/emctl start dbconsole"
        su oracle -c "$ORACLE_HOME/bin/lsnrctl start"
        touch /var/lock/oracle
        echo "OK"
        ;;
stop)
        echo -n "Shutting down Oracle: "
        su oracle -c "$ORACLE_HOME/bin/emctl stop dbconsole"
        su oracle -c "$ORACLE_HOME/bin/lsnrctl stop"
        su oracle -c "$ORACLE_HOME/bin/dbshut"
        rm -f /var/lock/oracle
        echo "OK"
        ;;
restart)
        echo -n "Shutting down Oracle: "
        su oracle -c "$ORACLE_HOME/bin/emctl stop dbconsole"
        su oracle -c "$ORACLE_HOME/bin/dbshut"
        rm -f /var/lock/oracle
        echo "OK"
        su oracle -c "$ORACLE_HOME/bin/dbstart"
        su oracle -c "$ORACLE_HOME/bin/emctl start dbconsole"
        touch /var/lock/oracle
        echo "OK"
        ;;
esac
exit 0
```

Agregamos el scripts a chkconfig

```
[root@oraserv5 scripts]# chkconfig --add oracleserv5
```

Damos los permisos adecuados al fichero:

```
[root@oraserv5 scripts]# chmod 755 /etc/init.d/oracleserv5
```

Añadimos el script a los niveles de ejecución 345:

```
[oracle@oraserv5 ~]$ chkconfig --level 3 oracleserv5
```

Si usamos la versión 10g Release 2 , tendremos el siguiente mensaje de error:
Failed to auto-start Oracle Net Listener using

/ade/vikrkuma_new/oracle/bin/tnslsnr

Tenemos que editar el archivo \$ORACLE_HOME/bin/dbstart y poner.
ORACLE_HOME_LISTNER=\$ORACLE_HOME

Ahora automáticamente debería arrancarse oracle al reiniciar el sistema.

Si queremos arrancarla o pararla manualmente podemos usar el comando:

```
[oracle@oraserv5 ~]$ service oracleserv5 start
```

o para pararla con

/etc/init.d/ oracleserv5 stop

```
[oracle@oraserv5 ~]$ service oracleserv5 stop
```

Instalación de Clientes

Objetivos: Aprender la instalación y conexión de clientes en Oracle.

Paquetes Necesarios

```
linux_11gR2_client.zip
```

1. Creamos Usuario, Grupos y Directorios para el cliente.

```
[root@oraserv5 ~]# groupadd oinstall
```

```
[root@oraserv5 ~]# groupadd dba
```

```
[root@oraserv5 ~]# useradd -u 526 -g oinstall -G dba oracle
```

```
[root@oraserv5 ~]# mkdir -p /opt/app/oracle/product/11.2.0/db_1
```

```
[root@oraserv5 ~]# chown -R oracle:oinstall /opt/app
```

2. Copiamos las variables en el archivo .bashrc o .bash_profile:

```
# Configuración de ORACLE
ORACLE_BASE=/opt/app/oracle; export ORACLE_BASE
ORACLE_HOME=$ORACLE_BASE/product/11.2.0/db_1; export ORACLE_HOME
LD_LIBRARY_PATH=/usr/lib/oracle/11.2.0.1/client/lib; export LD_LIBRARY_PATH
TNS_ADMIN=/opt/app/oracle/product/11.2.0/db_1/network/admin; export TNS_ADMIN
```

3. Actualizamos el profiles del usuario.

```
[root@oraserv5 ~]# source /home/oracle/.bashrc &&
source /home/oracle/.bash_profiles
```

4. Descargamos el paquete que tiene el cliente.

```
[root@oraserv5 ~]# su - oracle
```

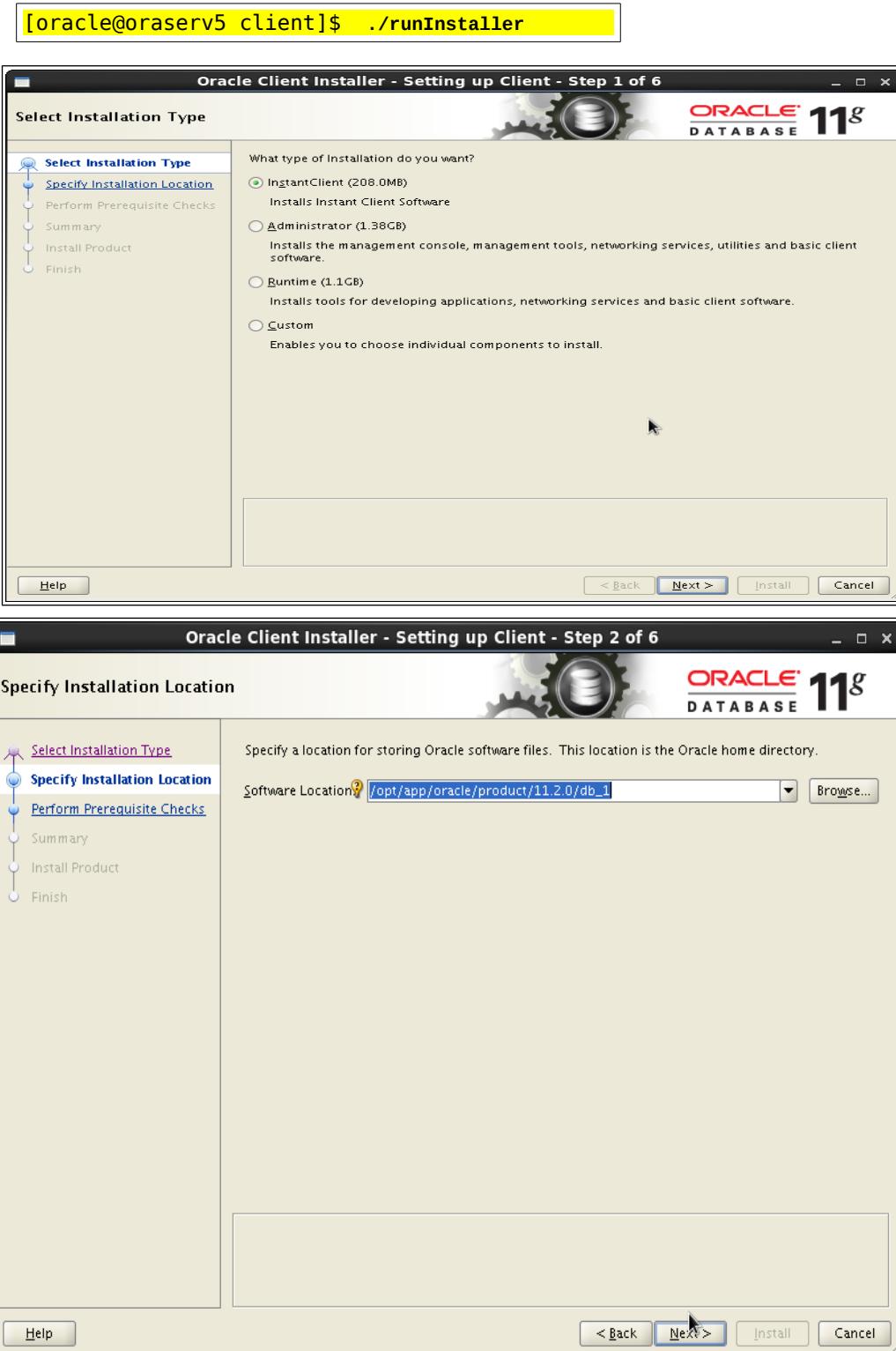
URL Descarga:

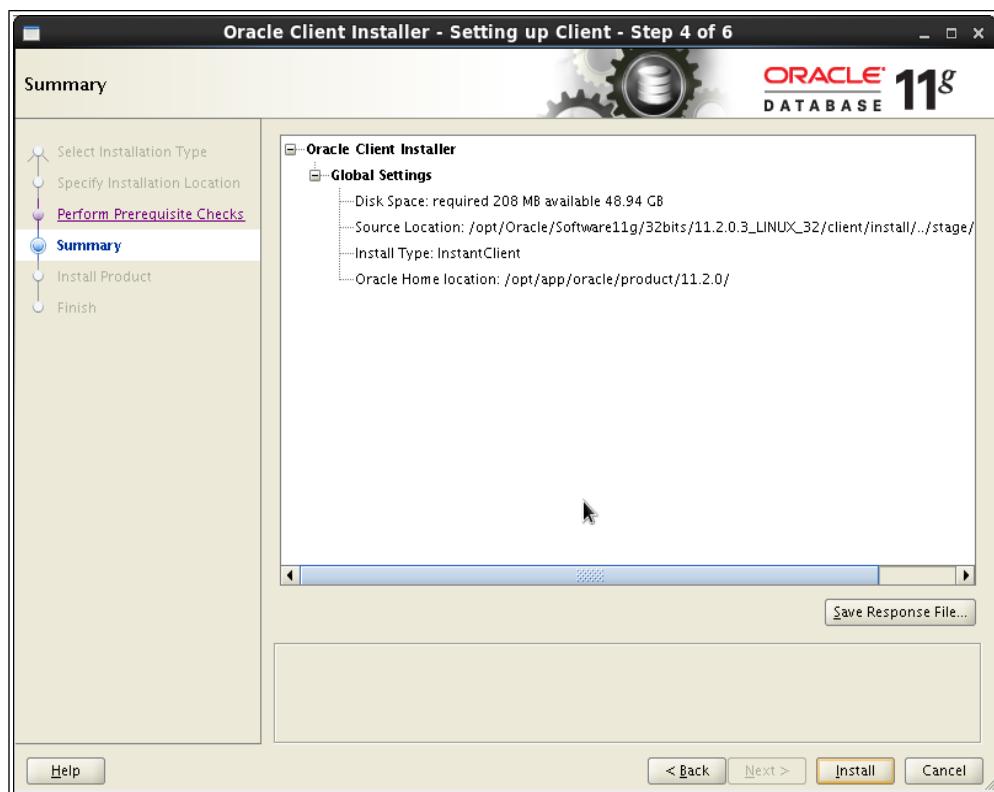
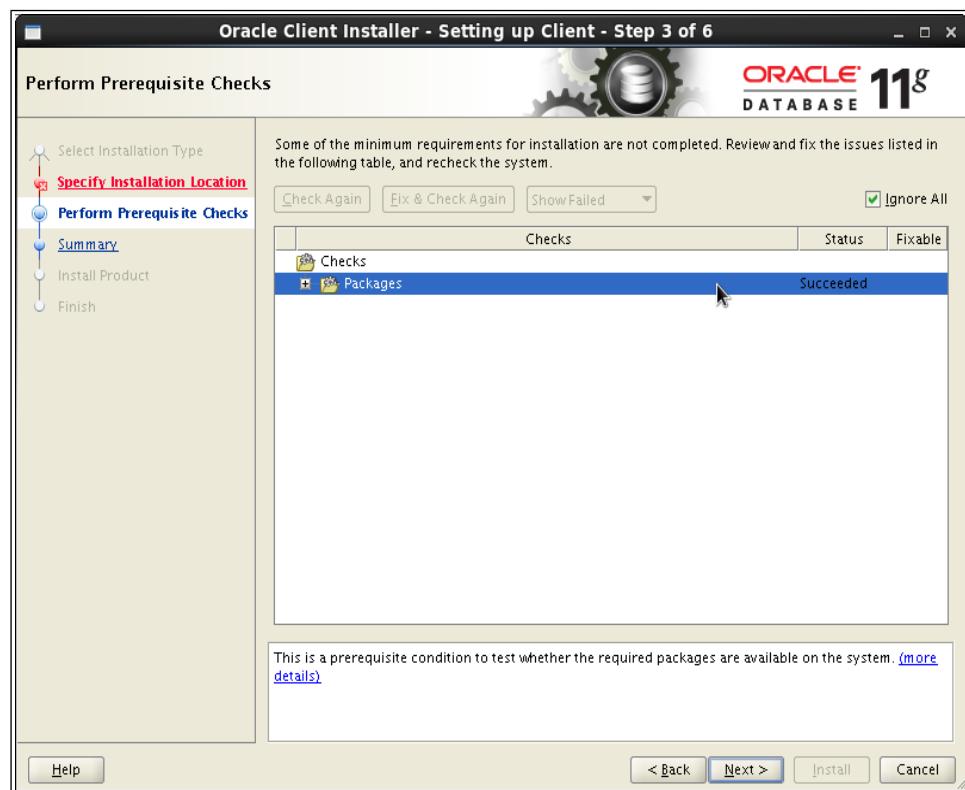
<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/112010-linuxsoft-085393.html>

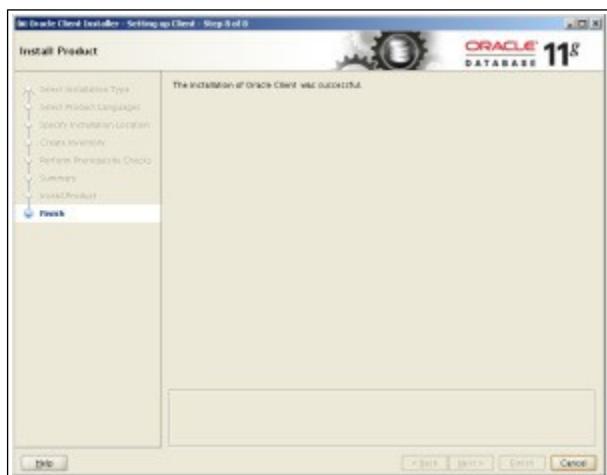
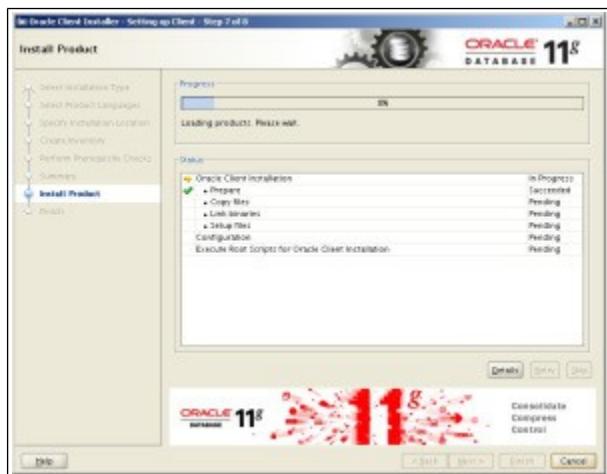
5. Descomprimir el paquetes del los cliente.

```
[oracle@oraserv5 ~]$ unzip linux_11gR2_client.zip
```

6. Instalación del cliente.







7. Copiamos las librerías de /usr/lib/oracle/11.2.0/client/lib a /usr/lib/ (Si es necesario).

```
[root@oraserv5 ~]# cp /usr/lib/oracle/11.2.0/client/lib /usr/lib
```

8. Agregamos la ruta /etc/profile.

```
PATH=$PATH:/usr/bin ; export PATH
```

9. Creamos las carpetas network/admin.

```
[root@oraserv5 ~]# mkdir -p /usr/lib/oracle/11.2.0/client/network/admin
```

10. creamos el tnsname.ora.

```
[root@oraserv5 ~]# vim /usr/lib/oracle/11.2.0/client/network/admin  
ORASERV5 =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.100.178 )(PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = oraserv5.fcl.d.acl)  
    )  
  )
```

11. Probamos la conexión con el sqlplus desde el cliente.

```
[oracle@oraserv5 ~]$ sqlplus hr/kikla@oraserv5
```

```
SQL> select * from dept;
```

Consejo para un GNU/Dba

Objetivos: Conocer las mejores practicas para un ambiente producción.

DBA debe: Instalar y configurar el Oracle software, seleccionar el hardware, manejar la base de datos y objetos de ella, establecer backups y tuning.

Tareas como el monitoreo y rotación de logs pueden hacerse desde el Enterprise Manager, Grid Control, scripts shell de Unix, DBMS_SCHEDULER, Perl, herramientas de base de datos de terceros o combinaciones de algunos de estos.

Se debería tener el `flash_recovery_area` en otro disco para tener mayor seguridad y performance.

Para aplicaciones transaccionales se debería setear el `db_block_size` a 8 kb y para data warehouse a 16 kb a más.

Limpiar periódicamente el archivo `alert_log` puede llenar disco duro.

Tener los índices en diferentes tablespace que las tablas y cada tablespace (sus datafiles) en diferente discos duros.

A veces es recomendable usar tablespace bigfile porque reduce el tiempo de actualización de cabeceras en un checkpoint al existir menos datafiles.

Para crear un Temporary Tablespace es mejor primero crear un tablespace normal, luego eliminarlo (`drop tablespace nombre`) y luego crear el Temporary Tablespace pero usando `reuse`, ejemplo: `create temporary tablespace nombre tempfile 'ruta' size # reuse`. Con la finalidad que se guarde un espacio desde que se crea el tempfile ya que un Temporary tablespace por default se reserva la primera vez que se hace una operación de ordenamiento y no cuando se crea, haciéndolo la primera vez un proceso lento.

Si nosotros deseamos borrar un datafile que fue agregado por equivocación a un tablespace y este datafile aun no se le ha agregado objetos, la mejor manera de eliminarlo es redimensionar ese datafile a un tamaño menor a 5 bloques de Oracle. Con esto hacemos que nunca sea utilizado para almacenar objetos y luego podemos excluirlo de la base de datos con el tiempo.

Para hacer un backup de un tablespace: `alter tablespace nombre begin backup` copiar los datafiles. `alter tablespace nombre end backup` Al tenerlo en `begin backup` permite que se siga copiando data en los datafiles pero se actualizará las cabeceras del datafile.

Si la conexión de red está lento puede fallar la conexión. El error es ORA-12535 y ORA-12547, hay que modificar el parámetro INBOUND_CONNECT_TIMEOUT. Revisar si el listener esta generando trace porque podría esto llenar el disco.

No se debería hacer tareas como reconstrucción de índices, tablas, estadísticas, carga de data en un Shared Server.

Se debería tener load balancing en el listener.ora si estamos en Shared Server, lo cual permitiría que el listener haga decisiones acerca de que dispatcher esté menos ocupado y enviarlo al cliente.

Se debería usar v\$session e ir viendo periódicamente el número de conexiones para determinar un correcto número de dispatchers.

Para aplicaciones web utilizar connection pooling

Se debería usar Shared Server cuando se utiliza más de 200 usuarios.

Si creamos un script administrativo donde no debemos colocar nuestro password, la mejor alternativa es usar una cuenta externa .

Nunca se debería dar permisos de un alter system a un usuario externo.

Setear la variable remote_os_authent a false.

Todas las cuentas que se crean en el DBCA deberían estar bloqueadas.

Debemos tener cuidado a los usuarios que tengan los privilegios create any job o procedure y además alter user porque ellos pueden correr programas con privilegios de otros usuarios.

Mover la tabla sys.aud\$ a otro tablespace.

Auditar la tabla sys.aud\$.

Para conseguir una mejor performance en las aplicaciones debemos tener seteado lo siguiente:

```
PLSQL_CODE_TYPE = NATIVE.  
PLSQL_DEBUG = FALSE.  
PLSQL_OPTIMIZE_MODE = 2.  
PLSQL_WARNING=DISABLE:ALL.
```

Los datafiles se deberían tener en autoextend hasta un tiempo y una vez que ya es constante el tamaño se le debería quitar la opción de autoextend.

No se debe usar memoria SWAP en el SGA.

No se debería ir más allá del 60% de la memoria RAM.

Al ingresar al Database Control hay un link llamado Resumen de Diagnóstico donde este tiene un link llamado Busquedas de Rendimiento (Performance Findings), ahí nosotros podemos dar click y nos mostrará los problemas de performance encontrados para afinar.

En el Database Control hay un tab llamado Rendimiento donde se encuentra una tabla con todos los aspectos que se pueden analizar información importante.

Debemos ubicar los objetos que están en estado INVALID ó UNUSABLE en la base de datos para corregirlos.

```
SQL> select owner, object_name, object_type from dba_objects  
where status = 'INVALID' or 'UNUSABLE';
```

Los logfiles deberían tener el mismo tamaño como recomendación.

Tener cada miembro de los redolog en disco separados, si en caso tenemos un redolog y un datafile en el mismo disco no debemos poner el UNDOTBS, SYSAUX, SYSTEM y algun tablespace asignado para índices en el mismo disco.

No debemos tener mas del 90% del tamaño del redo log más pequeño sin hacer un checkpoint.

Oracle recomienda tener los siguientes parámetros deshabilitados:

LOG_CHECKPOINT_TIMEOUT, indica la cantidad de segundos que un nuevo bloque modificado o nuevo en el buffer cache espera para ser escrito en disco.

LOG_CHECKPOINT_INTERVAL.

FAST_START_IO_TARGET, indica lo mismo que el FAST_START_MTTR pero el número indica el número de I/O en lugar de segundos para finalizar un instance recovery.

Priorizando actividades-diarias, semanales, mensuales, trimestrales o anuales

Veamos cuales son las actividades prioritarias que se necesitan cubrir. La calendarización dependerá del lugar de trabajo, necesidades de la aplicación y el peso que tengan las actividades en general.

Diarias

Respaldos – usualmente son incrementales o acumulativas, uno completo por semana y los logs son almacenados y enviados por correo al DBA en caso de fallas.

Alert Log de la base de datos – tales como errores ORA, notificaciones automáticas por correo, mensajes a localizadores.

ADRCI – Automatic Repository Utility and Log Rotation.

Espacio en el filesystem, CPU y estadísticas de I/O – requiere de apoyo del administrador del SO.

Instrucciones SQL – Sentencias que están entre el top 5 y 10.

Corrupción – logs de RMAN, logs de exports y/o datapump, dbverify, v\$database_block_corruption.

Crecimiento de tablespaces – crecerlos, administración de particiones, tablespaces temporales, undo.

Data Guard – revisar en los logs que la aplicación/transporte esté en sincronía.

Logs del listener SQL*NET – detección de intrusos.

Auditoría de logs y evidencias – detección de intrusos, quitar cuentas sin uso.

Dumps del core y usuario – espacio que ocupan, bugs de Oracle.

Creación de nuevas cuentas – debería ser al menos automatizado parcialmente.

Actualizar al usuario sobre la seguridad – al menos con 24 horas de anticipación.

Migrar esquemas y cambios al código o actualizaciones específicas de SQL.

Crecimiento de las tablas grandes, crecimiento uniforme del tablespace.

Llevar un registro de los cambios diarios a la base de datos – publicarlos para ciertas personas del staff de IT.

Semanales

Respaldos

usualmente de toda la base de datos.

Clonaciones para bases de datos no productivas – automatizado o con scripts.

Crecimiento de tablespaces – el diario acumulado en uno semanal.

Mejora de versión de Oracle o proyectos de migración para aplicar parches – actualizaciones significativas.

Pruebas en sitio de Data Guard.

Revisar actualizaciones de My Oracle Support(MOS) – nuevos parches, actualizaciones o nuevas versiones.

actualizaciones en la intranet local sobre procedimientos operacionales.

Mensuales

Clonaciones para bases de datos no productivas – automatizado o con scripts.

Monitoreo del crecimiento de tablespaces – el semanal acumulado en uno mensual.

Tendencias y previsiones – consumo de CPU, estadísticas de I/O, accesos

Cambio de passwords en producción – sys, system, wallet, schema, grid control, OAS.

Licenciamiento de uso de Oracle y cuanto abarca.

Poner en práctica escenarios de recuperación.

Trimestrales

Aplicación de CPUs(Critical Patch Update) y PSUs(Patch Set Updates) en producción con planeación de la suspensión del servicio. Aplicación de CPUs, PSUs una sola vez en instancias no productivas.

Monitoreo del crecimiento de tablespaces – el mensual acumulado en un anual

Cursos de actualización de Oracle – de Oracle University(en línea o presenciales), libros, encuentros informales.

Acumulación de tendencias y previsiones.

Anuales

Crecimiento de tablespaces – reporte anual.

Suma de tendencias y previsiones.

Ir a conferencias sobre Oracle – grupos de usuarios locales o nacionales.

actualizaciones de Oracle con suspensión de servicio planeado – versión + parches + PSUs + aplicación una sola vez.

Licenciamiento de software y renovación de servicios.

Evaluación y actualización de hardware.

Renovación de certificados SSL, Oracle Wallets.

Así es, se ve como un conjunto desolador de actividades qué realizar, sin embargo, se cuenta con el apoyo de herramientas tales como el OEM, Grid Control, monitoreo de terceros o scripts hechos a medida. Es por eso que se hace la reiteración de que la automatización de estas tareas es de gran importancia.

¿Cuál sería el procedimiento para aplicar un parche de una base Oracle?

Procedimiento :

Bajar el parche/patchset

Verificar los requerimientos del parchado: versiones de oui, opatch, rapidclone.

Verificar los requerimientos del sistema: parches del sistema operativo y pre-requisitos del sistema operativo.

Ejecutar tareas de preinstalación: verificar usuarios a usar, claves, sistemas impactados, riesgos, pasos a ejecutar para RAC, bajar instancia y listener, respaldar la base de datos, requisitos para los parámetros del kernel.

Chequear el contexto del sistema: nodos del RAC impactado, nodos de la base Standby o Dataguard, Options instaladas en la base, aplicaciones impactadas, en general se debe hacer un plan de pruebas como resultado de este paso.

Aplicación del parche: configurar el ambiente y la variable DISPLAY, ejecutar OUI (patchset) y aplicar el parche .

Ejecutar scripts de post-aplicación del parche. Tareas de post-instalación.

LVM (Manejo de volúmenes lógicos)

Objetivos: Aprendiendo el valioso concepto de RAID pero a nivel de particiones.

LVM es una implementación de un administrador de volúmenes lógicos para el kernel Linux. Se escribió originalmente en 1998 por Heinz Mauelshagen, que se basó en el administrador de volúmenes de Veritas usado en sistemas HP-UX.

LVM incluye muchas de las características que se esperan de un administrador de volúmenes, incluyendo:

- Redimensionado de grupos lógicos
- Redimensionado de volúmenes lógicos
- Instantáneas de sólo lectura (LVM2 ofrece lectura y escritura)
- RAID0 de volúmenes lógicos.

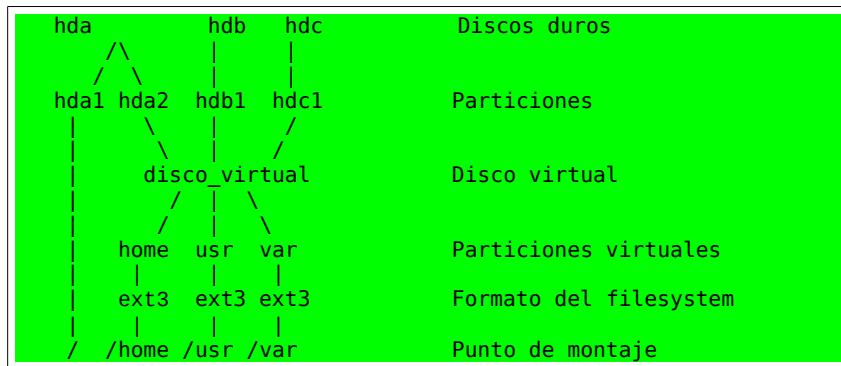
LVM no implementa RAID1 o RAID5, por lo que se recomienda usar software específico de RAID para estas operaciones, teniendo las LV por encima del RAID .

En la imagen de la derecha podemos observar cómo trabaja LVM. Los volúmenes lógicos agrupan particiones físicas de disco, y éstos a su vez, aunque no está representado en la figura, se engloban en un grupo lógico. De esta forma, /home se compone de hda3, hda4 y hdb3, y a su vez, /usr engloba a hda1, hda2, hdb1 y hdb2.

Es muy habitual que alguna partición (normalmente la más importante) tenga poco espacio mientras que otras tienen mucho espacio libre. Para evitar esta situación podemos usar volúmenes lógicos LVM (Logical Volume Manager). Veamos en qué consiste:

- combinamos varias particiones (o discos o dispositivos RAID) para formar un disco virtual o grupo de volúmenes.
- el disco virtual lo dividimos en particiones virtuales o volúmenes lógicos.
- esto es transparente al usuario: sólo veremos los volúmenes lógicos.
- la ventaja es que redimensionar los volúmenes lógicos o añadir/quitar un disco duro al disco virtual es razonablemente sencillo.

➤ **Gráficamente:**



LVM2: administración de volúmenes lógicos en Linux

LVM2 es un administrador de volúmenes lógicos desarrollado para el kernel de Linux, compatible con su predecesor LVM1. En la actualidad, LVM está disponible en la mayoría de los sistemas Linux para utilizarlo al momento de la instalación. De hecho, sistemas como Fedora utilizan LVM si se los partitiona como lo sugiere el instalador por defecto.

La primer ventaja fundamental de LVM es que nos quita el inconveniente de dimensionar exactamente las particiones tal cuál las necesitaremos, encontrándonos luego con que el esquema de particionamiento escogido no es el más adecuado. Este caso es muy fácil verlo con un ejemplo.

Se tiene un disco de 40G (pequeño para los tamaños actuales, lo reconozco, pero sirve para el ejemplo). Se desea instalar Linux allí entonces se decide particionarlo de la siguiente manera:

/boot	200M
/	4G
swap	1G
/home	34G

El problema al que nos enfrentamos en este caso (me ha ocurrido en alguna oportunidad) es quedarnos sin espacio en alguna partición y tener lugar de sobra en otra. Por ejemplo, supongamos que se llena el disco raíz y sin embargo tenemos aún 14G libres en /home. Eso representa una situación donde se hace un uso ineficiente del espacio en disco, además de un problema. La solución más obvia en este caso es hacer algún link simbólico apuntando a algún lugar de /home, pero es una solución bastante mala.

Con LVM, la solución a este problema es trivial, dado que se podría simplemente achicar la partición que contiene /home y aumentar luego el espacio asignado al directorio raíz.

Características de LVM2

LVM2 cuenta, básicamente, con las siguientes funcionalidades.

Redimensión de grupos de volúmenes y volúmenes lógicos en línea.

Crear instantáneas (snapshots) de lectura/escritura del sistema de archivos.

Constituir los volúmenes lógicos separados en los diferentes volúmenes físicos, de manera similar que RAID 0.

Mover los volúmenes lógicos entre los diferentes volúmenes físicos.

Conceptos básicos de LVM

Para entender cómo funciona LVM es necesario conocer algunos conceptos elementales, que son:

Volumen físico (PV): un PV es un disco rígido, una partición o un RAID.

Volumen lógico (LV): un LV es el equivalente a una partición tradicional.

Grupo de volúmenes (VG): un grupo de volúmenes reúne uno o más PVs. Los PVs pueden comenzar a utilizarse en LVM recién cuando pasan a formar parte de un VG.

Physical extent (PE): un PE es una porción de cada volumen físico, de tamaño fijo. Un volumen físico se divide en múltiples PEs del mismo tamaño.

Logical extent (LE): un LE es una porción de cada volumen lógico, de tamaño fijo. Un volumen lógico se divide en múltiples LEs del mismo tamaño.

Device mapper: es un framework genérico del kernel de Linux que permite realizar un mapeo de un dispositivo de bloques a otro. Es la herramienta fundamental en la que se basa LVM para hacer el mapeo de los dispositivos virtuales con los dispositivos físicos.

Ventajas de usar LVM en un sistema pequeño

- Una de las decisiones que afronta un usuario instalando GNU/Linux es cómo particionar el disco duro. La necesidad de estimar cuánto espacio será necesario para el sistema, para los temporales o para los datos personales, puede convertirse en algo problemático, por lo que muchos usuarios optan por crear una partición que ocupe todo el disco y allí introducir los datos. Aún habiendo estimado correctamente cuánto espacio se necesita para /home, /usr, /tmp, o cualquier otro directorio importante, es bastante común que nos quedemos sin espacio en estas particiones, cuando tenemos espacio de sobra en alguna otra.
- Con el uso de un administrador de volúmenes lógicos, el disco completo puede ser asignado a un único grupo lógico y definir distintos volúmenes lógicos para almacenar /home u otros directorios. En el caso que nos quedemos sin espacio, por ejemplo, en /home, y tenemos espacio en /opt, podríamos redimensionar /home y /opt y usar el espacio que le hemos quitado a /opt y añadírselo a /home. Hay que tener en cuenta, que para realizar esto, nuestro sistema de ficheros debe soportar el redimensionado por arriba y por abajo, como ReiserFS.
- Otra alternativa sería dejar una cierta cantidad de espacio del disco sin asignar y cuando fuera necesario, expandir el volumen.

Ventajas de usar LVM en un sistema grande

- Administrar un sistema con muchos discos es un trabajo que consume tiempo, y se hace particularmente complejo si el sistema contiene discos de distintos tamaños. Equilibrar los requerimientos de almacenamiento de distintos usuarios (a menudo conflictivos) puede ser una tarea muy laboriosa y compleja.
- Los grupos de usuarios (llamémosles por ejemplo administración, desarrollo, etc.) pueden tener sus volúmenes lógicos y éstos pueden crecer lo que sea necesario, y el administrador puede realizar las operaciones oportunas sobre dichos volúmenes.
- Cuando un nuevo disco se añade al sistema, no es necesario mover los datos de los usuarios. Simplemente se añade el nuevo disco al grupo lógico correspondiente y se expanden los volúmenes lógicos todo lo que se considere adecuado. También se pueden migrar los datos de discos antiguos a otros nuevos, de forma totalmente transparente al usuario.

Ejemplos

Ya que podemos crear un PV sobre un dispositivo cualquiera de bloques ya bien sean discos USB, raids, discos internos, cintas, etc., en esta serie de ejemplos nos centraremos en dispositivos del tipo /dev/sdX, y bajo la versión 2 de LVM.

No entraremos en la sintaxis de los comandos, ya que la ayuda que proporciona el man es muy extensa y completa.

Inicialización del dispositivo lógico

Primero borramos la tabla de particiones e inicializamos las particiones de disco.

```
[root@oraserv5 ~]# dd if=/dev/zero of=/dev/sda bs=1k count=1
```

```
[root@oraserv5 ~]# dd if=/dev/zero of=/dev/sdb bs=1k count=1
```

```
[root@oraserv5 ~]# pvcreate /dev/sda
```

```
[root@oraserv5 ~]# pvcreate /dev/sdb
```

En este paso, tenemos creados dos volúmenes físicos (PVs).

Creación de la VG

Creamos el grupo lógico (VG) de nombre vg_acl que contendrá las dos particiones que hemos creado anteriormente:

```
[root@oraserv5 ~]# vgcreate vg_acl /dev/sda /dev/sdb
```

Creación del LV

Ahora crearemos un volumen lógico que llamaremos lv1_acl:

```
[root@oraserv5 ~]# lvcreate -L1G -n lv1_acl vg_acl
```

```
lvcreate -- doing automatic backup of "vg_acl"
lvcreate -- logical volume "/dev/vg_acl/lv1_acl" successfully created
```

Creación del sistema de ficheros

En este apartado, debemos crear el sistema de ficheros sobre el LV que acabamos de definir. En este ejemplo, crearemos un sistema de ficheros JFS:

```
[jp@oraserv5 ~]$ mkfs.jfs /dev/vg_acl/lv1_acl
```

```
[jp@oraserv5 ~]$ mount -t jfs /dev/vg_acl/lv1_acl /mnt/acl
```

Recuerde agregar la entrada en /etc/fstab, de modo que al reiniciar, /mnt/acl vuelva a estar disponible.

Conclusión

LVM es un sistema muy interesante para utilizar ya sea en sistemas pequeños como en sistemas con muchos discos y esquemas complejos de particionamiento. Por su flexibilidad y sus capacidades puede reducir mucho el trabajo de mantenimiento de los equipos y cualquier cambio a nivel de almacenamiento.

En un próximo post explicaré cómo trabajar con LVM, desde la creación de los PVs, VGs y LVs, cómo redimensionarlos y eliminarlos e, incluso, cómo trabajar con las instantáneas.

ASM (Automatic Storage Management)

Objetivos: Es simplificar la administración de los archivos de base de datos como datafile, control file, spfile , log file y archive log.

¿Qué es Gestión Automática de Almacenamiento?

ASM proporciona una integración vertical del sistema de archivos y del gestor de volúmenes creado de forma específica para los archivos de las bases de datos Oracle. Además de gestionar máquinas de multiproceso simétrico (SMP) individuales, ASM soporta Oracle Real Application Clusters (RAC) y, por lo tanto, permite gestionar varios nodos de un cluster.

ASM distribuye la carga de entrada/salida (E/S) entre todos los recursos disponibles para optimizar el rendimiento mientras elimina la necesidad de ajuste manual de E/S. ASM ayuda a los DBA a gestionar los entornos de base de datos dinámicos ya que les permite aumentar el tamaño de las bases de datos sin tener que cerrarlas a la hora de ajustar la asignación de almacenamiento.

ASM puede mantener copias redundantes de los datos para ofrecer tolerancia ante fallos o se puede montar en mecanismos de almacenamiento suministrados por el proveedor. La gestión de datos se realiza seleccionando las características de fiabilidad y rendimiento deseadas para las clases de datos en lugar de hacerlo archivo por archivo con la intervención del usuario.

Las capacidades de ASM ahorran tiempo a los DBA al automatizar el almacenamiento manual y, en consecuencia, les permiten aumentar su capacidad para gestionar bases de datos más grandes (y en mayor número) con mayor eficiencia.

Automatic Storage Management distribuye los datos de bases de datos por todos los discos, crea y mantiene un grid de almacenamiento y proporciona el rendimiento de entrada/salida (E/S) más elevado con los mínimos costos de gestión. Conforme se agregan o borran discos, ASM vuelve a distribuir los datos de manera automática. (No es necesario disponer de un gestor de volúmenes lógicos para gestionar el sistema de archivos.) La duplicación opcional aumenta la disponibilidad de los datos y es posible agregar o borrar discos en línea.

ASM ($>=10g$). Gestor de volúmenes para bases de datos Oracle. Mejor rendimiento de E/S y fácil de gestionar. Maneja el espacio en forma de grupos de discos. Divide cada fichero en extensiones (de 128K o 1M) y las reparte entre los discos de un grupo (striping). Tb permite mirror, y lo hace a nivel de fichero (más granular q a nivel de disco), gestionando el mirror a nivel de extensión. Varias opciones de mirror (a nivel de grupos de discos): 2-way mirroring (1 copia por extensión), 3-way (2 copias) y unprotected (sin mirror). opciones de striping: fine (128Kb) y coarse (1M). Se puede desactivar mirror y/o striping.

ASM: Funciones Clave y Ventajas

ASM divide los archivos en extensiones (diferentes de las extensiones de archivos de datos descritas anteriormente) y las distribuye de manera uniforme entre todos los discos. Utiliza una técnica de índice para realizar el seguimiento de la colocación de cada extensión. Cuando varía la capacidad de almacenamiento, ASM no vuelve a segmentar todos los datos, sino que mueve una cantidad de datos proporcional a la cantidad de almacenamiento agregada o eliminada para redistribuir uniformemente los archivos y mantener una carga equilibrada en los discos. Todo ello se hace mientras la base de datos permanece activa.

Puede aumentar la velocidad de una operación de nuevo equilibrio para hacer que termine antes o reducirla para disminuir el impacto en el subsistema de E/S. ASM proporciona protección de duplicación sin necesidad de adquirir un gestor de volúmenes lógicos de otro fabricante. Una ventaja exclusiva de ASM radica en que la duplicación se aplica por archivo, no por volumen. Por lo tanto, un mismo grupo de discos puede contener una combinación de archivos duplicados y no duplicados.

ASM soporta archivos de datos, archivos log, archivos de control, archive logs, juegos de copia de seguridad de Recovery Manager (RMAN) y otros tipos de archivos de base de datos Oracle. ASM soporta RAC y elimina la necesidad de un gestor de volúmenes lógicos de cluster o de un sistema de archivos de cluster.

ASM: Conceptos

ASM no elimina ninguna de las funciones existentes de la base de datos. Las bases de datos existentes siguen funcionando como siempre. Puede crear nuevos archivos como archivos de ASM y dejar que los archivos existentes se administren del modo anterior, o bien puede migrarlos a ASM.

En el diagrama de la diapositiva se representan las relaciones que existen entre los diversos componentes de almacenamiento dentro de una base de datos Oracle que utiliza ASM. Las partes izquierda y central del diagrama muestran las relaciones que existen en versiones anteriores. En la derecha se encuentran los nuevos conceptos introducidos con ASM.

Los archivos de base de datos se pueden almacenar como archivos de ASM. La cabeza de la nueva jerarquía la ocupan los grupos de discos de ASM. Todos los archivos de ASM están en un único grupo de discos. Por el contrario, un grupo de discos puede contener archivos que pertenezcan a varias bases de datos y una única base de datos puede utilizar almacenamiento de varios grupos de discos. Como puede observar, un grupo de discos está formado por varios discos de ASM y cada disco de ASM pertenece a un único grupo de discos.

Los archivos de ASM siempre se distribuyen por todos los discos de ASM del grupo. Los discos de ASM están particionados en unidades de asignación de un megabyte. La unidad de asignación constituye el espacio de disco contiguo más pequeño que asigna ASM. ASM no permite dividir un bloque de Oracle en unidades de asignación.

Nota: El gráfico sólo tiene que ver con un tipo de archivo de ASM: archivo de datos. Sin embargo, ASM se puede utilizar para almacenar otros tipos de archivos de base de datos.

ASM necesita un tipo especial de instancia:

- Tiene init y orapw, pero no DD. Usuarios SYS y SYSTEM con autenticación SO (no más usuarios).
- Instancia: mount o no mount (nunca open). Memoria: de 60M a 120M.
- Comandos de gestión propios: “create | alter | drop diskgroup”.
- Nuevos procesos background:
 - RBAL: coordina el rebalanceo de discos en cada grupo.
 - ORBn (n=0..9): ejecuta el rebalanceo, moviendo extensiones entre discos.

Cada BD que usa ASM, tiene dos nuevos procesos background:

- OSMB: comunicación entre BD y la instancia ASM.
- RBAL: abre y cierra los discos en los grupos, en la parte de la BD.
 - Parámetros del init (para instancia ASM y/o BD q la usa):
- instance_type = ASM (para una BD es RDBMS)
- db_unique_name = +ASM (valor por defecto)
- asm_power_limit = 1 (máximo 11; velocidad rebalanceo; 1 = lento)
- asm_diskstring (limita los dispositivos de disco usables para grupos de discos; ejemplo: '/dev/hd*').

- `asm_diskgroups` (nombres de grupos de discos que se mostrarán automáticamente; por defecto vale `NULL` y los monta todos).
- `large_pool_size` (al menos 8M, para ejecutar los paquetes internos de uso de ASM)

AsmLib

ASMLib es una biblioteca de soporte para la característica de administración de almacenamiento automático de la base de datos Oracle. Automatic Storage Management (ASM) simplifica la administración de base de datos. Elimina la necesidad de que el DBA para gestionar directamente potencialmente miles de archivos de bases de datos Oracle, que sólo requiere la gestión de grupos de discos asignados a la base de datos Oracle. ASMLib permite una base de datos Oracle usando el acceso para ASM más eficiente y capaz de los grupos de discos que se utilizan.

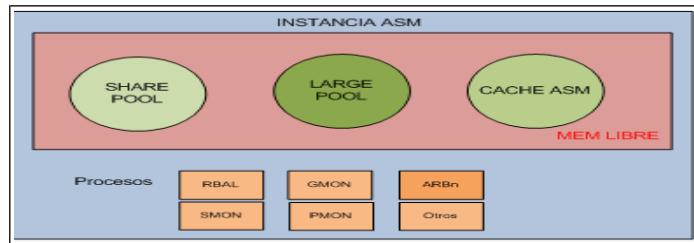
Automatic Storage Management (ASM) es la solución desarrollada por Oracle para automatizar el manejo del storage utilizado para base de datos. Es en esencia, un administrador de volúmenes y sistema de archivo. Inicialmente fue lanzado como uno de los features de la versión 10g de Oracle y esta presente en la versión actual con muchísimas mejoras importantísimas.

ASM brinda a los administradores las siguientes ventajas.

- Provee balanceo automático de carga y redefinición en línea de volúmenes.
- Provee striping y mirroring a nivel de archivos y no a nivel de volumen como RAID.
- Integración automática con Oracle RAC y Clusterware.
- Fácil integración y administración comparada con otros administradores de volúmenes como LVM.
- Elimina los llamados Hotspots en discos.
- Facilita la consolidación de almacenamiento para múltiples bases de datos.

Los componentes básicos de ASM son:

La Instancia ASM.



Es un tipo de instancia especial que sirve para proveer la interfaz entre la instancia de bases de datos normal (RDBMS) y el sistema de archivos. Usualmente esta instancia es de un tamaño pequeño y de poco impacto en el servidor. Consta de cuatro áreas de memoria principales:

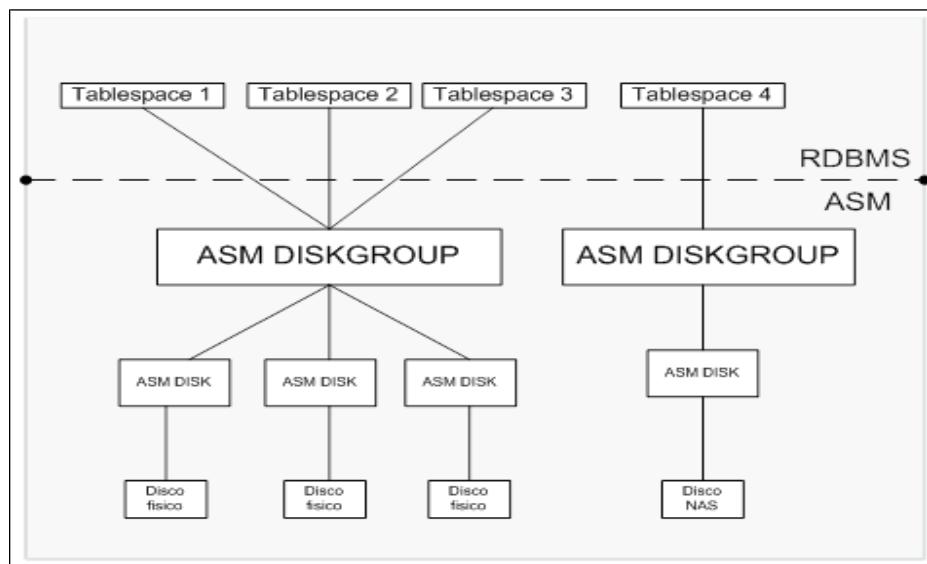
- Share Pool: Esta área de memoria almacena la metadata temporal.
- Large Pool: Usada para operaciones en paralelo.
- Asm Cache: Almacena bloques para lectura y escritura.
- Mem Free: Memoria libre para otros sub procesos.

También componen esta instancia varios server process que al igual que en una instancia rdbms tienen distintas funciones específicas. Para la instancia ASM algunos de los principales son:

- Rbal: Este proceso es el encargado de rebalancear los discos y abrir los archivos.
- Gmon: Realiza tareas a bajo nivel en los discos.
- ARBn: Realiza rebalanceo y otras operaciones.

La instancia ASM es la encargada de montar los diskgroups y hacerlos hábiles para uso.

Discos y Diskgroups:



Los discos son los volúmenes físicos o lógicos (También particiones, NAS) en donde los datos son almacenados, en términos de LVM serían los PV. Los diskgroups son agrupaciones de estos discos que en teoría comparten propiedades similares y son presentados con una unidad. ASM distribuye los datos en todos los discos que pertenecen a un diskgroup mejorando así el I/O de este.

Con la versión actual de ASM los siguientes tipos de archivos pueden ser almacenados en un ASM Diskgroup.

- Flashback logs
- Datapump files
- Archive redo logs
- Datafiles
- Tempfiles
- Controlfiles
- OCR y los Voting Disks
- Con el ACFS (ASM Cluster FileSystem) todo tipo de archivo incluso el mismo ORACLE_HOME.

ASM originalmente fue diseñado para facilitarles la vida a los administradores de bases de datos. Con la versión 11.2 nuevos roles administrativos fueron introducidos para separar las tareas de administración del ASM y la administración de la base de datos. Los roles ASMADMIN, ASMDBA y ASMOPER pueden ser asignados a un administrador para realizar las tareas administrativas del ASM.

Estas tareas administrativas pueden ser aun más simplificadas con la instalación del ASMLIB (Automatic Storage Management Library Driver), ASMLIB nos brinda:

- Simplifica la administración de discos y unidades RAW.
- Permite la configuración persistente de las reglas udev.
- Es lo recomendado por Oracle para instalaciones en Linux.

ASMLIB viene por defecto con la instalación del uek kernel de Oracle Enterprise Linux pero podemos descargar los binarios desde www.oracle.com o pueden instalarlo desde la media de Linux con la siguiente sentencia:

```
[root@oraserv5 ~]# yum -i install oracleasm
```

```
[root@oraserv5 ~]# rpm -uvh oracleasm*
```

Una vez instalado puedes configurarlo con:

```
[root@oraserv5 ~]# /etc/init.d/oracleasm configure
```

```
[root@oraserv5 ~]# /etc/init.d/oracleasm enable
```

Esto te permitirá definir el usuario y grupo dueño del driver, entre otras opciones.

Los discos ASM pueden ser creados mediante el asmlib con:

```
[root@oraserv5 ~]# /etc/init.d/oracleasm createdisk 'DISCO' '/ruta_del_disco'
```

Nota: Cuando el disco es creado aparecerán con candidate.

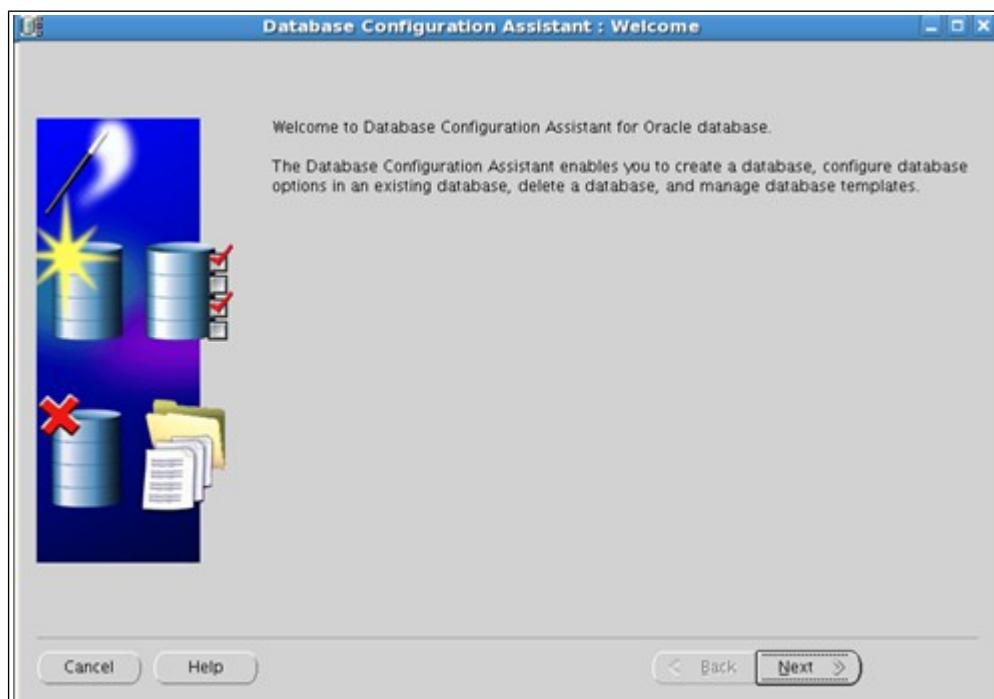
Instalación del ASM

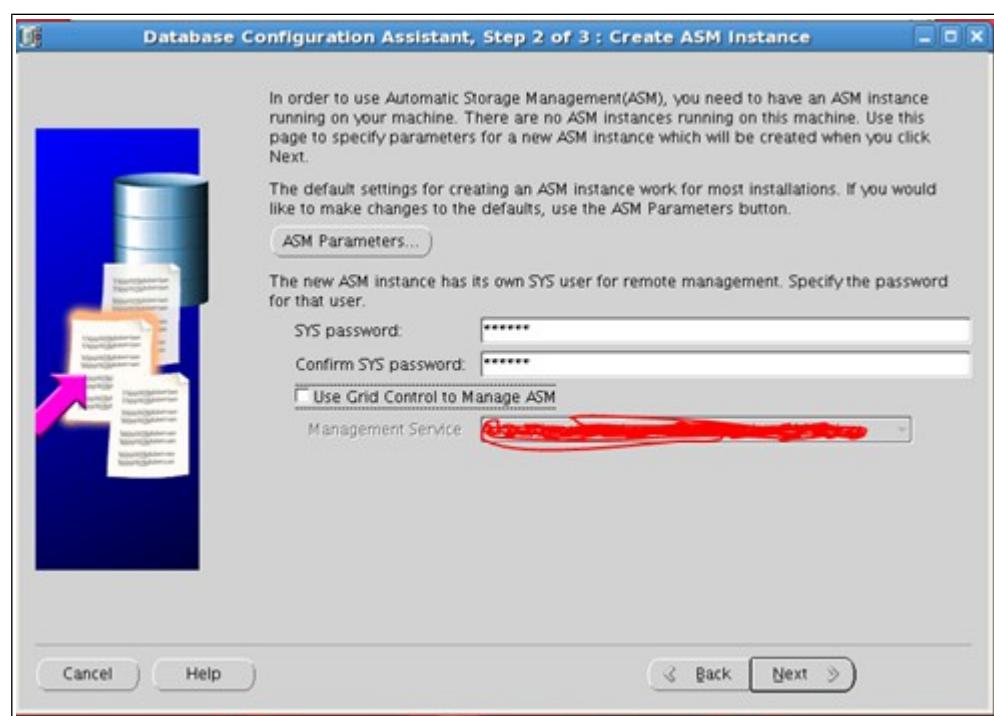
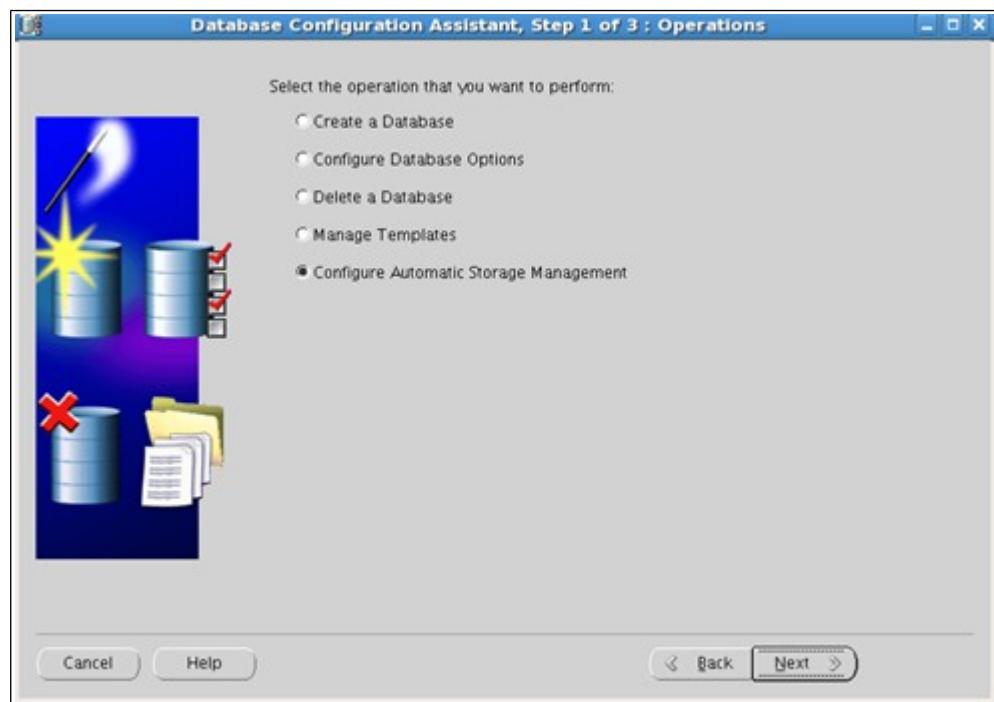
La instalación de la instancia ASM va a depender de la versión. Para las versiones menores a 11.2 la instalación de ASM viene junto con la instalación del software del rdbms y puede ser creada mediante el dbca.

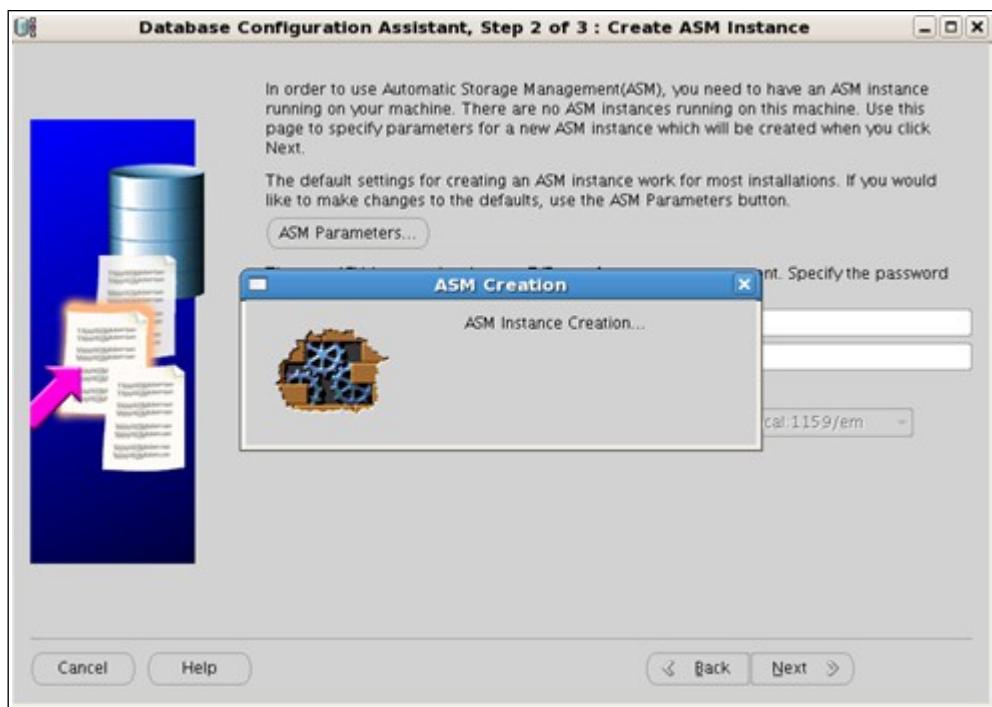
Antes de empezar debemos los servicios del Oracle Synchronization Services que son necesarios para la instancia ASM. Ejecutar el siguiente comando con el usuario root para iniciar el CCS.

```
[root@oraserv5 ~]# $ORACLE_HOME/bin/localconfig add
```

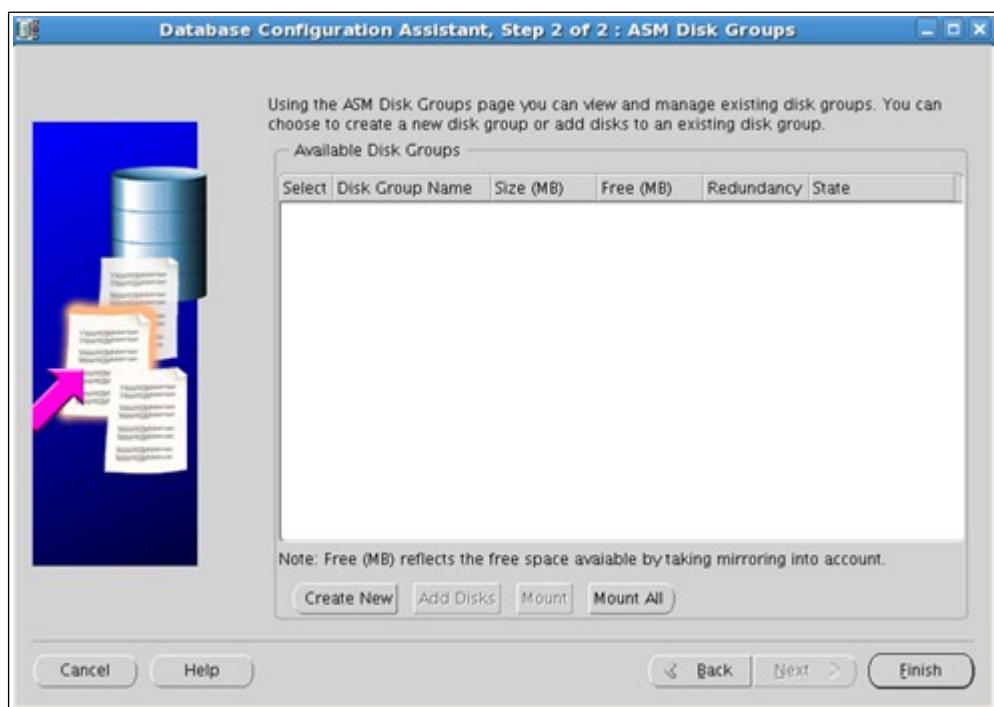
Luego con el usuario dueño de la base de datos ejecutar el dbca.







Una vez creada la instancia, pueden proceder con la administración de discos y diskgroups.



Verificando el status.

```
SQL> select instance_name,status,version from v$instance;

INSTANCE_NAME      STATUS      VERSION
-----  -----
+ASM                STARTED    10.2.0.4.0

SQL> [REDACTED]
```

En la versión 11.2, ASM viene integrado con la instalación del Oracle Clusterware.

Nota: Esta instalación la cubriremos en la guía de instalación del Oracle RAC 11g que pronto estaré publicando.

Ya para finalizar esta pequeña introducción, es importante mencionarles que al momento de usar ASM tengan pendiente los siguientes puntos:

- Utilizar un software owner distinto al dueño de las instancias rdbms. Esto permite tener más control al momento de upgrades y mantenimientos.
- Crear una sola instancia ASM por servidor. Importantísimo.
- Usar el ASMLIB cuando puedan. Solo esta disponible para Linux.
- Crear diskgroups con discos de iguales capacidades y características.
- Crear solo dos diskgroups. Usualmente son llamados DATA y FRA para datos y el flash_recovery respectivamente.

Rman Oracle Recovery Manager

Objetivos: Utilizar RMAN para crear y gestionar juegos de copia de seguridad y copias de imágenes. Recuperar la base de datos hasta un punto anterior en el tiempo Utilizar la tecnología Flashback de Oracle para recuperar la base de datos.

Este tema fue posible gracias a la colaboración del GNU/Instructor Michael Sanchez...

Introducción

RMAN es una herramienta de linea de comandos que permite realizar copias de seguridad y recuperaciones de una base de datos de destino (target database).

RMAN utiliza un repositorio (**repository**) para almacenar la información de su configuración, las copias de seguridad realizadas, la estructura de la base de datos destino, los ficheros de actualización almacenados, etc.

Este repositorio se almacena en el fichero de control de la base de datos destino. La duración de conservación de la información en el fichero de control viene determinada por el parámetro de inicialización CONTROL_FILE_RECORD_KEEP_TIME (7 días por defecto).

El repositorio se puede almacenar en un catálogo de recuperación (**recovery catalog**) que se materializa por medio de un esquema en otra base de datos. Un solo catálogo de recuperación se puede utilizar para centralizar los repositorios RMAN de varias bases de datos objetivo.

Emplear un catálogo de recuperación separado es interesante ya que la información de la copia de seguridad se preserva si todos los ficheros de control de la base de datos destino se pierden.

- Los ficheros de control son, por lo tanto, todavía más importantes cuando se utiliza RMAN sin catálogo de recuperación. Si pierde todos los ficheros de control de la base de datos objetivo, RMAN no tiene más información sobre las copias de seguridad disponibles. Si se parte de una copia de seguridad del fichero de control, RMAN no tendrá información sobre las copias de seguridad realizadas después de la copia de seguridad del fichero de control. Si no utiliza el catálogo de recuperación, es interesante aumentar el valor del parámetro CONTROL_FILE_RECORD_KEEP_TIME (al menos de 10 a 15 días).

Si define una zona de recuperación rápida (**flash recovery área**), gracias a los parámetros DB_RECOVERY_FILE_DEST y DB_RECOVERY_FILE_DEST_SIZE, se puede beneficiar de las funcionalidades de copia de seguridad y restauración automáticas en disco propuestas por RMAN. Adicionalmente, puede definir una política de conservación (**retention policy**) indicando cuánto tiempo se debe conservar una copia de seguridad. RMAN se encarga de gestionar el espacio de la zona de recuperación, rápidamente suprimiendo, si es necesario, las copias de seguridad obsoletas o las copias de seguridad copiadas en cinta.

Para cada operación de copia de seguridad o restauración, RMAN utiliza un canal (**channel**). Un canal es una conexión entre el cliente RMAN y un proceso servidor de la base de datos destino que accede a un periférico (disco o cinta).

Una copia de seguridad RMAN se puede hacer bajo la forma de una copia de imagen (image copy) o de un juego de copias de seguridad (**backup set**). Una copia de imagen es una copia idéntica del fichero (análogo a una copia por un comando del sistema operativo).

Un juego de copias de seguridad contiene uno o varios ficheros salvaguardados. Un juego de copias de seguridad incluye uno o varios ficheros; cada fichero de un juego se llama elemento de copia de seguridad (**backup piece**). Por defecto, un juego de copias de seguridad incluye un solo elemento de copia de seguridad, aunque es posible limitar el tamaño de estos elementos; en este caso, un juego de copia de seguridad puede contener varios elementos de copias de seguridad si el tamaño total de la copia de seguridad es superior al límite. El juego de copias de seguridad tiene un formato propietario de RMAN.

Para realizar copias de seguridad en cinta, RMAN se comunicará con un software de gestión de medios de comunicación.

RMAN ofrece un gran número de funcionalidades y opciones y se puede utilizar de diferentes maneras. En esta obra, presentaremos las funcionalidades de RMAN, necesarias y suficientes para establecer las estrategias de copia de seguridad/recuperación simple, adaptadas a un gran número de casos. Supondremos fundamentalmente que se ha definido una zona de recuperación, lo que permite simplificar un gran número de operaciones y que no se ha establecido ningún catálogo de recuperación separado.

Las funcionalidades de base de RMAN se describen en la documentación Oracle "Data' base Backup and Recovery User's Guide".

Ejecutar RMAN

Para ejecutar RMAN, es suficiente con ejecutar el comando rman en la línea de comandos del sistema operativo.

Sintaxis :

rman [lista_opciones]

```
[oracle@oraserv5 ~]$ rman [lista_opciones]
```

Se pueden utilizar las opciones siguientes:

TARGET [=] conexión
CATALOG [=] conexión

Cadena de conexión a la base de datos objetivo.
Cadena de conexión a la base de datos del catálogo de recuperación.

CMDFILE [=]

fichero Ruta del fichero que contiene los comandos RMAN a ejecutar.

LOG [=] fichero

Ruta del fichero diario de la actividad RMAN.

APPEND

Indica que el fichero diario se debe abrir en modo añadir.

USING valor [...] Define uno o varios valores para las variables de sustitución que pueden ser utilizadas en un fichero de comandos RMAN. En el fichero de comandos RMAN, las variables de sustitución se definen por la sintaxis &n (eventualmente seguida de un punto), siendo n un entero.

Los principios de conexión a la base destino son los mismos que con SQL*Plus" utilización por defecto de las variables de entorno (ORACLE_SID, LOCAL, TOW_TASK), utilización de un nombre de servicio de red, etc. Las variables de entorno como NLS_DATE_FORMAT y NLS_LANG influyen también en el formato de las fechas y el idioma de los mensajes mostrados por RMAN. Ademas, la conexión a la base de datos destino se efectúa implícitamente AS SYSDBA.

Ejecutar Rman sin conectarse

```
[oracle@oraserv5 ~]$ rman
```

```
RMAN>
```

Ejecutar Rman conectarse a la destino (Utilización de la variables de entorno y autentifican SYSDBA por el sistema operativo.

```
[oracle@oraserv5 ~]$ rman target /
```

Ejecutar Rman conectarse a la base datos destino (Utilización de un nombre servicio de red y autentificacion SYSDBA por un fichero de contraseña)

```
[oracle@oraserv5 ~]$ rman target sys/kikla@oraservacl
```

Ejecutar Rman y ejecutar un fichero de comandos.

```
[oracle@oraserv5 ~]$ rman cmdfile=backup.rcv log=backup.log append
```

Cada base de datos tiene un identificador único llamado DBID. RMAN muestra el DBID de la base de datos objetivo cuando se conecta. Es interesante anotar que este DBID puede llegar a ser útil para ciertas operaciones.

Algunos comandos útiles

Ciertos comandos RMAN deben terminar con un punto y coma, otros no (punto y coma es opcional). Los comandos RMAN que necesitan un punto y coma se pueden llenar en varias líneas, los otros no. Los comandos RMAN se llenan indiferentemente en mayúsculas o minúsculas.

Se pueden utilizar los comandos siguientes en RMAN:**@fichero**

Ejecuta un fichero de comandos.

@@fichero

Ejecuta un fichero de comandos en el mismo repositorio que el fichero de comandos actual.

SET ECHO ON | OFF

Activa o desactiva el "echo" de los comandos.

SPOOL LOG TO fichero [APPEND]

Escribe la salida RMAN en un fichero.

SPOOL LOG OFF

Detiene la escritura de la salida RMAN en un fichero.

STARTUP [opción]

Arranca la base de datos; las opciones son las mismas que en SQL*Plus.

SHUTDOWN [opción]

Detiene la base de datos; las opciones son las mismas que en SQL*Plus.

ALTER DATABASE MOUNT | OPEN ;

Monta o abre una base de datos.

CONNECT CATALOG conexión

Establece una conexión a la base de datos del catalogo.

CONNECT TARGET conexión

Establece una conexión a la base de datos objetivo.

HOST ['comando']; HOST ["comando"] ;

Ejecuta una consulta SQL en la base de datos objetivo. La consulta no debe terminarse por un punto y coma; si contiene apóstrofes, estos deben ser dobles.

EXIT O QUIT

Sale de RMAN.

RMAN> exit;

Si desea situar un comentario en el final de la linea, debe terminar el comando con un punto y coma (Aunque el punto y coma sea opcional para el comando).

RMAN> #Esto es un Anuncio o Comentario :)

Ademas, es posible agrupar los comandos Rman en un bloque delimitado por llaves y ejecutar este bloque con el comando **RUN**.

```
RMAN> {  
  Sentencias;  
}
```

Ciertos comandos (ALLOCATE, CHANNEL, SET) que se ejecutan en el interior de un bloque, tiene un ámbito limitado al bloque. Ademas, si un comando del bloque fracasa, la ejecución del bloque se detiene.

Ciertos comandos de Rman se deben ejecutar en el interior de un bloque (ALLOCATE CHANNEL, por ejemplo); otros no se pueden ejecutar en un bloque (SPOOL por ejemplo).

Configurar RMAN

RMAN tiene varios ajustes persistentes utilizados por defecto en el momento de las diferentes operaciones.

La configuración actual se puede visualizar ejecutando el comando SHOW ALL.

```
RMAN> SHOW ALL;
```

El comando # default indica que el parámetro es igual a su valor por defecto.

El comando CONFIGURE permite modificar los ajustes persistentes; el comando SHOW ALL muestra el valor de los ajustes utilizando la sintaxis del comando CONFIGURE.

Los principales ajustes son los siguientes:

Configurar los canales y los periféricos

Por defecto, el periférico utilizado es el disco (parámetro DEFAULT DEVICE TYPE), el destino por defecto de las copias de seguridad es la zona de recuperación rápida. Si esta ultima no esta definida, RMAN utiliza un destino por defecto que depende de la plataforma.

Si desea configurar el destino de la copia de seguridad, puede utilizar el comando siguiente:

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK opciones;
```

CONFIGURE CHANNEL DEVICE TYPE DISK opciones ;

La clausula opciones puede tomar uno o varios valores:

FORMAT 'format'

Ruta y formato del nombre del fichero para la copia de seguridad.

MAXPIECESIZE tamaño [K|M|G]

Tamaño máximo de los elementos de copia de seguridad. No hay límite por defecto.

En este ejemplo, el tamaño de los elementos de la copia de seguridad se limita a 2 Gb. Si el tamaño de la copia de seguridad es superior a 2 Gb, el juego de copias de seguridad contendrá varios elementos de copias de seguridad.

La opción format incluye una ruta y un formato de fichero. El formato de fichero utiliza generalmente una o varias de las variables siguientes:

- %U Nombre del fichero único cuya composición depende de la naturaleza de la copia de seguridad:
 - %u_%p_%c para un elemento de copia de seguridad;
 - data-D-%d_id-%I_TS-%N_FNO-%f_%u para una copia de imagen de un fichero de datos;
 - arch-D_%d_id-%I_S-%e_T-%h_A-%a_%u para una copia de imagen de un fichero de actualización almacenado;
 - cf-D_%d_id-%I_%u para una copia de imagen del fichero de control.
- %d Nombre de la base de datos.
- %l Identificador de la base de datos (DBID).
- %h Número de activación de la base de datos.
- %N Nombre del tablespace.
- %f Número de ficheros de datos.
- %e Número de secuencia del fichero de actualización almacenado.
- %h Número de instancia (thread) del fichero de actualización almacenado.
- %s Número del juego de copias de seguridad (backup set).
- %p Número del elemento de copia de seguridad (backup piece) en el interior de un juego de copias de seguridad.
- %c Número de copia del elemento de copia de seguridad (caso de una copia de seguridad replicada). 1 si la copia de seguridad no está replicada.
- %u Cadena única de 8 caracteres basada en el número del juego de copias de seguridad o de la copia de imagen y de la fecha/hora de la copia de seguridad.
- Si utiliza formatos personalizados, asegúrese de que el nombre del fichero generado es único, so pena de eliminar otras copias de seguridad.

En el comando **CONFIGURE CHANNEL DEVICE TYPE DISK**, cualquier opción no especificada, toma su valor por defecto; el valor anterior no se conserva.

Para modificar el tamaño máximo de los elementos de la copia de seguridad, asignando el formato por defecto, puede utilizar el comando siguiente:

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE 4G;
```

Para volver a la configuración por defecto, puede emplear el comando siguiente:

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK CLEAR;
```

Configurar la política de conservación

La política de conservación se puede definir en términos de ventana de restauración o de redundancia.

Una ventana de restauración indica una cantidad de días hacia atrás y, por lo tanto, el momento en el pasado, al que desea regresar. Una redundancia inicia cuantas copias de seguridad de cada fichero se deben conservar; es la política por defecto (con una redundancia de 1).

Para definir la política de conservación, utilice uno de los comandos siguientes:

- Ventana de restauración

```
RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF n DAYS ;
```

- Redundancia

```
RMAN> CONFIGURE RETENTION POLICY TO REDUNDANCY n ;
```

Cuando la zona de recuperación rápida se utiliza, RMAN suprime automáticamente las copias de seguridad obsoletas (si hay necesidad de espacio), apoyándose en la política de conservación por defecto y en el tamaño asignado a la zona de recuperación rápida (parámetro DB_RECOVERY_FILE_DEST_SIZE).

Para volver a la configuración por defecto, puede utilizar el comando siguiente:

```
RMAN> CONFIGURE RETENTION POLICY CLAER;
```

Configurar la copia de seguridad automática del fichero de control

La copia de seguridad automática del fichero de control se puede activar gracias al comando siguiente: CONFIGURE CONTROLFILE AUTOBACKUP ON ;

Cuando la copia de seguridad automática del fichero de control se activa, el fichero de control es, por defecto, salvaguardado en la zona de recuperación rápida.

Si desea definir un destino de la copia de seguridad específico por defecto, puede utilizar el comando siguiente:

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO 'formato' ;
```

La opción formato no puede ni debe incluir nada más que la variable %F (nombre único basado en el identificador de la base de datos, la fecha y un número de secuencia hexadecimal). Sustituya TO 'formato' por CLEAR para volver al formato por defecto.

Activar la copia de seguridad automática del fichero de control activa también la copia de seguridad automática del fichero de parámetros servidor. Cuando la copia de seguridad automática se activa, estos los ficheros son automáticamente salvaguardados cada vez que una modificación de la estructura de la base de datos o una copia de seguridad RMAN se registra en el fichero de control.

Activar la copia de seguridad automática del fichero de control es altamente aconsejado, sobretodo si no utiliza el catálogo de recuperación separado para RMAN. En caso de perdida de todos los ficheros de control, RMAN podrá restaurar estos últimos a partir de una copia de seguridad automática.

Utilización de la zona de recuperación rápida .

Oracle aconseja utilizar una zona de recuperación rápida para beneficiarse de un numero de funcionalidades automáticas relativas a las operaciones de copia de seguridad y de recuperación.

Si se configura una zona de recuperación rápida, se utiliza como destino por defecto de las copias de seguridad y de varios otros ficheros (por ejemplo, los ficheros de actualización almacenados si no se ha definido ningún destino de almacenamiento).

La cuota de espacio asignado a la zona de recuperación rápida (parámetro **DB_RECOVERY_FILE_DEST_SIZE**) debe ser especificada con cuidado, teniendo en cuenta el tamaño de los ficheros que allí se almacenan (copias de seguridad fundamentalmente) y la política de conservación de las copias de seguridad.

Desde el punto de vista de la seguridad, es altamente aconsejado utilizar un disco separado para la zona de recuperación rápida.

Los ficheros generados por Oracle en la zona de recuperación rápida se almacenan en diferentes subdirectorios, con formatos de nombre de ficheros especificados. Estos son los ficheros "gestionados por Oracle" (Oracle-Managed Files).

Los diferentes repositorios se definen de la forma siguiente:

- <DB_UNIQUE_NAME>/<tipo>[/<fecha>] (Unix/Linux);
- <DB_UNIQUE_NAME>\<TIPO>[\<fecha>] (Windows).

Con:

<DB_UNIQUE_NAME>

Nombre único de la base de datos definido por el parámetro DB_UNIQUE_NAME (por defecto igual al parámetro **DB_NAME**).

<TIPO>

Subdirectorio correspondiente al tipo de fichero archivelog (fichero de actualización almacenado), autobackup (copia de seguridad automática del fichero de control y del fichero de parámetros servidor), backupset (juego de copias de seguridad), controlfile (copia de imagen del fichero de control), datafile (copia de imagen del fichero de datos).

<TIPO>

Idéntico a <tipo> pero en mayúsculas.

<FECHA>

Subdirectorio correspondiente a la fecha (con formato AAAA_MM_DD). No existe para los repositorios controlfile y datafile.

Las reglas de nomenclatura de los ficheros gestionados por Oracle son las siguientes:

Tipo de fichero	Formato
Fichero de actualización almacenado	01_mf_<i>_<s>_<u>_.log
Copia de imagen del fichero de datos	01_mf_<t>_<u>_.dbf
Copia de imagen del fichero de seguridad	01_mf_<a>_<u>_.ctl
Juego de copias de seguridad	01_mf_<c>_<a>_<u>_.bkp
Copia de seguridad automática	01_mf_s_<h>_<u>_.bkp

Con:

- <t> nombre del tablespace;
- <u> cadena de 8 caracteres que garantiza la unicidad;
- <g> numero de grupo para los ficheros de actualización;
- <i> numero de instancia (thread) para los ficheros de actualización almacenados;
- <s> numero de secuencia para los ficheros de actualización almacenados;
- <a> nombre dado a la copia de seguridad (opción TAG del comando BACKUP);
- <c> cadena de 5 caracteres correspondiente al contenido del juego de copias de seguridad;
- <h> timestamp de la copia de seguridad automática (numero de segundos que han pasado desde una fecha interna fija).

La misma zona de recuperación rápida se puede compartir entre varias bases de datos, con la precaución de que estas ultimas tengan un nombre único de base de datos (parametro DB_UNIQUE_NAME) diferente.

El comando VALIDATE

El comando VALIDATE se puede utilizar en diferentes situaciones (antes de una copia de seguridad, antes de una restauración, etc.) para detectar eventuales problemas de corrupción o desaparición de ficheros.

Sintaxis simplificadas

RMAN> **VALIDATE que ;**

La cláusula que puede tomar uno de los valores siguientes (lista no exhaustiva):

DATABASE Verificación de la totalidad de la base de datos (ficheros de datos, ficheros de control y ficheros de parámetros servidor).

TABLESPACE lista_nombres
Verificación del tablespace..

DATAFILE lista_Numeros_o_nombres
Verificación de uno o varios ficheros de datos.

CURRENT CONTROLFILE Verificación del fichero de control actual.

SPFILE Verificación del fichero de parámetros del servidor.

ARCHIVELOG ALL Verificación de todos los ficheros de actualización almacenados (ALL se puede sustituir por diferentes cláusulas que permiten seleccionar los ficheros de actualización almacenados que se deben verificar).

BACKUPSET lista_claves
Verificación de uno o varios juegos de copias de seguridad.

RECOVERY AREA Verificación de todos los ficheros almacenados en la zona de recuperación rápida.

Copia de seguridad

Generalidades

El comando BACKUP permite efectuar una copia de seguridad. Para que este comando funcione, es necesario que la base de datos esté montada o abierta ya que RMAN necesita acceder al fichero de control de la base objetivo, fundamentalmente para registrar ahí la existencia de la copia de seguridad. Las copias de seguridad de base de datos abierta, solo son posibles si la base de datos funciona en modo ARCHIVELOG. Si la base de datos funciona en modo NOARCHIVELOG, es necesario parar la base de datos (correctamente) y después montarla:

```
RMAN> SHUTDOWN IMMEDIATE  
STARTUP MOUNT  
BACKUP ... ;
```

RMAN puede hacer una copia de seguridad de los ficheros de datos, de los ficheros de control, de los ficheros de actualización almacenados, el fichero de parámetros servidor o de los elementos de copia de seguridad (de una copia de seguridad anterior). Como se ha indicado anteriormente, una copia de seguridad RMAN se puede realizar bajo la forma de una copia de imagen (image copy) o de un luego de copias de seguridad (backup set). Por defecto, la copia de seguridad se efectúa en un juego de copias de seguridad.

Cuando RMAN efectuá una copia de seguridad de ficheros de datos en un juego de copias de seguridad, no hace copia de seguridad de los bloques que nunca se han utilizado de los ficheros, lo que permite ganar espacio. Adicionalmente, es posible comprimir el juego de copias de seguridad; esto ralentiza ligeramente la copia de seguridad, consume mas CPU, aunque disminuye el tamaño de la copia de seguridad de manera importante (normalmente, lo divide por 5). Estas dos funcionalidades no están disponibles en el caso de una copia de imagen (copia bit a bit del fichero de origen).

La sintaxis general del comando BACKUP es la siguiente:

```
RMAN> BACKUP [comentario] que  
[opción];
```

- El comando BACKUP tiene un gran numero de opciones. En esta obra, solo presentaremos las opciones utilizadas mas habitualmente.

La clausula comentario puede tener uno o varios de los valores siguientes:

INCREMENTAL LEVEL n [CUMULATIVE]

Indica que la copia de seguridad es una copia de seguridad incremental.

VALIDATE

Indica simplemente que se debe verificar que se puede realizar la copia de seguridad (comprueba la presencia de los ficheros y que no están corruptos). Esta opción es equivalente a la utilización del comando VALIDATE.

AS COPY O AS [COMPRESSED] BACKUPSET

Indica si es necesario hacer una copia de seguridad bajo la forma de una copia de imagen o de un juego de copias de seguridad, eventualmente comprimida.

La clausula que puede tener uno o varios de los valores siguientes:

DATABASE

Copia de seguridad de la totalidad de la base de datos.

TABLESPACE destino

Copia de seguridad de uno o varios tablespaces.

DATAFILE destino

Copia de seguridad de uno o varios ficheros de datos.

CURRENT CONTROLFILE

Copia de seguridad del fichero de control actual.

SPFILE

Copia de seguridad del fichero de parámetros servidor.

ARCHIVELOG destino

Copia de seguridad de los ficheros de actualización almacenados.

La clausula opción puede tener uno o varios de los valores siguientes:

INCLUDE CURRENT CONTROLFILE

Incluir el fichero de actualización almacenados en la copia de seguridad.

PLUS ARCHIVELOG

Incluir los ficheros de actualización almacenados en la copia de seguridad.

DELETE [ALL] INPUT

Eliminar los elementos salvaguardados (valido únicamente para una copia de seguridad de ficheros de actualización almacenados o una copia de seguridad del juego de copias de seguridad).

FORMAT [=] 'formato'

Especificar un formato para la copia de seguridad (ruta y formato de nombre de fichero).

TAG [=] 'nombre'

Asociar un nombre a la copia de seguridad.

NOT BACKED UP clausula_desde

Indica que solo se haga copia de seguridad de los elementos que no han sido previamente salvaguardados un cierto numero de veces o después de un cierto tiempo.

En el comando **BACKUP**, la única cláusula obligatoria es la cláusula que indica de que es necesario hacer copia de seguridad. Todas las otras cláusulas son opcionales y tienen valores por defecto.

Ciertos valores por defecto proceden de la configuración persistente de **RMAN**. Es el caso fundamentalmente del destino de la copia de seguridad y del formato del nombre de los ficheros (canal por defecto). La opción **FORMAT** del comando **BACKUP** permite especificar un destino diferente para la copia de seguridad. Salvo modificación de la configuración, la copia de seguridad en disco se efectúa por defecto en un juego de copias de seguridad no comprimido (equivalente a la opción **AS BACKUPSET**). Para efectuar una copia de seguridad en un juego de copias de seguridad comprimido, es necesario utilizar la opción **AS COMPRESSED BACKUPSET (BACKUP AS COMPRESSED BACKUPSET ...)**). Para hacer una copia de seguridad bajo la forma de una copia de imagen, es necesario utilizar la opción **AS COPY (BACKUP AS COPY ...)**.

Con la opción **VALIDATE**, el comando **BACKUP** no efectúa de hecho ninguna copia de seguridad; verifica simplemente que se podría realizar una copia de seguridad, es decir, que los ficheros sobre los que se hace la copia de seguridad, se acceden correctamente y no están corruptos.

La opción **TAG** permite asociar un nombre a la copia de seguridad, lo que permite posteriormente identificar fácilmente las copias de seguridad o las categorías de copias de seguridad. Este nombre se incluye en los nombres de ficheros generados por RMAN cuando se realiza una copia de seguridad en la zona de recuperación rápida.

La opción NOT BACKED UP tiene dos sintaxis:

- NOT BACKED UP SINCE TIME = 'fecha';
- NOT BACKED UP n TIMES.

La primera sintaxis permite hacer una copia de seguridad únicamente de los elementos que se han sido salvaguardados después de un cierto tiempo. La opción fecha puede ser una constante conforme al formato de fechas por defecto (**NLS_DATE_FORMAT** de la sesión RMAN) o una expresión (del tipo 'SYSDATE-1'). La segunda sintaxis permite hacer copia de seguridad únicamente de los elementos que no han sido salvaguardados un cierto número de veces; esta sintaxis solo se puede utilizar para los ficheros de actualización almacenados.

Las otras cláusulas se detallan en los puntos siguientes.

En el momento de realizar cada copia de seguridad, RMAN muestra numerosa información sobre las operaciones efectuadas.

En resumen, al hacer una copia de seguridad, encontraremos la información siguiente:

- los ficheros salvaguardados (por ejemplo "fichero de datos de entrada ..." para una copia de seguridad del fichero de datos);
- el nombre completo del fichero de copia de seguridad generado (por ejemplo "descriptor del elemento=" para un elemento de la copia de seguridad);
- si se ha hecho o no una copia de seguridad automática del fichero de control y del fichero de parámetros servidor.

Los ficheros de datos temporales (ficheros de datos de los tablespaces temporales gestionados localmente) no son salvaguardados (es inútil).

Copia de seguridad de la totalidad de la base de datos .

Para hacer una copia de seguridad de la totalidad de la base, es suficiente con utilizar la opción DATABASE en el comando BACKUP:

RMAN> BACKUP DATABASE ;

RMAN utiliza la información del fichero de control de la base de datos destino para definir la lista de los ficheros de datos a copiar. Adicionalmente, hace copia de seguridad del fichero de control y del fichero de parámetros servidor (ver más adelante).

El comando **BACKUP VALIDATE DATABASE** se puede utilizar para verificar que la base de datos está en "buen estado" (ningún fichero es inaccesible, ningún fichero está corrupto).

Copia de seguridad del tablespace o de los ficheros de datos individuales .

Para hacer una copia de seguridad individualmente de algunos tablespaces o de algunos ficheros de datos, se pueden utilizar las opciones TABLESPACE y/o DATAFILE en el comando BACKUP.

Las opciones TABLESPACE y DATAFILE se pueden utilizar simultáneamente en el mismo comando. Un tablespace se designa por su nombre y un fichero de datos por su número o por su nombre. En el momento de la copia de seguridad de un tablespace, RMAN utiliza la información del fichero de control de la base de datos destino para definir la lista de los ficheros de datos del tablespace y de las copias de seguridad.

Copia de seguridad del fichero de control y del fichero de parámetros servidor .

El fichero de control y el fichero de parámetros servidor son salvaguardados automáticamente en dos casos:

- Cuando la copia de seguridad automática del fichero de control esta activa.
- Cuando el fichero de datos numero 1 (el primer fichero de datos del tablespace SYSTEM) se salvaguarda.

En los otros casos, el fichero de control se puede salvaguardar explícitamente utilizando las opciones CURRENT CONTROLFILE o INCLUDE CURRENT CONTROLFILE en el comando BACKUP. Igualmente, el fichero de parámetros servidor se puede salvaguardar explícitamente utilizando la opción SPFILE.

Si la copia de seguridad automática del fichero de control esta activa, el fichero de control o el fichero de parámetros servidor se salvaguardan por duplicado en el momento de ser explícitamente salvaguardados.

El fichero de control y el fichero de parámetros servidor se salvaguardan en un juego de copias de seguridad separado.

- La copia de seguridad automática del fichero de control es mas interesante que una explícitamente copia de seguridad manual, fundamentalmente debido a que RMAN la puede restaurar automáticamente en caso de necesidad (esto no es el caso de una copia de seguridad manual). Ademas, en esta configuración, el fichero de control se guarda automáticamente cuando la configuración de los ficheros de la base de datos cambia.

Copia de seguridad de los ficheros de actualización almacenados .

Si los ficheros de actualización no se almacenan por duplicado en discos separados, o no se almacenan en la zona de recuperación rápida (lo que debe ser normalmente un disco separado), es altamente aconsejable hacer copias de seguridad; esto es lo que permite garantizar una restauración completa.

Los ficheros de actualización almacenados se pueden salvaguardar utilizando las opciones ARCHIVELOG o PLUS ARCHIVELOG en el comando BACKUP.

En los dos casos, si los ficheros de actualización se almacenan en diferentes destinos, se

El comando BAKCUP ARCHIVELOG destino permite hacer una copia de seguridad de los ficheros de actualización designados por la clausula destino. La clausula destino ofrece diferentes posibilidades entre las que encontramos:

ALL

Todos los ficheros de actualización almacenados. No se puede combinar con otras opciones.

FROM TIME 'fecha'

Todos los ficheros de actualización almacenados desde una fecha concreta.

UNTIL ITME 'fecha'

Todos los ficheros de actualización almacenados antes una fecha concreta.

TIME BETWEEN 'fecha1' AND 'fecha2'

Todos los ficheros de actualización almacenados entre dos fechas.

Si el comando incluye el fichero de actualización mas reciente (opción ALL o ausencia de De esta manera, toda la actividad de actualización generada antes del comienzo del comando se salvaguarda.

El comando **BACKUP ... PLUS ARCHIVELOG** permite hacer una copia de seguridad de los ficheros de actualización ademas de otros elementos (pero en un juego de copias de seguridad separado). Este comando efectuá las operaciones siguientes:

- almacenamiento del fichero de actualización actual;
- copia de seguridad de todos los ficheros de actualización almacenados (equivalente al comando **BACKUP ARCHIVELOG ALL**);
- copia de seguridad de los otros elementos;
- de nuevo, almacenamiento del fichero de actualización actual;
- copia de seguridad de los ficheros de actualización almacenados generados a partir del comienzo de la copia de seguridad.

De esta manera, todas las copias de seguridad de los ficheros de datos efectuadas durante la actualización la operación (en un estado incoherente) se pueden utilizar, ya que todos los ficheros de actualización necesarios han sido salvaguardados.

Utilización de la opción NOT BACKED UP .

La opción NOT BACKED UP se puede utilizar para copiar únicamente los ficheros de operación actualización almacenados que no han sido todavía salvaguardados un cierto numero de veces o desde un cierto tiempo.

Utilización de la opción DELETE [ALL] INPUT .

La opción DELETE [ALL] INPUT se puede utilizar para eliminar los ficheros de actualización almacenados que acaban de ser salvaguardados.

Si los ficheros de actualización se almacenan en destinos diferentes, hay diferencias entre las dos opciones:

- **DELETE INPUT** elimina únicamente la copia del fichero de actualización de actualización que se ha utilizado para la copia de seguridad.
- **DELETE ALL INPUT** elimina todas las copias del fichero de actualización salvaguardadas.

Copia de seguridad incremental.

Con RMAN, es posible realizar copias de seguridad incrementales, de la totalidad de la base de datos, de tablespaces individuales o de ficheros de datos individuales. El objetivo incremental es hacer copias de seguridad únicamente de los bloques que han sido modificados después de la última copia de seguridad. Las copias de seguridad incrementales tienen como principal interés reducir el tamaño de las copias de seguridad, fundamentalmente cuando la actividad de actualización es relativamente baja en la base de datos.

Para realizar una copia de seguridad incremental, es suficiente con incluir la opción

```
RMAN> INCREMENTAL LEVEL n [CUMULATIVE] en el comando BACKUP;
```

Una copia de seguridad incremental puede ser de nivel 0 o nivel 1, diferencial o acumulativa:

- Una copia de seguridad incremental de nivel 0 hace copia de seguridad de todos los bloques utilizados de los ficheros de datos. Es equivalente a una copia de seguridad completa (aunque una copia de seguridad completa no está considerada por RMAN como una copia de seguridad incremental de nivel 0).
- Una copia de seguridad incremental diferencial de nivel 1 hace una copia de seguridad de todos los bloques modificados después de la última copia de seguridad incremental de nivel 0 o 1. Es el comportamiento por consecuencia, la duración de la copia de seguridad no se reduce sensiblemente en relación a una copia de seguridad de nivel 0 (no hay ganancia en la lectura, simplemente se gana en la escritura).
- Una copia de seguridad incremental acumulativa de nivel 1 hace copia de seguridad de todos los bloques modificados después de la última copia de seguridad incremental de nivel 0.

Las copias de seguridad incrementales acumulativas son más interesantes para la velocidad de recuperación (menos copias de seguridad intermedias a aplicar), aunque necesitan más espacio de disco.

En el momento de realizar una copia de seguridad incremental de nivel 1, RMAN está obligado a leer todos los bloques utilizados para encontrar los que han sido modificados y deben, por lo tanto, ser salvaguardados. En consecuencia, la duración de la copia de seguridad no se reduce sensiblemente en relación a una copia de seguridad de nivel 0 (no hay ganancia en la lectura, simplemente se gana en la escritura).

Si desea reducir la duración de la copia de seguridad, se puede activar la funcionalidad de rastreo de los bloques modificados (block change tracking). Cuando esta funcionalidad está activada, Oracle guarda en un fichero un rastro de los bloques modificados. En el momento de realizar una copia de seguridad incremental de nivel 1, RMAN no tiene ya necesidad de recorrer todos los bloques utilizados; emplea el fichero de rastreo de los bloques modificados para identificar los bloques a salvaguardar.

Para activar la funcionalidad de rastreo de los bloques modificados, puede utilizar la sentencia SQL siguiente (no es un comando RMAN):

```
RMAN> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE 'fichero' ;
```

El fichero de la ruta completa y el nombre del fichero de rastreo.

Ejemplo de activación a partir de RMAN utilizando el comando SQL:

```
RMAN> SQL "ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE '
```

No hay ninguna recomendación particular en relación a la ubicación del fichero; se puede situar en el entorno de la base de datos o en la zona de recuperación rápida.

El fichero de rastreo de los bloques modificados no es especialmente voluminoso; Oracle anuncia un tamaño de 1/30 000 del tamaño de todos los bloques a rastrear, independientemente de la frecuencia de la actualización. El fichero tiene un tamaño mínimo de 10 Mb.

La vista V\$BLOCK_CHANGE_TRACKING da información sobre el rastro de los bloques modificados.

```
SQL> SELECT * FROM v$block_change_tracking;
```

Si el fichero de rastreo se pierde o se daña, la base de datos no podrá ser abierta (continuara en estado MOUNT). Para abrir la base de datos, es necesario desactivar el rastro y eventualmente, reactivarla si desea continuar utilizando la funcionalidad. No existe la posibilidad de hacer copia de seguridad y restauración del fichero de rastreo.

Para cambiar de ubicación el fichero de rastreo, puede utilizar la sentencia SQL ALTER DATABASE RENAME FILE, base de datos montada.

Para desactivar la funcionalidad de rastreo de los bloques modificados, puede utilizar la sentencia SQL siguiente:

```
RMAN> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
```

Después de la activación del rastreo, la primera copia de seguridad incremental de nivel 0 deberá recorrer todos los bloques utilizados, ya que el fichero de rastreo todavía no refleja el estado de los bloques. Es lo mismo después de la recreación del fichero de rastreo.

Emplear o no un fichero de rastreo de los bloques modificados no cambia nada en relación a los comandos a utilizar una copia de seguridad bajo la forma de copia de imagen y de actualizar esta copia de imagen mediante la aplicación regular de copias de seguridad incrementales (Incrementally Updated Backup). Para saber más sobre esto, consulte la relación documentación Oracle@ Database Backup and Recovery User's Guide.

Ejemplos de escenario

Preámbulo

Los escenarios presentados aquí se apoyan en dos hipótesis:

- Se utiliza una zona de recuperación rápida.
- Se ha activado la copia de seguridad automática de los ficheros de control.

Copia de seguridad completa de base de datos cerrada (coherente).

Los comandos siguientes permiten realizar una copia de seguridad completa de base de datos cerrada (por lo tanto, coherente):

```
SHUTDOWN IMMEDIATE;          # parar la base de datos
STARTUP MOUNT;               # montar la base de datos
BACKUP DATABASE              # realizar la copia de seguridad de      # la base de
datos
SQL "ALTER DATABASE OPEN";   # abrir la base de datos
```

Esta copia de seguridad es un ejemplo típico de lo que se hace cuando la base de datos funciona en modo NOARCHIVELOG.

c. Copia de seguridad completa de base de datos abierta (incoherente)

El comando siguiente permite realizar una copia de seguridad completa de base de datos abierta (por lo tanto, incoherente), con copia de seguridad de los ficheros de actualización almacenados, y eliminación de los ficheros de actualización almacenados salvaguardados:

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE ALL INPUT;
```

Esta copia de seguridad solo se puede efectuar cuando la base de datos funciona en modo ARCHIVELOG.

d. Copia de seguridad parcial de base de datos abierta

En este escenario, se hace una copia de seguridad de la totalidad de la base de datos en tres veces distintas, tres días distintos:

- Copia de seguridad 1: ficheros de datos 1 y 2

```
RMAN> BACKUP DATAFILE 1,2 PLUS ARCHIVELOG DELETE ALL INPUT;
```

- Copia de seguridad 2: ficheros de datos 3 y 4

```
RMAN> BACKUP DATAFILE 3,4 PLUS ARCHIVELOG DELETE ALL INPUT;
```

- Copia de seguridad 3: el resto

```
RMAN> BACKUP DATAFILE NO BACKED UP SINCE TIME='SYSDATE-3'
```

```
RMAN> PLUS ARCHIVELOG DELETE ALL INPUT;
```

En principio, es una variante del escenario anterior. El comando para la tercera copia de seguridad permite copiar todo lo que no ha sido salvaguardado desde hace tres días, incluidos todos los ficheros de datos nuevos.

Es técnicamente posible realizar copias de seguridad parciales de base de datos cerrada, aunque estas copias de seguridad solo se pueden utilizar si la base de datos funciona en modo ARCHIVELOG.

Copia de seguridad incremental .

En este escenario, las copias de seguridad incrementales acumulativas se realizan en un ciclo de una semana:

- Domingo: copia de seguridad incremental de nivel 0

```
RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE ;
```

- Lunes a Sábado: copia de seguridad incremental acumulativa de nivel 1

```
RMAN> BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE ;
```

En este ejemplo, suponemos que la base de datos funciona en modo ARCHIVELOG, lo que Sábado nos permite realizar la copia de seguridad de base de datos abierta; para ser totalmente rigurosos, es necesario tratar los ficheros de actualización almacenados (añadir una cláusula PLUS ARCHIVELOG, por ejemplo).

Este tipo de copia de seguridad se puede realizar también si la base de datos funciona en modo NOARCHIVELOG, añadiendo comandos siguientes:

```
SHUTDOWN IMMEDIATE;          # parar la base de datos
STARTUP MOUNT                # montar la base de datos
BACKUP INCREMENTAL...; 3       #realizar la copia de seguridad de la base de datos
SQL "ALTER DATABASE OPEN";    # abrir la base de datos
```

El repositorio RMAN.

Encontrar información sobre las copias de seguridad.

El comando LIST .

El comando LIST permite consultar el repositorio RMAN para mostrar información sobre las copia de seguridad y los ficheros de actualización almacenados.

Sintaxis 1

```
LIST objetivo [ BY FILE | SUMMARY ] [ filtro_copia_de_seguridad ];
    - objetivo
{ BACKUP | COPY } [ OF objetos ]
BACKUPSET
    -objetivos
DATABASE
DATAFILE lista_numeros_o_nombres
TABLESPACE lista_nombres
CONTROLFILE
SPFILE
ARCHIVELOG { ALL | filtro_fichero }
    -filtro_fichero
FROM TIME 'fecha'
UNTIL TIME 'fecha'
TIME BETWEEN 'fecha1' AND 'fecha2'
    -filtro_copia_de_seguridad
TAG [=] 'nombre'
COMPLETED { AFTER 'fecha1'
    | BEFORE 'fecha2'
    | BETWEEN 'fecha1' AND 'fecha2' }
```

Sintaxis 2

```
RMAN> LIST { BACKUP | BACKUPPIECE } { lista_claves | TAG [=] 'nombre' };
```

Sintaxis 3

```
RMAN> LIST ARCHIVELOG { ALL | filtro_fichero } [info_copia_de_seguridad];
```

- info_copia_de_seguridad
 BACKED UP n TIMES TO DEVICE TYPE [DISK | 'media']

En este libro no se presentan todas las opciones posibles.

La primera sintaxis permite mostrar información sobre las copias de seguridad registradas en el repositorio RMAN.

Por defecto, los comandos **LIST BACKUP**, **LIST COPY** Y **LIST BACKUPSET** listan todos los elementos registrados en el repositorio RMAN.

En el caso de los comandos **LIST BACKUP Y LIST COPY**, es posible especificar uno o varios objetos para mostrar únicamente las copias de seguridad de los objetos solicitados.

El último ejemplo lista las copias de seguridad de los ficheros de actualización almacenados hace más de un día, independientemente de la fecha de la copia de seguridad (puede estar fechada hace menos de un día); no hay que confundir el filtro de fecha de almacenamiento (opción `filtro_fichero`) y el filtro de fecha de copia de seguridad (opción `filtro_copia_de_seguridad`).

La opción `filtro_copia_de_seguridad` permite filtrar la lista gracias a un criterio sobre la copia de seguridad; fecha de la copia de seguridad y/o nombre asociado a la copia de seguridad (opción `TAG` del comando `BACKUP`).

El último ejemplo de la lista las copias de seguridad de los ficheros de actualización almacenados hace más de un día pero que hayan sido salvaguardados hace menos de un día.

Los comandos **LIST BACKUP OF** y **LIST BACKUPSET** listan las copias de seguridad por juego de copias de seguridad, con una visualización detallada que da el contenido de cada copia de seguridad.

La columna "Clave BS" da el número (clave) asignado por RMAN al juego de copias de seguridad

La opción `SUMMARY` permite obtener una visualización resumida (sin detalle del contenido de las copias de seguridad), organizada por juego de copias de seguridad.

La opción `BY FILE` permite obtener una visualización resumida, organizada por fichero salvaguardado.

La segunda sintaxis permite mostrar información sobre los juegos de copias de seguridad (`BACKUPSET`) o elementos de copia de seguridad (`BACKUPPIECE`) precisos (bien mediante una lista de claves, bien mediante el nombre asociado a la copia de seguridad gracias a la opción `TAG` del comando `BACKUP`).

Esta clave de un elemento de copia de seguridad ("Clave BP") no es forzosamente igual a la clave del juego de copias de seguridad ("Clave BS"), ya que un juego de copias de seguridad puede tener varios elementos de copia de seguridad, lo que genera un desfase en la numeración

La tercera sintaxis permite mostrar información sobre los ficheros de actualización almacenados considerados como disponibles por RMAN, es decir, no eliminados por RMAN (con la opción `DELETE INPUT`).

El comando REPORT

El comando REPORT permite realizar consultas mas avanzadas en el repositorio de RMAN.

Existen tres usos principales del comando REPORT:

- listar los elementos que necesitan una copia de seguridad;
- listar las copias de seguridad obsoletas;
- mostrar la lista de los ficheros de datos de la base de datos.

Listar los elementos que necesitan una copia de seguridad

Sintaxis

REPORT NEED BACKUP [condición] [objetos];

 - condición

DAYS [=] n

INCREMENTAL [=] n

RECOVERY WINDOW OF n DAYS

REDUNDANCY [=] n

 - objetos

DATABASE

DATAFILE lista_numeros_o_nombres

TABLESPACE lista_nombres

Por defecto, el comando **REPORT NEED BACKUP** muestra la lista de los ficheros que necesitan una copia de seguridad, teniendo en cuenta la política de conservación configurada (**CONFIGURE RETENTION POLICY**).

La opción condición permite precisar el criterio que el comando **REPORT** debe utilizar para determinar si un fichero necesita ser salvaguardado. Las condiciones posibles son:

DAYs [=] n

Ficheros de datos que necesitan mas de n días de aplicación de ficheros de actualización almacenados, para ser recuperados en caso de incidente.

INCREMENTAL [=] n

Ficheros de datos que necesitan mas de n aplicaciones de copia de seguridad incrementales para ser recuperados en caso de incidente.

RECOVERY WINDOW OF n **DAYS**

Una ventana de recuperación particular (misma sintaxis que en el comando **CONFIGURE RETENTION POLICY**).

REDUNDANCY [=] n

Una redundancia particular (misma sintaxis que en el comando **CONFIGURE RETENTION POLICY**).

La opción objetos permite centrar el interés en tablespaces o en ficheros de datos concretos.

Antes de ejecutar este comando, puede ser útil ejecutar el comando **CROSSCHECK** (ver mas adelante) para actualizar el estado de las copias de seguridad en el repositorio RMAN.

Listar las copias de seguridad obsoletas

Sintaxis

REPORT OBSOLETE [condición];

- condición

RECOVERY WINDOW OF n DAYS

REDUNDANCY [=] n

Por defecto, el comando **REPORT OBSOLETE** muestra las copias de seguridad obsoletas teniendo en cuenta la política de conservación configura (**CONFIGURE RETENTION POLICY**).

La opción condición permite precisar el criterio que debe utilizar el comando **REPORT** para determinar si una copia de seguridad es obsoleta. La sintaxis es la misma que en el comando **CONFIGURE RETENTION POLICY**.

Antes de ejecutar este comando, puede ser útil ejecutar el comando **CROSSCHECK** (ver mas adelante) para actualizar el estado de las copias de seguridad en el repositorio RMAN.

Mostrar la lista de los ficheros de datos de la base de datos

Sintaxis

RMAN> **REPORT SCHEMA ;**

Gestionar el repositorio RMAN

El comando **CROSSCHECK**

El comando **CROSSCHECK** permite verificar que la información registrada en el repositorio RMAN corresponde a ficheros que existen físicamente. Se puede producir un desfase si un fichero se elimina directamente a nivel del sistema operativo. El comando **CROSSCHECK** actualiza el estado del elemento en el repositorio RMAN aunque no elimina nada (ni el fichero físico, ni el registro en el repositorio).

Los estados posibles son los siguientes:

EXPIRED	El objeto no se ha encontrado a nivel del sistema operativo.
AVAILABLE	El objeto esta disponible y se puede utilizar por RMAN.
UNAVAILABLE	El objeto no esta disponible y no se puede utilizar por RMAN (como consecuencia de la utilización del comando CHANGE ... UNAVAILABLE - ver la documentación Oracle).

Un registro que ha sido marcado como **EXPIRED** después de un **CROSSCHECK**, puede volver a pasar **AVAILABLE** después de un nuevo **CROSSCHECK** si solo ha estado temporalmente inaccesible. Se puede también utilizar el comando **CHANGE ... AVAILABLE** para que un registro vuelva al estado **AVAILABLE** si el fichero físico esta otra vez accesible (ver la documentación Oracle).

Sintaxis 1

```
CROSSCHECK objetivo [ filtro_copia_de_seguridad ] ;
    - objetivo
{ BACKUP | COPY } [ OF objetos ]
BACKUPSET
    - objetos
DATABASE
DATAFILE lista_numeros_o_nombres
TABLESPACE lista_nombres
CONTROLFILE
SPFILE
ARCHIVELOG { ALL | filtro_fichero }
```

Sintaxis 2

```
RMAN> CROSSCHECK { BACKUPSET | BACKUPPIECE } { lista_claves | TAG [=] 'nombre' };
```

Sintaxis 3

```
RMAN> CROSSCHECK ARCHIVELOG { ALL | filtro_fichero };
```

No se presentan aquí todas las opciones posibles.

Las ventanas de sintaxis y opciones son las mismas que para el comando LIST. EL estado se muestra en el resultado del comando LIST. El comando LIST EXPIRED, variante del comando LIST, permite listar los elementos que tienen el estado EXPIRED.

El comando DELETE

El comando DELETE se puede utilizar para eliminar copias de seguridad. Elimina los ficheros físicos y el registro en el repositorio RMAN.

El comando DELETE tiene dos variantes principales para:

- eliminar copias de seguridad o ficheros de actualización específicos;
- eliminar copias de seguridad obsoletas.

Eliminar copias de seguridad o ficheros de actualización específicos

Sintaxis 1

```

DELETE [FORCE] [NOPROMPT] [EXPIRED] objetivo [ filtro_copia_de_seguridad ] ;
      - objetivo
{ BAKCUP | COPY } [ OF objetos ]
BACKUPSET
      - objetos
DATABASE
DATAFILE lista_numeros_o_nombres
TABLESPACE lista_nombres
CONTROLFILE
SPFILE
ARCHIVELOG { ALL | filtro_ficheros }
      - filtro_ficheros
FROM TIME 'fecha'
UNTIL TIME 'fecha'
TIME BETWEEN 'fecha1' AND 'fecha2'
      - filtro_copia_de_seguridad
TAG [=] 'nombre'
COMPLETED { AFTER 'fecha1'
      | BEFORE 'fecha2'
      | BETWEEN 'fecha1' AND 'fecha2' }

```

Sintaxis 2

```
RMAN> DELETE [FORCE] [NOPROMPT] [EXPIRED] { BACKUPSET | BACKUPPIECE }
{ lista_claves | TAG [=] 'nombre' };
```

Sintaxis 3

```
RMAN> DELETE [FORCE] [NOPROMPT] [EXPIRED] ARCHIVELOG { ALL | filtro_ficheros }
[info_copia_de_seguridad];
```

- info_copia_de_seguridad

BACKED UP n TIMES TO DEVICE TYPE [DISK | 'media']

Las variantes de sintaxis y opciones son las mismas que para el comando LIST.

La opción EXPIRED permite eliminar los elementos marcados EXPIRED en el repositorio RMAN (eventualmente, combinando con otros criterios).

Por defecto, RMAN lista los ficheros que va a eliminar y solicita confirmación para la eliminación. La opción NOPROMPT permite eliminar la solicitud de confirmación (aunque la lista de los ficheros eliminados se muestra siempre).

El comando DELETE genera un error si no hay concordancia entre el repositorio y los ficheros físicos:

- Un fichero se ha marcado EXPIRED en el repositorio aunque existe físicamente.
- Un fichero se ha marcado AVAILABLE en el repositorio aunque no existe físicamente.

Para resolver este problema, puede elegir:

- ejecutar el comando CROSSCHECK para actualizar el estado de los ficheros en el repositorio;
- utilizar la opción FORCE del comando DELETE;
- utilizar el comando CHANGE ... UNCATALOG para eliminar del repositorio una referencia a un fichero que ya no existe (ver la documentación Oracle).

Reflexione bien antes de eliminar cualquier cosa.

Eliminar las copias de seguridad obsoletas

Sintaxis 2

```
DELETE [FORCE] [NOPROMPT] OBSOLETE [ condición ];
      - condición
```

RECOVERY WINDOW OF n DAYS

REDUNDANCY [=] n

Cuando el comando se llama sin opciones, RMAN elimina las copias de seguridad obsoletas teniendo en cuenta la política de conservación configurada (CONFIGURE RETENTION POLICY).

La opción condición permite precisar el criterio que el comando DELTE debe utilizar para determinar si una copia de seguridad es obsoleta. La sintaxis es la misma que en el comando CONFIGURE RETENTION POLICY.

Si utiliza una zona de recuperación rápida, RMAN eliminara automáticamente las copias de seguridad obsoletas (teniendo en cuenta la política de conservación configurada), aunque solo si falta espacio.

El comando CATALOG

El comando CATALOG permite indicar a RMAN la existencia de ficheros de actualización almacenados o elementos de copia de seguridad que no están registrados en el repositorio RMAN.

Esta situación se puede producir en varias casos:

- Ha utilizado el comando DELETE sin darse cuenta y tiene todavía el fichero físico.
- Ha efectuado una recuperación con una copia de seguridad del fichero de control, que por lo tanto ya no contiene información sobre lo que RMAN ha hecho después de la copia de seguridad en cuestión.
- Ha creado de nuevo el fichero de control (no contenía nada).
- Se ha eliminado un registro del fichero de control mediante el valor del parámetro CONTROL_FILE_RECORD_KEEP_TIME, pero el fichero físico existe todavía y lo necesita para una recuperación.
- Ha cambiado de lugar un fichero físico.

Sintaxis

```
CATALOG { ARCHIVELOG | BACKUPPIECE } lista_ficheros ;
CATALOG { RECOVERY AREA | DB_RECOVERY_FILE_DEST } [NOPROMPT] ;
CATALOG START WITH 'ruta' [NOPROMPT] ;
```

La primera sintaxis permite catalogar los ficheros. Si cataloga un elemento ya catalogado, RMAN suprime la antigua referencia antes de crear una nueva.

La segunda sintaxis permite catalogar todos los ficheros cuyo nombre completo comience con una cierta cadena de caracteres (no puede contener caracteres comodines).

Con las dos ultimas sintaxis, RMAN solicita confirmación antes de catalogar un fichero, la opción NOPROMPT permite eliminar la solicitud de confirmación. Ademas, RMAN no cataloga los ficheros ya catalogados.

recuperación

Descripción general

La estrategia de recuperación depende de varios factores:

- De la naturaleza del fichero(s) dañado(s) o perdido(s):

- fichero de datos;
- fichero de control;
- fichero de parámetros servidor;
- fichero de actualización.

- De modo de funcionamiento de la base:

- ARCHIVELOG
- NOARCHIVELOG.
- De las copias de seguridad disponibles.

Que hacer en caso de problema?

1. Identificar la naturaleza del problema;
2. Definir el modo de actuar teniendo en cuenta el modo de funcionamiento de la base de datos y de las copias de seguridad disponibles.

Sobretodo, no se precipite y no dude en solicitar el soporte de Oracle.

Desde la versión 11, Oracle tiene un asesor para la recuperación de los datos (el Data Recovery Advisor) que permite diagnosticar y resolver fácilmente los incidentes (perdida o daño) de los datos en disco. Esta nueva herramienta se presenta en la sección Data Recovery Advisor.

Los términos "perdido" y "dañado" se utilizaran de manera indiferente para hacer referencia al incidente; en la practica, aunque el fichero este dañado o se pierda, los procedimientos de restauración son los mismos.

Una operación de recuperación se efectúa esencialmente con RMAN. Para ciertas etapas, SQL*Plus también puede ser necesario, principalmente para consultar algunas vistas del diccionario de datos; sera necesaria una conexión AS SYSDBA si la base no esta abierta.

Un consejo, antes de comenzar una operación de recuperación; realice si es posible, una copia de seguridad completa de la base de datos dañada. Esto le puede proporcionar un punto de retorno en caso de que la situación se agrave por una incorrecta manipulación. Al menos, realice una copia de seguridad del fichero de control y de los ficheros de actualización en linea (mediante una simple copia a nivel del sistema operativo).

En una operación de "restauración" o de "recuperación", existen dos etapas bien precisas y distintas:

- La etapa de restauración (restore) consiste en extraer de una copia de seguridad los ficheros necesarios.
- La etapa de recuperación (recover) consiste en aplicar los ficheros de actualización a los ficheros recuperados de la copia de seguridad.

Para ser rigurosos, sera necesario realizar una operación de "restauración y recuperación".

Principios generales de la recuperación

En modo NOARCHIVELOG

En modo NOARCHIVELOG, el procedimiento es muy sencillo:

- restaurar la ultima copia de seguridad completa de la base de datos;
- reiniciar la base de datos.

Todas las modificaciones hechas después de la ultima copia de seguridad se pierden.

A priori, la restauración en modo ARCHIVELOG no permite restablecer la base de datos al estado en el que se encontraba justo antes del incidente; únicamente permite restablecer la base de datos al estado en el que se encontraba en el momento de hacer la copia de seguridad.

No obstante, en ciertas situaciones, puede ser posible recuperar todo o parte de las modificaciones hechas después la ultima copia de seguridad.

El objetivo de las indicaciones dadas mas adelante es mostrar que no todo esta forzosamente perdido. En caso de problema en modo NOARCHIVELOG, no hay que dudar en llamar al servicio de soporte de Oracle para intentar, con ellos, realizar la recuperación mas completa posible. Por el contrario, para garantizar una recuperación completa en todas las situaciones (y simplificar el proceso de recuperación), es necesario hacer funcionar la base de datos en modo ARCHIVELOG.

Las situaciones son las siguiente:

- No ha tenido lugar un ciclo completo de basculación de los ficheros de actualización después de la copia de seguridad.
- El fichero de datos perdido no es critico para la base de datos (no pertenece al tablespace SYSTEM, ni al tablespace de anulación activo), ni para la aplicación (no es el tablespace principal de la aplicación).

- Todos los ficheros de control se han perdido aunque los otros ficheros (datos y actualización) están intactos.

si los ficheros de actualización no han seguido un ciclo completo de basculación después de la copia de seguridad utilizada, todas las actualización es efectuadas después de la copia de seguridad en cuestión, están todavía "disponibles" en los ficheros de actualización. En este caso, es necesario realizar una recuperación como si la base de datos estuviera en modo ARCHIVELOG (ver los escenarios correspondientes).

Si el fichero de datos perdido no es critico para la base de datos ni para la aplicación, y el problema apareció en el momento en que la base de datos estaba parada, la situación es quizás favorable ya que los ficheros que quedan son coherentes entre ellos: si este problema del fichero no existe, el próximo inicio no necesitará una recuperación de la instancia.

En este caso, es posible:

- Iniciar la base de datos en modo MOUNT

```
SQL> CONNECT / AS SYSDBA;
```

```
SQL> STARTUP MOUNT;
```

- Poner los ficheros de datos relacionados OFFLINE con la opción DROP

```
SQL> ALTER DATABASE DATAFILE 2
/opt/app/oradata/ORASERVACL/idx01.dbf' OFFLINE DROP;
```

- Abrir la bases datos

```
SQL> ALTER DATABASE OPEN;
```

- Eliminar el tablespace

```
SQL> DROP TABLESPACE indice;
```

- Y después, crear de nuevo el tablespace (y eventualmente su contenido)

```
SQL> CREATE TABLESPACE indice ...;
```

```
SQL> CREATE INDEX ... ;
```

Al final, el tablespace se elimina: esta técnica no es aplicable si el fichero de datos perdido es critico para la base de datos o para la aplicación. Por el contrario, es posible aplicarla para los tablespaces que contienen únicamente los índices (los datos no se pierden).

Si el problema apareció en el momento en el que la base de datos estaba en funcionamiento, la situación es mas problemática ya que los ficheros de datos que quedan pueden no ser coherentes y no existe un medio de saberlo.

Si no son coherentes, Oracle necesitará los ficheros de actualización en línea para hacerlos coherentes (es la recuperación de la instancia "clásica"). Si los ficheros de actualización están presentes o si únicamente se han perdido los ficheros de actualización INACTIVE, la técnica presentada anteriormente podrá utilizarse; si los ficheros de actualización CURRENT o ACTIVE se pierden, la técnica no podrá utilizarse (es necesario partir de la última copia de seguridad).

Si se pierden todos los ficheros de control, es una situación crítica y delicada, para la que existen diferentes posibilidades de recuperación. La documentación Oracle recomienda contactar con el servicio de soporte de Oracle.

En modo ARCHIVELOG

En modo ARCHIVELOG, el protocolo básico para una pérdida de fichero(s) de datos es el siguiente:

- restaurar la última copia de seguridad de cada fichero perdido;
- aplicar los ficheros de actualización (almacenados y después aquellos que están en línea);
- reiniciar la base de datos (si la recuperación no ha sido de base de datos abierta).

Se recuperan todas las modificaciones hechas después de las copias de seguridad utilizadas. La recuperación se llama completa.

Este tipo de recuperación es sencilla y no plantea problemas si hay, al menos, un fichero de control, un miembro por grupo de ficheros de actualización y todos los ficheros de actualización están disponibles.

Diferentes situaciones pueden conducir a una recuperación incompleta:

- voluntariamente para detenerse antes de una sentencia SQL poco afortunada;
- involuntariamente, si los ficheros de actualización se han perdido (un almacenado o todo el grupo de ficheros de actualización en línea).

En la terminología Oracle, una recuperación incompleta se llama point-in-time recovery.

Si se han perdido todos los ficheros de control, si se ha perdido todos un grupo de ficheros de actualización, o si falta un fichero de actualización almacenado, la recuperación completa será más delicada y en ciertos casos imposible (por ejemplo, si falta un fichero de actualización almacenado); entonces una recuperación incompleta puede ser posible y la base de datos no se restaura al estado en el que se encontraba justo antes del incidente sino a un estado anterior.

En ciertas situaciones (eliminación de una tabla importante, por ejemplo), la recuperación incompleta puede ser voluntaria; de nuevo, la base de datos no se restaura al estado en el que se encontraba justo antes del incidente sino a un estado anterior.

Sea cual sea el origen de la recuperación incompleta, todo lo que se ha hecho después del momento que corresponde al estado de recuperación de la base de datos, se pierde y debe rehacerse a mano: en una secuencia de aplicación de los ficheros de actualización, Oracle no puede "saltarse" alguna sentencia y después continuar.

En el momento de la restauración de las copias de seguridad, si las copias de seguridad son parciales, es necesario tomar la copia de seguridad mas reciente de cada fichero ordinario.

Los incidentes sobre los ficheros de control y de actualización

Los incidentes sobre los ficheros de control y los ficheros de actualización se pueden clasificar en dos categorías: "poco graves" y "muy graves".

Incidentes poco graves:

- perdida de uno o varios ficheros de control, puesto que queda al menos uno;
- perdida de uno o varios ficheros de actualización, puesto que queda al menos uno por grupo.

Incidentes mas graves y mas complejos de tratar:

- perdida de todos los ficheros de control: moderadamente grave si los otros ficheros estan intactos;
- perdida de todos los miembros de un grupo de ficheros de actualización: la gravedad depende del estado del grupo perdido (CURRENT,ACTIVE,INACTIVE).

Estas situaciones son evitables si se replican correctamente los ficheros de control y los ficheros de actualización. La perdida de todos los ficheros de control no es la situación mas compleja de tratar, si existen copias de seguridad recientes del fichero de control y si los otros ficheros (particularmente, los ficheros de actualización) estan intactos; en este caso, una recuperación completa es posible.

La perdida de todos los miembros de un grupo de ficheros de actualización es mas situación compleja de tratar; la situación de inicio debe ser analizada con cuidado (estado del grupo perdido, estado de los otros ficheros, etc.), con el fin de seleccionar el modo de actuar correcto. Para las situaciones complejas, es altamente aconsejable dejarse ayudar por el servicio de soporte de Oracle.

Identificar la naturaleza del problema

a. Mensajes de error relativos a los ficheros de control

Los mensajes de error mas frecuentes sobre los ficheros de control son los siguientes:

ORA-00204
ORA-00205
ORA-00206

Estos mensajes indican que, al menos, un fichero de control se ha dañado o perdido; es necesario consultar el fichero de las alertas de la instancia para saber mas, fundamentalmente para determinar los ficheros dañados y deducir que ficheros estan intactos, si hay alguno. En caso de problema sobre un fichero de control, la instancia se para. Al reiniciarse, la instancia permanece en estado NOMOUNT.

Mensajes de error relativos a los ficheros de actualización .

Los mensajes de error mas frecuentes sobre los ficheros de actualización son los siguientes

ORA-00313
ORA-00315
ORA-00316
ORA-00317
ORA-00318
ORA-00319
ORA-00320
ORA-00321

Estos mensajes se acompañan de uno o varios mensajes ORA-00312 dando el nombre del fichero:

ORA-00312: diario en linea n hilo p:fichero

En caso de problema en todo el grupo de ficheros de actualización, la instancia se detiene. En el momento de reiniciarse, la instancia permanece en estado MOUNT.

En caso de problema, es necesario consultar el fichero de alertas de la instancia y el fichero de traza del proceso LGWR.

c. Mensajes de error relativos a los ficheros de datos

Hay numerosos mensajes de error posibles relativos a los ficheros de datos, por ejemplo:

ORA-01157

Estos mensajes se acompañan de uno o varios mensajes ORA-01110 dando el nombre del fichero:

ORA-01110: fichero de datos n:fichero

EN modo NOARCHIVELOG, si la base de datos esta abierta y se produce un problema sobre un fichero de datos, la instancia se detiene. En modo ARCHIVELOG, es lo mismo permanece pero únicamente si el fichero de datos implicado es un fichero del tablespace SYSTEM o un fichero de datos del tablespace de anulación activo.

En el momento del reiniciarse, la instancia permanece en estado MOUNT.

En caso de problema, es necesario consultar el fichero de alertas de la instancia y el fichero de traza del proceso DBWR. Se pueden citar otros ficheros también dañados.

Cuando la base de datos esta montada o abierta, se puede consultar la vista V\$RECOVER_FILE para determinar la lista de los ficheros de datos sobre los que existe un problema.

Las columnas interesantes del vista V\$RECOVER_FILE son las siguientes:

FILE# Identificador del fichero (unión con V\$DATAFILE.FILE# para recuperar información complementaria sobre el fichero).

ONLINE_STATUS Estado del fichero (ONLINE u OFFLINE).

ERROR Naturaleza del error. Vacío si el error es desconocido y OFFLINE NORMAL si el fichero esta fuera de linea sin error (no hay necesidad de restauración en este caso).

El comando VALIDATE DATABASE también puede utilizarse para identificar los ficheros de datos perdidos o dañados.

Los comandos RMAN**a. Introducción**

En RMAN, las operaciones de restauración y de recuperación se efectúan respectivamente con los comandos RESTORE Y RECOVER.

El comando RESTORE permite restaurar los ficheros a partir de las copias de seguridad. El comando RECOVER permite proceder a una recuperación completa o incompleta.

La sintaxis general de estos dos comandos es del tipo:

RMAN> { RESTORE | RECOVER } objetivo [opciones] ;

Su principal responsabilidad, cuando utiliza estos comandos, es elegir bien el objetivo efectúan en función de la naturaleza del problema. Entonces, RMAN se encarga normalmente de todo: identificar las copias de seguridad a utilizar y extraer de ellas los ficheros necesarios; de seguridad si han sido salvaguardados y después eliminados.

Las opciones de estos dos comandos solo son necesarias para tratar los casos particulares: copia de seguridad no disponible, voluntad de volver a un instante en el pasado (recuperación incompleta), etc. En la gran mayoría de los casos, no debiera ser necesario.

Los principios de funcionamiento generales de estos comandos se van a presentar mas adelante y después veremos como utilizarlos en diferentes escenarios de restauración.

Los comandos RESTORE Y RECOVER tienen un gran numero de opciones. En esta obra, presentaremos únicamente las opciones utilizadas mas habitualmente.

El comando RESTORE .

La sintaxis simplificada del comando RESTORE es la siguiente:

```
RESTORE objetivos [opciones]
  - objetivos
DATABASE
DATAFILE lista_numeros_o_nombres
TABLESPACE lista_nombres
CONTROLFILE [TO 'destino'] [FROM AUTOBACKUP | 'copia_de_seguridad']
SPFILE [TO 'destino'] [FROM AUTOBACKUP | 'copia_de_seguridad']
ARCHIVELOG { ALL | filtro_fichero }
  - filtro_fichero
FROM TIME 'fecha'
UNTIL TIME 'fecha'
TIME BETWEEN 'fecha1' AND 'fecha2'
  - opciones
PREVIEW [SUMMARY]
VALIDATE
```

La opción objetivos permite indicar aquello que es necesario restaurar. La opción DATABASE permite restaurar la totalidad de la base de datos; solo debe utilizarse si desea o debe restaurar la totalidad de la base de datos. En modo ARCHIVELOG, si un fichero de números datos esta dañado, únicamente debe restaurar el fichero en cuestión, utilizando las opciones DATAFILE O TABLESPACE.

La opción PREVIEW es interesante para listar las copias de seguridad que necesita RMAN números para realizar la operación de restauración correspondiente. La opción SUMMARY permite números obtener una visualización resumida. La visualización es la misma que con el comando LIST.

La opción VALIDATE permite comprobar si la restauración correspondiente puede realizarse. RMAN accede a las copias de seguridad y verifica que puede extraer de ellas los ficheros necesarios. Existe también un comando VALIDATE BACKUPSET que permite probar los juegos de copias de seguridad específicos (ver la documentación Oracle).

El comando RECOVER .

La sintaxis del comando RECOVER es la siguiente:

```
RECOVER objetivo [opciones]
  - objetivo
DATABASE
DATAFILE lista_numeros_o_nombres
tablespace lista_nombres
  - opciones
DELETE ARCHIVELOG [MAXSIZE tamaño [K|M|G]]
```

La opción objetivo permite indicar lo que es necesario recuperar: la base de datos en su también totalidad, o los tablespaces o ficheros de datos específicos.

En el momento de la operación de recuperación, RMAN busca los ficheros de actualización almacenados que necesita, en primer lugar, en el disco. Los ficheros de actualización almacenados que faltan, son automáticamente restaurados a partir de la copia de seguridad, en el repositorio de almacenamiento definido por el parámetro LOG_ARCHIVE_DEST_1 (o en otro destino - ver el comando SET ARCHIVELOG DESTINATION en la documentación).

Al final de la operación, los ficheros de actualización almacenados restaurados en otro lugar diferente a la zona de recuperación rápida, no son eliminados por defecto. La opción DELETE ARCHIVELOG permite eliminar los ficheros de actualización almacenados restaurados que ya no son necesarios, a medida que se utilizan. La opción MAXSIZE permite, si es necesario, limitar el espacio utilizado por RMAN para los ficheros de actualización almacenados restaurados. Si se especifica esta opción, RMAN procederá a la restauración de los ficheros de actualización almacenados en varias etapas, para no exceder el tamaño indicado. Asegúrese de que el tamaño indicado es superior al tamaño de los ficheros de actualización almacenados, en caso contrario obtendrá un error.

La recuperación puede utilizar copias de seguridad incrementales o ficheros de actualización almacenados. Si es RMAN quien elige, utiliza prioritariamente las copias de seguridad incrementales.

Escenarios de recuperación .

a. Presentación

En esta obra, vamos a presentar los escenarios de recuperación de base siguientes:

- recuperación del fichero de parámetros servidor;
- recuperación de un fichero de control;
- recuperación de un fichero de actualización;
- recuperación completa de la totalidad de la base de datos en modo ARCHIVELOG;
- recuperación completa de una parte de la base de datos en modo ARCHIVELOG;
- recuperación de todos los ficheros de control en modo ARCHIVELOG;
- recuperación incompleta en modo ARCHIVELOG;
- recuperación en modo NOARCHIVELOG.

Adicionalmente, haremos referencia a dos casos particulares:

- recuperación en una ubicación diferente;
- tablespace temporal gestionado localmente.

En caso de recuperación real, puede que necesite combinar varios de estos escenarios de base. Por ejemplo, si ha perdido un fichero de control y un tablespace y ésta en modo ARCHIVELOG, aplicara los escenarios siguientes en la sentencia:

- recuperación de un fichero de control;
- recuperación completa de una parte de la base de datos en modo ARCHIVELOG.

Como regla general, si ha perdido el fichero de parámetros servidor, un fichero de control y/o un fichero de actualización, debe resolver primero estos problemas antes de tratar el caso de los ficheros de datos.

Todos estos escenarios se basan en las siguientes hipótesis:

- Ha activado la copia de seguridad automática del fichero de control y del fichero de parámetros servidor.
- Utiliza una zona de recuperación rápida.
- No utiliza una base de datos anexa para almacenar el catálogo RMAN.

Sea cual sea el escenario, si simplemente el fichero está temporalmente inaccesible (controlador de disco averiado, por ejemplo), no es necesaria una restauración; es suficiente con corregir el problema para tener el fichero de nuevo disponible y reiniciar la base de datos. Una restauración es posible no obstante, si no es posible esperar a que el problema se solucione.

Recuperación del fichero de parámetros servidor .

En caso de perdida del fichero de parámetros servidor, tiene dos posibilidades:

- Crearlo de nuevo a partir de un fichero de parámetros de texto (ver el capítulo Creación de una nueva base de datos).
- Recuperación a partir de una copia de seguridad RMAN.

Para recuperarlo a partir de una copia de seguridad automática RMAN situada en la zona de recuperación rápida, el modo de actuar es el siguiente:

- Indicar la instancia sin montar la base de datos (observe que RMAN va a utilizar un fichero de parámetros "temporal" para iniciar la instancia)

```
RMAN> STARTUP NOMOUNT  
error de inicio: ORA-01078
```

RMAN> STARTUP NOMOUNT
error de inicio: ORA-01078

- Restaurar el fichero de parámetros servidor a partir de una copia de seguridad automática especificando la ubicación de la zona de recuperación rápida y el nombre (o el nombre único) de la base de datos

```
RMAN> RESTORE SPFILE FORM AUTOBACKUP DB_RECOVERY_FILE_DEST  
'/opt/app/flash_recovery_area' DB_NAME oraservaci;
```

- Reiniciar la instancia y abrir la base de datos

```
RMAN> SHUTDOWN
```

```
RMAN> STARTUP
```

Si a la copia de seguridad automática no esta almacenada en la zona de recuperación rápida, el modo de actuar es diferente. Es necesario poner el DBID correspondiente a la base de datos (SET DBID ...), especificar el formato utilizado para las copias de seguridad automáticas (SET CONTROLFILE AUTOBACKUP FORMAT ...) antes de restaurar la copia de seguridad por un RESTORE SPFILE FROM AUTOBACKUP (sin otra opción).

Cambien es posible restaurar el fichero de parámetros servidor especificando la copia de seguridad a utilizar: RESTORE SPFILE FROM 'copia_de_seguridad'.

Recuperación de un fichero de control .

En caso de haber perdido uno o varios ficheros de control, pero cuanto todavía le queda al menos uno, no debe partir de una copia de seguridad del fichero de control. Simplemente duplique uno de los ficheros de control restantes para remplazar los ficheros perdidos.

Suponemos que la instancia esta parada.

El modo de actuar es el siguiente:

- utilizar el fichero de alerta de la instancia para identificar los ficheros de control dañados o perdidos y comprobar que queda al menos un fichero de control valido;
- duplicar una versión valida del fichero de control para ponerlo en lugar del fichero(s) de control dañado(s);
- reiniciar la base de datos (STARTUP).

Si un fichero de control esta duplicado en otra ubicación distinta a la ubicación de origen, es necesario modificar el parámetro CONTROL_FILES en el fichero de parámetros servidor. En lugar de reiniciar directamente la base de datos, es necesario proceder de la manera siguiente:

- Iniciar la instancia, sin montar la base de datos

```
50L> STARTUP NOMOUNT
```

- Modificar el parámetro CONTROL_FILES en el fichero de parámetros servidor

```
50L> ALTER SYSTEM SET CONTROL_FILE=
2 '/opt/app/oradata/ORASERVACL/control01.ctl',
3 '/opt/app/oradata2/ORASERVACL/control02.ctl' -- cambio
4 SCOPE=SPFILE;
```

- Reiniciar la instancia

```
50L> SHUTDOWN IMMEDIATE
```

```
50L> STARTUP
```

En RMAN se puede hacer la duplicación de una versión valida del fichero de control gracias a una variante del comando RESTORE CONTROLFILE.

```
RMAN> RESTORE CONTROLFILE FROM '/opt/app/oradata/ORASERVACL/control01.ctl' ;
```

El comando trata de una vez todos los ficheros de control que faltan basándose en el valor del parámetro CONTROL_FILES.

Es igualmente posible iniciar temporalmente con menos ficheros de control; en este caso, basándose sera también necesario modificar el parámetro CONTROL_FILES en el fichero de parámetros servidor.

Recuperación de un fichero de actualización .

Sis ha perdido uno o varios ficheros de actualización, pero al menos le queda uno por grupo, no tiene necesidad de realizar la restauración o la recuperación de la base de datos. Simplemente cree de nuevo los ficheros de actualización perdidos.

El modo de actuar es el siguiente:

- Identificar el (los) fichero(s) de actualización dañado(s) en el fichero de alerta de la instancia, en el fichero de traza de LGWR o en la vista V\$LOGFILE.
- Eliminar el miembro dañado .

```
SQL> ALTER DATABASE DROP LOGFILE MEMBER 'nombre_fichero'
```

- Añadir un nuevo miembro al grupo en cuestión

```
SQL> ALTER DATABASE ADD LOGFILE MEMBER 'nombre_fichero'  
2 TO GROUP numero;
```

- Repetir las dos operaciones anteriores con todos los miembros dañados.

Los ficheros de actualización dañados tienen una columna STATUS a INVALID en la vista V\$LOGFILE.

El fichero de actualización añadido puede situarse en otra ubicación si se situá en la misma ubicación que el anterior, sera necesario previamente eliminar físicamente el fichero antiguo (si esta presente) o utilizar la clausula REUSE en la sentencia SQL.

No podrá eliminar el miembro si pertenece al grupo actual. En este caso, es necesario cambiar de grupo actual ejecutando la sentencia SQL ALTER SYSTEM SWITCH LOGFILE. Esta sentencia SQL solo puede ejecutarse si la base de datos esta abierta. Si la sentencia base de datos esta cerrada, y no puede abrirse inmediatamente, puede posponer la corrección del problema a mas tarde o contentarse con crear de nuevo el miembro, la eliminación podrá tener lugar mas tarde, una vez que la base de datos este abierta.

Puede ser posible también funcionar temporalmente con menos miembros en un grupo de ficheros de actualización.

Recuperación completa de la totalidad de la base de datos en modo ARCHIVELOG.

Este escenario se basa en la hipótesis de que haya perdido todos los ficheros de datos. La instancia está parada.

El modo de actuar es el siguiente:

- Montar la base de datos

```
RMAN> STARTUP MOUNT
```

- Restaurar la base de datos

```
RMAN> RESTORE DATABASE;
```

- Recuperar la base de datos

```
RMAN> RECOVER DATABASE ;
```

- Abrir la base de datos

```
RMAN> ALTER DATABASE OPEN ;
```

Si no utiliza la zona de recuperación rápida para el almacenamiento, puede especificar la opción DELETE ARCHIVELOG en el comando RECOVER para eliminar los ficheros de actualización almacenados restaurados a medida que se utilizan y eventualmente, limitar el espacio utilizado por estos ficheros.

Recuperación completa de una parte de la base de datos en modo ARCHIVELOG .

Este escenario se basa en la hipótesis de que ha perdido uno o varios ficheros de datos (pero no todos).

Esta operación puede realizarse en base de datos cerrada o en base de datos abierta, de acuerdo con la naturaleza del problema.

- Si ha perdido un fichero de datos del tablespace SYSTEM o un fichero del tablespace de de anulación activo, la instancia se detiene y no podrá abrir la base de datos sin recuperar los ficheros en cuestión.
- Si se trata de otro fichero de datos, la base de datos puede continuar abierta. Por el contrario, si está cerrada, no puede abrirse.

Recuperación de base de datos cerrada

En este ejemplo, el fichero de datos del tablespace SYSTEM se ha perdido; la instancia está parada.

El modo de actuar es el siguiente:

- Montar la base de datos

```
RMAN> startup mount
```

- Restaurar los ficheros de datos deseados bien utilizando un RESTORE TABLESPACE, bien por medio de un RESTORE DATAFILE

```
RMAN> RESTORE TABLESPACE system ;
```

- Recuperar los ficheros de datos bien utilizando un RECOVER TABLESPACE, bien utilizando un RECOVER DATAFILE

```
RMAN> RECOVER TABLESPACE system ;
```

- Abrir la base de datos

```
RMAN> ALTER DATABASE OPEN ;
```

Recuperación de base de datos abierta .

En este ejemplo, el fichero de datos del tablespace INDEX se ha perdido (fichero de datos numero 6).

Si la base de datos esta cerrada pero desea realizar la recuperación de base de datos abierta (para que los usuarios puedan comenzar a trabajar), empiece por la primera parte del procedimiento. Si la base de datos esta abierta, pase directamente a la segunda parte del procedimiento.

La primera parte del procedimiento es la siguiente:

- Montar la base de datos

```
RMAN> STARTUP MOUNT
```

- Restaurar los ficheros de datos deseados bien utilizando un RESTORE TABLESPACE bien por medio de un RESTORE DATAFILE

```
RMAN> RESTORE TABLESPACE system ;
```

- Recuperar los ficheros de datos bien utilizando un RECOVER TABLESPACE, bien por medio de un RECOVER DATAFILE

```
RMAN> RECOVER TABLESPACE system ;
```

- Abrir la base de datos

```
RMAN> ALTER DATABASE OPEN ;
```

Recuperación de base de datos abierta .

En este ejemplo, el fichero de datos del tablespace INDEX se ha perdido (fichero de datos numero 6).

Si la base de datos esta cerrada pero desear realizar la recuperación de base de datos abierta (para que los usuarios puedan comenzar a trabajar), empiece por la primera parte del procedimiento. Si la base de datos esta abierta, pase directamente a la segunda parte del procedimiento.

La primera parte del procedimiento es la siguiente:

- Montar la base de datos

```
RMAN> STARTUP MOUNT;
```

- Poner a OFFLINE los ficheros de datos perdidos

```
RMAN> SQL "ALTER DATABASE DATAFILE 6 OFFLINE";
```

- Abrir la base de datos

```
RMAN> ALTER DATABASE OPEN;
```

La segunda parte del procedimiento es la siguiente:

- Pasar a OFFLINE los tablespaces en cuestión, debe utilizar la opción IMMEDIATE, ya que un fichero de datos no es accesible

```
RMAN> SQL "ALTER TABLESPACE indx OFFLINE IMMEDIATE";
```

- Restaurar los ficheros de datos deseados bien utilizando un RESTORE TABLESPACE, o bien utilizando un RESTORE DATAFILE

```
RMAN> RESTORE DATAFILE 6 ;
```

- Recuperar los ficheros de datos bien utilizando un RECOVER TABLESPACE, o bien utilizando un RECOVER DATAFILE

```
RMAN> RECOVER DATAFILE 6 ;
```

- Pasar a ONLINE los tablespaces en cuestión

```
RMAN> SQL "ALTER TABLESPACE indx ONLINE";
```

Recuperación de todos los ficheros de control en modo ARCHIVELOG .

En este escenario, suponemos que se han perdido todos los ficheros de control y también un fichero de datos. No se trata de una catástrofe ya que disponemos de copias de seguridad automáticas del fichero de control (en la zona de recuperación rápida) y los ficheros de actualización en linea están disponibles. La instancia esta parada.

El procedimiento es el siguiente:

- Iniciar la instancia sin montar la base de datos

```
RMAN> STARTUP NOMOUNT;
```

- Restaurar los ficheros de control a partir de una copia de seguridad automática (en la automáticas zona de recuperación rápida).

```
RMAN> RESTORE CONTROLFILE FROM AUTOBACKUP;
```

- Montar la base de datos

```
RMAN> ALTER DATABASE MOUNT;
```

- Restaurar los ficheros de datos perdidos (ya visto)

```
RMAN> RESTORE DATAFILE 5 ;
```

- Recuperar la base de datos (no únicamente los ficheros de datos, ya que partimos de una copia de seguridad de los ficheros de control)

```
RMAN> RECOVER DATABASE ;
```

- Abrir la base de datos en la opción RESETLOGS (obligatorio)

```
RMAN> ALTER DATABASE OPEN RESETLOGS ;
```

- Obtener una nueva "encarnación" de la base de datos

```
RMAN> LIST INCARNATION OF DATABASE ;
```

En el comando RESTORE CONTROLFILE FROM AUTOBACKUP, puede especificar las opciones DB_RECOVERY_FILE_DEST y DB_NAME (o DB_UNIQUE_NAME) si los valores actuales no son correctos. Por el contrario, si la copia de seguridad automática del fichero de control no esta almacenada en la zona de recuperación rápida, el modo de actuar es diferente. Es necesario poner el DBID correspondiente a la base de datos (SET DBID ...), especificar el formato utilizado para las copias de seguridad automáticas (SET CONTROLFILE AUTOBACKUP FORMAT ...) antes de restaurar la copia de seguridad por un RESTORE CONTROLFILE FROM AUTOBACKUP.

Cuando se parte de una copia de seguridad del fichero de control, RMAN efectuá automáticamente un CROSSCHECK y un CATALOG RECOVERY AREA para actualizar el repositorio o en los ficheros de control (que no están al día ya que provienen de una copia de seguridad , en función de la realidad física de los ficheros.

Ademas, debe abrirá la base de datos con la opción RESETLOGS. Incluso si la recuperación es completa, Oracle considera que es una nueva vía para la base de datos, una nueva "encarnación " de la base de datos. Los números de secuencia de los ficheros de actualización comienzan desde cero.

En las versiones anteriores de Oracle, todas las copias de seguridad y todos los ficheros de actualización almacenados anteriores a la apertura en modo RESETLOGS eran prácticamente inutilizables.

Desde la versión 10, este no es el caso. En el momento de una apertura en modo RESETLOGS, Oracle asocia un numero de activación a la "nueva" base de datos. Oracle utiliza este numero de activación en diferentes lugares, incluido el nombre de los ficheros de actualización almacenados (variable %r en el parámetro LOG_ARCHIVE_FORMAT). De esta manera, Oracle es capaz de asociar cualquier fichero a una encarnación de la base de datos.

El numero de activación actual puede consultarse en la columna INCARNATION de la vista V\$DATABASE. El histórico de las encarnaciones de una base de datos puede consultarse en la vista V\$DATABASE_INCarnation. En RMAN, el comando LIST INCarnation da la lista de las encarnaciones de la base de datos.

En el fichero de alertas de la instancia, encontrara también mensajes del tipo:

RESETLOGS after complete recovery through change 460307

.....

.....

- La noción de encarnación de base de datos es uno de los aspectos mas complejos de Oracle.

recuperación incompleta en modo ARCHIVELOG .

Este escenario va a ilustrar la técnica de recuperación incompleta, partiendo de una situación catastrófica: se ha perdido todo (fichero de parámetros servidor, ficheros de control, ficheros de datos y ficheros de actualización en linea). La instancia esta parada.

Una recuperación incompleta es necesaria en varios casos:

- perdida de todos los ficheros de actualización en linea (es el caso en este escenario);
- perdida de un fichero de actualización almacenado, necesario para una recuperación;
- como consecuencia de una sentencia SQL poco afortunada (DROP TABLE, DROP TABLESPACE, DROP USER, etc).

En todos los casos, es necesario identificar el punto de retorno deseado con una fecha/ hora, un numero SCN o un numero de secuencia del fichero de actualización.

Al final de la recuperación sera necesario, otra vez, abrir la base de datos con la opción RESETLOGS: es una nueva encarnación de la base de datos.

Este escenario es una combinación de escenarios ya estudiados.

El procedimiento es el siguiente:

- Iniciar la instancia sin montar la base de datos (RMAN utiliza un fichero de parámetros "temporal" ya que el fichero de parámetros servidor se ha perdido):

```
RMAN> STARTUP NOMOUNT
```

- Restaurar el fichero de parámetros servidor a partir de una copia de seguridad automática (almacenada en la zona de recuperación rápida para este ejemplo):

```
RMAN> RESTORE SPFILE FROM AUTOBACKUP  
DB_RECOVERY_FILE_DEST '/opt/app/oradata/flash_recovery_area'  
DB_NAME 'ORASERVAACL';
```

- Reiniciar la instancia sin montar la base de datos (inicio con el fichero de parámetros servidor restaurado):

```
RMAN> STARTUP NOMOUNT FORCE
```

- Restaurar los ficheros de control a partir de una copia de seguridad automática (almacenada en la zona de recuperación rápida para este ejemplo):

```
RMAN> RESTORE CONTROLFILE FROM AUTOBACKUP ;
```

- Montar la base de datos:

```
RMAN> ALTER DATABASE MOUNT ;
```

- Restaurar y recuperar la base de datos:

```
RMAN> RESTORE DATABASE;
```

```
RMAN> RECOVER DATABASE;
```

- Abrir la base de datos con la opción RESETLOGS:

```
RMAN> ALTER DATABASE OPEN RESETLOGS ;
```

En este escenario, con el modo de actuar utilizado aquí, es normal que el comando RECOVER se termine con un error porque falta un fichero de actualización. Inicialmente, el comando RESTORE ha efectuado automáticamente un CROSSCHECK y un CATALOG RECOVER AREA para actualizar el repositorio (fundamentalmente los ficheros de actualización almacenados disponibles) en los ficheros de control; el comando RECOVER va lo mas lejos posible con los elementos a su disposición. Antes de abrir la base de datos en modo RESETLOGS, asegúrese de que el numero de secuencia del ultimo fichero de actualización aplicado esta de acuerdo a lo que espera.

En caso de que desee precisar explícitamente el punto de retorno, es posible utilizar una clausula UNTIL en los comandos RESTORE y RECOVER; esta clausula ofrece varias opciones:

UNTIL SCN [=] n Hasta un numero SCN (no incluido).

UNTIL SEQUENCE[=] n Hasta un numero de secuencia de un fichero de actualización (no incluido).

UNTIL TIME [=] 'fecha' Hasta una fecha/hora (no incluida). Se puede especificar bajo la forma de una constante (en el formato de fecha actual) o una expresión del tip 'SYSDATE-1' o "TO_DATE(..)".

En un bloque RUN, es también posible el comando SET UNTIL antes de ejecutar los comandos RESTORE y RECOVER:

```
RMAN> RUN
{
  SET UNTIL ... ;
  RESTORE DATABASE ;
  RECOVER DATABASE ;}
```

recuperación en modo NOARCHIVELOG .

En este escenario, suponemos que se ha perdido toda o parte de la base de datos y que esta ultima funciona en modo NOARCHIVELOG.

En este caso, normalmente la única solución de recuperación consiste en restablecer la base de datos al estado en el que se encontraba en el momento de la ultima copia de seguridad completa de base de datos cerrada, esta ultima podría ser una copia de seguridad incremental.

No obstante, como hemos indicado anteriormente, puede realizarse una recuperación completa si los ficheros de actualización están disponibles y no ha habido un ciclo completo de basculamiento de los ficheros de actualización desde la última copia de seguridad. Puede intentar una restauración de tipo ARCHIVELOG (puntos recuperación completa de la totalidad de la base de datos en modo ARCHIVELOG y recuperación completa de una parte de la base de datos en modo ARCHIVELOG):

- restauración de los ficheros de datos dañados;
- recuperación de los ficheros de datos dañados.

Si la recuperación no tiene errores, se ha conseguido. En cambio, si la recuperación informa de un error del tipo siguiente, la situación es, a priori, desesperada:

En este caso, no queda mas que realizar una recuperación en modo NOARCHIVELOG, con el siguiente procedimiento:

- Iniciar la instancia sin montar la base de datos

```
RMAN> STARTUP NOMOUNT;
```

- Restaurar los ficheros de control a partir de una copia de seguridad automática (almacenados en la zona de recuperación rápida para este ejemplo)

```
RMAN> RESTORE CONTROLFILE FROM AUTOBACKUP;
```

- Montar la base de datos

```
RMAN> ALTER DATABASE MOUNT ;
```

- Restaurar la base de datos

```
RMAN> RESTORE DATABASE;
```

- Si utiliza copias de seguridad incrementales coherentes (base de datos cerrada) de la totalidad de la base de datos, el comando RESTORE anterior habrá restaurado la última copia de seguridad de nivel 0. Puede realizar una recuperación (RECOVER) con la opción NOREDO, para que RMAN aplique las copias de seguridad incrementales de nivel 1 posteriores a la copia de seguridad de nivel 0, sin aplicar los ficheros de actualización.

```
RMAN> RECOVER DATABASE NOREDO;
```

- Abrir la base de datos con la opción RESETLOGS (obligatorio)

```
RMAN> ALTER DATABASE OPEN RESETLOGS ;
```

RMAN es el componente de la base de datos Oracle que se utiliza para realizar operaciones de copia de seguridad y recuperación. Puede realizar copias de seguridad consistentes e inconsistentes, incrementales o completas, de la base de datos completa o de una parte de la misma.

RMAN utiliza un potente lenguaje de control de trabajos y de archivos de comandos propio, así como una API publicada que permite a RMAN interactuar con muchas soluciones de software de copia de seguridad conocidas.

RMAN puede almacenar copias de seguridad en disco para una recuperación rápida, o en cinta para su almacenamiento a largo plazo. Para que RMAN almacene las copias de seguridad en cinta, se debe configurar una interfaz al dispositivo de cinta conocida como capa de gestión de medios físicos (MML).

Configuración de Valores de Copia de Seguridad

En esta página de propiedades se gestionan los valores de copia de seguridad persistente que se utilizan para crear copias de seguridad. Hay valores distintos para disco y cinta. Los valores de cinta dependen de las capacidades de la biblioteca de gestión de medios físicos. Los valores de disco incluyen:

- **Parallelism:** Número de flujos de información de copia de seguridad distintos que desea crear. La mejor configuración para el paralelismo dependerá del hardware. Una única CPU, un único controlador de disco o un único servidor de disco no se beneficiarían de la realización de copias de seguridad paralelas. A medida que aumentan los recursos de hardware, el grado de paralelismo adecuado también aumenta.
- **Disk Backup Location:** Lugar donde se deben almacenar las copias de seguridad. El valor por defecto es el área de recuperación de flash. Si lo cambia, haga clic en "Test Disk Backup" para verificar que RMAN puede escribir en la nueva ubicación.
- **Disk Backup Type:** Seleccione Image Copy, Backup Set o Compressed Backup Set. Haga clic en el separador Backup Set para definir el tamaño máximo de los archivos de juego de copias de seguridad. (Los juegos de copias de seguridad se pueden dividir más si es necesario para un archivado más sencillo.)

Las credenciales de host son necesarias para que Enterprise Manager guarde los cambios a los valores de copia de seguridad.

Enterprise Manager proporciona una interfaz gráfica de las funciones de RMAN más utilizadas. El acceso a las operaciones avanzadas de copia de seguridad y recuperación se realiza a través del cliente de línea de comandos de RMAN.

Oracle Recovery Manager (RMAN), una línea de comandos y Oracle Enterprise Manager basado en herramientas, es el de Oracle método preferido para realizar copias de seguridad de manera eficiente y la recuperación de una base de datos Oracle . RMAN está diseñado para trabajar íntimamente con el servidor, proporcionando a nivel de bloque de detección de la corrupción durante la copia de seguridad y restauración. RMAN optimiza el rendimiento y el consumo de espacio durante la copia de seguridad con compresión de archivos de copia de seguridad de multiplexación y conjunto, y se integra con Oracle Secure Backup y productos de terceros-los medios de comunicación de gestión de backup en cinta.

RMAN se encarga de todos los procedimientos de base de datos subyacente antes y después de copia de seguridad o restaurar, la eliminación de la dependencia de los sistemas operativos y las secuencias de comandos SQL * Plus. Proporciona una interfaz común para las tareas de copia de seguridad a través de host diferente de funcionamiento de los sistemas, y ofrece características no están disponibles a través administrados por el usuario, tales como los métodos de paralelización de los flujos de backup / recuperación de datos, copias de seguridad de los archivos de política de retención, y la historia detallada de todas las copias de seguridad.

Componentes Básicos

Aunque RMAN tiene mas componentes de los que voy a mencionar, los siguientes yo los consideraría como los mínimos necesarios para poder tener una infraestructura segura para nuestros respaldos.

Cliente de RMAN

El cliente de RMAN es la aplicación de Oracle que maneja las operaciones de respaldos y recuperación, este puede utilizar Oracle Net para conectarse a través de la red, así que puede encontrarse en un servidor distinto a la Base de Datos primaria.

Base de Datos Primaria (Target Database)

Es la base de Datos que contiene los controlfiles,datafiles y en dado caso los archived redo logs que RMAN va a respaldar o restaurar.

Base de Datos de Catalogo y Esquema de Catalogo (Recovery and Schema Catalog Database)

Este componente es la base de datos que contiene el catalogo de recuperación, este catalogo contiene metadata que RMAN utiliza para respaldar y recuperar la base de datos primaria.

Aunque este componente no es obligatorio, ya que podemos respaldar utilizando nada mas el controlfile, si diría que es una buena practica tenerlo en nuestro ambiente de respaldos y recuperación.

Manejador de Medios (Media Manager)

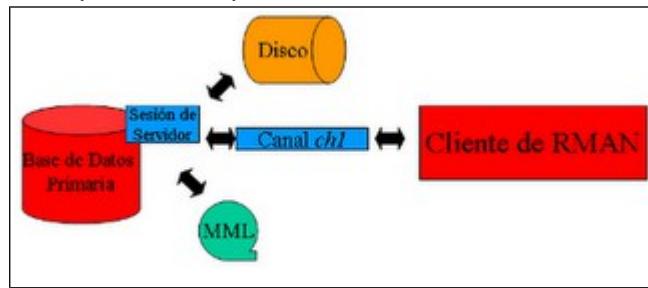
Para ponerlo de manera sencilla, este manejador es una aplicación de un vendor para mandar nuestros respaldos a cinta, como por ejemplo Netbackup o Tivoli.

Canales de RMAN

Una de las cosas mas importantes a entender es que el cliente de RMAN no hace los respaldos, restauraciones o recuperaciones, si no lo que hace es asigna sesiones de servidor en la Base de Datos Primaria y las dirige para que estas hagan las operaciones correspondientes.

En Unix un canal es un proceso de servidor, en Windows es un thread dentro del servicio de la Base de Datos.

En la figura que sigue puedes ver un pequeño ejemplo de como funcionan estos, estos canales son los que cuando hagamos nuestros respaldos, podamos hacer un respaldo en paralelo.



Algo importante a saber, los dispositivos soportados por RMAN para respaldos son disco y SBT (System Backup to Tape), un SBT es manejado por un vendor y normalmente estan conformadas por las librerías de cinta y los drives de cinta.

Una vez que logramos comprender lo mínimo necesario de la arquitectura de RMAN, en la siguiente entrada (Parte II) vamos a ver directamente como configurar nuestro ambiente de respaldo de RMAN, incluyendo la base de datos de catalogo y como hacer respaldos desde RMAN.

Pasos previos a la realización de un backup con el utilitario de rman.

1. Instalar la instancia en donde va a residir el catalogo de backup de rman
2. Crear el catalogo de recuperación
3. Colocar la base de datos de la cual queremos hacer backup en el modo correspondiente ARCHIVELOG/NOARCHIVELOG.
4. Registrar la base de datos de la cual queremos hacer el backup en el catalogo.
5. Crear el script almacenado en el catalogo de recuperación.
6. Ejecutar el script almacenado.

En esta entrada vamos a configurar un esquema de Catalogo para recuperación, así como conectarnos a el y hacer un respaldo de nuestra base de datos.

Conectarnos a RMAN

Existen varias maneras de conectarnos a RMAN, aquí hay varias, la que vamos a utilizar es la tercera, ya nos vamos a estar conectando a un esquema de catalogo de recuperación.

```
[oracle@oraserv5 ~]$ rman TARGET /  
# Autorización vía OS
```

```
[oracle@oraserv5 ~]$ rman TARGET SYS@oraserv5 NOCATALOG  
# RMAN pregunta por el password de SYS de la BD prod
```

```
[oracle@oraserv5 ~]$ rman TARGET / CATALOG aclr@oraserv5  
# RMAN pregunta por el password de rco de la BD catdb
```

Configuración del esquema de Catalogo de Recuperación

Respaldo de los respaldos

Si los respaldos se están haciendo en flashback recovery area, es conveniente respaldar también esta área que se encuentra en disco.

```
RMAN> backup recovery area ;
```

Esto respalda todos los archivos de recuperación en flashback recovery area: full backups, copies, incrementals, respaldos de controlfiles, y archivelogs. No se respaldan block change tracking files, controlfiles actuales ni redo logs. El único destino válido para estos respaldos es cinta.

Políticas de retención

Las políticas de retención nos ayudan a determinar qué respaldos todavía son necesarios y cuáles ya no debido a que han quedado obsoletos por un respaldo más reciente. Hay dos tipos de políticas de retención y son mutuamente excluyentes: redundancy y recovery window.

- Redundancy: determina cuantas copias de un archivo se necesitan antes de considerar una copia obsoleta. Si la redundancia es 1, cada que se respalde un archivo (copia), todas las copias anteriores son obsoletas. REDUNDANCY=1 es la política de retención por default.
- Recovery Window: determina el tiempo que debe ser retenido un archivo antes de ser obsoleto.

El catálogo de RMAN es donde se almacena este repositorio y aunque existe la posibilidad de almacenar parte de esta información en el control file de la base de datos de la cual se obtienen los backups (target database), esta opción se recomienda únicamente para bases de datos muy pequeñas. De igual forma al utilizar al control file para almacenar el repositorio de RMAN ciertas opciones estan restringidas, como por ejemplo el almacenamiento de scripts para automatizar procesos de backups. Las vistas dinámicas registran el flujo de la actividad de la bd.

♪Herramientas GUI para Oracle♪

Objetivos: Aprender el manejo y aprovechar la flexibilidad que brindan las herramientas: SQLDeveloper y Emacs.

SQLDeveloper

Oracle SQL Developer es la herramienta gráfica gratuita que proporciona Oracle para que no sea necesario utilizar herramientas de terceros (como el conocido TOAD, o el PL/SQL Developer) para desarrollar, o simplemente para ejecutar consultas o scripts SQL, tanto DML como DDL, sobre bases de datos Oracle. La apariencia y funcionalidad es similar a la de otras herramientas de este tipo, por lo que es una buena opción si no tenemos especial predilección por otras herramientas.

URL Descarga:

<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>
<http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>

Paquete Necesario.

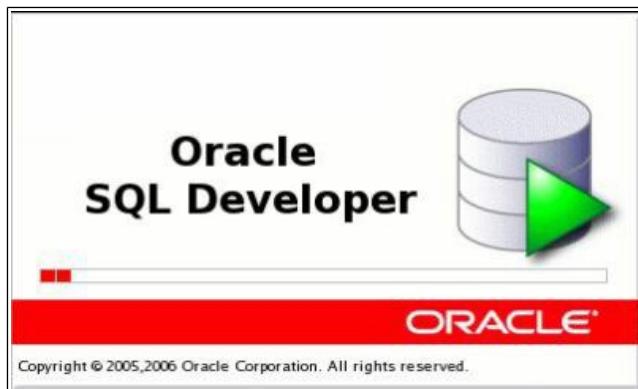
- `sqldeveloper-3.2.20.09.87-1.noarch.rpm`
- `jdk-7u11-linux-i586.rpm`

Instalación de los paquetes.

```
[root@oraserv5 ~]# rpm -Uvh sqldeveloper-3.2.20.09.87-1.noarch.rpm
```

```
[root@oraserv5 ~]# rpm -Uvh jdk-7u11-linux-i586.rpm
```

```
[oracle@oraserv5 ~]$ sqldeveloper
```



Tora

Tora es un conjunto de herramientas multiplataforma de software libre creado para ayudar a los administradores y desarrolladores de aplicaciones de bases de datos Oracle. También suministra soporte para MySQL y PostgreSQL. Es una herramienta muy valorada por los mismos ya que la herramienta que proporciona Oracle, el Enterprise Manager, no es demasiado intuitiva. Los desarrolladores para Oracle de Red Hat Linux afirman hacer un uso intensivo de la misma.

Independientemente del sistema operativo sobre el que se instale, necesita como paso previo que esté instalado el cliente de Oracle correspondiente a la misma o superior versión de base de datos sobre la que queremos actuar. Dicho cliente se puede instalar en su forma mínima, de modo Runtime.
Está disponible para los siguientes sistemas operativos:

Todos los sistemas que soporten POSIX (GNU/Linux, [[BSD]] y en general sistemas operativos tipo UNIX).

GNU/Nota: Usted debe tener instalado Client de Oracle para GNU/Linux.

URL Descarga:

<http://sourceforge.net/projects/tora/files/>

Paquete Necesario.

➤ tora-2.1.3.99.rev4555-1.3.i586.rpm

Instalación de los paquetes.

```
[root@oraserv5 ~]# rpm -Uvh tora-2.1.3.99.rev4555-1.3.i586.rpm
```

Crear 2 enlaces simbólicos que busca T0ra:

```
[root@oraserv5 lib]# cd /usr/lib/oracle/11.2/client/lib/
```

```
[root@oraserv5 lib]# ln -s libclntsh.so.11.2 libclntsh.so
```

```
[root@oraserv5 lib]# ln -s libocci.so.11.2 libocci.so
```

Exportar las librerías necesarias

Para un usuario en particular:

```
[root@oraserv5 ~]# echo "export LD_LIBRARY_PATH=/usr/lib/oracle/11.2/client/lib"
>> ~/.bash_profile
```

Para todo los usuarios del sistema:

```
[root@oraserv5 ~]# echo "export LD_LIBRARY_PATH=/usr/lib/oracle/11.2/client/lib"
>> /etc/bashrc
```

```
[oracle@oraserv5 ~]$ tora
```

**Fuentes:**

<http://oracleen espanol.blogspot.com>
<http://www.orasite.com/>
<http://www.nemaris.com.mx/>
<http://www.dataprix.com>
<http://www.oracle.com>
<http://wikipedia.org>
<http://eudba.wordpress.com>
<http://lecturasoracle.wordpress.com>
<http://pierfinazzi.wordpress.com>
<http://psoug.org>
<http://cavepopo.hd.free.fr>
<http://en.wikipedia.org>
<http://www.dataprix.com>
<http://gruffdba.wordpress.com>
<http://www.albertograjeda.com>
<http://blog.amit-agarwal.co.in>
<http://www.albertograjeda.com>
<http://deveshdba.wordpress.com>