

PRACTICA 7 – PREVIO
SQL *PLUS Y LENGUAJE DE DEFINICIÓN DE DATOS (DDL)

1.1. INTRODUCCIÓN.

En este documento se presentan conceptos básicos, lineamientos y buenas prácticas que deberán ser empleadas al interactuar con una base de datos Oracle, en especial con la herramienta SQL *Plus. Es importante que el aprendizaje adquirido al realizar las actividades de este documento previo sea aplicado en prácticas posteriores. El documento puede emplearse tanto para las asignaturas de Bases de Datos como para Bases de Datos Distribuidas.

1.2. LINEAMIENTOS PARA CREAR SCRIPTS E INCLUSIÓN DE CÓDIGO EN REPORTE

Las prácticas que se realizarán a partir de este momento requieren la inclusión de código SQL. Parte de la calificación del reporte incluye el formato y su correcta visualización del archivo en el reporte. Considerar los siguientes lineamientos que deben ser aplicados al momento de generar un reporte.

- Todos los scripts SQL deberán llevar el siguiente encabezado:

```
--@Autor:           <Nombre del autor o autores>
--@Fecha creación:  <Fecha de creación>
--@Descripción:     <Breve descripción del contenido del script>
```

- Para los casos en donde se solicite incluir código fuente en el reporte se deberá emplear un formato adecuado para que este sea entendible y no genere demasiado espacio. Las líneas de código no deben exceder los 80 caracteres.
- Emplear el tipo de Fuente Courier New tamaño 8. Alineado a la izquierda (no justificar).
- Se recomienda ampliamente emplear un editor SQL que permita resaltar la sintaxis SQL empleando colores, así como algunas opciones de formato. Ejemplos: Sublime Text Editor, Eclipse, IntelliJ IDEA, etc.
- Tomar como ejemplo las muestras que se incluyen en este documento. El código fue copiado de la herramienta Eclipse e IntelliJ IDEA. Estas herramientas permiten conservar el formato del código al copiar y pegar hacia el documento. Otra opción es incluir “Impresiones de pantalla” del código tal cual como se muestra en el editor. Se recomienda la primera opción.

Ejemplo:

Código SQL correctamente formateado listo para ser presentado en un reporte:

```
set serveroutput on
declare
    --declaración del cursor
    cursor cur_datos_profesor is
    select p.nombre as nombre_profesor, p.apellido_paterno, p.apellido_materno,
           a.nombre as nombre_asignatura, count(*) cursos
    from profesor p, curso c, asignatura a
    where p.profesor_id=c.profesor_id
    and c.asignatura_id = a.asignatura_id
    group by p.nombre, p.apellido_paterno, p.apellido_materno, a.nombre;

begin
    dbms_output.put_line('resultados obtenidos');
    dbms_output.put_line(
        'nombre apellido paterno apellido materno asignatura #cursos');
    for p in cur_datos_profesor loop
        dbms_output.put_line( p.nombre_profesor
                               ||' , '||p.apellido_paterno
                               ||' , '||p.apellido_materno
                               ||' , '||p.nombre_asignatura
                               ||' , '||p.cursos);
    end loop;
end;
/
```

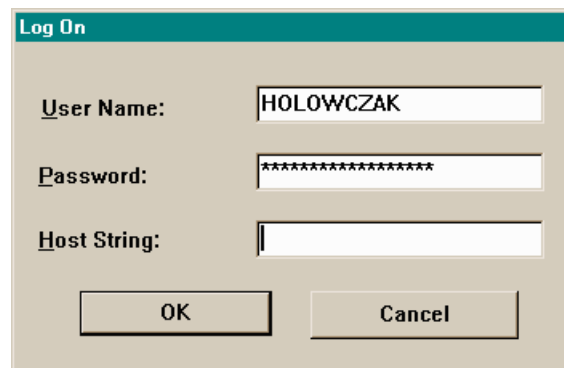
1.3. SQL *Plus

SQL *Plus es ampliamente utilizado por DBAs y desarrolladores para interactuar con una base de datos Oracle. Cualquier sentencia SQL puede ser ejecutada con SQL *Plus, programas escritos en PL/SQL, mostrar consultas, así como realizar tareas de administración en la base de datos. Esta herramienta se instala de manera automática al instalar Oracle.

SQL *Plus representa la interface a línea de comandos. A pesar de su gran utilidad, en algunas ocasiones su uso se vuelve un poco complicado principalmente por las limitantes de ser una herramienta a línea de comandos. Existen múltiples aplicaciones e interfaces gráficas que facilitan la interacción de un usuario con la instancia Oracle. Ejemplos: Oracle SQL Developer, TOAD (Windows), TORA (linux). A pesar de su existencia, es de vital importancia adquirir conocimientos sólidos en SQL *Plus, ya que, en ambientes de producción, normalmente no se cuenta con alguna herramienta gráfica. En prácticas subsecuentes se revisarán algunas de estas herramientas.

1.3.1. Entrando a SQL *PLUS

En sistemas Windows, para iniciar con SQL *PLUS, se muestra una ventana como la siguiente:



Host String se refiere al "Alias de la base de datos", este valor que se configuró en la práctica 2. Si no se especifica este valor, por default se emplea el valor de la variable de ambiente ORACLE_SID

1.4. ADMINISTRACIÓN DE SERVICIOS.

1.4.1. Iniciando el listener.

Como se mencionó en prácticas anteriores, el listener permite a clientes conectarse al servidor de base de datos, en especial clientes remotos empleando conexiones TCP/IP. La sintaxis para iniciar el listener es:

```
lsnrctl start <listener_name>
lsnrctl stop <listener_name>
lsnrctl status <listener_name>
```

Por default, el valor de <listener_name> es "LISTENER".

Cabe mencionar que si la conexión se realiza en el mismo servidor donde se encuentra la BD, no es necesario iniciar el listener.

1.4.2. Métodos de acceso en Sql *Plus.

- Solo el usuario SYS (administrador de la BD) puede realizar cualquier operación administrativa.
- Cuando la instancia no ha sido iniciada, no es posible autenticar usuarios ya que los datos de autenticación se almacenan en el diccionario de datos, por lo que podrá emplearse hasta que la instancia sea iniciada.
- Para resolver lo anterior, la autenticación del usuario se realiza a través del sistema operativo. El usuario oracle creado a nivel de sistema operativo en prácticas anteriores es el único usuario que puede adquirir la identidad del usuario SYS dentro de SQL *Plus sin ser autenticado a través del diccionario de datos.

```
su -l oracle (proporcionar password).
sqlplus / as sysdba
sql> show user
USER is "SYS"
```

A nivel básico, la sintaxis para autenticar a un usuario empleando el diccionario de datos de la BD es:

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

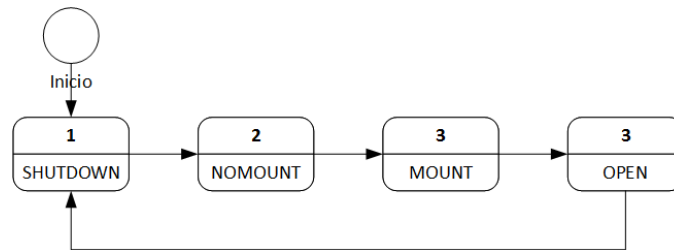
[@bd] es opcional y se emplea para conectarse a una instancia ubicada en un servidor remoto) .
[as sysdba] se emplea para únicamente para los usuarios administradores como el usuario SYS.

La siguiente tabla muestra un resumen de las formas comunmente empleadas para acceder a Sql *Plus:

Comando	descripción
<pre>su -l oracle (proporcionar password). 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 commando accede a Sql *Plus como usuario SYS empleando el sistema operativo como mecanismo de autenticación. Únicamente el usuario <code>oracle</code> del sistema operativo puede emplear esta estrategia ya que el es el dueño del software de la BD. En la práctica, solo los DBAs deberían conocer en password del usuario <code>Oracle</code> • Esta estrategia se emplea para acceder a Sql *Plus cuando el diccionario de datos no está disponible para autenticar. Comunmente se emplea para acceder a Sql *Plus para levantar la instancia. • Notar la clausula <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.
<code>sqlplus /nolog</code>	<ul style="list-style-type: none"> • Accede a Sql *Plus sin autenticar. En este caso, el usuario puede invocar comandos que no requieran 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
<code>sqlplus <login>/<password></code>	<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. Ejemplo: <code>sqlplus jorge/miPassword</code> • En este modo, no se requiere hacer uso del usuario <code>oracle</code> del sistema operativo. • Notar que no se hace uso de la instrucción <code>as dba</code> ya que el usuario no es administrador.
<code>sqlplus <login></code>	<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> • Para efectos del curso y para agilizar la interacción con la instancia, se puede especificar el password de forma explícita, pero no en la práctica. ¡El password se puede obtener aunque nadie observe al usuario teclear el password explícitamente!
<code>sqlplus sys as sysdba</code>	<ul style="list-style-type: none"> • Accede a Sql *Plus como usuario SYS empleando el diccionario de datos como mecanismo de autenticación en lugar de emplear el Sistema operativo. • La instancia debió haber sido iniciada anteriormente. • En este escenario no se requiere hacer uso del usuario <code>oracle</code> del Sistema operativo, pero si se requiere conocer el password que se le asignó al usuario SYS en la base de datos. • Para efectos de las prácticas de este curso, el password que se le asignó en practicas anteriores fue "system". • Similar a la estrategia de validación por Sistema operativo, solo los DBAs deberían conocer el password del usuario SYS.
<pre>connect / as sysdba connect sys as sysdba connect sys/<password> as sysdba connect <login>/<password></pre>	<ul style="list-style-type: none"> • Estos comandos funcionan similar a los ejemplos anteriores correspondientes. • La diferencia es que se hace uso del commando <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

1.4.3. Iniciar y detener la instancia.

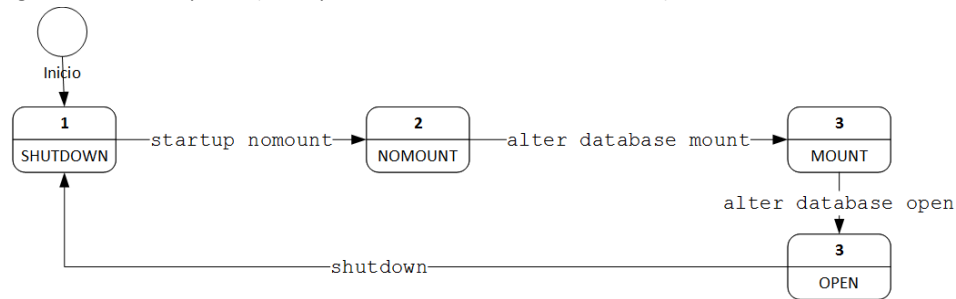
- Recordando, la base de datos y la instancia son 2 cosas separadas. Para iniciar a interactuar con la base de datos, esta debe pasar por varios estados:



- El estado inicial es shutdown.
- El comando `startup` se emplea para iniciar la instancia en alguno de los 3 estados siguientes: nomount, mount, open. Es decir:

Comando	Descripción
<code>startup nomount</code>	La instancia pasa de estado shutdown a nomount
<code>startup mount</code>	La instancia pasa de estado shutdown a nomount, y después a mount.
<code>startup open</code>	La instancia pasa del estado shutdown a nomount, después a mount y finalmente a open
<code>startup</code>	Si se especifica el comando <code>startup</code> sin ninguna de las 3 opciones anteriores, la base de datos pasa por todos los estados hasta llegar al estado open. Es decir, es equivalente al comando <code>startup open</code> Representa la forma más simple para levantar una instancia y poder comenzar a trabajar.

- Para comenzar a trabajar con la base de datos de forma normal, el estado de la instancia debe ser open.
- Los estados anteriores se emplean para realizar diversas acciones que no requiere el uso de los datos, por ejemplo, para realizar algunas tareas administrativas, de recuperación, etc.
- Una vez que se ha iniciado la instancia en alguno de los estados anteriores, se puede emplear el comando `alter database` para que la instancia pase de un estado al siguiente estado superior (no es posible saltar ni retroceder estados).



- Observar que el primer comando debe ser `startup nomount`. A partir de este punto, se puede emplear el comando `alter database` para pasar al siguiente estado.

1.4.3.1. Actividad 1

Ejecutar las siguientes instrucciones en el servidor linux que tiene instalado Oracle. **C1. Incluir en el reporte** únicamente el resultado de ejecutar el inciso "N".

- Abrir una terminal (Linux), autenticarse como `oracle` (empleando `su -l oracle`)
- Iniciar el listener empleando `lsnrctl start`
- Entrar a la consola de `sqlplus` empleando `sqlplus /nolog`, al entrar a la consola de `sqlplus`, el prompt de la terminal cambia a `SQL>`
- Autenticarse como `sysdba` para poder iniciar la instancia empleando `connect / as sysdba`
- Iniciar la instancia empleando `startup nomount`
- Montar la base de datos empleando `alter database mount`
- Abrir la base de datos empleando `alter database open`
- Detener la instancia de la base de datos empleando `shutdown immediate`
- Ejecutar `disconnect`. Este comando termina la sesión y regresa a la consola de `SQL *PLUS`.
- Ejecutar `exit` para salir de la consola de `SQL *Plus`
- Acceder nuevamente a `Sql *Plus`, pero ahora con la instrucción `sqlplus / as sysdba`
- Es posible abrir la base de datos en un solo paso escribiendo el comando `startup` (sin opción). Ejecutar el comando, verificar la salida.
- Ejecutar el comando `show user` el cual indica el usuario actual que está en sesión.
- Abrir otra terminal, sin usar el usuario `oracle` del sistema operativo, entrar a `Sql *Plus` como usuario `SYS` empleando el password proporcionado en prácticas anteriores. Se requiere especificarlo debido a que no se emplea el usuario `oracle` para autenticar. Aplicar el tip de seguridad explicado anteriormente para no mostrar el password en el comando.

- En caso de obtener un error de autenticación o de no recordar el password asignado, aplicar las siguientes acciones para actualizar el password del usuario SYS:

```
su -l oracle
sqlplus / as sysdba
SQL> alter user sys identified by <password>
```

Sustituir <password> por el valor que se desea actualizar. En prácticas anteriores se asignó el valor “system”. Intentar nuevamente la ejecución del inciso anterior.

O. Detener la instancia empleando el commando `shutdown immediate`

1.5. CREACIÓN DE USUARIOS.

Para poder realizar el acceso a la base de datos, es necesario crear un usuario. Durante el proceso de instalación se crean varios usuarios por default, todos ellos bloqueados a excepción de los 2 siguientes:

- SYS: Es el dueño del diccionario de datos y sus objetos asociados.
- SYSTEM: Usuario que tiene acceso a todos los objetos de la base de datos.

1.5.1. Usuarios y esquemas

- Cualquier usuario al que se le da permisos para crear objetos se asocia con un “esquema”. Un esquema contiene los objetos que son creados por un usuario. El esquema tiene el mismo nombre que el usuario.
- Cada uno de los objetos pertenecientes a un esquema puede ser referenciado empleando como prefijo el nombre de su esquema. Por ejemplo, si la tabla `estudiante` fue creada por el usuario `practicaDDL`, es posible hacer referencia a la tabla escribiendo `practicaddl.estudiante`. Esto es útil especialmente cuando el objeto es referenciado por otro usuario diferente al dueño de la tabla.
- Para entender la relación entre usuario y esquema, podemos hacer la siguiente analogía: Estudiante que vive en la casa de sus Papás. El dueño de la casa es el Papá, él tiene acceso a todos los objetos de la casa, y otorga permisos al hijo para hacer uso de ellos: la recamara, la cocina, la TV, etc. De esta forma el estudiante tendría el rol de un usuario valido, pero no es dueño o no tiene un esquema asociado, únicamente tiene permisos para hacer uso de los objetos de la casa (objetos pertenecientes al usuario dueño del esquema).
- Los usuarios que pueden crear nuevos usuarios son SYS y SYSTEM.

1.5.2. Los esquemas SYS y SYSTEM

- Todas las bases de datos Oracle incluyen la definición de los usuarios SYS y SYSTEM, y como consecuencia, sus correspondientes esquemas. Ambos usuarios son considerados como usuarios administradores. Su uso en la práctica debe estar restringido para DBAs autorizados para realizar tareas de naturaleza administrativa. Por ejemplo, detener o levantar la instancia, administración de usuarios, administración de recursos: memoria, discos, etc. Ambos usuarios y esquemas se crean al momento de crear una BD.
- En el esquema SYS se almacenan las tablas y vistas principales que integran al diccionario de datos. Estos objetos son considerados como críticos para el correcto funcionamiento de la BD. Estos objetos son administrados por el propio manejador, por lo que, ningún usuario debe acceder o modificarlos.
- En el esquema SYSTEM se almacenan tablas y vistas adicionales que muestran información administrativa, así como tablas y vistas internas empleadas por diversas herramientas y opciones que ofrece la BD.

Por lo anterior, se hace especial énfasis en no emplear estos 2 usuarios para crear nuevos objetos o para realizar prácticas o tareas que no tengan que ver con tareas administrativas.

1.5.3. Administración de usuarios en Oracle.

- La sintaxis empleada para crear un usuario en Oracle es la siguiente:

```
create user <username> identified
[by <password> | externally | globally as <external_name>]
[default tablespace <tablespace>]
[temporary tablespace <tablespace>]
[{quota <num> [k | m] on <tablespace_name> | unlimited on <tablespace_name>}...]
[profile <profile_name>]
[password expire]
[account [lock|unlock]
```

- `default tablespace` indica el nombre del tablespace asociado para almacenar los objetos del nuevo usuario.
- `account` se emplea para bloquear al usuario al momento de crearlo. Por default es UNLOCK, es decir, el usuario puede conectarse a la base de datos inmediatamente después de crearse.
- `password expire`. El password expira inmediatamente después que se crea al usuario, útil para que la primera o la siguiente vez que entre en sesión, el sistema solicite el cambio de password.
- `quota` indica la cantidad de espacio que puede gastar un usuario en el tablespace asignado.

Ejemplo:

```
create user myuser identified by myuser quota 100m on users;
```

- Esta sentencia crea un usuario `myuser` con password `myuser`. Por default, se asocia al tablespace `users`. El usuario tendrá como máximo 100MB de espacio disponible (la M que aparece en la sintaxis se refiere a MB).
- La sintaxis para modificar la definición de un usuario es:

```
alter user <username> identified
[by <password> | externally | globally as <external_name>]
[default tablespace <tablespace>]
[temporary tablespace <tablespace>]
[{quota <num> [k | m] on <tablespace_name> | unlimited on <tablespace_name>}...]
[profile <profile_name>]
[password expire]
[account [lock|unlock]
```

- Para eliminar a un usuario se emplea:

```
drop user <username> [cascade]
```

- Si el usuario contiene objetos en su esquema, Oracle genera un error, no es posible eliminar usuarios con objetos existentes, excepto que en la instrucción se especifique `cascade`.
- `cascade` elimina todos los objetos y datos existentes de un usuario.
- Para obtener los objetos que le pertenecen a un usuario se emplea la siguiente sentencia SQL:

```
select object_name, object_type from dba_objects where owner='<user_name>';
```

1.5.4. Otorgar y remover privilegios a un usuario

- La creación de un nuevo usuario permite que este sea autenticado en la base de datos. Sin embargo, no tiene autorización o permiso alguno para realizar acciones en la base de datos: crear sesión, crear objetos, etc. El siguiente paso posterior a la creación de un usuario es otorgar privilegios. En Oracle existen 2 tipos de privilegios: System privileges y Object privileges

1.5.4.1. System privileges

Permiten a un usuario realizar acciones que alteran el estado de la base de datos. La siguiente lista representa los privilegios comúnmente empleados.

```
create session
create table
create view
create synonym
create public synonym
create procedure
create trigger
create sequence
create user
alter user
drop any table
alter any table
backup any table
select any table
insert any table
update any table
delete any table
```

- Los privilegios a nivel sistema son otorgados por el DBA (usuario sys), mientras que los privilegios hacia los objetos son otorgados por el usuario dueño del objeto. La sintaxis empleada para otorgar o remover privilegios es:

```
grant <privilege>[,<privilege>,....]
to <username> [,<username>,...]
[with admin option];
```

- Como se puede observar, es posible asignar varios privilegios a la vez. No es necesario salir de sesión, los cambios toman efecto de forma inmediata.

Ejemplo:

```
grant create table,create sequence to myuser;
```

1.5.4.2. Object privileges

- Permiten a un usuario manipular objetos, leer datos, modificar su estructura, etc. Recordando, el dueño del objeto (tabla, vista o secuencia), tiene por definición todos los privilegios sobre el objeto. Sin embargo, estos privilegios pueden ser otorgados a usuarios que no sean dueños del objeto.

Ejemplo:

Suponer que la tabla EMPLEADO pertenece al usuario HR, y se desea otorgarle permisos de SELECT al usuario DEMO:

```
grant select on employees to demo;
```

- En este ejemplo, el DBA otorga privilegios al usuario MYUSER para poder crear tablas y secuencias. La opción WITH ADMIN OPTION otorga al usuario los privilegios o autoridad para que pueda otorgar esos mismos privilegios a otros usuarios.

1.5.5. Roles

- Como se puede observar, asignar privilegios a este nivel, puede representar una tarea repetitiva, en especial cuando el DBA desea crear varios usuarios con los mismos privilegios. En Oracle existe el concepto de Rol. Un rol agrupa a una serie de privilegios que pueden ser otorgados a varios usuarios. A su vez un usuario puede tener varios roles.
- Para crear un rol en Oracle, se emplea la siguiente sintaxis:

```
create role <role_name>;
```

Ejemplo:

```
create role manager; -- emplear el usuario sys
```

Para definir los privilegios que tendrá este rol, se emplea la siguiente sintaxis:

```
grant <privilege1, privilege2,..., privilegen> to <role_name>;
```

Ejemplo:

La siguiente instrucción asocia los privilegios para crear tablas, vistas al rol manager.

```
grant create table, create view to manager;
```

Para asignar un rol a un usuario se emplea la siguiente sintaxis:

```
grant <role_name> to <user>;
```

Ejemplo:

La siguiente instrucción asigna al usuario alice, el rol manager.

```
grant manager to alice;
```

1.6. SESIONES EN SQL *PLUS

- Una vez que se ha realizado la autenticación de un usuario se inicia una sesión. Cada usuario conectado a la instancia tiene su propia área de memoria asignada y sus datos particulares a su sesión.

Es importante seguir los siguientes lineamientos para trabajar de forma correcta y evitar un uso incorrecto de las sesiones del usuario:

- **No se recomienda** trabajar con el usuario `oracle` del sistema operativo ni con el usuario `SYS` de la BD. Dichos usuarios solo **deben** emplearse para hacer tareas administrativas como son: levantar o detener la instancia, crear usuarios, etc.
- Se deberá emplear una nueva terminal para interactuar con la base de datos en la que se hará uso de un usuario diferente al usuario `oracle`, y de un usuario diferente al usuario `SYS` de la base de datos. Mantener la terminal donde se está conectado como usuario `SYS` únicamente si se requiere realizar alguna otra tarea administrativa.

Lo anterior se describe con la siguiente secuencia de comandos:

A. **Terminal 1:** Empleada únicamente para operaciones administrativas:

1. Cambiarse al usuario Oracle: `su -l oracle`
2. Entrar a SQL *Plus empleando la autenticación del Sistema Operativo: `sqlplus / as sysdba`
3. Realizar las tareas de administración, por ejemplo, levantar la instancia.

B. **Terminal 2:** Empleada para trabajar con esquemas y usuarios de la BD que no implican actividades de administración. Por ejemplo: Resolver los ejercicios de las prácticas.

1. Abrir una terminal. El usuario del sistema operativo que se usará es un usuario ordinario (no usar al usuario `oracle`). Por ejemplo, el usuario `jorge`.
2. Una vez que la instancia ha sido levantada empleando la Terminal 1, entrar a SQL *Plus con un usuario ordinario de la BD, (no con el usuario `SYS`): `sqlplus <bd_user>/<bd_user_password>`
3. Realizar las tareas solicitadas con el usuario `<bd_user>`

1.6.1.1. Actividad 2.

Ejecutar las siguientes instrucciones para ilustrar los puntos anteriores.

- Se creará un usuario llamado `<iniciales>_p07_previo`.
- Sustituir `<iniciales>` con las 3 primeras letras del nombre y apellidos del alumno. Ejemplo: `jrc_p07_previo`.
- En caso de que sea posible realizar la práctica en equipo, cada integrante deberá crear este usuario en su ambiente.

Terminal 1:

```
jorge@lap-red-mint ~ $ su -l oracle
Contraseña:
oracle@lap-red-mint ~ $ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Thu Nov 2 19:14:00 2017

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 1073741824 bytes
Fixed Size      2932632 bytes
Variable Size   750780520 bytes
Database Buffers 314572800 bytes
Redo Buffers    5455872 bytes
Database mounted.
Database opened.
SQL> show user
USER is "SYS"
```

Como se puede observar, en la terminal 1 se hace uso del usuario `oracle` (sistema operativo) para levantar la instancia. A nivel de la base de datos el usuario que está en sesión es `SYS`.

Las siguientes instrucciones se ejecutan en la terminal 1 ya que se va a crear un usuario. Recordando, la creación de usuarios se realiza empleando el usuario `SYS`.


```
SQL> create user jrc_p07_previo identified by jorge
      2 quota unlimited on users;
```

User created.

```
SQL> grant create table, create session to jrc_p07_previo;
```

Grant succeeded.

```
SQL>
```

En este ejemplo se ha creado al usuario `jrc_p07_previo` con privilegios para crear sesiones y crear tablas.

Una vez que se ha creado al usuario, todos los objetos y actividades que realizará deberán aplicarse en la Terminal 2 sin emplear a los usuarios oracle y SYS:

Terminal 2:

```
jorge@lap-red-mint ~ $ whoami
```

```
jorge
```

```
jorge@lap-red-mint ~ $ sqlplus jrc_p07_previo
```

```
password *****
```

```
SQL*Plus: Release 12.1.0.2.0 Production on Thu Nov 2 19:23:16 2017
```

```
Copyright (c) 1982, 2014, Oracle. All rights reserved.
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
```

```
SQL> show user
```

```
USER is "JRC_P7_PREVIO"
```

```
SQL>
```

- Observar que a nivel de sistema operativo se emplea el usuario `jorge` (no oracle).
- Observar que el usuario a nivel de la BD es el usuario `jrc_p07_previo`.
- Esta terminal representa el lugar correcto para realizar tareas propias del usuario creado. **C2. Incluir en el reporte** la salida de la terminal 2 empleando el usuario creado. Si la práctica se realiza en equipo, **incluir una salida por cada integrante**.

1.7. PRINCIPALES CARACTERÍSTICAS DE SQL *PLUS

- Las sentencias SQL a ejecutar en SQL *PLUS pueden expandirse en varias líneas. Por cada línea aparecerá un número a la izquierda de la consola indicando que la sentencia puede continuar.
- Se emplea “;” para terminar una sentencia SQL, al presionar “enter”, se invoca su ejecución.
- Para comandos que son propios de SQL *Plus no requieren “;”.
- SQL *PLUS cuenta con un **buffer** el cual contiene la última secuencia SQL ejecutada. Este buffer puede ser editado o almacenado en un archivo.
- Escribiendo “/” en una nueva línea, o el comando `run` provoca la ejecución de la última instrucción almacenada en el buffer.
- Solo sentencias SQL y bloques PL/SQL son almacenadas en el buffer. Recordar que en SQL *PLUS se puede ejecutar tanto sentencias SQL como comandos propios de la herramienta: `connect`, `disconnect`, `show user`, etc.

1.7.1. Configurando SQL *Plus - Historial de comandos.

La configuración inicial de la interfaz de SQL *Plus es muy limitada. No se cuenta con historial de comandos capturados previamente, las flechas de navegación del cursor `<-` `^` `->` no permiten corregir comandos. Los siguientes puntos permiten configurar la interfaz para mejorar la experiencia de uso.

1.7.1.1. Uso del buffer.

Para ilustrar su uso, ejecutar las siguientes instrucciones SQL capturada en SQL *PLUS haciendo uso de la terminal 2 con el usuario creado anteriormente. Observar la salida generada.

```
SQL> select count(*)
      2 form user_tables;
form user_tables
      *
```

```
ERROR at line 2:
```

```
ORA-00923: palabra clave FROM no encontrada donde se esperaba
```

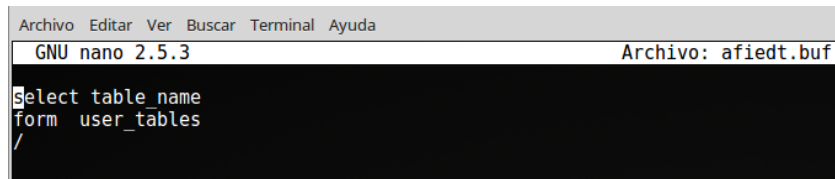
- Como se puede observar en la sentencia anterior, se intenta mostrar el número de tablas que le pertenecen al usuario SYS. La palabra `from` está mal escrita.
- Por las limitantes del editor, no es posible corregir los errores sin tener que re- escribir la sentencia.
- Una de las estrategias más sencillas para corregir es emplear un editor externo a SQL *Plus, por lo regular un editor de texto del sistema operativo: `notepad` en Windows, `nano`, `vi` o `pico` para el caso de UNIX. Para habilitar el uso del editor dentro de SQL *Plus se debe indicar el editor a emplear estableciendo el valor de una variable llamada `_EDITOR`

```
SQL> define _EDITOR=nano
SQL>
```

Internamente, SQL *Plus realiza la invocación del comando configurado. Para modificar el contenido del buffer empleando en este caso el editor `nano`, escribir el siguiente comando:

```
SQL> edit
```

Observar que, al ejecutar el comando, se abre el editor configurado (en este caso `nano`) con el contenido de la sentencia SQL. La corrección de los errores es muy sencilla ya que el editor nos permite modificar libremente el contenido del buffer, como cualquier archivo de texto.



- Para volver a ejecutar la sentencia, se debe corregir el error, guardar los cambios (para el caso del editor `nano` o `pico`, los cambios se salvan presionando `ctrl + o`).
- El siguiente paso es cerrar el editor empleando `ctrl + x`
- Al cerrar el editor, observar que aparece nuevamente la consola de SQL *Plus.
- Para ejecutar nuevamente el comando, simplemente escribir el comando `run`, o emplear `"/`.

Adicional al comando `run`, existe el comando `list` o `L` en su forma corta, que permite escribir nuevamente la última sentencia sin ejecutarla.

```
SQL> edit
Wrote file afiedt.buf

 1 select count(*)
 2* from user_tables
SYS-SQL> run
 1 select count(*)
 2* from user_tables

COUNT(*)
-----
0
```

- Se vuelve a mostrar la sentencia. Observar que aparece un `"*"` en el contenido del buffer, el cual indica la línea actual.

1.7.1.2. Site and user profile

Para no tener que establecer el valor de la variable `_EDITOR` cada vez que se acceda a SQL *Plus es posible hacer uso de una funcionalidad llamada "Site and User profiles". Esta funcionalidad permite ejecutar comandos de forma automática justo después de ejecutar el comando `sqlplus`. Lo anterior permite personalizar la configuración y evitar la ejecución manual cada vez que se acceda a SQL *Plus. Existen 2 archivos donde se pueden agregar estos comandos.

- A nivel global para todos los usuarios (Site Profile):
 - Existe un archivo llamado `glogin.sql` ubicado en el directorio `$ORACLE_HOME/sqlplus/admin/` (no olvidar que `$ORACLE_HOME` es una variable y su valor se emplea para determinar la ruta completa, por ejemplo: `/u01/app/oracle/product/12.1.0/dbhome_1`).
- A nivel local para un solo usuario (User profile)
 - En este caso el archivo no existe, se debe crear un archivo de texto llamado `login.sql` en el directorio donde se va a invocar a SQL *Plus.

1.7.1.3. Actividad 3:

- Agregar las siguientes líneas al archivo `glogin.sql` (emplear el usuario Oracle para editar).

```
define _editor=nano
set sqlprompt "_user-SQL> "
set trimspool on
```

- Observar que en la segunda instrucción se modifica el “prompt” para agregar el nombre del usuario que está en sesión.
- La tercera instrucción elimina los espacios al final del archivo “spool”. Más adelante se explica la utilidad de este archivo.

1.7.1.4. Rlwrap

RIWrap Es una herramienta bastante útil que permite habilitar el historial de comandos dentro de SQL* Plus empleando las flechas de dirección tal cual se hace en una terminal del sistema operativo.

1.7.1.5. Actividad 4:

- Instalar el paquete

```
sudo apt-get install rlwrap (En Distribuciones Ubuntu/Mint/Debian)
sudo dnf install rlwrap (En Distribuciones Fedora)
sudo dnf install rlwrap (En Distribuciones Oracle Linux)
```

- Para hacer uso de rlwrap se deberá escribir el comando `rlwrap sqlplus`.
- Se recomienda crear un “alias” para evitar hacer esto cada vez que se acceda a SQL *Plus.
 - Agregar la siguiente línea en el archivo `/etc/bash.bashrc` (En Distribuciones Ubuntu/Mint/Debian)
 - Agregar la siguiente línea en el archivo `/etc/bashrc` (En Distribuciones Fedora/Oracle Linux)

```
#alias globales
alias sqlplus='rlwrap sqlplus'
```

- Cerrar sesión y volver a entrar para que los cambios tomen efecto.
- Abrir una nueva y entrar a SQL *Plus de forma normal. Verificar el uso del historial de comandos haciendo uso de las flechas de dirección.

1.7.1.6. Formato de columnas.

- En algunas ocasiones el resultado de una consulta en SQL *Plus suele presentar un formato difícil de leer. En el siguiente ejemplo se muestra el nombre del usuario, la fecha de creación y la fecha de expiración de un usuario:

```
select username, to_char(created, 'DD/MM/YYYY HH:MI:SS') fecha_creacion,
to_char(expiry_date, 'DD/MM/YYYY HH:MI:SS') fecha_expiracion
from user_users;
```

La salida de la consulta se ve así:

```
USERNAME
-----
FECHA_CREACION    FECHA_EXPIRACION
-----
EJEMPLOS
13/10/2015 08:40:05 10/04/2016 08:40:05
```

- Lo anterior se debe a que SQL *Plus reserva la longitud máxima de una columna para mostrarla en pantalla. En este caso el campo `username` está definido como un `varchar2(128)`. Por esta razón aparecen demasiados espacios y guiones para el campo `username`.
- Para corregir lo anterior, es posible limitar el número de caracteres máximo empleados para mostrar una columna empleando el comando `column`

```
col[umn] <column_name> format A<size>
```

Ejemplo:

```
sql> column username format A30
```

Al ejecutar nuevamente la sentencia, se obtiene una salida mucho mejor:

```
USUARIO            FECHA_CREACION    FECHA_EXPIRACION
-----
EJEMPLOS           13/10/2015 08:40:05 10/04/2016 08:40:05
```

- Otros comandos que pueden mejorar la apariencia de los resultados, en especial si se requiere un mayor tamaño del renglón o del número de renglones a mostrar (tamaño de página) son:

```
set pagesize <valor>
set linesize <valor>
```

1.7.1.7. Spool

- Spool es una herramienta que permite exportar o almacenar las instrucciones y salida de comandos que se ejecutan en SQL *Plus.
- La herramienta es muy útil ya que permite guardar en un archivo absolutamente todos los comandos y resultados que se obtienen al interactuar con SQL *Plus y evita la necesidad de anotar o respaldar manualmente todos los comandos aprendidos.

Sintaxis para hacer uso del Spool:

```
spo[ol] [file_name[.ext]] [cre[ate] | rep[lace] | app[end]] | off | out]
```

- Se recomienda habilitar el spool una vez que se ha levantado la instancia en la **Terminal 2**.

Ejemplo 1:

```
sql> spool /home/jorge/mis-comandos.txt create
```

- Este comando habilita el spool, el contenido lo guardará en el archivo /home/jorge/mis-comandos.txt
- Si el archivo existe, se generará error.
- Para evitar problemas de permisos de creación y escritura de archivos, el usuario a nivel de sistema operativo que se está empleando en la terminal donde se ejecuta SQL *Plus debe tener permisos para escribir y crear archivos en la ruta configurada.

Ejemplo:

```
jorge@lap-red-mint ~ $ sqlplus jrc_p07_previo
password: *****

SQL*Plus: Release 12.1.0.2.0 Production on Thu Nov 2 19:55:27 2017

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Jue Nov 02 2017 19:54:10 -06:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> spool /home/jorge/spool.txt create
SQL>
```

- Observar que el usuario a nivel de sistema operativo es el usuario `jorge` (marcado en rojo).
- En la BD se emplea el usuario `jrc_p07_previo`.
- EL Spool se va a guardar en `/home/jorge/spool.txt`. Esto no generará problemas de permisos ya que el usuario `/home/jorge` le pertenece al usuario `jorge`.

Ejemplo 2:

```
sql> spool /home/jorge/mis-comandos.txt append
```

- Este ejemplo es útil cuando se desea continuar agregando contenido a un archivo existente.

Ejemplo 3:

```
sql> spool /home/jorge/mis-comandos.txt off
```

- Este ejemplo detiene el spool y escribe el contenido al archivo.
- Notar que en el archivo no se realiza escritura hasta que se ejecuta este comando, o al salir de SQL *Plus con el comando `exit`. (No cerrar la terminal abruptamente).
- Es posible volver a habilitar el spool posterior a la ejecución del comando `off`.

Tip: El spool no se detiene si se cambia de usuario empleando el comando `connect`. El spool se conserva mientras no se cierre la terminal o se ejecute el comando `exit`.

1.7.2. Ejecutando Scripts en SQL *Plus

- Esta funcionalidad permite ejecutar archivos SQL, típicamente generados a partir de una herramienta de modelado.
- Para ejecutar un archivo con sentencias SQL, se emplea `@<file_name>` o el comando `start <file_name>`.
- En algunos casos es posible invocar la ejecución de un archivo SQL dentro de otro. Para ello se emplea `@<file_name>`.
- Si no se especifica una ruta junto con el nombre del archivo, SQL *PLUS realiza la búsqueda en el mismo directorio donde se encuentra el archivo que lo invoca.
- En su defecto, se puede especificar una ruta relativa o una ruta absoluta. Ejemplos:

```
@@/tmp/mis_scripts/script1.sql  (ruta absoluta)
@@mis_scripts/script1.sql      (ruta relativa)
@@../mis_scripts/script1.sql    (ruta relativa)
```

Importante: Scripts SQL **NO** deben guardarse dentro de los directorios donde está instalada la BD, tampoco se debe hacer uso del usuario oracle para manipularlos.

1.7.2.1. Actividad 5.

Considerar la existencia de los siguientes scripts SQL:

- s-00-asignaturas-elimina-usuario.sql
- s-01-asignaturas-crea-usuario.sql
- s-02-asignaturas-ddl.sql
- s-03-asignaturas-carga-inicial.sql
- s-04-asignaturas-main.sql
- El contenido del archivo s-00-asignaturas-elimina-usuario.sql se muestra a continuación.

```
--@Autor:          Jorge Rodriguez Campos
--@Fecha creación: dd/mm/yyyy
--@Descripción:     Elimina al usuario jrc_p07_previo en caso de existir.
set serveroutput on
declare
  v_count number(1,0);
begin
  select count(*) into v_count
  from dba_users
  where username ='JRC_P07_PREVIO';

  if v_count > 0 then
    execute immediate 'drop user JRC_P07_PREVIO cascade';
    dbms_output.put_line('Usuario eliminado');
  else
    dbms_output.put_line('El usuario no existe, no se requiere eliminar');
  end if;
end;
/
```

- En general este script contiene un programa que elimina al usuario en caso de existir.
- El contenido del archivo s-01-asignaturas-crea-usuario.sql se muestra a continuación.

```
--@Autor:          Jorge Rodriguez Campos
--@Fecha creación: dd/mm/yyyy
--@Descripción:     Creación de Usuarios

create user jrc_p07_previo identified by jorge quota unlimited on users;
grant create table, create session to jrc_p07_previo;
```

- El contenido del archivo s-02-asignaturas-ddl.sql se muestra a continuación. Contiene la definición de 2 tablas:

```
--@Autor:           Jorge Rodriguez Campos
--@Fecha creación:  dd/mm/yyyy
--@Descripción:     Creación de objetos

--Tabla plan_estudios
create table plan_estudios(
  plan_estudios_id  number(3,2) constraint plan_estudios_pk primary key,
  clave             varchar2(10) not null,
  fecha_aprobacion  date not null,
  fecha_inicio      date not null,
  fecha_fin         date
);

--Tabla asignatura
create table asignatura(
  asignatura_id      number(10, 0)    not null,
  nombre             varchar2(50)     not null,
  creditos           number(2, 0)     not null,
  asignatura_requerida_id  number(10, 0),
  plan_estudios_id   number(2, 0)     not null,
  constraint asignatura_pk primary key (asignatura_id),
  constraint asignatura_plan_fk foreign key (plan_estudios_id)
  references plan_estudios(plan_estudios_id),
  constraint asignatura_req_fk foreign key (asignatura_requerida_id)
  references asignatura(asignatura_id)
);
```

- El contenido del archivo s-03-asignaturas-carga-inicial.sql se muestra a continuación. Contiene la inserción de registros en ambas tablas:

```
--@Autor:           Jorge Rodriguez Campos
--@Fecha creación:  dd/mm/yyyy
--@Descripción:     Script de carga inicial

prompt realizando la carga de planes de estudios y asignaturas.

insert into plan_estudios(plan_estudios_id,clave,fecha_aprobacion,fecha_inicio,fecha_fin)
values (1,'PL-001',sysdate,sysdate,sysdate+365);

insert into asignatura(asignatura_id,nombre,creditos,plan_estudios_id)
values (1,'algebra',10,1);

insert into asignatura(asignatura_id,nombre,creditos,asignatura_requerida_id,plan_estudios_id)
values (2,'algebra lineal',10,1,1);
commit;
```

- Para ejecutar los scripts antes mencionados se creará uno nuevo llamado s-04-asignaturas-main.sql
- El script deberá invocar a los 4 archivos. El Script deberá conectarse a la instancia empleando los usuarios correspondientes para ejecutar las instrucciones de los scripts.
- En general la estructura del archivo es la siguiente:

```
--@Autor:           Jorge Rodriguez Campos
--@Fecha creación:  dd/mm/yyyy
--@Descripción:     Archivo principal
whenever sqlerror exit rollback;

prompt conectando como sys para eliminar/crear al usuario
connect sys as sysdba

prompt eliminando al usuario jrc_p07_previo en caso de existir
-- completar

prompt creando usuario jrc_p07_previo
--completar

prompt conectando como usuario jrc_p07_previo
--completar con el comando connect

prompt creando tablas
--completar

prompt cargando datos
--completar

prompt Listo!
```

- A. Crear los 5 scripts SQL empleando los nombres de archivos indicados. Guardar todos los scripts en algún directorio, por ejemplo: `/home/jorge/BD/practicas/p07/previo`.
- B. Revisar el código de cada script y sustituir el nombre del usuario con las iniciales de cada alumno
- C. Completar el archivo `s-04-asignaturas-main.sql` y ejecutarlo. **C3. Incluir en el reporte** el contenido y el resultado de su ejecución (únicamente de este script).
- D. En caso de ser una práctica en equipo, se deberá incluir el contenido y salida de cada integrante.
- E. No Olvidar las convenciones vistas en el documento: Los scripts no deben ubicarse dentro de los directorios de instalación de la BD, no deben pertenecer al usuario oracle ni ubicarse dentro de `/home/oracle`. Usar la 'Terminal 2' para ejecutar.

1.8. CONTINUAR CON EL DESARROLLO DE LA PRÁCTICA.