

WWW.ACL.EDU.DO



ACADEMIA
CODIGO
LIBRE

LIBROS PRACTICAS MANUALES SCRIPTS VIDEOS

Practica GCOA/DBA-I



Aprendiendo Practicando
GNU Certified Oracle Administrator
Ing. José Paredes

www.codigolibre.org

Práctica Administración de Base de Datos I

Academia Código Libre
Septiembre 2014.-

Valentín Jiménez
GNU/Instructor GCOA

Documentación realizada por:

Ing. José Paredes
GNU/Instructor GCOA

Meidy Alvarez
GNU/Instructora GCOA



Aprendiendo Practicando GCOA/DBA-I

En esta práctica usted aprenderá lo necesario para usted manejar Oracle sobre GNU/Linux y muchas cosas más a nivel práctico: Comandos Unix/Linux, Editores (Vim-Emacs), Manejo de Paquetes (Rpm-Deb-Tar), Variables, Localizar la Base Datos, Los Objetos de la Bases Datos, Nombre de los Objetos, Diccionario de Datos, Iniciar y Detener una Instancia, Los Procesos, Control Files, DataFiles, TableSpaces, Archived Red Logs. Si completas el 100% de estos ejercicios podremos garantizar su aprendizaje.

Para iniciar Esta practica usted debe tener Oracle 11g Instalado.

Los diferente Prompt que tendrá:

[Oracle@oraservacl ~]\$ Usuario Dbá en el S.O.

[root@oraservacl ~]# Administrador del S.O.

SQL> Para Ejecutar Sentencias SQL en la BD.

0) Verificamos que distribución, estamos utilizando.

```
[Oracle@oraservacl ~]$ cat /etc/issue
```

```
[Oracle@oraservacl ~]$ lsb_release -d
```

1) Ver el tiempo qué tiene nuestro server encendido.

```
[Oracle@oraservacl ~]$ uptime
```

2) Verificamos la versión del kernel y la arquitectura.

```
[Oracle@oraservacl ~]$ uname -a
```

3) Determinamos que arquitectura tenemos instalada.

```
[Oracle@oraservacl ~]$ arch
```

4) Determinamos que arquitectura tenemos instalada.

```
[Oracle@oraservacl ~]$ uname -m
```

5) Determinamos que arquitectura tenemos instalada y que arquitectura soporta nuestro hardware.

```
[Oracle@oraservacl ~]$ lscpu
```

6) Verificamos la cantidad de memorias disponibles.

```
[Oracle@oraservacl ~]$ free -tm
```

7) Liste los Sistemas de Archivos montados actualmente con su espacio disponible.

```
[Oracle@oraservacl ~]$ df -TH
```

8) Obtenga información del CPU de su server actual.

```
[Oracle@oraservacl ~]$ cat /proc/cpuinfo
```

9) Liste las particiones actuales que tiene su sistema.

```
[Oracle@oraservacl ~]$ su -
```

```
[root@oraservacl ~]# fdisk -l
```

10) Si tenemos una instalación LVM, para ver los volúmenes lógicos.

```
[root@oraservacl ~]# lvscan
```

11) Liste el contenido de los siguientes directorios.

```
/opt/  
/opt/app  
/opt/app/oracle  
/opt/app/oracle/product/11.2.0/db_1/  
$ORACLE_BASE/oradata
```

12) Determine el tamaño de los siguientes Directorios.

```
[root@oraservacl ~]# du -sch /opt
```

```
[root@oraservacl ~]# du -sch /home
```

```
[root@oraservacl ~]# du -sch /var
```

```
[root@oraservacl ~]# du -sch /boot
```

```
[root@oraservacl ~]# du -sch /etc
```

13) Desplegar los directorios que ocupan mayor espacio en /opt/app.

```
[root@oraservacl ~]# du -sh /opt/app/* |sort -nr| head -5
```

14) Salir del usuario root.

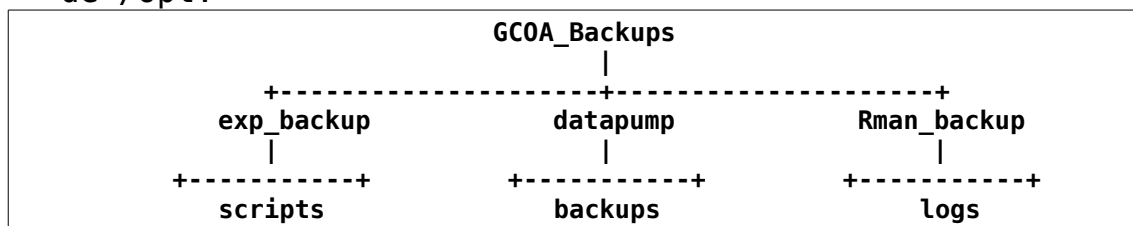
```
[root@oraservacl ~]# exit
```

15) Verifique qué está logueado con el usuario Oracle.

```
[Oracle@oraservacl ~]$ whoami
```

```
[Oracle@oraservacl ~]$ id
```

16) Crear los siguientes directorios con el usuario root dentro de /opt:



```
[Oracle@oraservacl ~]$ su -
```

```
[root@oraservacl ~]# cd /opt && mkdir GCOA_Backups && cd GCOA_Backups && mkdir exp_backup datapump Rman_backup && mkdir exp_backup/scripts && mkdir datapump/backups && mkdir Rman_backup/logs
```

17) Vamos a ver la estructura de directorio en forma jerárquica.

```
[root@oraservacl GCOA_Backups]# tree
```

18) Listar en orden descendente, fecha y detalle el directorio GCOA_Dba.

```
[root@oraservacl GCOA_Backups]# ls -ltrh ; cd
```

19) Verifique si el usuario "Oracle" está creado.

```
[root@oraservacl ~]# cat /etc/passwd |grep oracle
```

20) Mostramos el identificador y grupos del usuario oracle.

```
[root@oraservacl ~]# id oracle
```

21) Verifique si los grupos "oinstall y Db" están creados.

```
[root@oraservacl ~]# cat /etc/group |grep oinstall
```

```
[root@oraservacl ~]# cat /etc/group |grep db
```

22) Verifiquemos si el usuario oracle pertenece a los grupos "oinstall y Db".

```
[root@oraservacl ~]# su - oracle
```

```
[Oracle@oraservacl ~]$ groups
```

23) Vamos a cambiarle el dueño y grupo al directorio GCOA_Backups.

```
[Oracle@oraservacl ~]$ su -
```

```
[root@oraservacl ~]# chown -R oracle.oinstall /opt/GCOA_Backups
```

GNUNota: Comprobar que los mismos fueron cambiados con el comando "STAT".

24) Vamos a ver los parámetros del kernel.

```
[root@oraservacl ~]# cat /etc/sysctl.conf
```

25) Vamos a ver el archivo que contiene los límites de los recursos utilizados por los usuarios.

```
[root@oraservacl ~]# cat /etc/security/limits.conf
```

26) El programa login define su nombre de servicio como login e instala el archivo de configuración PAM.

```
[root@oraservacl ~]# cat /etc/pam.d/login
```

27) Determinamos el estatus de SELinux.

```
[root@oraservacl ~]# sestatus
```

```
[root@oraservacl ~]# getenforce
```

```
[root@oraservacl ~]# cat /etc/selinux/config
```

28) SELinux está habilitado lo deshabilitemos.

```
[root@oraservacl ~]# setenforce 0
```

29) Saber el estatus del servicio iptables.

```
[root@oraservacl ~]# service iptables status
```

30) Detenemos iptables.

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

31) Revisemos los puertos abiertos.

```
[root@oraservacl ~]# nmap localhost
```

GNUNota: Si no tiene nmap instalado debe instalarlo.

32) Para mostrar todos los datos de conexiones abiertas.

```
[root@oraservacl ~]# netstat -putona
```

33) Si quiero saber que está pasando por el puerto 1521.

```
[root@oraservacl ~]# netstat -putona |grep ':1521'
```

34) Verifique si estos paquetes se encuentran instalados.

```
[root@oraservacl ~]# rpm -q compat-libstdc++-33 elfutils-libelf-devel glibc-devel-2.5 gcc gcc-c++ libaio-devel libstdc++-devel unixODBC unixODBC-devel sysstat binutils elfutils-libelf glibc glibc-common libaio libgcc libstdc++ make
```

35) Vamos a imprimir las variables de entorno, logueado con el usuario "oracle".

```
[root@oraservacl ~]# su - oracle
```

```
[oracle@oraservacl ~]$ echo $Tab Tab
```

36) Vamos a imprimir las variables de ambiente, logueado con el usuario "oracle".

```
[oracle@oraservacl ~]$ env
```

37) Comprobar la asignación de variables de entorno necesarias para conectarnos a la BD.

```
[oracle@oraservacl ~]$ echo $ORACLE_HOME
```

```
[oracle@oraservacl ~]$ echo $ORACLE_SID
```

```
[oracle@oraservacl ~]$ echo $ORACLE_BASE
```

```
[oracle@oraservacl ~]$ echo $PATH
```

38) Asignar la variable NLS_LANG para nuestro país y comprobar el cambio en las respuestas de Oracle desde sqlplus.

```
[oracle@oraservacl ~]$ export NLS_DATE_FORMAT='dd-mm-yyyy hh24:mi:ss'
```

```
[oracle@oraservacl ~]$ export NLS_LANG=american_america
```

```
[oracle@oraservacl ~]$ echo $NLS_LANG
```

```
[oracle@oraservacl ~]$ echo $NLS_DATE_FORMAT
```

39) Vamos a imprimir las variables del Sistema.

```
[oracle@oraservacl ~]$ set
```

40) Imprimir las variables cargadas al usuario oracle.

```
[oracle@oraservacl ~]$ cat .bashrc
```

```
[oracle@oraservacl ~]$ cat .bash_profile
```

41) Vamos a imprimir todos los alias.

```
[oracle@oraservacl ~]$ alias
```

42) Vamos agregar algunos alias necesarios para un DBA.

```
[oracle@oraservacl ~]$ alias sys='${ORACLE_HOME}/bin/sqlplus / as sysdba'
```

```
[oracle@oraservacl ~]$ alias ohome='cd $ORACLE_HOME'
```

```
[oracle@oraservacl ~]$ alias c='clear'
```

```
[oracle@oraservacl ~]$ alias dbs='cd ${ORACLE_HOME}/dbs'
```

```
[oracle@oraservacl ~]$ alias aler="locate alert_${ORACLE_SID} | xargs grep ORA- |less"
```

GNUNota: Agregar estos alias permanente y probar cada uno.

43) Veamos el nombre de nuestro server.

```
[Oracle@oraservacl ~]$ hostname
```

44) Verificamos si tenemos el nombre calificado en /etc/hosts.

```
[Oracle@oraservacl ~]$ cat /etc/hosts
```


45) Comprobamos que resolvemos nuestro nombre.

```
[Oracle@oraservacl ~]$ ping -ac 3 $ORACLE_SID
```

46) Vemos el IP de nuestro server.

```
[Oracle@oraservacl ~]$ ifconfig
```

47) Vemos el IP de nuestro server.

```
[Oracle@oraservacl ~]$ hostname -I
```

48) Vemos nuestras tarjetas de red con sus red.

```
[Oracle@oraservacl ~]$ ip addr show
```

49) Para saber cual es su HOME.

```
[Oracle@oraservacl ~]$ echo $HOME
```

50) Para saber la ruta donde usted se encuentra.

```
[Oracle@oraservacl ~]$ pwd
```

51) Actualizamos la BD del comando locate.

```
[Oracle@oraservacl ~]$ su -
```

```
[root@oraservacl ~]# updatedb
```

```
[root@oraservacl ~]# exit
```

52) Localizar los ficheros init.ora.

```
[oracle@oraservacl ~]$ locate init.ora
```

53) Para buscarlo utilizando el comando find.

```
[oracle@oraservacl ~]$ find /opt/app/ -iname "*init.ora*"
```

54) Listar el directorio donde se almacenan los archivos de inicializacion.

```
[oracle@oraservacl ~]$ ls -l $ORACLE_HOME/dbs/
```

55) Listar el directorio donde se encuentran los ficheros init.ora.

```
[oracle@oraservacl ~]$ ls -l $ORACLE_BASE/admin/$ORACLE_SID/pfile
```

56) Localizar los archivo init.ora creado desde nuestra instalación.

```
[oracle@oraservacl ~]$ locate init.ora.
```

57) Ver el contenido del init.ora creado después de nuestra instalación.

```
[oracle@oraservacl ~]$ locate init.ora. |xargs cat |less
```

58) Verifique el tamaño de la Base Dato Oracle instalada.

```
[oracle@oraservacl ~]$ du -sch /opt/app
```

59) Localizar los Data Files.

```
[oracle@oraservacl ~]$ locate .dbf
```

60) Buscar los Data Files en su S.O.

```
[oracle@oraservacl ~]$ find /opt/app/ -iname "*.dbf*"
```

61) Saber que tamaño ocupan los Datafiles en el S.O.

```
[oracle@oraservacl ~]$ du -sch $ORACLE_BASE/oradata/$ORACLE_SID/*.dbf
```

62) Saber que tamaño ocupan los Datafiles en el S.O.

```
[oracle@oraservacl ~]$ locate .dbf |grep .dbf |xargs du -sch
```

63) Localizar los Control Files.

```
[oracle@oraservacl ~]$ locate .ctl
```

64) Buscar los Control Files en su S.O.

```
[oracle@oraservacl ~]$ find /opt/app/ -iname "*.ctl*"
```

65) Saber que tamaño ocupan los Control Files en el S.O.

```
[oracle@oraservacl ~]$ du -sch $ORACLE_BASE/oradata/$ORACLE_SID/*.ctl
```

Nota: El \$ORACLE_SID puede estar en mayúscula.

66) Saber que tamaño ocupan los Control Files en el S.O.

```
[oracle@oraservacl ~]$ locate .ctl |grep .ctl |xargs du -sch
```

67) Localizar los archivos de Redo Log.

```
[oracle@oraservacl ~]$ locate .log |grep redo
```

68) Buscar los Redo Log en su S.O.

```
[oracle@oraservacl ~]$ find /opt/app/ -iname "*redo*"
```

69) Saber que tamaño ocupan los Redo Log en el S.O.

```
[oracle@oraservacl ~]$ find /opt/app/ -iname "*redo*" | xargs du -sch
```

70) Saber que tamaño ocupan los Redo Log en el S.O.

```
[oracle@oraservacl ~]$ locate redo |grep .log | xargs du -sch
```

71) Localizar los archivo de alerta.

```
[oracle@oraservacl ~]$ locate alert_${ORACLE_SID}
```

72) Viendo el contenido el archivo de alerta.

```
[oracle@oraservacl ~]$ locate alert_${ORACLE_SID} |xargs cat |less
```

73) Identificar los procesos que componen la instancia.

```
[oracle@oraservacl ~]$ ps aux |grep ${ORACLE_SID} |grep -v grep
```

74) Comprobar los ficheros que componen la BD y ubicarlos en la estructura OFA.

```
[oracle@oraservacl ~]$ ls -lR /opt/app/
```

75) Listar los binarios que tenemos disponibles para el usuario oracle.

```
[oracle@oraservacl ~]$ ls ${ORACLE_HOME}/bin
```

76) Localizar la ruta de la aplicación sqlplus.

```
[oracle@oraservacl ~]$ which sqlplus
```

77) Solicitar la ayuda para utilizar el sqlplus.

```
[oracle@oraservacl ~]$ sqlplus -H
```

78) Vamos a acceder a la BD creada con sqlplus.

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

79) Cambiar el prompt de SQLPLUS.

```
SQL> set sqlprompt "_user'@'_connect_identifier > "
```

80) Si ponemos help index, nos dará un índice con la ayuda de la que disponemos.

```
SQL> help index
```

81) Si deseamos organizar y saber el tiempo de la sentencia realizada en nuestro sqlplus.

```
SQL> SET TIMING ON
```

```
SQL> SET PAGESIZE 37
```

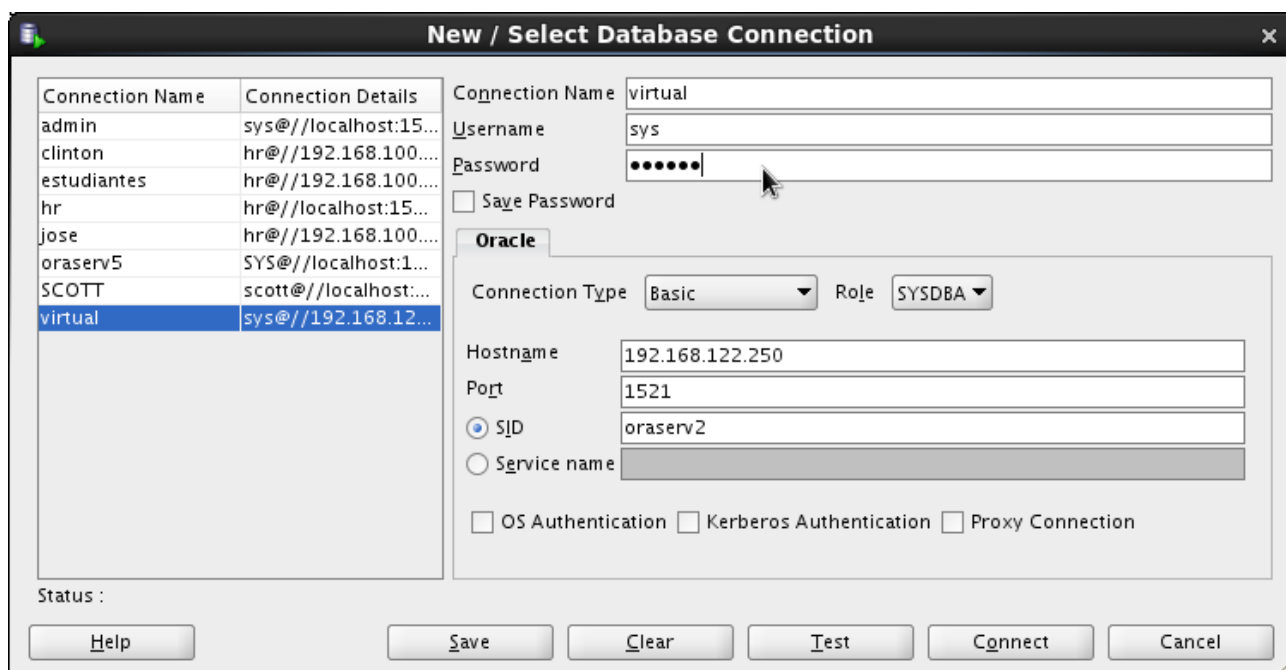
```
SQL> SET LINESIZE 40
```

82) Conocer nuestro catalog.

```
SQL> SELECT *  
      FROM cat;
```

83) Vamos a lanzar el sqldeveloper para realizar los siguientes ejercicios y apreciar mejor su salida.

```
[oracle@oraservacl ~]$ locate sqldeveloper.sh |xargs sh &
```



84) Definición del Diccionario de datos (incluye todas las vistas y tablas de la Base de Datos).

```
SQL> SELECT *  
      FROM dictionary;
```

85) Tablas propiedad del usuario actual.

```
SQL> SELECT *  
      FROM user_tables;
```

86) Listar todas las tablas de la BD.

```
SQL> SELECT *  
      FROM all_tables;
```

87) Todos los objetos propiedad del usuario conectado a Oracle.

```
SQL> SELECT *  
      FROM user_catalog;
```

88) Saber el nombre de su server.

```
SQL> SELECT HOST_NAME  
      FROM v$instance;
```

89) Obtener la IP del servidor de la base de datos Oracle Database.

```
SQL> SELECT utl_inaddr.get_host_address IP  
      FROM dual;
```

90) Saber el nombre de tu instancia.

```
SQL> SELECT INSTANCE_NAME  
      FROM v$instance;
```

91) Saber el nombre de tu instancia desde SQLPLUS.

```
SQL> show parameter instance_name;
```

Nota: No funciona en SQLDEVELOPER.

92) Saber el nombre global de tu instancia.

```
SQL> SELECT PROPERTY_VALUE, DESCRIPTION  
      FROM database_properties  
      WHERE PROPERTY_NAME = 'GLOBAL_DB_NAME';
```

93) Saber el nombre del servicio.

```
SQL> show parameter service_names;
```

Nota: No funciona en SQLDEVELOPER.

94) Saber el estado actual de tu instancia.

```
SQL> SELECT STATUS  
      FROM v$instance;
```

95) Saber la arquitectura que tenemos

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

96) Saber la version de su BD Oracle instalada.

```
SQL> SELECT PROPERTY_VALUE "Version Oracle"  
      FROM database_properties  
      WHERE PROPERTY_NAME = 'NLS_RDBMS_VERSION' ;
```

97) Productos Oracle instalados y la versión.

```
SQL> SELECT *  
      FROM product_component_version;
```

98) Para saber en que modo estamos trabajando en nuestra BD.

```
SQL> SELECT name, log_mode  
      FROM v$database;
```

99) Saber el juego de carácter por defecto.

```
SQL> SELECT PROPERTY_VALUE, DESCRIPTION  
      FROM database_properties  
      WHERE PROPERTY_NAME = 'NLS_CHARACTERSET';
```

100) Ver la estructura de directorios principales del ADR desde SQL, así como el total de subdirectorios y ficheros que lo componen. Consulta también el contenido del fichero de log desde SQL.

```
SQL> SELECT rpad(name,21), rpad(value,70)  
      FROM V$DIAG_INFO;
```

101) Ubicación y nombre del fichero spfile.

```
SQL> SELECT value  
      FROM v$system_parameter  
      WHERE name = 'spfile';
```

102) Ubicación y nombre del fichero spfile.

```
SQL> show parameter spfile;
```

Nota: No funciona en SQLDEVELOPER.

103) Ubicación y número de ficheros de control.

```
SQL> SELECT value  
      FROM v$system_parameter  
      WHERE name ='control_files';
```

104) Ubicación y número de ficheros de control.

```
SQL> show parameter control_files;
```

Nota: No funciona en SQLDEVELOPER.

105) Todos los ficheros de datos y su ubicación.

```
SQL> SELECT name  
      FROM V$DATAFILE;
```

106) Todos los ficheros de datos y su ubicación.

```
SQL> SELECT FILE_NAME, STATUS  
      FROM dba_data_files;
```

107) Todos los Espacios de tablas "TableSpace.

```
SQL> SELECT name  
      FROM V$TABLESPACE;
```

108) Ubicación y número de ficheros log.

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

109) Los Tablespaces y propietarios de los mismos.

```
SQL> SELECT owner, decode(partition_name, null, segment_name,  
      segment_name || ':' || partition_name) name,  
      segment_type, tablespace_name, bytes, initial_extent,  
      next_extent, PCT_INCREASE, extents, max_extents  
      FROM dba_segments  
      WHERE 1=1  
      And extents > 1  
      ORDER BY 9 desc, 3;
```

- 110) Consulta que muestra los tablespaces, el espacio utilizado, el espacio libre y los ficheros de datos de los mismos.

```
SQL> SELECT t.tablespace_name "Tablespace", t.status "Estado",  
          ROUND(MAX(d.bytes)/1024/1024,2) "MB Tamaño",  
          ROUND((MAX(d.bytes)/1024/1024) -  
            (SUM(decode(f.bytes, NULL,0, f.bytes))/1024/1024),2) "MB Usados",  
          ROUND(SUM(decode(f.bytes, NULL,0, f.bytes))/1024/1024,2) "MB Libres",  
          t.pct_increase "% incremento",  
          SUBSTR(d.file_name,1,80) "Fichero de datos"  
        FROM DBA_FREE_SPACE f, DBA_DATA_FILES d, DBA_TABLESPACES t  
        WHERE t.tablespace_name = d.tablespace_name AND  
              f.tablespace_name(+) = d.tablespace_name  
        AND f.file_id(+) = d.file_id GROUP BY t.tablespace_name,  
              d.file_name, t.pct_increase, t.status ORDER BY 1,3 DESC;
```

- 111) Identificar los procesos que componen instancia.

```
SQL> SELECT username, program  
        FROM v$process  
        WHERE background IS NOT NULL;
```

- 112) Identificar los procesos que componen instancia con su descripción.

```
SQL> SELECT name Procesos, description Descripción  
        FROM v$process a, v$bgprocess b  
        WHERE a.ADDR=b.PADDR;
```

- 113) Vista que muestra los parámetros generales de Oracle.

```
SQL> SELECT *  
        FROM v$system_parameter;
```

- 114) Ver el tamaño de la SGA de la BD y las cachés que la componen.

```
SQL> SELECT *  
        FROM v$sgainfo;
```

- 115) Ver el tamaño de la SGA de la BD y las cachés que la componen.

```
SQL> SELECT *  
        FROM v$sgastat;
```

- 116) ¿Cuánto ocupa la Dictionary cache y la Library cache en tu BD? (v\$sgastat).

```
SQL> SELECT *  
        FROM v$sgastat  
        WHERE name like '%cache';
```


117) Ver la actividad de la Library Cache (v\$librarycache).

```
SQL> SELECT namespace, pinhitratio
       FROM v$librarycache;
```

118) Ver las sentencias SQL que guarda la SharedPool(v\$sqlarea).

```
SQL> SELECT SQL_TEXT, PERSISTENT_MEM, EXECUTIONS, LOADS, DISK_READS, CPU_TIME, ELAPSED_TIME
       FROM v$sqlarea
       ORDER BY DISK_READS DESC;
```

119) Localizar el proceso "servidor" asociado a mi sesión (v\$process y v\$session). ¿Es un servidor dedicado o compartido?.

```
SQL> SELECT a.SERVER, a.username dbuser, a.OSUSER, a.PROCESS user_process,
       a.machine, a.terminal, a.program user_program, b.spid server_process,
       b.program server_program
       FROM v$session a, v$process b
       WHERE a.PADDR=b.ADDR;
```

120) Ver el tamaño de la shared pool.

```
SQL> show parameter shared_pool_size
```

121) Ver la shared pool.

```
SQL> SELECT *
       FROM v$sga_dynamic_components
       WHERE component='shared pool';
```

122) Memoria Share_Pool libre y usada.

```
SQL> SELECT name, to_number(value) bytes
       FROM v$parameter WHERE name = 'shared_pool_size'
       union all
       SELECT name, bytes
       FROM v$sgastat WHERE pool = 'shared pool'
       AND name = 'free memory';
```

123) Aciertos de la caché (no debe superar el 1 por ciento).

```
SQL> SELECT sum(pins) Ejecuciones, sum(reloads) Fallos_cache,
       trunc(sum(reloads)/sum(pins)*100,2) Porcentaje_aciertos
       FROM v$librarycache
       WHERE namespace in ('TABLE/PROCEDURE', 'SQL AREA', 'BODY', 'TRIGGER');
```

124) Vista que muestra las conexiones actuales a Oracle.

```
SQL> SELECT osuser, username, machine, program
       FROM v$session
       ORDER BY osuser;
```

125) Vista que muestra el número de conexiones actuales a Oracle agrupado por aplicación que realiza la conexión.

```
SQL> SELECT program Aplicacion, count(program) Numero_Sesiones
      FROM v$session
      GROUP BY program
      ORDER BY Numero_Sesiones desc;
```

126) Vista que muestra los usuarios de Oracle conectados y el número de sesiones por usuario.

```
SQL> SELECT username Usuario_Oracle, count(username) Numero_Sesiones
      FROM v$session
      GROUP BY username ORDER BY Numero_Sesiones desc;
```

127) Propietarios de objetos y número de objetos por propietarios.

```
SQL> SELECT owner, count(owner) Numero
      FROM dba_objects
      GROUP BY owner
      ORDER BY Numero desc;
```

128) Muestra los datos de una tabla especificada (en este caso todas las tablas que lleven la cadena "EMPL0").

```
SQL> SELECT *
      FROM ALL_ALL_TABLES
      WHERE upper(table_name) like '%EMPL0%';
```

129) Muestra los disparadores (triggers) de la base de datos Oracle Database.

```
SQL> SELECT *
      FROM ALL_TRIGGERS;
```

130) Vista que muestra todos los procedimientos almacenados con su dueño.

```
SQL> SELECT owner Dueño, procedure_name
      FROM ALL_procedures;
```

131) Vista que muestra todas las funciones almacenadas con su dueño.

```
SQL> SELECT owner Dueño, object_name
      "Function" FROM ALL_OBJECTS
      WHERE OBJECT_TYPE = 'FUNCTION';
```

132) Roles y privilegios por roles.

```
SQL> SELECT *  
      FROM role_sys_privs;
```

133) Reglas de integridad y columnas a las que afectan.

```
SQL> SELECT constraint_name, column_name  
      FROM sys.all_cons_columns;
```

134) Tablas de las que es propietario un usuario, en este caso "HR".

```
SQL> SELECT DISTINCT TABLE_NAME  
      FROM ALL_ALL_TABLES  
      WHERE OWNER LIKE 'HR';
```

135) Otra forma menos efectiva de ver las tablas de las que es propietario un usuario, en este caso "HR".

```
SQL> SELECT table_owner, table_name  
      FROM sys.all_synonyms  
      WHERE table_owner like 'HR';
```

136) Parámetros de Oracle, valor actual y su descripción.

```
SQL> SELECT v.name, v.value value, decode(ISSYS_MODIFIABLE, 'DEFERRED',  
      'TRUE', 'FALSE') ISSYS_MODIFIABLE, decode(v.isDefault, 'TRUE', 'YES',  
      'FALSE', 'NO') "DEFAULT", DECODE(SES_MODIFIABLE, 'IMMEDIATE',  
      'YES', 'FALSE', 'NO', 'DEFERRED', 'NO', 'YES') SES_MODIFIABLE,  
      DECODE(ISSYS_MODIFIABLE, 'IMMEDIATE', 'YES', 'FALSE', 'NO',  
      'DEFERRED', 'YES', 'YES') SYS_MODIFIABLE , v.description  
      FROM V$PARAMETER v  
      WHERE name not like 'nls%' ORDER BY 1;
```

137) Usuarios de Oracle y todos sus datos (fecha de creación, estado, id, nombre, tablespace temporal,...).

```
SQL> SELECT *  
      FROM dba_users;
```

138) Ver todos los usuarios actuales de la BD.

```
SQL> SELECT *  
      FROM all_users;
```

139) Últimas consultas SQL ejecutadas en Oracle y usuario que las ejecutó.

```
SQL> SELECT distinct vs.sql_text, vs.sharable_mem,
vs.persistent_mem, vs.runtime_mem, vs.sorts,
vs.executions, vs.parse_calls, vs.module,
vs.buffer_gets, vs.disk_reads, vs.version_count,
vs.users_opening, vs.loads,
to_char(to_date(vs.first_load_time,
'YYYY-MM-DD HH24:MI:SS'),'MM/DD HH24:MI:SS') first_load_time,
rawtohex(vs.address) address, vs.hash_value hash_value ,
rows_processed , vs.command_type, vs.parsing_user_id ,
OPTIMIZER_MODE , au.USERNAME parseuser
FROM v$sqlarea vs , all_users au
WHERE (parsing_user_id != 0) AND
(au.user_id(+)=vs.parsing_user_id)
and (executions >= 1) ORDER BY buffer_gets/executions desc;
```

140) Obtener los roles existentes en Oracle Database.

```
SQL> SELECT *
FROM DBA_ROLES;
```

141) Cursores abiertos por usuario.

```
SQL> SELECT b.sid, a.username, b.value Cursores_Abiertos
FROM v$session a,
v$sesstat b,
v$statname c
WHERE c.name in ('opened cursors current')
and b.statistic# = c.statistic#
and a.sid = b.sid
and a.username is not null
and b.value >0
ORDER BY 3;
```

142) Tamaño ocupado por la base de datos.

```
SQL> SELECT sum(BYTES)/1024/1024 MB
FROM DBA_EXTENTS;
```

143) Tamaño de los ficheros de datos de la base de datos.

```
SQL> SELECT sum(bytes)/1024/1024 MB
FROM dba_data_files;
```

144) Espacio ocupado por los usuarios.

```
SQL> SELECT owner, SUM(BYTES)/1024/1024
FROM DBA_EXTENTS MB GROUP BY owner;
```

145) Espacio ocupado por los diferentes segmentos (tablas, índices, undo, rollback, cluster, ...).

```
SQL> SELECT SEGMENT_TYPE, SUM(BYTES)/1024/1024
      FROM DBA_EXTENTS MB
      GROUP BY SEGMENT_TYPE;
```

146) Espacio ocupado por todos los objetos de la base de datos, muestra los objetos que más ocupan primero.

```
SQL> SELECT SEGMENT_NAME, SUM(BYTES)/1024/1024
      FROM DBA_EXTENTS MB
      GROUP BY SEGMENT_NAME
      ORDER BY 2 DESC ;
```

147) Obtener todas las funciones de Oracle: NVL, ABS, LTRIM.

```
SQL> SELECT distinct object_name
      FROM all_arguments
      WHERE package_name = 'STANDARD'
      ORDER BY object_name;
```

Arranque y parada de la BD.

148) Busca y consulta el fichero de alert de la BD. Comprueba el último arranque de la BD .

```
[oracle@oraservacl ~] locate alert_${ORACLE_SID}.log |xargs tail -n 30
```

149) Localizar los ficheros de traza.

```
[oracle@oraservacl ~]$ locate trace |grep .trc |head -10
```

150) Listar los ficheros de traza de forma recursiva.

```
[oracle@oraservacl ~]$ locate trace |grep .trc |xargs ls -ltrh
```

151) Vemos el contenido de un fichero de traza.

```
[oracle@oraservacl ~]$ locate trace |grep .trc |head -1 | xargs cat |less
```

GNUNota: Los ficheros de traza de los procesos background y de usuario, en 11g se generan en el mismo directorio donde ya hemos localizado el fichero de alert, bajo el subdirectorio "diag" situado en el directorio que indica el parámetro DIAGNOSTIC_DEST.

152) El contenido del fichero de log lo puedes sacar desde SQL de la vista X\$DBGALERTEXT.

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

```
SQL> SELECT lpad(' ',lvl,' ')||logical_file file_name  
FROM X$DBGDIREXT;
```

```
SQL> SELECT message_text  
FROM X$DBGALERTEXT WHERE ROWNUM<=20;
```

153) Parar la BD de forma normal, con algún usuario conectado. ¿Qué ocurre? ¿y qué sucede cuando todos los usuarios se desconectan? Repetir la parada de la BD, estando conectado algún usuario, de forma que no espere a que se desconecten.

154) Abrimos una nueva sesión con el usuario system, desde otra terminal.

```
[oracle@oraservacl ~]$ sqlplus system/kikla
```

155) Confirmamos esta sesión con el usuario system.

```
SQL> show user
```

156) Abrimos una nueva sesión con el usuario sys, desde otra terminal.

```
SQL> exit
```

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

157) Confirmamos esta sesión con el usuario sys.

```
SQL> show user
```

158) Desde el usuario Sys paramos la BD de forma normal.

```
SQL> shutdown
```

GNUNota: Se queda esperando a que se desconecten los usuarios.

159) Desde la sesión del usuario system.

```
SQL> exit
```

GNUNota: Al cerrar la sesión del usuario system que está abierta, termina de cerrar sin problemas.

160) Se puede ver mirando las últimas líneas del archivo alert.

```
[oracle@oraservacl ~]$ locate alert_${ORACLE_SID}.log |xargs tail -f
```

161) Arranca de nuevo la BD y repite la parada, estando conectado algún usuario, de forma que no espere a que se desconecten.

Subimos la BD Datos.

```
SQL> startup
```

Abrimos una nueva sesión con el usuario system, desde otra terminal.

```
[oracle@oraservacl ~]$ sqlplus system/kikla
```

Confirmamos esta sesión con el usuario system.

```
SQL> show user
```

Desde el usuario Sys paramos la BD de forma immediate.

```
SQL> shutdown immediate
```

GNUNota: Esta forma no espera que se desconecten los usuarios.

162) Localizar el fichero init.ora de nuestra BD.

```
[oracle@oraservacl ~]$ locate init.ora
```

163) Vamos a ver el contenido del archivo init.ora creado desde nuestra instalación.

```
[oracle@oraservacl ~]$ locate init.ora. |xargs cat |less
```

GNUNota: Los dígitos después de .ora serán diferentes para usted.

164) Listar el directorio que contiene archivos de inicio.

```
[oracle@oraservacl ~]$ ls -l $ORACLE_HOME/dbs/
```

165) Localizar el fichero spfile de nuestra BD.

```
[oracle@oraservacl ~]$ locate spfile
```

166) Vamos a ver el contenido de nuestro spfile.

```
[oracle@oraservacl ~]$ cat $ORACLE_HOME/dbs/spfile${ORACLE_SID}.ora
```

167) Vamos a iniciar como DBA.

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

168) Vamos a detener la BD.

```
SQL> shutdown immediate
```

169) Vamos a iniciar la BD.

```
SQL> startup
```

170) Vamos a ver los parámetros de los procesos, por defecto son 150.

```
SQL> show parameter processes
```

171) Vamos crear un archivo init\$ORACLE_SID.ora a partir del spfile que tenemos.

```
SQL> CREATE pfile FROM spfile;
```

```
SQL> exit
```

```
[oracle@oraservacl ~]$ ls -ltrh $ORACLE_HOME/dbs/
```

172) Puedes editar el init\$ORACLE_SID.ora con vim o también añadir al final del mismo una línea con el nuevo valor del parámetro.

```
[oracle@oraservacl ~]$ cd $ORACLE_HOME/dbs/
```

```
[oracle@oraservacl dbs]# vim init$ORACLE_SID.ora
# Processes and Sessions
#####
processes=400
```

173) Vamos a iniciar como DBA.

```
[oracle@oraservacl dbs]$ sqlplus sys/kikla as sysdba
```

174) Vamos a detener la BD.

```
SQL> shutdown immediate
```

175) Vamos a iniciar la BD.

```
SQL> startup pfile=init$ORACLE_SID.ora
```

176) Comprobamos el valor actual del parámetro processes.

```
SQL> show parameter processes
```


177) Crear un spfile a partir del init.ora que hay en \$ORACLE_HOME/dbs.

```
SQL> CREATE spfile FROM pfile;
```

178) Vamos a ver el contenido del spfile creado.

```
SQL> !cat $ORACLE_HOME/dbs/spfile$ORACLE_SID.ora
```

179) Vamos a detener la BD.

```
SQL> shutdown immediate
```

180) Vamos a iniciar la BD.

```
SQL> startup
```

181) Comprobamos el limite de procesos alterado.

```
SQL> SELECT name, value Process  
FROM V$parameter  
WHERE name='processes';
```

182) Modificar el parametro del spfile, dinamicamente.

```
SQL> alter system set processes=200 scope=spfile;
```

183) Comprobamos el valor actual del parámetro processes.

```
SQL> show parameter processes
```

184) Comprobamos los cambios.

```
SQL> !cat $ORACLE_HOME/dbs/spfile$ORACLE_SID.ora
```

```
SQL> exit
```

185) Monitoreemos los movimientos de la BD desde el alert.

```
[oracle@oraservacl ~] locate alert_$ORACLE_SID.log |xargs tail -f
```

GNUNota: Recomendamos abrir otra terminal y realizar este ejercicio.

```

Total System Global Area 1603411968 bytes
Fixed Size                2213776 bytes
Variable Size             603981936 bytes
Database Buffers          989855744 bytes
Redo Buffers              7360512 bytes
Database mounted.
SQL> alter database open read only;

Database altered.

SQL> create table system.mitabla (c1 varchar2(2)) tablespace users;
create table system.mitabla (c1 varchar2(2)) tablespace users
*
ERROR at line 1:
ORA-00604: error occurred at recursive SQL level 1
ORA-16000: database open for read-only access

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>

```

en modo
y volver
read-only,
se estado (si
odo Read-0

```

ARCH: Archival disabled due to shutdown: 1089
Shutting down archive processes
Archiving is disabled
Wed Apr 03 15:23:27 2013
ARCH shutting downWed Apr 03 15:23:27 2013
ARCH shutting down
ARCH1: Archival stopped
ARCH3: Archival stopped
Wed Apr 03 15:23:27 2013
ARCH shutting down
ARCH2: Archival stopped
Wed Apr 03 15:23:27 2013
ARCH shutting down
ARCH0: Archival stopped
ARCH: Archival disabled due to shutdown: 1089
Shutting down archive processes
Archiving is disabled
Archive process shutdown avoided: 0 active
Wed Apr 03 15:23:29 2013
Stopping background process VKTM:
Wed Apr 03 15:23:31 2013
Instance shutdown complete

```

186) Hacemos login a la BD como DBA.

```
[oracle@oraservacl dbs]$ sqlplus sys/kikla as sysdba
```

187) Vamos a detener la BD.

```
SQL> shutdown immediate
```

188) Arrancar solo la instancia (NOMOUNT) y consultar algún parámetro. Qué ocurre al acceder a V\$CONTROLFILE y por qué.

```
SQL> startup nomount
```

```
SQL> show parameter processes
```

```
SQL> SELECT count(*)
      FROM v$controlfile;
```

```
SQL> SELECT *
      FROM dba_users;
```

GNUNota: Base de Datos no abierta, solo se permiten consultas en tablas/vistas fijas.

189) Ahora montar la BD y volver a consultar V\$CONTROLFILE. ¿Qué sucede al leer DBA_USERS?.

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

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

```
SQL> SELECT *
      FROM dba_users;
```

```
SQL> SELECT count(*)
      FROM v$datafile;
```

```
SQL> SELECT count(*)  
      FROM v$logfile;
```

```
SQL> shutdown immediate
```

190) Cerrar la BD y abrirla en modo READ ONLY, y crear una tabla.
Activar modo READ WRITE y volver a crear la tabla.

```
SQL> startup mount
```

GNUNota: Para abrir la BD en modo read-only, primero debemos arrancar la BD sin abrirla (MOUNT), si no está ya en ese estado (si está abierta la cerraremos).

191) Para abrir la BD en modo Read-Only.

```
SQL> alter database open read only;
```

192) Creamos una tabla.

```
SQL> CREATE table hr.acl (libro varchar2(20)) tablespace users;
```

193) Paramos la BD Datos.

```
SQL> shutdown immediate
```

194) Subimos la BD Datos.

```
SQL> startup
```

195) Creamos la tabla.

```
SQL> CREATE TABLE hr.acl (libro varchar2(20)) tablespace users;
```

```
SQL> exit
```

GNUNota: Como hemos visto, en modo read-only no podemos crear una tabla pues los ficheros de la BD están en modo de sólo lectura. Si abrimos la BD en modo normal (read-write), no tendremos problemas para crear la tabla.

196) Vamos a consultas algunas vistas estáticas.

```
SQL> SELECT username, ACCOUNT_STATUS, CREATED  
      FROM dba_users;
```

```
SQL> SELECT rpad(username,14), rpad(osuser,10), rpad(machine,15), rpad(terminal,10),  
          rpad(program,27)  
      FROM v$session  
     WHERE username is NOT NULL  
     ORDER BY username;
```

197) Impedir las conexiones de usuarios, de modo que el DBA sí pueda conectarse.

```
SQL> startup restrict
```

GNUNota: Trate de realizar una conexión con un usuario que no sea el Db.

198) Forzar un checkpoint y hacer un insert en la tabla SCOTT.DEPT. Inmediatamente después hacer SHUTDOWN ABORT. Arrancar y comprobar el contenido de la tabla SCOTT.DEPT. Volver a repetir el insert, haciendo commit antes del SHUTDOWN ABORT; y comprueba el contenido de la tabla SCOTT.DEPT. Activar el modo restringido en caliente.

```
SQL> connect / as sysdba
```

```
SQL> alter system checkpoint;
```

```
SQL> insert into scott.dept values (53,'Capacitacion','ACL');
```

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

```
SQL> SHUTDOWN ABORT
```

```
SQL> startup
```

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

Ahora Si :)

```
SQL> alter system checkpoint;
```

```
SQL> insert into scott.dept values (53,'Capacitacion','ACL');
```

```
SQL> commit;
```

```
SQL> SHUTDOWN ABORT
```

```
SQL> startup
```

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

GNUNota: Se trata de comprobar los mecanismos de recuperación de la instancia. Vamos a insertar una fila en SCOTT.DEPT y a simular una caída de la BD con SHUTDOWN ABORT. Comprobaremos que, como no se hizo COMMIT, los datos insertados no están, al arrancar de nuevo la BD. Si repetimos el experimento haciendo COMMIT después del INSERT, comprobaremos que la información no se pierde; ya que al hacer commit estamos forzando el volcado a disco de los buffers redolog, que serán leídos en el siguiente arranque al recuperarse la instancia.

Recuerda que el checkpoint es el evento que fuerza la escritura de los bloques de datos que hay en la caché de datos de la SGA, a disco, por parte del DBWR (previo volcado de los buffer redo log), por parte del LGWR.

199) Conectarse como SCOTT y haz update sobre DEPT (sin hacer commit). Desde otra sesión, cierra la BD con SHUTDOWN TRANSACTIONAL. ¿Qué pasa al hacer commit en la sesión de SCOTT?

```
SQL> connect scott/kikla
```

```
SQL> update dept set dname='Ventas' WHERE deptno=30;
```

Abrimos otra sesión y nos conectamos como SYS.

```
SQL> conn sys/kikla as sysdba
```

```
SQL> shutdown transactional
```

GNUNota: Abrimos una terminal para monitoreo de los movimientos de la BD desde el alert.

```
[oracle@oraservacl ~]$ locate alert_${ORACLE_SID}.log |xargs tail -f
```

Confirmamos la transacción desde la sesión con el usuario scott:)

```
SQL> commit;
```

GNUNota: Una vez que SCOTT ha hecho commit vemos como se cierra la BD.



```
oracle@oraserv5:/opt/app/oracle/diag/rdbms/oraserv5/oraserv5/trace
File Edit View Search Terminal Help

^[
Thu Apr 04 16:32:58 2013
Shutting down instance (transactional)
Stopping background process SMC0
Shutting down instance: further logons disabled
Thu Apr 04 16:32:59 2013
Stopping background process QMNC
Stopping background process CJQ0
Stopping background process MMNL
Stopping background process MMON
Thu Apr 04 16:33:54 2013
All transactions complete. Performing immediate shutdown
```

FICHERO DE CONTROL.

200) Localizar el fichero de control desde el S0.

```
[oracle@oraservacl ~]$ locate .ctl | grep control |xargs ls -ltrh
```

201) Ahora veremos cómo obtener la misma información desde la propia BD, de forma más directa.

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

202) Consultar la información de la BD relativa al contenido del fichero de control. Forzar un checkpoint y volver a consultar. Consultar los registros que contiene el fichero de control.

```
SQL> SELECT CONTROLFILE_TYPE type,
      to_char(CONTROLFILE_CREATED,'dd/mm/yyyy hh24:mi') CREATED,
      CONTROLFILE_SEQUENCE# sequence,CONTROLFILE_CHANGE# change,
      to_char(CONTROLFILE_TIME,'dd/mm/yyyy hh24:mi') time
      FROM v$database;
```

GNUNota: En la vista V\$DATABASE hay una serie de columnas, con el prefijo CONTROLFILE, que nos dan información sobre el fichero de control, por ejemplo la fecha de creación del fichero de control (CONTROLFILE_CREATED) o la última vez que se actualizó la copia del fichero de control (CONTROLFILE_TIME).

203) Si forzamos un checkpoint, se actualiza el fichero de control y podremos ver cómo cambia el valor de las columnas V\$DATABASE.CONTROLFILE_SEQUENCE#. Y V\$DATABASE.CONTROLFILE_TIME

```
SQL> alter system checkpoint;
```

```
SQL> SELECT CONTROLFILE_TYPE type,
      to_char(CONTROLFILE_CREATED,'dd/mm/yyyy hh24:mi') CREATED,
      CONTROLFILE_SEQUENCE# sequence,CONTROLFILE_CHANGE# change,
      to_char(CONTROLFILE_TIME,'dd/mm/yyyy hh24:mi') time
      FROM v$database;
```

204) En la vista V\$CONTROLFILE_RECORD_SECTION se pueden ver las distintas secciones que componen el fichero de control y los registros de los que consta cada una; así como el número de registros de cada tipo usados hasta el momento.

```
SQL> SELECT TYPE,RECORDS_USED,RECORDS_TOTAL
      FROM v$controlfile_record_section;
```

205) Añade una copia al fichero de control de la BD en “/opt/app/controlfiles/”.

Cerramos la BD.

```
SQL> shutdown immediate
```

Salimos de SQL.

```
SQL> exit
```

Creamos el directorio donde copiaremos el nuevo controlfiles.

```
[oracle@oraservacl ~]$ mkdir -p /opt/app/controlfiles/
```

Copiamos el nuevo controlfile en el nuevo directorio.

```
[oracle@oraservacl ~]$ cp -vpi $ORACLE_BASE/oradata/$ORACLE_SID/control01.ctl  
/opt/app/controlfiles/control03.ctl
```

Comprobamos que la copia fue exitosa.

```
[oracle@oraservacl ~]$ ls -l /opt/app/controlfiles/
```

Con un editor abrir el archivo init\$ORACLE_SID y agregar al final de la línea donde están los controlfile la ruta del nuevo controlfile.

```
[oracle@oraservacl ~]$ vim $ORACLE_HOME/dbs/init$ORACLE_SID.ora  
*.control_files='/opt/app/oracle/oradata/oraserv2/control01.ctl',  
'/opt/app/oracle/fast_recovery_area/oraserv2/control02.ctl',  
'/opt/app/controlfiles/control03.ctl'
```

Confirmamos que esta agregada la ruta del nuevo controlfiles.

```
[oracle@oraservacl ~]$ cat $ORACLE_HOME/dbs/init$ORACLE_SID.ora
```

Iniciamos sesión con el usuario sys.

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

Ahora iniciamos la BD con los parámetros de nuestro pfile.

```
SQL> startup pfile=init$ORACLE_SID.ora
```

Confirmamos desde la BD que el nuevo controlfile está agregado.

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

206) Sacar una copia de seguridad del fichero de control.

En primer lugar sacaremos una copia sobre un fichero de traza, que contendrá las sentencias sql necesarias para crear un nuevo fichero de control.

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

```
SQL> exit
```

Vemos al final la sentencia.

```
[oracle@oraservacl ~]$ locate alert_$ORACLE_SID |xargs tail
```

GNUNota: Observar el fichero de traza generado y luego ver su contenido.

Ejecutamos la sentencia para realizar la copia del controlfile.

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

Confirmamos que se creo la copia.

```
SQL> !ls -ltrh /opt/app/controlfiles/
```


TableSpaces y DataFiles.

207) Consultar los tablespaces que componen la BD desde la tabla solo accesible por el DBA.

```
SQL> SELECT tablespace_name, extent_management  
FROM dba_tablespaces;
```

208) Consultar los tablespaces que componen la BD desde las tablas virtuales.

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

209) Consultar los tablespaces que componen la BD desde la tabla user_tablespaces.

```
SQL> SELECT *  
FROM user_tablespaces;
```

210) Especifica la ubicación predeterminada para los archivos de datos administrados por Oracle.

```
SQL> show parameter DB_CREATE_FILE_DEST
```

211) Vemos la descripción de la tabla que contiene información de los Data Files.

```
SQL> desc dba_data_files
```

212) Para ver todos los datafiles de una base de datos.

```
SQL> SELECT file_name DataFiles  
FROM dba_data_files;
```

213) Listar los tablespaces con la ubicación de sus datafile.

```
SQL> SELECT tablespace_name TableSpace, File_name Datafile  
FROM dba_data_files;
```

214) Saber el número máximo de datafile que puede tener su base de datos.

```
SQL> show parameter db_files;
```

215) Para conocer el tamaño ocupado de los DataFiles.

```
SQL> SELECT sum(bytes)/1024/1024 MB  
FROM dba_data_files;
```

216) Ver el espacio total de los TableSpaces.

```
SQL> SELECT tablespace_name TableSpace,  
        ROUND(sum(to_char(bytes)/1024/1024),'9999.99') MB  
        FROM dba_data_files b  
        WHERE tablespace_name NOT LIKE 'TEMP%'  
        GROUP BY b.tablespace_name;
```

217) Listar los tablespace con su tamaño y espacios libres.

```
SQL> SELECT tablespace_name,  
        SUM (mbtotal) / 1024 / 1024 mbtotal,  
        SUM ( mbtotal - mblibre) / 1024 / 1024 mbusado,  
        SUM (mblibre) / 1024 / 1024 mblibre,  
        round(((SUM ( mbtotal - mblibre) / 1024 / 1024) * 100) / (SUM (mbtotal) /  
1024 / 1024), 2) Porc_ocu,  
        round(((SUM (mblibre) / 1024 / 1024) *100) / (SUM (mbtotal) / 1024 /  
1024),2) Porc_libre  
        FROM (SELECT tablespace_name, bytes mbtotal, 0 mblibre  
        FROM dba_data_files  
        UNION ALL  
        SELECT tablespace_name, 0 mbtotal, bytes mblibre  
        FROM dba_free_space)  
        GROUP BY tablespace_name  
        ORDER BY 6;
```

218) Listar los tablespace con los ficheros que forman parte de cada tablespace, con su crecimiento dinámico.

```
SQL> SELECT rpad(a.TABLESPACE_NAME,10) tablespace,rpad(a.FILE_NAME,40) fichero,  
        to_char(a.BYTES/1024/1024,'999.99') MB,  
        to_char(a.increment_by*b.value/1024/1024,'99.99') nextmb,  
        to_char(a.MAXBYTES/1024/1024,'99999.99') maxmb  
        FROM DBA_DATA_FILES a, v$parameter b  
        WHERE b.name='db_block_size';
```

219) Consultar la hora que se realizaron los checkpoint.

```
SQL> SELECT rpad(name,40) name,CHECKPOINT_CHANGE#,  
        to_char(CHECKPOINT_TIME,'dd/mm/yyyy hh24:mi') CHECKPOINT_TIME  
        FROM v$datafile;
```

220) Para ver los “ocupantes” del tablespace SYSAUX, y cuánto ocupa cada uno consultaremos .

```
SQL> SELECT rpad(occupant_name,30), space_usage_kbytes  
        FROM v$sysaux_occupants;
```

221) Vamos a crear un tablespace llamado "acl" en con su datafile en /opt/app/datafiles su tamaño sera de 80MB.

Creamos el directorio donde almacenaremos el DataFile.

```
[oracle@oraservacl ~]$ mkdir -p /opt/app/datafiles
```

Sentencia para crear el TableSpace con su Datafile.

```
SQL> CREATE tablespace acl
      datafile '/opt/app/datafiles/acl01.dbf'
      size 20M;
```

Confirmamos que esta creado el TableSpace.

```
SQL> SELECT name Tablespaces
      FROM v$tablespace;
```

Vemos el tamaño físicamente ocupado por el Datafile.

```
SQL> !du -sch /opt/app/datafiles/acl01.dbf
```

Vemos el tamaño del TableSpace.

```
SQL> SELECT tablespace_name TableSpace,
      ROUND(sum(to_char(bytes)/1024/1024), '9999.99') MB
      FROM dba_data_files b
      WHERE tablespace_name LIKE 'AC%'
      GROUP BY b.tablespace_name;
```

222) Ahora vamos aumentar el tamaño del Datafile acl01.dbf a 80MB.

Alteramos su tamaño físicamente.

```
SQL> ALTER DATABASE DATAFILE '/opt/app/datafiles/acl01.dbf' RESIZE 80M;
```

Vemos el aumento del tamaño físicamente por el Datafile.

```
SQL> !du -sch /opt/app/datafiles/acl01.dbf
```

Vemos el tamaño del TableSpace.

```
SQL> SELECT tablespace_name TableSpace,
      ROUND(sum(to_char(bytes)/1024/1024), '9999.99') MB
      FROM dba_data_files b
      WHERE tablespace_name LIKE 'AC%'
      GROUP BY b.tablespace_name;
```

223) Vamos a agregar un DataFiles llamado fcld01 al TableSpace acl.

Añadimos el Datafile fcld01.

```
SQL> ALTER TABLESPACE acl ADD DATAFILE '/opt/app/datafiles/fcld01.dbf'
      SIZE 50M;
```

Vemos que esta creado físicamente por el Datafile.

```
SQL> !ls -ltrh /opt/app/datafiles/fcld01.dbf
```

Vemos el total de MB que tiene el TableSpace ACL.

```
SQL> SELECT tablespace_name TableSpace,
      ROUND(sum(to_char(bytes)/1024/1024), '9999.99') MB
      FROM dba_data_files b
      WHERE tablespace_name LIKE 'AC%'
      GROUP BY b.tablespace_name;
```

224) Comprobar cuales Datafiles pertenecen al TableSpace ACL.

```
SQL> SELECT tablespace_name TableSpace, File_name Datafile
      FROM dba_data_files
      WHERE tablespace_name='ACL';
```

225) Vamos a modificar el TableSpaces acl01 que sea auto extend y su crecimiento sea cada 5M con un máximo de 100M.

```
SQL> alter database datafile '/opt/app/datafiles/acl01.dbf' autoextend ON next 5M
      maxsize 100M;
```

226) Modificar la disponibilidad de los datafiles.

```
SQL> alter database datafile '/opt/app/datafiles/acl01.dbf' offline;
```

GNUNota: Si está en modo NoArchiveLog no se puede poner offline a menos que sea para eliminarlo, porque una vez puesto offline no se podrá poner online.

227) Para sincronizar un datafile que se quedo offline se hace el recover.

```
SQL> recover datafile '/opt/app/datafiles/acl01.dbf';
```

GNUNota: Como no está sincronizado el datafile con el resto de los datafile da error y pide que se haga recovery.

228) Volvemos a poner el DataFile online.

```
SQL> alter database datafile '/opt/app/datafiles/acl01.dbf' online;
```

229) Renombrar y Reubicar DataFiles.

Paso1-Poner El Tablespace Offline.

```
SQL> alter tablespace acl offline;
```

```
SQL> exit;
```

Paso2-Renombrar o Reubicar El Archivo En El S.O.

```
[oracle@oraservacl ~]$ mv /opt/app/datafiles/fcld01.dbf $ORACLE_BASE/oradata
```

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

Paso3-Renombrar desde la BD la Ruta.

```
SQL> alter tablespace acl rename datafile '/opt/app/datafiles/fcld01.dbf'
to '$ORACLE_BASE/oradata/fcld01.dbf';
```

Paso3-Comprobamos La Ubicación En La Base Datos.

```
SQL> SELECT tablespace_name, FILE_NAME, SUM(BYTES)/1024/1024 Mega
FROM DBA_DATA_FILES WHERE TABLESPACE_NAME='ACL'
GROUP BY tablespace_name, FILE_NAME;
```

Paso4-Poner el tablespace online y hacer un backup de la base de datos.

```
SQL> alter tablespace ACL online;
```

GNUNota: No olvidar hacer una copia de los Control Files.

230) Para Eliminar el DataFile fcld01.

```
SQL> alter tablespace ACL drop datafile '/opt/app/datafiles/fcld01.dbf';
```

231) Crea el tablespace datacl01, con el fichero /opt/app/datafiles/datacl01.dbf, con un tamaño de 1M. Créalo de nuevo, manejado localmente. Crea 3 tablas (acl01, acl02, acl03) de 256K sobre dicho tablespace. Borra la tabla acl02 y crea una tabla acl04 de 384K. ¿Qué ocurre y por qué?.

```
SQL> CREATE TABLESPACE datacl01
datafile '/opt/app/datafiles/datacl01.dbf' size 1M
reuse extent management local uniform size 128K;
```

```
SQL> CREATE table acl01 (fcl VARCHAR2(4000))
tablespace datacl01 storage (initial 256K minextents 1);
```

```
SQL> CREATE table acl02 (fcl VARCHAR2(4000))
tablespace datacl01 storage (initial 256K minextents 1);
```

```
SQL> CREATE table acl03 (fcl VARCHAR2(4000))
tablespace datacl01 storage (initial 256K minextents 1);
```

```
SQL> drop table acl02 purge;
```

GNUNota: A partir de 10g, si RECYCLEBIN está activado, hay que añadir la opción “purge” a la sentencia “drop table” para que la tabla se borre realmente, y no se quede en la “papelera”.

```
SQL> SELECT bytes
      FROM dba_free_space
      WHERE tablespace_name='DATA01';
```

```
SQL> CREATE table acl04 (fcl VARCHAR2(4000))
      tablespace data01 storage (initial 384K minextents 1);
```

```
SQL> SELECT bytes
      FROM dba_free_space
      WHERE tablespace_name='DATA01';
```

GNUNota1: Al tratarse de un tablespace local, con “uniform size” de 128K, habiendo 2 extensiones libres de 128 y 256K, se puede crear una tabla de 384K, pues realmente asigna trozos de 128K, y hay 3, dos de los cuales están juntos.

GNUNota2: Como se observa, siendo el tablespace local, se aprovechan los huecos que se generan con el borrado de tablas. La cuestión clave, en este caso, es encontrar un tamaño adecuado para “uniform size”. Una posibilidad es tener varios tablespaces: pequeño, mediano y grande con uniform size, por ejemplo, de 8K, 256K y 8M:

- DAT08K: para guardar tablas de menos de 1M de tamaño.
- DAT0256K: para tablas de entre 1M hasta 32M.
- DAT08M: para aquellas mayores de 32M.

232) Poner el tablespace acl01 en modo READ-ONLY. Inserta una fila en alguna de sus tablas, ¿qué ocurre?. Borra la tabla acl01, ¿por qué se puede borrar?. Deja el tablespace data01 en modo READ-WRITE. Repite el insert, ahora sobre acl03.

```
SQL> alter tablespace data01 read only;
```

```
SQL> insert into acl01 values ('Somos Mas que GNU');
```

GNUNota: En un tablespace read only no se pueden actualizar datos (insert, update, delete), pero sí se pueden borrar objetos (drop table), ya que el borrado de objetos sólo afecta al DD (y no al tablespace).

```
SQL> DROP TABLE acl01 purge;
```

```
SQL> alter tablespace data01 read write;
```

```
SQL> insert into acl03 values ('Somos Mas que GNU');
```

```
SQL> commit;
```

233) Crea una tabla acl05 de 512K en el tablespace datacl01, ¿qué ocurre y por qué?. Activa el autoextend de su fichero, ajustando next 256K y maxsize 2M. Vuelve a crear la tabla acl05.

```
SQL> SELECT bytes
      FROM dba_free_space
      WHERE tablespace_name='DATACL01';
```

```
SQL> CREATE table acl05 (C1 VARCHAR2(4000))
      tablespace datacl01 storage (initial 512K minextents 1);
```

GNUNota1: Cuando el tablespace se llena no será posible crear más objetos, a no ser que esté activado el 'autoextend', de forma que permita la ampliación del tablespace lo suficiente para albergar el nuevo objeto.

GNUNota2: No se puede crear una tabla de 512K porque solo quedan 256K libres en el tablespace.

```
SQL> alter database datafile '/opt/app/datafiles/datacl01.dbf'
      autoextend on next 256K maxsize 2M;
```

```
SQL> CREATE table acl05 (fcl0 VARCHAR2(4000))
      Tablespace datacl01 storage (initial 512K minextents 1);
```

```
SQL> SELECT bytes
      FROM dba_free_space
      WHERE tablespace_name='DATACL01';
```

```
SQL> SELECT rpad(segment_name,20) segmento, segment_type tipo, bytes
      FROM dba_extents
      WHERE tablespace_name='DATACL01';
```

234) Crea el tablespace indacl01 de 1M
'/opt/app/datafiles/indacl01.dbf'. Moverlo
'/opt/app/datafiles2/indacl01.dbf'.

```
SQL> CREATE tablespace indacl01 datafile '/opt/app/datafiles/indacl01.dbf' size 1M
      reuse extent management local uniform size 64K;
```

```
SQL> alter tablespace indacl01 offline;
```

GNUNota: Para mover o renombrar un fichero de un tablespace, este último debe estar OFFLINE.

```
SQL> !mv /opt/app/datafiles/indacl01.dbf /opt/app/datafiles2/indacl01.dbf
```

```
SQL> alter tablespace indacl01 rename datafile
      '/opt/app/datafiles/indacl01.dbf' to '/opt/app/datafiles2/indacl01.dbf';
```

```
SQL> alter tablespace indacl01 online;
```

```
SQL> SELECT file_name FROM dba_data_files;
```

235) Redimensionar el tablespace datacl01 a 1M ¿qué ocurre y por qué? Redimensionar el tablespace datacl01 a 512K, y añadirle otro fichero de 512K.

```
SQL> alter database datafile '/opt/app/datafiles/datacl01.dbf' resize 1M;
```

GNUNota: Se puede redimensionar un fichero reduciendo su tamaño solo si el espacio recortado está libre. En el caso de datacl01 no es posible reducirlo a 1M porque tiene espacio ocupado en el espacio recortado. Sin embargo, indacl01 sí que puede ser reducido a 512K porque está vacío.

```
SQL> alter database datafile '/opt/app/datafiles2/indacl01.dbf' resize 512K;
```

```
SQL> SELECT rpad(file_name,50),bytes  
FROM dba_data_files  
WHERE tablespace_name='INDACL01';
```

236) Crear un tablespace de “undo” UNDO_acl. Convertirlo en el tablespace de “undo” activo. Crear una tabla acl06 en el nuevo tablespace, ¿qué ocurre? Consultar la vista V\$UNDOSTAT.

```
SQL> CREATE undo tablespace undo_acl  
datafile '/opt/app/datafiles/undo_acl01.dbf' size 1M;
```

```
SQL> SELECT *  
FROM dba_tablespaces  
WHERE tablespace_name='UNDO_ACL';
```

```
SQL> CREATE TABLE acl06 (fclcd VARCHAR2(2)) TABLESPACE UNDO_ACL;
```

GNUNota: Podemos tener varios tablespaces de “undo”, pero solo uno de ellos estará activo en un momento dado (el indicado en el parámetro undo_tablespace). No se pueden crear objetos sobre un tablespace de “undo” (como puede ser una tabla). Al cambiar de tablespace “undo” activo (con undo_tablespace), los segmentos de rollback que contiene el nuevo tablespace pasan a estar “online”, mientras que los del tablespace anterior se ponen offline.

```
SQL> SHOW PARAMETER UNDO_TABLESPACE
```

```
SQL> SELECT SEGMENT_NAME, TABLESPACE_NAME, STATUS  
FROM DBA_ROLLBACK_SEGS;
```

```
SQL> ALTER SYSTEM SET UNDO_TABLESPACE=UNDO_ACL;
```

```
SQL> SELECT SEGMENT_NAME, TABLESPACE_NAME, STATUS  
FROM DBA_ROLLBACK_SEGS;
```



```
SQL> SELECT to_char(BEGIN_TIME,'dd/mm/yyyy hh24:mi') BEGIN_TIME,  
           b.name TABLESPACE, UNDOBLKS, TXNCOUNT  
FROM V$UNDOSTAT a, v$tablespace b  
WHERE A.UNDOTSN=B.TS#  
ORDER BY BEGIN_TIME;
```

GNUNota: En la vista V\$UNDOSTAT se guarda información valiosa cuando se usa “undo automático”, pues podemos ver estadísticas (cada 10 minutos) del tablespace de undo usado (UNDOTSN), bloques de rollback consumidos (UNDOBLKS) y el no total de transacciones de dicho periodo (TXNCOUNT).

237) Crear un tablespace temporal TEMP_acl de 1M. ¿Qué ocurre al ponerlo read-only? Haz que sea el tablespace temporal por defecto de la base de datos ¿que pasa al ponerlo offline? ¿y si lo borramos?

```
SQL> show parameter sort_area_size
```

```
SQL> CREATE temporary tablespace temp_acl tempfile  
'/opt/app/datafiles/temp_acl.dbf' size 1M extent management local uniform size 64K;
```

```
SQL> alter tablespace temp_acl read only;
```

GNUNota: Un tablespace temporal no puede ponerse en modo “read only”, y tampoco se puede poner offline. El tablespace temporal por defecto de la BD, además, no puede ser borrado.

```
SQL> alter tablespace temp_acl offline;
```

```
SQL> alter database default temporary tablespace temp_acl;
```

```
SQL> alter database default temporary tablespace temp;
```

```
SQL> drop tablespace temp_acl;
```

238) Comprobar el tablespace por defecto y tablespace temporal que tiene definido cada usuario que existe en la BD. Modificar el tablespace por defecto de la BD por datacl01 y volver a hacer la comprobación anterior. Renombrar el tablespace datacl01 por datosfcl01 y hacer de nuevo la comprobación.

```
SQL> SELECT USERNAME, rpad(DEFAULT_TABLESPACE,20), rpad(TEMPORARY_TABLESPACE,20)  
FROM DBA_USERS;
```

```
SQL> alter database default tablespace datacl01;
```

```
SQL> SELECT USERNAME, rpad(DEFAULT_TABLESPACE,20), rpad(TEMPORARY_TABLESPACE,20)  
FROM DBA_USERS;
```

GNUNota: Como se observa, para los usuarios OUTLN, SYS, SYSTEM y DBSNMP no cambia el tablespace por defecto.

```
SQL> alter tablespace datacl01 rename to datosfld;
```

```
SQL> SELECT USERNAME, rpad(DEFAULT_TABLESPACE, 20), rpad(TEMPORARY_TABLESPACE, 20)
FROM DBA_USERS;
```

239) Crear los tablespaces tempacl01 Y tempacl02 de 4M cada uno y asignarlos al grupo GTEMP. Asignar al usuario scott el tablespace temporal GTEMP.

```
SQL> CREATE temporary tablespace tempacl01
tempfile '/opt/app/datafiles/tempacl01.dbf' size 4M
extent management local uniform size 64K;
```

```
SQL> CREATE temporary tablespace tempacl02
tempfile '/opt/app/datafiles/tempacl02.dbf' size 4M
extent management local uniform size 64K;
```

```
SQL> ALTER TABLESPACE tempacl01 TABLESPACE GROUP GTEMP;
```

```
SQL> ALTER TABLESPACE tempacl02 TABLESPACE GROUP GTEMP;
```

```
SQL> ALTER USER SCOTT TEMPORARY TABLESPACE GTEMP;
```

```
SQL> SELECT USERNAME, rpad(DEFAULT_TABLESPACE, 20), rpad(TEMPORARY_TABLESPACE, 20)
FROM DBA_USERS WHERE USERNAME='SCOTT';
```

240) También podemos crear un TableSpace que se puede extender el tamaño de su espacio cada vez que se requiera, definiendo el tamaño de crecimiento y el máximo del mismo.

```
SQL> CREATE tablespace DATACL03 datafile '/opt/app/datafiles/datacl03.dbf' size 50M
autoextend ON next 5M maxsize 500M;
```

241) Para eliminar el TableSpace datacl03.

```
SQL> drop tablespace DATACL03;
```

GNUNota: Investigar la opción "INCLUDING CONTENTS".

242) Modificamos la ubicación predeterminada para los DataFiles.

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST = '/opt/app/datafiles';
```

243) Confirmamos nuestra nueva ubicación por defecto para los DataFiles.

```
SQL> show parameter DB_CREATE_FILE_DEST
```

244) Creamos un TableSpace llamado pomavid_data.

```
SQL> CREATE TABLESPACE pomavid_data;
```

245) Verificamos si esta creado el TableSpace.

```
SQL> SELECT *  
      FROM v$tablespace;
```

246) Verificamos si esta creado el Datafile.

```
SQL> !ls /opt/app/datafiles/ORASERV5/datafile/
```

247) Verificamos si esta creado el DataFile y su ubicación.

```
SQL> SELECT tablespace_name TableSpace, File_name Datafile  
      FROM dba_data_files  
      WHERE tablespace_name LIKE 'POMA%';
```

```
SQL> exit;
```

248) Vemos el tamaño del DataFile pomavid_data.

```
[oracle@oraservacl ~]$ find /opt/app/ -iname "*pomavi*" |xargs du -sch
```

FICHEROS REDO LOG.

249) Localizar los ficheros redolog de la BD. ¿Cuántos grupos hay y cuántos miembros tiene cada grupo? ¿están correctamente distribuidos?.

```
[oracle@oraservacl ~]$ find $ORACLE_BASE -iname "*redo*" |grep .log |xargs du -sch
```

```
SQL> SELECT member  
      FROM v$logfile;
```

250) Comprobar el fichero redo log activo. ¿Qué ocurre al forzar un "log switch"? ¿y al forzar un checkpoint?.

```
SQL> SELECT *  
      FROM v$log;
```

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

GNUNota: Al forzar un "log switch", se provoca el cambio de fichero redo log (current)

GNUNota1: El grupo "ACTIVE" es aquel que se acaba de llenar y cuyo checkpoint está pendiente de terminar. Un Checkpoint NO supone un "log switch". Podemos consultar el avance de los checkpoints en V\$DATABASE.CONTROLFILE_SEQUENCE#.

```
SQL> SELECT CONTROLFILE_SEQUENCE#,checkpoint_change#,LOG_MODE  
      FROM v$database;
```

Cambiamos el redolog group activo.

```
SQL> alter system checkpoint;
```

```
SQL> SELECT CONTROLFILE_SEQUENCE#,checkpoint_change#,LOG_MODE  
      FROM v$database;
```

251) Vamos a añadir un miembro mas a cada grupo en otra ubicación.

Vemos los grupos actuales.

```
SQL> SELECT GROUP#, MEMBERS, STATUS  
      FROM v$log;
```

Creamos el directorio donde almacenaremos los nuevos logs.

```
SQL> !mkdir /opt/app/redolog
```

Agregamos los nuevos grupos.

```
SQL> alter database add logfile member  
      '/opt/app/redolog/redo11.log' to group 1,  
      '/opt/app/redolog/redo12.log' to group 2,  
      '/opt/app/redolog/redo13.log' to group 3;
```

Vemos los miembros agregados.

```
SQL> SELECT member  
      FROM v$logfile;
```

Vemos la cantidad por grupo.

```
SQL> SELECT GROUP#,MEMBERS,STATUS  
      FROM v$log;
```

Para ver el tamaño de los redolog.

```
SQL> !find /opt/app -iname "redo*" | grep .log |xargs du -sch
```

252) Añadamos un grupo más (grupo 4), con dos miembros de 4M: /opt/app/redolog/redo04.log y /opt/app/redolog/redo14.log. Añade 2 grupos más (grupo 5 y 6), con las mismas características.

```
SQL> alter database add logfile group 4  
      ('/opt/app/redolog/redo04.log',  
       '/opt/app/redolog/redo14.log') size 4M;
```

GNUNota: Si en 10g u 11g, con Linux, intentamos crear un fichero redolog de menos de 4M, por ejemplo de 2M, nos dará el error: ORA-00336: el tamaño de los bloques 4096 del archivo log es inferior al mínimo de 8192 de bloques esto es porque el tamaño del bloque que Oracle usa para el redolog es de 512bytes (en Linux), y como indica el error, el numero mínimo de bloques que puede tener un fichero redolog en 10g es de 8192 bloques, en este caso de 512bytes, que son 4M.

```
SQL> alter database add logfile group 5  
      ('/opt/app/redolog/redo05.log',  
       '/opt/app/redolog/redo15.log') size 4M;
```

```
SQL> alter database add logfile group 6  
      ('/opt/app/redolog/redo06.log',  
       '/opt/app/redolog/redo16.log') size 4M;
```

```
SQL> SELECT member  
      FROM v$logfile;
```

```
SQL> SELECT GROUP#,MEMBERS,STATUS  
      FROM v$log;
```

GNUNota: Finalmente haremos los “log switch” que hagan falta para que se usen los nuevos redolog que hemos creado.

```
SQL> alter system switch logfile;
```

253) Elimina los miembros del grupo 1, de uno en uno. ¿Qué ocurre al eliminar el último?. Borrar los grupos 1, 2 y 3. ¡¡¡ Ojo con borrar el redo log activo !!!

```
SQL> alter database drop logfile member  
      '$ORACLE_BASE/oradata/$ORACLE_SID/redo01.log';
```

```
ORA-00362: miembro se requiere para formar un archivo de registro válido en el grupo 1
```

```
SQL> alter database drop logfile member '/opt/app/redolog/redo11.log';
```

GNUNota: Vamos a ver cómo borrar miembros de redo, y también grupos redolog. No podemos borrar el redo log activo (current). Si intentamos borrar el último miembro que quede en un grupo, no podremos hacerlo. Los borrados son siempre a nivel de BD, no se borran los ficheros a nivel del S.O habría que borrarlos una vez que hayamos hecho el borrado a nivel de BD ¡¡¡ ojo con esta operación, y borremos un redolog de la BD.

254) Vamos a borrar los 3 grupos que tienen miembros de 10M; para quedarnos sólo con los nuevos de 4M.

```
SQL> alter database drop logfile group 1;
```

```
SQL> alter database drop logfile group 2;
```

```
SQL> alter database drop logfile group 3;
```

GNUNota: Recuerda que no se puede borrar el fichero de redo "current", sobre el que está escribiendo el LGWR. Así que forzaremos un "log switch".

```
SQL> SELECT GROUP#,MEMBERS,STATUS  
      FROM v$log;
```

```
SQL> alter system switch logfile;
```

```
SQL> alter database drop logfile group 4;
```

GNUNota: Tampoco podemos borrar un fichero de redo si está "active", y por tanto pendiente de que finalice el checkpoint correspondiente. Podemos hacer otro "log switch" para acelerar el proceso, y si sigue "ACTIVE", forzar el checkpoint con "alter system checkpoint;"

```
SQL> SELECT GROUP#,MEMBERS,STATUS  
      FROM v$log;
```

```
SQL> alter system switch logfile;
```

```
SQL> SELECT GROUP#,MEMBERS,STATUS  
      FROM v$log;
```

```
SQL> alter database drop logfile group 4;
```

```
SQL> SELECT member
```

```
FROM v$logfile;
```

255) Cambiar el nombre de los miembros de redo de los grupos 4 a acfredo4.log, acfredo14.log.

```
SQL> shutdown immediate
```

GNUNota: La BD debe estar solo montada !!! por tanto hay que cerrar la BD y después solo montarla. A continuación se cambia el nombre a los ficheros (desde el S.O.); y entonces modificamos el nombre a los redolog (desde la BD). Si intentamos cambiar el nombre en la BD, antes de hacerlo desde el S.O, cascará. Finalmente abrimos la BD y veremos los nuevos nombres en V\$LOGFILE.

```
SQL> startup mount
```

```
SQL> SELECT member  
FROM v$logfile;
```

```
SQL> alter database rename file '/opt/app/redolog/redo04.log' to  
'/opt/app/redolog/acfredo04.log';
```

GNUNota: No podemos hacer el “rename” en el fichero de control si antes no lo hemos hecho a nivel del S.O.

```
SQL> !mv /opt/app/redolog/redo04.log /opt/app/redolog/acfredo4.log
```

```
SQL> alter database rename file '/opt/app/redolog/redo04.log' to  
'/opt/app/redolog/acfredo4.log';
```

```
SQL> !mv /opt/app/redolog/redo14.log /opt/app/redolog/acfredo14.log
```

```
SQL> alter database rename file '/opt/app/redolog/redo14.log' to  
'/opt/app/redolog/acfredo14.log';
```

```
SQL> SELECT group#, rpad(member,40) member  
FROM v$logfile;
```

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

256) Habilitar el modo archivelog en Oracle 11g.

Cómo saber en qué modo está una base de datos.

```
SQL> SELECT name, log_mode  
FROM v$database;
```

Nos aseguramos que la DB no está en modo archivelog.

```
SQL> ARCHIVE LOG LIST
```

Creamos el directorio destino de los ficheros de archivado.

```
SQL> !mkdir /opt/app/archivelog_acl
```

Ahora configuramos el destino de los ficheros de archivado.

```
SQL> alter system set log_archive_dest_1='location=/opt/app/archivelog_acl'  
scope=spfile;
```

Paramos la instancia y la iniciamos montada.

```
SQL> shutdown immediate
```

```
SQL> startup mount;
```

Arrancamos el proceso de archivado.

```
SQL> alter database archivelog;
```

abrimos la base de datos.

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

Por último comprobamos que el sistema de archivado está funcionando.

```
SQL> archive log list
```

```
SQL> alter system switch logfile;
```

```
SQL> !ls -lrt /opt/app/archivelog_acl
```


257) Desactivar el modo ARCHIVELOG en Oracle 11g.

Paramos la instancia.

```
SQL> shutdown immediate
```

La arrancamos montandola.

```
SQL> startup mount;
```

Arrancamos el proceso de archivado.

```
SQL> alter database noarchivelog;
```

Abrimos la base de datos.

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

Por último comprobamos que el sistema de archivado está funcionando.

```
SQL> archive log list
```

Desactivamos el archivado automático.

```
SQL> alter system archive log stop
```

Fuente:

<http://www.codigolibre.org>
<http://acl.edu.do>
<http://dbagroup.cl>
<http://myorastuff.blogspot.com>
<http://idevelopment.info>
<http://www.oracle-base.com>
<http://itdavid.blogspot.com>
<http://alexechavarriaperez.blogspot.com>
<http://oracledbacr.blogspot.com>
<http://setijoagus.wordpress.com>
<http://cursos.atica.um.es>
<http://oracle.com>
<http://psoug.org/>