

UD1. Sistemas Gestores de Bases de Datos

Índice

Implantación de SGBD.....	2
Instalación, creación y configuración de bases de datos.....	2
Fases en la implantación de una base de datos.....	3
Proceso de instalación.....	5
Por que Oracle?.....	5
Conexión a la base de datos.....	7
Conexión usando SQL*Plus.....	9
De que está compuesta una base de datos de Oracle?.....	12
Que es una pluggable database?.....	14
Consulta el catálogo de Oracle: Tablespace y datafiles.....	15
Consulta el catálogo de Oracle: parámetros de instancia.....	16
Instalando el esquema del usuario de ejemplo HR.....	17
comandos del SQL*PLUS.....	18
Comando DESC[RIBE].....	18
Comando START o @.....	18
Comando HISTORY.....	19
Comando COLUMN.....	19
Comando SHOW.....	20
Comando SPOOL.....	20
Comando DEFINE.....	21
Variables de sustitución.....	21
Paso de valores a un script.....	21
Otros comandos.....	22
El diccionario de datos de Oracle.....	23
Vistas estáticas.....	23
Vistas dinámicas de rendimiento.....	25

Implantación de SGBD

Las bases de datos forman parte de cualquier sistema de información. Junto a estas se encuentran todas las aplicaciones que permiten su gestión y su explotación, además de aquellas que sirven a la mecanización de los procesos de la organización propietaria de ellas. Estas aplicaciones requieren de accesos eficientes al lugar donde se encuentra el gigantesco repositorio de datos. Las tecnologías de mantenimiento usadas son diversas y las arquitecturas de almacenamiento varían desde las locales, centralizadas y distribuidas, hasta las alojadas en la nube.

Los sistemas gestores de bases de datos son las herramientas que permiten la gestión completa de las bases de datos y toda su explotación. Atendiendo a cuestiones técnicas de la naturaleza de los datos, de los sistemas para su obtención y el modo en el que estos se pueden organizar, existen diversas arquitecturas. Las arquitecturas actuales son las relacionales, las orientadas a la documentación y los Big Data, y, por tanto, la oferta de sistemas gestores de bases de datos es amplia, desde Oracle, MySQL y PostgreSQL, hasta, por ejemplo, MongoDB.

En esta unidad se estudian las consideraciones que tener en cuenta en el momento de implantar un sistema gestor de bases de datos, sus componentes, características fundamentales, su elección y verificación y la documentación del proceso completo de instalación. Se analizan los ficheros que componen una base de datos y la gestión del almacenamiento en soportes. Se profundiza en la arquitectura del diccionario de datos de Oracle para su uso en cualquier acción que acometer en la consola de comandos.

Instalación, creación y configuración de bases de datos

Los sistemas gestores de bases de datos se convierten en la herramienta informática que permite definir, crear y dar mantenimiento a las bases de datos, y luego su administración, con el fin de proporcionar acceso controlado, gestión eficiente del espacio de almacenamiento y explotación de los recursos, entre otros objetivos principales. Esta gestión completa actúa como interfaz intermedia entre las aplicaciones que requieren, crean y usan los datos y el almacén masivo de datos.

Desde este punto de vista, la base de datos es la colección de todos los datos creados y manipulados por los usuarios a través de las aplicaciones, y las aplicaciones son los agentes que usan el sistema gestor de bases de datos. El propósito de cada aplicación puede variar, desde permitir el alta, la baja y la modificación de los datos almacenados en repositorios, hasta la administración del sistema gestor. Oracle define el concepto de **base de datos** como la suma de una **instancia** y un conjunto de **ficheros** de la base de datos. Alrededor de la base de datos, que se puede definir como el núcleo del sistema, se encuentran desde dentro hacia fuera el sistema operativo, el sistema gestor de bases de datos, la red y el conjunto de aplicaciones.

Existen varias opciones para la configuración del software y del hardware con relación al servidor del sistema gestor Oracle. La opción **cliente-servidor** implica la existencia de una máquina central encargada de controlar todas las peticiones de los clientes. A esta máquina se le conoce como servidor central y provee un servicio a toda la intranet. Este servicio está ejecutándose constantemente en esa máquina con el fin de detectar cualquier petición de un cliente Oracle. Cuando esto ocurre se crea un canal de comunicación entre el cliente y el servidor por el que van las peticiones. El servidor es capaz de gestionar un gran número de canales de comunicación con los clientes.

Otra opción es la **distribuida**. Esta opción permite operaciones que afectan a varias bases de datos ubicadas en lugares distintos de toda la intranet y usa una serie de componentes para la gestión transaccional. Oracle también puede gestionar réplicas o snapshots de tablas enteras en diferentes bases de datos, donde el modelo de actualización se convierte en un modelo centralizado.

La opción avanzada de **replicación** permite diseñar aplicaciones que gestionan bases de datos distribuidas debido a que existe una réplica exacta de un conjunto de objetos distribuidos por los diferentes nodos de toda la arquitectura. Así, se puede hacer cualquier operación sobre el objetivo replicado.

También existe la opción paralela de consultas, que descompone una operación en varios procesadores para acelerar la ejecución. La opción de servidor paralelo ofrece alta escalabilidad en bases de datos de otra naturaleza, como los llamados OLTP, sistemas orientados a la decisión y los almacenes de datos. Esta opción se base en la concurrencia de varios nodos hacia la base de datos. En este contexto entran en juego los sistemas clusters, los servidores express OLAP, los servidores web, etcétera.

Fases en la implantación de una base de datos



Estas fases principales son las siguientes:

Fase de implantación:

Instalar y actualizar el sistema gestor de bases de datos, tomando decisiones sobre el mejor producto para la organización, instalando componentes para la seguridad y actualizando en todo momento cualquier software de gestión.

Diseñar las estructuras de la base de datos, los **ficheros** permanentes donde se almacenan, los ficheros de actualización que permiten procesos de recuperación y los ficheros de control que gestionan todos los aspectos claves de la base de datos.

Diseñar, construir y mantener los **ficheros de parámetros** gestionando en todo momento los valores más adecuados para la correcta explotación de la base de datos.

Crear, mantener y gestionar las bases de datos, así como las **instancias** necesarias para cada usuario que requiera explotarlas. La gestión de bases de datos conectables en entornos multicontenedores o **multitenant** también es un objetivo muy importante.

Controlar los accesos de los usuarios a la parte de la base de datos que estos puedan gestionar. La **gestión de usuarios** es una tarea muy importante.

Fase de mantenimiento

Controlar la **evolución de los datos en las tablas y en los ficheros físicos** donde se almacenan. Una correcta gestión del almacenamiento influirá en una eficiente explotación de los medios de almacenamiento masivos con su ahorro en costes.

Monitorizar la explotación de la base de datos por parte de los usuarios y las aplicaciones, detectando en todo momento posibles procesos no eficientes o con propósitos malintencionados. Gestionar las **auditorías** sobre el uso de la base de datos.

Asegurar la arquitectura física de la base de datos a entornos web en los que se posibilita la explotación externa de usuarios, a través de cortafuegos, servidores proxy y configuraciones **DMZ**.

Tener en cuenta en todo momento las políticas de **seguridad** actuales de la organización, y saber detectar aquellas que potencialmente pueden no ser útiles o mejorables y trabajar con los técnicos de seguridad al respecto.

Analizar el **rendimiento** de la base de datos y ajustarla para obtener la mayor eficiencia posible.

Establecer políticas claras de **copias de seguridad** de los datos y alta disponibilidad si es posible.

Detectar y mantener los sistemas de **balanceo de carga** y réplica.

Proceso de instalación



Una vez evaluado el hardware del servidor procedemos a la instalación del software que será el gestor de base de datos **Oracle Database** para esto decidiremos el número de ficheros de datos que requerimos (datafiles), instancia (procesos y tamaño de memoria que se dedicará) ficheros de control, ficheros de log (redo logs) ficheros de log archivados (arch logs) y ficheros de traza (log de transacciones, auditorias, alertas...)

Por que Oracle?

Oracle es un SGBD relacional, multiplataforma, confiable y seguro. Su arquitectura principal es cliente/servidor y está orientándose a otros tipos de modelos. Usa también el lenguaje JSON. Ofrece una versión gratuita Express Edition. Es un producto muy competitivo.

Oracle se define como gestor de bases de datos objeto-relacional, por lo que puede manejar los datos a través de tablas y a través de objetos. Para esto, en sus versiones últimas, se ha incluido el manejo del lenguaje JSON para las estructuras orientadas a objetos.

Oracle se puede instalar en cualquier sistema operativo. Su versión completa se denomina **Enterprise Edition** y también dispone de otras versiones más reducidas, pero que se adaptan a la mayoría de las organizaciones. Dispone de una **versión gratuita de prueba y de uso académico denominada Express Edition**. También en la web de oracle podemos encontrar máquinas virtuales .ova de Oracle Linux con El Oracle Database Enterprise Edition junto a herramientas de aprendizaje son las **Pre-Built Developer VMs** (para VirtualBox).

ORACLE XE

- Se puede usar como se quiera: aprendizaje, desarrollo, producción
- limitada a 5GB de datos

ORACLE ENTERPRISE EDITION

- Se baja de la web de oracle. Se puede usar pero sin licencia no tiene soporte
- Funcionalidad completa
- Versiones principales:
Versiones muy antiguas : 11g (sin multitenant), 12c (multitenant)
19c (LTS, la más usada), 21c
23 c (última versión, cloud)

PRE-BUILT DEVELOPER VMs

- Vienen en una ova con Oracle Linux listos para usar
 - Se usan para aprendizaje y desarrollo
 - Está prohibido poner las máquinas en producción
- Oracle WebCenter Portal VM (desarrollo de aplicaciones Java)
Oracle Big Data Lite VM
Database App Development VM (aprendizaje de BBDD de proposito general)

PRÁCTICA

INSTALACIÓN DE ORACLE DATABASE EXPRESS EDITION

Pasos:

1 . Baja la última versión

Selecciona [la última versión de Oracle Express Edition](#) y procede a su instalación. Es recomendable usar una máquina virtual con al menos 2 GB de RAM e 11GB de disco duro libre. Fíjate que se puede instalar en un Oracle Linux (tendríamos que crear una máquina virtual con el) o en Windows. Vamos a instalarlo en una máquina virtual con Windows 10.

2. Para instalarlo seguiremos [este documento de Oracle](#).

Para la mayor parte, sigue las instrucciones del documento anterior. Salvo:

En lugar de doble-click en el fichero setup.exe file, haz click con el botón derecho y selecciona **Run as Administrator**.

Acepta todo por defecto.

Cuando te salga el mensaje del bloqueo firewall de windows defender permite acceso.

3. Ten muy en cuenta el paso correspondiente a la introducción de la contraseña que usarán las cuentas de usuario SYS, SYSTEM y PDBADMIN. Esta contraseña sera la que se emplee para conectarse a la base de datos por primera vez a través de la cuenta SYS AS SYSDBA.

4. Ten paciencia, le lleva mucho tiempo acabar. Apunta en un documento la info que aparece: contenedor, conexión y url del Enterprise Manager.

Conexión a la base de datos

La conexión a la base de datos conlleva establecer un canal de comunicación con el servidor con el fin de realizar operaciones sobre los datos. Para ello, se requiere un programa que permita la comunicación y el envío de sentencias y comandos SQL.

Los comandos enviados al servidor son interpretados por el procesador del lenguaje o parser y luego por el motor de la base de datos. Con el fin de administrar la base de datos, estos comandos se pueden enviar directamente a través de la herramienta SQL *Plus o usando una herramienta gráfica como Oracle Enterprise Manager Database Express u Oracle Enterprise Manager Cloud Control. También es habitual usar SQL Developer, más usada por los desarrolladores para crear y probar diferentes esquemas y aplicaciones.

PRÁCTICA

INSTALACIÓN DE ORACLE SQL Developer

SQL Developer es una herramienta como TOAD que permite conectarte a una base de datos y hacer tareas de administración o de desarrollo, tanto en SQL como en PL/SQL
Pasos:

1. Mira que versión tienes de JAVA. Si no tienes java SE Development Kit 8 (Java Version 1.8.x), 11, o 17 instalado (por ejemplo porque estamos usando una máquina virtual), si es así podemos elegir la versión con **JDK 8 incluido**

Para ver tu versión de java, abre cmd y teclea: java -version

Si tienes JDK 8, 11, o 17 instalado, elige el **instalador que requiere el JDK**.

2. Baja el [programa de esta página](#). Recuerda bajar el instalador con JDK o sin JDK incluido dependiendo de si tienes el JAVA instalado o no. Tendrás que tener un usuario en la web de Oracle. Si no estás dado/a de alta, date.

2. Sigue las [instrucciones de esta página](#). Te recomiendo que extraigas el sqldeveloper al directorio C:\Program Files\Oracle\ o C:

Al pulsar en sqldeveloper.exe dependiendo de donde descomprimiéramos el programa, tendremos acceso al sqldeveloper.

Vamos a crear varias conexiones a distintos usuarios. Por ejemplo una conexión con el usuario system al contenedor (SID XE) y a la pluggable database (nombre del servicio XEPDB1) según donde nos conectemos usaremos uno u otro.

También podemos crear una conexión con el usuario SYS as sysdba



Pulsamos el signo + para crear conexiones

Nueva / Seleccionar Conexión a Base de Datos

Nombre de Cone... Detalles de Cone...

Nombre de la conexión

Usuario y contraseña

Cadena de conexión
SID si me conecto al
container
Service name si me
conecto a la pluggable

Probar antes de
conectar

Nombre system xe

Tipo de Base de Datos Oracle

Información de usuario Usuario de Proxy

Tipo de autenticación Por defecto

Usuario system Rol valor por defecto

Contraseña

Guardar Contraseña

Tipo de Conexión Básico

Detalles Avanzado

Nombre del Host localhost

Puerto 1521

☒ SID xe

☐ Nombre del Servicio xepdb1

Estado: Correcto

Ayuda Guardar Borrar Probar Conectar Cancelar

Otra conexión de
system pero a la
pluggable database

Nombre del Host localhost

Puerto 1521

☐ SID

☒ Nombre del Servicio xepdb1

Nueva / Seleccionar Conexión a Base de Datos

Nombre de Cone... Detalles de Cone...

system pdb1 system@//localh...

system xe system@//localh...

Conexión de SYS tiene
que ser con el rol
SYSDBA

Nombre sys

Tipo de Base de Datos Oracle

Información de usuario Usuario de Proxy

Tipo de autenticación Por defecto

Usuario sys Rol SYSDBA

Contraseña

Guardar Contraseña

Tipo de Conexión Básico

Detalles Avanzado

Nombre del Host localhost

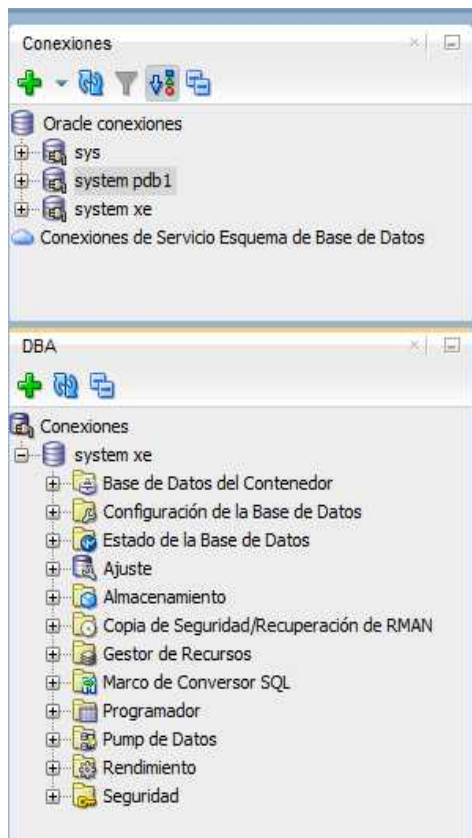
Puerto 1521

☒ SID xe

☐ Nombre del Servicio

Estado: Correcto

Ayuda Guardar Borrar Probar Conectar Cancelar



También podemos ver la "vista DBA" que nos permite acceder a multitud de tareas de administración fácilmente.

Para habilitarla tenemos que ir al menú

VER-> DBA.

Aparecerá un recuadro nuevo (podemos cerrar el de informes) en el panel izquierdo y pulsando el signo + escogeremos la conexión con la que queremos trabajar (por ejemplo system)

La conexión a la base de datos puede ser local, en la misma máquina donde está el servidor Oracle, o remota. La máquina que se quiere conectar a la base de datos debe tener instalado el producto Oracle Client o una instalación de Instant Client. La versión completa de Oracle Database dispone de las herramientas para el cliente y para el servidor.

Conexión usando SQL*Plus

La conexión a Oracle permite la conexión a una instancia. En el servidor debe existir como mínimo una instancia. En arquitecturas más avanzadas es habitual tener habilitadas y configuradas más de una instancia. Una instancia permite tantas conexiones como la máquina pueda.

Lo primero que debe hacerse para usar SQL Plus es tener abierta la consola de comandos del sistema operativo. Desde procesos batch se pueden ejecutar conexiones a la base de datos, salir de la conexión y seguir con el proceso batch del sistema operativo. Es posible que sea necesario configurar algunas variables de entorno del sistema operativo. La variable de entorno en Windows llamada ORACLE_HOME contiene el directorio donde se ha instalado Oracle. Además, la variable ORACLE_SID contiene el identificador único de la base de datos. Estas variables se pueden establecer escribiendo, por ejemplo, SET ORACLE_SID=XE. Otra variable de interés es PATH que debe contener el directorio bin que cuelga de ORACLE_HOME, es decir, PATH=ORACLE_HOME/bin.

Para ejecutar sqlplus basta con:

a) Escribir en la consola de comandos el texto **sqlplus** y poner usuario y password cuando nos lo pida

b) Escribir en la consola **sqlplus** y la cadena de conexión

sqlplus usuario[/password] [@SID o SERVICE NAME]

Si no ponemos el password nos lo pedirá y si no ponemos el SID cogerá el SID que tenga en la variable de entorno.

c) escribir

sqlplus /nolog

y a continuación, se usa el comando **CONN[ECT]** para conectarse a una instancia Oracle. El motor de Oracle solicitará el nombre de una cuenta de usuario existente en el servidor, por ejemplo, SYSTEM, SYS AS SYSDBA, y su contraseña.

La sintaxis del comando **CONNECT** es la siguiente:

CONN[ECT] [cláusulaAutenticación]

[AS {SYSASM | SYSBACKUP | SYSDBA | SYSDG | SYSOPER | SYSTAC | SYSKM }]

La cláusula de autenticación tiene la sintaxis siguiente:

{nombreUsuario | / }[@nombreServicioRed/]

Cuando se usa / en vez del nombre del usuario, la autenticación debe hacerse de forma externa a la solicitud de conexión, ya sea a través del sistema operativo o a través de un servidor externo de autenticación. También se puede conectar indicando la cuenta de usuario directamente en el comando connect.

Si la conexión es a un servidor remoto, es necesario indicar el nombre del servicio de red. Para ello, se emplea el carácter @, separando el nombre del nombre del servicio de red. Este nombre puede estar compuesto del nombre de host o dirección IP, el nombre FQDN de un recurso dentro de un controlador de dominio y el puerto de comunicación.

Para indicar el número del puerto que usa el servidor remoto, se usa el carácter de dos puntos detrás del nombre de host o IP. El siguiente ejemplo pretende conectarse al host *servidor* por el puerto 1521 usando la cuenta de usuario director:

connect director@servidor:1521/XE equivale a

connect director@192.168.21.1:1521/XE

Ejemplos:

COMANDOS	EQUIVALE A
C:\>sqlplus SQL>connect system	C:\>sqlplus system
SQL> connect system	SQL>connect system@XE
SQL> connect system	C:\>sqlplus system@localhost:1521/XE
SQL> connect sys as sysdba	SQL> connect / as sysdba

El comando connect / as sysdba es una autenticación usando el usuario del sistema operativo. Hay que usar el usuario con el que instalamos Oracle. Así supuestamente, si sabemos la password del usuario del SO podemos logarnos en Oracle como si fuésemos dba. Realmente también podemos entrar con cualquier usuario o incluso inexistentes as sysdba. La siguiente sentencia funciona porque estamos entrando con autenticación del sistema operativo: c:\>sqlplus nonexisto/non as sysdba

Existe el parámetro **REMOTE_LOGIN_PASSWORDFILE** que puede ser:

NONE : sólo podemos logarnos como sysdba usando autenticación del SO

EXCLUSIVE : Permite dar el privilegio SYSDBA a otros usuarios además de SYS

SHARED: El único usuario reconocido como SYSDBA es SYS

Si activamos este parámetro debemos tener creado un archivo de claves que nos permite conectarnos a oracle como SYS AS SYSDBA sin pertenecer al grupo dba o desde un puesto remoto realizando la autenticación contra el archivo. Para crear el archivo de claves usamos la utilidad orapwd

```
#> orapwd file=nombre_archivo password=clave entries=maximo_usuarios
```

siendo nombre_archivo orapwd\$ORACLE_SID y estará en \$ORACLE_HOME/dbs

Ahora podremos añadir al usuario en el archivo de claves:

```
SQL> GRANT SYSDBA to usuario
```

Podemos también ver la vista V\$PWFILERS donde estarán los usuarios sysdba o sysoper.

Ahora ya podemos conectarnos con el usuario concreto as sysdba aunque el usuario conectado siempre será SYS

Por qué tantos usuarios dba y métodos de acceso?

Hay dos tipos de conexiones:

- Interna

La cadena de conexión es simplemente el nombre del SID o del Service Name, solo sirve para conectarse al servidor local. Para que funcione el SID o Service Name tiene que estar dado de alta en el TNSNAMES.ORA

Por ejemplo, podemos poner:

```
connect system
```

```
connect system@XE
```

```
connect system@XEPDB1
```

Los dos primeros funcionarán, pero no el tercero pues no está dado de alta en el tnsnames.ora.

- Externa

La cadena de conexión es `usuario/password@servidor[ip:puerto/SID|ServiceName]`

Si no ponemos el password nos lo pedirá

Por ejemplo podemos poner:

```
connect system/oracle@localhost:1521/XE
```

```
connect system/oracle@localhost:1521/XEPDB1
```

Ambas funcionarán.

PRÁCTICA

MODIFICAR EL TNSNAMES.ORA

Verifica que te puedes conectar con sistem de forma interna y externa al contenedor (XE) pero no de forma interna a la pluggable database (XEPDB1) dando un error TNS.

Abre el tnsnames.ora que está en c:\app\usuario\product\21c\homes\OraDB21Home1\network\admin y copia el bloque completo XE = (Description.... cambiando el nombre de XE por XEPDB1 en los dos sitios que aparece.

Prueba ahora a conectarte internamente con system/password@XEPDB1

De que está compuesta una base de datos de Oracle?

El Oracle **software** (es el programa que instalas)

A **base de datos** que es una colección de archivos en uno o más discos. La base de datos contiene datos sobre el usuario, metadatos (por ejemplo el diccionario de datos) y estructuras de control (controlfile y redo logs)

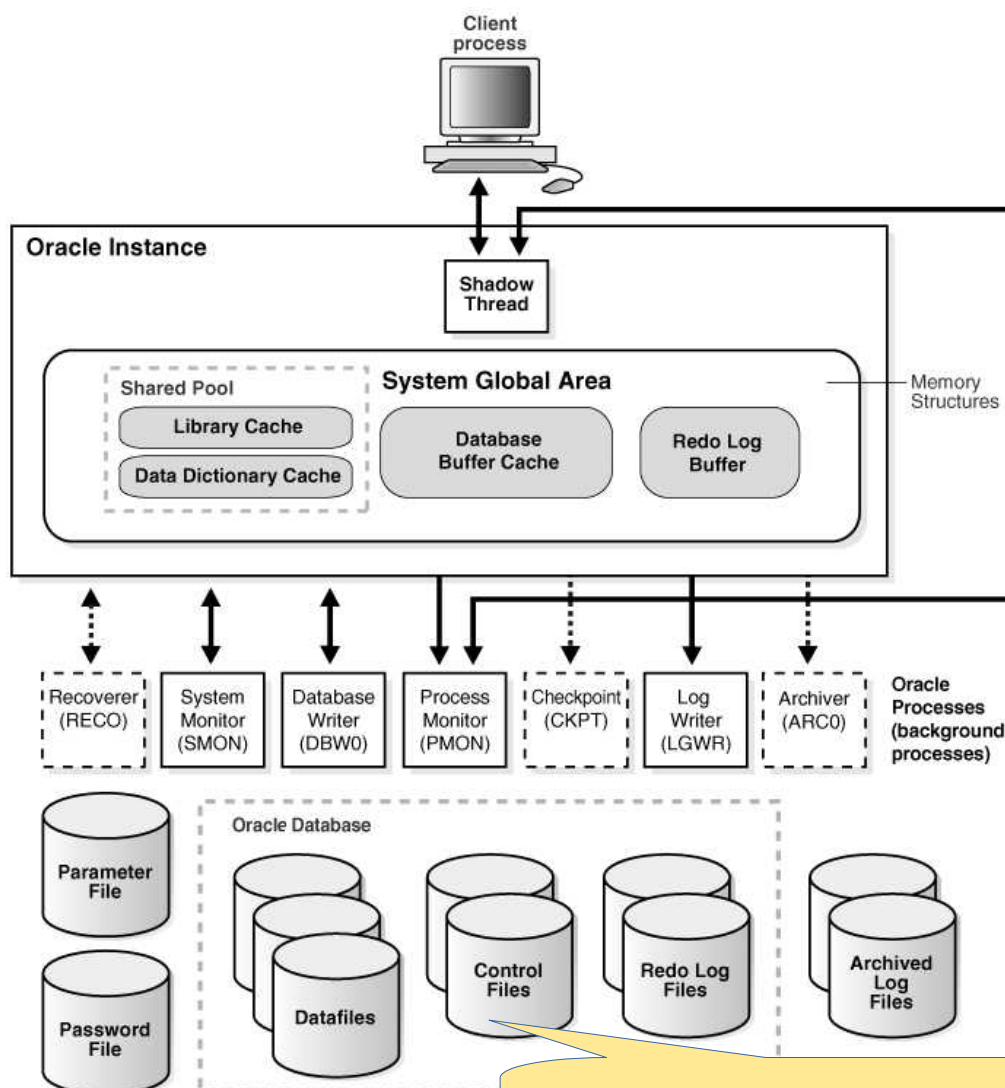
La **instancia** de Oracle que está compuesta por los procesos que corren en el ordenador en modo background y la memoria

Procesos de background que son los procesos o threads que hacen el trabajo de acceder, guardar, monitorizar y recuperar los datos del usuario, los metadatos e información de control que se encuentra en los discos.

La **shared memory** es el área de memoria usada de modo compartido por los procesos de background.

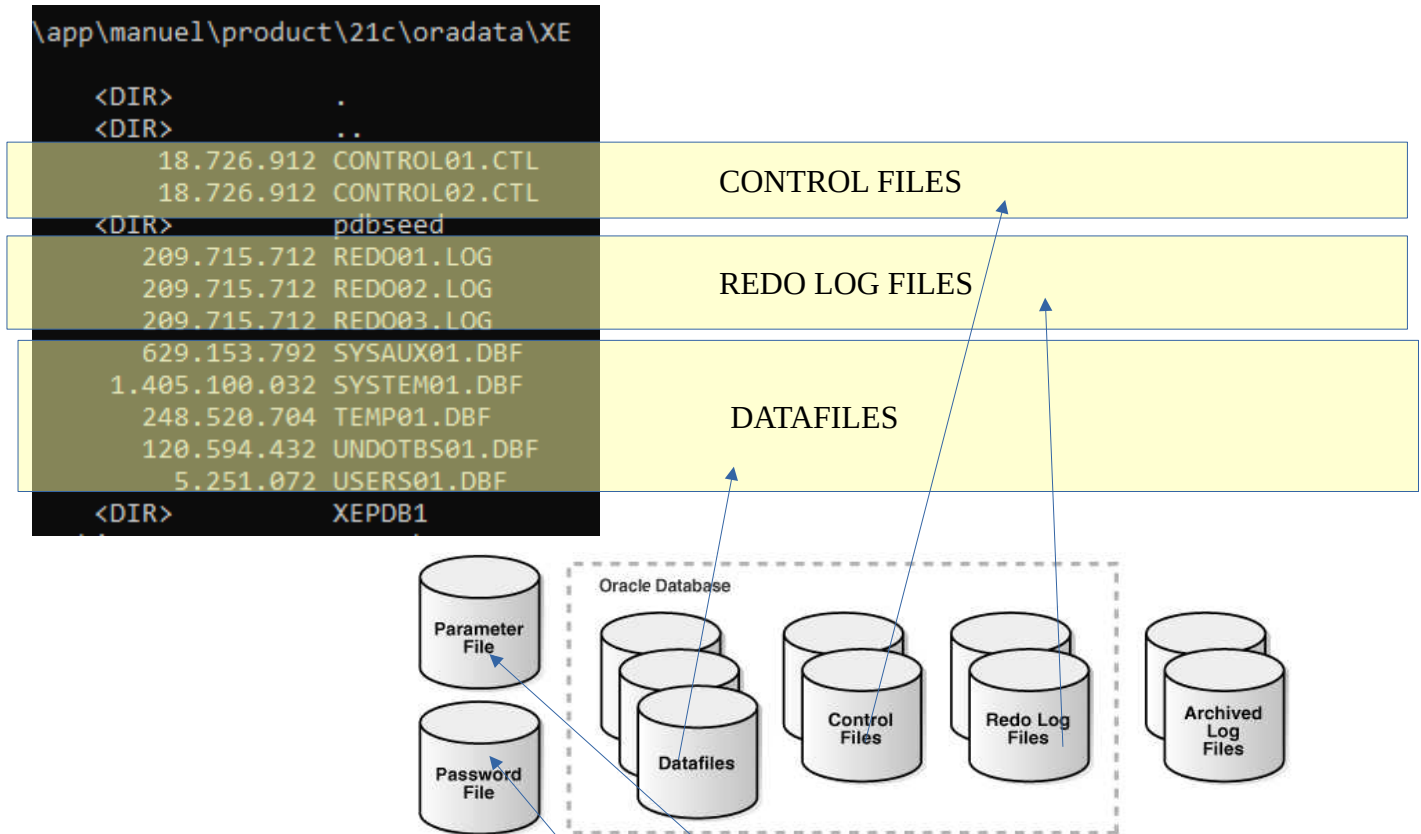
Procesos Servidor que trabajan para las aplicaciones y usuarios conectados trabajando con la memoria y el almacenamiento temporal PGA. Ejecutan las sentencias SQL y devuelven los resultados al usuario.

Oracle Net. Es una capa de software que habilita a las aplicaciones para comunicarse sobre la red. El Listener escucha las peticiones de conexión y el tnsnames describe las cadenas de conexión

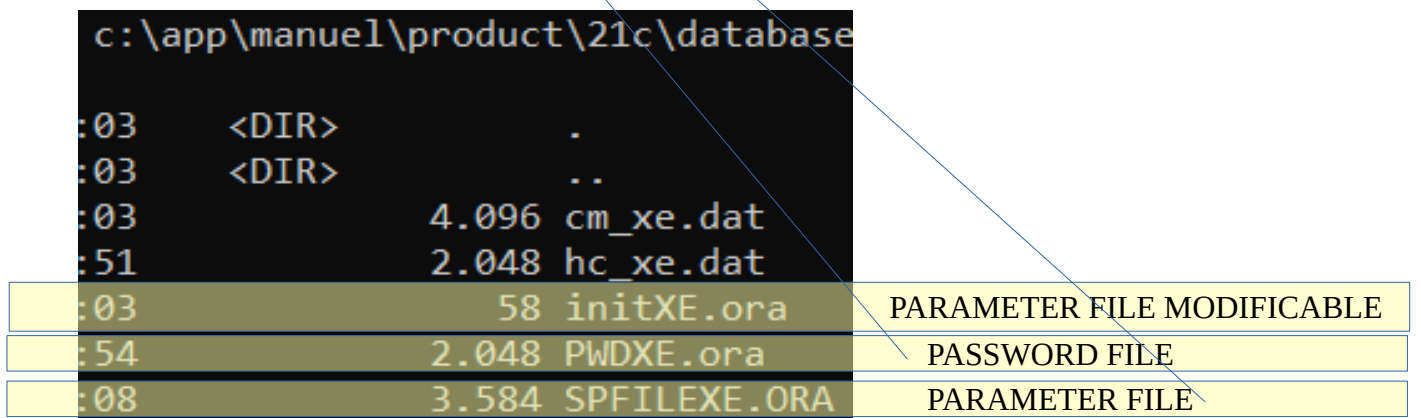


Los ficheros están en el directorio **oradata**

En mi caso por ejemplo



En database están los archivos de inicialización de Oracle



```

DEFAULT_SERVICE_LISTENER = XE

SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
(SID_NAME = CLRExtProc)
(ORACLE_HOME = C:\app\manuel\product\21c\dbhomeXE)
(PROGRAM = extproc)
(ENVS = "EXTPROC_DLLS=ONLY:C:\app\manuel\product\21c\dbhomeXE\bin\oraclr.dll")
)
)

LISTENER =
(DESCRIPTION_LIST =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = DESKTOP-SF84IHC)(PORT = 1521))
(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
)
)
)
  
```

El archivo **listener.ora** contiene el puerto en el que escucha oracle, protocolo y servicio por defecto. Este archivo está en el network\admin, en mi caso por ejemplo c:\app\manuel\product\21c\homes\OraDB21Home1\network\admin\

Podemos trabajar con el listener con el script `lsnrctl`, por ejemplo

`c:\>lsnrctl status` nos devolverá el estatus del listener y que instancia está corriendo, podemos pararlo y arrancarlo con `start`, `stop`

Que es una pluggable database?

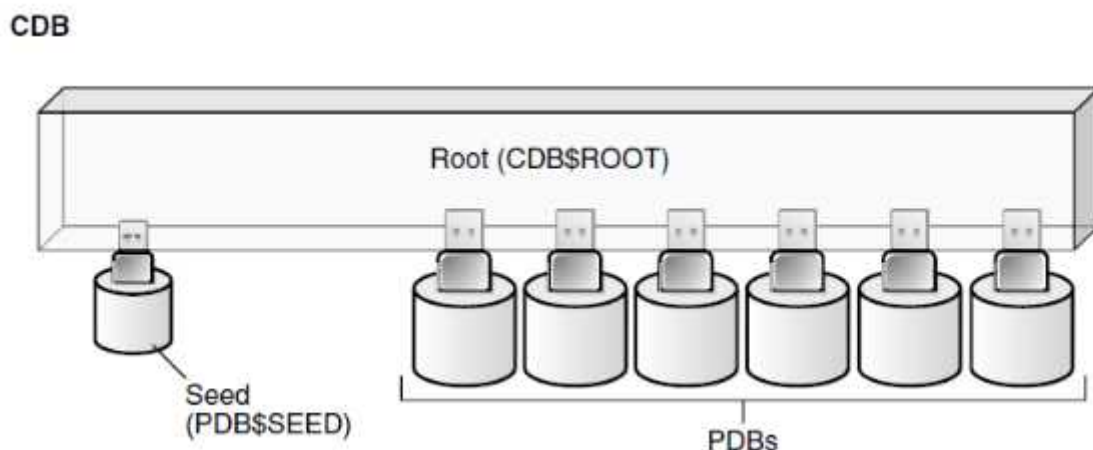
Desde Oracle 12c se simplifica la administración de entornos donde se tienen múltiples Bases de Datos independientes en un solo servidor y a su vez está preparada para soportar soluciones de Cloud con un modelo de servicio PaaS (Platform as a Service) al poder dinámicamente conectar y desconectar múltiples bases de datos de manera sencilla y dando escalabilidad a nuestra solución.

Para ello, Oracle introduce dos nuevos conceptos llamados: CDB (Container Database) y PDB (Pluggable Database). Traduciendo los términos, CDB sería una Base de Datos Contenedor y PDB una Base de Datos Agrupable, enchufable o con traducción similar.

Un CDB mantiene una colección de PDBs y una PDB es una colección de esquemas que en las versiones anteriores representaría a una base de datos Oracle.

Cada CDB tiene 3 componentes:

- a) Root, llamado CDB\$ROOT, el cual es una colección de esquemas pertenecientes al diccionario de datos que estarán disponibles para todos los PDBs.
- b) Seed, llamado PDB\$SEED, el cual es un template que podemos usar para crear nuevos PDB.
- c) PDBs, donde cada PDB cuenta con su propio identificador (GUID). Cada PDB mantiene su propio data dictionary, haciéndolo portable a diferentes CDB. Un CDB mantiene el diccionario de datos referente a la información del sistema, que cualquier PDB puede consultar.



Igual que en `oradata\XE` están los archivos de XE, los de la pluggable database estarán en el directorio `xepdb1` que está dentro de este.

Cada PDB mantiene su propio tablespace SYSTEM y SYSAUX, además de los propios tablespaces que nosotros vayamos creando.

Las operaciones de backups pueden ser realizadas tanto a nivel de CDB de PDB.

Asimismo es importante mencionar que ahora nuestros usuarios serán de 2 tipos: Usuarios Comunes (son aquellos usuarios que existen en toda PDB y con el mismo ID) y Usuarios Locales que solo existen en un PDB.

También estará permitido trabajar con base de datos en configuración NO-CDB. Cuando instalamos Oracle podremos decidir no tener contenedor y pluggable databases sinó tener una única base de datos. Esto lo haremos cuando veamos el tema de Oracle Forms.

Conectándose con sys as sysdba podemos ver las pluggable databases que tenemos con el comando `show pdbs`

Podemos saber en que database estamos (pluggables o contenedor) con el comando `SQL>show con_name`

Así mismo podemos consultar la vista dba_pdbs por ejemplo

```
SQL>select pdb_id,pdb_name,status from dba_pdbs;
```

Consulta el catálogo de Oracle: Tablespaces y datafiles

Hay muchas vistas en el catálogo:

Las que empiezan por user_ se refieren a información sobre los objetos del usuario por ejemplo user_tables , user_views, ...

Las que empiezan por all_ se refieren a información sobre los objetos sobre los que el usuario tiene permisos (aunque no sean suyos)

Las que empiezan por dba_ se refieren a información sobre todos los objetos de la base de datos y solo puede ser consultada por un usuario dba.

PRÁCTICA

CONSULTAR EL CATÁLOGO, TABLESPACES Y DATAFILES

Los tablespaces son la unidad lógica en la que se engloban los objetos de la base de datos y que se corresponde con al menos un datafile (que es un fichero físico de la base de datos que se encuentra en el oradata). Existe un tablespace especial que es el temporal que no tiene asociado un datafile sinó un tempfile. Una vista para obtener información de los ficheros de datos es V\$DATAFILE

1. Conéctate con sys as sysdba al contenedor (XE) y consulta los tablespaces

```
SQL>connect sys@XE as sysdba
```

```
SQL>select tablespace_name from dba_tablespaces
```

2. Consulta los datafiles, como las columnas pueden ser muy largas vamos a formatearlas

```
SQL> set linesize 190
```

```
SQL> column file_name format a60
```

```
SQL> select file_name, tablespace_name from dba_data_files;
```

3. Conéctate ahora a la pluggable database y haz lo mismo. Mira las diferencias

4. Consulta ahora las vistas dba_temp_files

Consulta el catálogo de Oracle: parámetros de instancia

Podemos consultar los **logfiles** con la vista V\$LOGFILE

```
SQL> select group#, status, member from V$LOGFILE
```

Los archivos de log (logfiles) son muy importantes ya que permiten la recuperación de los datos en caso de interrupción anormal del sistema hasta el último commit que se haya realizado. Los archivos de log están también en el oradata como hemos visto.

Por defecto hay tres grupos y un miembro en cada grupo. Oracle aconseja que haya más de un miembro por grupo y que estén en discos distintos por seguridad.

Consultando el archivo de control (**controlfile**) con la vista V\$CONTROLFILE

```
SQL> select name,status,is_recovery_dest_file from V$CONTROLFILE;
```

El archivo de control es muy importante por eso está duplicado por si uno falla. Oracle recomienda que esté cada copia en un disco distinto por si falla un disco poder recuperar la base de datos usando la otra copia del controlfile. Sin el controlfile la base de datos no arranca.

Consultando donde está el archivo de inicialización (**SPFILE**)

```
SQL> show parameters spfile;
```

El fichero de inicialización contiene parámetros de la instancia como el número de procesos, tamaño de memoria, etc.

Consultando los **Archive Logs** (Log archivados)

```
SQL> archive log list
```

Veremos que el archive log está desactivado. El archive log es un método de backup que permite recuperar la base de datos con el último backup hasta justo el último commit realizado antes de que fallase.

las vistas V\$ARCHIVED_LOG y V\$ARCHIVE_DEST nos dan información sobre los log archivados.

Consultando la memoria compartida System Global Area (**SGA**)

```
SQL> show parameters sga
```

el parámetro sga_max_size nos dirá el tamaño de la memoria (Oracle XE viene limitado a no más de 2GB de memoria)

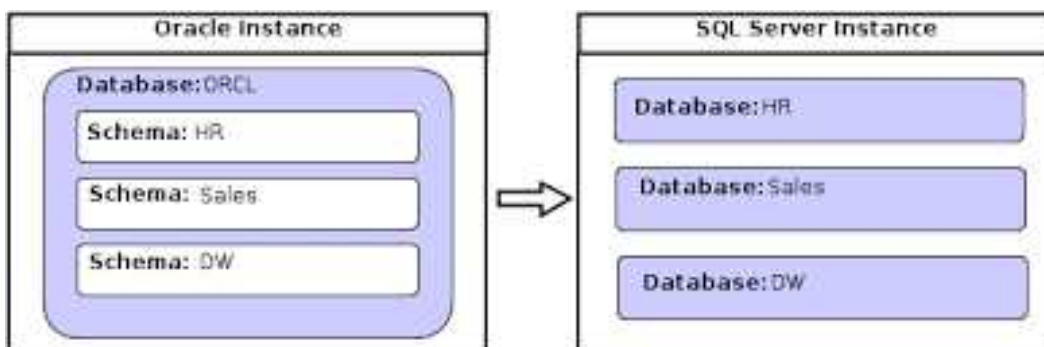
Consultando los **cpu's** (fíjate en el parámetro `cpu_count`)

`SQL> show parameters cpu`

Oracle XE viene limitado, realmente Oracle cuando crea la instancia se fija en cuanto memoria tiene el sistema y cuantas CPU's así pueden aparecer valores grandes, sin embargo solo usará 2 cpu's y 2GB de memoria como máximo.

Instalando el esquema del usuario de ejemplo HR

HR es un usuario de ejemplo de Oracle con una serie de tablas. Recordemos que Oracle es distinto a otras bases de datos como Mariadb. En Oracle los usuarios son propietarios de sus tablas y a esto se le llama esquema de datos.



En Mariadb se crean databases con `create database` y se le dan permisos a los usuarios para usarlas y que creen ahí sus objetos. En Oracle cada usuario es propietario de sus objetos (y por tanto nadie puede quitarle los permisos sobre ellos), esos objetos conforman el esquema de datos del usuario, y simplemente se pueden dar permisos a otros usuarios.

Para crear el usuario de prueba hr de Oracle tenemos que correr un script `.sql`. Para hacerlo usamos `start` o `@nombre_del_fichero.sql` en `sqlplus`

PRÁCTICA

CREAR ESQUEMA HR

1. Vete a [este site](#) y bájate el zip de la última versión de los esquemas
2. Descomprime el zip
3. Desde el terminal vete al directorio `human_resources`, verás un archivo llamado `hr_install.sql` este archivo es el que tienes que correr en Oracle.
4. Lógate en Oracle usando el `sqlplus` con el usuario `system` o `sys` en la pluggable database (ya que el usuario HR será un usuario solo de esa pluggable).

```
SQL> connect system@XEPDB1
```

```
SQL> @hr_install.sql
```

Te preguntará por la password de hr (ponle por ejemplo `oracle`), el tablespace (puedes darle a enter y se crearán las tablas en el tablespace `USERS` que es el que tiene por defecto).

Una vez haya terminado saldrá del `sqlplus`. Prueba a entrar con el usuario `hr` y lista sus tablas.

Fíjate que no podrás entrar con `connect hr/oracle` porque eso intenta conectarse a XE, tienes que poner la pluggable database

comandos del SQL*PLUS

EL comando HELP muestra una ayuda sobre la sintaxis, uso y ejemplos de un conjunto de comandos útiles para la consola de comandos. Escribiendo HELP INDEX se obtienen los comandos sobre los que se puede mostrar información. En la Figura se muestran estos comandos.

Comando HELP			
@	COPY	PASSWORD	SHOW
@@	DEFINE	PAUSE	SHUTDOWN
/	DEL	PRINT	SPOOL
ACCEPT	DESCRIBE	PROMPT	SQLPLUS
APPEND	DISCONNECT	QUIT	START
ARCHIVE LOG	EDIT	RECOVER	STARTUP
ATTRIBUTE	EXECUTE	REMARK	STORE
BREAK	EXIT	REPFOOTER	TIMING
BTITLE	GET	REPHEADER	TTITLE
CHANGE	HELP	RESERVED WORDS (SQL)	UNDEFINE
CLEAR	HISTORY	RESERVED WORDS (PL/SQL)	VARIABLE
COLUMN	HOST	RUN	WHENEVER OSERROR
COMPUTE	INPUT	SAVE	WHENEVER SQLERROR
CONNECT	LIST	SET	XQUERY

Comando DESC[RIBE]

Se usa para conocer la descripción de una tabla, vista o sinónimo. También se puede usar para conocerla especificación de un procedimiento o de una función. Este comando se emplea mucho para conocer las columnas de las vistas del diccionario de datos. La sintaxis de este comando es la siguiente:

El siguiente comando muestra las columnas de la vista DBA_TABLES:

DESCRIBE DBA_TABLES

Comando START o @

Ya hemos visto que se usa START o @ para ejecutar un script sql. Aunque la extensión predeterminada de los scripts Oracle es .sql, se puede usar otra, siempre que se indique en su nombre completo, ya que, en realidad, se trata de un fichero plano.

En el path del fichero se puede usar el carácter ? con el fin de indicar la ruta hacia el directorio Oracle Home. En las rutas se puede usar tanto el carácter / como el carácter \.

Por ejemplo, para ejecutar el script utlpwdmg ubicado en el directorio /rdbms/admin alojado en el directorio Oracle Home se usaría el comando @?/rdbms/admin/utlpwdmg.

Comando HISTORY

Permite almacenar, listar, ejecutar y editar los diferentes comandos introducidos en una sesión de la consola de SQL *Plus. La sintaxis de este comando es la que sigue:

HIST[ORY] [N { RUN | EDIT | DEL[ETE]}] | [CLEAR]

Si no se indica ninguna opción muestra el historial de comandos ejecutados. El valor N hace referencia al número de la lista de comandos. Se usará RUN para volver a ejecutar el comando, EDIT para editarlo antes de ejecutarlo y DELETE para eliminar el comando de la lista. La opción CLEAR elimina todo el historial. Por ejemplo, el comando siguiente ejecuta el comando decimoquinto de la lista

hist

hist 15 run

Para activar el historial se usa el comando SET HIST[ORY] ON y SET HISTORY OFF para desactivarlo.

Comando COLUMN

Este comando permite configurar una serie de atributos para las columnas que se muestran en una consulta. La sintaxis de este comando es como sigue:

COL[UMN] [{Columna | expr} [opciones...]

Los valores de opciones pueden ser ALI[AS] alias, CLE[AR], ENTMAP {ON|OFF}, FOLD_A[FTER], FOLD_B[EFORE], FOR[MAT] formato HEA[DING] texto, JUS[TIFY] {L[EFT] C[ENTER] R[IGHT]}, LIKE {expr | alias}, NEWL[INE], NEW_V[ALUE] variable, NOPRI[NT] PRI[NT], NUL[L] texto, OLD__V[ALUE] variable, ON | OFF y WRA[PPED] WOR[D_WRAPPED] | TRU[NCATED].

Las opciones más usadas son FORMAT y HEADING, que permiten indicar el formato de salida y la cabecera de una columna. Con la opción A seguida de un número es posible indicar el número de caracteres para esa columna. En el siguiente ejemplo se declara el formato y la cabecera para las columnas product, version_full y status en la consulta de más abajo:

COL product FOR A29 HEA Producto

COL version_fu11 FOR A19 HEA Version

COL status FOR A19 HEA Estado

```
SELECT PRODUCT, VERSION_FULL, STATUS  
FROM PRODUCT_COMPONENT_VERSION;
```

También se puede usar el alias para formatear si lo usa la consulta:

COL pr FOR A29 HEA Producto

COL ve FOR A19 HEA

COL st FOR A19 HEA Estado

```
SELECT PRODUCT pr, VERSION_FULL ve, STATUS st  
FROM PRODUCT_COMPONENT_VERSION;
```

Comando SHOW

El comando SHOW es distinto al comando SHOW PARAMETERS, ya que el primero muestra el valor de las variables del sistema o del entorno de SQL *Plus, y el segundo el valor de un parámetro de inicialización. La sintaxis de este comando es la siguiente:

SHO[W] opción

La opción puede ser uno de estos valores: variables del sistema, ALL, BTITLE, CON_ID, CON_NAME, EDITION, ERR[ORS] [{ANALYTICVIEW | ATTRIBUTE DIMENSION | HIERARCHY | FUNCTION | PROCEDURE | PACKAGE | PACKAGE BODY | TRIGGER | VIEW | TYPE | TYPE BODY | DIMENSION | JAVA CLASS} [esquema | nombre], LNO, PARAMETERS [nombre_parámetro], PDBS, PNO, RECYC[LEBIN] [nombre_ordinal], REL[EASE], REPF[OOTER], REPH[EADER], SGA, SPOOL[L], SPPARAMETERS [nombre_parámetro], SQLCODE, TTI[TLE] y USER.

En la lista siguiente se destacan los más interesantes:

SHOW ALL: muestra el valor de todas las variables del sistema.

SHOW AUTOCOMMIT: muestra el contenido de la variable del sistema con nombre AUTOCOMMIT.

SHOW CON_ID: muestra el identificador del contenedor actual.

SHOW CON_NAME: muestra el nombre del contenedor actual.

SHOW BTITLE: muestra el título para la siguiente consulta.

SHOW PDBS: muestra el identificador y el nombre de cada uno de los contenedores visibles por el usuario.

SHOW TITTLE: muestra la cabecera título de la siguiente consulta.

SHOW SGA: muestra información del área de memoria compartida.

SHOW SPOOL: muestra con OFF y ON si la salida a fichero está activada.

SHOW USER: muestra el usuario activo.

Así como el comando SHOW puede mostrar el valor de una variable del sistema, como, por ejemplo, SHOW PAGESIZE, el comando SET permite introducirle un valor. Para conocer todas las variables del sistema y los valores que pueden tomar se puede escribir el comando HELP SET.

Comando SPOOL

Permite almacenar el resultado de un comando en un fichero. La sintaxis es la siguiente:

SPO[OL] [nombre_fichero[.ext]] [CRE[ATE] | REP[LACE] | APP[END]] | OFF | OUT

Por ejemplo, para activar SPOOL e indicar que se vuelquen los resultados de los comandos al fichero c:\oradata\salida.txt se escribe:

SPOOL c:\oradata\saLida.txt

Si no se indica extensión, Oracle añade .lst. Si el fichero ya existe y se quiere reemplazar, se usa la opción REPLACE. Si se quiere añadir información nueva al final del fichero, se usa la opción APPEND. La opción SPOOL se activa con **SPOOL ON** y se desactiva con **SPOOL OFF**

Comando DEFINE

Se usa DEFINE para definir una variable de sustitución. En este caso no es necesario usar comillas simples cuando es una cadena. La sintaxis es la siguiente.

DEFINE variable = valor

Cuando la variable es seguida por una letra, número, punto o subrayado es necesario utilizar un punto para delimitar el fin del nombre de la variable. Por ejemplo, se define una variable que se llama Prefijo con el texto user_. Para ello, se escribe el comando siguiente:

DEFINE prefijo=user_

Para ejecutar la consulta **SELECT COUNT(*) FROM USER_CLUSTERS;** usando la variable Prefijo se escribe:

SELECT COUNT(*) FROM &Prefijo.CLUSTERS;

Variables de sustitución

Las variables de sustitución reemplazan a un valor que se solicita por teclado. Un nombre precedido por el símbolo & obliga a Oracle a solicitar un valor para una variable con ese nombre. El siguiente ejemplo ejecuta una consulta para mostrar los productos del tipo que se solicita por teclado:

SELECT * FROM Producto WHERE Tipo='&TipoProducto';

Una variable de sustitución se entrecomilla cuando se trata de una cadena; en caso contrario se escribirá sin comillas.

Por defecto, cuando SQL *Plus efectúa una sustitución muestra un mensaje con la sentencia SQL antes y después de la sustitución. Esta funcionalidad se activa usando **SET VERIFY ON | OFF**.

Paso de valores a un script

En el contenido de un script se usan las variables de sustitución &1, &2, &3, etc., que sustituyen a los argumentos pasados al script. Un script sql se ejecuta con START o desde el sistema operativo ejecutando el comando sqlplus y usando el carácter @ para indicar el nombre del fichero. El siguiente ejemplo se ejecuta desde Windows; se conecta con la cuenta de usuario alumno y ejecuta el script llamado selectProducto.sql. A este script se le pasan dos argumentos, el primero C y el segundo nombre.

Ejemplo haciendo uso de la línea de comandos:

sqlplus alumno/pass@selectProducto.sql C nombre

Un ejemplo del contenido del script selectProducto.sql es el siguiente:

```
SELECT &2 FROM Producto WHERE Tipo='&1';
```

Sustituyendo los argumentos, sería equivalente a la consulta siguiente:

```
SELECT nombre FROM Producto WHERE Tipo='C';
```

Otros comandos

A continuación se detallan algunos comandos interesantes, cuya ayuda se puede obtener escribiendo HELP y el nombre del comando:

TIMING: muestra el tiempo requerido para ejecutar un comando SQL. Para que se muestre en cada consulta se escribe **SET TIMING ON**. Su sintaxis es

TIMI[NG]] [START texto | SHOW | STOP].

HOST: abre una conexión a la consola de comandos del sistema operativo o ejecuta un comando del sistema operativo si este se indica. Se escribe EXIT para volver a SQLPLUS. Su sintaxis es **HO[ST] [comando]**.

PASSWORD: permite cambiar la contraseña del usuario activo o de un usuario si se indica después de PASSWORD. Su sintaxis es **PASSW[ORD] [nombreUsuario]**.

El diccionario de datos de Oracle

El diccionario de datos es un conjunto de tablas y vistas que dan información sobre el contenido de una base de datos: las estructuras de almacenamiento, los usuarios y sus derechos, los objetos como tablas, vistas, índices, procedimientos, funciones, etc. A estos datos, a los que hay que sumar datos sobre la propia base de datos (llamados metadatos) componen el diccionario de datos. Hay dos grupos de tablas y vistas en el diccionario de datos: las tablas y las vistas estáticas; las tablas y las vistas dinámicas de rendimiento.

Las tablas y las vistas estáticas están basadas en las verdaderas tablas almacenadas en el tablespace System. Son accesibles únicamente cuando la base de datos está completamente abierta. Las tablas y las vistas dinámicas de rendimiento no están basadas en verdaderas tablas, sino en la información en memoria o en extractos del fichero de control. No obstante, se consultan como si fueran verdaderas tablas y vistas y dan información sobre el funcionamiento de la base de datos. La mayoría son accesibles, aunque la base de datos no este completamente abierta. El diccionario de datos pertenece a SYS y se almacena en el tablespace SYSTEM.

Los objetos del diccionario de datos a los que un usuario puede acceder se encuentran en la vista DICTIONARY y es propiedad de SYS

Vistas estáticas

Las vistas estáticas comienzan con el prefijo USER_ o ALL_. Las primeras hacen referencia a a visión del usuario de la base de datos, es decir, obtienen los objetos que el usuario posee. De este modo, con la vista USER_TABLES el usuario obtiene información de sus tablas. Los sinónimos para estas vistas son obj, tabs, cols, ind, cols, ind syn y seq.

```
SELECT COUNT(*) FROM USER_OBJECTS; --equivale a  
SELECT COUNT(*) FROM OBJ;
```

El prefijo ALL_ hace referencia a los objetos a los que el usuario tiene acceso, bien porque sean suyos o porque tenga privilegios otorgados por los que son sus dueños.

El prefijo DBA_ sólo es usado por las vistas a las que tiene acceso el DBA y se refieren a todos los objetos de la base de datos sean de quien sean.

Esta es una lista de las vistas estáticas:

- USER|ALL|DBA_OBJECTS: información de cualquier objeto de la base de datos.
- USER|ALL|DBA_TABLES: información de las tablas.
- USER|ALL|DBA_TAB_COLUMNS: información de las columnas de las tablas.
- USER|ALL|DBA_INDEXES: información de los índices.
- USER|ALL|DBA_TRIGGERS: información sobre los disparadores.
- USER|ALL|DBA_CONSTRAINTS: información sobre las restricciones.
- USER|ALL|DBA_CONS_COLUMNS: información sobre las columnas de las restricciones.
- USER|ALL|DBA_VIEWS: información de las vistas.
- USER|ALL|DBA_SYNONYMS: información de los sinónimos.
- USER|ALL|DBA_SEQUENCES: información de las secuencias.

Las vistas estáticas se pueblan con el diccionario de datos y por tanto tiene que estar la base de datos abierta para poder usarlas.

Un caso especial es la vista DBA_OBJECTS, esta vista incluye información sobre los objetos de la base de datos, sus campos son:

Nombre	¿Nulo?	Tipo
OWNER		VARCHAR2(128)
OBJECT_NAME		VARCHAR2(128)
SUBOBJECT_NAME		VARCHAR2(128)
OBJECT_ID		NUMBER
DATA_OBJECT_ID		NUMBER
OBJECT_TYPE		VARCHAR2(23)
CREATED		DATE
LAST_DDL_TIME		DATE
TIMESTAMP		VARCHAR2(19)
STATUS		VARCHAR2(7)
TEMPORARY		VARCHAR2(1)
GENERATED		VARCHAR2(1)
SECONDARY		VARCHAR2(1)
NAMESPACE		NUMBER
EDITION_NAME		VARCHAR2(128)
SHARING		VARCHAR2(18)
EDITIONABLE		VARCHAR2(1)
ORACLE_MAINTAINED		VARCHAR2(1)
APPLICATION		VARCHAR2(1)
DEFAULT_COLLATION		VARCHAR2(100)
DUPLICATED		VARCHAR2(1)
SHARDED		VARCHAR2(1)
IMPORTED_OBJECT		VARCHAR2(1)
CREATED_APPID		NUMBER
CREATED_VSNID		NUMBER
MODIFIED_APPID		NUMBER
MODIFIED_VSNID		NUMBER

La columna object_type discrimina el tipo de objeto. Así, si se quiere obtener información sobre las restricciones, el valor de object_type sería 'CONSTRAINTS'. Por ejemplo:

```
SELECT * FROM DBA_OBJECTS WHERE OBJECT_TYPE='CONSTRAINTS';
```

Para conocer todos los tipos que existen se puede usar esta misma vista consultando el distinct object_type, por ejemplo:

```
SELECT DISTINCT OBJECT_TYPE FROM DBA_OBJECTS;
```

Por ejemplo la columna CREATED nos dice la fecha de creación del objeto. Así podemos saber en que fecha se creó una tabla:

```
SELECT OBJECT_NAME, CREATED FROM DBA_OBJECTS WHERE OBJECT_TYPE='TABLES'
```

Vistas dinámicas de rendimiento

Las vistas dinámicas de rendimiento tienen el prefijo V\$ y la mayoría solo son accesibles por el administrador. Son sinónimos de las vistas que empiezan por V_

Las vistas de interpretación dinámicas ofrecen información sobre la instancia y sobre la base de datos. La mayoría de las vistas tienen información de la instancia y el resto del fichero de control. Las vistas de interpretación dinámicas son creadas en el startup, actualizadas durante la vida de la instancia, y desaparecen con el cierre. Son pobladas por la instancia o por el fichero de control; las vistas DBA_, ALL_ y USER_ son pobladas del diccionario de datos.

Esta es una lista de las más usadas:

V\$SQL : tiene una fila por cada sentencia corriente almacenada en la Shared Pool

V\$DATABASE: sobre la base de datos

V\$DATAFILE: Sobre los datafiles. Accesible con la base de datos en MOUNT

V\$LOGFILE: Sobre los ficheros de log

V\$TABLESPACE : Sobre los tablespaces.

V\$INSTANCE o V\$SYSTAT: sobre la instancia actual. Accesible desde NOMOUNT

V\$OPTION: sobre las opciones de instalación en el servidor.

V\$PARAMETER: sobre los parámetros de inicialización.

V\$PROCESS: sobre los procesos activos.

V\$SESSION: sobre la sesión actual.

V\$SGA y V\$SGAINFO: sobre la memoria compartida SGA.

V\$VERSION: sobre los componentes y sus versiones.

PRÁCTICA

CONSULTA EL DICCIONARIO DE DATOS

Usando la vista del diccionario que porceda, construye la consulta para mostrar el número de objetos creados durante los últimos 30 días.

Crea un procedimiento almacenado al que le pases el id de un usuario de la base de datos y te devuelva por pantalla el nombre de todas sus tablas.