

## TEMA 2

### Instalación y configuración de una Base de Datos.

#### 2.1. CONSIDERACIONES ANTES DE INSTALAR.

Para el caso de Oracle, una base de datos puede crearse al mismo tiempo en el que se instala el software, o puede crearse posteriormente. Existen 2 formas para crear una base de datos:

- A través de la herramienta `dbca` (Oracle Database Configuration Assistant). Esta opción se revisó en el ejercicio práctico 3.
- A través de línea de comandos con la instrucción `create database`.








#### 2.1.1. Planeación previa a la creación de la Base de Datos

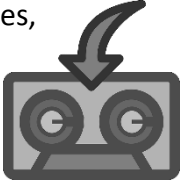
La siguiente lista muestra las acciones recomendadas que deben realizarse antes de comenzar con la creación de una nueva base de datos:

| Num. | Actividad de planeación   |
|------|---|
| 1.   | Con base al modelo relacional aprobado y validado, realizar una estimación de espacio en disco que será necesario reservar para cada una de las tablas e índices. <ul style="list-style-type: none"><li>• Esta estimación se debe proyectar a lo largo de varios años.</li></ul>  |
| 2.   | Diseñar la estructura de directorios y ubicaciones de todos los archivos que integran a la base de datos. <ul style="list-style-type: none"><li>• Una correcta distribución de archivos en distintos directorios y en discos diferentes permitirá un rendimiento mucho mejor para realizar operaciones I/O Ejemplos:<ul style="list-style-type: none"><li>○ Almacenar los archivos REDO Log en discos separados.</li><li>○ Almacenar los data files en discos diferentes para evitar contención.</li><li>○ Controlar la densidad de datos: número de registros a almacenar por cada bloque de datos.</li><li>○ Ubicar al Fast Recovery Area en un dispositivo de almacenamiento diferente al de los data files.</li></ul></li><li>• En el caso de Oracle, existen 2 principales herramientas y conceptos que permiten simplificar estas tareas:<ul style="list-style-type: none"><li>○ Oracle Managed files.</li><li>○ Automatic Storage Management (ASM)</li></ul></li></ul> |
| 3.   | Diseñar y determinar el nombre global de la base de datos. Esta tarea se debe realizar con base a la estructura de red, de tal forma que su nombre global sea único. <ul style="list-style-type: none"><li>• El nombre global se forma por 2 elementos representados por los parámetros:<ul style="list-style-type: none"><li>○ <code>DB_NAME</code></li><li>○ <code>DB_DOMAIN</code></li></ul></li></ul> Ejemplo:<br><code>cursobd.fi.unam</code> , <code>cursobd</code> representa el nombre y <code>fi.unam</code> al dominio.   |
| 4.   | Conocer y familiarizarse con un archivo muy importante llamado <b>Server parameter file</b> . <ul style="list-style-type: none"><li>• Como su nombre lo indica, contiene la definición de los parámetros de la base de datos.</li></ul>   |
| 5.   | Seleccionar el juego de caracteres para ser almacenados en la BD. <ul style="list-style-type: none"><li>• Muy importante seleccionar el valor correcto durante la creación de la BD.</li></ul>  |

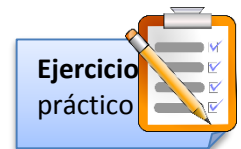


| Num. | Actividad de planeación   |
|------|---|
|      | <ul style="list-style-type: none"> <li>• Modificar su valor posterior a su creación no es una tarea factible y puede llevar a pérdida o inconsistencia en los datos.</li> <li>• El valor a seleccionar depende de requerimientos como:             <ul style="list-style-type: none"> <li>○ Soporte para almacenar datos en múltiples lenguajes</li> <li>○ Restricciones por parte de aplicaciones o compatibilidad.</li> </ul> </li> <li>• La recomendación para Oracle es el uso de un juego de caracteres es AL32UTF8 y representa el valor por default.</li> </ul> <p>Responder las siguientes preguntas como parte del <u>ejercicio práctico 01</u>.</p> <p>A. ¿Qué significa AL32UTF8? Ojo: No es equivalente a UTF-8.</p> <p>B. ¿Cuál es la longitud máxima que puede tener un carácter con esta configuración?</p> <p>C. ¿En qué casos un carácter requeriría la longitud máxima para poder almacenarse?</p> <p>D. ¿Por qué Oracle recomienda este juego de caracteres? Listar beneficios.</p> <p>E. ¿Qué desventajas y en qué situaciones no se recomendaría este valor?</p>    |
| 6.   | <p>Configurar la zona horaria de la Base de Datos.</p> <ul style="list-style-type: none"> <li>• Oracle emplea por default el siguiente archivo para validar zonas horarias:<br/> <code>\$ORACLE_HOME/oracore/zoneinfo/timezlg_11.dat</code><br/> <code>\$ORACLE_HOME/oracore/zoneinfo/little/timezlg_11.dat</code><br/> <code>\$ORACLE_HOME/oracore/zoneinfo/big/timezlg_11.dat</code> </li> </ul>   |
| 7.   | <p>Definir el tamaño máximo de un bloque de datos.</p> <ul style="list-style-type: none"> <li>• Se especifica con el parámetro <code>db_block_size</code>.</li> <li>• Valores típicos son 4K y 8K.</li> <li>• En caso de no especificarse se emplea el valor del sistema operativo, el cual suele ser adecuado.</li> <li>• En caso de ser diferente al del sistema operativo, este debe ser un múltiplo. Por ejemplo, si en el s.o. es 2k (2048), en la BD podría ser <code>db_block_size=4096</code></li> <li>• Este valor no puede modificarse después de la creación de la BD.</li> <li>• Un valor mayor de bloque de datos ofrece mayor eficiencia en disco y operaciones I/O en memoria. Se recomienda aumentar su valor con respecto al s.o. en las siguientes condiciones:             <ul style="list-style-type: none"> <li>○ La Base de datos está instalada en un servidor con grandes capacidades de hardware, en especial, memoria y discos duros muy rápidos.</li> <li>○ El tamaño de bloque del s.o. es muy pequeño, por ejemplo, de 1K. Si la BD hereda este valor, se tendrá una excesiva actividad I/O reduciendo desempeño.</li> </ul> </li> </ul> <p><u>Ejemplos:</u></p> <ul style="list-style-type: none"> <li>• Comando en s.o. para mostrar el tamaño del bloque:<br/> <code>sudo blockdev --getbsz /dev/sda5</code> </li> <li>• Mostrar el tamaño del bloque en BD:<br/> <code>SYS-SQL&gt; show parameter db_block_size</code> </li> </ul>  |
| 8.   | <p>Definir el tamaño del bloque de datos para almacenar Redo Log Files</p> <ul style="list-style-type: none"> <li>• A diferencia del tamaño de bloque de los data files que pueden tener tamaños entre 2K y 32K para el caso de los archivos Redo Log, el tamaño por default coincide con el tamaño físico del sector del disco. Típicamente 512 Bytes. Son tamaños menores a los de los archivos de datos.</li> <li>• Algunos discos con capacidades mayores de almacenamiento ofrecen tamaños de sectores mayores, p.e. 4K. Esta capacidad ofrece beneficios, pero para la base de datos, un tamaño mayor para bloques Redo Log implica una condición llamada "Redo wastage"             <ul style="list-style-type: none"> <li>○ Redo wastage representa el número de bytes de un bloque que fueron escritos a disco antes de que dicho bloque se llenara. P.e. Suponer que el bloque es de 4K. En la operación normal de la Base de Datos, se requiere escribir constantemente el contenido de bloques</li> </ul> </li> </ul>   |

| Num. | Actividad de planeación   |
|------|---|
|      | <p>Redo Log a disco que frecuentemente no se alcanzan a llenar. Estos espacios vacíos representarían un “desperdicio” ya que se escriben a disco bloques incompletos. Una similitud sería: Un vuelo de avión deja más ganancias cuando se llena al 100% que realizarlo a un porcentaje menor, por ejemplo al 60%.</p> <ul style="list-style-type: none"> <li>○ Cuando el disco ofrece un sector mayor, al crear una BD el tamaño se ajusta automáticamente. En estos casos se debe verificar si es adecuado dejar este tamaño o realizar algún ajuste.</li> </ul> <p>La siguiente instrucción muestra las estadísticas de la cantidad de bytes “desperdiciados” que se obtienen al realizar las escrituras a disco a través del proceso de background LGWR.</p> <pre>select name, value from v\$sysstat where name = 'REDO WASTAGE';</pre> <p>La siguiente instrucción muestra el tamaño de bloque asignado para archivos Redo Log.</p> <pre>select blocksize from v\$log;</pre>       |
| 9.   | <p>Definir el tamaño inicial del tablespace <code>sysaux</code></p> <ul style="list-style-type: none"> <li>• Este tablespace es creado al crear la BD.</li> <li>• Actúa como tablespace auxiliar al tablespace SYS. Empleado por herramientas y funcionalidades adicionales de la BD.</li> <li>• Su existencia permite la distribución de la carga de trabajo entre estos 2 tablespaces: <code>system</code> y <code>sysaux</code>.</li> <li>• El tamaño del tablespace <code>sysaux</code> dependerá de las funcionalidades y componentes instalados.</li> <li>• Su tamaño debe ser de al menos 400 MB.</li> <li>• Es importante establecer parámetros y configuraciones iniciales como <code>permanent</code>, <code>read write</code>, <code>extend management local</code>, <code>segment space management auto</code>. Posterior a la creación de la BD ya no es posible modificar.</li> <li>• La siguiente sentencia muestra información de los componentes instalados en una BD.</li> </ul> <pre>select * from v\$sysaux occupants</pre>                           |
| 10.  | <p>Creación de un tablespace por default.</p> <ul style="list-style-type: none"> <li>• Es altamente recomendable realizar la creación de un tablespace por default, el cual será asignado a los usuarios que no tienen roles de administración.</li> <li>• Herramientas como <code>dbca</code>, crean el tablespace <code>users</code>.</li> <li>• El tablespace por default se crea al incluir la cláusula <code>default tablespace</code> dentro de la instrucción <code>create database</code>. En caso de no especificarse, se hace uso del tablespace <code>system</code>.</li> </ul>  |
| 11.  | <p>Creación de un tablespace para administrar los datos tipo <i>undo</i>.</p> <ul style="list-style-type: none"> <li>• La BD necesita realizar la administración de datos empleados para realizar operaciones de rollback, y en general datos requeridos para “deshacer (undo)” operaciones realizadas anteriormente.</li> <li>• Estos datos se guardan en la BD en forma de registros y son conocidos como <i>datos undo</i>.</li> <li>• La base de datos puede realizar la administración automática de los <i>datos undo</i>.</li> <li>• Sin embargo, en otras situaciones, se requiere configuración y almacenamiento adicional.</li> <li>• Los datos <i>undo</i> pueden ser almacenados en un tablespace especial destinado solo para realizar este tipo de operaciones.</li> </ul> <p>Por su importancia, se explica esta actividad en un capítulo separado. Al momento de la creación de la BD es posible especificar la forma en la que se realizará su administración.</p>  |
| 12.  | <p>Configurar el directorio <code>ORACLE_HOME</code> en modo “solo lectura”.</p> <ul style="list-style-type: none"> <li>• A partir de Oracle 18c es posible configurar el directorio en modo “solo lectura”.</li> </ul>   |

| Num. | Actividad de planeación  |
|------|--|
|      | <ul style="list-style-type: none"> <li>• Esto evita la modificación de cualquier archivo.</li> <li>• Útil para la aplicación de parches y para crear “Imágenes de Bases de datos” cuyas copias pueden ser distribuidas a diversos servidores y facilitar su creación sin repetir configuraciones.</li> </ul>   |
| 13.  | <p>Desarrollar una estrategia para respaldos y recuperación de tal forma que la base de datos quede protegida ante un evento de falla.</p> <ul style="list-style-type: none"> <li>• Archivos importantes deben ser “multiplexados” como son: control files, redo log, archived redo log, etc.</li> <li>• Los conceptos asociados con esta estrategia se revisan en capítulos posteriores.</li> </ul>  |
| 14.  | Familiarizarse con operaciones comunes como iniciar y detener una instancia, modificar disponibilidad, suspender una BD, activar e inactivar una BD, fases de inicio y término, etc.   |

Revisar el documento correspondiente al ejercicio práctico 01 para poner en práctica los conceptos vistos hasta el momento.



## 2.2. INSTALACIÓN DE LA BASE DE DATOS EMPLEANDO HERRAMIENTAS GRÁFICAS.

- El proceso para crear una base de datos Oracle se explica en los ejercicios prácticos anteriores.
- En estas prácticas se mencionan los requerimientos mínimos de instalación basadas en “*Oracle Database Installation Guide*”.
- Se emplea la herramienta “Oracle Universal Installer”. Permite guiar paso a paso el proceso de instalación del software de Oracle.
- Se hace uso de dbca, herramienta gráfica que realiza la creación de una base de datos de forma gráfica.
- DBCA permite la creación de una base de datos en modo no interactivo/silencioso a través del uso de línea de comandos

### Ejemplo:

```
dbca -silent -createDatabase -templateName General_Purpose.dbc \
  -gdbname oradb.example.com \
  -sid oradb \
  -characterSet AL32UTF8 \
  -memoryPercentage 30 \
  -emConfiguration DBEXPRESS
```

```
Enter SYSTEM user password:
password
Enter SYS user password:
password
Copying database files
1% complete
3% complete
...
```

- Para mayores detalles, revisar la documentación oficial.

## 2.3. CREACIÓN DE LA BASE DE DATOS EMPLEANDO LA INSTRUCCIÓN CREATE DATABASE

La creación de una base de datos a partir de la instrucción `create database` ofrece diversas ventajas:

- Posibilidad de crear scripts que realicen la creación
- Mejora el entendimiento y comprensión de todos los conceptos involucrados durante la planeación, y configuración de la nueva base de datos.
- Permite conocer y entender a mayor profundidad las actividades y configuraciones que deben realizarse posterior a su creación para poder contar con una base de datos completamente funcional.

Esta sección está organizada en una serie de pasos que permitirán realizar la creación de una nueva base de datos empleando la instrucción `create database`.

### 2.3.1. Paso 1: Configurar variables de entorno.

#### 2.3.1.1. ORACLE\_SID

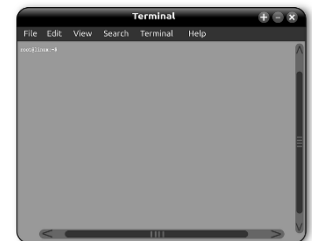
En prácticas anteriores se configuró esta variable. Para ilustrar esta sección se creará una nueva base de datos empleando la siguiente convención para nombrado: `<iniciales>bda<N>` Donde:

- `<iniciales>`: Iniciales del alumno iniciando con el nombre
- `<N>`: Número de base de datos. A lo largo del curso podrían crearse varias bases de datos para propósitos académicos. En este caso `N = 2` ya que se trata de la segunda base de datos que se creará.
- Todos los caracteres son en minúsculas. Ejemplo: `jrcbda2`

#### 2.3.1.2. Otras variables.

Asegurarse que las siguientes variables estén configuradas correctamente:

- `ORACLE_HOME`
- `PATH` (debe contener el directorio `bin` donde encuentra el ejecutable `sqlplus`), típicamente `$ORACLE_HOME/bin`



### 2.3.2. Paso 2: Seleccionar el método de autenticación.

Para poder realizar la creación de una base de datos, es necesario estar autenticado con privilegios de administración. Se pueden seleccionar alguno de los siguientes métodos:

- Autenticación por sistema operativo. En este caso, el usuario que va a realizar la creación de la BD debe pertenecer a alguno de los grupos del s.o. mencionados anteriormente.
- Autenticación a través de archivo de passwords. El archivo debe crearse antes de iniciar con la creación de la base de datos.

### 2.3.3. Paso 3: Crear el archivo de parámetros: PFILE

Al momento de iniciar una instancia Oracle, se lee un archivo de parámetros. El archivo puede tener 2 formatos:

- Archivo de texto que puede ser editado con cualquier editor llamado *Parameter File* (PFILE)

- Archivo binario (formato preferido) Puede ser creado a partir del PFILE y modificado dinámicamente por la base de datos. Al archivo en formato binario se le conoce como *server parameter file (SPFILE)*. Estos conceptos se revisarán más adelante.

### 2.3.3.1. Creación de un archivo de parámetros PFILE

- La forma más sencilla es hacer uso de la plantilla que se encuentra en el directorio `$ORACLE_HOME/dbs` El nombre de la plantilla es `init.ora`

#### Ejemplo:

Abrir el archivo `init.ora` y revisar su contenido

- Una vez creado, es posible iniciar una instancia. Posterior a ello, es posible crear un *server parameter file*.
- Por default, el archivo de parámetros se ubica en `$ORACLE_HOME/dbs`
- El nombre del archivo de parámetros debe tener la siguiente nomenclatura: `init<ORACLE_SID>.ora` Por ejemplo: `initjrcbda1.ora`
- En caso de no cumplir con las convenciones anteriores, se deberá especificar manualmente el nombre y ubicación del archivo al momento de levantar la instancia.

#### Ejemplo:

```
startup pfile="/myfiles/myparamfile.ora"
```

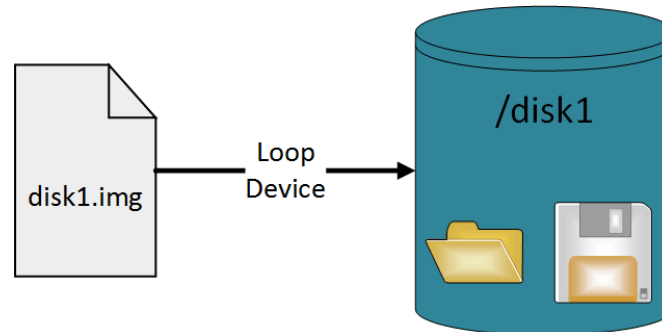
- El archivo debe contener al menos los siguientes parámetros. Todos los demás tomarán sus valores por default.

| Nombre del parámetro   | Descripción  |
|--|--|
| db_name<br>En realidad este es el único parámetro requerido. | Nombre de la base de datos. Debe corresponder con el valor especificado en la instrucción <code>create database</code> así como con el valor de la variable <code>ORACLE_SID</code> .<br><u>Ejemplo:</u><br><code>db_name=jrcbda2</code>   |
| control_files  | No es obligatorio pero muy recomendado. En caso de no especificarse, los archivos se crean en el mismo directorio donde se encuentra el archivo de parámetros. Si se usa, permite la posibilidad de multiplexar archivos (crear y mantener copias para ser empleadas en caso de falla o daño irreparable de este archivo).<br><u>Ejemplo:</u><br><code>control_files=(/u01/app/oracle/oradata/JRCBDA2/control01.ctl,<br/>                  /u02/app/oracle/oradata/JRCBDA2/control02.ctl)</code> <ul style="list-style-type: none"> <li>• Para tener los beneficios del multiplexado, cada archivo debe ubicarse en dispositivos de almacenamiento diferentes.</li> <li>• En este ejemplo se hace uso de la ruta que emplea <code>dbca</code> para crear los archivos de control.</li> </ul> |
| memory_target  | No obligatorio, pero conveniente. Especifica la cantidad máxima de memoria que será asignada a la instancia, habilita la administración automática.  |

### 2.3.3.2. Simulación de discos adicionales.

Para efectos del curso bastará con realizar una simulación de dispositivos de almacenamiento adicionales empleando **loop devices** que permitan realizar el multiplexado de los principales archivos de la base de datos. Posteriormente se hará uso de técnicas más adecuadas: ASM (Automatic Storage Management).

Un loop device es una especie de “pseudo-dispositivo” que permite a un archivo simple ser visto como un dispositivo de bloques (raw device), es decir, puede ser visto como un dispositivo de almacenamiento que puede contener un conjunto de archivos.



- Para poder hacer uso de un loop device, este debe asociarse con un archivo físico del sistema operativo. Por ejemplo: `disk1.img`.
- Si el archivo original tiene su propio sistema de archivos este puede ser montado como si fuera un disco o una partición. Ejemplo de esta técnica son los Cds y los Floppy disks.
- En Linux este tipo de dispositivos están representados por los nodos `/dev/loop0`, `/dev/loop1`, etc.
- Por default se crea cierto número de loop devices al iniciar el sistema, y se pueden crear adicionales con el comando `mknod`.
- Para efectos de este tema, se crearán 2 dispositivos de este tipo `disk2.img` y `disk3.img` que será montados en `/u02` y `/u03` respectivamente. Posteriormente serán creados otros más.
- Este ejercicio permitirá contar con 3 estructuras de directorios para distribuir los archivos de la base de datos: `/u01` (ya existente), `/u02` y `/u03`.

Para implementar esta funcionalidad, crear un script llamado `s-01-crea-loop-devices-root.sh` que realice las siguientes acciones. El script deberá ser ejecutado con el usuario `root`.

*Nota:* Los scripts que se describen a partir de este punto y hasta el final del documento serán ejecutados como parte de los ejercicios prácticos 2,3 y 4.

1. Crear una carpeta `/unam-bda`
2. Cambiarse a dicha carpeta y ejecutar el siguiente comando para crear un archivo que representará a cada uno de los 2 loop devices. Notar que el comando crea un archivo binario de 1GB.

```
dd if=/dev/zero of=disk2.img bs=100M count=10
```

Realizar lo mismo para `disk3.img`

3. Comprobar la creación de los archivos con el siguiente comando:

```
du -sh disk*.img
```

4. Crear cada loop device asociándolo a su archivo creado en el punto anterior haciendo uso del comando `losetup`

```
losetup -fP disk2.img
```

Realizar lo mismo para `disk3.img`

- La opción `-f` ubica o localiza al primer loop device disponible. Generalmente, en Linux se crean 8 loop devices al iniciar el sistema. Esto se puede comprobar ejecutando `ls /dev/loop*`
- La opción `-P` obliga al Kernel a volver a leer la tabla de particiones para incluir a los nuevos loop devices creados.

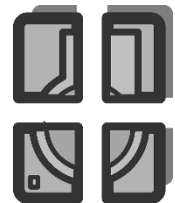
5. Para confirmar la creación del loop device ejecutar :

```
losetup -a
```

6. El siguiente punto es dar formato a los archivos creados para que puedan ser montados. Se emplea el mismo formato que el sistema de archivos Linux: `ext4`.

```
mkfs.ext4 disk2.img
```

Realizar lo mismo para `disk3.img`



7. Crear el directorio `u02` y `u03` donde los dispositivos serán montados.

```
mkdir /u02
```

```
mkdir /u03
```

8. El siguiente comando se emplea para montar el loop device y poderlo emplear como si se tratara de un disco o partición nueva (esta instrucción no se debe incluir en el script).

```
mount -o loop /dev/loop0 /u02
```

- Asumiendo que el loop device creado fue `loop0`, el comando anterior permite montar el archivo en el directorio `/u02` el cual tendrá una capacidad de 1Gb.
  - El comando `mount` funciona, pero tiene un inconveniente: Al reiniciar o apagar el equipo, los loop devices serán desmontados y eliminados.
  - Para resolver este problema se puede hacer uso del archivo `/etc/fstab` en lugar de la ejecución del comando `mount`
  - Durante el inicio del equipo, este archivo es leído. El archivo contiene una lista de todos los dispositivos que deben ser montados durante el arranque. Por lo tanto, se pueden agregar instrucciones para montar los loop devices creados en los pasos anteriores.
9. Antes de editar el archivo `/etc/fstab`, guardar los cambios del script creado y ejecutarlo como usuario `root`. Posteriormente, ejecutar el comando `mount` del paso anterior en una terminal. Si el comando no marca error alguno, editar el archivo `/etc/fstab` y agregar las siguientes líneas al final:

```
#loop devices agregados para la materia BDA
#Asegurarse que los archivos img existan.
/unam-bda/disk2.img      /u02      auto      loop      0      0
```



Realizar lo mismo para `disk3.img`

- La primera columna indica la ruta absoluta donde se encuentra el archivo `img` con formato `ext4`.
- La segunda columna indica la ruta donde será montado.
- Las columnas restantes indican los valores por default para montar el dispositivo.

10. Para probar que todo ha salido correctamente, ejecutar con el usuario `root`: `mount -a`. Este comando lee el contenido del archivo de `/etc/fstab` y realiza el montado de todos los dispositivos listados en el archivo.

Para comprobar que el script funcionó, reiniciar el equipo. Posteriormente ejecutar:

`df -h | grep u0` para confirmar que los dispositivos fueron montados correctamente. El comando deberá mostrar una salida similar a la siguiente:

```
/dev/loop0      969M  2.5M  900M   1% /u02
/dev/loop2      969M  2.5M  900M   1% /u03
```

En este ejemplo, se ha montado el loop device `/dev/loop0` en `/u02`. Se tiene ocupado el 1% de un total de 969 MB.

**Nota** importante acerca de la variable `ORACLE_SID`.

- Hasta este punto `ORACLE_SID` apunta a la base de datos `<nombre>bda1` creada en ejercicios prácticos anteriores. Su valor fue configurado de forma global.
- Durante el curso serán creadas varias bases de datos. En este capítulo se creará una nueva base de datos `<nombre>bda2`.
- Lo anterior implica que la variable `ORACLE_SID` deberá ajustarse a este nuevo valor.
- Típicamente en un ambiente en el que existen varias bases de datos instaladas en el mismo servidor esta variable suele eliminarse de la configuración global y cada vez que se desee acceder a una determinada instancia, se deberá especificar su valor con el siguiente comando:

```
export ORACLE_SID=<nombre>bda<n>
```

Notar que esta configuración pierde efecto al abrir una nueva terminal, o al cambiarse de sesión de usuario en el s.o. Para efectos del curso, se aplicará la configuración sugerida:

- Abrir el script ubicado en `/etc/profile.d` en el que se realizó la definición de las variables de entorno, comentar las siguientes líneas:

```
# Variables de entorno para Oracle.
export TMP=/tmp
export TMPDIR=/tmp
export ORACLE_HOSTNAME=pc-jrc-ora.fi.unam
#export ORACLE_UNQNAME=jrcbda1
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/18.0.0/dbhome_1
export ORA_INVENTORY=/u01/app/oraInventory
#export ORACLE_SID=jrcbda1
```

Una vez que se han creado los puntos de montaje que simulan dispositivos de almacenamiento adicionales, el proceso continúa con la creación del archivo de passwords y parámetros.

Crear un script `s-02-crea-pwd-param-oracle.sh` que realice las siguientes acciones:

1. Establecer temporalmente la variable `ORACLE_SID` con el valor `<iniciales>bda2`
2. Crear un archivo de passwords únicamente con el usuario `SYS` empleando como password `hola1234*`
3. Crear un archivo de parámetros en la ruta `$ORACLE_HOME/dbs` empleando las convenciones de nombrado.
  - Deberá contener la definición de los 3 parámetros de la tabla anterior.
  - El nombre de la BD debe coincidir con el valor de `$ORACLE_SID`. Especificar los siguientes archivos de control:

```
/u01/app/oracle/oradata/<ORACLE_SID>/control01.ctl  
/u02/app/oracle/oradata/<ORACLE_SID>/control02.ctl  
/u03/app/oracle/oradata/<ORACLE_SID>/control03.ctl
```

- Sustituir `<ORACLE_SID>` por el valor real en mayúsculas. Por ejemplo:  
`/u01/app/oracle/oradata/JRCBDA2/control01.ctl`

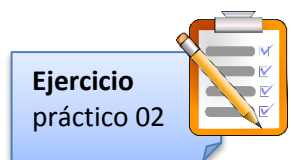


- Asignar 768 MB de memoria.
4. Ejecutar el script empleando el usuario `oracle`.
  5. Ejecutar el siguiente comando para validar que tanto el archivo de parámetros como el archivo de passwords fueron creados.

```
ls -l $ORACLE_HOME/dbs/*<oracle_sid>*
```

- Sustituir `<oracle_sid>` con el valor de la variable en minúsculas.

Revisar el documento correspondiente al ejercicio práctico 02 para realizar las actividades mencionadas hasta este punto.



#### **2.3.4. Paso 4: Crear el archivo de parámetros: SPFILE**

- En la sección anterior se creó el archivo de parámetros PFILE. Como se mencionó anteriormente, para crear una BD se recomienda hacer uso de un SPFILE en lugar de un PFILE.
- Esta actividad permite crear un archivo de parámetros binario SPFILE a partir del archivo PFILE.
- La principal ventaja de su existencia, es la posibilidad de modificar o actualizar los parámetros de la BD empleando la sentencia `alter system`. Los cambios pueden ser actualizados en este archivo y volverse permanentes (estos conceptos se revisan más adelante).

Para crear el archivo SPFILE crear un script llamado `s-01-crea-spfile-ordinario.sql`. El script deberá realizar las siguientes acciones.

1. Autenticarse como usuario `sys` empleando el password indicado en el archivo de passwords creado anteriormente.

## 2. Iniciar la instancia en modo mount.

Hasta este punto es posible iniciar la instancia (áreas de memoria y procesos de background) sin aun existir una base de datos. A este proceso se le conoce como iniciar una instancia en modo nomount.

```
startup nomount
```

## 3. Agregar la siguiente instrucción para crear el archivo binario.

```
create spfile from pfile;
```

La instrucción toma como entrada el archivo de parámetros de texto (PFILE) y genera un archivo binario *Server Parameter File* (SPFILE) en la ubicación por default. En caso que alguno o ambos archivos no se encuentren en las ubicaciones por default, se especifican las rutas absolutas:

```
create spfile='/u01/oracle/dbs/test_spfile.ora'
from pfile='/u01/oracle/dbs/test_init.ora';
```

## 4. Agregar la siguiente instrucción para verificar la creación del *Server parameter file* (SPFILE):

```
!ls $ORACLE_HOME/dbs/spfile<nombre>bda2.ora
```

- El script deberá ser invocado por el usuario ordinario del sistema operativo, no olvidar inicializar ORACLE\_SID con el valor <nombre>bda2.

### Ejemplo:

```
export ORACLE_SID=jrcbda2
sqlplus /nolog
start s-01-crea-spfile-ordinario.sql
```

### 2.3.5. Paso 5: Ejecutar la instrucción create database

La siguiente actividad es ejecutar la instrucción `create database`. Revisar cuidadosamente el siguiente código.

### Ejemplo:

```
create database jrcbda2
user sys identified by system2
user system identified by system2
logfile group 1 (
  '/u01/app/oracle/oradata/JRCBDA2/redo01a.log',
  '/u02/app/oracle/oradata/JRCBDA2/redo01b.log',
  '/u03/app/oracle/oradata/JRCBDA2/redo01c.log') size 50m blocksize 512,
group 2 (
  '/u01/app/oracle/oradata/JRCBDA2/redo02a.log',
  '/u02/app/oracle/oradata/JRCBDA2/redo02b.log',
  '/u03/app/oracle/oradata/JRCBDA2/redo02c.log') size 50m blocksize 512,
group 3 (
  '/u01/app/oracle/oradata/JRCBDA2/redo03a.log',
  '/u02/app/oracle/oradata/JRCBDA2/redo03b.log',
  '/u03/app/oracle/oradata/JRCBDA2/redo03c.log') size 50m blocksize 512
```



```

maxloghistory 1
maxlogfiles 16
maxlogmembers 3
maxdatafiles 1024
character set AL32UTF8
national character set AL16UTF16
extent management local
datafile '/u01/app/oracle/oradata/JRCBDA2/system01.dbf'
    size 700m reuse autoextend on next 10240k maxsize unlimited
sysaux datafile '/u01/app/oracle/oradata/JRCBDA2/sysaux01.dbf'
    size 550m reuse autoextend on next 10240k maxsize unlimited
default tablespace users
    datafile '/u01/app/oracle/oradata/JRCBDA2/users01.dbf'
    size 500m reuse autoextend on maxsize unlimited
default temporary tablespace tempts1
    tempfile '/u01/app/oracle/oradata/JRCBDA2/temp01.dbf'
    size 20m reuse autoextend on next 640k maxsize unlimited
undo tablespace undotbs1
    datafile '/u01/app/oracle/oradata/JRCBDA2/undotbs01.dbf'
    size 200m reuse autoextend on next 5120k maxsize unlimited;

```

. Antes de su ejecución es necesario crear los siguientes directorios:

- Directorio(s) donde se guardarán los *data files*
- Directorio(s) donde se guardarán los archivos *redo logs*.
- Directorio(s) donde se crearán los archivos *control files*.

Los principales parámetros del comando `create database` se describen a continuación.

```
create database <database_name>
```

- En esta instrucción se especifica en nombre de la base de datos.
- Su valor corresponde con el valor de la variable `ORACLE_SID` y del valor configurado en el archivo de parámetros `db name`

```
user sys identified by <sys_password>
```

```
user system identified by <system password>
```

- Especifica los valores para el password de los usuarios `sys` y `system`.
- Si se omiten se emplean los valores por default: `change_on_install` and `manager`.
- Se recomienda asignar sus valores.

```
logfile group <n>
```

- Define los N grupos de archivos Redo Log. En cada grupo se pueden definir N miembros.
- Los Redo Log Files son muy importantes ya que en ellos se guardan todos los movimientos que se aplican a los datos. Para garantizar consistencia y evitar pérdidas de datos, es posible crear N grupos. En cada grupo se tienen N archivos redundantes, los cuales deben ubicarse en discos diferentes. Si un archivo se daña, la BD se puede recuperar tomando los datos de los otros archivos del grupo.
  - Estos conceptos se revisarán más adelante.

¿Cuántos miembros y grupos se deben crear?

- La respuesta básicamente se obtiene haciendo pruebas de tal forma que los grupos deben ser los mínimos posibles para que el proceso de background `LGWR` trabaje de forma óptima.
  - En algunos casos la instancia puede requerir únicamente 2.
  - En otras situaciones se pueden requerir 3 para garantizar que siempre existe un grupo disponible para escribir datos en sus archivos.

- La forma más sencilla de determinar los valores óptimos es analizar los *trace files* del LGWR y el llamado Alert Log de la BD (más adelante se explica la forma de acceder).
- Si los logs indican que LGWR tiene que esperar frecuentemente para que un grupo sea liberado (ya sea porque se está ejecutando un *checkpoint* o porque el grupo aún no ha sido *archivado*), se deben agregar más grupos.
- EL parámetro MAXLOGFILES indica el número máximo de grupos que pueden existir.
- El parámetro MAXLOGMEMBERS indica el número máximo de miembros en cada grupo.
- Este parámetro también especifica el tamaño del bloque para los archivos Redo Log, el cual es de 512 bytes (coincide con el tamaño del sector del disco, revisado en el tema anterior).

#### maxloghistory

Útil únicamente en sistemas con Oracle RAC en modo archivado.

#### character set

Parámetro empleado para indicar el juego de caracteres de la base de datos. Como se mencionó en secciones anteriores, el valor recomendado es AL32UTF8.

#### national character set

Su valor se emplea para ser utilizado en columnas de tablas que especifiquen los siguientes tipos de datos: `nchar`, `nclob` y `nvarchar2`. A nivel general, esta configuración permite que ciertas columnas de tablas específicas guarden datos representadas por juegos de caracteres diferentes al juego de caracteres global configurado a través del parámetro anterior `character set`.

#### extent management local

En esta sección se especifica la creación de tablespaces que requiere una base de datos para operar, así como la definición de data files y sus ubicaciones.

- Un tablespace administrado de forma local hace uso de una especie de matriz de bits almacenado en cada data file empleada para administrar las extensiones y bloques de datos (saber que bloques están disponibles para ser sobrescritos, etc.).
- La primera instrucción `datafile` que aparece permite la creación del tablespace `system` cuyos datos serán almacenados en el data file `system0.dbf`.
- La instrucción `sysaux datafile` se encarga de crear el tablespace `sysaux` empleando el archivo `sysaux01.dbf`.
- La instrucción `default tablespace` permite crear el tablespace por default que se les asigna a todos los usuarios. Sus datos se almacenan en `users01.dbf`.
- La instrucción `default temporary tablespace` crea el tablespace temporal de la base de datos.
- La instrucción `undo tablespace` se emplea para crear el tablespace donde se guardan los datos tipo *undo*. El manejo de datos Undo se revisará más adelante. Se emplea el archivo `undotbs.dbf`.
- La instrucción `user_data tablespace` se encarga de crear un tablespace para guardar datos de usuarios así como datos de algunas funcionalidades que ofrece la BD. Por ejemplo, datos del soporte XML que ofrece Oracle.

Posterior a la ejecución del comando, el status de la instancia se actualiza a OPEN. Esto significa que los usuarios administradores ya pueden crear sesiones y realizar tareas administrativas. Sin embargo, antes de comenzar a utilizar la BD de forma normal se requieren pasos adicionales que se describen a continuación.

Realizar las siguientes acciones para ejecutar la instrucción anterior.

- Crear un Shell script `s-02-crea-directorios-root.sh` que realice las siguientes acciones:
  1. Crear la estructura de directorios necesarios para ejecutar la instrucción `create database` descritos anteriormente.
  2. Cambiar el dueño de todos los directorios creados al usuario `oracle`. Los directorios `/u02` y `/u03` le deben pertenecer al usuario `root`, y los subdirectorios le pertenecerán al usuario `oracle`.
  3. Cambiar el grupo de todos los directorios creados al grupo `oinstall`.
  4. Cambiar los permisos de los directorios creados en el punto anterior a `754 (rwxr-xr--)`. Ejecutar el script empleando al usuario `root`.
- Crear un script SQL `s-04-crea-bd-ordinario.sql` que realice las siguientes acciones:
  1. Iniciar en sesión con el usuario `sys` haciendo uso del archivo de passwords.
  2. En caso de no tener la instancia iniciada en modo `mount`, iniciarla.
  3. Justo antes de la instrucción `create database`, agregar `whenever sqlerror exit rollback` para detener la ejecución del script en caso de error al intentar crear la BD.
  4. Incluir la instrucción `create database` tomando como base el ejemplo anterior:
- Sustituir el valor de `JRCBDA2` con el valor correspondiente para `ORACLE_SID`.
- Notar que la base de datos a crear contará con 3 grupos de Redo Logs, cada uno a su vez con 3 miembros que representan copias multiplexadas para minimizar pérdidas. Verificar las rutas empleadas: `/u01`, `/u02`, `/u03`.
- Asignar un máximo de 16 grupos y un máximo de 3 miembros por grupo.
- Asignar un máximo de 1024 data files
- El juego de caracteres será `AL32UTF8` y el juego de caracteres nacional será `AL16UTF16`
- Agregar la siguiente configuración de tablespaces empleando `extent management local` y el directorio creado anteriormente.
  - Crear los tablespaces `system`, `sysaux`, `users`, `temptst1`, `undotbs1` con los siguientes nombres respectivamente: `system01.dbf`, `sysaux01.dbf`, `users01.dbf`, `temp01.dbf` y `undotbs01.dbf`. Emplear los mismos parámetros mostrados en el ejemplo de creación de una BD.
- 5. Notar que el password asignado para el usuario `sys` es `system2`. Este valor es diferente al password configurado durante la creación del archivo de passwords. Para homologar ambos valores, posterior a la ejecución de la instrucción `create database`, incluir las siguientes instrucciones para homologar el password tanto para el usuario `sys` como para el usuario `system`.

```
alter user sys identified by system2;  
alter user system identified by system2;
```

El script deberá ser invocado con el usuario ordinario del Sistema operativo.

**Ejemplo:**

```
export ORACLE_SID=jrcbda2  
sqlplus /nolog  
start s-04-crea-bd-ordinario.sql
```

### 2.3.5.1. Algunas consideraciones en caso de tener errores al crear la BD

- Asegurarse que los directorios donde se guardarán los archivos Redo Log y los data files existan.
- Asegurarse que el SPFILE contenga la definición de los archivos de control y que dichos archivos existan.
- Asegurarse que la instancia haya sido levantada en modo nomount.
- Si se modifica el SPFILE asegurarse de reiniciar la instancia entre cada intento.
- Revisar el Alert Log.

Para revisar el contenido de este log, realizar las siguientes acciones:

- Ejecutar la siguiente sentencia el sqlplus

```
SQL> col name format a25
SQL> col value format a70
SQL> select name,value from v$diag_info;
```

| NAME                  | VALUE  |
|-----------------------|--|
| Diag Enabled          | TRUE   |
| ADR Base              | /u01/app/oracle  |
| ADR Home              | /u01/app/oracle/diag/rdbms/jrcbda2/jrcbda2                             |
| <b>Diag Trace</b>     | <b>/u01/app/oracle/diag/rdbms/jrcbda2/jrcbda2/trace</b>                |
| Diag Alert            | /u01/app/oracle/diag/rdbms/jrcbda2/jrcbda2/alert                       |
| Diag Incident         | /u01/app/oracle/diag/rdbms/jrcbda2/jrcbda2/incident                    |
| Diag Cdump            | /u01/app/oracle/diag/rdbms/jrcbda2/jrcbda2/cdump                       |
| Health Monitor        | /u01/app/oracle/diag/rdbms/jrcbda2/jrcbda2/hm                          |
| Default Trace File    | /u01/app/oracle/diag/rdbms/jrcbda2/jrcbda2/trace/jrcbda2_ora_25679.trc |
| Active Problem Count  | 0  |
| Active Incident Count | 0  |

- Abrir una terminal y cambiarse a la ruta marcada en negritas.
- Revisar el archivo alert\_<ORACLE\_SID>.log
- Analizar los archivos trace. Dichos archivos están asociados a un número de proceso. Si el error obtenido muestra un número de proceso, se puede consultar el archivo con el número correspondiente.

Para mostrar la ubicación de estos archivos, realizar las siguientes consultas:

- Muestra la ruta del archivo trace de la sesión actual:

```
select value from v$diag_info where name = 'Default Trace File';
```

- Muestra todos los archivos trace de la instancia:

```
select value from v$diag_info where name = 'Diag Trace';
```

- Muestra un trace file asociado a un número de proceso:

```
select pid, program, tracefile from v$process;
```

- Para reintentar la creación de una BD se deberán borrar todos los archivos creados por intentos anteriores.

### 2.3.6. Paso 6: Ejecución de scripts SQL posterior a la creación de la BD.

La última actividad a realizar para contar con una BD totalmente funcional es la ejecución de un conjunto de scripts que en general se encargan de crear todas las tablas y vistas que integran al diccionario de datos.

#### 2.3.6.1. Ejecución de scripts como usuario sys.

```
@?/rdbms/admin/catalog.sql
@?/rdbms/admin/catproc.sql
@?/rdbms/admin/utlrlp.sql
```



#### 2.3.6.2. Ejecución de script como usuario system

```
@?/sqlplus/admin/pupbld.sql
```

- El símbolo “@” se emplea para indicarle a sqlplus que ejecute un archivo con instrucciones.
- El símbolo “?” se emplea para hacer referencia al valor de la variable ORACLE\_HOME. De lo anterior, la ruta de los script sería equivalente a escribir \$ORACLE\_HOME/rdbms/admin/utlrlp.sql

En la siguiente tabla se muestra el significado de cada uno:

| Nombre del archivo | Descripción  |
|--------------------|--|
| catalog.sql        | Se encarga de crear las vistas del diccionario de datos, las vistas dinámicas que muestran datos relacionados con el desempeño de la BD, sinónimos públicos y otros objetos. |
| catproc.sql        | Ejecuta y configura todos los objetos necesarios para poder hacer uso de la programación PL/SQL  |
| utlrlp.sql         | Recompila todos los módulos PL/SQL que se encuentran en estado inválido debido a la reciente creación de la BD.  |
| pupbld.sql         | Configura SQL *Plus para que sea posible la habilitación y des habilitación de comandos SQL para usuarios.   |

Crear un script llamado s-05-crea-diccionario-datos-ordinario.sql que realice las siguientes actividades:

1. Entrar a sesión como usuario sys
  2. Invocar la ejecución de los 3 scripts mostrados anteriormente
  3. Entrar a sesión como usuario system
  4. Invocar la ejecución de los scripts correspondientes.
  5. Ejecutar el script. Ojo: El tiempo de ejecución de estos archivos es alto. Alrededor de 30 min. Es normal observar algunos errores, en especial errores de objetos inexistentes que se obtienen debido a que estos scripts intentan eliminar primero al objeto para asegurar su correcta creación o sobrescritura.
- El script debe ser invocado con el usuario ordinario del s.o.

#### Ejemplo:

```
export ORACLE_SID=jrcbda2
sqlplus /nolog
start s-05-crea-diccionario-datos-ordinario.sql
```



Revisar el documento correspondiente al ejercicio práctico 03 para realizar las actividades estudiadas hasta el momento.

**Ejercicio  
práctico 03**

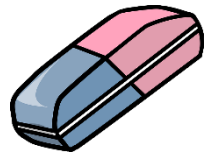


## 2.4. ELIMINACIÓN DE BASES DE DATOS.

Para realizar la eliminación de una base de datos ejecutar la siguiente instrucción como usuario sys en sqlplus:

```
drop database;
```

- Ojo: Esta instrucción elimina TODOS los datos que pudiera tener.
- La instrucción realiza las siguientes acciones:
  - Primero elimina los archivos de control.
  - Elimina todos los archivos que se especifican en los archivos de control.
  - Posteriormente a ello, detiene la instancia.
  - No tiene efecto sobre respaldos o sobre archivos Redo Log archivados.



En la práctica, se recomienda realizar las siguientes actividades para realizar una correcta eliminación de una base de datos.

### 2.4.1. Paso 1. Conexión a sqlplus.

Conectarse a sqlplus como usuario sys

```
$ export ORACLE_SID=mydb  
$ sqlplus / as sysdba
```

### 2.4.2. Paso 2: Ejecución de scripts para consulta.

```
SQL> select name from v$datafile;  
SQL> select name from v$controlfile;  
SQL> select member from v$logfile;
```

La ejecución de estos scripts permitirá identificar los archivos que deben ser eliminados por la instrucción `drop database`. Posterior a la eliminación es recomendable verificar que efectivamente han sido eliminados.

- Consulta 1:

```
sys@jrcbda2> select name from v$datafile;
```

NAME

```
-----  
/u01/app/oracle/oradata/JRCBDA2/system01.dbf  
/u01/app/oracle/oradata/JRCBDA2/sysaux01.dbf  
/u01/app/oracle/oradata/JRCBDA2/undotbs01.dbf  
/u01/app/oracle/oradata/JRCBDA2/users01.dbf  
/u01/app/oracle/oradata/JRCBDA2/usertbs01.dbf  
/u01/app/oracle/oradata/JRCBDA2/apps01.dbf  
/u01/app/oracle/oradata/JRCBDA2/indx01.dbf
```

- Consulta 2:

```
sys@jrcbda2> select name from v$controlfile;
```

NAME

```
-----  
/u01/app/oracle/oradata/JRCBDA2/control01.ctl  
/u02/app/oracle/oradata/JRCBDA2/control02.ctl  
/u03/app/oracle/oradata/JRCBDA2/control03.ctl
```

- Consulta 3:

```
sys@jrcbda2> select member from v$logfile;
```

MEMBER

```
-----  
/u01/app/oracle/oradata/JRCBDA2/redo01a.log  
/u02/app/oracle/oradata/JRCBDA2/redo01b.log  
/u03/app/oracle/oradata/JRCBDA2/redo01c.log  
/u01/app/oracle/oradata/JRCBDA2/redo02a.log  
/u02/app/oracle/oradata/JRCBDA2/redo02b.log  
/u03/app/oracle/oradata/JRCBDA2/redo02c.log  
/u01/app/oracle/oradata/JRCBDA2/redo03a.log  
/u02/app/oracle/oradata/JRCBDA2/redo03b.log  
/u03/app/oracle/oradata/JRCBDA2/redo03c.log
```

#### **2.4.3. Paso 3: Detener la instancia.**

```
shutdown immediate;
```

#### **2.4.4. Paso 4: Iniciar la base de datos en modo exclusivo.**

```
startup mount exclusive restrict;
```

#### **2.4.5. Paso 5: Ejecutar drop database.**

```
drop database;
```

#### **2.4.6. Paso 6: Validaciones posteriores a la eliminación**

- Verificar que los archivos listados anteriormente fueron eliminados de forma correcta.
- Eliminar los directorios en caso de no requerirse. Para los archivos del ejemplo, se puede eliminar el directorio JRCBDA2.
- Cambiarse al directorio donde el archivo de parámetros y el archivo de passwords fueron creados, eliminarlos en caso de existir. Para el ejemplo, estos archivos fueron creados en \$ORACLE\_HOME/dbs
- Eliminar en caso de ser necesario la entrada que se agregó en /etc/oratab en donde se hace referencia al nombre de la base de datos difunta.

### **2.5. ADMINISTRACIÓN DE LOS PARÁMETROS DE LA BASE DE DATOS.**



- En secciones anteriores se revisó el concepto de PFILE y SPFILE
- Se mencionó la conveniencia de contar con un archivo binario que permita ser administrado de forma dinámica a través de sentencias SQL llamado *Server Parameter File* SPFILE.
- Al ejecutar el comando `startup`, la instancia realiza una búsqueda de este archivo en el siguiente orden:
  - Si la instrucción `startup` no contiene el parámetro `pfile`, se busca el SPFILE en la ubicación por default.
  - En caso de no encontrarlo, se busca el PFILE en la ubicación por default.
- La recomendación es hacer uso de un SPFILE en su ubicación por default.

### 2.5.1. Creación de un SPFILE.

- En secciones anteriores se realizó un ejercicio en el que a partir de un PFILE se realiza la creación de un SPFILE empleando la instrucción `create spfile from pfile;`
- Esta instrucción puede ser ejecutada antes o después de haber iniciado la instancia. Si la instancia ya está iniciada, no es posible sobrescribir el archivo que están en uso.

#### Ejemplos:

Considerando que la instancia está iniciada en status OPEN, crear los siguientes archivos PFILE, abrirlos, revisar diferencias.

- Crear un PFILE a partir de los parámetros existentes en memoria en el directorio `/tmp`

```
create pfile='/tmp/pfile.txt' from memory;
```

- Crear un PFILE a partir del SPFILE en el directorio `/tmp`

```
create pfile='/tmp/pfile2.txt' from spfile;
```

### 2.5.2. Métodos empleados para consultar parámetros.

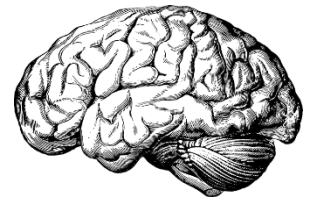
| Método                            | Descripción  |
|-----------------------------------|--|
| <code>show parameters</code>      | Comando SQL *Plus. Muestra el valor de un parámetro empleado en la instancia actual.   |
| <code>show sparameters</code>     | Comando SQL *Plus. Muestra el valor de un parámetro del SPFILE. El valor de un parámetro puede ser ajustado en memoria.  |
| <code>v\$parameter</code>         | Muestra los valores de los parámetros empleados para la sesión actual del usuario.   |
| <code>v\$parameter2</code>        | La diferencia con la anterior, es que los parámetros que aceptan una lista de valores son mostrados como registros de la vista (cada elemento de la lista es un registro). |
| <code>v\$system_parameter</code>  | Muestra los valores de los parámetros empleados pero ahora a nivel de instancia.   |
| <code>v\$system_parameter2</code> | La diferencia con la anterior, es que los parámetros que aceptan una lista de valores son mostrados como registros de la vista (cada elemento de la lista es un registro). |
| <code>v\$sparameter</code>        | Contiene la lista de parámetros especificados en el SPFILE   |

### 2.5.3. Parámetros básicos.

Existen alrededor de 30 parámetros considerados como básicos, los cuales deben ser identificados y estar familiarizados con ellos.

**Ejemplo:**

```
select name,type,value
from v$parameter
where isbasic='TRUE'
order by name;
```



| Nombre                      | Valor  | Default           |
|-----------------------------|--|-------------------|
| cluster_database            | Indica si la BD está configurada en modo Cluster (RAC) o en modo instancia simple.                                     | FALSE             |
| compatible                  | El número de versión que será emulada para garantizar compatibilidad.<br>Típicamente coincide con la versión de la BD. |                   |
| control_files               | Nombre y ubicación de los archivos de control.   | ?=/dbs/cntrl@.dbf |
| db_block_size               | Tamaño del bloque de datos por default empleado para dar formato a los data files.                                     | 8192              |
| db_create_file_dest         | La ubicación por default de los data files.  | NONE              |
| db_create_online_log_dest_1 | Ubicación por default de los Redo Log files  | NONE              |
| db_create_online_log_dest_2 | Ubicación por default donde se almacenan las copias de los Archivos Redo Logs (Archivos multiplexados).                | NONE              |
| db_domain                   | Nombre del dominio de la base de datos.  | NULL              |
| db_name                     | Nombre de la base de datos.  | NONE              |
| db_recovery_file_dest       | La ubicación del área rápida de recuperación   | NONE              |
| db_recovery_file_dest_size  | La cantidad de datos que puede ser escrita en el área rápida de recuperación.  | 0                 |
| db_unique_name              | Identificador único que distingue a 2 bases de datos que tienen el mismo valor del parámetro db_name.                  | NONE              |
| instance_number             | Empleado en ambientes RAC para distinguir a 2 o más instancias que administran una misma base de datos.                | 0                 |
| log_archive_dest_1          | La ubicación donde se almacenan los Redo Logs archivados   | NONE              |
| log_archive_dest_2          | La ubicación donde se almacenan los Redo Logs archivados y multiplexados   | NONE              |
| log_archive_dest_state_1    | Indicador que verifica si la ubicación configurada en el parámetro log_archive_dest_1 estará habilitada o no.          | enable            |
| log_archive_dest_state_2    | Indicador que verifica si la ubicación configurada en el parámetro log_archive_dest_2 estará habilitada o              | enable            |
| nls_language                | El lenguaje empleado en la instancia. A partir de este valor se definen formatos (fechas, valores monetarios, etc).    | AMERICAN          |
| nls_territory               | La ubicación geográfica donde se encuentra la base de datos. A partir de este valor se definen formatos                | AMERICA           |

| Nombre                      | Valor  | Default    |
|-----------------------------|--|------------|
| open_cursors                | En número máximo de áreas de trabajo SQL que una sesión puede abrir de forma concurrente.  | 50         |
| pga_aggregate_target        | La cantidad total de memoria que la instancia puede asignar a la PGA.  | 0          |
| processes                   | El número máximo de procesos (incluyendo los procesos internos de la base de datos) que pueden conectarse a la instancia.  | 0          |
| remote_listener             | Lista de direcciones IP donde se encuentran listener ejecutándose. Esta lista es empleada por la instancia para conectarse o registrarse ante dichos listener. Una vez registrada, el listener podrá redirigir las peticiones que recibe hacia la instancia. | NULL       |
| remote_login_passwordfile   |  | exclusive  |
| sessions                    | El número máximo de sesiones concurrentes que pueden ser creadas en la instancia.  | 4294967295 |
| sga_target                  | El tamaño o cantidad de memoria asignada a la SGA  | 0          |
| shared_servers              | Numero de procesos creados para atender peticiones empleados para casos donde la BD se configura en modo compartido "Shared"   | 4294967294 |
| star_transformation_enabled | Bandera que le indica al optimizador de sentencias SQL si hace uso de conversiones tipo "estrella" para reescribirla.  | FALSE      |
| undo_tablespace             | En caso de emplear la administración automática de datos UNDO, esta parámetro indica la ubicación de los datos UNDO.   | NONE       |

#### 2.5.4. Modificación de parámetros.

Existen 2 tipos de parámetros:

- Parámetros dinámicos: Los cambios toman efecto de forma inmediata sin requerir reinicio.
- Parámetros estáticos: Los cambios no toman efecto en la instancia de forma inmediata. Se deben escribir al SPFILE y requieren reinicio.
- Para realizar la modificación de un parámetro se emplean las siguientes instrucciones

```
alter system set <parametro>=<valor> [scope={spfile|memory|both}]
```

```
alter session set <parametro>=<valor>
```

- Para el caso del parámetro `scope`, sus valores se describen a continuación:



| Cláusula scope | Descripción   |
|----------------|---|
| scope=spfile   | <ul style="list-style-type: none"> <li>• El cambio es aplicado únicamente en el SPFILE.</li> <li>• Los cambios toman efecto hasta que se reinicia la instancia. El cambio se vuelve permanente.</li> <li>• Valido para parámetros estáticos.</li> </ul> |
| scope=memory   | <ul style="list-style-type: none"> <li>• El cambio se aplica directamente en memoria.</li> <li>• El cambio es aplicado a la instancia actual.</li> </ul>  |

| Cláusula scope | Descripción  |
|----------------|--|
|                | <ul style="list-style-type: none"> <li>El cambio no es permanente ya que no se afecta al SPFILE</li> <li>Válido para parámetros dinámicos.</li> </ul>  |
| scope=both     | <ul style="list-style-type: none"> <li>El cambio se aplica tanto en memoria como en el SPFILE.</li> <li>Su cambio es inmediato y permanente.</li> <li>Válido para parámetros dinámicos.</li> </ul> |

Valores por default:


- scope=both si la instancia fue levantada con un SPFILE
- scope=memory si la instancia fue levantada con un PFILE

### Ejemplo:

```
alter system set sec_max_failed_login_attempts=3
comment='reduce from 10 for tighter security.' scope=spfile;
```

En el ejemplo se cambia a 3 el número máximo de intentos fallidos al hacer login antes de cortar la conexión.

- Para regresar al valor por default de un parámetro se emplea la instrucción `alter system reset`
- Dependiendo de la naturaleza del parámetro se emplea alguna de las 2 sentencias anteriores, o inclusive ambas.
- La vista del diccionario de datos `v$parameter` define los siguientes atributos cuyos valores permiten conocer la forma en la que se puede actualizar el valor de un parámetro:

| Nombre de la columna /atributo   | Descripción   |   |
|--|---|---|
| isses_modifiable   | Indica si el parámetro puede ser modificado con la instrucción <code>alter session (true/false)</code>  |   |
|  issys_modifiable | Indica si el parámetro puede ser modificado con la instrucción <code>alter system</code> y el momento en el que el cambio toma efecto. La columna puede tener los siguientes valores: |   |
|  | Valor de la columna   | Descripción   |
|  | immediate   | <ul style="list-style-type: none"> <li>El parámetro puede ser modificado con <code>alter system</code> sin importar si se emplea PFILE o SPFILE para levantar la instancia.</li> <li>El cambio toma efecto de forma inmediata.</li> </ul>   |
|  | deferred  | <ul style="list-style-type: none"> <li>El Parámetro puede ser cambiado con <code>alter system</code> sin importar si se emplea PFILE o SPFILE para levantar la instancia.</li> <li>El cambio toma efecto en <b>sesiones</b> subsecuentes.</li> </ul>  |
|  | false   | <ul style="list-style-type: none"> <li>El Parámetro no puede ser cambiado con <code>alter system</code> a menos que la instancia haya sido iniciada con un SPFILE.</li> <li>Lo anterior se debe a que el cambio DEBE almacenarse para que este sea permanente. Si la instancia fue levantada con un PFILE, el nuevo valor no podría ser almacenado, recordar que la instancia administra los parámetros únicamente en el SPFILE.</li> </ul> |

- El cambio toma efecto en subsecuentes **instancias** (al reiniciar).

Tlp: Todos los parámetros cuyo valor de la columna `issys_modifiable` es FALSE son *estáticos*. De lo contrario serán dinámicos.

### Ejercicio en clase 1



- A. Generar una sentencia SQL que muestre los parámetros de la siguiente figura ordenados por nombre.

| NAME               | VALUE                 | DEFAULT_VALUE | ISDEFAULT | ISSES_MODIFIABLE | ISSYS_MODIFIABLE |
|--------------------|-----------------------|---------------|-----------|------------------|------------------|
| 1 db_domain        | (null)                | NULL          | TRUE      | FALSE            | FALSE            |
| 2 java_jit_enabled | TRUE                  | TRUE          | TRUE      | TRUE             | IMMEDIATE        |
| 3 nls_date_format  | dd/mm/yyyy hh24:mi:ss | NONE          | TRUE      | TRUE             | FALSE            |
| 4 processes        | 360                   | 0             | TRUE      | FALSE            | FALSE            |
| 5 sessions         | 568                   | 4294967295    | TRUE      | FALSE            | IMMEDIATE        |

- B. Generar la sentencia necesaria para modificar su valor con base a los requerimientos de la siguiente tabla. Si el valor del parámetro es una cadena, especificarla entre comillas simples.

| Parámetro        | Nuevo Valor           | Requerimientos   |
|------------------|-----------------------|--|
| nls_date_format  | dd/mm/yyyy hh24:mi:ss | El cambio solo debe aplicarse para la sesión actual.   |
| processes        | 300                   | El cambio debe ser permanente  |
| sessions         | 500                   | El cambio debe ser permanente  |
| java_jit_enabled | false                 | <p>Este parámetro habilita o deshabilita el compilador JIT de Java. Por default está habilitado y permite incrementar el desempeño de aplicaciones Java que corren en la BD ya que el compilador genera código nativo. Sus valores son TRUE, FALSE.</p> <ol style="list-style-type: none"> <li>1) Deshabilitar el compilador únicamente a nivel sesión</li> <li>2) Comprobar el resultado consultando su valor:</li> <li>3) ¿Qué valor tendrá el parámetro si se abre una nueva sesión?</li> <li>4) Deshabilitar el compilador a nivel instancia.</li> <li>5) Comprobar el resultado del inciso anterior.</li> </ol> |

**Ejercicio**  
práctico 04

