**Universidad Nacional de Costa Rica**

**Administración de Bases de Datos**

Creación de una base de datos Oracle

Para obtener más información, visite  [ww w.slinfo .una.ac.cr](http://www.slinfo.una.ac.cr/)

Autor: Steven R. Brenes Chavarria

Email:  [sb renes @una.cr](mailto:sbrenes@una.cr)  Fecha: 1 de Agosto del 2013

**INTRODUCCIÓN**

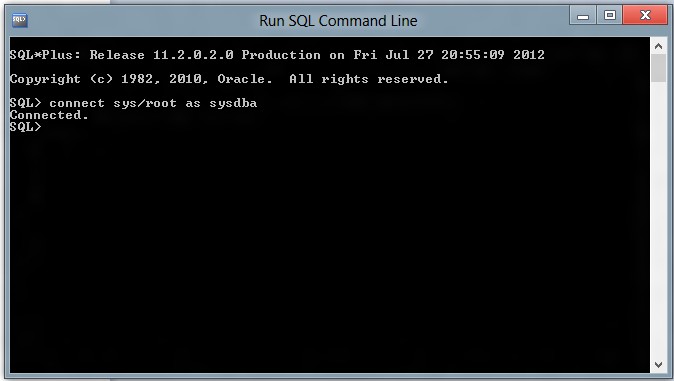
Las bases de datos realizan una serie de procesos para garantizar la integridad de los datos y la consistencia de los archivos del sistema operativo. Es por ello que la simple acción de arrancar una base de datos se convierta fácilmente en N pasos intermedios de los cuales debemos conocer a profundidad para detectar errores cuando se nos presenten.

**ARRANQUE DE UNA BASE DE DATOS**

Antes de iniciar una base de datos, debemos estar conectados al SGBD mediante un usuario con rol de super administrador. Solamente este usuario es capaz de arrancar y apagar la base de datos. El nombre de usuario super administrador en Oracle se denomina **sys.**

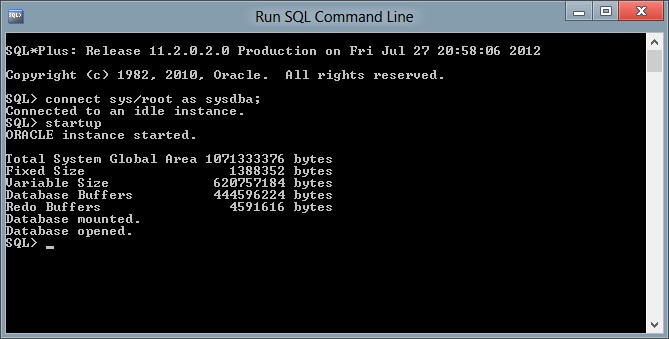
El primer paso para iniciar una base de datos, es autentificarnos como super administrador. Para ello ejecutamos la siguiente instrucción en la consola de Oracle:

SQL> CONNECT sys/root as SYSDBA



Una vez nos salga el mensaje de Conectado exitosamente, debemos escribir la siguiente instrucción:

SQL> STARTUP OPEN;



Oracle primeramente lee el archivo init.ora, localiza los ficheros de control, crea e inicia al SGA y arranca todos los procesos. En este punto la base de datos esta arrancada.

Estos datos significan que el SGA fue instanciado correctamente en el servidor. Sin embargo la base de datos ha pasado por tres estados previos antes de estar lista para recibir consultas. La instrucción STARTUP OPEN realmente realiza tres instrucciones (nomount, mount y open) antes de estar disponible para el DBA. Desde luego el DBA también puede instanciar una base de datos en un estado intermedio con las siguientes instrucciones:

**NOMOUNT**

La instrucción de nomount se utiliza cuando se crea una base de datos o cuando, debido a una falla física se ha pedido un archivo de control. Para iniciar una base de datos en modo NOMOUNT se debe ejecutar la siguiente instrucción:

SQL> STARTUP NOMOUNT;

**MOUNT**

El estado de MOUNT abre los ficheros de control para localizar los ficheros de datos y los redo log, pero no se realiza ninguna comprobación de ellos en este momento. Luego bloquea la instancia para evitar intervenciones de otras instancias.

Existen muchas razones por las que los DBA necesitan montar la base en modo mount, entre las principales están las modificaciones a la base de datos como:

1. Efectuar recuperaciones

2. Poner Offline/online un tablespace

3. Recolocar los ficheros de datos y redo log.

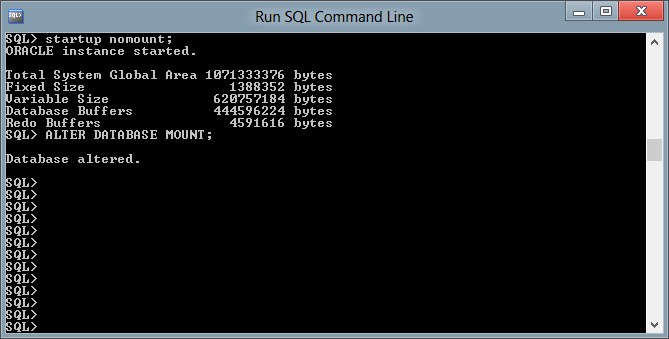
4. Crear/borras/modificar un nuevo grupo o miembro de redo log.

Se utiliza la siguiente instrucción para montar la base en modo Mount:

SQL> STARTUP MOUNT;

Para pasar de una base de datos en estado NOMOUNT a un estado MOUNT basta ejecutar la siguiente instrucción:

SQL> ALTER DATABASE MOUNT;



**OPEN**

El estado de OPEN, bloquea los ficheros de datos y abre todos los ficheros redo log. Si la instancia abre una Base de datos que se cerró de forma anormal, entonces el proceso OPEN recupera de los redo log el estado de consistencia mas reciente y pone en línea la base de datos.

Para iniciar una base de datos en modo OPEN se debe ejecutar:

SQL> ALTER DATABASE OPEN; SQL> STARTUP OPEN;

**DETENER DE UNA BASE DE DATOS**

Detener una base de datos, va depender el estado en la que se encuentre. Existen tres formas diferentes de apagar una base de datos. La sintaxis es la siguiente:

SQL> SHUTDOWN [normal | immediate | abort];

**Shutdown Normal**

La operación de pagado normal, realiza las siguientes acciones en orden: se impide el acceso a la base de datos (se bloquea), espera a que todos los usuarios completen todas sus peticiones y se desconecten del servidor. Purga todos los buffers de datos y cachés de redo log, actualizando los ficheros de datos y de redo log, se eliminan los bloqueos de ficheros, se completan las transacciones en marcha, se actualizan las cabeceras de ficheros, elimina los threads, libera los bloqueos de la BD por parte de la instancia, y sincroniza los ficheros de control y de datos.

**Shutdown Immediate**

El proceso de detención de la base de datos mediante el immediate, termina todos los procesos de la base de datos en curso, para ello realiza un rollback a todas las transacciones no confirmadas. Luego la BD es detenida. Una desventaja de este mecanismo de apagado es que los usuarios son expulsados, la ventaja es que la BD queda en un estado consistente y si errores.

**Shutdown abort**

Es el último recurso a utilizar, debe ser usado solamente cuando los demás mecanismos fallas. Por ejemplo ante la caída de un proceso que impida el apago normal. Este modo de apagado tiene la característica que aborta todas las transacciones incluyendo las que se están realizando en un momento dado. **No se realizan rollback** a las transacciones abortadas, es por ello que la BD queda en un estado de inconsistencia, que debe ser reparado con los redo log la próxima vez que la BD inicie.

**CREACIÓN DE UNA BASE DE DATOS**

Para crear una base de datos Oracle se debe tener la siguiente información:

1. Ubicación del archivo INIT.ORA

2. Nombre y tamaños de los tablespaces a crear

3. Estructura del servidor creada (discos duros, conexiones FTP, etc)

Primero que todos, debemos modificar el siguiente archivo llamado **init.ora** por la configuración que necesitemos.

# $Header: rdbms/admin/init.ora /main/23 2009/05/15 13:35:38 ysarig Exp $

#

# Copyright (c) 1991, 1997, 1998 by Oracle Corporation

# NAME

# init.ora

# FUNCTION

# NOTES

# MODIFIED

# ysarig 05/14/09 - Updating compatible to 11.2

# ysarig 08/13/07 - Fixing the sample for 11g

# atsukerm 08/06/98 - fix for 8.1.

# hpiao 06/05/97 - fix for 803

# glavash 05/12/97 - add oracle\_trace\_enable comment

# hpiao 04/22/97 - remove ifile=, events=, etc.

# alingelb 09/19/94 - remove vms-specific stuff

# dpawson 07/07/93 - add more comments regarded archive start

# maporter 10/29/92 - Add vms\_sga\_use\_gblpagfile=TRUE

# jloaiza 03/07/92 - change ALPHA to BETA

# danderso 02/26/92 - change db\_block\_cache\_protect to

\_db\_block\_cache\_p

|  |  |  |  |
| --- | --- | --- | --- |
| # | ghallmar | 02/03/92 | - db\_directory -> db\_domain |
| # | maporter | 01/12/92 | - merge changes from branch 1.8.308.1 |
| # | maporter | 12/21/91 | - bug 76493: Add control\_files parameter |
| # | wbridge | 12/03/91 | - use of %c in archive format is discouraged |
| # | ghallmar | 12/02/91 | - add global\_names=true, db\_directory=us.acme.com |
| # | thayes | 11/27/91 | - Change default for cache\_clone |
| # | jloaiza | 08/13/91 | - merge changes from branch 1.7.100.1 |
| # | jloaiza | 07/31/91 | - add debug stuff |

|  |  |  |  |
| --- | --- | --- | --- |
| # | rlim | 04/29/91 | - removal of char\_is\_varchar2 |
| # | Bridge | 03/12/91 - | log\_allocation no longer exists |
| # | Wijaya | 02/05/91 - | remove obsolete parameters |
| # |  |  |  |

###########################################################################

###

# Example INIT.ORA file

#

# This file is provided by Oracle Corporation to help you start by providing

# a starting point to customize your RDBMS installation for your site.

#

# NOTE: The values that are used in this file are only intended to be used

# as a starting point. You may want to adjust/tune those values to your

# specific hardware and needs. You may also consider using Database

# Configuration Assistant tool (DBCA) to create INIT file and to size your

# initial set of tablespaces based on the user input.

###########################################################################

####

# Change '<ORACLE\_BASE>' to point to the oracle base (the one you specify at install time)

db\_name='XE' memory\_target=1G processes = 150

audit\_file\_dest='<ORACLE\_BASE>/admin/orcl/adump'

audit\_trail ='db' db\_block\_size=8192 db\_domain=''

db\_recovery\_file\_dest='<ORACLE\_BASE>/flash\_recovery\_area' db\_recovery\_file\_dest\_size=2G diagnostic\_dest='<ORACLE\_BASE>' dispatchers='(PROTOCOL=TCP) (SERVICE=ORCLXDB)' open\_cursors=300

remote\_login\_passwordfile='EXCLUSIVE' undo\_tablespace='UNDOTBS1'

# You may want to ensure that control files are created on separate physical devices

control\_files = (ora\_control1, ora\_control2)

compatible ='11.2.0'

Posteriormente creamos la estructura d archivos que necesitamos. Recuerde que los tablespaces deben quedar separados físicamente incluyendo los ControlFiles para garantizar mayor seguridad en la integridad.

Después de crear la estructura física del servidor, debemos invocar la siguiente instrucción que va crear la base de datos propiamente:

create database "<nombre\_bd>"

logfile group 1 ('<nombre\_bd>/redo1.log') size 100M, group 2 ('<nombre\_bd>/redo2.log') size 100M, group 3 ('<nombre\_bd>/redo3.log') size 100M

character set WE8ISO8859P1

national character set utf8

datafile '<nombre\_bd>/system.dbf' size 500M autoextend on next 10M maxsize unlimited extent management local

sysaux datafile '<nombre\_bd>/sysaux.dbf' size 100M autoextend on next 10M

maxsize unlimited

undo tablespace undotbs1 datafile '<nombre\_bd>/undotbs1.dbf' size 100M

default temporary tablespace temp tempfile '<nombre\_bd>/temp01.dbf' size

100M;

**CREATE TABLESPACE <nombre> DATAFILE *<nombredatafile.dbf>*SIZE 10M AUTOEXTEND ON NEXT 10M MAXSIZE 100M ;**

Finalmente ejecutamos las siguientes instrucciones para cargar el catalogo del diccionario de datos y el catálogo de procedimientos:

[@C:\oraclexe\app\oracle\product\11.2.0\server\rdbms\admin\catalog.sql](mailto:@C:)

[@C:\oraclexe\app\oracle\product\11.2.0\server\rdbms\admin\catproc.sql](mailto:@C:)

**Iniciar una base de datos creada manualmente**

Si queremos iniciar una base de datos creada por nosotros, es necesario contar con la ubicación del

**INIT.ORA**, una vez tengamos la ruta ejecutamos la siguiente instrucción:

SQL> STARTUP [PFILE=@ruta](mailto:@ruta)