

Manual de Administración de Base de Datos I

Fundación Código Libre Dominicano
Julio 2008.-

Documentación realizada por:

Ing. Jose Paredes
Director de Operaciones

Yolanda Suarez
GNU/Instructor DBA

Francis Aristy
GNU/Instructor DBA

Jesus Rafael Sanchez Medrano
Director de Investigación Científica

Version 1.0

Índice de contenido

Copyright.....	i
Prefacio.....	ii
Instalación de GNU/Linux.....	1
Distribuciones.....	1
Particiones.....	2
Sistema de archivos.....	2
Cuenta Root.....	3
Comandos su - y sudo.....	3
El editor VI.....	4
Instalación de Oracle 10g.....	5
Elementos del servidor Oracle.....	5
Definiciones.....	6
Requerimientos de Hardware y Software:	6
Paquetes Necesarios.....	7
Instalar los paquetes:.....	7
Descripción de parámetros del Kernel a modificar:.....	7
Pasos a seguir para la modificación de parámetros del Kernel:.....	8
Cuenta Oracle.....	9
Creación de Directorios.....	9
Variables del usuario Oracle.....	10
Pasos para la instalación del software:.....	11
Creación de Una Base Datos.....	16
Creación de una Base de Datos con DBCA.....	17
Arquitectura OFA.....	18
Registros de historial (Logs).....	19
Practica:.....	19
Introducción SQL.....	20
Componentes del SQL.....	20
Lenguaje de Descripción de Datos (DDL):.....	21
Lenguaje de Manipulación de Datos (DML):.....	21
Clausulas.....	22
Operadores Lógicos.....	22
Operadores de Comparación:.....	23

Funciones de Agregados.....	24
Practica.....	25
Arranque y Parada.....	26
Arrancar la Base Datos.....	26
Viendo el proceso de arranque:.....	27
Parada de la Base Datos.....	28
Archivos Esenciales para iniciar la Base Datos.....	28
Practica:.....	29
Diccionario de Datos.....	30
¿Que es el Diccionario de Datos?.....	30
Consultas Útiles en el Diccionario de Datos.....	31
Practica.....	33
Procesos Background.....	34
SMON.....	34
DBWR.....	34
PMON.....	34
CKPT.....	35
LGWR.....	35
ARCH	35
Comandos para manejar procesos GNU/Linux.....	36
Comando ps:.....	36
Comando kill:.....	36
Comando killall:.....	36
Comando jobs:.....	36
Comando top:.....	37
Practica:.....	37
Fichero de Control.....	38
Contenido del fichero de control.....	38
Multiplexar fichero de control.....	38
Backup del fichero de control.....	39
Practica:.....	39
Usuarios, roles, privilegios y perfiles.....	40
Usuarios Administradores.....	40
Usuarios y su autenticación.....	40
Creación, modificación y eliminación de usuarios.....	41

Modificación de usuario.....	42
Eliminación de usuarios.....	42
Perfiles.....	43
Crear perfiles.....	43
Modificar un perfil.....	44
Asignación de perfil.....	44
Borrar un perfil.....	44
Privilegios.....	45
Otorgar privilegios.....	45
Quitar privilegios.....	45
ROLES.....	46
Características de los roles:.....	46
Roles predefinidos:.....	46
Creación de Roles.....	47
Modificación de roles.....	47
Roles por defecto.....	47
Revocar un rol.....	47
Eliminación de roles.....	48
Practica:.....	48
Copias de seguridad y recuperación.....	49
Modos de Operación de la DB.....	49
Modo ARCHIVELOG.....	49
Copia logica export/import.....	50
Practica:.....	51
Herramientas GUI para Oracle.....	53
Tora.....	53
Paquetes Necesarios.....	53
Instalación de Clientes.....	53
Emacs.....	54
Comandos mas comunes en relación con las ventanas.....	54
Comandos básicos de emacs.....	55
Para conectar desde Emacs a una Base Datos.....	56

Copyright

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

Prefacio

En estas paginas 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 de GNU/Linux

Objetivos: En este capítulo tratamos de familiarizarnos con el entorno Linux, y que usted se familiarice con los conceptos necesarios que un DBA debe saber sobre la instalación de GNU/Linux.

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 más 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 obligatoria se debe contar con cuatro particiones:

- **swap** Debe asignarse el doble del tamaño del RAM físico.
- **/var**
- **/opt**
- **/** El restante del disco.

Un esquema de particionamiento optimo dispone de una partición **/boot** que contendría la imagen del kernel con no mas de 100 MB y las particiones de **/home** para los datos y documentos de la estación de trabajo y **/var** para sistema servidores por separado.

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, Servicios
proc	Sistema de archivo virtual (información del kernel)

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.

```
[sahirys@sarys ~]$ su oracle
password: XXXXXX
```

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.

```
[sahirys@sarys ~]$ su - oracle
password: XXXXXX
```

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

```
[sahirys@sarys ~]$ su -c 'lsnrctl status'
password: XXXXXX

[sahirys@sarys ~]$ su - -c 'ifconfig'
password: XXXXXX
```

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.

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** : Nos sirve para buscar una palabra y sustituirla la sintaxis es : **%s/viejo/nuevo/**.
- **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.
- **p** : 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.

Instalación de Oracle 10g

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 10g 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:

1. **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.
2. **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 (controlfiles).
 - La estructura lógica está formada por los tablespaces 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 de 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 procesos 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.

Requerimientos de Hardware y Software:

Los requerimientos que Oracle recomienda para su instalación:

Requerimientos del Sistema	Mínimo	Recomendado
Memoria RAM	512 MB	1 GB
Almacenamiento SWAP	1 GB	2 GB
Almacenamiento Temporal	400 MB	1 GB
Espacio en Disco	1.5 GB Software	1.5 GB Software
	1.5 GB Base de Datos	1.5 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`

Revisamos cuanto tenemos de disponibilidad en recursos de almacenamiento:

- **Revisando el directorio /tmp:** `du -sch /tmp`
- **Revisando la disponibilidad del disco:** `df -h`

Paquetes Necesarios

- | | |
|---------------------|---------------------------|
| ◆ binutils | ◆ compatgcc |
| ◆ compatdb | ◆ compatgcc + + |
| ◆ controlcenter | ◆ compatlibstdc + + |
| ◆ gcc gcc + + | ◆ openmotif21 |
| ◆ glibc | ◆ setarch |
| ◆ glibccommon | ◆ xorgx11 |
| ◆ gnomelibs | ◆ compatlibstdc + + devel |
| ◆ libstdc + + | ◆ expat |
| ◆ libstdc + + devel | ◆ fontconfig |
| ◆ make | ◆ freetype |
| ◆ pdksh | ◆ zlib11 |
| ◆ sysstat | ◆ glibc6devel |
| ◆ xscreensaver | ◆ libmotif3 |
| ◆ libaio | ◆ lesstif2 |

Instalar los paquetes:

```
yum install binutils* compat* control-center* gcc* libstdc* make*  
pdksh* sysstat* xscreensaver* libaio* openmotif* glibc* gnome-libs*  
setarch* xorg-x11* expat* fontconfig* freetype* zlib-11* libmotif*  
lesstif* unixODBC* elfutils-devel
```

Descripción de parámetros del Kernel a modificar:

- **SEMMNI = 100:** cantidad máxima de conjuntos de semáforos
- **SEMMNS = 256:** cantidad máxima de semáforos, sólo para la instalación inicial
- **SEMMNS = 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"
- **SEMMSL = 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.
- **SHVMX = 32767:** valor máximo de un semáforo.

Pasos a seguir para la modificación de parámetros del Kernel:

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

```
kernel.shmmax = 536870912
kernel.shmmni = 4096
kernel.shmall = 2097152
# semaphores: semmsl, semmns, semopm, semmni
kernel.sem = 250 32000 100 128
fs.filemax = 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:

```
sysctl -p
```

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

Editamos 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. Desabilitar la SeLinux `/etc/selinux/config`:

```
SELINUX=disabled
```

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:

```
groupadd oinstall
groupadd dba
useradd -g oinstall -G dba oracle
passwd oracle
```

Creación de Directorios

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

```
mkdir -p /opt/app/oracle/product/10.1.0/db_1
```

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

```
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:

```
[sahirys@sarys ~]$vi /etc/redhat-release
Red Hat Linux release 2.1 (drupal)
```

Variables del usuario Oracle

```
TMP=/tmp; export TMP
TMPDIR=$TMP; export TMPDIR

ORACLE_BASE=/opt/app/oracle; export ORACLE_BASE
ORACLE_HOME=$ORACLE_BASE/product/10.1.0/db_1; export ORACLE_HOME
ORACLE_SID=server01; export ORACLE_SID
ORACLE_TERM=xterm; export ORACLE_TERM

PATH=/usr/sbin:$PATH; export PATH
PATH=$ORACLE_HOME/bin:$PATH

export PATH

LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib; export LD_LIBRARY_PATH
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
export CLASSPATH

if [ $USER = "oracle" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
        ulimit -p 16384
        ulimit -n 65536
    else
        ulimit -u 16384 -n 65536
    fi
fi

LIBXCB_ALLOW_SLOPPY_LOCK=1; export LIBXCB_ALLOW_SLOPPY_LOCK
```

Luego se ejecuta el instalador:

```
./runInstaller
```


Pasos para la instalación del software:

1. Selección de método de instalación
 - Elegimos el método de de instalación avanzada:

Oracle Database 11g Installation - Select a Product to Install

Select Installation Method

☒ **Basic Installation**
Perform full Oracle Database 11g installation with standard configuration options requiring minimal input. This option uses file system for storage, and a single password for all database accounts.

Oracle Base Location: Browse...

Oracle Home Location: Browse...

Installation Type:

UNIX DBA Group:

☒ Create Starter Database (additional 1482MB)

Global Database Name:

Database Password: Confirm Password:

This password is used for the SYS, SYSTEM, SYSMAN, and DBSNMP accounts.

☐ **Advanced Installation**
Allows advanced selections such as different passwords for the SYS, SYSTEM, SYSMAN, and DBSNMP accounts, database character set, product languages, automated backups, custom installation, and alternative storage options such as Automatic Storage Management.

Buttons: Help, Back, Next, Install, Cancel

2. Seleccionamos el directorio de inventario credenciales (guardado en oraInventory)
 - /opt/app/oracle/oraInventory

Oracle Universal Installer: Specify Inventory directory

Specify Inventory directory and credentials

You are starting your first installation on this host. As part of this install, you need to specify a directory for installer files. This is called the "inventory directory". Within the inventory directory, the installer automatically sets up subdirectories for each product to contain inventory data and will consume typically 150 Kilobytes per product.

Enter the full path of the inventory directory:

Browse...

You can specify an Operating System group that has write permission to the above inventory directory. You can leave the field blank if you want to perform the above operations as a Superuser.

Specify Operating System group name:

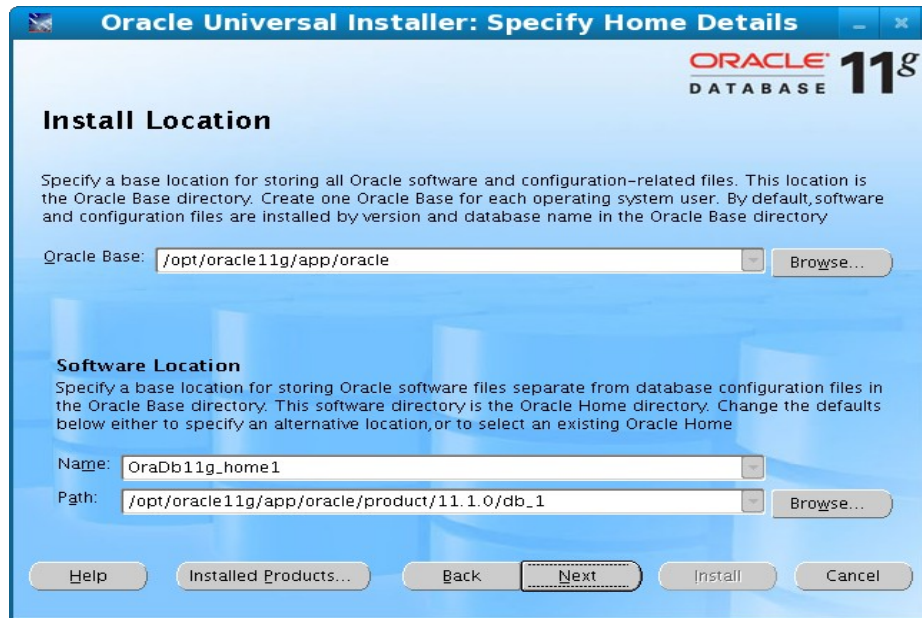
Buttons: Help, Installed Products..., Back, Next, Install, Cancel

- También especificamos el nombre del grupo del s.o. a oinstall

3. Nos saldrá una mini pantalla especificando que ejecutemos el script `/opt/app/oracle/orainventory/orainstroot.sh` con el usuario root:

```
[root@sarys ~]# /opt/app/oracle/oraInventory/orainstRoot.sh
Creating the Oracle inventory pointer file (/etc/oraInst.loc)
Changing groupname of /opt/app/oracle/oraInventory to oinstall.
```

4. Detallar la ruta donde vamos a instalar el producto: en este caso lo nombraremos `oracle10g_home` y la ruta a instalar seria `/opt/app/oracle/product/10.1.0/db_1`

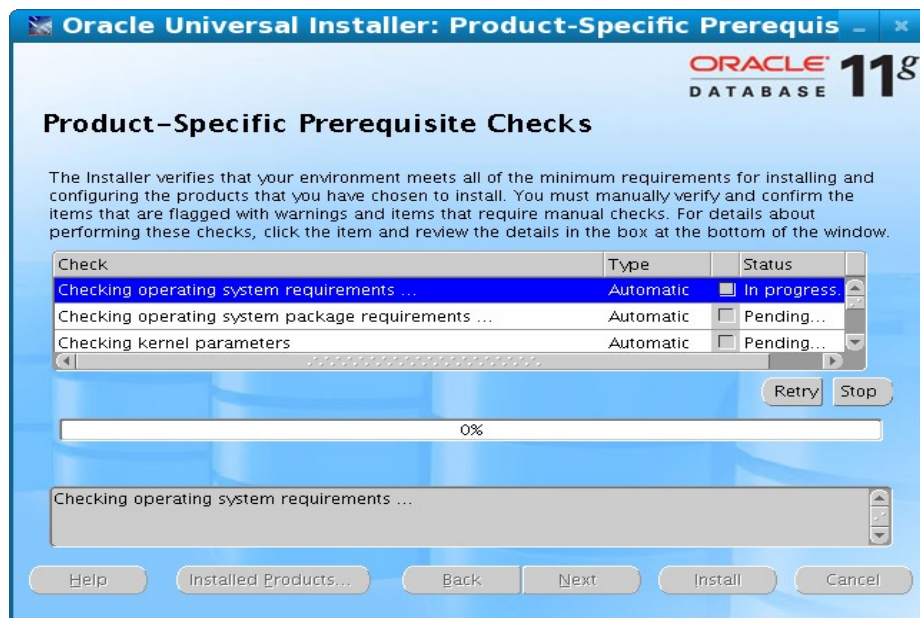


5. Seleccionamos el tipo de instalación:

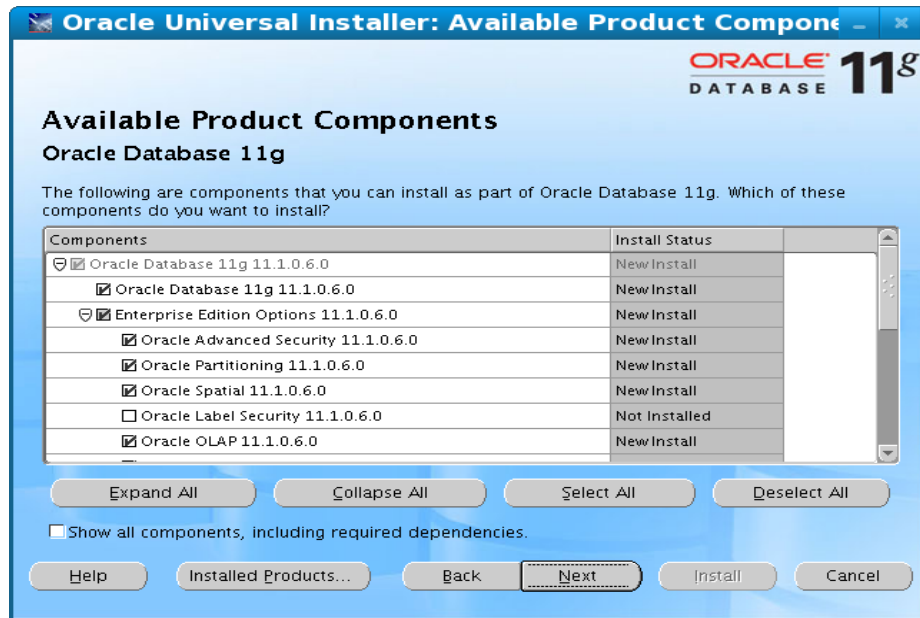
- Elegimos personalizada. Aquí tenemos también la opción de instalar las opciones de lenguaje del oracle:



6. Luego pasamos a la comprobación de pre requisitos específicos del producto:

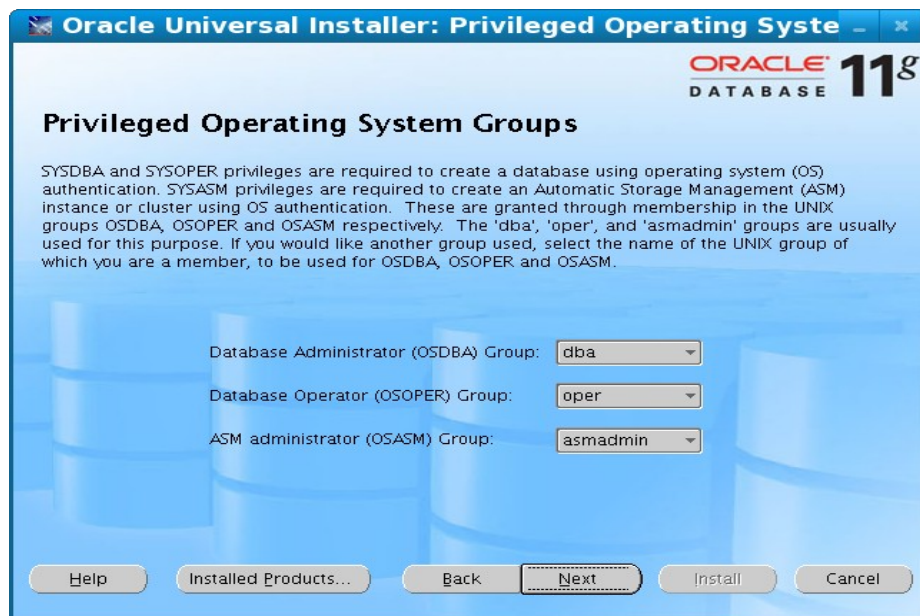


7. Luego veremos los componentes disponibles para instalar:



8. Definimos el grupo del sistema operativo con privilegios:

- grupo de los administradores de la base de datos DBA
- grupo de los operadores de la base de datos DBA



9. En este punto se nos pide crear la base de datos. Le decimos no para luego instalarla.



10. Nos mostrara el sumario de todos los pasos que hemos hecho hasta ahora:



11. Vamos a instalar!

Estos pasos de instalación, son aplicables tanto para la versión 10g como para la 11g.

12. Finalizando la instalación nos saldrá una pantalla solicitando ejecutar el script `/opt/app/oracle/product/10.1.0/db_1/root.sh` con el usuario root.

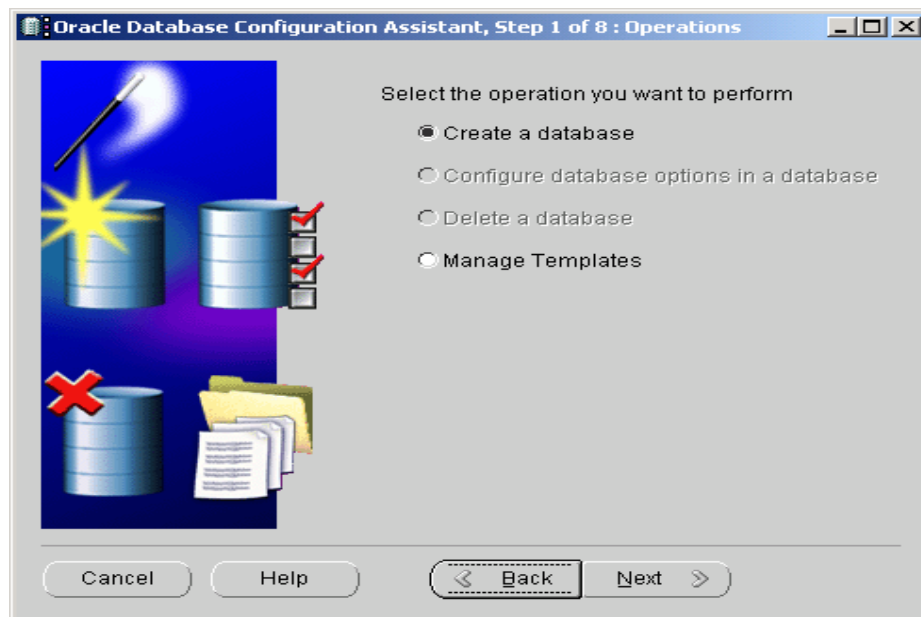
```
[root@sarys ~]# /opt/app/oracle/product/10.1.0/db_1/root.sh
Running Oracle10 root.sh script...
The following environment variables are set as:
  ORACLE_OWNER= oracle
  ORACLE_HOME=/opt/app/oracle/product/10.1.0/db_1
```

13. Configuremos el listener. las opciones por defecto serán suficientes por el momento.

Fin de la instalación Ahora podemos ver el registro de la instalación en `/opt/app/oracle/orainventory/logs/installactions200...(fecha y hora.log)`

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

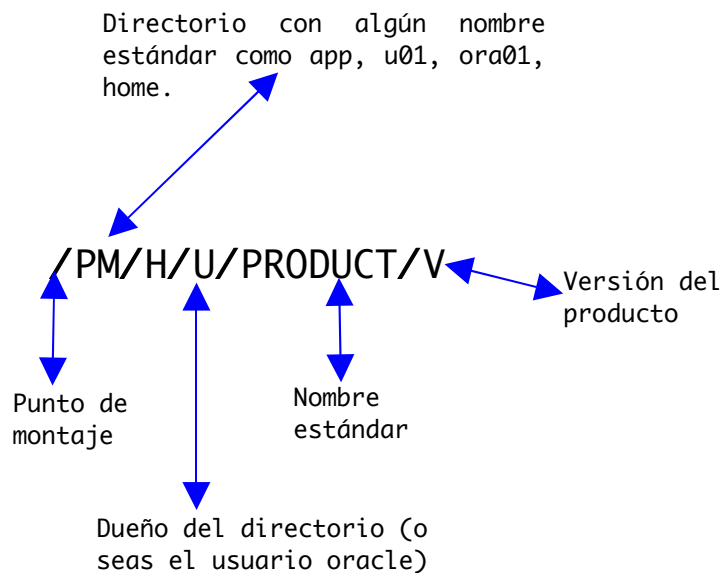
1. Lo primero que veremos sera la pantalla de bienvenida. Presionar el botón “siguiente”:
2. Seleccionamos la opción a realizar:
 - Crear base de datos.
 - Configurar opciones de base de datos
 - Suprimir base de datos
 - Gestionar plantilla
 - Configurar la gestión automática de almacenamiento
3. En este caso crearemos una base de datos.
4. Seleccionamos una plantilla para crear la base de datos, en nuestro caso una de propósito general.
5. 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.
 - sary.fclld.local
 - sarys
6. Lo siguiente es configurar la base de datos con enterprise manager.
7. Elegimos las contraseña para los usuarios que crea oracle.
8. Seleccionamos el mecanismo de almacenamiento de la base de dato. Por ahora solo file system.
9. 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.
10. Elegimos las opciones de recuperación para la base de datos.
 - {ORACLE_BASE}/flash_recovery_area
11. Instalamos los esquema de ejemplos. Nos servirá para las practicas.
12. La siguiente pantalla nos pedirá datos como la memoria, tamaño, juego de caracteres y modo de conexión Los valores por defecto nos servirán
13. Luego veremos el almacenamiento de la base de datos.
14. Luego seleccionamos la opciones para crear la base de datos. Cotejamos crear.
15. Por ultimo veremos el resumen de los detalles de la base de datos a crear.

Arquitectura OFA

Oracle Flexible Architecture, la arquitectura flexible de oracle es una estructura de directorios que propone oracle para la fácil localización de cualquier fichero del servidor. OFA propone los puntos de montaje de los directorios, además 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 sería:



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

```
ORACLE_BASE = /opt/app/oracle
ORACLE_HOME = $ORACLE_BASE/product/10.1.0
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
```


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:
 1. Arranque y parada
 2. Parámetros del init sin valores por defecto,
 3. Arranque de los procesos background,
 4. Cambio de fichero redolog (log switch),
 5. Creación de tablespaces y segmentos de rollback,
 6. Comandos alter (alter database, alter tablespace, etc),
 7. Errores del tipo ORA600 y otros tipos de errores que indican falta de espacio en los sistemas /opt/app/oracle/product/11.1.0/db_1/network/admin/sqlnet.ora (llenado de tablas, índices, tablespaces, etc).

Practica:

Comprobar los requerimientos HW y SW para la instalación de Oracle10g sobre Linux.

1. Verificar la cantidad memoria Ram y Swap
2. Verificar el espacio en discos
3. Versión del Kernel
4. Paquetes necesarios
5. Acceso a los ejecutables
6. Entorno gráfico
7. Parámetros del kernel
8. Puntos de montaje para el sw y las bases de datos
9. Grupos dba y oinstall
10. Usuario oracle con grupo primario dba y secundario oinstall
11. Directorios /opt/app/oracle y /opt/oradata, etc.
12. Variables de entorno del usuario oracle justo antes de instalar
13. Identificar los procesos que componen instancia
14. Comprobar cuánto ocupa el sw de Oracle10g.
15. Buscar el fichero de log de la instalación y echarle un vistazo.
16. Ver el tamaño de la SGA de la BD
17. Comprobar ficheros que componen la BD y ubicarlos en la estructura OFA

Introducción SQL

Objetivos: En esta capítulo explicamos una breve introducción al estandar 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 se consulta estructurado surgido de un proyecto de investigación de IBM para el acceso de las base 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 ademas 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 description lenguaje)**, lenguaje de definición de datos, incluye ordenes para definir, modificar o borrar las tablas en las que se almacenan los datos y de las relaciones entre estas.
- **DCL (data control lenguaje)**, 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 lenguaje)**, lenguaje de manipulación de datos, nos permite recupera los datos almacenados en la base de datos y tambien incluye ordenes permitir al usuario actualizar, insertar o modificar datos existentes en la tabla.

Lenguaje de Descripción de Datos (DDL):

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

```
CREATE TABLE MiTabla (  
    a number,  
    b varchar2(10)  
);
```

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

```
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:

```
ALTER TABLE MiTabla ADD COLUMN (c number);  
ALTER TABLE MiTabla ADD CONSTRAINT a UNIQUE (clave)
```

Lenguaje de Manipulación de Datos (DML):

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

```
SELECT col1, col2 FROM mitabla;  
SELECT ename, sal, hiredate FROM emp;
```

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

```
INSERT INTO MiTabla (col01, col02) VALUES (12, 'Suarez');  
INSERT INTO dept (deptno, dname) VALUES (45, 'jaguar');
```

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

```
UPDATE MiTabla SET (col01, col02) = (15, 'Paredes');  
UPDATE emp  
    SET (job, deptno) = (select job, deptno from empno = 7499)  
    WHERE empno = 7698;
```

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

```
DELETE from MiTabla where col02='Paredes';  
DELETE FROM emp  
    WHERE deptno=20;
```

Clausulas

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

```
SELECT * FROM dept;
```

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

```
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 conector y, evalúa dos condiciones y devuelve un valor de verdad solo si ambos son ciertos.

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

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

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

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

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

Operadores de Comparación:

- > mayor que...

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

- < menor que...

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

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

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

- **BETWEEN**. Usado para especificar un intervalo de valores.

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

- **LIKE**. Usado en la comparación de un modelo.

```
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.

```
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.

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

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

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

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

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

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

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

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

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

Practica

1. Crear una tabla llamada **dba1** con los siguientes campos:

```
Id_Dba PK
Nombre
Teléfono
Edad
Profesión
Correo
```

2. Crear una tabla llamada **server1** con los siguientes campos:

```
Id_Server
Dba FK de dba1 (Id_Dba)
Nom_Server
IP_Server
Desc_Server
Fecha_alta
```

3. Inserte al menos 10 registros en Ambas tablas
4. Seleccione los dba que su nombre empiece con J
5. Actualice la profesión de una dba
6. Borre el servidor que su ip termine en 10
7. Diga la cantidad de servidores por dba
8. Seleccione los server que su fecha alta sea mayor a la fecha actual
9. Diga la cantidad de servidores
10. Liste el último dba en administrar un servidor

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 línea (off-line).

Arrancar la Base Datos

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, evitable, cuyo nombre sigue el patrón:

`init$ORACLE_SID.ora.`

- **spfile:** es un fichero binario, no evitable pero visualizable, cuyo nombre sigue el patrón:

`spfile$ORACLE_SID.ora.`

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

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/dbs/sanjuaninit.ora`
- **STARTUP NOMOUNT:** para crear la base de datos.
- **STARTUP MOUNT:** para renombrar datafiles, activar.
- **ARCHIVELOG:** 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 SESSION`.
- **STARTUP FORCE:** hace `SHUTDOWN ABORT` y arranca la BD.

Viendo el proceso de arranque:

```
[oracle@sanjuan ~]$ sqlplus sys/kikla as sysdba@sanjuan/sanjuan
SQL> startup nomount
```

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

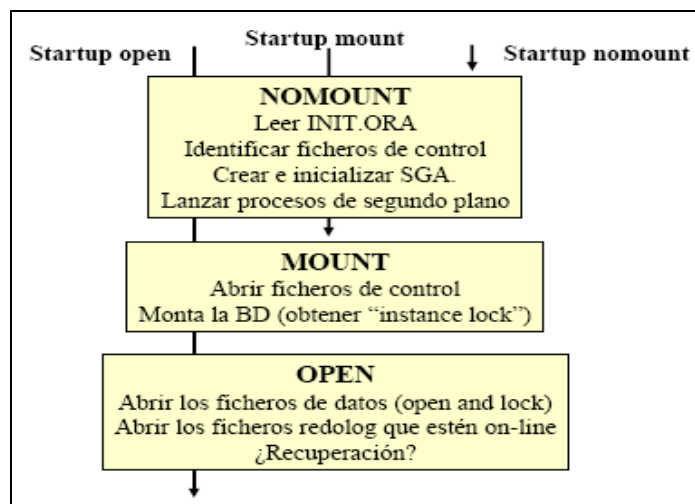
```
SQL> alter database mount;
SQL> alter database open database opened
```

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).

Archivos Esenciales para iniciar la Base Datos

1. `init.ora`
2. `spfilesanjuan.ora`

Practica:

1. Localizar el fichero **init.ora** (y **spfile** si existe) de nuestra BD
2. Ver parámetros: **db_block_size**, **sga_target**, **ga_max_size**, **shared_pool_size**, **db_cache_size**, **log_buffer**, **processes**.
3. Arrancar la BD y comprobar valor de parámetros, ¿ha tomado el nuevo valor? ¿por qué?
4. Parar la BD y arrancar.
5. Parar la BD de forma normal, con algún usuario conectado. ¿Qué ocurre? ¿y qué sucede cuando todos los usuarios se desconectan?
6. Repetir la parada de la BD, estando conectado algún usuario, de forma que no espere a que se desconecten.
7. Arranca sólo la instancia (**NOMOUNT**) y consulta algún parámetro (**sga_target**, **db_cache_size**, **shared_pool_size**, **log_buffer**, **processes**, etc). Qué ocurre al acceder a **V\$CONTROLFILE**.
8. Ahora monta la BD y vuelve a consultar **V\$CONTROLFILE**. Qué sucede al leer **DBA_USERS**
9. Impedir las conexiones de usuarios, de modo que el **DBA** sí pueda conectarse. Intenta conectarte como **scott/tiger**. Volver a permitir conexiones de usuarios.

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.

¿Que 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 **DICTIONARY_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, ALL_SEQUENCES, ALL_VIEWS,

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

USER_TABLES, USER_SEQUENCES, ...

- 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 el se refleja en el diccionario. También se puede ver la estructura de una tabla del diccionario:

```
DESCRIBE ALL_TABLES;  
DESCRIBE ALL_INDEXES;  
DESCRIBE ALL_SEQUENCES;
```

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:

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

Para los índices:

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

Para las secuencias:

```
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:

```
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.

```
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.

```
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.

```
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

```
SELECT * FROM V$CONTROLFILE
```

Para ver el nombre de la instancia y su estado

```
SELECT INSTANCE_NAME, DATABASE_STATUS
FROM V$INTANCE;
```

Para obtener información acerca de los procesos iniciados

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

Para conocer el nombre de la base de datos actual;

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

Practica

Consultar las siguientes vistas en la base de datos:

```
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 (Usuarios de la BD).
```

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 optimizan de los recursos que estos procesos puedan requerir en un momento dado.

SMON

Este proceso se encarga de recuperar 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.

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 nombretsp BEGIN BACKUP.
```

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.

LGWR

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

- 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.

ARCH

Este es un 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”.

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:

- **-a** para mostrar los procesos de otros usuarios.
- **-f** para ver los procesos en forma de arbol
- **-u** para ver nombre de usuario que lo inicio y tiempo del proceso
- **-x** procesos que no poseen terminales asociados a estos (como los deamons)
- **-U** para ver los procesos de un usuario en particular

```
[oracle@sanjuan ~]$ ps -aux  
[oracle@sanjuan ~]$ ps -aux|grep ora
```

****NOTA:** `grep` es un comando nos permite filtrar lo que buscamos, en este caso el mostrara todos los procesos en cuya ocurrencia aparezca la palabra “ora”.

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.

Encender el listener y enviarlo a background:

```
[oracle@sanjuan ~]$ lsnrctl &  
[1] 3533
```

Ponerlo en primer plano nuevamente. Notese que tomamos el numero de jobs “[1]” para subirlo a primer plano. Para saber cuantos jobs hemos iniciado suficiente con ejecutar el comando jobs con la opción -l

```
[oracle@sanjuan ~]$ jobs -l  
[1]+  3533 Running                  lsnrctl &  
[oracle@sanjuan ~]$ fg %1
```

Comando top:

Una utilidad muy usada y muy versátil para el monitoreo en tiempo real del estado de los procesos y de otras variantes del sistema es el programa llamado `top`, se ejecuta desde la línea de comandos, es interactivo y por defecto se actualiza cada 3 segundos

Practica:

1. Ver todos los procesos del usuario Oracle
2. Ver cantidad de memoria de los procesos Oracle
3. Identificar los procesos que componen instancia.
4. También lo podemos hacer consultando la vista dinámica `V$PROCESS` (o directamente en la vista `V$BGPROCESS`), en la BD en cuestión
5. Ver la prioridad de los procesos de Oracle.

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

```
CREATE DATABASE ;
```

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:

```
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:

```
ALTER DATABASE BACKUP CONTROLFILE TO  
'/opt/app/oracle/oradata/sanjuan/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 "SARYS" NORESETLOGS
NOARCHIVELOG
MAXLOGFILES 32
MAXLOGMEMBERS 2
MAXDATAFILES 1000
MAXINSTANCES 1
MAXLOGHISTORY 292
LOGFILE
GROUP 1 '/opt/app/oracle/oradata/sanjuan/redo01.log' SIZE 10M,
GROUP 2 '/opt/app/oracle/oradata/sanjuan/redo02.log' SIZE 10M,
GROUP 3 '/opt/app/oracle/oradata/sanjuan/redo03.log' SIZE 10M
DATAFILE
'/opt/app/oracle/oradata/sanjuan/system01.dbf',
'/opt/app/oracle/oradata/sanjuan/undo_rbs01.dbf',
'/opt/app/oracle/oradata/sanjuan/sysaux01.dbf',
'/opt/app/oracle/oradata/sanjuan/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/sanjuan_ora_26373.trc
SQL> alter database backup controlfile to
'/opt/app/oracle/oradata/control_copia1.ctl';
Modified Database.
SQL> select name from v$controlfile;
```

Practica:

1. Localizar el fichero de control desde el S.O y desde la propia BD.
2. Añade una copia al fichero de control de la BD en /u04/oradata/\$ORACLE_SID.
3. Sacar una copia de seguridad del fichero de control.
4. Diga el tamaño de los ficheros de control en el S.O.

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.

Usuarios Administradores

Cuando creamos una BDO se crean automáticamente los usuarios SYS y SYSTEM, ambos con el rol DBA. El SYS, cuya clave inicial es change_on_install, es el propietario del DD y habitualmente se usa para arrancar y parar la base de datos, así como para modificar los componentes de la misma (como instalar nuevas opciones). El usuario SYSTEM, con clave inicial manager, es el DBA por excelencia. Se usará para las tareas administrativas habituales: alta de usuarios, creación de tablespaces, etc.

Para conectar como SYS:

```
CONNECT SYS AS SYSDBA
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 válida de usuarios. Para acceder a la misma un usuario debe ejecutar un aplicación y conectarse a la instancia usando un nombre válido 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):

```
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.

```
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** (¡Error!).
- **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** (¡Error!).
- **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.

```
Create user PERPINAN
identified by PASSWORD
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**”.

```
ALTER USER <usuario>
IDENTIFIED BY <contraseña>/EXTERNALLY
DEFAULT TABLESPACE <espacio>
TEMPORARY TABLESPACE <espacio>
QUOTA <xx>K/UNLIMITED ON <espacio>
DEFAULT 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.

```
DROP USER <usuario>;
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.

```
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**. Numero 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.

```
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.

```
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.

```
DROP PROFILE <perfil>;  
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 privilegios 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**.

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

```
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 otorgo. 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 adquieren 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 auditoría 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.

```
CREATE ROLE <rol> IDENTIFIED BY <contraseña>;  
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 fclد;  
SQL> grant create session to fclد;
```

Ahora asignamos a un usuario al rol creado:

```
SQL> gran fclد 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.

```
ALTER ROLE <rol> NOT IDENTIFIED/ IDENTIFIED BY <contraseña>;  
GRANT <rol> TO <usuario>/<rol>/PUBLIC;  
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**.

```
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.

```
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**.

```
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.

Practica:

1. Buscar en la documentación en línea y en bd el contenido de las vistas:
2. Conectarse como usuario **SYSTEM** a la base y crear un usuario llamado “**administrador**” autenticado por la base de datos.
3. Averiguar qué privilegios de sistema, roles y privilegios sobre objetos tiene concedidos el usuario “**administrador**”.
4. Otorgar el privilegio “**CREATE SESSION**” al usuario “**administrador**” e intentar de nuevo la conexión sqlplus.
5. Conectarse como usuario “**administrador**” y crear un usuario llamado “**prueba00**” que tenga como “**tablespace**” por defecto **USERS** y como “**tablespace**” temporal **TEMP**; asignar una cuota de 0K en el “**tablespace**” **USERS**. ¿Es posible hacerlo?.
6. Conectado como usuario **SYSTEM**, otorgar el privilegio “**create user**” al usuario “**administrador**” y repetir el ejercicio anterior.
7. Averiguar que usuarios de la base de datos tienen asignado el privilegio “**create user**” de forma directa, ¿qué vista debe ser consultada?.
8. Crear un rol llamado “**ADMIN**”, asignarle los privilegios “**create session**”, “**create user**” y “**CREATE ROLE**”. Asignarlo al usuario administrador.
9. Consultar los privilegios de sistema que tiene asignados de forma directa el usuario “**administrador**”, revocarlos y asignarle el rol “**admin.**”.
10. Consultar que perfiles tiene asignados cada usuario de la base de datos.
11. Crear un perfil llamado “**DESARROLLO**” con las siguientes especificaciones:

Sessions_per_user	2
Cpu_per_session	unlimited
Cpu_per_call	6000
Connect_time	480
Idle_time	2
Failed_login_attempts	2
Password_life_time	120

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.

Modos de Operación de la DB

Modo ARCHIVELOG

El archivado de los “**redo log**” está habilitado. Un grupo de “**redo**” no puede reutilizarse por **LGR** 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.

Por defecto, la BD se crea en modo **NOARCHIVELOG** (con **CREATE DATABASE**). Si activamos el modo **ARCHIVELOG** se irán archivando los ficheros redo log conforme se llenan (cada vez que ocurre un “**log switch**”).

LOG_ARCHIVE_START=TRUE, activa archivado automático (en 10g no hace falta). El proceso ARCH irá archivando el grupo redo log lleno, después de cada “log switch”, en el directorio indicado por el parámetro **LOG_ARCHIVE_DEST** (por defecto **\$ORACLE_HOME/dbs/arch**).

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

Cada vez que se archiva un redo log, en el fichero de control se guarda el nombre del redolog archivado, número de secuencia, y números **SCN** más alto y más bajo.

El redo log que se ha llenado no puede reutilizarse hasta que ocurra un checkpoint y haya sido copiado por el proceso **ARCH**.

Poner BD en modo **ARCHIVELOG**: **SHUTDOWN**, backup (por seguridad),

Configurar **log_archive_dest** en el **INIT**, **STARTUP MOUNT**, activar archivado (**ALTER DATABASE ARCHIVELOG;**), abrir BD (**ALTER DATABASE OPEN;**), parar BD, y hacer backup (pues ha cambiado el fichero de control y la copia anterior ya no nos sirve).

Copia logica export/import

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 da datos completa. Esta utilidad permite almacenar tanto las definiciones de las tablas como los datos que contengan. También permite el intercambio de información entre maquinas, bases de datos o versiones del servidor de Oracle. Es muy útil para proporcionar protección contra las pérdidas 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:

```
exp keyword = valor
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

```
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:

```
imp keyword = valor  
imp keyword = (valor1,valor2,...)
```

y 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 una 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.

```
imp estudiante01/kikla file=hr.dmp full=yes
```

Practica:

1. Comprobar que usuarios y roles tienen asignado el rol “**exp_full_database**”.
2. ¿Podría un usuario que tuviera asignado el rol “dba” realizar una exportación total de la base de datos?, ¿por qué?
3. Como usuario “**scott**”, realizar una exportación de su esquema. la exportación se llamara “**scott_scott03.dmp**” y el registro en el fichero log “**scott_scott03.log**”. ¿Se ha realizado con éxito la exportación?
4. Como usuario “**scott**”, intentar realizar una exportación total de la base de datos. La exportación en el fichero “**total_scott04.dmp**” y el registro en el fichero “**total_scott04.log**”. ¿Se ha realizado con éxito la exportación?, ¿por qué?
5. Conectado como usuario “**scott**”, realizar una exportación de sus tablas “**emp**” y “**dept**”. Los parámetros se almacenarán en un fichero llamado “**par_scott05**”, la exportación en el fichero “**tablas_scott05.dmp**” y el registro en el fichero “**tablas_scott05.log**”. ¿Hubo éxito en la exportación?

El contenido del fichero de parámetros será:

```
buffer=102400
file=/export/CURS001/tablas_scott05.dmp
log=/export/CURS001/tablas_scott05.log
tables=(EMP, DEPT)
compress=y
```

6. Comprobar que usuarios y roles tienen asignado el rol “**imp_full_database**”
7. Asignar el rol “**imp_full_database**” al usuario “administrador”.
8. Como usuario “**administrador**”, importar la tabla “**scott.emp**” desde el fichero “**scott_scott03.dmp**”. Crear un fichero de parámetros llamado “**par_imp_admin01**” (parámetros **file**, **fromuser** y **tables**), el registro se guardara en el fichero “**imp_tabla_emp**”. ¿Termina con éxito la importación?. El contenido del fichero de parámetros de importación es:

```
buffer=102400
file=/home/oracle/dba/scott_scott03.dmp
log=/home/oracle/dba/imp_tabla_emp.log
tables=(EMP)
fromuser=scott
touser=scott
```

Herramientas GUI para Oracle

Objetivos: Aprender el manejo y aprovechar la flexibilidad que brindan las dos herramientas principales que tiene a la mano todo administrador de base de datos de Oracle sobre GNU/Linux: Tora y Emacs.

Tora

Tora fue originalmente escrita por Henrik Johnson, con las adiciones y mejoras de la comunidad de código abierto, y ahora está únicamente desarrollado como un proyecto Open Source. Cualquier referencia a un juicio o la compra de licencias de licencias se aplican exclusivamente a los binarios de MS Windows con versiones anteriores a 1.3.15. Las versiones mas recientes se son 100% con licencia GPL y puede ser utilizado libremente para cualquier fin.

Paquetes Necesarios

- tora-1.3.16-1rhel4.i386
- tora-oracle-1.3.16-1.rhel.i386
- oracle-instantclient-basic-10.2.0.1-1.i386.rpm
- oracle-instantclient-devel-10.2.0.1-1.i386.rpm
- oracle-instantclient-sqlplus-10.2.0.1-1.i386.rpm

Instalación de Clientes

1. Instalar los paquetes RPM de los clientes

```
oracle-instantclient-basic-10.2.0.1-1.i386.rpm
oracle-instantclient-devel-10.2.0.1-1.i386.rpm
oracle-instantclient-sqlplus-10.2.0.1-1.i386.rpm
```

2. Luego Instalamos los paquetes convertidos

```
rpm -Uvh oracle-instantclient-basic-10.2.0.1-1.i386.rpm
rpm -Uvh oracle-instantclient-devel-10.2.0.1-1.i386.rpm
rpm -Uvh oracle-instantclient-sqlplus-10.2.0.1-1.i386.rpm
rpm -Uvh tora-1.3.16-1rhel4.i386
rpm -Uvh tora-oracle-1.3.16-1.rhel.i386
```

3. Copiamos las variables en el archivo **.bashrc**:

```
# Configuración de ORACLE
ORACLE_HOME=/usr/lib/oracle/10.2.0.1/client
PATH=$ORACLE_HOME/bin:$PATH
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export ORACLE_HOME
export PATH
export LD_LIBRARY_PATH
```

4. Copiamos las librerías de `/usr/lib/oracle/10.2.0.1/client/lib` a `/usr/lib/` (Si es necesario)

```
cp /usr/lib/oracle/10.2.0.1/client/lib /usr/lib
```

5. Agregamos la ruta `/etc/profile`

```
PATH=$PATH:/usr/bin ; export PATH
```

6. Creamos las carpetas **network/admin**

```
mkdir -p /usr/lib/oracle/10.2.0.1/client/network/admin
```

7. Copiamos el **tnsname.ora** y el **listener.ora** del server

```
scp listener.ora tnsname.ora cliente@ip:ruta
```

8. Problemas la conexión con el **sqlplus**

```
sqlplus hr/kikla@//150.1.1.31/GNU.Linux
```

****Nota:** el nombre de la BD tiene que ser el Global Configuración

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-INITIO (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.