

TEMA 1

Conceptos básicos en Administración de Bases de Datos.

1.1. ADMINISTRACIÓN DE BASES DE DATOS.

Para comprender de forma adecuada y desde un enfoque práctico, los conceptos del curso hacen referencia a la base de datos Oracle en ambientes Linux/Unix. Sin embargo, en la mayoría de los casos, dichos conceptos pueden aplicarse de forma muy similar a otros manejadores y sistemas operativos aplicando ligeras modificaciones.

¿Qué debemos aprender en el curso?

- Comprender el funcionamiento interno de una base de datos.
- Con base al conocimiento anterior, ser capaces de contar con una BD que opere de manera óptima bajo diversos ambientes o escenarios.
- Capacidad de prevenir y resolver problemas asociados con su correcto funcionamiento: “Cuidar la Salud de la Base de Datos” tanto de forma proactiva como reactiva.
- Lo anterior permite contar con el activo más preciado de cualquier organización:
 - *Datos consistentes, verídicos y completos.*

Ejercicio
práctico 01



¿Debo aprender Administración de BDs aunque no tenga planeado ser DBA?

- *¡Por supuesto!* Conocer el funcionamiento interno y la forma en la que se debe emplear una base de datos para que opere de manera adecuada, permite desarrollar aplicaciones que hagan uso óptimo de la base de datos, produciendo una armonía perfecta entre Aplicaciones, DBMS y la propia BD.



1.2. TAREAS FUNDAMENTALES DEL ADMINISTRADOR DE LA BASE DE DATOS (DBA).

La siguiente lista muestra las principales tareas que todo administrador de una base de datos debe realizar.

Tarea: Evaluar el Hardware en el que el Sistema de Base de Datos estará operando. Se deben responder preguntas como:

- ¿Cuántos discos serán necesarios?
- ¿Cuántos cartuchos de cinta se requieren para aplicar respaldos?
- ¿Qué capacidades de memoria y procesador requiere la BD?
- ¿Qué características debe contar la Red de Datos donde operará la BD?



Tarea: Instalación del Software de la Base De Datos.

- Instalación y configuración del software necesario para crear una BD.
- Instalación de herramientas de administración, generalmente en ubicaciones remotas. Esto particularmente en ambientes donde la administración se realiza de forma remota.

Tarea: Planeación de la Base de Datos.

- Capacidad de revisión y aprobación de diseños lógicos de bases de datos.
- Diseño de las estructuras físicas de almacenamiento: número de data files, particiones, tablespaces, distribución de data files en discos
- Establecer una correcta correspondencia entre el diseño lógico (modelos relacionales) y el diseño físico de la BD.
- Estrategia de respaldos. El diseño lógico puede verse modificado para beneficiar el esquema de respaldos.
- Planear o considerar el crecimiento de la BD a lo largo del tiempo.

Tarea: Planeación de funciones básicas de la BD:



- Creación de bases de datos
- Iniciar y detener bases de datos.
- Respalidar bases de datos.
- Actualizar el software de la Base de datos
- Aplicación de parches (corrección de errores, mejoras o seguridad).
- Administrar usuarios de la BD.
- Replicar o clonar ambientes de Bases de Datos en otros servidores.
- Trabajar constantemente en el afinamiento y desempeño de la BD.

1.3. HERRAMIENTAS DE ADMINISTRACIÓN

Existen 2 tipos principales de herramientas para realizar la administración de la base de datos:

- A línea de comandos
 - Empleando sentencias SQL y comandos propios en SQL*Plus
 - Empleando Scripts SQL, shell Scripts, RMAN scripts, Perl scripts.
- A través del uso de herramientas gráficas.
 - SQL Developer.
 - *Oracle Enterprise Manager Database Express* (EM Express) o *Oracle Enterprise Manager Cloud Control* (Cloud Control).
 - Interfaces de administración para servicios *Cloud Computing*.



- A través de herramientas de administración proporcionados por terceros.

1.3.1. Programación PL/SQL

La programación y creación de Scripts SQL es una habilidad fundamental para realizar administración de bases de datos. Gran parte de las actividades de administración suelen automatizarse a través del uso de programas PL/SQL. A nivel básico, es indispensable contar con los siguientes conceptos:

- Manejo de elementos básicos de programación: uso de variables, sentencias de control, operadores.
- Programación de bloques anónimos, procedimientos almacenados y funciones.
- Manejo de cursores
- Manejo de excepciones
- Ejecución de código SQL de forma dinámica
- Conocimientos sólidos en SQL, manejo de funciones.

1.3.2. Programación en Shell.

Como administrador de base de datos particularmente en sistemas Unix/Linux, el uso de la programación en Shell representa una de las herramientas básicas e indispensables. Los principales aspectos de programación que deben dominarse son:

- Lógica básica de programación: declaración de variables, uso de funciones, sentencias de control.
- Manejo de variables de entorno.
- Manipulación de archivos leer, escribir, administrar permisos de acceso.
- Manejo de comandos básicos: `cat`, `grep`, `find`, `ps`, `chgrp`, `chown`, `more`, `cp`, `mv`, `ln`, redireccionamiento de salida, etc.

**Ejercicio
práctico 02**



1.3.3. Uso de SQL*Plus

Representa la principal herramienta a línea de comandos para interactuar y administrar una instancia de base de datos Oracle.

La primera actividad a realizar dentro de la herramienta es realizar la conexión hacia la instancia:

- *De forma local:* El servidor se encuentra en la misma máquina
- *De forma remota:* El cliente remoto instala un pequeño software llamado Cliente de Oracle mismo que realizará una conexión hacia un servidor remoto. Dicho servidor deberá contar un servicio llamado *listener* encargado de atender peticiones remotas.

1.3.3.1. Revisión de variables de entorno.

En cualquier instalación de una base de datos se requieren de variables de entorno previamente configuradas, es especial cuando se trabaja a línea de comandos.

- Para el caso de Oracle, se requieren las siguientes variables como mínimo para que SQL*Plus pueda operar correctamente

Variable	Descripción
ORACLE_SID	Indica el nombre de la instancia de la Base de datos a la que se desea conectar
ORACLE_HOME	Directorio donde se encuentra la instalación y ejecutables de la base de datos

LD_LIBRARY_PATH	Lista de directorios donde se encuentran librerías del sistema (típicamente escritas en lenguaje C) requeridas para el correcto funcionamiento tanto de la BD como de SQL*Plus
-----------------	--

1.3.3.2. Personalización de SQL*Plus

Para tener una mejor experiencia con SQL *Plus en ambientes Linux, aplicar las siguientes configuraciones:

- *Instalar y configurar rlwrap*

```
sudo yum install rlwrap
```

```
#crear el script en caso de no existir.
```

```
sudo nano /etc/profile.d/oracle-env.sh
```

```
#incluir las siguientes líneas
```

```
alias sqlplus='rlwrap sqlplus'
```



- *Configurar glogin.sql* para personalizar el uso del buffer y el formato del prompt.

```
su -l oracle
```

```
cd $ORACLE_HOME/sqlplus/admin
```

```
nano glogin.sql
```

```
--agregar las siguientes líneas
```

```
--define el editor para el buffer
```

```
define _editor=nano
```

```
--personalizar el prompt
```

```
define prompt_value=idle
```

```
col prompt_name new_value prompt_value
```

```
col prompt_name noprint
```

```
set heading off
```

```
set termout off
```

```
select lower(sys_context('userenv','current_user'))
```

```
||'@'||sys_context('userenv','db_name')) as prompt_name
```

```
from dual;
```

```
set sqlprompt '&prompt_value>'
```

```
set heading on
```

```
set termout on
```

```
col prompt_name print
```

- En general, estas líneas personalizan el prompt “SQL>” para mostrar el usuario y el nombre de la base de datos a la que se está conectado.
- Lo anterior permite identificar fácil y visualmente la instancia y el usuario actuales, en especial en servidores donde se tienen varias bases de datos creadas y evitar así confusiones.
- La instrucción `define _editor=nano` establece al editor nano como herramienta para editar/modificar la última sentencia SQL capturada en sqlplus a través del uso de comando edit. Revisar el documento `BD/practicas/practica7/practica7-previo.pdf` para mayores detalles.

1.3.3.3. Accediendo a SQL *Plus

A nivel básico, la sintaxis para autenticar a un usuario empleando el diccionario de datos de la BD se muestra a continuación (solo se muestran las opciones más comunes).

```
sqlplus [<usuario>]/[<password>|<nolog>][@bd] [as sysdba]
```


Dentro del SQL*Plus es posible autenticarse empleando el comando connect

```
conn[ect] [logon] [as {sysoper|sysdba|sysbackup|sysdg syskm|sysrac}]
```

Donde [logon] es:

```
{username | /}[@connect_identifier]
```

La siguiente tabla muestra un resumen de las formas comúnmente empleadas para acceder a SQL*Plus. En los siguientes temas se revisan los conceptos para comprender los mecanismos de autenticación así como los privilegios de administración.

Comando	Descripción
<pre>su -l oracle</pre> <pre>sqlplus / as sysdba</pre> 	<ul style="list-style-type: none"> • Observar que al invocar al comando <code>sqlplus</code> no se especifica ni usuario ni password. • Este comando accede a SQL*Plus como usuario SYS empleando el sistema operativo como mecanismo de autenticación • Notar la cláusula <code>as sysdba</code>. Esta instrucción solo debe ser empleada para autenticarse como usuario SYS o como cualquier otro usuario que tenga rol de administrador (dba) en la Base de datos.
<pre>sqlplus /nolog</pre>	<ul style="list-style-type: none"> • Accede a SQL*Plus sin autenticar. En este caso, el usuario puede invocar comandos que no requieren autenticación. • El usuario que accede de esta manera está muy limitado a ejecutar comandos en SQL*Plus. Muy pocos comandos pueden ser ejecutados en este modo. Por ejemplo: <code>show user</code>, etc. • Se puede emplear el comando <code>connect</code> para autenticarse sin salir de SQL*Plus
<pre>sqlplus <login>/<password></pre>	<ul style="list-style-type: none"> • Empleado para acceder a SQL*Plus con login y password de algún usuario existente en la base de datos. • La instancia ya debió haber sido iniciada y el usuario debe existir dentro de la base de datos, se emplea el diccionario de datos para autenticar. <u>Ejemplo:</u> <code>sqlplus jorge/miPassword</code>
<pre>sqlplus <login></pre>	<ul style="list-style-type: none"> • Similar al caso anterior, pero sin especificar password. • Se considera una forma segura de acceder a SQL*Plus ya que el password no se escribe de forma explícita. • Sql *Plus detecta que no se especifica el password y lo solicitará ocultando los caracteres empleando <code>*****</code>
<pre>su -l myuser</pre> <pre>sqlplus sys as sysdba</pre>	<ul style="list-style-type: none"> • En este ejemplo se intenta acceder a SQL*Plus sin emplear autenticación del sistema operativo ya que se emplea al usuario <code>myuser</code> que no pertenece al grupo de usuarios dueños de la BD. • En caso de no usar sistema operativo como medio de autenticación, se emplea el archivo de passwords.
<pre>connect / as sysdba</pre> <pre>connect sys as sysdba</pre>	<ul style="list-style-type: none"> • Estos comandos funcionan similar a los ejemplos anteriores correspondientes.

Comando	Descripción
connect sys/<password> as sysdba connect <login>/<password>	<ul style="list-style-type: none"> La diferencia es que se hace uso del comando <code>connect</code> o <code>conn</code> en su forma corta. El comando <code>connect</code> es un comando de SQL*Plus por lo que se debe ejecutar dentro de la herramienta. Útil cuando se desea cambiar de usuario sin tener que salir de SQL*Plus

1.4. IDENTIFICACIÓN DE LA VERSIÓN DE LA BASE DE DATOS.

A nivel general, cualquier base de datos hace uso de una serie de caracteres y/o dígitos para especificar la versión de la base de datos a emplear.

- Para el caso de Oracle se emplean 5 dígitos.
- Se definen 2 tipos de versiones:
 - `version`
 - `version_full` (Actualización de la versión mayor de forma trimestral).



Formada por los siguientes componentes. Ejemplo: **18.0.0.0.0**

18	0	0	0	0
Major Release.	Release Update Version.	Release update Revision version.	version Increment.	Reservado para uso futuro.
Formada por los 2 últimos dígitos del año en el que se libera por primera vez.	Numero de actualización. Oracle produce actualizaciones trimestrales.	Representa el número de revisión de la actualización trimestral.	Empleado para actualizaciones en versiones futuras	

1.5. CUENTAS DE USUARIO DE ADMINISTRACIÓN Y PRIVILEGIOS.

Dependiendo el sistema operativo, una base de datos puede requerir de las siguientes cuentas de usuarios para poder ser administrada:

- Cuenta de usuario administrador a *nivel del sistema operativo*. Permite administrar el control de acceso a los archivos que integran el software de la base de datos (depende del sistema operativo). Para el caso de sistemas Unix/Linux se requiere la existencia del usuario `oracle`.
- Cuentas de usuarios administradores a *nivel de la base de datos*.

1.5.1. Cuentas de usuario a nivel de la base de datos.

Oracle requiere la existencia de los siguientes usuarios encargados de administrar diferentes áreas de la BD. Estos usuarios son creados al momento de crear una base de datos.

Nombre del usuario	Descripción
<code>sys</code>	Cuenta con todos los privilegios asignados. Todas las tablas y vistas del diccionario de datos son almacenadas en el esquema SYS. Usuarios no modifican estas vistas y tablas, y <u>tampoco debe crear nuevas</u> .

system	Se le asigna el rol DBA. Se emplea para crear tablas y vistas adicionales que muestran información de administración empleadas por herramientas y funcionalidades adicionales que ofrece la BD. De forma similar, no se debe emplear este esquema para crear nuevas tablas.
sysbackup	Facilita el manejo y administración de respaldos empleando SQL *Plus o RMAN (Recovery Manager).
sysdg	Se encarga de realizar operaciones con funcionalidades de "Data Guard". Data Guard ofrece la implementación de requerimientos como Alta disponibilidad, protección de datos, recuperación ante desastres, etc., a través de operaciones de creación, mantenimiento y monitoreo de BDs.
syskm	Empleado para realizar y ejecutar funcionalidades asociadas con Cifrado de datos, generación de KeyStores, etc.
sysrac	Empleado para realizar operaciones de administración en ambientes clusterizados a través de Oracle RAC. Conexión entre nodos del clúster, etc.

- Para las últimas 4 cuentas, existe un privilegio de administración con el mismo nombre.
- Existe un rol muy importante llamado DBA. Este rol contiene prácticamente todos los privilegios existentes en una BD. Se crea al crear una base de datos.



1.5.2. Privilegios de administración.

Requeridos por los usuarios administradores para realizar sus tareas. No confundir este concepto con el concepto de Rol.

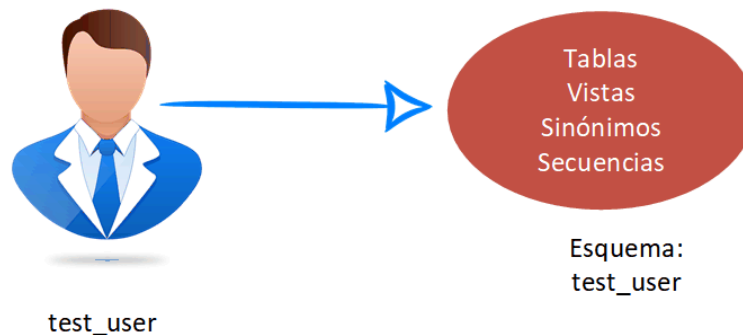
Privilegio	Operaciones principales permitidas.
sysdba	<code>startup, shutdown</code> <code>alter database: open, mount, back up, change character set</code> <code>create database</code> <code>drop database</code> <code>create spfile</code> <code>alter database archivelog</code> <code>alter database recover</code> Permite realizar la mayoría de las operaciones incluida la posibilidad de ver los datos de los usuarios, representa el privilegio con el mayor poder.
sysoper	<code>startup, shutdown</code> <code>create spfile</code> <code>alter database: open, mount, back up</code> <code>alter database archivelog</code> <code>alter database recover</code> (solo recuperación completa). No puede ver los datos de los usuarios.
sysbackup	La lista de operaciones que puede realizar este privilegio está definida a través del rol sysbackup
sysdg	La lista de operaciones que puede realizar este privilegio está definida a través del rol sysdg
syskm	La lista de operaciones que puede realizar este privilegio está definida a través del rol syskm
sysrac	La lista de operaciones que puede realizar este privilegio está definida a través del rol sysrac

Para hacer uso de los privilegios de administración se requiere realizar lo siguiente:

- El usuario debe tener asignado alguno de estos privilegios, por ejemplo, el usuario SYS se los ha asignado anteriormente.
- Para hacer uso de dicho privilegio, el usuario debe conectarse a la BD indicando el privilegio de administración con el que desea conectarse empleando la cláusula `as`.

Cuando un usuario se conecta a la BD sin hacer uso de un privilegio de administración ocurre lo siguiente:

- Al entrar a sesión, el usuario tiene asociado a un esquema cuyo nombre coincide con su nombre de usuario. Por ejemplo, el usuario `test_user` tiene un esquema llamado `test_user`.



Cuando un usuario se conecta a la BD empleando un privilegio de administración, la regla anterior cambia.

- Al conectarse como `sysdba`, el esquema asignado es `sys`.
- Al conectarse como `sysoper`, el esquema asignado es `public`.
- Al conectarse como `sysbackup`, `sysdg`, y `sysrac` el esquema asignado es `sys`.
- Al conectarse como `syskm`, el esquema asignado es `syskm`

Otra característica del uso de los privilegios de administración es que el usuario también cambia:

- Al conectarse como `sysdba`, el usuario asignado es `sys`.
- Al conectarse como `sysoper`, el usuario asignado es `public`.
- Al conectarse como `sysbackup`, `sysdg`, `syskm` y `sysrac` el usuario asignado es `sysbackup`, `sysdg`, `syskm` y `sysrac` respectivamente.

Para entender lo anterior, realizar el siguiente ejercicio:

Ejercicio en clase 1



- Conectarse a la BD como usuario `sys` empleando el privilegio de administración `sysdba`.
- Crear un usuario empleando el nombre del alumno: `<nombre>01` con cuota ilimitada en el `tablespace users`.
- Asignarle privilegio para crear sesión y crear tablas (ojo, estos no son privilegios administrativos).
- Asignarle los siguientes privilegios administrativos: `sysdba` y `sysoper`.
- Conectarse como usuario `<nombre>01` sin hacer uso de los privilegios administrativos.
 - Revisar el nombre de usuario asignado.
 - Revisar el nombre del esquema asignado. ¿Qué valor debería obtenerse?
 - Crear una tabla llamada `test` con un campo `id` numérico.

- d. Ejecutar `select * from test` para comprobar su existencia.
- F. Salir de sesión
- G. Conectarse ahora como usuario `<nombre>01` con el privilegio de administración `sysdba`
 - a. ¿Quién debería ser el usuario conectado en la BD?
 - b. ¿Cuál debería ser el nombre del esquema asociado al usuario?
 - c. ¿Qué le sucedió a la tabla `test`?
- H. Hacer el mismo ejercicio del inciso anterior, pero ahora con el privilegio `sysoper`.



1.5.3. Métodos de autenticación para usuarios con privilegios de administración

- A través del diccionario de datos (cuando la BD está iniciada).
- A través de un archivo de passwords (útil cuando la BD no está iniciada).
- A través del sistema operativo.
- Empleando un servicio de autenticación externo (directory-based authentication service), por ejemplo, Oracle Internet Directory.

1.5.3.1. Autenticación vía sistema operativo.

Para realizar la autenticación vía sistema operativo, el usuario debe pertenecer a un grupo especial. La siguiente tabla ilustra los grupos del sistema operativo en Linux/Unix y su correspondiente privilegio.

Al pertenecer a alguno de los grupos de usuarios, la autenticación se otorga sin tener que especificar usuario y password.

Nombre del grupo del sistema operativo	Privilegio de administración.
dba, oinstall	sysdba
oper	sysoper
backupdba	sysbackup
dgdba	sysdg
kmdba	syskm
racdba	sysrac

Ejemplos de conexión empleando el sistema operativo (el usuario pertenece a alguno de los grupos anteriores).

```
connect / as sysdba
connect / as sysoper
```

1.5.3.2. Autenticación empleando archivo de passwords.

- El archivo de passwords se crea durante el proceso de creación de la base de datos empleando `dbca` (Database Configuration Assistant).
- El archivo debe contar con un nombre y ubicación con base a las siguientes reglas:
 - En Unix/Linux el nombre debe ser `orapwORACLE_SID`. `ORACLE_SID` corresponde al valor de la variable de entorno revisada anteriormente.
 - Debe ubicarse en `$ORACLE_BASE/dbs` para casos donde se ha decidido definir al directorio `ORACLE_HOME` como *read only*, (funcionalidad agregada en 18c). De lo contrario se ubica en `$ORACLE_HOME/dbs`.
- Para hacer uso del archivo de passwords, un usuario con privilegios de administración podrá autenticarse en SQL*Plus aun con la base de datos detenida empleando las siguientes instrucciones:

Ejemplo:

```
sqlplus /nolog
connect sys as sysdba
```

o de forma equivalente:

```
sqlplus sys as sysdba
```

- En ambos casos, el sistema le solicitará el password y será validado contra el archivo de passwords.

En los ejemplos anteriores, el usuario que ejecuta las instrucciones no requiere ser autenticado vía Sistema operativo. En caso de que el usuario pertenezca a alguno de los grupos mencionados anteriormente, se empleará autenticación por sistema operativo la cual tiene mayor precedencia.

Ejercicio en clase 2.

- A. Analizar las siguientes instrucciones y contestar la pregunta indicada en los comentarios.

```
jorge@lap-red-mint ~ $ su -l oracle
Password:
oracle@lap-red-mint:~$ sqlplus /nolog
```

```
-SQL> --¿Qué sucederá al ejecutar la siguiente instrucción?
-SQL> connect usr_noexiste as sysdba
```

- Una forma de alimentar el archivo de password es a través de SQL*Plus. El archivo se modifica automáticamente cuando se le asigna un privilegio de administración a algún usuario. Por ejemplo, al ejecutar la siguiente instrucción se crea una nueva entrada al archivo para que el usuario jorge01 se pueda autenticar empleando el archivo de passwords.

```
SYS-SQL> grant sysdba, sysoper to jorge01;
```

Ejercicio en clase 3

- Crear un usuario <nombre>01 en caso de no existir.
- Asignarle el privilegio sysdba en caso de no existir
- Consultar la vista v\$pwfile_users la cual permite visualizar a los usuarios que cuentan con privilegios de administración. Confirmar que el usuario <alumno> se encuentra en dicha lista.
- Detener la instancia, salir de SQL*Plus, y en caso de estar conectado al servidor como usuario oracle, cerrar sesión.
- Sin entrar a sesión con el usuario oracle, realizar la autenticación en SQL *Plus para hacer uso del archivo de passwords.
- Levantar la instancia.
- Eliminar el privilegio de administración al usuario <nombre>01.
- Revisar el contenido de la vista para validar los resultados.
- Revisar la fecha de modificación del archivo de passwords para comprobar su actualización.

- Para realizar la administración del archivo de passwords se emplea el comando `orapwd`.

```
orapwd FILE=filename
[FORCE={y|n}]
[ASM={y|n}]
[DBUNIQUENAME=dbname]
[FORMAT={12.2|12}]
[SYS={y|n|password|external('sys-external-name')|
  global('sys-directory-DN')}]
[SYSBACKUP={y|n|password|
  external('sysbackup-external-name')|global('sysbackup-directory-DN')}]
[SYSDG={y|n|password|external('sysdg-external-name')|
  global('sysdg-directory-DN')}]
[SYSKM={y|n|password|external('syskm-external-name')|
  global('syskm-directory-DN')}]
[DELETE={y|n}]
[INPUT_FILE=input-fname]
```



Ejercicio en clase 4

Simulación de pérdida del archivo de passwords.

- Detener la instancia en caso de estar levantada.
- Conectarse como usuario `oracle` en s.o.
- Mover el archivo de passwords a `/home/oracle/backups` para simular que este se ha perdido o se ha dañado.
- ¿Qué pasará al no existir el archivo de passwords?
 - ¿Levantará la instancia?
 - ¿Quién podría autenticar?
 - ¿Qué contendría la vista `v$pwfile_users`?
 - Asumiendo que el archivo se perdió, ¿Habría forma de recuperarse de esta pérdida? En caso afirmativo, generar las acciones necesarias para reconstruir el archivo.

Con base a los ejercicios anteriores, se tiene el siguiente resumen:

- Autenticación por sistema operativo tiene mayor prioridad. Por lo tanto, si el usuario está autenticado a nivel del sistema operativo y pertenece a los grupos vistos anteriormente, podría acceder con privilegios de administración.
- Si se desea hacer uso del archivo de passwords, emplear `sqlplus sys as sysdba`, `sqlplus jorge as sysdba`, etc.
 - Solo usuarios con privilegios de administración podrán aparecer en el archivo de passwords.
 - El usuario `sys` por definición tiene el privilegio `sysdba` y por tanto está en el archivo de passwords.
 - A partir de la versión 12c, el usuario `sys` **siempre** es autenticado por archivo de passwords.
 - Para que el usuario `jorge` aparezca en el archivo de password, se le tuvo que haber asignado el privilegio de administración: `grant sysdba to jorge`.
- Si la autenticación por sistema operativo falla, el archivo de password **siempre** se emplea para autenticar usuarios con privilegios de autenticación.
- Si el archivo de passwords se pierde o se daña, la única forma de autenticar a un usuario de administración será a través del sistema operativo, aunque la instancia esté iniciada.

5. Finalmente, la autenticación por diccionario de datos se emplea para usuarios que no tienen privilegios de administración.

Ejercicio
práctico 04

