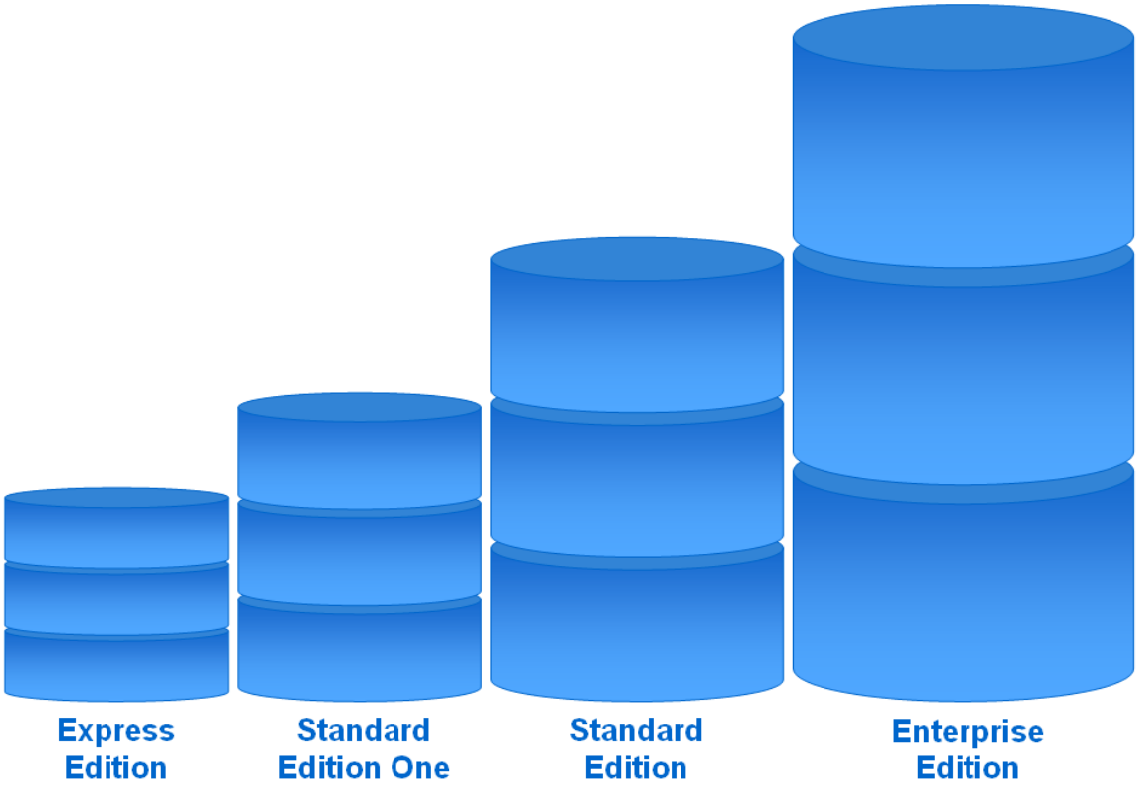
Perú Dev

Consultoría y Capacitación



Oracle Database 10g

Administración I

Primera Edición

Eric Gustavo Coronel Castillo

Febrero - 2009

Oracle Database 10g Administración I

Derechos Reservados © 2009 Eric Gustavo Coronel Castillo

Primera Edición

Contacto

Email: gcoronelc@gmail.com

Teléfono: (511) 996-664-457

Lima - Perú

Contenido

Lección 01: Arquitectura de la Base de Datos Oracle 10g

Arquitectura de Oracle 10g 2
La Instancia de Oracle 3
La Base de Datos de Oracle 4

Lección 02: Introducción a las Estructuras de Almacenamiento

Estructuras Lógicas 14
Estructuras Físicas 16
Esquemas de Base de Datos 18

Lección 03: El Diccionario de Datos

Introducción al Diccionario de Datos 22
Componentes Privados 23
Componentes Públicos 28

Lección 04: Usuarios Administradores

Administrador de la Seguridad y Privilegios de la Base de Datos 32
Autenticación del Administrador de Base de Datos 34
Creación y Manteniendo el Archivo de Contraseñas 44

Lección 05: Inicio y Parada de la Base de Datos

Introducción 52
Inicio de una Base de Datos de Oracle 53
Comando: STARTUP 55
Comando: SHUTDOWN 56
Comando: ALTER DATABASE 59

Lección 06: Acceso a la Base de Datos Oracle

Verificación de Servicios Windows 62
Identificación del Servicio y la Instancia de Base de Datos 65
Configuración del Método de Nomenclatura Local 75
Configuración del Proceso Escucha 77
Glosario 79

Lección 07: Administración de la Instancia

Parámetros de Inicialización 84
Dimensionando el SGA 94
Manejo de Sesiones 97
Archivos de Alertas 103

Lección 08: Administración del Archivo de Control

¿Qué es el Archivo de Control? 106
Establecer el Nombre de los Archivos de Control 107
Multiplexado el Archivo de Control 108
Creando Copias Adicionales, Renombrando y Reubicando Archivos de Control 109
Mostrando Información del Archivo de Control 111

Lección 09: Administración del los Redo Logs

¿Que es el Redo Log? 114
Planificación de los Redo Log Files 118
Creando Groupos y Miembros Redo Log 119
Eliminando Grupos y Miembros Redo Log 121
Consultando Información sobre los Redo Log 123
Ejercicio 123

Lección 10: Administración de Tablespaces

Esquema de la Base de Datos 126
Guía para Administrar Tablespaces 128
Administración de Tablespaces 129
Tablespace Administrado Localmente 131
Tablespaces Temporales 137
Especificando Bloques NO Estándares para Tablespaces 139
Gestionando Tablespaces 140
Gestionando Tablespaces 140

Lección 11: Usuarios y Recursos

Administración de Usuarios 144
Administración de Recursos con Perfiles 149
Verificar Información sobre Usuarios y Perfiles 160

Lección 12: Privilegios y Roles

Entendiendo Privilegios y Roles 164
Administración de Roles 169
Concediendo Privilegios y Roles 171
Revocando Privilegios y Roles 176
Verificar Información sobre Privilegios y Roles 180

Lección 01

Arquitectura de la Base de Datos Oracle 10g

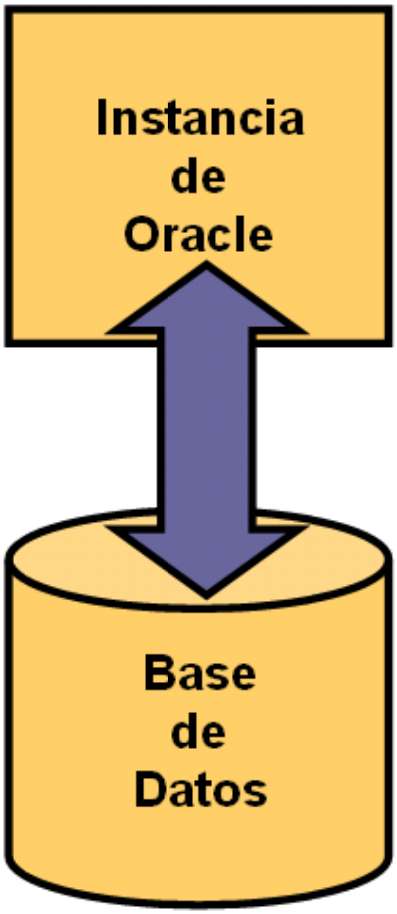
Contenido

- Arquitectura de Oracle 10g 2
  - Esquema General 2
  - Esquema Ampliado 3
- La Instancia de Oracle 4
  - Estructuras en Memoria 4
  - El System Global Area (SGA) 5
  - Procesos de Fondo 7
- La Base de Datos de Oracle 9

Arquitectura de Oracle 10g

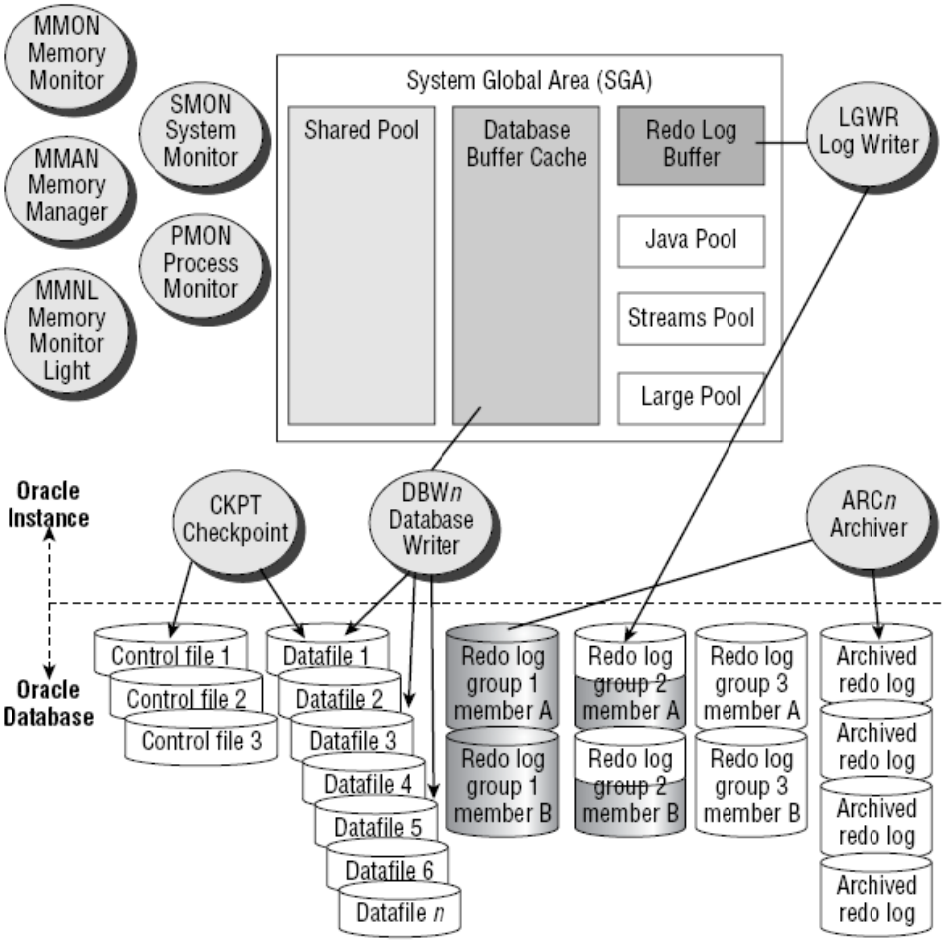
Esquema General

El servidor Oracle está compuesto de varios archivos, procesos y estructuras en memoria que se agrupan en dos componentes: La Instancia Oracle y la Base de Datos Oracle.



Por cada instancia de Oracle se tiene una sola base de datos, en un servidor se pueden crear varias instancias, pero se recomienda solo una, porque cada instancia consume muchos recursos.

Esquema Ampliado



Para consultar el nombre de la instancia ejecute la siguiente consulta:

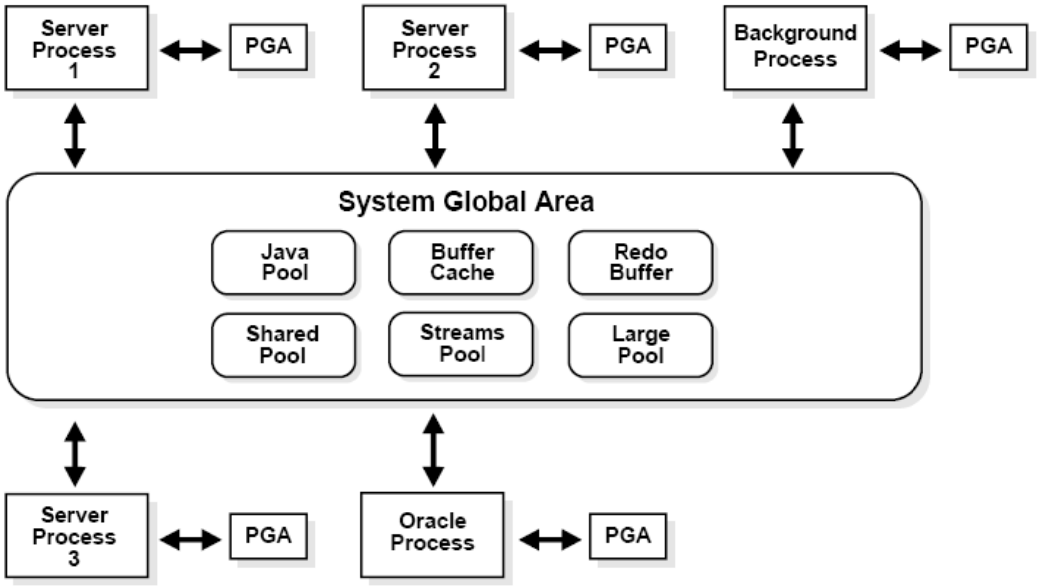
```
SQL> select instance_name from v$instance;   [Enter]

INSTANCE_NAME
----------------
Orcl
```

Para consultar el nombre de la base de datos ejecute la siguiente consulta:

```
SQL> select name from v$database;    [Enter]

NAME
---------
ORCL
```

La Instancia de Oracle

Estructuras en Memoria



Oracle utiliza la memoria para almacenar información como:

- Código de los programas.
- Información sobre las sesiones actuales, incluso si no está activa.
- La información necesaria durante la ejecución de un programa, Por ejemplo, el estado actual de una consulta de la cual se están trayendo las filas.
- Información que es compartida y se comunica entre procesos de Oracle. Por ejemplo, información bloqueada.
- Contiene los datos que también se almacenan permanentemente en memoria periférica. Por ejemplo, los bloques de los datos y entradas del redo log.

Las estructuras básicas de memoria asociadas con Oracle incluyen:

- System Global Area (SGA), que es compartida por todos los procesos del servidor y procesos de fondo.
- Program Global Area (PGA), que es privado a cada proceso del servidor y procesos de fondo; hay un PGA para cada proceso.

El System Global Area (SGA)

El SGA está conformado por tres componentes obligatorios y tres opcionales.

Componentes Obligatorios del SGA

El siguiente cuadro muestra los componentes obligatorios del SGA.

Componente SGA	Descripción
Shared Pool	Contiene las sentencias SQL usadas mas recientemente que han sido ejecutadas por los usuarios de la base de datos.
Database Buffer Cache	Contiene los datos que han sido mas recientemente accedidos por los usuarios de la base de datos.
Redo Log Buffer	Almacena información de las transacciones para propósitos de recuperación.

Componentes Opcionales del SGA

El siguiente cuadro muestra los componentes opcionales del SGA.

Componente SGA	Descripción
Java Pool	Almacena el código de las aplicaciones y objetos Java que se han usado más recientemente cuando se utiliza el JVM de Oracle.
Large Pool	Almacén de datos para grandes operaciones, como actividades de backup y restauración de Recovery Manager (RMAN) y componentes compartidos del servidor.
Streams Pool	Almacén de data asociada con requerimientos hechos a la cola de mensajes cuando Oracle's Advanced Queuing esta en uso.

Para consultar el SGA ejecute el siguiente comando:

```
SQL> show sga   [Enter]

Total System Global Area  289406976 bytes
Fixed Size                  1247528 bytes
Variable Size              83887832 bytes
Database Buffers          197132288 bytes
Redo Buffers                7139328 bytes
```

La salida de este comando muestra el tamaño total de SGA, es 289,406,976 bytes. Este tamaño total está compuesto de un espacio variable (Variable Size) que lo conforma el Shared Pool, el Large Pool, y el Java Pool (1,247,528 bytes), el Database Buffer Cache (197,132,288 bytes), el Redo Log Buffer (7,139,328 bytes), y el espacio adicional (1,247,528 bytes) que guarda información usada por los procesos de fondo de la instancia.

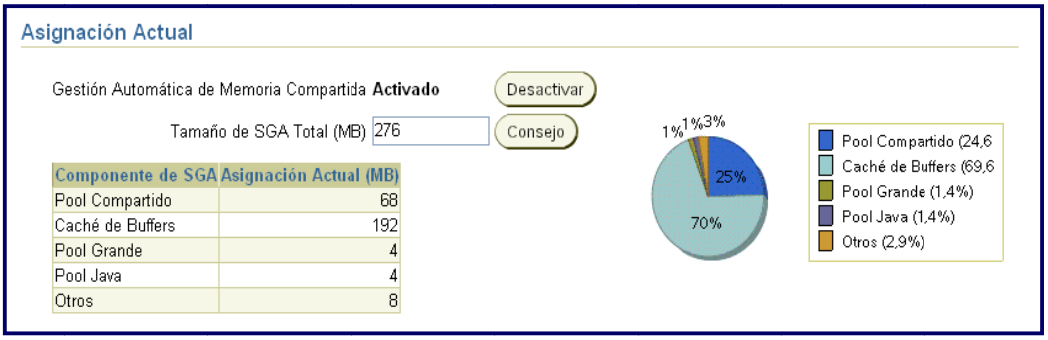
También podemos utilizar la vista V$SGA para consultar información sobre el SGA, tal como se ilustra a continuación:

```
SQL> select * from v$sga;  [Enter]

NAME                     VALUE
-------------------- ----------
Fixed Size              1247528
Variable Size          83887832
Database Buffers      197132288
Redo Buffers            7139328
```

La vista V$SGA_DYNAMIC_COMPONENTS permite consultar detalles adicionales sobre el espacio asignado dentro del SGA; según lo muestra la siguiente consulta:

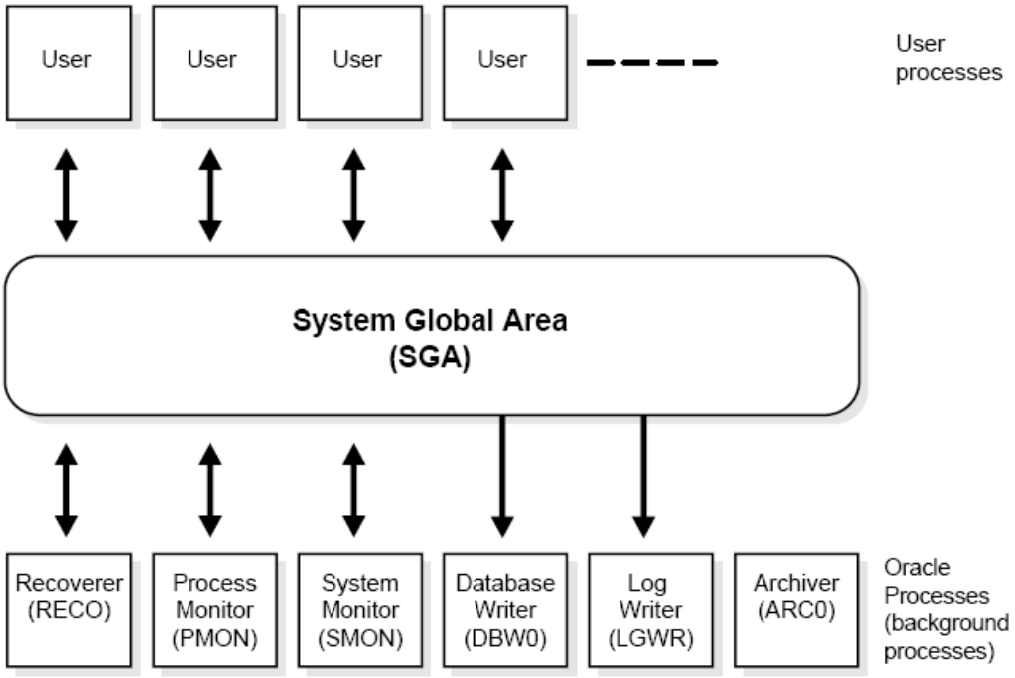
```
SQL> select component,current_size from v$sga_dynamic_components; [Enter]

COMPONENT                      CURRENT_SIZE
------------------------------ ------------
shared pool                        75497472
large pool                          4194304
java pool                           4194304
streams pool                              0
DEFAULT buffer cache              197132288
KEEP buffer cache                         0
RECYCLE buffer cache                      0
DEFAULT 2K buffer cache                   0
DEFAULT 4K buffer cache                   0
DEFAULT 8K buffer cache                   0
DEFAULT 16K buffer cache                  0
DEFAULT 32K buffer cache                  0
ASM Buffer Cache                          0

13 filas seleccionadas.
```

También se puede usar el Enterprise Manager Database Control para consultar el tamaño de cada componente del SGA, tal como se muestra a continuación:



Procesos de Fondo

Hay muchos tipos de procesos de fondo de Oracle. Cada uno realiza un trabajo específico que ayuda a manejar la instancia. Oracle requieren cinco procesos de fondo de manera obligatoria, y varios procesos de fondo son opcionales. Los procesos de fondo opcionales pueden o no ser utilizados dependiendo de cuales son las características opcionales de Oracle que se están utilizando en la base de datos.



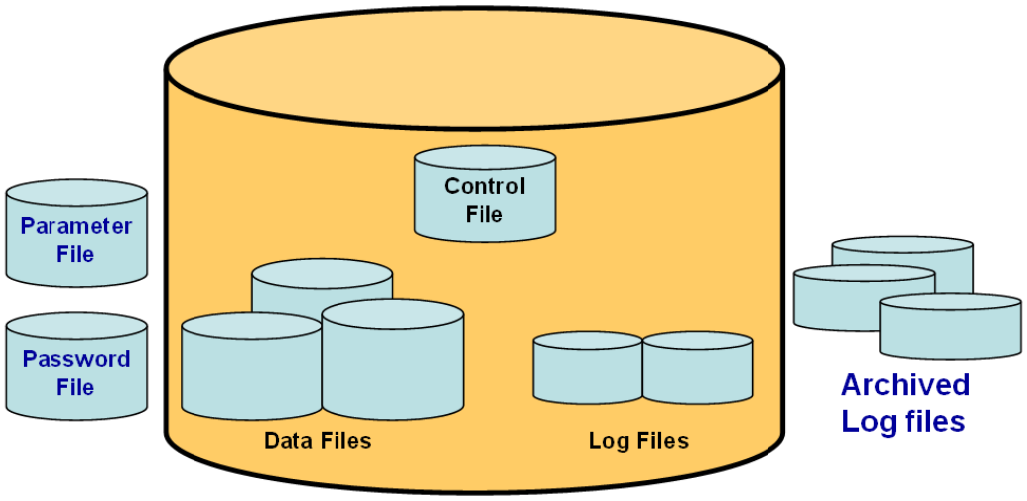
Procesos de Fondo Obligatorios

El siguiente cuadro muestra los procesos de fondo obligatorios:

Process Name	Operating System Process	Description
System Monitor	SMON	Es el proceso encargado de recuperar la instancia y abrir la base de datos en caso de fallas en la instancia.
Process Monitor	PMON	Este proceso se encarga de liberar los recursos del servidor asignados a la conexión de un usuario en caso que la conexión a través del proceso servidor asignado falle.
Database Writer	DBWn*	Escribe los bloques modificados de la base de datos del Database Buffer Cache del SGA a los archivos de datos (datafiles) en disco.
Log Writer	LGWR	Escribe la información de recuperación de transacciones del Redo Log Buffer del SGA a los archivos Redo Log en línea en disco.
Checkpoint	CKPT	Actualiza los archivos de la base de datos cada vez que ocurre un evento Checkpoint.

* La n en cualquier nombre de proceso del sistema operativo significa que más de uno de estos procesos pueden estar en funcionamiento. En estos casos, la n se substituye por un valor numérico. Por ejemplo, si cuatro procesos Writer de la base de datos están funcionando, sus nombres de proceso en el nivel del sistema operativo son DBW0, DBW1, DBW2, y DBW3.

La Base de Datos de Oracle



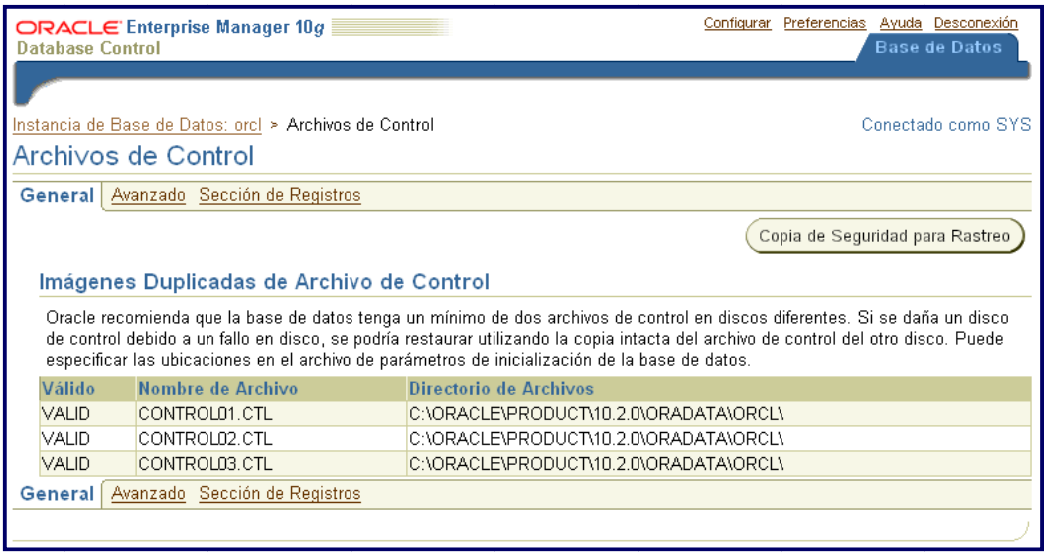
Una instancia es una estructura temporal en memoria, pero la base de datos de Oracle se compone de un sistema de archivos físicos que residen en los discos del servidor. Estos archivos se llaman control files, datafiles, y redo logs. Los archivos físicos adicionales que se asocian a la base de datos de Oracle, pero que técnicamente no son parte de la base de datos, son el password file, el PFILE y SPFILE, y los archived redo log files. El siguiente cuadro resume el papel de cada uno de estos juegos de los archivos en la arquitectura de la base de datos.

Tipo de Archivo	Información Contenida en el Archivo
Control	Ubicación de los otros archivos, nombre de la base de datos, tamaño del bloque de la base de datos, juego de caracteres de la base de datos, e información de recuperación física. Estos archivos se requieren para abrir la base de datos.
Datafile	Toda la data de las aplicaciones y la metadata interna.
Redo Log	Registra todos los cambios hechos en la base de datos; usado para propósitos de recuperación.
Parameter (PFILE o SPFILEE)	Parámetros de configuración del SGA, características opcionales de Oracle, y procesos de background.
Archived Log	Copia el contenido de los redo log en línea, usado para propósitos de recuperación.
Password	Es un archivo opcional usado para almacenar el nombre de los usuarios a los que se les ha otorgado el privilegio SYSDBA y/o SYSOPER.

Podemos consultar los archivos de control ejecutando la siguiente consulta:

```
SQL> select name from v$controlfile;    [Enter]

NAME
-------------------------------------------------------
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\CONTROL01.CTL
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\CONTROL02.CTL
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\CONTROL03.CTL
```

También podemos realizar la misma consulta en EM Database Control:



Podemos consultar los archivos de datos ejecutando la siguiente consulta:

```
SQL> select name from v$datafile;    [Enter]

NAME
---------------------------------------------------------
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\SYSTEM01.DBF
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\UNDOTBS01.DBF
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\SYSAUX01.DBF
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\USERS01.DBF
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\EXAMPLE01.DBF
```

También podemos realizar la consulta de sobre loa *datafiles* en EM Database Control:

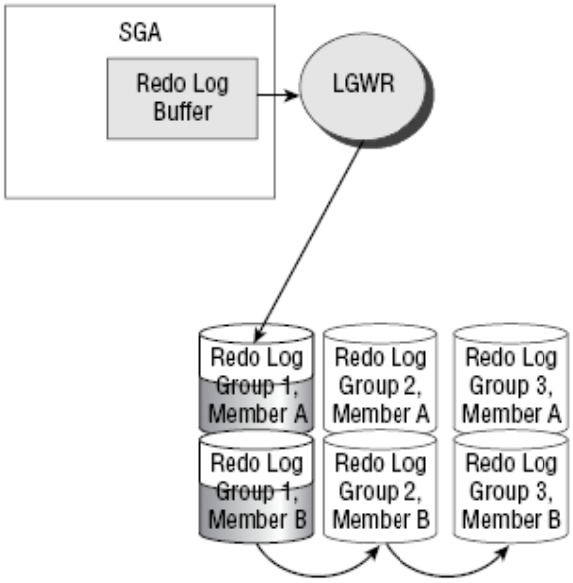
Para consultar los grupos Redo Logs ejecute la siguiente consulta:

```
SQL> select group#, members from v$log;   [Enter]

    GROUP#    MEMBERS
---------- ----------
         1          1
         2          1
         3          1
```

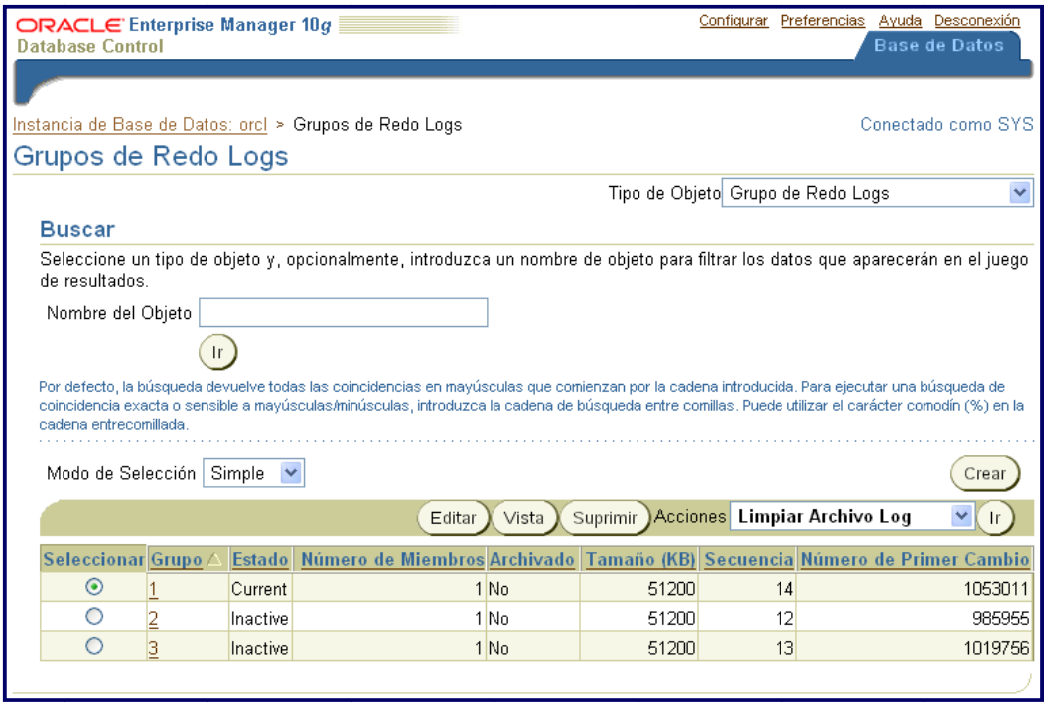
Esta consulta nos indica que tenemos tres grupos, y cada grupo tiene un solo miembro, o sea, un archivo. Para consultar los archivos miembros de cada grupo ejecute la siguiente consulta:

```
SQL> select group#, member from v$logfile order by 1;     [Enter]

    GROUP# MEMBER
---------- --------------------------------------------------
         1 C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\REDO01.LOG
         2 C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\REDO02.LOG
         3 C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\REDO03.LOG
```

El siguiente grafico ilustra la forma como trabajan los archivos redo log:



También podemos consultar los redo logs en EM Database Control:



En esta ventana para consultar los archivos miembros de un grupo, hacemos clic en el número que corresponde al grupo a consultar, en la columna ***Grupo*.**

Lección 02

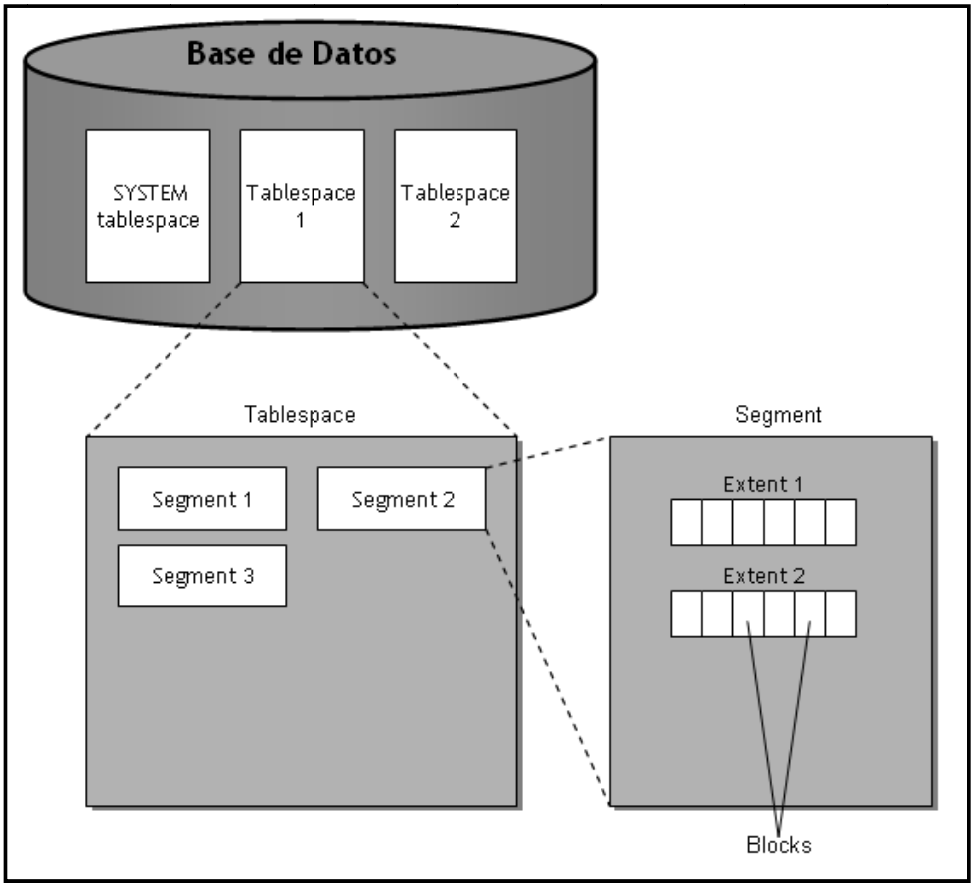
Introducción a las Estructuras de Almacenamiento

Contenido

- Estructuras Lógicas 14
- Estructuras Físicas 16
- Esquemas de Base de Datos 18

Estructuras Lógicas

Oracle divide la base de datos en unidades más pequeñas para manejar, almacenar, y recuperar los datos eficientemente. El siguiente grafico muestra la estructura lógica de la base de datos.



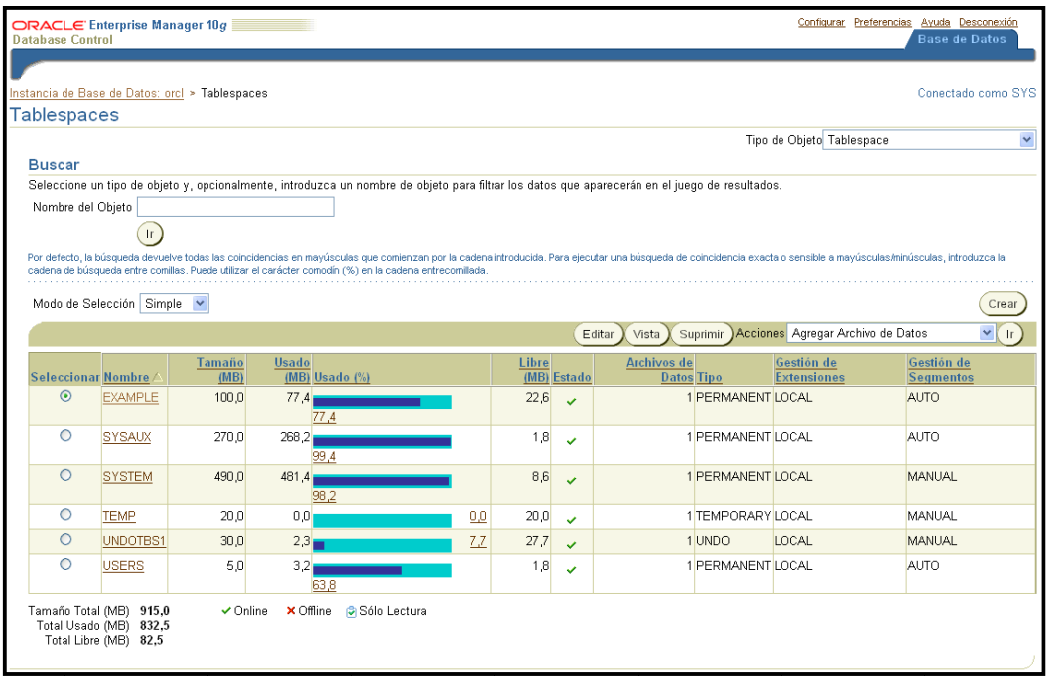
La siguiente consulta muestra los tablespaces que se crean junto con la base de datos:

```
SQL> select name from v$tablespace;     [Enter]

NAME
------------------------------
SYSTEM
UNDOTBS1
SYSAUX
USERS
TEMP
EXAMPLE

6 filas seleccionadas.
```

A continuación tenemos una breve descripción de los tablespaces que se crean junto con la base de datos.

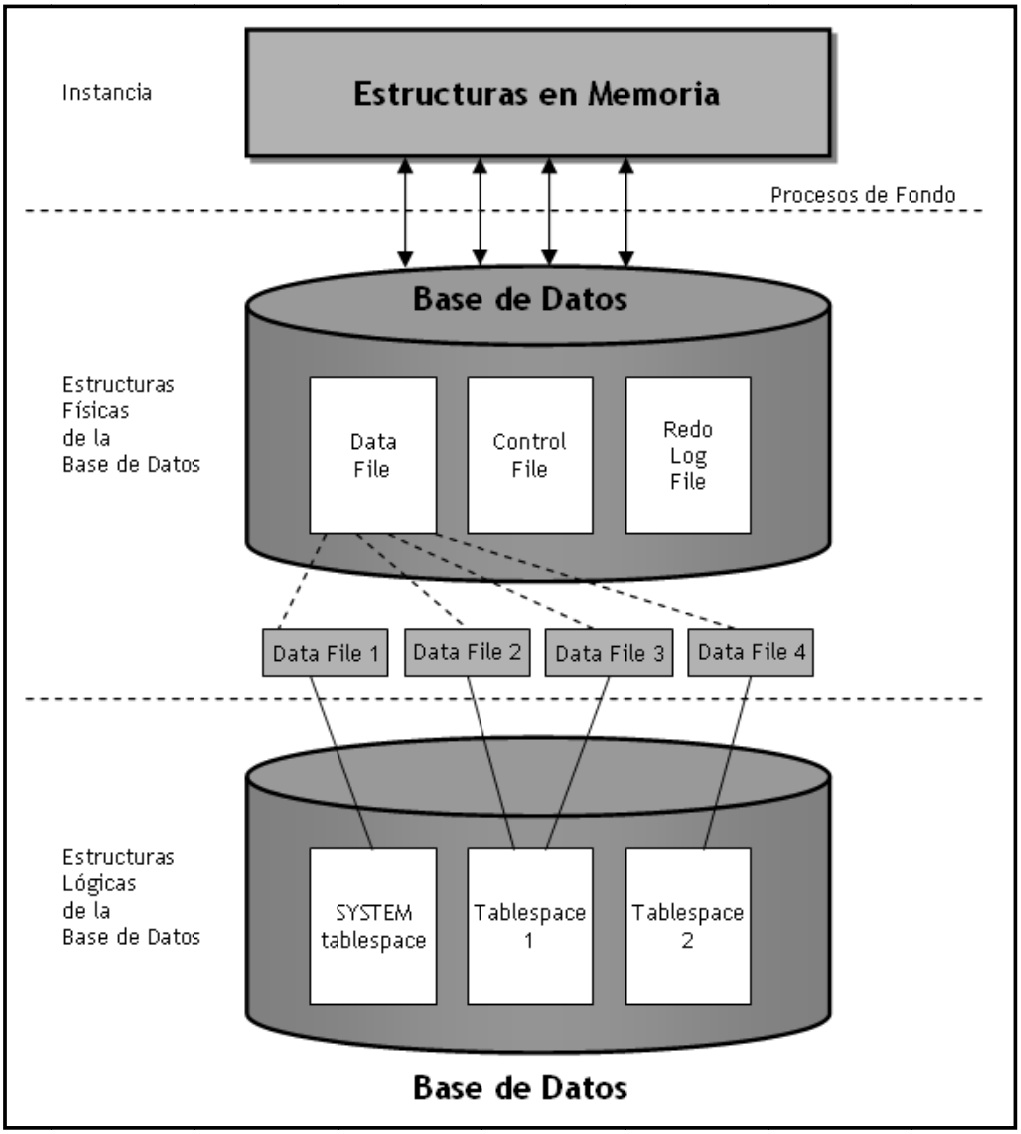
Tablaspace	Descripción
system	El tablespace SYSTEM se utiliza para el diccionario de los datos y código PL/SQL, y no debe ser usado para almacenar objetos de otros esquemas.
undotbs1	Almacena información de las transacciones para los propósitos consistencia y recuperación.
sysaux	El tablespace SYSAUX es nuevo a Oracle10g y almacena segmentos usados para opciones de la base de datos, tales como Automatic Workload Repository (repositorio que guarda periódicamente información relativa a la actividad del sistema), Online Analytical Processing (OLAP), y Spatial.
users	Utilizado como tablespace por defecto para usuarios de la base de datos.
temp	Usado para operaciones de grandes ordenamiento.

También podemos consultar los tablespaces utilizando la herramienta EM Database Control, tal como se ilustra a continuación:



Estructuras Físicas

El siguiente grafico muestra la relación que existe entre las estructuras físicas y las estructuras lógicas de almacenamiento.



Cada tablespace tiene asignado por lo menos un datafile. Algunos tablespace, por razones de administración y rendimiento pueden estar conformados por varios datafile. La siguiente consulta nos muestra los datafiles por tablespace.

```
SQL> select tablespace_name, file_name from dba_data_files;  [Enter]

TABLESPACE_NAME      FILE_NAME
-------------------- ----------------------------------------------------
USERS                C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\USERS01.DBF
SYSAUX               C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\SYSAUX01.DBF
UNDOTBS1             C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\UNDOTBS01.DBF
SYSTEM               C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\SYSTEM01.DBF
EXAMPLE              C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\EXAMPLE01.DBF
```

Para consultar los archivos asociados con los tablespaces temporales, la consulta es la siguiente:

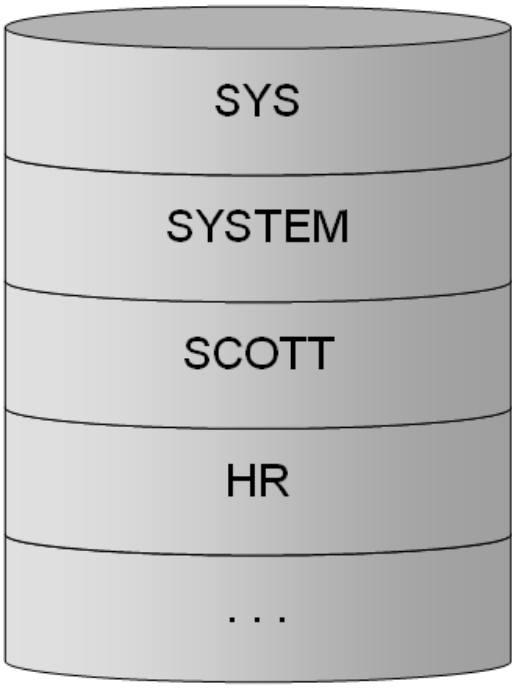
```
SQL> select tablespace_name, file_name from dba_temp_files;  [Enter]

TABLESPACE_NAME      FILE_NAME
-------------------- ----------------------------------------------------
TEMP                 C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\TEMP01.DBF
```

También podemos usar el EM para consultar información sobre loa datafiles.

Esquemas de Base de Datos

Un esquema es una colección de objetos de la base de datos pertenecientes un usuario específico de la base de datos. En una base de datos de Oracle, el esquema tiene el mismo nombre que el usuario de la base de datos, así que los dos términos son sinónimos.

Por lo tanto, podemos ver la base de datos como un conjunto de esquemas, tal como se ilustra en el siguiente grafico:



Para consultar información de los usuarios, podemos ejecutar la siguiente consulta:

```
SQL> select username, default_tablespace, temporary_tablespace
  2  from dba_users
  3  where username in ('SYS', 'SYSTEM', 'SCOTT', 'HR');    [Enter]

USERNAME             DEFAULT_TABLESPACE   TEMPORARY_TABLESPACE
-------------------- -------------------- --------------------
SCOTT                USERS                TEMP
HR                   USERS                TEMP
SYS                  SYSTEM               TEMP
SYSTEM               SYSTEM               TEMP
```

Podemos apreciar los tablespace por defecto de cada usuario y el tablespace temporal asignado a cada usuario.

El usuario SYS es el dueño de todas las tablas base y vistas de acceso a la data del diccionario de datos. Ningún otro usuario debe alterar nunca (UPDATE, DELETE, o INSERT) cualquier fila u objetos del esquema SYS, porque tal actividad puede comprometer la integridad de los datos. El administrador de la base de datos debe tener estricto control de esta cuenta.

También podemos utilizar el EM para consultar información sobre los usuarios de la base de datos.

Página en Blanco

Lección 03

El Diccionario de Datos

Contenido

- Introducción al Diccionario de Datos 22
- Componentes Privados 23
  - Tablas X$ 23
  - Vistas Dinámicas (V_$) 25
  - Tablas $ 27
- Componentes Públicos 28

Introducción al Diccionario de Datos

Una de las partes más importantes de una base de datos Oracle es su diccionario de datos, que es un sistema de solo lectura de tablas que proporciona la información sobre la base de datos. Un diccionario de los datos contiene:

- La definición de los objetos de todos los esquemas (tablas, vistas, índices, sinónimos, secuencias, funciones, procedimientos, etc.).
- Cuanto espacio ha sido asignado a cada esquema, y cuanto ha sido utilizado.
- Los valores por defecto de las columnas.
- Información de restricciones de integridad.
- Los nombres de los usuarios de Oracle.
- Privilegios y roles otorgados a cada usuario.
- Información de auditoria, como por ejemplo, que usuarios han iniciado sesión y cuanto tiempo estuvieron activas.
- Etc.

El diccionario de datos es una estructura de tablas y vistas, similar a otros datos de la base de datos. Todas las tablas y vistas del diccionario de datos están almacenadas en el tablespace SYSTEM.

No solo el diccionario de datos es todo en cada base de datos Oracle, esta es también una herramienta importante para todos los usuarios, para los diseñadores de aplicaciones de usuarios finales y para los administradores de base de datos. Utilice las sentencias SQL para realizar consultas al diccionario de los datos. Debido a que el diccionario de datos es de solo lectura, solo se puede consultar (instrucciones SELECT) a las tablas y vistas.

Por su complejidad y extensión, el diccionario de datos se ha dividido, en dos niveles de objetos: privado y público.

Componentes Privados

Se denomina componente privado a todos aquellos grupos de objetos (tablas y/o vistas) que conforman el Diccionario de Datos que por su naturaleza sólo deberían ser acezados y/o utilizados por usuarios que normalmente son Administradores de la Base de Datos. Este componente esta, a su vez integrado por los siguientes grupos de objetos:

- Tablas X$
- Vistas Dinámicas (V_$)
- Tablas $

Tablas X$

Son estructuras en memoria presentadas como tablas y sirven como base para las vistas V$, normalmente estas tablas no se encuentran documentadas. Para ver su contenido un DBA tiene que hacerlo como usuario SYS. El problema que se tiene con estas tablas, es que, su nombre y el nombre de sus columnas no son descriptivos, por lo tanto resulta difícil darse cuenta del significado de su contenido. Es de suponer que por su importancia, Oracle haya decidido usar esas convenciones para nombrarlas.

No hay documentación acerca de las tablas X$ prevista por Oracle Corporation; sin embargo es posible mediante la revisión y el análisis del contenido de las mismas deducir su utilidad en la creación de las vistas V$.

Las tablas X$ son creadas en memoria al momento del inicio de la instancia y mantenidas en tiempo real guardando información de la actividad de la Base de Datos desde el comienzo de la instancia.

El único comando permitido sobre estas tablas es el SELECT, ocurriendo un error si se intenta ejecutar cualquier tipo de instrucción diferente al SELECT.

```
SQL> conn / as sysdba   [Enter]
Conectado.

SQL> select banner from x$version;   [Enter]

BANNER
----------------------------------------------------------------
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Prod
PL/SQL Release 10.2.0.1.0 - Production
CORE    10.2.0.1.0      Production
TNS for 32-bit Windows: Version 10.2.0.1.0 - Production
NLSRTL Version 10.2.0.1.0 – Production
```

No se puede otorgar el privilegio de SELECT sobre esas tablas a ningún usuario.

```
SQL> conn / as sysdba   [Enter]
Conectado.

SQL> grant select on x$option to scott;   [Enter]
grant select on x$option to scott
                *
ERROR en línea 1:
ORA-02030: sólo se puede seleccionar de tablas/vistas fijas
```

La única operación disponible para permitir acceso sobre estas tablas a usuarios diferentes a SYS es la creación de vistas sobre ellas y con esas vistas permitir el acceso, de la siguiente forma:

```
SQL> conn / as sysdba     [Enter]
Conectado.

SQL> create view prueba$option as select * from x$option;    [Enter]

Vista creada.

SQL> create  public synonym x$option for prueba$option;     [Enter]

Sinónimo creado.

SQL> grant select on prueba$option to public;               [Enter]

Concesión terminada correctamente.

SQL> conn scott/tigre           [Enter]
Conectado.

SQL> select * from x$option;    [Enter]

ADDR           INDX    INST_ID PARAMETER                        VALUE
-------- ---------- ---------- -------------------------------- ----------
04B79704          0          1 Partitioning                     TRUE
04B79704          1          1 Objects                          TRUE
04B79704          2          1 Real Application Clusters        FALSE
04B79704          3          1 Advanced replication             TRUE
04B79704          4          1 Bit-mapped indexes               TRUE
04B79704          5          1 Connection multiplexing          TRUE
04B79704          6          1 Connection pooling               TRUE
04B79704          7          1 Database queuing                 TRUE
04B79704          8          1 Incremental backup and recovery  TRUE
04B79704          9          1 Instead-of triggers              TRUE
04B79704         10          1 Parallel backup and recovery     TRUE
. . .
. . .
```

Las tablas X$ no pueden ser modificadas de ninguna manera y su mantenimiento lo desarrolla Oracle de manera automática. La forma fácil de ver su información es mediante el uso de las vistas V$,

```
SQL> conn / as sysdba          [Enter]
Conectado.

SQL> select * from x$ksmsd;    [Enter]

ADDR           INDX    INST_ID KSMSDNAM                KSMSDVAL
-------- ---------- ---------- -------------------- ----------
03BF0BB0          0          1 Fixed Size              1247528
03BF0BD0          1          1 Variable Size         109053656
03BF0BF0          2          1 Database Buffers      171966464
03BF0C10          3          1 Redo Buffers            7139328

SQL> select * from v$sga;      [Enter]

NAME                      VALUE
-------------------- ----------
Fixed Size              1247528
Variable Size         109053656
Database Buffers      171966464
Redo Buffers            7139328
```

Vistas Dinámicas (V_$)

Oracle contiene un conjunto de vistas fundamentales que son mantenidas por el servidor de base de datos y accesible por el usuario administrador de la base de datos SYS. Estas vistas se llaman "dynamic performance views" porque son continuamente actualizadas mientras la base de datos está abierta y en uso, y su contenido se relaciona sobre todo con su funcionamiento.

Aunque estas vistas parecen ser tablas de la base de datos, no son. Estas vistas proporcionan datos en las estructuras internas del disco y las estructuras de memoria. Podemos realizar consultar de estas vistas, pero nunca podemos actualizarlas ó alterarlas.

El código contenido en "catalog.sql" contiene las definiciones de estas vistas y sus correspondientes sinónimos públicos.

Las vistas dinámicas reales son identificadas por el prefijo V_$. Los sinónimos públicos para estas vistas tienen el prefijo V$. Los administradores de la base de datos y otros usuarios solo deben tener acceso a las vistas V$, y no a las vistas V_$.

Las vistas dinámicas son utilizadas por Oracle Enterprise Manager, que es la interfaz principal para acceder a información sobre funcionamiento del sistema. Después de que la instancia es iniciada, las vistas V$ que leen información de la memoria son accesibles. Las vistas que leen los datos del disco requieren que la base de datos esté montada, y algunas requieren que la base de datos esté abierta.

```
SQL> conn / as sysdba        [Enter]
Conectado.

SQL> shutdown immediate      [Enter]
Base de datos cerrada.
Base de datos desmontada.
Instancia ORACLE cerrada.

SQL> select * from v$sga;    [Enter]
select * from v$sga
*
ERROR en línea 1:
ORA-01034: ORACLE not available

SQL> startup                 [Enter]
Instancia ORACLE iniciada.

Total System Global Area  289406976 bytes
Fixed Size                  1247528 bytes
Variable Size             109053656 bytes
Database Buffers          171966464 bytes
Redo Buffers                7139328 bytes
Base de datos montada.
Base de datos abierta.

SQL> select * from v$sga;    [Enter]

NAME                      VALUE
------------------- ----------
Fixed Size              1247528
Variable Size         109053656
Database Buffers      171966464
Redo Buffers            7139328
```

Tablas $

El otro grupo de objetos dentro del Diccionario de Datos, es el conjunto de las tablas $. Estas tablas tienen nombres más comprensibles que las anteriores tablas **X$**, tales como **COL$**, **TS$**, **USER$** o **TAB$**. Este conjunto de tablas es creado por la ejecución del archivo **sql.bsq**. De la misma manera que con las tablas **X$**, se recomienda que solo sean usuarios administradores (DBA) los que accedan estas tablas.

```
SQL> conn / as sysdba        [Enter]
Conectado.

SQL> select name from ts$;   [Enter]

NAME
------------------------------
EXAMPLE
SYSAUX
SYSTEM
TEMP
UNDOTBS1
UNDOTBS2
USERS

7 filas seleccionadas.
```

Componentes Públicos

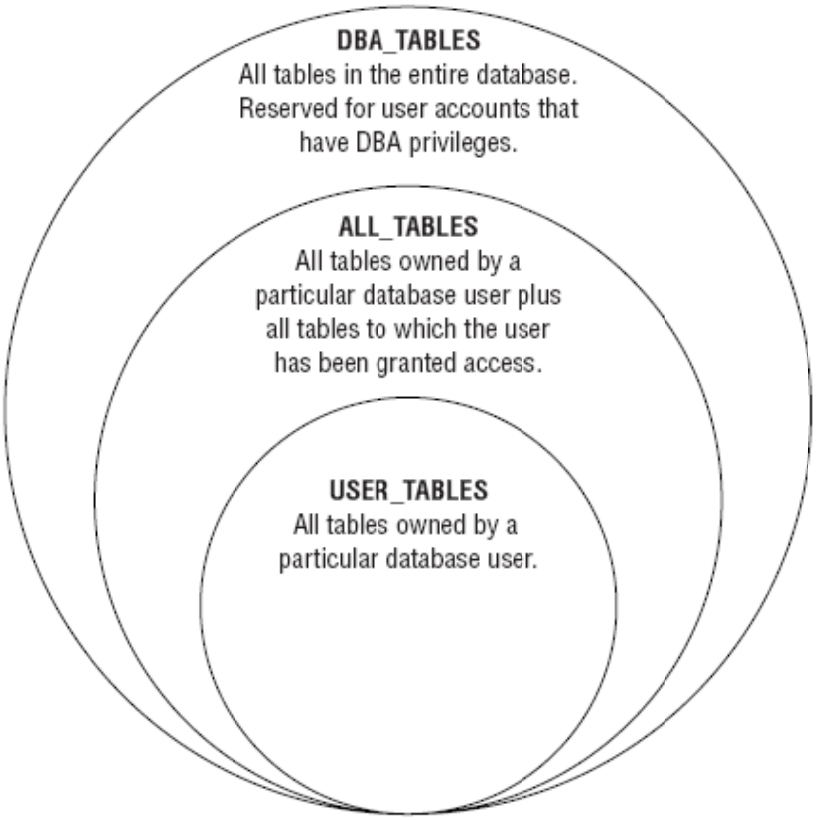
Dentro de este grupo de objetos Oracle están todas las vistas que sobre el componente privado son creadas por los diferentes archivos cat*.sql, siendo el más conocido el catalog.sql, que crea vistas asociadas a las tablas X$, $ y a las vistas V$ o V_$.

Dependiendo de qué características están instaladas y configuradas, una base de datos de Oracle 10g puede contener más de 1,300 vistas del diccionario de los datos. Las vistas del diccionario de datos tienen nombres que comienzan con el DBA_, ALL_, y USER_.

Estas vistas son la manera más amigable de observar la información contenida en el Diccionario de Datos. Son el grupo de objetos Oracle más conocido, como alternativa de seguridad en el manejo del Diccionario de Datos por parte de los diferentes usuarios involucrados en el manejo de la base de datos.

La diferencia entre las vistas DBA_, ALL_, USER_ se puede ilustrar usando como ejemplo la vista del diccionario de datos DBA_TABLES. La vista DBA_TABLES muestra información sobre todas las tablas en la base de datos. La vista ALL_TABLES, a pesar de su nombre, muestra solamente a tablas que un usuario particular de la base de datos posee o tiene acceso. Por ejemplo, si inicia sesión en la base de datos como usuario SCOTT, la vista ALL_TABLES mostrará todas las tablas pertenecientes al usuario SCOTT y las tablas sobre las cuales a SCOTT otros usuarios le han otorgado acceso. La vista USER_TABLES muestra solamente los objetos pertenecientes a un usuario. Si el usuario SCOTT consulta la vista USER_TABLES, sólo las tablas que él posee se mostraran en el resultado. El siguiente gráfico muestra una representación gráfica de la relación entre las vistas **DBA**_, **ALL**_, y **USER**_.

A comparison of data dictionary views



Las vistas **DBA_** proporcionan la información más amplia de la metadata y son generalmente las vistas del diccionario de datos usadas por los DBAs. La siguiente tabla muestra una descripción sobre los tipos de vistas.

Prefijo	Descripción
USER_	Muestran los objetos creados por el usuario que hace la consulta.
ALL_	Muestran los objetos a los cuales cada usuario tiene acceso, es decir todos los que él ha creado como también aquellos sobre los cuales tiene permitido el acceso y no le pertenecen.
DBA_	Muestran todos los objetos creados a nivel de la Base de Datos. Disponibles para usuarios administradores.

Como las vistas ALL_ y DBA_ muestran objetos pertenecientes a diferentes usuarios, en ellas se debe estipular el propietario de cada objeto referenciado.

Normalmente toda vista DBA_, tiene su contraparte USER_ y ALL_, con contadas excepciones que no ameritan la presencia de esas vistas, como por ejemplo: dba_tablespaces, **dba_data_files**, **dba_rollback_segs**, etc.

```
SQL> conn scott/tigre      [Enter]
Conectado.

SQL> select table_name from user_tables;     [Enter]

TABLE_NAME
------------------------------
DEPT
EMP
BONUS
SALGRADE
```

Lección 04

Usuarios Administradores

Contenido

- Administrador de la Seguridad y Privilegios de la Base de Datos 32
  - Cuentas del Sistema Operativo Administradoras de la Base de Datos 32
  - Nombres de Usuarios Administradores de la Base de Datos 32
- Autenticación del Administrador de Base de Datos 34
  - Privilegios Administrativos 34
  - Seleccionar un Método de Autenticación 37
  - Usando Autenticación del Sistema Operativo (OS) 39
  - Usando la Autenticación del Archivo de contraseñas 42
- Creación y Manteniendo el Archivo de Contraseñas 44
  - Usando ORAPWD 44
  - Estableciendo el valor de REMOTE_LOGIN_PASSWORDFILE 46
  - Adicionando Usuarios al Archivo de Contraseñas 47
  - Mantenimiento del Archivo de Contraseñas 49

Administrador de la Seguridad y Privilegios de la Base de Datos

Para llevar a cabo las tareas administrativas, un DBA de Oracle necesita privilegios específicos dentro de la base de datos y posiblemente en el sistema operativo del servidor en el cual la base de datos esta funcionando. El acceso a una cuenta de administrador de la base de datos debe ser celosamente controlado.

Cuentas del Sistema Operativo Administradoras de la Base de Datos

Para realizar muchas de las tareas administrativas de una base de datos, es necesario poder ejecutar comandos del sistema operativo. Dependiendo del sistema operativo en que se ejecuta Oracle, puede ser que se necesite una cuenta del sistema operativo o una identificación para acceder al sistema operativo. Si es así, la cuenta del sistema operativo pude requerir más privilegios o derechas de acceso que muchos usuarios de la base de datos requieren (por ejemplo, realizar la instalación del software de Oracle). Aunque no es necesario guardar los archivos de Oracle en nuestra cuenta personal, si debemos tener acceso a ellos.

Nombres de Usuarios Administradores de la Base de Datos

Dos cuentas de usuario son creadas automáticamente con la base de datos, como se ilustra en el siguiente cuadro.

- SYS
- SYSTEM

Durante la instalación de la base de datos Oracle se establecen las contraseñas de los usuarios SYS y SYSTEM.

Se sugiere también crear por lo menos un usuario adicional, y concederle el rol DBA, para utilizarlo al realizar tareas administrativas diarias. Se recomienda no utilizar las cuentas SYS y SYSTEM para estos propósitos.

SYS

Cuando se crea una base de datos, el usuario SYS se crea automáticamente y se le asigna el rol de **DBA**.

Todas las tablas y vistas del diccionario de datos son almacenadas en el esquema SYS. Estas tablas y vistas son críticas para la operación de la base de datos Oracle. Para mantener la integridad del diccionario de datos, las tablas en el esquema SYS son manipuladas solamente por Oracle. Nunca deben ser modificadas por cualquier usuario o administrador de la base de datos, y nadie debe crear cualquier tabla en el esquema del usuario SYS. (Sin embargo, se pueden cambiar los parámetros de almacenamiento del diccionario de datos en caso de que sea necesario.)

Asegúrese de que la mayoría de los usuarios de la base de datos nunca puedan conectarse con la cuenta SYS.

SYSTEM

Cuando se crea una base de datos, el usuario SYSTEM se crea automáticamente y se le asigna el rol de **DBA**.

El usuario SYSTEM es usado para crear tablas y vistas adicionales para mostrar información administrativa, y tablas y vistas internas usadas por varias opciones y herramientas de Oracle. Nunca se debe crear tablas en el esquema SYSTEM de interés de usuarios individuales.

Rol DBA

Un rol predefinido, de nombre DBA, se crea automáticamente con cada base de datos de Oracle. Este rol contiene la mayoría de los privilegios del sistema de base de datos. Por lo tanto, es de muy gran alcance y debe ser concedido solamente a los administradores completamente funcionales de la base de datos.

Nota:

El papel del DBA no incluye los privilegios del sistema SYSDBA ó SYSOPER. Éstos son los privilegios administrativos especiales que permiten a un administrador realizar tareas básicas de la administración de la base de datos, tales como crear la base de datos, el arranque y la parada de la instancia.

Autenticación del Administrador de Base de Datos

Como **DBA**, realizamos a menudo operaciones especiales tales como Iniciar y Parar la base de datos. Porque solamente un DBA debe realizar estas operaciones, los usernames del administrador de la base de datos requieren un esquema seguro de autentificación.

Privilegios Administrativos

Los privilegios administrativos que se requieren para que un administrador pueda realizar operaciones básicas de la base de datos se conceden con dos privilegios especiales del sistema, SYSDBA y SYSOPER. Se debe tener uno de estos dos privilegios, dependiendo del nivel de autorización que se requiera.

Nota

Los privilegios del sistema SYSDBA y SYSOPER permiten el acceso a la instancia de la base de datos aun cuando la base de datos no está abierta.

El control de estos privilegios está totalmente fuera de la base de datos.

SYSDBA y SYSOPER

Las siguientes son las operaciones que están autorizadas para los privilegios del sistema **SYSDBA** y de **SYSOPER**:

Privilegio del Sistema	Operación Autorizada
SYSDBA	▪ Ejecutar operaciones de STARTUP y SHUTDOWN ▪ ALTER DATABASE: OPEN, MOUNT, BACKUP, o cambiar el conjunto de caracteres ▪ CREATE DATABASE ▪ ARCHIVELOG y RECOVER ▪ CREATE SPFILE ▪ Incluye el privilegio RESTRICTED SESSION Este privilegio del sistema permite conectarnos como usuario SYS.
SYSOPER	▪ Ejecutar operaciones de STARTUP y SHUTDOWN ▪ ALTER DATABASE: OPEN, MOUNT, BACKUP, o cambiar el conjunto de caracteres ▪ ARCHIVELOG y RECOVER ▪ CREATE SPFILE ▪ Incluye el privilegio RESTRICTED SESSION Este privilegio permite a un usuario realizar tareas operacionales básicas, pero sin la capacidad de ver datos del otros usuario.

El contexto en el cual se autorizan a utilizar estos privilegios depende del método de autentificación que se elige utilizar.

Cuando nos conectamos con privilegios de SYSDBA o SYSOPER, nos estamos conectando con un esquema por defecto, no con el esquema que se asocia generalmente a su username. Para SYSDBA este esquema es SYS; para SYSOPER el esquema es PUBLIC.

Conectándonos con Privilegios Administrativos: Ejemplo

En este ejemplo se comprueba que a un usuario se le asigna el esquema SYS cuando se conecta con el privilegio del sistema SYSDBA.

Paso 1

Como primer paso, crearemos una tabla como usuario scott, y luego insertaremos una fila.

```
SQL> conn scott/tiger      [Enter]

Conectado.

SQL> show user             [Enter]

USER es "SCOTT"

SQL> create table test( dato varchar2(30) );           [Enter]

Tabla creada.

SQL> insert into test values( 'Gustavo Coronel' );     [Enter]

1 fila creada.

SQL> select * from test;   [Enter]

DATO
------------------------------
Gustavo Coronel
```

Paso 2

Como segundo paso, otorgaremos a scott el privilegio SYSDBA.

```
SQL> conn sys/admin as sysdba     [Enter]

Conectado.

SQL> grant sysdba to scott;       [Enter]

Concesión terminada correctamente.
```

Paso 3

Ahora, comprobaremos que cuando *scott* se conecta utilizando el privilegio SYSDBA, el esquema por defecto ya no es *scott*, sino *sys*.

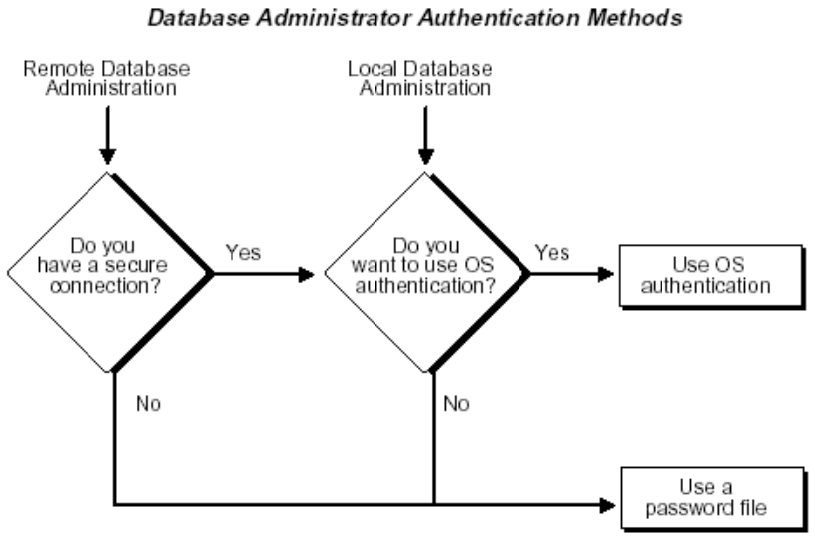
```
SQL> conn scott/tiger as sysdba    [Enter]

Conectado.

SQL> show user                     [Enter]

USER es "SYS"

SQL> select * from test;           [Enter]

select * from test
              *
ERROR en línea 1:
ORA-00942: la tabla o vista no existe
```

Seleccionar un Método de Autenticación

Los métodos siguientes están disponibles para autenticar administradores de la base de datos:

- Autenticación por el Sistema Operativo (OS)
- Autenticación mediante Archivo de Contraseñas (Password Files)

Su opción es afectada si administramos la base de datos localmente en la misma máquina donde reside la base de datos, o si administramos diversas bases de datos de un solo cliente remoto. La siguiente figura ilustra las opciones que tenemos para los esquemas de autentificación del administrador de la base de datos.



Si estamos realizando la administración remota de la base de datos, debemos consultar la documentación de red de Oracle para determinar si estamos utilizando una conexión segura. La mayoría de los protocolos comunes de conexión, tales como TCP/IP y DECnet, no son seguros.

Conexiones Remotas No-Seguras

Para conectar con Oracle como usuario privilegiado sobre una conexión no-segura, debemos estar autenticados por un archivo de contraseñas. Al usar la autentificación del archivo de contraseña, la base de datos utiliza un archivo de contraseña para no perder la identificación de los usernames de la base de datos a los se han concedido el privilegio de sistema **SYSDBA** o **SYSOPER**.

Conexiones Locales y Conexiones Remotas Seguras

Para conectarnos con Oracle como usuario privilegiado sobre una conexión local o una conexión remota segura, tenemos las siguientes opciones:

- Podemos conectarnos y ser autenticados por un archivo de contraseñas, siempre y cuando la base de datos tenga un archivo de contraseñas y tengamos el privilegio del sistema SYSDBA o SYSOPER.
- Si el servidor no está utilizando un archivo de contraseñas, o si no tenemos el privilegio SYSDBA o SYSOPER, y por lo tanto no estamos en el archivo de contraseñas, podemos utilizar la autentificación del OS. En la mayoría de los sistemas operativos, la autentificación del OS para los administradores de la base de datos implica colocar el username del OS del administrador de la base de datos en un grupo especial, generalmente designado con el nombre OSDBA.

Usando Autenticación del Sistema Operativo (OS)

OSDBA y OSOPER

Son dos grupos de usuarios especiales del sistema operativo para controlar las conexiones de los usuarios administradores de la base de datos usando la autenticación del OS. Estos grupos son genéricamente conocidos como OSDBA y OSOPER. Los grupos son creados y los nombres específicos asignados como parte del proceso de instalación de la base de datos. Los nombres específicos dependen del sistema operativo y se ilustran en el siguiente cuadro:

Grupo del Sistema Operativo	UNIX	Windows
OSDBA	dba	ORA_DBA
OSOPER	oper	ORA_OPER

Los nombres por defecto asumidos por el instalador universal de Oracle pueden ser eliminados. Podemos crear los grupos OSDBA y OSOPER específicos según el sistema operativo.

A continuación hacemos una descripción cómo un usuario que pertenece al grupo OSDBA u OSOPER afecta la conexión con Oracle:

- Si es un miembro del grupo OSDBA, y especifica AS SYSDBA cuando se conecta con la base de datos, lo estará haciendo con el privilegio de sistema SYSDBA.
- Si es un miembro del grupo OSOPER, y especifica AS SYSOPER cuando se conecta con la base de datos, lo estará haciendo con el privilegio de sistema SYSOPER.
- Si no es miembro de estos grupos del sistema operativo e intenta conectarse como SYSDBA o SYSOPER, el comando CONNECT fallará.

Preparando la Autenticación del SO

Para habilitar la autentificación de un usuario administrador que usa el sistema operativo debemos hacer el siguiente:

1. Crear un usuario en el sistema operativo.
2. Agregar el usuario a los grupos OSDBA u OSOPER definidos en el sistema operativo.

Conectándonos usando la Autenticación del SO

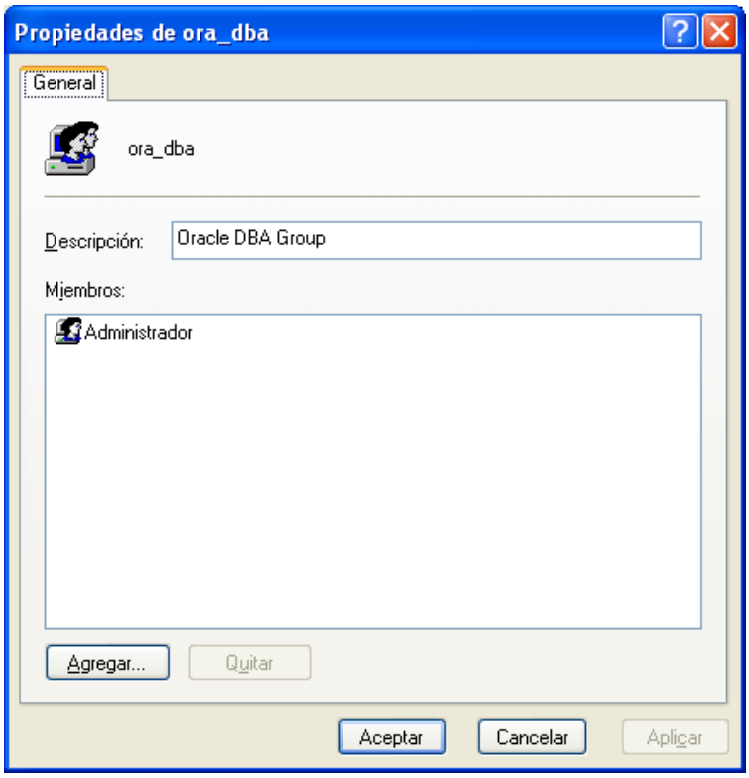
Un usuario puede ser autenticado, habilitándolo como usuario administrativo, y conectarse con una base de datos local con uno de los comandos siguientes del SQL*Plus:

```
CONNECT / AS SYSDBA

CONNECT / AS SYSOPER
```

Para una conexión remota con la base de datos sobre una conexión segura, el usuario debe también especificar el nombre del servicio de red de la base de datos remota:

```
CONNECT /@net_service_name AS SYSDBA

CONNECT /@net_service_name AS SYSOPER
```

El siguiente gráfico muestra los miembros del grupo *ORA_DBA*.



A continuación tenemos un ejemplo de como se realiza la conexión utilizando autenticación del sistema operativo.

```
SQL> conn / as sysdba      [Enter]

Conectado.

SQL> show user             [Enter]

USER es "SYS"
```

Ejercicio Experimental

El siguiente ejercicio debe desarrollarlo usted.

1. Crear el usuario del SO de nombre *sergio*.
2. Agregue el usuario *sergio* al grupo Administradores.
3. Cierre todas las aplicaciones, cierre la sesión actual, e inicie sesión con el usuario *sergio*.
4. Intente conectarse a Oracle como SYSDBA, comente el resultado con su compañero.
5. Agregue el usuario *sergio* al grupo ORA_DBA.
6. Intente nuevamente conectarse a Oracle como SYSDBA, comente el resultado con su compañero.

Usando la Autenticación del Archivo de contraseñas

Esta sección describe cómo autenticar a un usuario administrativo usando la autentificación del archivo de la contraseña. Le sugiero retirar a todos los miembros de los grupos del sistema operativo ORA_DBA y ORA_OPER.

Preparando el Archivos de Contraseñas de Autenticación

Para permitir la autentificación de un usuario administrativo que usa la autentificación del archivo de contraseñas se debe hacer el siguiente:

1. Si no existe, crear el archivo de contraseñas usando el utilitario ORAPWD:

```
ORAPWD FILE=filename PASSWORD=password ENTRIES=max_users
```

2. Establecer el parámetro de inicialización REMOTE_LOGIN_PASSWORDFILE en EXCLUSIVE.

> **Nota**
>
> REMOTE_LOGIN_PASSWORDFILE es un parámetro de inicialización estático y por lo tanto no puede ser cambiado sin el reinicio de la base de datos.

3. Conectarse a la base de datos como usuario SYS (o con otro usuario que tenga privilegios administrativos).
4. Si el usuario no existe en la base de datos, crear el usuario.
5. Otorgar el privilegio del sistema SYSDBA ó SYSOPER al usuario.

Conectándonos usando autenticación del archivo de contraseñas

Los usuarios con privilegios administrativos pueden conectarse y autenticarse en una base de datos local ó remota usando el comando SQL*Plus CONNECT. Deben conectar con su username y contraseña, y con la cláusula AS SYSDBA o AS SYSOPER.

Por ejemplo, si al usuario scott se le ha concedido el privilegio SYSDBA, él puede conectarse de la siguiente manera:

```
CONNECT scott/tiger AS SYSDBA
```

Si se le concede el privilegio SYSOPER, él podría conectarse de la siguiente manera:

```
CONNECT scott/tiger AS SYSOPER
```

Notas

La autentificación del sistema operativo toma precedencia sobre la autentificación del archivo de contraseñas. Específicamente, si pertenece al grupo OSDBA u OSOPER del sistema operativo, y se conecta como SYSDBA o SYSOPER, se conectará con privilegios administrativos sin importar el username/password que se especifique.

Si no pertenece al grupo OSDBA ú OSOPER, y tampoco está en el archivo de contraseñas, entonces la conexión fallará.

Ejercicio Dirigido

El siguiente ejercicio debe resolverlo junto con su profesor.

1. Establezca el valor del parámetro REMOTE_LOGIN_PASSWORDFILE en EXCLUSIVE.
2. Quite todos los miembros de los grupos del sistema operativo ORA_DBA y ORA_OPER.
3. Crear el archivo de contraseñas con las siguientes características:
    3. Contraseña del usuario SYS: super
    4. Número máximo de entradas: 10
4. Crear el usuario gustavo, asígnele la contraseña clever.
5. Concédale el privilegio SYSDBA a gustavo.
6. Verifique el contenido del archivo de contraseñas.
7. Verifique la conexión de gustavo como usuario administrador.

Creación y Manteniendo el Archivo de Contraseñas

Podemos crear un archivo de contraseñas usando el programa utilitario para la creación del archivo de contraseñas, ORAPWD. Para algunos sistemas operativos, podemos crear este archivo como parte de su instalación estándar.

Usando ORAPWD

Cuando se invoca el programa utilitario de creación del archivo de contraseñas sin proveer ningún parámetro, se recibe un mensaje que indica el uso apropiado del comando, tal como se ilustra a continuación:

```
C:\> orapwd          [Enter]
Usage: orapwd file=<fname> password=<password> entries=<users> force=<y/n>

  where
    file - name of password file (mand),
    password - password for SYS (mand),
    entries - maximum number of distinct DBA and OPERs (opt),
    force - whether to overwrite existing file (opt),
  There are no spaces around the equal-to (=) character.

C:\>
```

El siguiente comando crea un archivo de contraseñas de nombre acct.pwd que permite hasta 30 usuarios privilegiados con diversas contraseñas. En este ejemplo, el archivo se crea inicialmente con la contraseña secret para los usuarios que conectan como SYS.

```
C:\> ORAPWD FILE=acct.pwd PASSWORD=secret ENTRIES=30    [Enter]
```

Nota

Es críticamente importante para la seguridad del sistema proteger el archivo de contraseñas y las variables de entorno que identifican su localización. Cualquier usuario con acceso podría potencialmente comprometer la seguridad de la conexión.

A continuación pasamos a describir los parámetros del utilitario ORAPWD.

FILE Este parámetro establece el nombre del archivo de contraseñas a ser creado. Se debe especificar el nombre completo y la ruta del archivo. El contenido de este archivo es encriptado, y no es posible leerlo directamente. Este parámetro es obligatorio.

Los tipos de nombre de archivo permitidos para el archivo de contraseñas son específicos del sistema operativo. Algunos sistemas operativos requieren que el archivo de contraseñas tenga un formato específico y localizado en un directorio específico. Otros sistemas operativos permiten el uso de las variables de entorno para especificar el nombre y la localización del archivo de contraseñas. Debemos revisar la documentación específica de Oracle del sistema operativo para los nombres y las localizaciones permitidas en su plataforma.

PASSWORD En este parámetro establecemos la contraseña para el usuario SYS. Si usamos la sentencia ALTER USER para cambiar la contraseña de SYS después de conectarse a la base de datos, la contraseña almacenada en el diccionario de datos y la contraseña almacenada en el archivo de contraseñas ambas son actualizadas. Este parámetro es obligatorio.

ENTRIES Este parámetro especifica el número de entradas que se requiere que el archivo de contraseñas acepte. Este número corresponde al número de usuarios distintos permitidos para conectar a la base de datos como SYSDBA o SYSOPER. El número real de entradas permisibles puede ser más alto que el número de usuarios porque la utilidad ORAPWD continúa asignando entradas de contraseña hasta que se llena un bloque del sistema operativo. Por ejemplo, si el tamaño del bloque del sistema operativo es de 512 Bytes, esto aloja cuatro entradas de contraseña. El número de entradas de contraseña alojadas será siempre múltiplo de cuatro.

FORCE Este parámetro, si se establece en Y, le permite sobre-escribir un archivo de contraseñas existente. Devuelve un error si existe un archivo de contraseñas del mismo nombre y se omite este parámetro o se establece en N.

Estableciendo el valor de REMOTE_LOGIN_PASSWORDFILE

Además de crear el archivo de la contraseña, debemos también fijar el parámetro de inicialización REMOTE_LOGIN_PASSWORDFILE en el valor apropiado. Los valores reconocidos se describen a continuación:

Valor	Descripción
NONE	Fijar este parámetro en NONE hace que Oracle funcione como si no existiese el archivo de contraseñas. Es decir, no se permite ninguna conexión privilegiada sobre conexiones no-seguras.
EXCLUSIVE	Un archivo de contraseñas EXCLUSIVE se puede utilizar solamente con una base de datos. Usar un archivo de contraseñas EXCLUSIVE permite agregar, modificar, y suprimir usuarios. También permite cambiar la contraseña de SYS con el comando ALTER USER. Este es el valor por defecto.
SHARED	Un archivo de contraseñas SHARED puede ser utilizado por múltiples bases de datos que funcionan en el mismo servidor. Sin embargo, el archivo no puede ser modificado. Esto significa que no se puede agregar nuevos usuarios a un archivo de contraseñas SHARED. Todos los usuarios que necesitan los privilegios del sistema SYSDBA y/o SYSOPER deben ser agregados al archivo de contraseñas cuando REMOTE_LOGIN_PASSWORDFILE se establece en EXCLUSIVE. Después que se agreguen a todos los usuarios, se puede cambiar REMOTE_LOGIN_PASSWORDFILE a SHARED, y después comparte el archivo. Esta opción es útil si se tiene uno o más DBAs que administran bases de datos múltiples.

Adicionando Usuarios al Archivo de Contraseñas

Cuando concede privilegios SYSDBA y/o SYSOPER a un usuario, el nombre de usuario y la información de su privilegio se agregan al archivo de contraseñas. Si el servidor no tiene un archivo de contraseñas EXCLUSIVE (es decir, si el parámetro de inicialización REMOTE_LOGIN_PASSWORDFILE se ha establecido en NONE ó SHARED) se mostrará un mensaje de error si tratamos de conceder estos privilegios.

El nombre de usuario se mantiene en el archivo de contraseñas solamente mientras ese usuario tiene por lo menos uno de estos dos privilegios. Si se revoca ambos privilegios, se elimina el usuario del archivo de contraseñas.

Creando un Archivo de Contraseñas y Agregándole Nuevos Usuarios

Use el siguiente procedimiento para crear un nuevo archivo de contraseñas, y agregarle nuevos usuarios:

1. Proceda a crear un nuevo archivo de contraseñas.
2. Establezca el parámetro de inicialización REMOTE_LOGIN_PASSWORDFILE en EXCLUSIVE.
3. Conéctese a Oracle con el privilegio SYSDBA.
4. Inicie la instancia y crear la base de datos si es necesario, o monte e inicie una base de datos existente.
5. Crear los usuarios necesarios. Otorgar los privilegios SYSDBA y/o SYSOPER a usted mismo y otros usuarios apropiadamente.

Otorgando y Revocando Privilegios SYSDBA y SYSOPER

Para otorgar el privilegio SYSDBA y/o SYSOPER debe utilizar la instrucción GRANT, su sintaxis es la siguiente:

```
GRANT privilegio TO usuario;
```

Para revocar el privilegio debe utilizar la instrucción es REVOKE, su sintaxis es la siguiente:

```
REVOKE privilegio FROM usuario;
```

Consultando los Miembros del Archivo de Contraseñas

Podemos usar la vista V$PWFILE_USERS para consultar los usuarios que tienen los privilegios SYSDBA y SYSOPER para la base de datos. Esta vista tiene las siguientes columnas:

Columna	Descripción
USERNAME	Esta columna contiene el nombre del usuario que es reconocido por el archivo de contraseñas.
SYSDBA	Si el valor de esta columna es TRUE, entonces el usuario puede iniciar sesión con el privilegio SYSDBA.
SYSOPER	Si el valor de esta columna es TRUE, entonces el usuario puede iniciar sesión con el privilegio SYSOPER.

En el siguiente ejemplo se ilustra el uso de la vista V$PWFILE_USERS.

```
SQL> conn sys/admin as sysdba     [Enter]

Conectado.

SQL> desc v$pwfile_users          [Enter]

Nombre                                    ¿Nulo?   Tipo
----------------------------------------- -------- -------------------
USERNAME                                           VARCHAR2(30)
SYSDBA                                             VARCHAR2(5)
SYSOPER                                            VARCHAR2(5)

SQL> select * from v$pwfile_users;     [Enter]

USERNAME                       SYSDB SYSOPER
------------------------------ ----- --------
SYS                            TRUE  TRUE
SCOTT                          TRUE  FALSE
```

Mantenimiento del Archivo de Contraseñas

Ampliar el número de usuarios del archivo de contraseñas

Si se recibe el error de archivo (ORA-1996) cuando se intenta conceder los privilegios del sistema SYSDBA o SYSOPER a un usuario, se debe crear un archivo de contraseñas más grande y re-conceder los privilegios a los usuarios.

Reemplazar el archivo de contraseñas:

Use el siguiente procedimiento para reemplazar el archivo de contraseñas:

1. Anote los usuarios que tienen privilegios SYSDBA ó SYSOPER consultando la vista V$PWFILE_USERS.
2. Cierre la base de datos con la instrucción SHUTDOWN.
3. Elimine si el archivo de contraseñas existente.
4. Se debe crear nuevamente el archivo de contraseñas con el utilitario ORAPWD. Debemos asegurarnos que estamos aumentando el número de entradas.
5. Finalmente proceda a agregar los usuarios al archivo de contraseñas.

Eliminar el Archivo de Contraseñas

Si determina que ya no se requiere más de un archivo de contraseñas para autenticar a usuarios, puede eliminar este archivo y reestablecer el parámetro de inicialización REMOTE_LOGIN_PASSWORDFILE a NONE. Después de que eliminemos este archivo, sólo los usuarios que puedan ser autenticados por el sistema operativo podrán realizar operaciones de administración en la base de datos.

Página en Blanco

Lección 05

Inicio y Parada de la Base de Datos

Contenido

- Introducción 52
- Inicio de una Base de Datos de Oracle 53
  - Fases del Inicio de la Base de Datos 53
- Comando: STARTUP 55
- Comando: SHUTDOWN 56
  - SHUTDOWN con la cláusula NORMAL 56
  - SHUTDOWN con la cláusula IMMEDIATE 56
  - SHUTDOWN con la cláusula TRANSACTIONAL 57
  - SHUTDOWN con la cláusula ABORT 58
- Comando: ALTER DATABASE 59

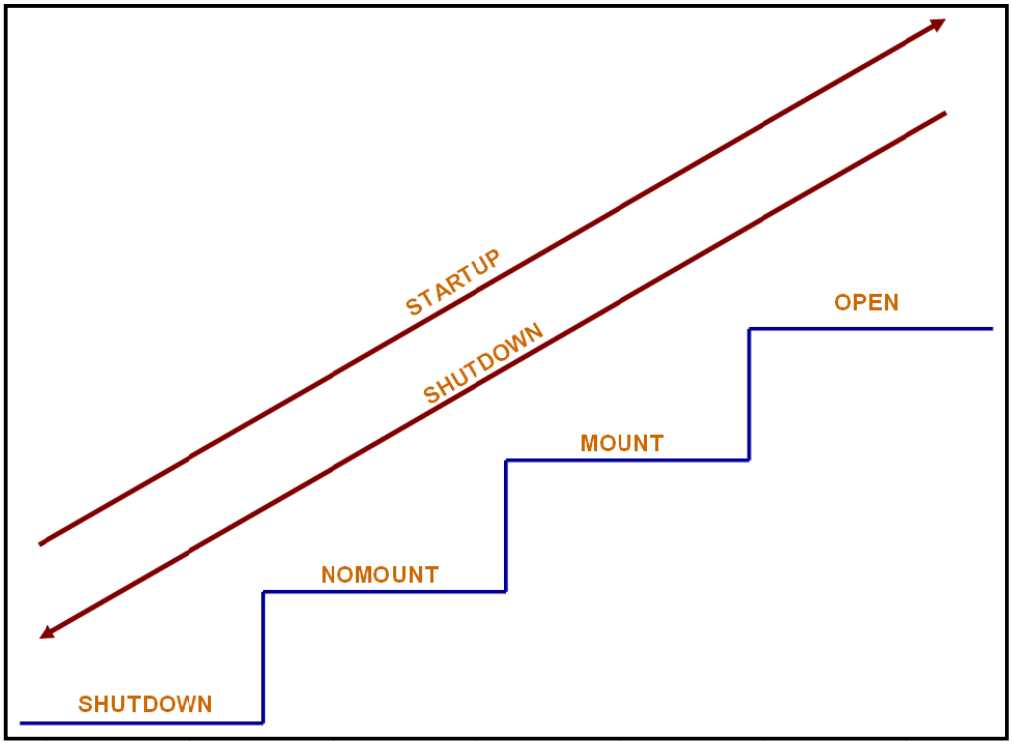
Introducción

El DBA, es el responsable del inicio y parada de la instancia de Oracle. Oracle proporciona a los administradores autorizados las opciones de realizar esta tarea usando una variedad de interfaces. Es importante entender las opciones que están disponibles para iniciar y cerrar la instancia de Oracle, y cuando las varias opciones pueden o deben ser utilizadas.

Para iniciar o cerrar una instancia de Oracle, necesita estar conectado con la base de datos con los privilegios apropiados. Hay dos privilegios especiales que están disponibles para usuarios que van a realizar estas tareas: SYSDBA y SYSOPER. El privilegio SYSDBA es un privilegio que permite realizar cualquier tarea de la base de datos. La autorización SYSOPER es un privilegio de menor alcance que permite operaciones de inicio y parada pero restringe otras tareas administrativas, tales como el acceso a los objetos de esquemas de las aplicaciones. Estas autorizaciones se manejan en el archivo de contraseñas o vía el sistema operativo.

Cuando una base de datos está recientemente instalada, sólo el usuario SYS puede conectarse con la base de datos con el privilegio SYSDBA. Se puede otorgar este privilegio y el privilegio SYSOPER a otros usuario para ejecutar esta tarea sin la necesidad de conectarse con el usuario SYS.

El siguiente gráfico muestra las diferentes fases del proceso de inicio y parada de una base de datos de Oracle.

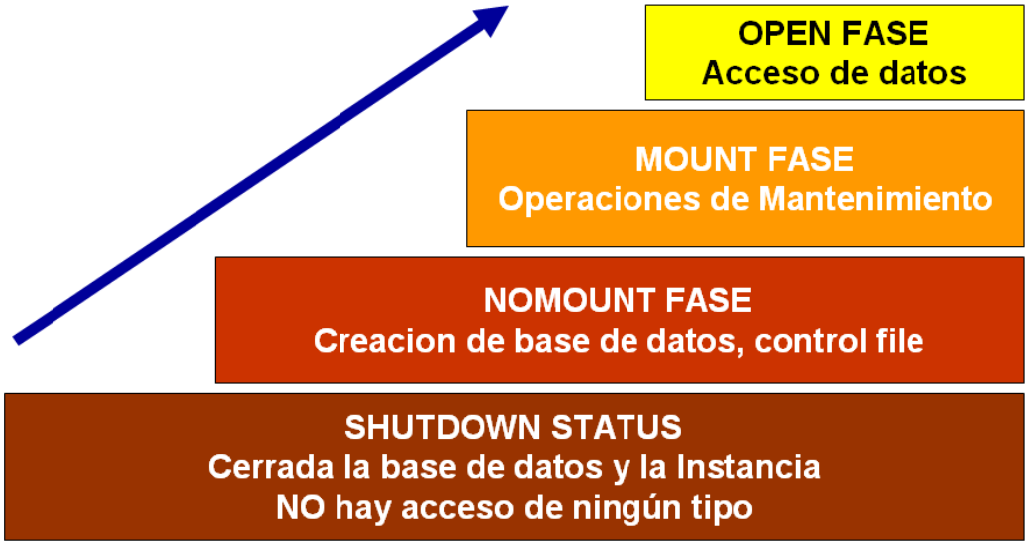


Inicio de una Base de Datos de Oracle

Fases del Inicio de la Base de Datos

Según lo descrito en la Lección 01, la instancia de Oracle se compone de un conjunto de estructuras lógicas en memoria y de procesos de fondo que interactúan con los usuarios para la comunicación con la base de datos de Oracle. Cuando Oracle esta iniciado, las estructuras en memoria y los procesos de fondo están inicializados y abiertos para que los usuarios puedan comunicarse con la base de datos de Oracle.

Siempre que se inicia una base de datos de Oracle, pasa por una serie de pasos que asegurar su consistencia. Cuando se inicia, una base de datos pasa por tres estados: NOMOUNT, MOUNT, y OPEN, tal como se ilustra en la siguiente figura.



Fase: NOMOUNT

Una base de datos en esta fase incluye las siguientes tareas:

- Se lee el archivo de parámetros.
- Se identifican los archivos de control.
- Se inician los procesos de fondo.
- Se inician las estructuras en memoria, pero no son vinculadas con las estructuras en disco.
- La base de datos no esta disponible para los usuarios.

Si una base de datos se encuentra en esta fase, se pueden realizar ciertas tareas. Una de las más comunes es la creación de una nueva base de datos.

Ocasionalmente, una base de datos no puede pasar a la fase siguiente (MOUNT) y se mantiene en la fase NOMOUNT. Por ejemplo, esto puede ocurrir si se tiene problemas para acceder a las estructuras del archivo de control, que contiene información importante para continuar con el proceso de inicio de la base de datos. Si se dañan estas estructuras o no están disponibles, el proceso de inicio de la base de datos no puede continuar hasta que se resuelva este problema.

Fase: MOUNT

En esta fase, primero se deben ejecutar todas las tareas para llegar a la fase NOMOUNT, y luego se ejecutan las siguientes tareas:

- Se asocia los archivos de la base de datos con la instancia previamente inicializada.
- Se localizan y abren los archivos de control indicados en el archivo de parámetros.
- Se lee el archivo de control y se obtiene el estado de los archivos de datos y los redo log files.

Ciertas tareas administrativas se pueden realizar mientras la base de datos esta en esta fase, como por ejemplo:

- Renombrar los archivos de datos.
- Realizar un proceso de recovery.
- Cambiar la ubicación física de los archivos.
- Poner la base de datos en modo Archive Log.

Fase: OPEN

En esta fase, primero se deben ejecutar las tareas para llegar a la fase MOUNT, y luego se ejecutan las siguientes tareas:

- Se abren los archivos de datos.
- Se abren los redo log files en línea.

En esta fase la base de datos esta disponible para todos los usuarios.

Comando: STARTUP

Este comando permite iniciar una instancia y abrir la base de datos. Cuando se utiliza sin parámetros, inicia la instancia, monta y abre la base de datos.

Sintaxis

```
STARTUP
    [NOMOUNT | MOUNT | OPEN]
    [PFILE/SPFILE=ruta_nombre_archivo]
    [RESTRICT]
    [FORCE]
```

Donde:

NOMOUNT	Inicia la instancia, pero no monta la base de datos.
MOUNT	Inicia la instancia, y monta la base de datos.
OPEN	Inicia la instancia, monta y abre la base de datos.
PFILE/SPFILE	Permite indicar el archivo de parámetros.
RESTRICT	Abre la base de datos en modo restringido. Solo los usuarios con privilegio RESTRICTED SESSION podrán iniciar sesión.
FORCE	Fuerza el inicio de una base de datos. Si la base de datos esta en funcionamiento, en primer lugar realiza ejecuta un SHUTDOWN.

Comando: SHUTDOWN

En algunos casos, es necesario cerrar la base de datos, por ejemplo, para ejecutar backups en frió, regularmente para programar o realizar mejoras de la base de datos. Cualquiera de estos casos, es necesario entender las opciones de la parada de una base de datos.

Sintaxis

```
SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT]
```

SHUTDOWN con la cláusula NORMAL

Para cerrar una base de datos en situaciones normales, utilice el comando SHUTDOWN con la cláusula NORMAL:

```
SHUTDOWN NORMAL
```

La cláusula NORMAL es opcional, porque éste es el método de parada por defecto si no se proporciona ninguna cláusula.

La parada normal de la base de datos procede con las siguientes condiciones:

- No se permite nuevas conexiones.
- Espera a que todos los usuarios actualmente conectados se desconecten para proceder a cerrar la base de datos.

El próximo inicio de la base de datos no requerirá ningún procedimiento para recuperar la instancia.

SHUTDOWN con la cláusula IMMEDIATE

Utilice la parada inmediata de la base de datos solamente en las siguientes situaciones:

- Para iniciar un backup automatizado y desatendido.
- Cuando un corte de energía va a ocurrir muy pronto.
- Cuando la base de datos o una de sus aplicaciones esta funcionando irregularmente y no puede contactarse con el usuario para indicarle que terminen su sesión ó no pueden terminar su sesión.

Para cerrar una base de datos inmediatamente, se utiliza el comando SHUTDOWN con la cláusula IMMEDIATE:

```
SHUTDOWN IMMEDIATE
```

La parada inmediata de la base de datos procede con las siguientes condiciones:

- No se permite nuevas conexiones, tampoco se pueden iniciar nuevas transacciones.
- No espera a que las sesiones activas sean terminadas.
- Las transacciones pendientes son canceladas, y los usuarios son desconectados.

El próximo inicio de la base de datos no requerirá ningún procedimiento para recuperar la instancia.

SHUTDOWN con la cláusula TRANSACTIONAL

Cuando se requiere realizar una parada de la instancia que permita terminar las transacciones activas, se debe utilizar el comando SHUTDOWN con la cláusula TRANSACTIONAL:

```
SHUTDOWN TRANSACTIONAL
```

La parada transaccional de la base de datos procede con las siguientes condiciones:

- No se permite nuevas conexiones, tampoco se pueden iniciar nuevas transacciones.
- Después que las transacciones son terminadas, los clientes son desconectados de la base de datos.

El próximo inicio de la base de datos no requerirá ningún procedimiento para recuperar la instancia.

Una parada transaccional previene a clientes pierdan su trabajo, y al mismo tiempo, no requiere que todos los usuarios terminar su sesión.

SHUTDOWN con la cláusula ABORT

Se puede cerrar la base de datos instantáneamente abortando la instancia de la base de datos. Si es posible, se debe realizar este tipo de parada solamente en las situaciones siguientes:

- La base de datos o una de sus aplicaciones está funcionando irregularmente y ninguno de los otros tipos de parada funcionan.
- Se necesita cerrar la base de datos instantáneamente (por ejemplo, si se sabe un corte de energía va a ocurrir en un minuto).
- Se ha encontrado problemas al iniciar la instancia de la base de datos.

Cuando se necesita una parada de la base de datos abortando transacciones y conexiones de usuarios, ejecute el comando SHUTDOWN con la cláusula ABORT:

```
SHUTDOWN ABORT
```

La parada abortada de la base de datos procede con las siguientes condiciones:

- No se permite nuevas conexiones, tampoco se pueden iniciar nuevas transacciones.
- Las sentencias SQL activas del cliente que son procesadas por la base de datos son terminadas inmediatamente.
- Las transacciones activas son canceladas.
- No se espera a que la conexiones activas sean desconectadas, son desconectadas inmediatamente.

El próximo inicio de la base de datos requerirá un procedimiento para recuperar la instancia.

Comando: ALTER DATABASE

Cuando se inicia la base de datos con el comando STARTUP NOMOUNT, para pasar a la siguiente fase MOUNT u OPEN se debe ejecutar el comando ALTER DATABASE:

Sintaxis

```
ALTER DATABASE  MOUNT | OPEN  [ READ ONLY | READ WRITE ]
```

Donde:

MOUNT	Monta la base de datos.
OPEN	Abre la base de datos, y por lo tanto, estará disponible para los usuarios.
READ ONLY	Abre la base de datos en modo de solo lectura.
READ WRITE	Abre la base de datos en modo lectura y escritura.

Página en Blanco

Lección 06

Acceso a la Base de Datos Oracle

Contenido

- Verificación de Servicios Windows 62
  - Servicio de la Base de Datos 62
  - Servicio del Proceso Escucha 62
  - Tipos de Inicio de los Servicios 63
  - Estados de un Servicio 63
  - La Ventana de Servicios 64
- Identificación del Servicio y la Instancia de Base de Datos 65
  - Servicio de Base de Datos 65
  - Instancias de la Base de Datos 67
  - Servicio de Accesibilidad 70
  - Proceso de Servidor Dedicado 72
  - Nomenclatura 73
- Configuración del Método de Nomenclatura Local 75
  - Configurar el archivo tnsname.ora durante la instalación 75
  - Configurar el archivo tnsname.ora después de la instalación 75
- Configuración del Proceso Escucha 77
  - Configuración del Proceso Escucha Durante la Instalación 77
- Glosario 78

Verificación de Servicios Windows

De la relación de servicios creados durante la instalación de Oracle, por ahora nos interesa básicamente dos, el primero relacionado con la base de datos y el segundo con el proceso escucha.

Servicio de la Base de Datos

Este servicio esta relacionado con la instancia y la base de datos, la estructura del nombre de este servicio es:

OracleService**SID**

Donde SID representa el nombre de la instancia, por ejemplo, si el nombre de la instancia es EGCC, el nombre del servicio es:

OracleServiceEGCC

Servicio del Proceso Escucha

Este servicio esta relacionado con la disponibilidad de la base de datos para el acceso remoto, el nombre por defecto de este servicio es:

OracleOraDb10g_home1TNSListener

Tipos de Inicio de los Servicios

Los servicios tienen tres tipos de inicios, tal como se describe en el siguiente cuadro:

Tipo de Inicio	Descripción
Manual	El servicio debe ser iniciado de manera explicita con el comando `net` o desde la ventana de servicios.
Automático	El servicio se inicia de manera automática junto con el sistema operativo.
Deshabilitado	El servicio no esta disponible, por lo tanto no podrá ser iniciado.

En cualquier momento se puede ejecutar el comando **net** para modificar el estado del servicio, este comando se ejecuta desde la consola de comandos.

Estados de un Servicio

Un servicio puede tener uno de los estados que se describen en el siguiente cuadro:

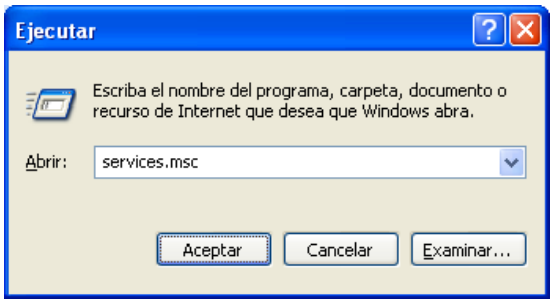
Estado	Descripción
Iniciado	El servicio esta funcionando y disponible.
Detenido	El servicio no esta disponible.
Pausado	El servicio esta funcionando, pero no esta disponible. Se asume que es un estado temporal.

La Ventana de Servicios

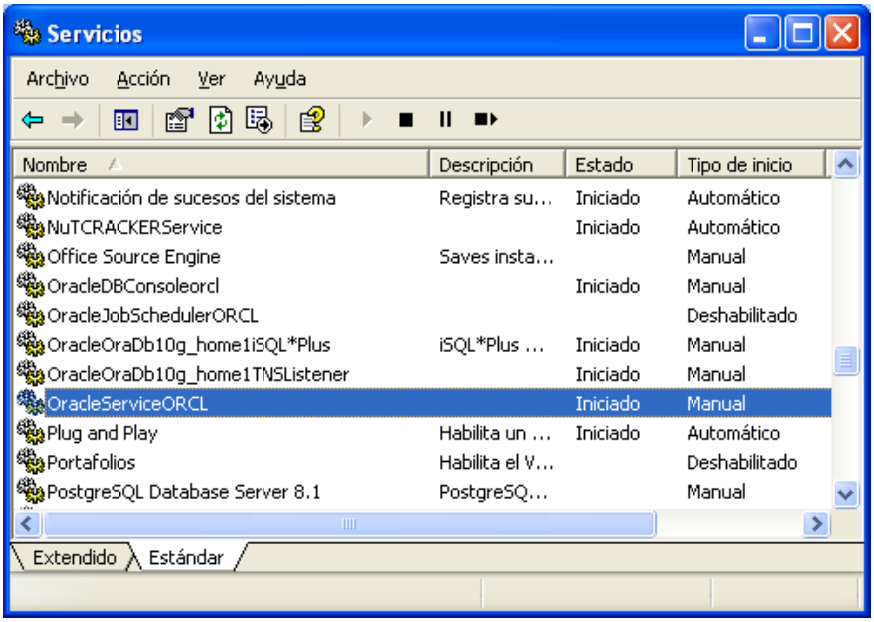
La ventana de servicios permite administrar los servicios, para cargar esta ventana debe seguir los siguientes pasos:

1. Desde el menú **Inicio** ejecute el comando **Ejecutar**.

2. En el campo abrir escriba el comando: **services.msc**



3. Luego haga clic en el botón **Aceptar**.

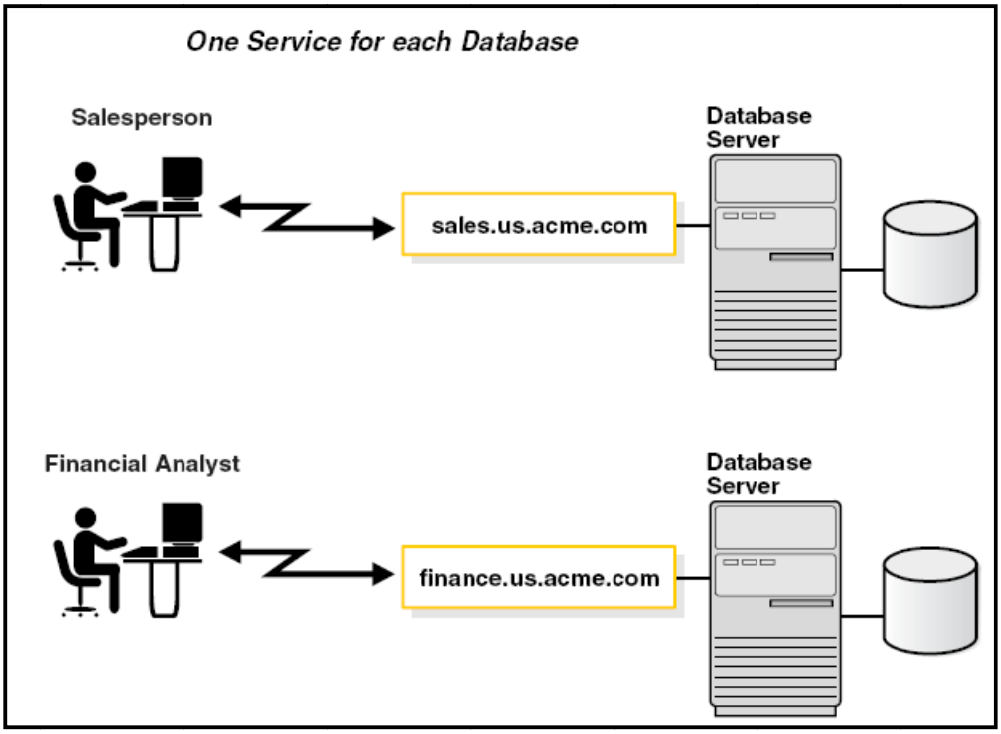


Identificación del Servicio y la Instancia de Base de Datos

Servicio de Base de Datos

Una base de datos Oracle esta representada para los clientes como un servicio; es decir, la base de datos realiza el trabajo a nombre de los clientes. Una base de datos puede tener uno o más servicios asociados a ella.

La siguiente figura muestra dos bases de datos, cada uno con su propio servicio de base de datos para los clientes de Intranet. Un servicio, sales.us.acme.com, permite a vendedores tener acceso a la base de datos de ventas. Otro servicio, finance.us.acme.com, permite a analistas financieros tener acceso a la base de datos de finanzas.



Las bases de datos de ventas y finanzas son identificadas por un nombre de servicio, **sales.us.acme.com** y **finance.us.acme.com** respectivamente. El nombre del servicio es especificado por el parámetro de inicialización SERVICE_NAMES en el archivo de parámetros de inicialización. El nombre del servicio por defecto es el nombre global de la base de datos, el nombre abarca el nombre de la base de datos (parámetro DB_NAME) y el nombre del dominio (parámetro DB_DOMAIN). En el caso de **sales.us.acme.com**, **sales** es el nombre de la base de datos y **us.acme.com** es el nombre del dominio.

Nota

Es posible cambiar dinámicamente el valor del parámetro de SERVICE_NAMES con la instrucción SQL ALTER SYSTEM cuando la base de datos está en servicio.

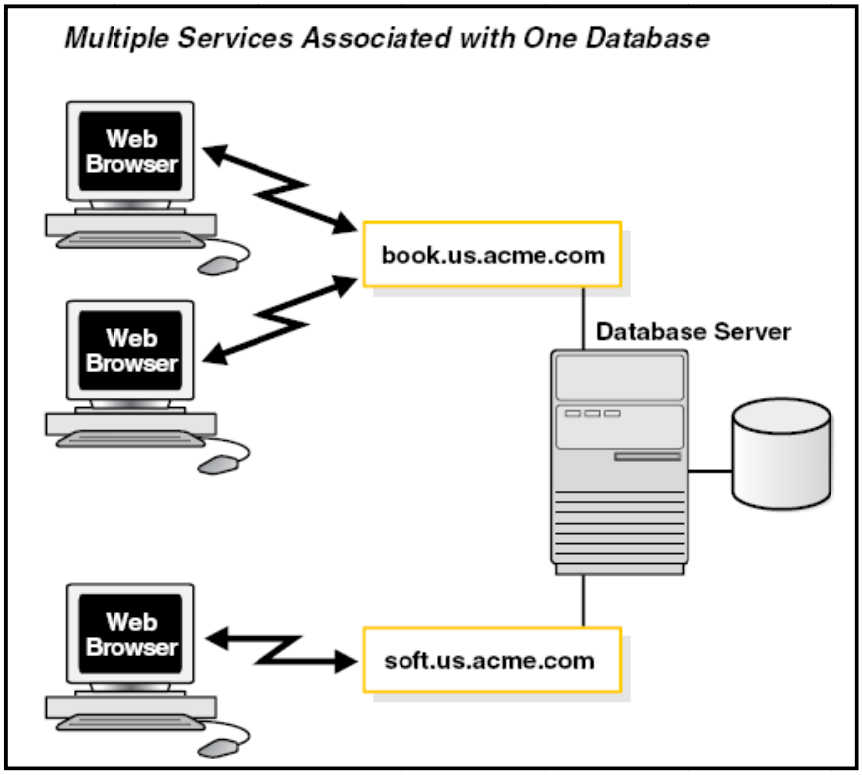
Las siguientes instrucciones consultan el valor de los parámetros SERVICE_NAMES, DB_NAME, y DB_DOMAIN.

```
SQL> show parameter service_names    [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
service_names                        string      orcl.www.perudev.com

SQL> show parameter db_name          [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
db_name                              string      orcl

SQL> show parameter db_domain        [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
db_domain                            string      www.perudev.com
```

Una base de datos puede tener múltiples servicios asociados a ella. La siguiente figura muestra una base de datos que tiene dos diferentes servicios para los clientes Web. Un servicio, **book.us.acme.com**, dedicado a clientes que hacen compras de libros. El otro servicio, **soft.us.acme.com**, dedicado a clientes que hacen compras de software.

Tener múltiples servicios asociados a una base de datos permite las siguientes funcionalidades:

- Una base de datos puede ser identificada de diversas maneras por diversos clientes.
- Un administrador de la base de datos puede limitar o reservar recursos del sistema. Este nivel de control permite una asignación de recursos mejor a los clientes que solicitan uno de estos servicios.



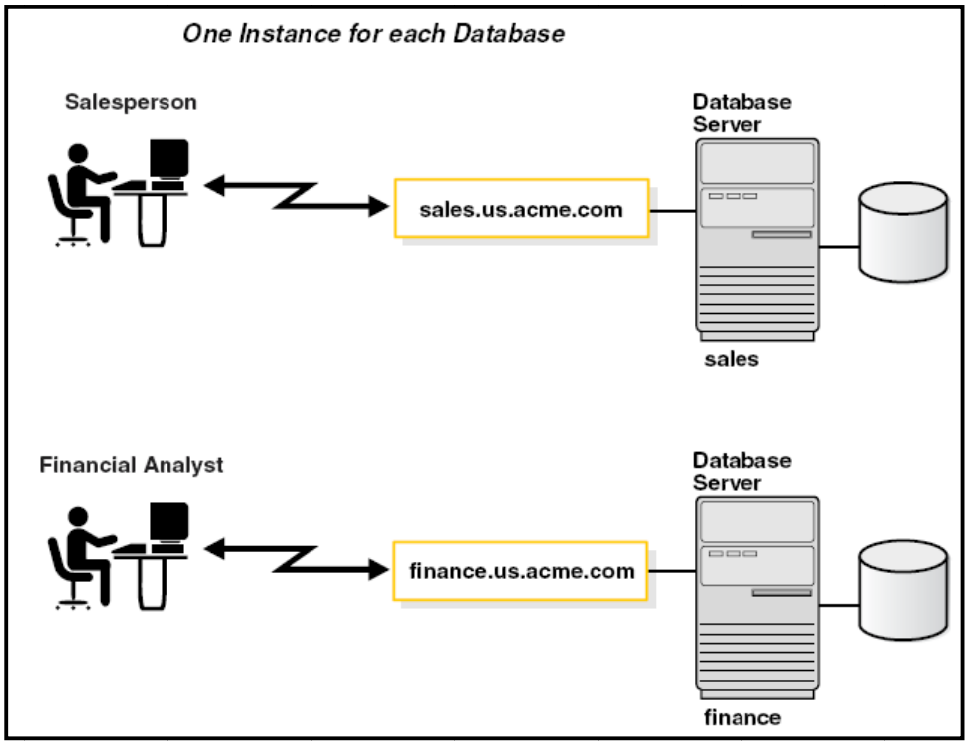
Instancias de la Base de Datos

Una base de datos tiene por lo menos una instancia. Un instancia esta compuesta de un área de memoria llamada *System Global Area* **(SGA)** y procesos de Oracle. La memoria y los procesos de una instancia manejan eficientemente los datos de la base de datos asociada y sirven a usuarios de la base de datos.

Nota

Una instancia también maneja otros servicios, como por ejemplo: **Oracle XML DB**.

La siguiente figura muestra dos instancias, **sales** y **finance**, asociadas con sus respectivas bases de datos.

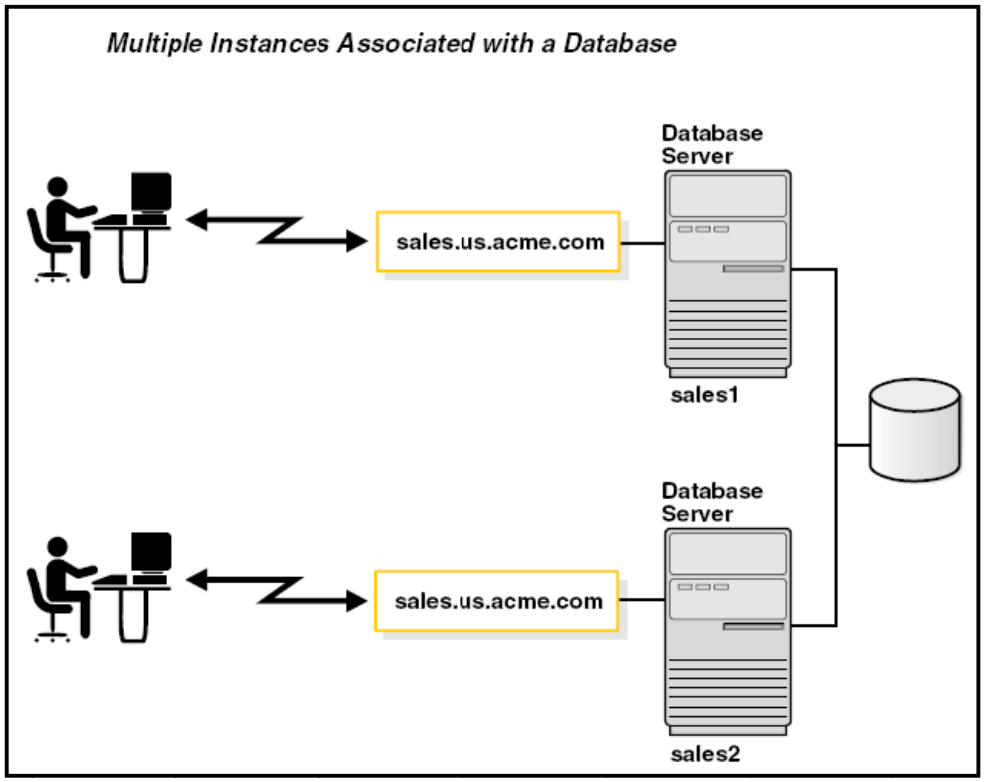


Como servicios, las instancias son identificadas por un **nombre de instancia**, *sales* y *finance* en este ejemplo. El nombre de la instancia es especificado por el parámetro INSTANCE_NAME en el archivo del parámetro de inicialización. El nombre de la instancia por defecto es ***Oracle System Identifier* (SID)** de la instancia.

```
SQL> show parameter instance_name    [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- --------------
instance_name                        string      orcl
```

Algunas arquitecturas de hardware permiten que múltiples computadoras compartan el acceso a los datos, al software, o a los dispositivos periféricos. ***Oracle Real Application Clusters*** pueden aprovechar de tal arquitectura para ejecutar múltiples instancias en diversas computadoras que compartan una sola base de datos física.

La siguiente figura muestra la configuración de **Oracle Real Application Clusters**, en este ejemplo, dos instancias, *sales1* y *sales2*, están asociadas con una base de datos, *sales.us.acme.com*.



Servicio de Accesibilidad

Para conectarse a un servicio de base de datos, los clientes usan un **descriptor de conexión** que provee la ubicación de la base de datos y el nombre del servicio de base de datos, El siguiente ejemplo muestra un **descriptor de conexión** que habilita a los clientes para conectarse a un servicio de base de datos llamado *sales.us.acme.com*.

```
(DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
    (CONNECT_DATA=
        (SERVICE_NAME=sales.us.acme.com)
    )
)
```

La sección **ADDRESS** del **descriptor de conexión** es realmente la **dirección de protocolo** del **proceso escucha**. Para conectarse a un servicio de base de datos, los clientes primero se contactan con un **proceso escucha** que típicamente reside en el servidor de base de datos. El proceso escucha recibe las solicitudes de conexión de los clientes y entrega estos requerimientos al servidor de base de datos. Una vez establecida la conexión, el cliente y el servidor se comunican directamente.

Como una dirección comercial, el proceso escucha es configurado para aceptar peticiones de clientes en una dirección de protocolo. Esta dirección define el protocolo que el proceso escucha está escuchando y alguna otro información específica del protocolo. Por ejemplo, el proceso escucha podría ser configurado para escuchar en la siguiente dirección del protocolo:

```
(DESCRIPTION=
    ( ADDRESS = (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521) )
)
```

Este ejemplo muestra a una dirección de protocolo TCP/IP que especifica el host y un número de puerto del proceso escucha. Los clientes configurados con esta misma dirección de protocolo pueden enviar solicitudes de conexión a este proceso escucha.

El **descriptor de conexión** también especifica el nombre del servicio de la base de datos con la cual los clientes intentan establecer una conexión. El proceso escucha conoce los servicios para los cuales puede manejar solicitudes de conexión, porque una base de datos de Oracle registra dinámicamente esta información en el proceso escucha. Este proceso del registro se llama **registro del servicio**. También provee al proceso escucha información sobre las instancias de la base de datos y los manejadores del servicio disponibles para cada instancia. Los manejadores de servicio actúan como puntos de conexión a un servidor de base de datos Oracle. Un manejador del servicio puede ser un **despachador** o un **servidor dedicado**.

Si se requiere la conexión a una instancia específica de la base de datos, entonces los clientes también pueden especificar el INSTANCE_NAME de una instancia particular en el **descriptor de conexión**. Por ejemplo, el siguiente **descriptor de conexión** especifica el nombre de instancia **sales1** que esta asociada con **sales.us.acme.com**.

```
(DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
    (CONNECT_DATA=
        (SERVICE_NAME=sales.us.acme.com)
        (INSTANCE_NAME=sales1)
    )
)
```

Alternativamente, los clientes que siempre quieren usar un tipo particular de manejador de servicio pueden usar un **descriptor de conexión** que especifica el tipo de manejador de servicio. En el siguiente ejemplo, un **descriptor de conexión** es configurado para usar un despachador para una configuración de servidor compartido, indicado por: (SERVER = shared).

```
(DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
    (CONNECT_DATA=
        (SERVICE_NAME=sales.us.acme.com)
        (SERVER=shared)
    )
)
```

Si queremos que el cliente use un servidor dedicado, debemos especificar (SERVER=dedicated) en lugar de (SERVER=shared). Si el parámetro SERVER no esta establecido, entonces la configuración de servidor compartido es asumido. Sin embargo, el cliente usará servidor dedicado si no esta disponible un despachador.

Cuando el proceso escucha recibe los requerimientos del cliente, este selecciona uno de los manejadores de servicios que fueron previamente registrados. Dependiendo del tipo de manejador de servicio seleccionado, el protocolo de comunicación usado, y el sistema operativo del servidor de base de datos, el proceso escucha ejecuta una de las siguientes acciones.

- Da la petición de conexión directamente a un despachador.
- Envía un mensaje al cliente con la localización del despachador ó el proceso de servidor dedicado. El cliente entonces se conecta directamente con el despachador o el proceso dedicado del servidor.
- Genera un proceso de servidor dedicado y pasa la conexión del cliente al proceso de servidor dedicado.

Una vez que el proceso escucha haya terminado la operación de conexión para el cliente, el cliente se comunica con el servidor de la base de datos Oracle sin la participación del proceso escucha. El proceso escucha reasume la esperar a escuchar nuevas sesiones entrantes de la red.

Proceso de Servidor Dedicado

Una configuración de servidor dedicado, el proceso escucha inicia un proceso de servidor dedicado para cada solicitud de conexión de los clientes dedicado al mantenimiento del cliente. Una vez que la sesión es finalizada, el proceso de servidor dedicado finaliza. Por que un proceso de servidor dedicado tiene que ser iniciado para cada conexión, esta configuración requiere más recursos del sistema que la configuración de servidor compartido.

Un proceso de servidor dedicado es un tipo de servicio que el proceso escucha inicia cuando recibe una solicitud de un cliente. Para terminar la conexión cliente/servidor, una de las siguientes acciones debe ocurrir:

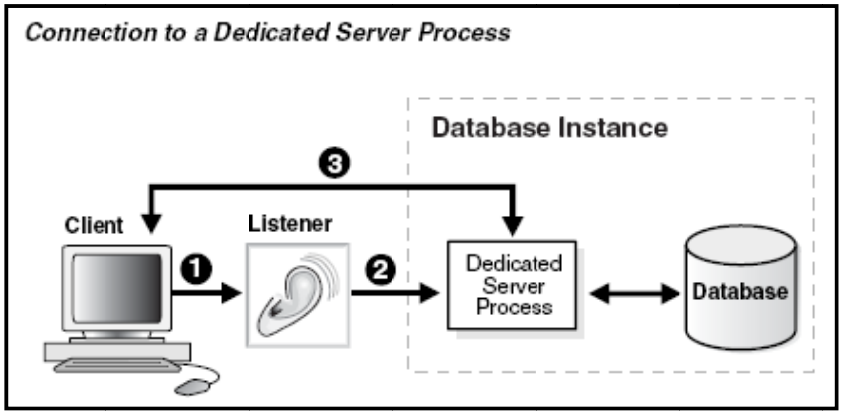
- El servidor dedicado hereda la solicitud de conexión desde el proceso escucha.
- El servidor dedicado informa al proceso escucha su dirección de protocolo. El proceso escucha pasa la dirección de protocolo al cliente en un mensaje de respuesta y termina la conexión. El cliente se conecta con el servidor dedicado usando directamente la dirección de protocolo.

Si el cliente y el servidor de base de datos existen en la misma computadora, la conexión del cliente puede pasar directamente a un proceso de servidor dedicado sin ir a través del proceso escucha. La aplicación que inicia la sesión genera un proceso de servidor dedicado para la solicitud de conexión. Esto sucede automáticamente si la aplicación que es usada para iniciar la sesión en la base de datos esta en la misma computadora de la base de datos.

Nota

Para que los clientes remotos puedan conectarse con el servidor dedicado, el proceso escucha y la instancia de la base de datos deben funcionar en la misma la computadora.

La siguiente figura muestra un proceso escucha pasando la solicitud de conexión de un cliente a un proceso de servidor dedicado.



1. El proceso escucha recibe la solicitud de conexión del cliente.

2. El proceso escucha inicia un proceso de servidor dedicado, y el servidor dedicado hereda la solicitud de conexión desde el proceso escucha.

3. El cliente es ahora conectado directamente con el servidor dedicado.

Nomenclatura

Los usuarios inician una solicitud de conexión proporcionando una **cadena de conexión**. Una cadena de conexión incluye un nombre de usuario y una contraseña, junto con un **identificador de conexión**. Un identificador de conexión puede ser el mismo descriptor de conexión o un nombre que resuelve a un descriptor de conexión. Uno de los identificadores de conexión más comunes es un **nombre de servicio de red**, un nombre sencillo para un servicio. Los ejemplos siguientes muestran una cadena de conexión que utiliza un descriptor de conexión completo como identificador de conexión y otra cadena de conexión que utiliza el nombre de servicio de red **sales** como identificador de conexión.

```
CONNECT
scott/tiger@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server1)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com)))

CONNECT scott/tiger@sales
```

Cuando el nombre de servicio de red **sales** es usado, el proceso de conexión resuelve el nombre **sales** a un descriptor de conexión. Esta información del servicio de red es almacenada en una o más repositorios de información a la que acceden con métodos de nomenclatura.

El proceso para establecer una sesión del cliente con la ayuda de un método de nomenclatura es como sigue:

1. El cliente inicia una solicitud de conexión proporcionando un identificador de conexión.

2. El identificador de conexión es resuelto a un descriptor de conexión por un método de nomenclatura. Esta información se vuelve al cliente.

3. El cliente realiza la solicitud de conexión a la dirección proporcionada por el descriptor de conexión.

4. El proceso escucha recibe la solicitud y la dirige al servidor de base de datos apropiada.

5. La conexión es aceptada por el servidor de base de datos.

Oracle Net provee los siguientes métodos de nomenclatura:

- Nomenclatura Local
- Nomenclatura de Directorios
- Nomenclatura de conexión sencilla
- Nomenclatura Externa

Nomenclatura Local

El método de nomenclatura local almacena los nombres de servicio de red y sus descriptores de conexión en un archivo de configuración denominado tnsnames.ora.

Nomenclatura de Directorios

El método de nomenclatura de directorios almacena identificadores de conexión en un servidor de directorios LDAP para acceder a un servicio de base de datos.

Nomenclatura de conexión sencilla

La nomenclatura de conexión sencilla permite a los clientes 10g conectarse con un servidor de base de datos Oracle usando una cadena de conexión TCP/IP que consiste del nombre del host y opcionalmente el puerto y el nombre del servicio.

```
CONNECT username/password@host[:port][/service_name]
```

La nomenclatura de conexión sencilla no requiere configuración.

Nomenclatura Externa

El método de nomenclatura externa almacena los nombres de servicio de red en un soporte de servicio de nomenclatura que no es de Oracle. Estos soportes de servicios de terceras personas incluyen:

- Servicio de Información de Red (NIS) Nombramiento externo
- Entorno de Computación Distribuida (Distributed Computing Environment) (DCE) Servicio de Directorio de Celdas (CDS)

Configuración del Método de Nomenclatura Local

El método de nomenclatura local adiciona nombres de servicios de red al archivo **tnsnames.ora**. Cada nombre de servicio de red apunta a un descriptor de conexión. El siguiente ejemplo muestra un nombre de servicio de red apuntando a un descriptor de conexión:

```
sales=(DESCRIPTION=
    (ADDRESS=
        (PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)
    )
    (CONNECT_DATA=
        (SERVICE_NAME=sales.us.acme.com)
    )
)
```

En este ejemplo, el nombre de servicio de red `ventas` esta apuntando al descriptor de conexión contenido en DESCRIPTION. DESCRIPTION contiene la dirección de protocolo e identifica el servicio de base de datos destino.

Es posible configurar la nomenclatura local durante o después de la instalación.

Configurar el archivo tnsname.ora durante la instalación

Oracle Net Configuration Assistant permite configurar los nombres de servicio de red para los clientes. Oracle Universal Installer ejecuta Oracle Net Configuration Assistant después de la instalación del software. La configuración varía dependiendo del modo de instalación.

Si realiza una instalación personalizada Oracle Net Configuration Assistant le permite seleccionar el método de nomenclatura a usar. Si selecciona Nomenclatura Local, le permitirá configurar el nombre del servicio de red en el archivo `tnsnames.ora` para conectarse con el servicio de base de datos Oracle.

Configurar el archivo tnsname.ora después de la instalación

Se puede agregar nombres de servicios de red al archivo tnsnames.ora en cualquier momento después de la instalación. Para configurar el método de nomenclatura local, puede utilizar una de las siguientes herramientas:

- Oracle Enterprise Manager
- Oracle Net Manager
- Oracle Net Configuration Assistant

Configuración del Proceso Escucha

El Proceso Escucha es un proceso separado que funciona en la computadora donde se encuentra el servidor de base de datos. Recibe solicitudes de conexión entrantes del cliente y maneja el tráfico de estas solicitudes al servidor de base de datos.

Porque todos los parámetros de configuración tienen valores por defecto, es posible comenzar y utilizar a un proceso escucha que no esta configurado. Este proceso escucha por defecto tiene por nombre LISTENER, no soporta servicios que se están iniciando, y escucha en la siguiente dirección de protocolo TCP/IP:

```
(ADDRESS=(PROTOCOL=tcp)(HOST=host_name)(PORT=1521))
```

Los servicios soportados, es decir, los servicios a los cuales el proceso escucha remite las solicitudes del cliente, se pueden configurar en el archivo de listener.ora o esta información se puede registrar dinámicamente en el proceso escucha. Esta característica de registro dinámico se llama **registro de servicio**. El registro es realizado por proceso PMON (un proceso de fondo de la instancia) de cada instancia de base de datos que tenga la configuración necesaria en el archivo del parámetro de inicialización de la base de datos. El registro dinámico del servicio no requiere ninguna configuración en el archivo de listener.ora.

Configuración del Proceso Escucha Durante la Instalación

Oracle Universal Installer ejecuta **Oracle Net Configuration Assistant** durante la instalación del software. Este programa le permite configurar la dirección de protocolo y el servicio que escucha para una base de datos de Oracle.

Durante la instalación de un servidor Enterprise Edition o Standard Edition, Oracle Net Configuration Assistant configura automáticamente el proceso escucha con el nombre LISTENER que tiene una dirección de protocolo TCP/IP que escucha para la base de datos Oracle. Durante una instalación personalizada, Oracle Net Configuration Assistant permite configurar el nombre del proceso escucha y la dirección de protocolo que uno elija.

Además, una dirección del protocolo IPC que escucha las llamadas a procedimientos externos es configuro automáticamente, sin tener en cuenta el tipo de la instalación.

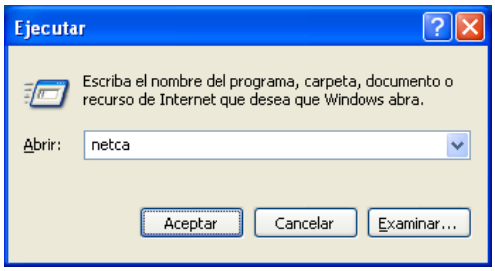
Oracle Net Configuration Assistant también configura automáticamente la información sobre el servicio para los procedimientos externos en el archivo listener.ora.

En el siguiente script tenemos un ejemplo

Ejemplo de un listener: listener.ora

```
LISTENER=
   (DESCRIPTION=
      (ADDRESS_LIST=
         (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
         (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))
      )
   )

SID_LIST_LISTENER=
   (SID_LIST=
      (SID_DESC=
         (SID_NAME=plsextproc)
         (ORACLE_HOME=/oracle10g)
         (PROGRAM=extproc)
      )
   )
```

Para ejecutar **Oracle Net Configuration Assistant** desde la ventana **Ejecutar** ejecute el programa **netca**, tal como se ilustra a continuación:



Esta herramienta nos permitirá realizar cualquier configuración del proceso escucha posterior a la instalación.

Glosario

ADDRESS

Dirección de protocolo de red del listener.

Cuando se realiza una conexión, el cliente y el listener se configuran con direcciones de protocolo idénticas. El cliente utiliza esta dirección para enviar la solicitud de conexión a una ubicación de listener concreta, mientras el listener "recibe" solicitudes de dicha dirección.

A continuación, se muestra una dirección de ejemplo:

```
(ADDRESS=
    (PROTOCOL=tcp)
    (HOST=sales-server)
    (PORT=1521)
)
```

Archivo tnsnames.ora

Archivo de configuración que contiene nombres de servicio de red asignados a descriptores de conexión. Este archivo se utiliza para el método de nomenclatura local.

El archivo tnsnames.ora debe residir en una de las siguientes ubicaciones:

- Directorio especificado por la variable de entorno TNS_ADMIN. Si la variable de entorno TNS_ADMIN no está definida como una variable en Windows NT, puede que esté en el registro.
- Directorio de configuración global del nodo. Para Sun Solaris, este directorio es `/var/opt/oracle`. Windows NT no tiene un directorio normal.
  El directorio `$ORACLE_HOME/network/admin` en UNIX o el directorio `ORACLE_HOME\network\admin` en sistemas operativos Windows.

Cadena de Conexión

(Connect String) Información que el usuario transfiere a un servicio para conectarse, como el nombre de usuario, la contraseña y el identificador de conexión:

Descriptor de Conexión

(Connect Descriptor) Descripción con formato especial del destino de una conexión de red. Un descriptor de conexión contiene información sobre el servicio destino y la ruta de red.

Descripción formateada de forma especial del destino de una conexión de red.

Los descriptores de conexión contienen el servicio de destino que se identifica mediante el nombre de servicio de las bases de datos Oracle10g, Oracle9i u Oracle8i o el identificador del sistema Oracle (SID) de las bases de datos Oracle8 u Oracle7.

Dirección de Protocolo

(Protocol Address) Identifica la dirección de red de un objeto de red.

Cuando se realiza una conexión, el cliente y el receptor de la solicitud, por ejemplo el listener, se configuran con direcciones de protocolo idénticas. El cliente utiliza esta dirección para enviar la solicitud de conexión a una ubicación concreta de un objeto de la red, mientras que el receptor "recibe" solicitudes de dicha dirección. Es importante instalar los mismos protocolos para el receptor y el cliente de la conexión, así como configurar las mismas direcciones.

La dirección de protocolo consta de un elemento denominado ADDRESS.

Identificador de Conexión

(Connect Identifier) Nombre, nombre de servicio de red o nombre de servicio que se resuelve en un descriptor de conexión. Los usuarios inician una solicitud de conexión mediante la transferencia de un nombre de usuario y una contraseña junto con un identificador en una cadena de conexión para el servicio al que desea conectarse

Método de Nomenclatura

(Naming Methods) Método de resolución utilizado por una aplicación cliente para resolver un identificador de conexión en un descriptor de conexión al conectarse a un servicio de la base de datos. La Red de Oracle soporta cuatro métodos de nomenclatura:

- Nomenclatura Local
- Nomenclatura de Directorios
- Nomenclatura de conexión sencilla (denominado anteriormente Nomenclatura del host)
- Nomenclatura Externa

Nombre de Servicio de Red

(Net Service Name) Nombre sencillo del servicio que se resuelve en un descriptor de conexión. Los usuarios inician una solicitud de conexión mediante el envío de un nombre de usuario y una contraseña junto con un identificador en una cadena de conexión para el servicio al que desea conectarse:

```
CONNECT usuario/contraseña@nombre_servicio_red
```

Según sus necesidades, los nombres de servicio de red se pueden almacenar en distintas ubicaciones, por ejemplo:

- Archivo de configuración local, tnsnames.ora, de cada cliente
- Servidor de directorios
- Oracle Names Server
- Servicio de nomenclatura externa, incluyendo Servicio de Información de Red (NIS) o Servicio de Directorio de Celdas (CDS)

Nomenclatura Local

(Local Naming) Método de nomenclatura que resuelve el nombre de servicio de red, almacenado en el archivo tnsnames.ora del cliente en un descriptor de conexión. La nomenclatura local es más aconsejable para redes de distribución simple con un número reducido de servicios que cambian en muy raras ocasiones.

Proceso Escucha

(Listener) Proceso que reside en el servidor cuya responsabilidad es la de recibir solicitudes de conexión de clientes y gestionar el tráfico al servidor.

Cada vez que un cliente solicita una sesión de red con un servidor, el listener recibe la solicitud real. Si la información del cliente coincide con la información del listener, éste otorga una conexión al servidor.

Registro de Servicio

(Service Registration) Función mediante la cual el proceso PMON (proceso en segundo plano de instancias) registra información automáticamente en el proceso escucha. Ya que esta información se registra en el proceso escucha, no es necesario configurar el archivo listener.ora con esta información estática.

El registro del servicio proporciona al proceso escucha la siguiente información:

- Nombre de servicio para cada instancia en ejecución de la base de datos
- Nombres de instancias de la base de datos
- Manejadores de servicios (distribuidores y servidor dedicado) disponibles para la instancia
- Información de carga de nodos, instancias y distribuidores

Esta información permite al proceso escucha determinar cómo se debe realizar el servicio de solicitud de conexión de un cliente.

Servicio de Directorio de Celdas (CDS)	Método de nomenclatura externa que permite a los usuarios utilizar de forma transparente las herramientas y aplicaciones de Oracle para acceder a las bases de datos Oracle en un entorno DCE (Distributed Computing Environment).
Servicio de Información de Red (NIS)	(Network Information Service) Protocolo cliente/servidor de las Páginas Amarillas (yp) de Sun Microsystems para la distribución de datos sobre la configuración del sistema (como los nombres de usuario y del host) entre las computadoras de una red.
Servidor Dedicado	(Dedicated Server) Proceso del servidor dedicado a una conexión del cliente.
SID	(Oracle System Identifier) Nombre que identifica una instancia concreta de una base de datos en ejecución. Por cada base de datos, hay al menos una instancia que hace referencia a ella. Para las bases de datos anteriores a Oracle8i, se utilizaba el SID para identificar la base de datos. El SID se incluía en la parte del descriptor de conexión del archivo tnsnames.ora y en la definición del listener en el archivo listener.ora. Los descriptores de conexión utilizados por los clientes para la conexión a la base de datos Oracle8 u Oracle7 deben estar configurados con el parámetro SID.

Lección 07

Administración de la Instancia

Contenido

- Parámetros de Inicialización 84
  - El archivo initSID.ORA (PFILE) 85
  - El Archivo SPFileSID.ORA (SPFILE) 87
  - Modificando Valores de los Parámetros 88
- Dimensionando el SGA 94
- Manejo de Sesiones 97
  - Sesión Restringida 97
  - Finalizar una Sesión 100
  - Terminar una Sesión 101
- Archivos de Alertas 103

Parámetros de Inicialización

Cuando se inicia la instancia el servidor Oracle tiene que hacer la lectura de los parámetros de inicialización desde el archivo de parámetros, para definir las estructuras del SGA, rutas de directorio, entre otros parámetros. Hay dos tipos de archivo de parámetros:

- Archivo de parámetro estático PFILE llamado comúnmente initSID.ORA donde SID es el identificador de la instancia
- Archivo de parámetros persistente, SPFILE comúnmente llamado como spfileSID.ORA.

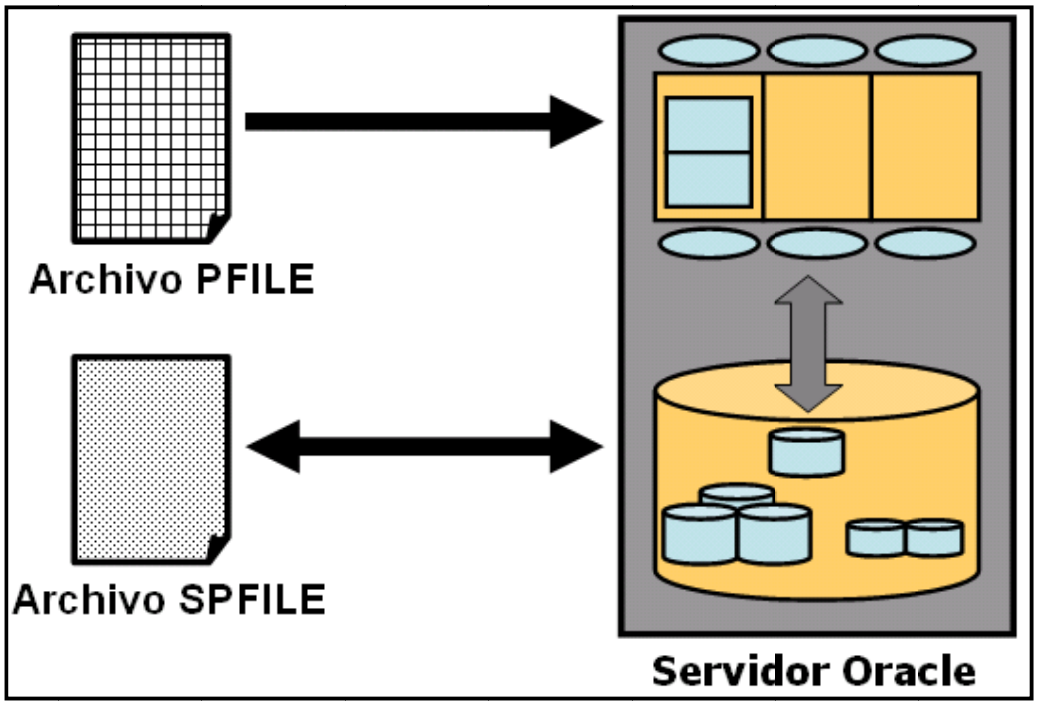


Figura 7 . 1 Estructura de PFILE y SPFILE

El archivo initSID.ORA (PFILE)

Es un archivo de texto que puede ser editado con un editor de de texto del sistema operativo, si el pfile es modificado para refrescar los cambios en la instancia esta debe ser reiniciada, algunos parámetros son dinámicos esto permite que sean modificados mientras esta activa la instancia sin embargo estos cambios no se reflejan en el pfile.

La ubicación por defecto de es archivo es: %ORACLE_HOME%\database.

A continuación se muestra un ejemplo de archivo PFILE.

```
###########################################################################
# Copyright (c) 1991, 2001, 2002 by Oracle Corporation
###########################################################################

##############################################
# Cache and I/O
##############################################
db_block_size=8192
db_file_multiblock_read_count=8

##############################################
# Job Queues
##############################################
job_queue_processes=10

##############################################
# File Configuration
##############################################
control_files=("C:\oracle\product\10.2.0/oradata/egcc/oradata\control01.ctl",
"C:\oracle\product\10.2.0/oradata/egcc/oradata\control02.ctl",
"C:\oracle\product\10.2.0/oradata/egcc/oradata\control03.ctl")

##############################################
# Cursors and Library Cache
##############################################
open_cursors=300

##############################################
# Diagnostics and Statistics
##############################################
background_dump_dest=C:\oracle\product\10.2.0/admin/egcc/bdump
core_dump_dest=C:\oracle\product\10.2.0/admin/egcc/cdump
user_dump_dest=C:\oracle\product\10.2.0/admin/egcc/udump

##############################################
# Database Identification
##############################################
```

```
db_domain="www.perudev.com"
db_name=egcc

###########################################
# SGA Memory
###########################################
sga_target=167772160

###########################################
# NLS
###########################################
nls_language="SPANISH"
nls_territory="SPAIN"

###########################################
# Processes and Sessions
###########################################
processes=150

###########################################
# System Managed Undo and Rollback Segments
###########################################
undo_management=AUTO
undo_tablespace=UNDOTBS1

###########################################
# Security and Auditing
###########################################
audit_file_dest=C:\oracle\product\10.2.0/admin/egcc/adump
remote_login_passwordfile=EXCLUSIVE

###########################################
# Shared Server
###########################################
dispatchers="(PROTOCOL=TCP) (SERVICE=egccXDB)"

###########################################
# Miscellaneous
###########################################
compatible=10.2.0.1.0

###########################################
```

```
# Sort, Hash Joins, Bitmap Indexes
##########################################
pga_aggregate_target=16777216
```

Reglas para especificar parámetros

- Todos los parámetros son opcionales
- Especificar los valores en el formato siguiente: parámetro = valor
- El servidor tiene valores por defecto para cada parámetro que dependen del SO
- Los parámetro pueden ser especificados en cualquier orden
- Los comentarios de línea van después del signo #
- Se puede incluir un archivo de parámetros dentro de otro con el parámetro IFILE
- Un parámetro con valores múltiples, los valores se deben especificar entre paréntesis separados con comas

El Archivo SPFileSID.ORA (SPFILE)

Es un archivo binario que es mantenido por el servidor Oracle, el spfile permite que los cambios persistan a los procesos de iniciar y bajar la base de datos.

Creación del archivo SPFILE

El SPFILE se crea a partir de un PFILE usando el comando CREATE SPFILE, este comando debe ser ejecutado ante de abrir la base de datos.

Sintaxis:

```
CREATE SPFILE [= 'nombre y ruta del archivo spfile']
    FROM PFILE [= 'nombre y ruta del archivo pfile'];
```

Ejemplo 1

```
SQL> conn / as sysdba              [Enter]
Connected.

SQL> shutdown immediate            [Enter]
Database closed.
Database dismounted.
ORACLE instance shut down.
```

```
SQL> create spfile from pfile;    [Enter]
File created.

SQL> startup                      [Enter]
ORACLE instance started.

Total System Global Area   72424008 bytes
Fixed Size                   453192 bytes
Variable Size              54525952 bytes
Database Buffers           16777216 bytes
Redo Buffers                 667648 bytes
Database mounted.
Database opened.
```

Modificando Valores de los Parámetros

Para realizar cambios de valor de uno ó más parámetros existen varias alternativas:

- Cambiando el archivo PFILE
- Cambiando con ALTER SESSION
- Cambiando con ALTER SYSTEM

Algunos parámetros son dinámicos, lo que quiere decir, que se pueden modificar su valor sin necesidad de bajar la base de datos.

Cambiando el archivo PFILE

Si realizamos cambios en el archivo PFILE, para que tengan efecto estos cambios debemos bajar la base de datos, crear el archivo SPFILE, y volver a cargar la base de datos.

Verificar si un parámetro es dinámico

Tenemos las vistas V$PARAMETER y V$SYSTEM_PARAMETER para consultar información sobre la modificación de los parámetros. En el siguiente cuadro se explica la información de dos columnas sobre los parámetros.

ISSES_MODIFIABLE	Indica Si el parámetro puede ser modificado con ALTER SESSION.
ISSYS_MODIFIABLE	Indica Si el parámetro puede ser modificado con ALTER SYSTEM.

Ejemplo 2

```
SQL> column name format a40            [Enter]

SQL> select name, isses_modifiable, issys_modifiable
  2  from v$system_parameter
  3  where issys_modifiable != 'FALSE'
  4  order by 1;                       [Enter]

NAME                                     ISSES ISSYS_MOD
---------------------------------------- ----- ---------
aq_tm_processes                          FALSE IMMEDIATE
archive_lag_target                       FALSE IMMEDIATE
asm_diskgroups                           FALSE IMMEDIATE
asm_diskstring                           FALSE IMMEDIATE
asm_power_limit                          TRUE  IMMEDIATE
audit_file_dest                          FALSE DEFERRED
background_dump_dest                     FALSE IMMEDIATE
backup_tape_io_slaves                    FALSE DEFERRED
circuits                                 FALSE IMMEDIATE
commit_write                             TRUE  IMMEDIATE
control_file_record_keep_time            FALSE IMMEDIATE
core_dump_dest                           FALSE IMMEDIATE
cpu_count                                FALSE IMMEDIATE
create_stored_outlines                   TRUE  IMMEDIATE
cursor_sharing                           TRUE  IMMEDIATE
db_block_checking                        TRUE  IMMEDIATE
db_block_checksum                        FALSE IMMEDIATE
db_cache_advice                          FALSE IMMEDIATE
db_cache_size                            FALSE IMMEDIATE
db_create_file_dest                      TRUE  IMMEDIATE
db_create_online_log_dest_1              TRUE  IMMEDIATE
db_create_online_log_dest_2              TRUE  IMMEDIATE
db_create_online_log_dest_3              TRUE  IMMEDIATE
db_create_online_log_dest_4              TRUE  IMMEDIATE
db_create_online_log_dest_5              TRUE  IMMEDIATE
db_file_multiblock_read_count            TRUE  IMMEDIATE
db_flashback_retention_target            FALSE IMMEDIATE
db_keep_cache_size                       FALSE IMMEDIATE
db_recovery_file_dest                    FALSE IMMEDIATE
db_recovery_file_dest_size               FALSE IMMEDIATE
db_recycle_cache_size                    FALSE IMMEDIATE
db_16k_cache_size                        FALSE IMMEDIATE
db_2k_cache_size                         FALSE IMMEDIATE
db_32k_cache_size                        FALSE IMMEDIATE
db_4k_cache_size                         FALSE IMMEDIATE
db_8k_cache_size                         FALSE IMMEDIATE
ddl_wait_for_locks                       TRUE  IMMEDIATE
dg_broker_config_file1                   FALSE IMMEDIATE
dg_broker_config_file2                   FALSE IMMEDIATE
dg_broker_start                          FALSE IMMEDIATE
dispatchers                              FALSE IMMEDIATE
drs_start                                FALSE IMMEDIATE
fal_client                               FALSE IMMEDIATE
fal_server                               FALSE IMMEDIATE
fast_start_io_target                     FALSE IMMEDIATE
fast_start_mttr_target                   FALSE IMMEDIATE
fast_start_parallel_rollback             FALSE IMMEDIATE
file_mapping                             FALSE IMMEDIATE
fixed_date                               FALSE IMMEDIATE
global_names                             TRUE  IMMEDIATE
hs_autoregister                          FALSE IMMEDIATE
java_pool_size                           FALSE IMMEDIATE
job_queue_processes                      FALSE IMMEDIATE
```

```
large_pool_size                          FALSE IMMEDIATE
ldap_directory_access                    FALSE IMMEDIATE
license_max_sessions                     FALSE IMMEDIATE
license_max_users                        FALSE IMMEDIATE
license_sessions_warning                 FALSE IMMEDIATE
local_listener                           FALSE IMMEDIATE
log_archive_config                       FALSE IMMEDIATE
log_archive_dest                         FALSE IMMEDIATE
log_archive_dest_state_1                 TRUE  IMMEDIATE
log_archive_dest_state_10                TRUE  IMMEDIATE
log_archive_dest_state_2                 TRUE  IMMEDIATE
log_archive_dest_state_3                 TRUE  IMMEDIATE
log_archive_dest_state_4                 TRUE  IMMEDIATE
log_archive_dest_state_5                 TRUE  IMMEDIATE
log_archive_dest_state_6                 TRUE  IMMEDIATE
log_archive_dest_state_7                 TRUE  IMMEDIATE
log_archive_dest_state_8                 TRUE  IMMEDIATE
log_archive_dest_state_9                 TRUE  IMMEDIATE
log_archive_dest_1                       TRUE  IMMEDIATE
log_archive_dest_10                      TRUE  IMMEDIATE
log_archive_dest_2                       TRUE  IMMEDIATE
log_archive_dest_3                       TRUE  IMMEDIATE
log_archive_dest_4                       TRUE  IMMEDIATE
log_archive_dest_5                       TRUE  IMMEDIATE
log_archive_dest_6                       TRUE  IMMEDIATE
log_archive_dest_7                       TRUE  IMMEDIATE
log_archive_dest_8                       TRUE  IMMEDIATE
log_archive_dest_9                       TRUE  IMMEDIATE
log_archive_duplex_dest                  FALSE IMMEDIATE
log_archive_local_first                  FALSE IMMEDIATE
log_archive_max_processes                FALSE IMMEDIATE
log_archive_min_succeed_dest             TRUE  IMMEDIATE
log_archive_trace                        FALSE IMMEDIATE
log_checkpoint_interval                  FALSE IMMEDIATE
log_checkpoints_to_alert                 FALSE IMMEDIATE
log_checkpoint_timeout                   FALSE IMMEDIATE
max_dispatchers                          FALSE IMMEDIATE
max_dump_file_size                       TRUE  IMMEDIATE
max_shared_servers                       FALSE IMMEDIATE
nls_length_semantics                     TRUE  IMMEDIATE
nls_nchar_conv_excp                      TRUE  IMMEDIATE
object_cache_max_size_percent            TRUE  DEFERRED
object_cache_optimal_size                TRUE  DEFERRED
olap_page_pool_size                      TRUE  DEFERRED
open_cursors                             FALSE IMMEDIATE
optimizer_dynamic_sampling               TRUE  IMMEDIATE
optimizer_features_enable                TRUE  IMMEDIATE
optimizer_index_caching                  TRUE  IMMEDIATE
optimizer_index_cost_adj                 TRUE  IMMEDIATE
optimizer_mode                           TRUE  IMMEDIATE
optimizer_secure_view_merging            FALSE IMMEDIATE
parallel_adaptive_multi_user             FALSE IMMEDIATE
parallel_instance_group                  TRUE  IMMEDIATE
parallel_max_servers                     FALSE IMMEDIATE
parallel_min_servers                     FALSE IMMEDIATE
parallel_threads_per_cpu                 FALSE IMMEDIATE
pga_aggregate_target                     FALSE IMMEDIATE
plsql_ccflags                            TRUE  IMMEDIATE
plsql_code_type                          TRUE  IMMEDIATE
plsql_compiler_flags                     TRUE  IMMEDIATE
plsql_debug                              TRUE  IMMEDIATE
plsql_native_library_dir                 FALSE IMMEDIATE
plsql_native_library_subdir_count        FALSE IMMEDIATE
plsql_optimize_level                     TRUE  IMMEDIATE
plsql_v2_compatibility                   TRUE  IMMEDIATE
```

```
plsql_warnings                          TRUE  IMMEDIATE
query_rewrite_enabled                   TRUE  IMMEDIATE
query_rewrite_integrity                 TRUE  IMMEDIATE
recyclebin                              TRUE  IMMEDIATE
remote_dependencies_mode                TRUE  IMMEDIATE
remote_listener                         FALSE IMMEDIATE
resource_limit                          FALSE IMMEDIATE
resource_manager_plan                   FALSE IMMEDIATE
resumable_timeout                       TRUE  IMMEDIATE
service_names                           FALSE IMMEDIATE
sga_target                              FALSE IMMEDIATE
shared_pool_size                        FALSE IMMEDIATE
shared_servers                          FALSE IMMEDIATE
shared_server_sessions                  FALSE IMMEDIATE
skip_unusable_indexes                   TRUE  IMMEDIATE
smtp_out_server                         TRUE  IMMEDIATE
sort_area_retained_size                 TRUE  DEFERRED
sort_area_size                          TRUE  DEFERRED
sql_trace                               TRUE  IMMEDIATE
sqltune_category                        TRUE  IMMEDIATE
standby_archive_dest                    FALSE IMMEDIATE
standby_file_management                 FALSE IMMEDIATE
star_transformation_enabled             TRUE  IMMEDIATE
statistics_level                        TRUE  IMMEDIATE
streams_pool_size                       FALSE IMMEDIATE
thread                                  FALSE IMMEDIATE
timed_os_statistics                     TRUE  IMMEDIATE
timed_statistics                        TRUE  IMMEDIATE
trace_enabled                           FALSE IMMEDIATE
undo_retention                          FALSE IMMEDIATE
undo_tablespace                         FALSE IMMEDIATE
user_dump_dest                          FALSE IMMEDIATE
workarea_size_policy                    TRUE  IMMEDIATE

151 filas seleccionadas.
```

Modificando Parámetros con ALTER SESSION

Sintaxis:

ALTER SESSION SET parámetro = valor;

Ejemplo 3

```
SQL> conn system/manager                                [Enter]
Connected.

SQL> show parameter session_cached_cursors              [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
session_cached_cursors               integer     0

SQL> alter session set  session_cached_cursors = 30;    [Enter]

Session altered.
```

```
SQL>  show parameter session_cached_cursors            [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------------------
session_cached_cursors               integer     30
```

Modificando Parámetros con ALTER SYSTEM

Sintaxis:

```
ALTER SYSTEM SET parámetro = valor
   [DEFERRED] [SCOPE={MEMORY|SPFILE|BOTH}];
```

Ejemplo 4

```
SQL> conn / as sysdba                        [Enter]
Connected.

SQL> show parameter open_cursors             [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- -----------------------
open_cursors                         integer     300

SQL> alter system set open_cursors = 200;    [Enter]
System altered.

SQL> show parameter open_cursors             [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- -----------------------
open_cursors                         integer     200
```

Ejemplo 5

```
SQL> conn / as sysdba                                        [Enter]
Connected.

SQL> show parameter java_pool_size;                          [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
java_pool_size                       big integer 20971520

SQL> alter system set java_pool_size = 0 scope = spfile;     [Enter]
System altered.
```

```
SQL> show parameter java_pool_size;                          [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- -----------------------
java_pool_size                       big integer 20971520

SQL> shutdown immediate                                      [Enter]
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup                                                 [Enter]
ORACLE instance started.

Total System Global Area   72424008 bytes
Fixed Size                   453192 bytes
Variable Size              54525952 bytes
Database Buffers           16777216 bytes
Redo Buffers                 667648 bytes
Database mounted.
Database opened.

SQL> show parameter java_pool_size;                          [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- -----------------------
java_pool_size                       big integer 0
```

Dimensionando el SGA

El tamaño de SGA pude variar sin tener que reiniciar el servidor, es por ello que se define un parámetro llamado SGA_MAX_SIZE y SGA_TARGET para limitar el crecimiento.

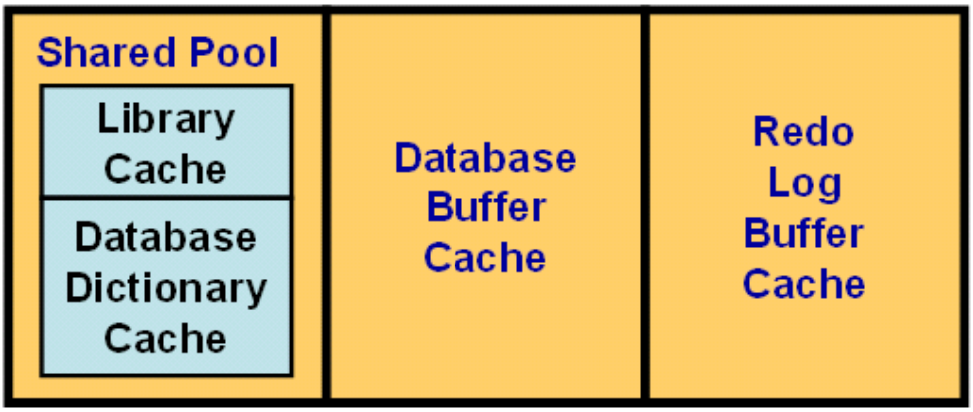


Figura 7 . 2 El SGA

El tamaño del SGA esta determinado por los siguientes parámetros de inicialización:

SGA_MAX_SIZE	Establece el máximo tamaño que puede alojar el SGA cuando se levanta la instancia de base de datos. Este parámetro permitirá aumentar el tamaño de la SGA sin necesidad de iniciar la instancia, teniendo en cuenta que el total del SGA no exceda este parámetro.
SGA_TARGET	(Oracle 10g r1) Especifica el total de tamaño que dispondrá la SGA cuando la instancia se inicia. Si utilizamos este parámetro no tendremos necesidad de definir los valores para db_cache_size, shared_pool_size, large_pool_size, java_pool_size puesto que oracle automáticamente ajusta estos componentes incluyendo stream_pool_size.
DB_CACHE_SIZE	Determina el tamaño del Database Buffer Cache en bytes.
LOG_BUFFER	Determina el número de bytes del Redo Log Buffer Cache.
SHARED_POOL_SIZE	Especifica el tamaño en bytes del Shared Pool.

STREAMS_POOL_SIZE	Determina el tamaño de Oracle Streams, que le permite a la versión EE compartir datos y eventos en un ambiente distribuido. Cuando el parámetro STREAMS_POOL_SIZE no es asignado, se utiliza memoria del shared pool, hasta un 10% de su tamaño.
LARGE_POOL_SIZE	Indica el tamaño en bytes del Large Pool por defecto es cero.
DB_BLOCK_SIZE	Determina el tamaño del bloque Oracle primario en bytes, el bloque Oracle es la unidad de lectura o escritura.

El SGA como estructura dinámica utiliza una unidad de granularida el cual es un espacio contiguo de memoria, esta unidad depende del tamaño definido para el SGA, así la unidad de granularida es 4MB si el SGA es menor o igual que 1 GB y 16 MB si el SGA es mayor a 1 GB.

El tamaño de las estructuras del SGA puede cambiar en proporciones múltiplos de la unidad de granularida, podemos consultar V$SGA_DYNAMIC_COMPONENTS para consultar los componentes dinámicos del SGA y el tamaño granular utilizado.

Ejemplo 6

```
SQL> select component, current_size, granule_size
  2  from v$sga_dynamic_components;                    [Enter]

COMPONENT                           CURRENT_SIZE GRANULE_SIZE
----------------------------------- ------------ ------------
shared pool                             62914560      4194304
large pool                               4194304      4194304
java pool                                4194304      4194304
streams pool                                   0      4194304
DEFAULT buffer cache                    88080384      4194304
KEEP buffer cache                              0      4194304
RECYCLE buffer cache                           0      4194304
DEFAULT 2K buffer cache                        0      4194304
DEFAULT 4K buffer cache                        0      4194304
DEFAULT 8K buffer cache                        0      4194304
DEFAULT 16K buffer cache                       0      4194304
DEFAULT 32K buffer cache                       0      4194304
ASM Buffer Cache                               0      4194304

13 filas seleccionadas.


SQL> show parameter db_cache_size                      [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- --------------
db_cache_size                        big integer 44M

SQL> alter system set db_cache_size = 47M;             [Enter]

Sistema modificado.

SQL> show parameter db_cache_size                      [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- ----------------
db_cache_size                        big integer 48M
```

Manejo de Sesiones

Sesión Restringida

La sesión restringida es muy útil, por ejemplo, cuando realizamos el mantenimiento de la estructura, ó realizamos tareas de importación ó exportación de la base de datos.

Cuando realizamos tareas de mantenimiento como administradores debemos asegurarnos de que seamos lo únicos conectados a la base de datos, para eso debemos terminar las sesiones de otros usuarios de manera explicita.

Habilitar – Deshabilitar el acceso restringido a la base de datos

Para restringir el acceso a la base de datos se puede usar el comando ALTER SYSTEM.

Sintaxis:

```
ALTER SYSTEM {ENABLE | DISABLE} RESTRICTED SESSION
```

Ejemplo 7

```
SQL> conn / as sysdba                              [Enter]
Connected.

SQL> select logins from v$instance;                [Enter]

LOGINS
----------
ALLOWED

SQL> alter system enable  restricted session;      [Enter]
System altered.

SQL> select logins from v$instance;                [Enter]

LOGINS
----------
RESTRICTED

SQL> conn scott/tiger                              [Enter]
ERROR:
ORA-01035: ORACLE only available to users with RESTRICTED SESSION privilege


Warning: You are no longer connected to ORACLE.
```

También podemos restringir el acceso a la base de datos con el comando STARTUP, cuando iniciamos la base de datos.

Sintaxis:

```
STARTUP RESTRICT
```

Ejemplo 8

```
SQL> conn / as sysdba                      [Enter]
Connected.

SQL> shutdown immediate;                   [Enter]
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup restrict                      [Enter]
ORACLE instance started.

Total System Global Area   93395628 bytes
Fixed Size                   453292 bytes
Variable Size              75497472 bytes
Database Buffers           16777216 bytes
Redo Buffers                 667648 bytes
Database mounted.
Database opened.

SQL> select logins from v$instance;        [Enter]

LOGINS
----------
RESTRICTED
```

Quienes pueden iniciar sesión en modo restringido

Para poder iniciar sesión en modo restringido, los usuarios deben tener los siguientes privilegios del sistema:

- SYSDBA
- SYSOPER
- RESTRICTED SESSION

El rol DBA también concede el privilegio RESTRICTED SESSION, por lo tanto, los usuarios a los que se asigna este rol también podrán iniciar sesión en modo restringido.

Inicialmente los usuarios que pueden conectarse en modo restringido son:

- SYS
- SYSTEM

Conceder y Revocar el privilegio RESTRCITED SESSION

Conceder privilegio RESTRICTED SESSION

Sintaxis:

GRANT privilegio/rol TO usuario

Ejemplo 9

```
SQL> conn / as sysdba                    [Enter]
Connected.

SQL> grant restricted session to scott;  [Enter]
Grant succeeded.

SQL> conn scott/tiger                    [Enter]
Connected.
```

Revocar privilegio RESTRICTED SESSION

Sintaxis:

REVOKE privilegio/rol FROM usuario

Ejemplo 10

```
SQL> conn / as sysdba                      [Enter]
Connected.

SQL> revoke restricted session from scott; [Enter]
Revoke succeeded.

SQL> conn scott/tiger                      [Enter]
ERROR:
ORA-01035: ORACLE only available to users with RESTRICTED SESSION privilege


Warning: You are no longer connected to ORACLE.
```

Finalizar una Sesión

Sintaxis:

ALTER SYSTEM DISCONNECT SESSION 'sid,serial#'
{POST_TRANSACTION | IMMEDIATE}

Ejemplo 11

```
SQL> select sid,serial#,username,status,BLOCKING_SESSION
   2 from v$session;                                        [Enter]

   SID    SERIAL# USERNAME        STATUS   BLOCKING_SESSION
------ ---------- --------------- -------- ----------------
   141         18 SCOTT           ACTIVE                144
   144         16 SCOTT           INACTIVE
   146         70                 ACTIVE
   148         18 SCOTT           ACTIVE                144
   149          2                 ACTIVE
   152         11 SYS             ACTIVE
   154          1                 ACTIVE
   158          4                 ACTIVE
   160          1                 ACTIVE
   161          1                 ACTIVE
   162          1                 ACTIVE
   163          1                 ACTIVE
   164          1                 ACTIVE
   165          1                 ACTIVE
   166          1                 ACTIVE
   167          1                 ACTIVE
   168          1                 ACTIVE
   169          1                 ACTIVE
   170          1                 ACTIVE

19 filas seleccionadas.

SQL> alter system
   2 disconnect session '144,16' post_transaction;        [Enter]
System altered.
```

Si scott después de terminar transacción intenta hacer otra operación, Oracle le mostrará el siguiente mensaje de error:

ERROR at line 1:

ORA-00028: your session has been killed

Terminar una Sesión

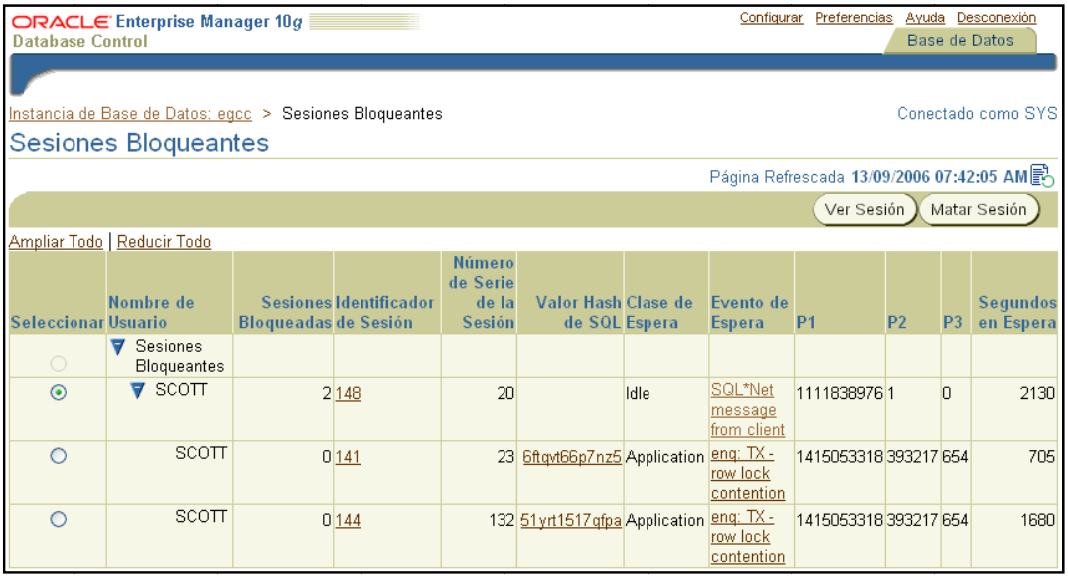
Sintaxis:

ALTER SYSTEM KILL SESSION 'sid,serial#' [IMMEDIATE]

Ejemplo 12

```
SQL> select SID,SERIAL#,USERNAME,status from v$session;   [Enter]

       SID    SERIAL# USERNAME                       STATUS
---------- ---------- ------------------------------ --------
         1          1                                ACTIVE
         2          1                                ACTIVE
         3          1                                ACTIVE
         4          1                                ACTIVE
         5          1                                ACTIVE
         6          1                                ACTIVE
         7          1                                ACTIVE
         8          1                                ACTIVE
         9          5 SYS                            ACTIVE
        12        195 SCOTT                          INACTIVE

10 rows selected.

SQL> alter system kill session '12,195' immediate;        [Enter]

System altered.

SQL> select SID,SERIAL#,USERNAME,status from v$session;   [Enter]

       SID    SERIAL# USERNAME                       STATUS
---------- ---------- ------------------------------ --------
         1          1                                ACTIVE
         2          1                                ACTIVE
         3          1                                ACTIVE
         4          1                                ACTIVE
         5          1                                ACTIVE
         6          1                                ACTIVE
         7          1                                ACTIVE
         8          1                                ACTIVE
         9          5 SYS                            ACTIVE
        12        195 SCOTT                          KILLED

10 rows selected.
```

También podemos revisar los bloqueos de las sesiones con la herramienta Oracle EM, tal como se muestra en la siguiente figura.



Seleccionar	Nombre de Usuario	Sesiones Bloqueadas	Identificador de Sesión	Número de Serie de la Sesión	Valor Hash de SQL	Clase de Espera	Evento de Espera	P1	P2	P3	Segundos en Espera
	Sesiones Bloqueantes										
	SCOTT	2	148	20		Idle	SQL*Net message from client	1111838976	1	0	2130
	SCOTT	0	141	23	6ftqvt66p7nz5	Application	enq: TX - row lock contention	1415053318	393217	654	705
	SCOTT	0	144	132	51yrt1517qfpa	Application	enq: TX - row lock contention	1415053318	393217	654	1680

Figura 7 . 3 Bloqueos de cesiones.

Archivos de Alertas

El archivo de alertas alert_SID.log registra los comandos y resultados de los principales eventos mientras la base de datos esta operativa, cada instancia tiene una archivo de alertas, sino existe la instancia crea uno durante el inicio, este archivo crece día a día así que debe ser revisarlo y darle mantenimiento, puedes optar por eliminar parte del contenido ó eliminarlo periódicamente previo backup si deseas guardar la información histórica del archivo de alertas.

El Archivo de alertas contiene la siguiente información:

- Fecha y Hora de inicio y parada de la base de datos
- Una lista de los parámetros de inicialización que no tiene valores por defecto
- El inicio de los procesos background
- El thread de inicio de la instancia
- El numero de secuencia de log Writer
- Información de los switch de los log files
- Creación de los tablespaces y segmentos de UNDO
- Sentencias de tipo ALTER que han sido utilizadas
- Información relacionada con el error de tipo ORA-600

La localización del archivo de alerta la define el parámetro BACKGROUND_DUMP_DEST.

Ejemplo 13

1. Bajar la base de datos.

```
SQL> shutdown immediate      [Enter]
Database closed.
Database dismounted.
ORACLE instance shut down.
```

2. Eliminar el archivo de alertas.

3. Iniciar la base de datos.

```
SQL> startup
ORACLE instance started.

Total System Global Area   72424008 bytes
Fixed Size                   453192 bytes
Variable Size              54525952 bytes
Database Buffers           16777216 bytes
Redo Buffers                 667648 bytes
Database mounted.
Database opened.
```

4. Revisar el archivo de alertas

Lección 08

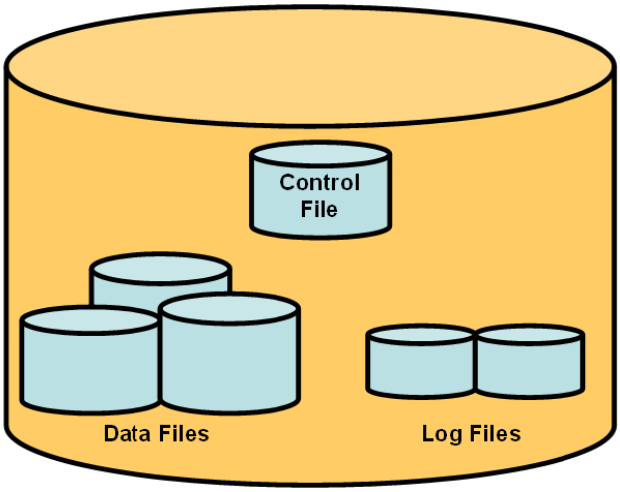
Administración del Archivo de Control

Contenido

- ¿Qué es el Archivo de Control? 106
- Establecer el Nombre de los Archivos de Control 107
- Multiplexado el Archivo de Control 108
- Creando Copias Adicionales, Renombrando y Reubicando Archivos de Control 109
- Mostrando Información del Archivo de Control 111

¿Qué es el Archivo de Control?

Toda base de datos tiene un archivo de control, qué es un archivo binario donde se registra la estructura física de la base de datos.



El archivo de control contiene:

- El nombre de la base de datos
- Nombre y ubicación de los data files y redo log files
- Fecha y hora de la creación de la base de datos
- El número actual de la secuencia del log
- Información de los checkpoints

El archivo de control debe estar disponible para escritura por el servidor de base de datos siempre que la base de datos esté abierta. Sin el archivo de control, la base de datos no puede ser montada y es difícil recuperarla.

El archivo de control de una base de datos se crea al mismo tiempo que la base de datos. Por defecto, por lo menos una copia del archivo de control se crea durante la creación de base de datos. En algunos sistemas operativos por defecto es crear copias múltiples. Debes crear dos o más copias del archivo de control durante la creación de base de datos. Puedes también crear archivos de control más adelante, si pierdes archivos de control o deseas cambiar valores particulares en los archivos de control.

Establecer el Nombre de los Archivos de Control

Los nombres de los archivos de control se especifican en el parámetro de inicialización CONTROL_FILES en el archivo de parámetros de la base de datos. La instancia reconoce y abre toda la lista de archivos durante la puesta en marcha, y la instancia escribe y mantiene toda la lista de archivos de control durante la operación de la base de datos.

Ejemplo 1

```
SQL> show parameter control_files         [Enter]

NAME                     TYPE        VALUE
------------------------ ----------- ------------------------------
control_files            string      C:\ORACLE\PRODUCT\10.2.0\ORADA
                                     TA\EGCC\CONTROL01.CTL, C:\ORAC
                                     LE\PRODUCT\10.2.0\ORADATA\EGCC
                                     \CONTROL02.CTL, C:\ORACLE\PROD
                                     UCT\10.2.0\ORADATA\EGCC\CONTRO
                                     L03.CTL
```

Si no especifica los archivos en el parámetro CONTROL_FILES antes de la creación de base de datos:

- Si no esta utilizando el Administrador de Archivos de Oracle (Oracle-Managed Files), entonces la base de datos crea un archivo de control y usa un nombre de archivo por defecto. El nombre por defecto depende del sistema operativo.
- Si estas utilizando el Administrador de Archivos de Oracle, entonces debe establecer el parámetro de inicialización que determina el nombre y la ubicación de los archivos de control,
- Si estas utilizando el Administrador de Almacenamiento Automático (Automatic Storage Management), puedes poner nombres de archivos incompletos en ASM en los parámetros de inicialización DB_CREATE_FILE_DEST y DB_RECOVERY_FILE_DEST. ASM crea automáticamente los archivos de control en los lugares apropiados.

Multiplexado el Archivo de Control

Cada base de datos debe tener por lo menos 2 archivos de control, cada uno almacenado en diferentes discos. Si un archivo de control es dañado debido a una falla del disco, la instancia asociada debe ser cerrada. Una vez que se repare el disco, el archivo de control dañado se puede restaurar usando la copia intacta del archivo de control del otro disco y la instancia puede ser reiniciada. En este caso, no se requiere ninguna recuperación de los medios.

El comportamiento de los archivos de control multiplexados es como se describe a continuación:

- La base de datos escribe los nombres de archivo de control en el parámetro CONTROL_FILES, en el archivo del parámetro de inicialización de la base de datos.
- La base de datos lee solamente el primer archivo en el parámetro CONTROL_FILES durante la operación de la base de datos.
- Si uno de los archivos de control llega a ser inasequible durante la operación de la base de datos, la instancia llega a ser inoperable y debe ser abortada.

Nota

Oracle recomienda fuertemente que toda base de datos tenga un mínimo de dos archivos de control y que estén situados en discos físicos separados.

Una forma para multiplexar archivos de control es almacenar una copia del archivo de control en discos diferentes. Almacenando archivos de control en discos diferentes, reducimos al mínimo el riesgo cuando falle uno de los discos. Lo que se busca es evitar hacer un proceso de recovery.

Creando Copias Adicionales, Renombrando y Reubicando Archivos de Control

Se puede crear una copia adicional del archivo de control para multiplexarlo, copiando un archivo de control existente a una nueva ubicación y agregando el nombre del archivo a la lista de los archivos de control. De manera similar, podemos renombrar un archivo de control existente cambiando el nombre del archivo o cambiando su ubicación, y cambiando el nombre del archivo en la lista de archivos de control. En ambos casos, para garantizar que los archivos de control no cambian durante el proceso, cerrar la base de datos antes de copiar el archivo de control.

Para agregar una copia multiplexada del archivo de control actual o renombrar un archivo de control, debe seguir los siguientes pasos:

1. Cerrar la base de datos
2. Copie un archivo de control existente a una nueva ubicación, usando comandos del Sistema Operativo.
3. Corrija el parámetro CONTROL_FILES en el archivo de parámetro de inicialización de base de datos para añadir el nuevo nombre del archivo de control, o cambiar el nombre del archivo de control existente.
4. Inicie de nuevo la base de datos.

Ejemplo 2

1. Cerramos la base de datos.

```
SQL> shutdown immediate    [Enter]
Base de datos cerrada.
Base de datos desmontada.
Instancia ORACLE cerrada.
```

2. Realizamos copia del archivo de control en C:\CTLFILE y D:\CTLFILE.

3. Modificamos el parámetro CONTROL_FILES en el archivo de parámetros de inicialización.

```
SQL> startup nomount                          [Enter]
Instancia ORACLE iniciada.

Total System Global Area  167772160 bytes
Fixed Size                  1247876 bytes
Variable Size              71304572 bytes
Database Buffers           88080384 bytes
Redo Buffers                7139328 bytes

SQL> alter system set
  2  control_files='c:\ctlfile\control01.ctl','d:\ctlfile\control02.ctl'
  3  scope=spfile;                              [Enter]

Sistema modificado.
```

4. Iniciamos nuevamente la base de datos.

```
SQL> shutdown immediate                       [Enter]
ORA-01507: base de datos sin montar

Instancia ORACLE cerrada.

SQL> startup                                  [Enter]
Instancia ORACLE iniciada.

Total System Global Area  167772160 bytes
Fixed Size                  1247876 bytes
Variable Size              71304572 bytes
Database Buffers           88080384 bytes
Redo Buffers                7139328 bytes
Base de datos montada.
Base de datos abierta.

SQL> show parameter control_files             [Enter]

NAME                 TYPE        VALUE
-------------------- ----------- ------------------------------
control_files        string      C:\CTLFILE\CONTROL01.CTL, D:\C
                                 TLFILE\CONTROL02.CTL
```

Mostrando Información del Archivo de Control

Las siguientes vistas muestran información acerca de los archivos de control:

Vista	Descripción
V$DATABASE	Muestra información de la base de datos desde el archivo de control.
V$CONTROLFILE	Lista los nombres de los archivos de control
V$CONTROLFILE_RECORD_SECTION	Muestra las distintas secciones que componen el archivo de control y los registros de los que consta cada una; así como el número de registros de cada tipo usados hasta el momento.
V$PARAMETER	Puede ser utilizada para mostrar nombres de los archivos de control como fueron especificadas en el parámetro de inicialización CONTROL FILES.

Ejemplo 3

```
SQL> select name from v$controlfile;      [Enter]

NAME
------------------------------
C:\CTLFILE\CONTROL01.CTL
D:\CTLFILE\CONTROL02.CTL
```

Página en Blanco

Lección 09

Administración del los Redo Logs

Contenido

- ¿Que es el Redo Log? 114
  - Redo Threads 114
  - Contenido del Redo Log 114
  - Como Oracle Database Graba en los Redo Log 115
- Planificación de los Redo Log Files 118
  - Multiplexado de los Redo Log Files 118
- Creando Groupos y Miembros Redo Log 119
  - Creando Grupos Redo Log 119
  - Creando Miembros Redo Log 120
- Eliminando Grupos y Miembros Redo Log 121
  - Eliminando Grupos Redo Log 121
  - Eliminando Miembros Redo Log 121
- Consultando Información sobre los Redo Log 123
- Ejercicio 123

¿Que es el Redo Log?

La estructura de mayor importancia para operaciones de recuperación es el **Redo Log**, que consiste en dos o más archivos preasignados que almacenan todos los cambios hechos a la base de datos tal como ellos ocurren. Cada instancia de una Base de datos de Oracle tiene un redo log asociado para proteger la base de datos en caso de que la instancia falle.

Redo Threads

Cada instancia de base de datos tiene su propio **redo log groups**. Estos redo log groups, multiplexados o no, llaman a un **thread** de la instancia de redo. En configuraciones típicas, sólo una instancia de base de datos tiene acceso a una Base de datos de Oracle, tan sólo un hilo está presente. En Oracle Real Application Clusters, sin embargo, dos o más instancias simultáneamente tiene acceso a una sola base de datos y cada instancia tiene su propio thread de redo.

En esta lección describiré como configurar y administrar el redo log en una instancia de Base de datos de Oracle estándar. El número de thread que puede ser asumido es 1 en todas las discusiones y los ejemplos desarrollados.

Contenido del Redo Log

Los Redo Log Files son llenados con redo records. Un redo record, también llamado un redo entry, está compuesto de un grupo de vectores de cambio[1], cada uno de los cuales es una descripción de un cambio hecho a un solo bloque en la base de datos. Por ejemplo, si cambia un valor del sueldo en una tabla de empleados, genera un redo record que contiene los vectores de cambio que describen los cambios al bloque del segmento de datos para la tabla, el bloque de datos de segmento deshacer, y la tabla de transacción de los segmentos deshacer.

Los redo records son guardados de una manera circular en el redo log buffer del SGA y son almacenados en uno de los redo log files por el proceso de fondo de la base de datos Log Writer (LGWR). Siempre que una transacción sea confirmada, LGWR graba los redo records de la transacción desde el redo log buffer del SGA a un redo log file, y asigna un SYSTEM CHANGE NUMBER (SCN) para identificar los redo records para cada transacción confirmada. Sólo cuando todos los redo records asociados con una transacción dada están seguros en disco el proceso de usuario notifica que la transacción ha sido confirmada.

Los redo records también puede ser grabados a un redo log file antes de que la transacción correspondiente sea confirmada. Si el redo log buffer se llena, u otra transacción es confirmada, LGWR envía todos los redo log entries en el redo log buffer a un redo log file,

[1] **Vector de Cambio:** Describe un cambio simple en un bloque de datos de la BD. Entre otros datos, contiene el número de versión, el código de la transacción, y la dirección del bloque afectado.

aun cuando algunos redo records puede no ser confirmados. Si fuera necesario, la base de datos puede cancelar estos cambios.

Como Oracle Database Graba en los Redo Log

El redo log de una base de datos consiste en dos o más redo log files. La base de datos requiere un mínimo de dos archivos para garantizar que uno está siempre disponible para grabar mientras el otro está siendo archivado (si la base de datos está en el modo ARCHIVELOG).

El proceso LGWR graba en el redo log files de una manera circular. Cuando el redo log files actual se llena, el LGWR empieza a grabar en el siguiente redo log file disponible. Cuando el último redo log file esta llenado, LGWR retorna al primer redo log file y graba en el él, comenzando el ciclo otra vez. La Figura 9.1 ilustra el proceso de grabación circular de los redo log files. Los números al lado de cada línea indican la secuencia en que LGWR graba a cada redo log file.

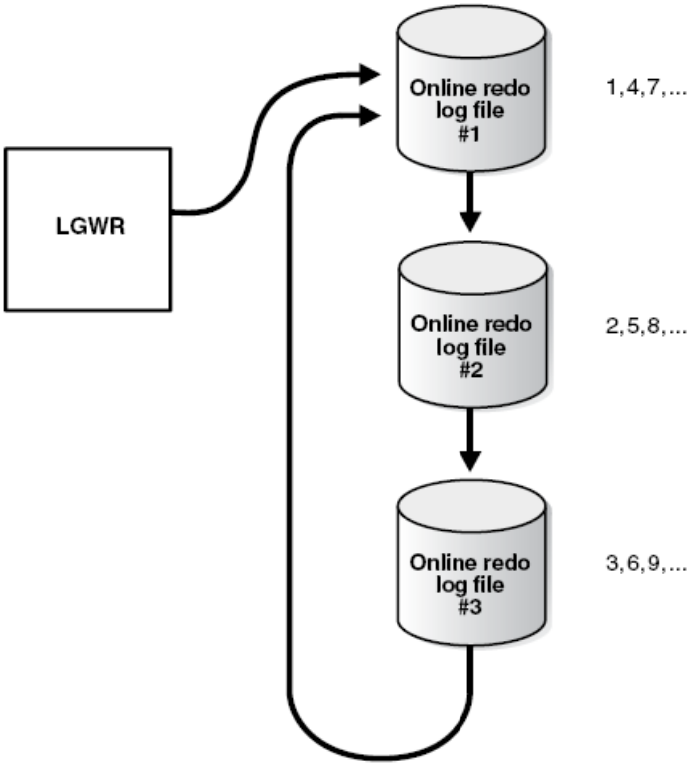


Figura 9 . 1 Reutilización de los Redo Log Files por el proceso LGWR

Llenado los redo log files están disponibles para ser reutilización por LGWR, dependiendo si esta habilitado el modo archivado.

- Si el archivado esta deshabilitado (la base de datos está en modo NOARCHIVELOG), un redo log file lleno vuelve a estar disponible después de que los cambios registrados en él son grabados en los archivos de datos.
- Si el archivado esta habilitado (la base de datos está en el modo ARCHIVELOG), un redo log file lleno vuelve a estar disponible después de que los cambios registrados en él han sido grabados en los archivos de datos y el archivo ha sido archivado.

Active (Current) y Inactive Redo Log Files

Ejemplo 1

```
SQL> select group#,members,status,sequence# from v$log;   [Enter]

    GROUP#    MEMBERS STATUS            SEQUENCE#
---------- ---------- ---------------- ----------
         1          1 CURRENT                  50
         2          1 INACTIVE                 48
         3          1 INACTIVE                 49
```

La base de datos Oracle utiliza solo un redo log file al mismo tiempo para almacenar los redo records desde el redo log buffer. El redo log file que LGWR actualmente esta escribiendo es llamado **current** redo log file.

Los redo log files que son requeridos para un proceso de recuperación de la instancia son llamados **active** redo log files. Los redo log files que ya no son requeridos para un proceso de recuperación de la instancia son llamados **inactive** redo log files.

Si la base de datos está en el modo ARCHIVELOG, entonces la base de datos no puede reutilizar o sobre-escribir un redo log file en línea activo (active) hasta que uno de los procesos de archivado de fondo (ARCn) haya archivado su contenido. Si la base de datos está en el modo NOARCHIVELOG, entonces cuando el último redo log file es llenado, LGWR continua reutilizando el primer archivo activo (active) disponible.

Log Switches y Log Sequence Numbers

Un log switch es el punto en el cual la base de datos deja de grabar en un redo log file y comienza a grabar en otro. Normalmente, un log switch ocurre cuando el current redo log file está completamente lleno y la grabación debe continuar en el siguiente redo log file. Sin embargo, se puede configurar log switch para que ocurran a intervalos regulares, independientemente de si el current log file está completamente lleno. También puede forzar log switch manualmente.

La Base de datos Oracle asigna a cada redo log file un nuevo log sequence number cada vez que un log switch ocurre y LGWR empieza a grabar. Cuando la base de datos archiva los redo log files, los log archivados conservan su log sequence number. A un redo log file que es cíclicamente utilizado se le asignara el siguiente log sequence number disponible.

Cada redo log file en línea o archivado es únicamente identificado por su log sequence number. Durante una falla, la instancia, o la recuperación de medios, la base de datos aplica correctamente los redo log files en orden ascendente usando el log sequence number de los redo log files.

Ejemplo 2

```
SQL> alter system switch logfile;                          [Enter]

Sistema modificado.

SQL> select group#,members,status,sequence# from v$log;    [Enter]

    GROUP#    MEMBERS STATUS            SEQUENCE#
---------- ---------- ---------------- ----------
         1          1 ACTIVE                   50
         2          1 CURRENT                  51
         3          1 INACTIVE                 49
```

Planificación de los Redo Log Files

Multiplexado de los Redo Log Files

La base de datos Oracle te deja multiplexar los redo log files de una instancia para salvaguardar contra daños a archivos individuales. Cuando los redo log files están multiplexados, el LGWR graba simultáneamente la misma información del redo log en múltiples redo log files idénticos, de tal modo que eliminamos fallas que puedan presentarse a un solo redo log file.

> **Nota**
>
> Oracle recomienda multiplexar los redo log files. La pérdida de datos de un redo log file puede ser catastrófica si se requiere un proceso de recuperación.

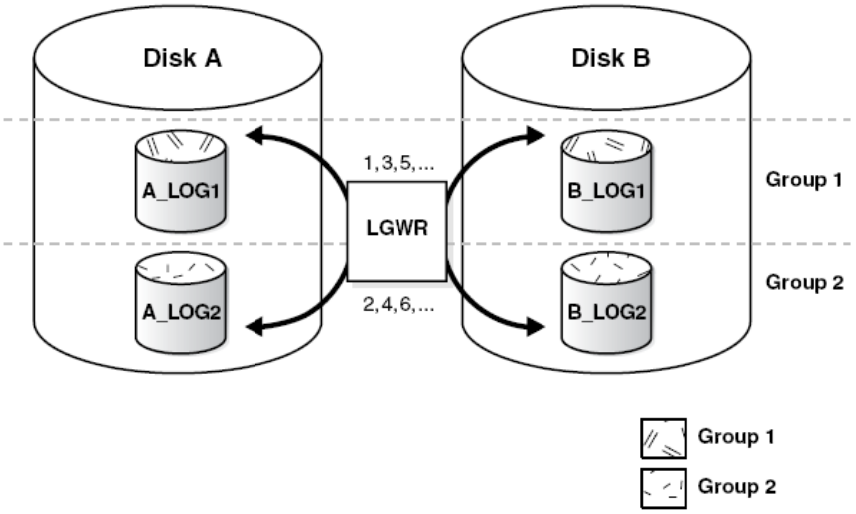


Figura 9 . 2 Multiplexado de los Redo Log Files

El multiplexado de los redo log files se llaman **groups** (grupos). Cada redo log files en un grupo se llama un **member** (miembro). En Figura 9.2, A_LOG1 y B_LOG1 ambos son miembros del grupo 1, A_LOG2 y B_LOG2 ambos son miembros del grupo 2, y así sucesivamente. Los miembros en un grupo debe ser exactamente del mismo tamaño.

Cada miembro de un log file group es simultáneamente activo (es decir, el LGWR graba simultáneamente en cada uno de ellos) según lo indicado por los log sequence number idénticos asignados por el LGWR. En la Figura 9.2, primer LGWR escribe simultáneamente en A_LOG1 y B_LOG1. Luego graba simultáneamente en A_LOG2 y B_LOG2, y así sucesivamente. LGWR nunca escribe simultáneamente en miembros de diversos grupos (por ejemplo, en A_LOG1 y B_LOG2).

Creando Groupos y Miembros Redo Log

Creando Grupos Redo Log

Sintaxis

ALTER DATABASE

ADD LOGFILE [GROUP #]

('especificación de archivo 1', 'especificación de archivo 1')

SIZE n {K|M}

Ejemplo 3

```
SQL> alter database
  2  add logfile
  3     ('C:\oracle\product\10.2.0\oradata\egcc\redo04a.log',
  4     'C:\oracle\product\10.2.0\oradata\egcc\redo04b.log')
  5  size 10M;                                                    [Enter]

Base de datos modificada.

SQL> select group#,members,status,sequence# from v$log;          [Enter]

    GROUP#    MEMBERS STATUS            SEQUENCE#
---------- ---------- ---------------- ----------
         1          1 INACTIVE                 50
         2          1 CURRENT                  51
         3          1 INACTIVE                 49
         4          2 UNUSED                    0

SQL> select group#, member from v$logfile
  2  order by 1 asc, 2 asc;                                       [Enter]

    GROUP# MEMBER
---------- --------------------------------------------------
         1 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO01.LOG
         2 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO02.LOG
         3 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO03.LOG
         4 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO04A.LOG
         4 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO04B.LOG
```

Creando Miembros Redo Log

Sintaxis

```
ALTER DATABASE
   ADD LOGFILE MEMBER 'especificación de archivo'
   TO GROUP #;
```

Ejemplo 4

```
SQL> select group#,members,status,sequence# from v$log;   [Enter]

    GROUP#    MEMBERS STATUS            SEQUENCE#
---------- ---------- ---------------- ----------
         1          1 INACTIVE                 50
         2          1 CURRENT                  51
         3          1 INACTIVE                 49
         4          2 UNUSED                    0

SQL> alter database add logfile member
  2  'C:\oracle\product\10.2.0\oradata\egcc\redo01B.log'
  3  to group 1;                                          [Enter]

Base de datos modificada.

SQL> select group#,members,status,sequence# from v$log;   [Enter]

    GROUP#    MEMBERS STATUS            SEQUENCE#
---------- ---------- ---------------- ----------
         1          2 INACTIVE                 50
         2          1 CURRENT                  51
         3          1 INACTIVE                 49
         4          2 UNUSED                    0

SQL> select group#, member from v$logfile
  2  order by 1 asc, 2 asc;                               [Enter]

    GROUP# MEMBER
---------- --------------------------------------------------
         1 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO01B.LOG
         1 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO01.LOG
         2 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO02.LOG
         3 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO03.LOG
         4 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO04A.LOG
         4 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO04B.LOG

6 filas seleccionadas.
```

Eliminando Grupos y Miembros Redo Log

Eliminando Grupos Redo Log

Sintaxis

ALTER DATABASE DROP LOGFILE GROUP #;

Ejemplo 5

```
SQL> select group#,members,status,sequence# from v$log;   [Enter]

    GROUP#    MEMBERS STATUS            SEQUENCE#
---------- ---------- ---------------- ----------
         1          2 INACTIVE                 50
         2          1 CURRENT                  51
         3          1 INACTIVE                 49
         4          2 UNUSED                    0

SQL> alter database drop logfile group 3;                 [Enter]

Base de datos modificada.

SQL> select group#,members,status,sequence# from v$log;   [Enter]

    GROUP#    MEMBERS STATUS            SEQUENCE#
---------- ---------- ---------------- ----------
         1          2 INACTIVE                 50
         2          1 CURRENT                  51
         4          2 UNUSED                    0
```

Eliminando Miembros Redo Log

Sintaxis

ALTER DATABASE DROP LOGFILE MEMBER ' especificación de archivo ';

Ejemplo 6

```
SQL> select group#,members,status,sequence# from v$log;     [Enter]

    GROUP#    MEMBERS STATUS            SEQUENCE#
---------- ---------- ---------------- ----------
         1          2 INACTIVE                 50
         2          1 CURRENT                  51
         4          2 UNUSED                    0
```

```
SQL> alter system switch logfile;                          [Enter]

Sistema modificado.

SQL> alter system switch logfile;                          [Enter]

Sistema modificado.

SQL> alter system switch logfile;                          [Enter]

Sistema modificado.

SQL> select group#,members,status,sequence# from v$log;    [Enter]

    GROUP#    MEMBERS STATUS            SEQUENCE#
---------- ---------- ---------------- ----------
         1          2 INACTIVE                 53
         2          1 CURRENT                  54
         4          2 INACTIVE                 52

SQL> select group#, member from v$logfile
  2  order by 1, 2;                                        [Enter]

    GROUP# MEMBER
---------- --------------------------------------------------
         1 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO01B.LOG
         1 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO01.LOG
         2 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO02.LOG
         4 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO04A.LOG
         4 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO04B.LOG

SQL> alter database drop logfile member
  2  'C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO01.LOG';    [Enter]

Base de datos modificada.

SQL> select group#, member from v$logfile
  2  order by 1, 2;                                        [Enter]

    GROUP# MEMBER
---------- --------------------------------------------------
         1 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO01B.LOG
         2 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO02.LOG
         4 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO04A.LOG
         4 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\REDO04B.LOG
```

Consultando Información sobre los Redo Log

Las siguientes vistas proveen información sobre los redo logs.

Vista	Descripción
V$LOG	Muestra información de los grupos redo logs.
V$LOGFILE	Muestra información de los grupos redo logs y sus miembros.
V$LOG_HISTORY	Muestra información histórica de los redo logs.

Ejercicio

Configure el redo log de su base de datos para que cumpla con los siguientes requisitos:

1. Debe tener 3 grupos redo log: 1, 2, y 3.
2. Cada grupo debe tener 2 miembros.
3. El tamaño de cada archivo debe ser de 100 Mb.

Página en Blanco

Lección 10

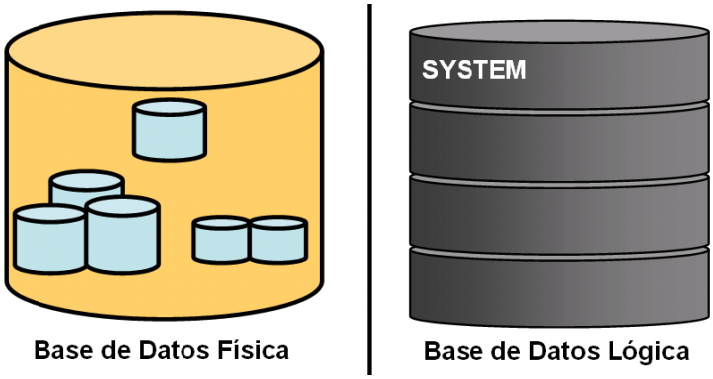
Administración de Tablespaces

Contenido

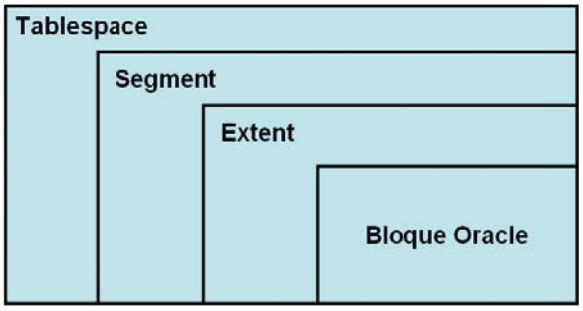
- Esquema de la Base de Datos 126
  - Tipos de Segmentos 127
- Guía para Administrar Tablespaces 128
  - Uso de Múltiples Tablespaces 128
  - Asignar Cuotas a los Usuarios en los Tablespace 128
- Administración de Tablespaces 129
  - Tablespace Administrado Localmente 129
  - Tablespace Administrando por el Diccionario 130
- Tablespace Administrado Localmente 131
  - Sintaxis General 131
  - Cláusula EXTENT 133
  - Cláusula AUTOEXTEND 136
- Tablespaces Temporales 137
  - Introducción 137
  - Crear un Tablespace Temporal Administrado Localmente 137
- Especificando Bloques NO Estándares para Tablespaces 139
- Gestionando Tablespaces 140
  - Monitoreando el Espacio Libre 140
  - Agregando un nuevo archivo 140
  - Tablespace ONLINE - OFFLINE 141
  - Tablespace READ ONLY – READ WRITE 142
  - Eliminando un Tablespace 142

Esquema de la Base de Datos

Una base de datos físicamente esta constituida de control files, log files y data files, los data files se agrupan en estructuras lógicas llamadas Tablespaces, una base de datos debe tener por lo menos un Tablespace llamado SYSTEM sin embargo se recomienda crear mas de uno para distribuir otras estructuras lógicas que almacenan datos.



Además de los Tablespace el servidor Oracle se compone de otras estructuras lógicas para gestionar el almacenamiento de datos, como se ilustra a continuación.



Bloque Oracle	Un bloque Oracle tiene un tamaño en bytes determinado por el parámetro DB_BLOCK_SIZE.
Extent	Es un conjunto de bloques oracle contiguos.
Segment	Se compone de uno o mas Extents.
Tablespace	Esta compuesto por uno o mas Segment.

Tipos de Segmentos

Un segmento está definido como cualquier entidad que consume espacio físico de almacenamiento dentro de la base de datos. Algunos de los tipos más comunes se describen en la siguiente tabla:

Tipo de Segmento	Descripción
Table	Almacenan datos en estructuras de filas y columnas.
Index	Mejoran el acceso a los datos de las tablas.
Rollback	Segmento especial usado para mantener consistencia en las lecturas durante las transacciones de los usuarios y ejecutar recuperación de transacción.
Partition	Divide una tabla en piezas más pequeñas, piezas más manejables para propósitos de desempeño.

Ejemplo 1

En este ejemplo consultamos los segmentos del esquema HR.

```
SQL> column segment_name format a20                    [Enter]

SQL> column tablespace_name format a20                 [Enter]

SQL> select segment_name, segment_type, tablespace_name, bytes
  2  from dba_segments
  3  where owner='SCOTT';                              [Enter]

SEGMENT_NAME       SEGMENT_TYPE       TABLESPACE_NAME      BYTES
----------------- ------------------ -------------------- ----------
PK_DEPT            INDEX              USERS                65536
DEPT               TABLE              USERS                65536
EMP                TABLE              USERS                65536
PK_EMP             INDEX              USERS                65536
BONUS              TABLE              USERS                65536
SALGRADE           TABLE              USERS                65536

6 filas seleccionadas.
```

Guía para Administrar Tablespaces

Uso de Múltiples Tablespaces

Usando múltiples tablespaces tenemos mayor flexibilidad en la ejecución de operaciones de base de datos. Por ejemplo, cuando una base de datos tiene múltiples tablespaces, podemos realizar las tareas siguientes:

- Separar los datos del usuario de los datos del diccionario de datos para reducir la disputa entre los objetos del diccionario y los objetos de los esquemas para los mismos datafiles.
- Separe los datos de una aplicación de otras, para impedir que múltiples aplicaciones sean afectadas si un tablespace debe ser puesto en OFFLINE.
- Almacenar los datafiles de los tablespaces en discos separados para reducir la disputa de las operaciones de I/O.
- Poner un tablespace OFFLINE mientras que otros permanecen en línea, con tal que en conjunto este siempre disponible.
- Reservar un tablespace para un particular tipo de uso de la base de datos, como una alta actividad de actualización, solo lectura, o como segmento de almacenamiento temporal. Esto permite optimizar el uso de los tablespaces.
- Backup de tablespaces individuales.

Asignar Cuotas a los Usuarios en los Tablespace

Conceda a los usuarios que tendrán privilegios para crear objetos una cuota en el tablespace donde tendrá su segmento para sus objetos.

Administración de Tablespaces

Tablespace Administrado Localmente

Se trata de un tablespace que gestiona sus propias extensiones, utiliza un mapa de bits en cada archivo de datos para mantener un seguimiento del estado libre ó utilizado de las extensiones.

Cada vez que una extensión es asignada o se libera para volver a utilizarla, Oracle cambia los valores del mapa de bits para que refleje el nuevo estado de las extensiones. Estos cambios no generan información de rollback ya que no actualizan las tablas del diccionario de datos.

Los tablespaces administrados localmente presentan los siguientes beneficios:

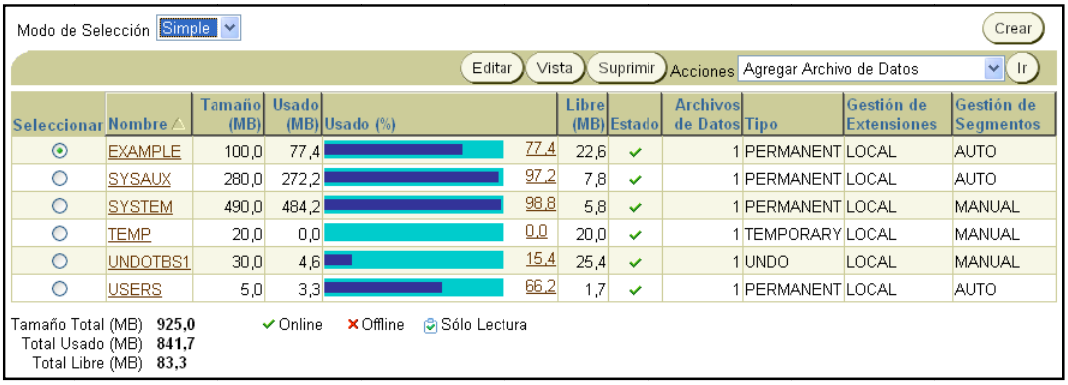
- Mejora la concurrencia y velocidad de operaciones de espacio, por que la asignación y liberación de espacio modifica recursos administrados localmente (mapa de bits almacenados en la cabecera de los archivos).
- Mejora el desempeño, por que operaciones recursivas que se pueden producir en los tablespace administrados por el diccionario de datos son eliminadas.
- Se permiten bases de datos de reserva leíbles, porque administramos localmente tablespaces temporales (por ejemplo, para ordenamientos) y así no generan ningún undo o redo.
- Simplifica la asignación de espacio, cuando la cláusula AUTOALLOCATE es especificada, el tamaño apropiado para el extent es automáticamente seleccionado.
- Reduce la confianza del usuario en el diccionario de datos por que la información necesaria es almacenada en la cabecera de los archivos y los bloques de mapa de bits.
- Unir extents libres es innecesario para los tablespaces administrados localmente.

Todos los tablespaces, incluido el tablespace SYSTEM, pueden ser administrados localmente.

Adicionalmente, el paquete DBMS_SPACE_ADMIN provee procedimientos para tablespaces administrados localmente.

Ejemplo 2

```
SQL> conn / as sysdba                                  [Enter]
Conectado.

SQL> select tablespace_name, extent_management
  2  from dba_tablespaces;                             [Enter]

TABLESPACE_NAME                EXTENT_MAN
------------------------------ ----------
SYSTEM                         LOCAL
UNDOTBS1                       LOCAL
SYSAUX                         LOCAL
TEMP                           LOCAL
USERS                          LOCAL
EXAMPLE                        LOCAL

6 filas seleccionadas.
```



Seleccionar	Nombre	Tamaño (MB)	Usado (MB)	Usado (%)	Libre (MB)	Estado	Archivos de Datos	Tipo	Gestión de Extensiones	Gestión de Segmentos
◉	EXAMPLE	100,0	77,4	77,4	22,6	✓	1	PERMANENT	LOCAL	AUTO
○	SYSAUX	280,0	272,2	97,2	7,8	✓	1	PERMANENT	LOCAL	AUTO
○	SYSTEM	490,0	484,2	98,8	5,8	✓	1	PERMANENT	LOCAL	MANUAL
○	TEMP	20,0	0,0	0,0	20,0	✓	1	TEMPORARY	LOCAL	MANUAL
○	UNDOTBS1	30,0	4,6	15,4	25,4	✓	1	UNDO	LOCAL	MANUAL
○	USERS	5,0	3,3	66,2	1,7	✓	1	PERMANENT	LOCAL	AUTO

Firura 10 . 1 Gestión de tablespaces desde Oracle EM

Tablespace Administrando por el Diccionario

Cuando iniciamos Oracle 10g, la opción por defecto para la administración de los extent cuando creamos un tablespace es *Administrado Localmente*. Sin embargo, podemos especificar explícitamente cuando queremos crear un tablespace administrado por el diccionario, Oracle actualiza las tablas apropiadas en el diccionario cuando un extent es asignado o liberado para rehusarlo.

Tablespace Administrado Localmente

Sintaxis General

```
CREATE TABLESPACE nombre
   DATA FILE nombre SIZE n [K|M|G|T]  [REUSE]
   [ AUTOEXTEND { OFF | ON [NEXT n [K|M|G|T] ]  [MAXSIZE n [K|M|G|T]] } ]
   [ BLOCKSIZE n [K] ]
   [ ONLINE | OFFLINE ]
   [ EXTENT MANAGEMENT
      { DICTIONARY | LOCAL [ AUTOALLOCATE | UNIFORM [SIZE n [K|M|G|T] ] ] } ]
```

Ejemplo 3

```
SQL> conn / as sysdba                                          [Enter]
Conectado.

SQL> create tablespace ts_demo
  2  datafile 'C:\oracle\product\10.2.0\oradata\egcc\ts_demo.dbf'
  3  size 5 M;                                                 [Enter]

Tablespace creado.
```

En este ejemplo se esta creando el tablespace **ts_demo** con parámetros por defecto:

- Estado **ReadWrite**
- Tipo **Permanente**
- Gestión de Extensiones **Local**
- Tipo de Asignación **Automático**
- Gestión de Espacio de Segmento **Automático**
- Ampliar automáticamente **Deshabilitado**

Ejemplo 4

Objetos generales para los ejemplos de esta lección.

```
SQL> create sequence sq_table;                                  [Enter]
Secuencia creada.

SQL> create or replace procedure sp_fill_table
  2  ( tabla varchar2, n number )
  3  as
  4    cmd varchar2(500);
  5    s varchar2(200) := 'Oracle is PowerFull';
  6  begin
  7    cmd := 'insert into ' || tabla
  8        || ' values(sq_table.nextval,''' || s || ''')';
  9    for k in 1..n loop
 10      execute immediate ( cmd );
 11      commit;
 12    end loop;
 13  end;
 14  /                                                           [Enter]

Procedimiento creado.
```

Ejemplo 5

```
SQL> create table tb_demo(
  2    id number constraint pk_tb_demo primary key,
  3    dato varchar2(200)
  4  ) tablespace ts_demo;                                [Enter]

Tabla creada.

SQL> call sp_fill_table( 'tb_demo', 62300 );            [Enter]
Llamada terminada.

SQL> select segment_name, segment_type, extent_id, blocks, bytes
  2  from dba_extents
  3  where tablespace_name = 'TS_DEMO';                   [Enter]

SEGMENT_NAME         SEGMENT_TYPE        EXTENT_ID     BLOCKS      BYTES
-------------------- ------------------ ---------- ---------- ----------
TB_DEMO              TABLE                       0          8      65536
TB_DEMO              TABLE                       1          8      65536
. . .
. . .
TB_DEMO              TABLE                      14          8      65536
TB_DEMO              TABLE                      15          8      65536
TB_DEMO              TABLE                      16        128    1048576
TB_DEMO              TABLE                      17        128    1048576
PK_TB_DEMO           INDEX                       0          8      65536
PK_TB_DEMO           INDEX                       1          8      65536
. . .
. . .
PK_TB_DEMO           INDEX                      14          8      65536
PK_TB_DEMO           INDEX                      15          8      65536

34 filas seleccionadas.
```

Cláusula EXTENT

Sintaxis

```
[ EXTENT MANAGEMENT
   { DICTIONARY | LOCAL [ AUTOALLOCATE | UNIFORM [SIZE n [K|M|G|T] ] ] } ]
```

Para crear un tablespace administrado localmente, debemos especificar LOCAL en la cláusula EXTENT MANAGEMENT de la instrucción CREATE TABLESPACE.

Tenemos dos opciones, podemos hacer que Oracle maneje los extent automáticamente con la opción AUTOALLOCATE (opción por defecto), o podemos especificar que el tablespace sea manejado con extent uniformes de un tamaño específico mediante la cláusula UNIFORM SIZE.

Ejemplo 6

```
SQL> create tablespace ts_a
  2  datafile 'C:\oracle\product\10.2.0\oradata\egcc\ts_a.dbf' size 5 M
  4  extent management local autoallocate;                      [Enter]

Tablespace creado.

SQL> create table tb_a(
  2    id number constraint pk_tb_a primary key,
  3    dato varchar2(200)
  4  ) tablespace ts_a;                                          [Enter]

Tabla creada.

SQL> call sp_fill_table( 'tb_a', 62300 );                       [Enter]

Llamada terminada.

SQL> select segment_name, segment_type, extent_id, blocks, bytes
  2  from dba_extents
  3  where tablespace_name = 'TS_A';                             [Enter]

SEGMENT_NAME         SEGMENT_TYPE        EXTENT_ID     BLOCKS      BYTES
-------------------- ------------------ ---------- ---------- ----------
TB_A                 TABLE                       0          8      65536
TB_A                 TABLE                       1          8      65536
. . .
. . .
TB_A                 TABLE                      14          8      65536
TB_A                 TABLE                      15          8      65536
TB_A                 TABLE                      16        128    1048576
TB_A                 TABLE                      17        128    1048576
PK_TB_A              INDEX                       0          8      65536
PK_TB_A              INDEX                       1          8      65536
. . .
. . .
PK_TB_A              INDEX                      14          8      65536
PK_TB_A              INDEX                      15          8      65536

34 filas seleccionadas.
```

El resultado que se obtiene es similar al del Ejemplo 4.

Ejemplo 7

```
SQL> create tablespace ts_b
  2  datafile 'C:\oracle\product\10.2.0\oradata\egcc\ts_b.dbf' size 5 M
  3  extent management local uniform;                          [Enter]

Tablespace creado.


SQL> create table tb_b(
  2    id number constraint pk_tb_b primary key,
  3    dato varchar2(200)
  4  ) tablespace ts_b;                                        [Enter]

Tabla creada.

SQL> call sp_fill_table( 'tb_b', 62300 );                      [Enter]

Llamada terminada.

SQL> select segment_name, segment_type, extent_id, blocks, bytes
  2  from dba_extents
  3  where tablespace_name = 'TS_B';                           [Enter]

SEGMENT_NAME         SEGMENT_TYPE        EXTENT_ID     BLOCKS      BYTES
-------------------- ------------------ ---------- ---------- ----------
TB_B                 TABLE                       0        128    1048576
TB_B                 TABLE                       1        128    1048576
TB_B                 TABLE                       2        128    1048576
PK_TB_B              INDEX                       0        128    1048576
```

En este ejemplo se ha especificado que el tamaño del extent es uniforme, pero no se ha indicado el tamaño, entonces asume el valor por defecto, este es de 1 MB.

Ejemplo 8

```
SQL> create tablespace ts_c
  2  datafile 'C:\oracle\product\10.2.0\oradata\egcc\ts_c.dbf' size 2M
  3  extent management local uniform size 127 K;                [Enter]

Tablespace creado.

SQL> create table tb_c(
  2    id number constraint pk_tb_c primary key,
  3    dato varchar2(200)
  4  ) tablespace ts_c;                                          [Enter]

Tabla creada.

SQL> call sp_fill_table( 'tb_c', 10000 );                       [Enter]

Llamada terminada.

SQL> select segment_name, segment_type, extent_id, blocks, bytes
  2  from dba_extents
  3  where tablespace_name = 'TS_C';                             [Enter]

SEGMENT_NAME         SEGMENT_TYPE        EXTENT_ID     BLOCKS      BYTES
-------------------- ------------------ ---------- ---------- ----------
TB_C                 TABLE                       0         16     131072
TB_C                 TABLE                       1         16     131072
TB_C                 TABLE                       2         16     131072
PK_TB_C              INDEX                       0         16     131072
PK_TB_C              INDEX                       1         16     131072
```

En este caso, especificamos que el extent tenga un tamaño de 127K, pero no es posible por que tiene que ser un múltiplo de 8K, debido a que el parámetro DB_BLOCK_SIZE es de 8K, por lo tanto es redondeado a 128K, que es lo mismo a 16 bloques.

```
SQL> show parameter db_block_size                               [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
-
db_block_size                        integer     8192
```

Cláusula AUTOEXTEND

Sintaxis

```
[ AUTOEXTEND { OFF | ON [NEXT n [K|M|G|T] ]  [MAXSIZE n [K|M|G|T]] } ]
```

Donde:

OFF	Deshabilita el crecimiento automático del tablespace.
ON	Habilita el crecimiento automático del tablespace.
NEXT	Se utiliza para especificar el siguiente incremento del tamaño en disco cuando mas extent son requeridos. El valor predeterminado es el tamaño de un bloque.
MAXSIZE	Esta cláusula se utiliza para especificar el máximo espacio permitido para expandir automáticamente el archivo de datos.
UNLIMITED	Esta cláusula indica que no existe límite para el crecimiento del archivo de datos.

Ejemplo 9

```
SQL> create tablespace ts_d
  2  datafile 'C:\oracle\product\10.2.0\oradata\egcc\ts_d.dbf' size 500 K
  3  autoextend on next 500 K maxsize 3 M
  4  extent management local uniform size 256 K;          [Enter]

Tablespace creado.

SQL> create table tb_d(
  2    id number constraint pk_tb_d primary key,
  3    dato varchar2(200)
  4  ) tablespace ts_d;                                    [Enter]

Tabla creada.

SQL> call sp_fill_table( 'tb_d', 20000 );                 [Enter]

Llamada terminada.

SQL> select segment_name, segment_type, extent_id, blocks, bytes
  2  from dba_extents
  3  where tablespace_name = 'TS_D';                       [Enter]

SEGMENT_NAME         SEGMENT_TYPE        EXTENT_ID     BLOCKS      BYTES
-------------------- ------------------ ---------- ---------- ----------
TB_D                 TABLE                       0         32     262144
TB_D                 TABLE                       1         32     262144
TB_D                 TABLE                       2         32     262144
PK_TB_D              INDEX                       0         32     262144
PK_TB_D              INDEX                       1         32     262144
```

Tablespaces Temporales

Introducción

Para mejorar la concurrencia de múltiples operaciones de ordenamiento, reducir su costo, o evitar que la Base de Datos Oracle administre operaciones de espacio, debemos crear tablespaces temporales. Un tablespace temporal puede ser compartido por múltiples usuarios.

Dentro de un tablespace temporal, todas las operaciones de ordenamiento para una instancia dada y el tablespace comparten un solo segmento de ordenamiento. El segmento de ordenamiento existe para cada instancia que ejecuta operaciones de ordenamiento dentro de un tablespace determinado. El segmento de ordenamiento es creado por la primera instrucción que use el tablespace temporal para ordenamiento, después de iniciar la base de datos, y es liberado solo cuando se baja la instancia. Un extent no puede ser compartido por múltiples transacciones.

Podemos consultar la asignación y liberación de espacio dentro de un segmento de ordenamiento en un tablespace temporal usando la vista V$SORT_SEGMENT. La vista V$TEMPSEG_USAGE identifica los usuarios actuales en esos segmentos.

No es posible crear objetos explícitamente en un tablespace temporal.

Ejemplo 10

En este ejemplo consultamos el nombre del tablespace temporal.

```
SQL> select TABLESPACE_NAME from v$sort_segment;       [Enter]

TABLESPACE_NAME
-------------------------------
TEMP
```

Crear un Tablespace Temporal Administrado Localmente

Debido a que el manejo del espacio es mucho más simple y más eficiente en un tablespace administrado localmente, ellos son los más apropiados para tablespaces temporales. Los tablespaces temporales administrados localmente usan **tempfiles**, los cuales no modifican data fuera del tablespace temporal o generan ningún redo para data del tablespace temporal.

Para consultar información sobre los tempfiles podemos usar las vistas V$TEMPFILE y DBA_TEMP_FILES análogo a las vistas V$DATAFILE y DBA_DATA_FILES.

Para crear un tablespace temporal debemos usar la instrucción CREATE TEMPORARY TABLESPACE, y necesitamos el privilegio del sistema CREATE TABLESPACE.

Sintaxis

```
CREATE TEMPORARY TABLESPACE nombre
   TEMPFILE nombre SIZE n [K|M|G|T] [REUSE]
   [ AUTOEXTEND { OFF | ON [NEXT n [K|M|G|T] ] [MAXSIZE n [K|M|G|T]] } ]
   [ EXTENT MANAGEMENT
      { DICTIONARY | LOCAL [ UNIFORM [SIZE n [K|M|G|T] ] ]
```

Ejemplo 11

```
SQL> create temporary tablespace temp_a
  2  tempfile 'C:\oracle\product\10.2.0\oradata\egcc\TEMP_A.DBF'
  3  size 5 M reuse
  4  extent management local uniform size 1M;        [Enter]

Tablespace creado.
```

Para optimizar el desempeño de las operaciones de ordenamiento en el tablespace temporal, el valor especificado en UNIFORM SIZE debe ser múltiplo del parámetro SORT_AREA_SIZE.

Especificando Bloques NO Estándares para Tablespaces

Podemos crear tablespaces con tamaños de bloques diferentes al estándar definido por el parámetro DB_BLOCK_SIZE. Esta característica permite transportar tablespaces con tamaños de bloques diferentes entre bases de datos.

La cláusula BLOCKSIZE en la instrucción CREATE TABLESPACE permite especificar el tamaño del bloque diferente al tamaño del bloque estándar. Sin embargo, el buffer cache en el SGA debe ser configurado para tamaños de bloques no estándar.

Ejemplo 12

```
SQL> alter system set db_2k_cache_size = 8M scope = spfile;    [Enter]

Sistema modificado.

SQL> show parameter db_2k_cache_size                           [Enter]

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------
db_2k_cache_size                     big integer 8M

SQL> create tablespace ts_e
  2  datafile 'C:\oracle\product\10.2.0\oradata\egcc\ts_e.dbf' size 2M
  3  extent management local uniform size 128K
  4  blocksize 2K;                                             [Enter]

Tablespace creado.

SQL> create table tb_e(
  2     id number primary key,
  3     valor varchar2(200)
  4  ) tablespace ts_e;                                        [Enter]

Tabla creada.

SQL> select segment_name, segment_type, extent_id, blocks, bytes
  2  from dba_extents
  3  where tablespace_name = 'TS_E';                           [Enter]

SEGMENT_NAME         SEGMENT_TYPE        EXTENT_ID     BLOCKS      BYTES
-------------------- ------------------ ---------- ---------- ----------
TB_E                 TABLE                       0         64     131072
SYS_C005523          INDEX                       0         64     131072
```

Gestionando Tablespaces

Monitoreando el Espacio Libre

Para monitorear el espacio libre de un tablespace tenemos las vistas:

- dba_free_space
- dba_free_space_coalesced

Ejemplo 13

```
SQL> select block_id, bytes, blocks
  2  from dba_free_space
  3  where tablespace_name = 'SYSTEM'
  4  order by block_id;                    [Enter]

  BLOCK_ID      BYTES     BLOCKS
---------- ---------- ----------
     61361     720896         88
     62089    5177344        632
```

Agregando un nuevo archivo

Ejemplo 14

```
SQL> alter tablespace ts_d
  2  add datafile 'C:\oracle\product\10.2.0\oradata\egcc\ts_d2.dbf'
  3  size 1M;                                                  [Enter]

Tablespace modificado.

SQL> select ts#, name from v$tablespace where name = 'TS_D';   [Enter]

       TS# NAME
---------- ------------------------------
        12 TS_D


SQL> select ts#, name, block_size, status
  2  from v$datafile where ts# = 12;                            [Enter]

  TS# NAME                                              BLOCK_SIZE STATUS
----- ------------------------------------------------ ---------- -------
   12 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\TS_D.DBF          8192 ONLINE
   12 C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\TS_D2.DBF         8192 ONLINE
```

Tablespace ONLINE - OFFLINE

Ejemplo 15

```
SQL> select count(*) from tb_b;        [Enter]

  COUNT(*)
----------
     62300

SQL> alter tablespace ts_b offline;    [Enter]

Tablespace modificado.

SQL> select count(*) from tb_b;        [Enter]
select count(*) from tb_b
                     *
ERROR en línea 1:
ORA-00376: el archivo 8 no se puede leer en este momento
ORA-01110: archivo de datos 8: 'C:\ORACLE\PRODUCT\10.2.0\ORADATA\TS_B.DBF'

SQL> alter tablespace ts_b online;     [Enter]

Tablespace modificado.

SQL> select count(*) from tb_b;        [Enter]

  COUNT(*)
----------
     62300
```

Tablespace READ ONLY – READ WRITE

Ejemplo 16

```
SQL> insert into tb_e values ( 500, 'www.perudev.com' );       [Enter]

1 fila creada.

SQL> select * from tb_e;                                       [Enter]

        ID VALOR
---------- ------------------------------
       500 www.perudev.com

SQL> alter  tablespace ts_e read only;                         [Enter]

Tablespace modificado.

SQL> insert into tb_e values ( 501, 'Oracle is Powerful' );    [Enter]
insert into tb_e values ( 501, 'Oracle is Powerful' )
            *
ERROR en línea 1:
ORA-00372: el archivo 12 no se puede modificar en este momento
ORA-01110: archivo de datos 12:
'C:\ORACLE\PRODUCT\10.2.0\ORADATA\EGCC\TS_E.DBF'

SQL> select * from tb_e;                                       [Enter]

        ID VALOR
---------- ------------------------------
       500 www.perudev.com
```

Eliminando un Tablespace

```
SQL> select name from v$tablespace where name = 'TS_B';        [Enter]

NAME
------------------------------
TS_B

SQL> drop tablespace ts_b including contents and datafiles;    [Enter]

Tablespace borrado.

SQL> select name from v$tablespace where name = 'TS_B';        [Enter]

ninguna fila seleccionada
```

Lección 11

Usuarios y Recursos

Contenido

- Administración de Usuarios 144
  - Guía para Crear Usuarios 144
  - Creación de un Nuevo Usuario 145
  - Modificando Usuarios 147
  - Eliminando Usuarios 148
  - Administración de Recursos con Perfiles 149
- Habilitar y Deshabilitar los Limites de Recursos 149
  - Creación de Perfiles 151
  - Modificando Perfiles 155
  - Eliminar Perfiles 155
  - Contraseñas Complejas 156
- Verificar Información sobre Usuarios y Perfiles 160
  - Usuarios y su Información Asociada 160
  - Tablaspaces y Quotas 160
  - Perfiles y Limites asignados 161

Administración de Usuarios

Cada base de datos Oracle tiene una lista de usuarios de base de datos validos. Para acceder a una base de datos, un usuario necesita una aplicación y conectarse a una instancia de la base de datos usando un nombre de usuario valido definido en la base de datos.

La seguridad del servidor y la administración de los usuarios de la base de datos Oracle están directamente relacionadas, por ello resulta imprescindible administrar el acceso a los servicios del servidor Oracle de manera adecuada, permitiendo a los usuarios realizar solo las operaciones para las cuales han sido creados.

Guía para Crear Usuarios

Para crear un usuario debemos tener en cuenta:

1. Identificar correctamente los tablespaces que utilizará el usuario para almacenas sus objetos en la base de datos.

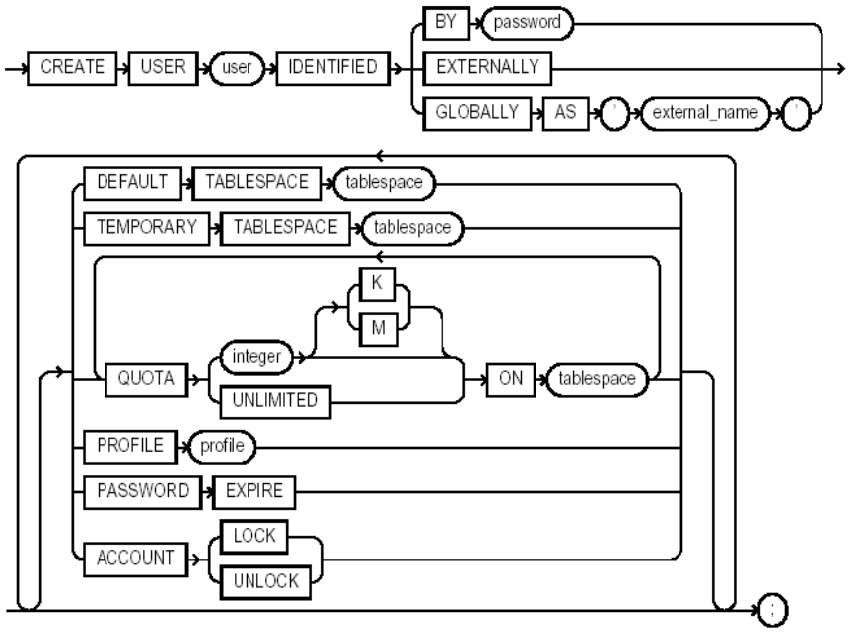
2. Asignar cuotas de espacio a los usuarios en los Tablespaces disponibles.

3. Asignar un tablespace por defecto y un tablespace temporal.

4. Asignar los privilegios y roles pertinentes.

Creación de un Nuevo Usuario

Sintaxis



Ejemplo 1

Para crear un usuario necesitamos tener privilegio CREATE USER. Primero crearemos el tablespace para los datos.

```
SQL> create tablespace ts_alm
  2  datafile 'E:\oracle\oradata\dbegcc\ts_alm.ora' size 1M
  3  autoextend on next 512K maxsize 5M
  4  extent management local uniform size 128K;                [Enter]

Tablespace created.
```

Debemos crear un tablespace temporal.

```
SQL> create temporary tablespace ts_alm_temp
  2  tempfile 'E:\oracle\oradata\dbegcc\ts_alm_temp.ora' size 1M
  3  extent management local uniform size 128K;                [Enter]

Tablespace created.
```

Ahora procedemos a crear el usuario.

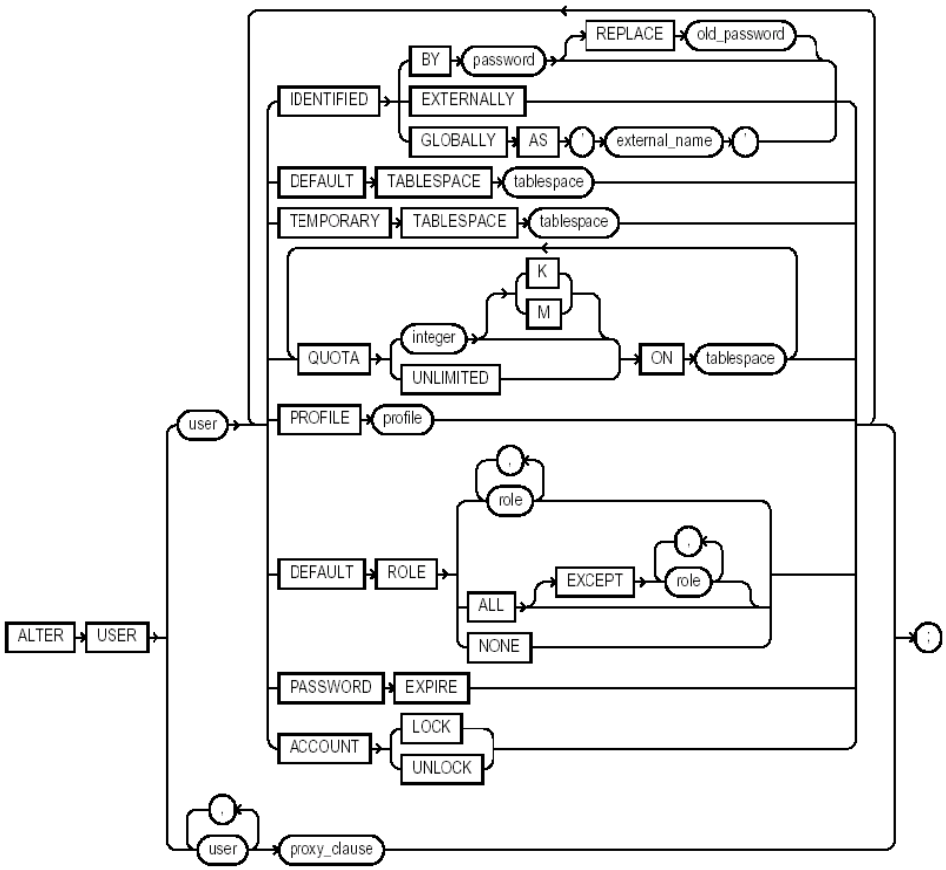
```
SQL> create user angelica
  2  identified by suerte
  3  default tablespace ts_alm
  4  quota 1M on ts_alm
  5  temporary tablespace ts_alm_temp;                    [Enter]

User created.
```

Para que el usuario pueda iniciar sesión y operar objetos de la base de datos debe tener privilegios, por ejemplo, el privilegio CREATE SESSION permite crear sesiones.

```
SQL> grant create session to angelica;                    [Enter]
Grant succeeded.

SQL> conn angelica/suerte                                 [Enter]
Connected.
```

Modificando Usuarios

Sintaxis



Ejemplo 2

Cambiar la contraseña de angelica, debe ser luck.

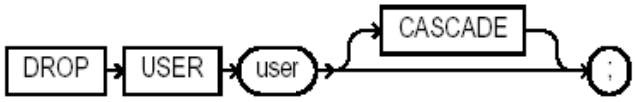
```
SQL> alter user angelica
  2  identified by luck;                                  [Enter]

User altered.

SQL> conn angelica/luck                                   [Enter]
Connected.
```

Su cuota en el tablespace ts_alm debe ser ilimitado.

```
SQL> alter user angelica quota unlimited on ts_alm;       [Enter]
User altered.
```

Eliminando Usuarios

Sintaxis



Ejemplo 3

```
SQL> conn / as sysdba                                        [Enter]
Connected.

SQL> grant create table to angelica;                         [Enter]
Grant succeeded.

SQL> conn angelica/luck                                      [Enter]
Connected.

SQL> create table test(
  2     id number primary key,
  3     valor varchar2(30)
  4  );                                                      [Enter]

Table created.

SQL> insert into test values(1,'Angelica es Beautiful');  [Enter]

1 row created.

SQL> commit;                                                 [Enter]

Commit complete.

SQL> conn / as sysdba                                        [Enter]
Connected.

SQL> select table_name, owner
  2  from dba_tables
  3  where tablespace_name = 'TS_ALM';                       [Enter]

TABLE_NAME                     OWNER
------------------------------ ------------------------------
TEST                           ANGELICA

SQL> drop user angelica;                                     [Enter]
drop user angelica
*
ERROR at line 1:
ORA-01922: CASCADE must be specified to drop 'ANGELICA'

SQL> drop user angelica cascade;                             [Enter]

User dropped.
```

Administración de Recursos con Perfiles

Un perfil es un nombre que agrupa un conjunto de límites de recursos. El perfil de un usuario limita el uso de recursos de la base de datos y la instancia definidos en el perfil. Se pude asignar un perfil a cada usuario; y un perfil por defecto para todos los usuarios que no tienen un perfil especifico. Para que los perfiles tengan efecto, los límites de recursos deben estar habilitados para la base de datos.

Habilitar y Deshabilitar los Limites de Recursos

Antes de Iniciar la Base de Datos

Los limites de recursos se pueden habilitar o deshabilitar con el parámetro RESOURCE_LIMIT en el archivo de parámetros. Los valores que puede tomar son:

TRUE Habilita la aplicación de límites de recursos definidos en un perfil.

FALSE Valor por defecto. Deshabilita la aplicación de límites de recursos definidos en un perfil.

Cuando la Base de Datos esta Inicializada

Si no es posible cerrar la base de datos temporalmente, debemos utilizar la instrucción ALTER SYSTEM para modificar el parámetro RESOURCE_LIMIT.

Sintaxis

```
ALTER SYSTEM
    SET RESOURCE_LIMIT = {TRUE|FALSE};
```

Ejemplo 4

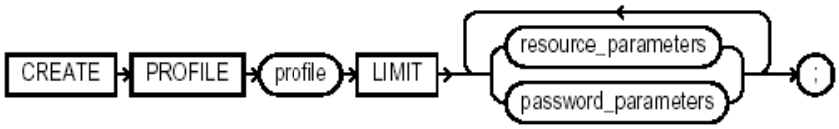
```
SQL> conn / as sysdba                    [Enter]
Connected.

SQL> create profile pf_demo limit
  2  sessions_per_user 2;                [Enter]

Profile created.

SQL> create user egcc
  2  identified by egcc
  3  profile pf_demo;                    [Enter]

User created.

SQL> grant connect to egcc;              [Enter]

Grant succeeded.
```

Ahora procedamos a cargar tres instancias de SQLPlusW, iniciemos dos sesiones con el usuario **egcc**, no tendremos ningún problema. Luego en la tercera instancia de SQLPlusW tratemos de iniciar una nueva sesión con el usuario **egcc**, el resultado se muestra a continuación:

```
SQL> conn egcc/egcc                      [Enter]
ERROR:
ORA-02391: exceeded simultaneous SESSIONS_PER_USER limit


Warning: You are no longer connected to ORACLE.
```

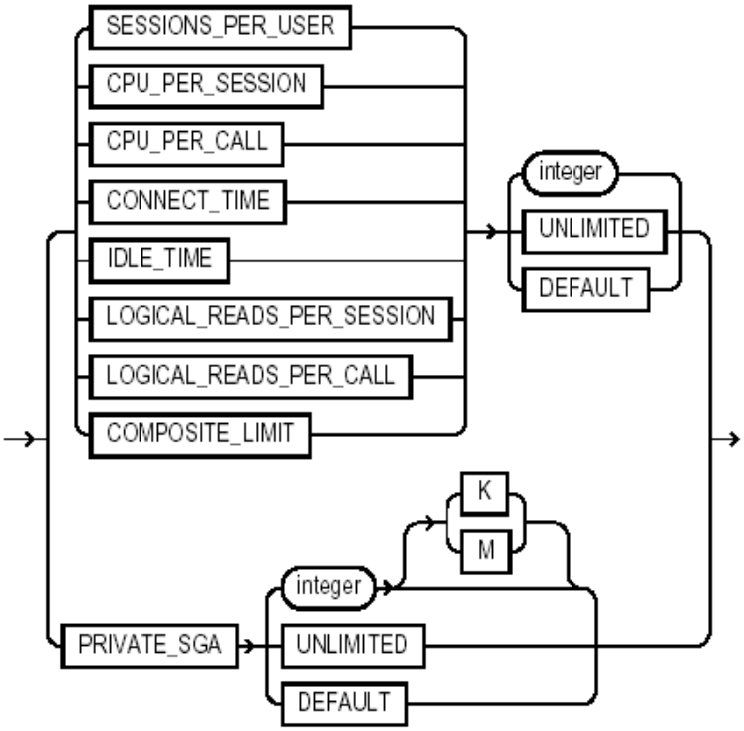
Creación de Perfiles

Sintaxis



resource_parameters

Sintaxis



SESSIONS_PER_USER	Número de sesiones concurrentes por usuario.
CPU_PER_SESSION	Tiempo limite de CPU por sesión, expresado en centésimas de segundo.
CPU_PER_CALL	Tiempo limite de CPU por llamada, expresado en centésimos de segundo.
CONNECT_TIME	Tiempo limite por sesión, expresado en minutos.

IDLE_TIME	Tiempo limite continuo de periodos de inactividad, expresado en minutos.
LOGICAL_READS_PER_SESSION	Número permitido de bloques de datos leídos en una sesión.
LOGICAL_READS_PER_CALL	Número permitido de bloques leídos por llamada para procesar una instrucción.
PRIVATE_SGA	Cantidad de espacio privado que una sesión puede destinar en el shared pool del SGA, expresado en bytes. Se debe usar **K** o **M** para especificar este limite en kilobytes o megabytes.
COMPOSITE_LIMIT	Especifica el costo total de recursos para una sesión, expresado en **service units**.

Ejemplo 5

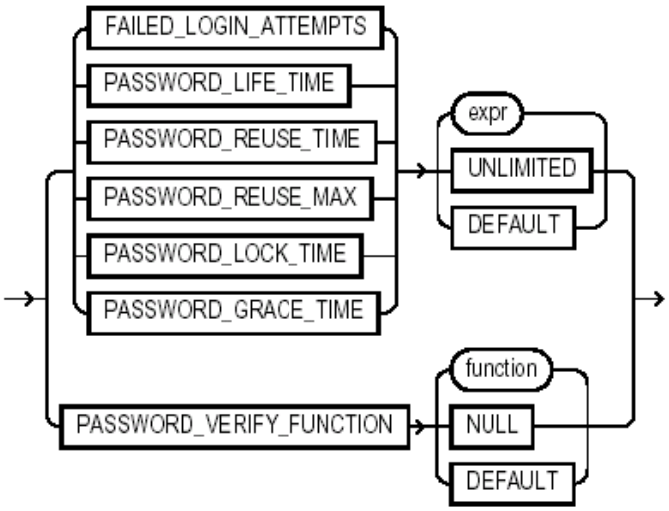
```
SQL> conn / as sysdba                    [Enter]
Connected.

SQL> create profile pf_cpu limit
  2  cpu_per_session 1;                  [Enter]

Profile created.

SQL> alter user egcc
  2  profile pf_cpu;                     [Enter]

User altered.

SQL> conn egcc/egcc                      [Enter]
Connected.
```

Después de realizar varias consultas obtenemos el siguiente resultado:

```
SQL> select user from dual;              [Enter]
*
ERROR at line 1:
ORA-02392: exceeded session limit on CPU usage, you are being logged off
```

password_parameters

Sintaxis



FAILED_LOGIN_ATTEMPTS	Número de intentos fallidos antes de que la cuenta del usuario sea bloqueada.
PASSWORD_LIFE_TIME	Especifique el número de días que la misma contraseña puede usarse. La contraseña expira si no se cambia dentro de este período.
PASSWORD_REUSE_TIME	Especifique el número de días que deben pasar para utilizar la misma clave.
PASSWORD_REUSE_MAX	Número de cambios necesarios para utilizar la misma clave.
PASSWORD_LOCK_TIME	Número de días que una cuenta de usuario estará bloqueada.
PASSWORD_GRACE_TIME	Número de días que un usuario puede trabajar con la base de datos con la clave expirada.
PASSWORD_VERIFY_FUNCTION	Permite realizar una verificación sofisticada del acceso a la base de datos.

Ejemplo 6

```
SQL> conn / as sysdba;                    [Enter]
Connected.

SQL> create profile pf_pass limit
  2  failed_login_attempts 3
  3  password_lock_time 3/1440;           [Enter]

Profile created.

SQL> alter user egcc
  2  profile pf_pass;                     [Enter]

User altered.
```

Ahora intente iniciar una sesión con el usuario **egcc**, pero con una contraseña inválida, en el cuarto intento la cuenta será bloqueada.

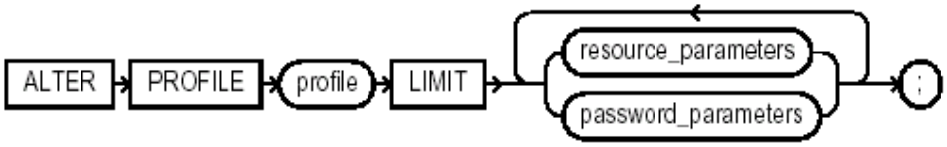
```
SQL> conn egcc/1234                       [Enter]
ERROR:
ORA-01017: invalid username/password; logon denied

SQL> conn egcc/1234                       [Enter]
ERROR:
ORA-01017: invalid username/password; logon denied

SQL> conn egcc/1234                       [Enter]
ERROR:
ORA-01017: invalid username/password; logon denied

SQL> conn egcc/1234                       [Enter]
ERROR:
ORA-28000: the account is locked
```

Después de 3 minutos intente conectarse nuevamente pero con la contraseña correcta.

```
SQL> conn egcc/egcc                       [Enter]
Connected.
```

Modificando Perfiles

Sintaxis



Ejemplo 7

Modifiquemos el perfil pf_pass de tal manera que el número de intentos para bloquear la clave sea 2.

```
SQL> conn / as sysdba;                [Enter]
Connected.

SQL> alter profile pf_pass limit
  2  failed_login_attempts 2;         [Enter]

Profile altered.
```

Ahora puede probar el funcionamiento del perfil.

Eliminar Perfiles

Sintaxis

Ejemplo 8

```
SQL> conn / as sysdba;                  [Enter]
Connected.

SQL> select username, profile
  2  from dba_users
  3  where username = 'EGCC';           [Enter]

USERNAME                       PROFILE
------------------------------ --------------------
EGCC                           PF_PASS

SQL> drop profile pf_pass cascade;      [Enter]

Profile dropped.

SQL> select username, profile, account_status
  2  from dba_users
  3  where username = 'EGCC';           [Enter]

USERNAME                       PROFILE
------------------------------ --------------------
EGCC                           DEFAULT
```

Contraseñas Complejas

Podemos utilizar contraseñas complejas que pueden ser verificadas por una función creada en PL/SQL.

Esta función debe tener el siguiente formato:

```
routine_name
(
   userid_parameter IN VARCHAR(30),
   password_parameter IN VARCHAR (30),
   old_password_parameter IN VARCHAR (30)
)
RETURN BOOLEAN
```

Oracle provee una función (UTLPWDMG.SQL) que se encuentra en **%ORACLE_HOME%/RDBMS\ADMIN**, esta función establece los siguiente parametros:

- La contraseña debe tener una longitud mínima de 4 caracteres.
- La contraseña no debe ser igual a nombre de usuario.
- La contraseña debe tener por lo menos una letra, un número, y un signo de puntuación.
- La contraseña no debe ser simple o una palabra obvia.
- La contraseña debe ser diferente a la contraseña anterior en por lo menos tres caracteres.

Ejemplo 9

Proceda a crear la función contenida en el archivo **UTLPWDMG.SQL**.

```
SQL> conn / as sysdba                                   [Enter]
Connected.

SQL> CREATE OR REPLACE FUNCTION verify_function
  2  (username varchar2,
  3    password varchar2,
  4    old_password varchar2)
  5    RETURN boolean IS
  6     n boolean;
  7     m integer;
  8     differ integer;
  9     isdigit boolean;
 10     ischar  boolean;
 11     ispunct boolean;
 12     digitarray varchar2(20);
 13     punctarray varchar2(25);
 14     chararray varchar2(52);
 15  BEGIN
 16     digitarray:= '0123456789';
 17     chararray:= 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
 18     punctarray:='!"#$%&()``*+,-/:;<=>?_';
 19
 20     -- Check if the password is same as the username
 21     IF NLS_LOWER(password) = NLS_LOWER(username) THEN
 22       raise_application_error(-20001, 'Password same as or similar to user');
 23     END IF;
 24
 25     -- Check for the minimum length of the password
 26     IF length(password) < 4 THEN
 27        raise_application_error(-20002, 'Password length less than 4');
 28     END IF;
 29
 30     -- Check if the password is too simple. A dictionary of words may be
 31     -- maintained and a check may be made so as not to allow the words
 32     -- that are too simple for the password.
 33     IF NLS_LOWER(password) IN ('welcome', 'database', 'account', 'user',
             'password', 'oracle', 'computer', 'abcd') THEN
 34        raise_application_error(-20002, 'Password too simple');
 35     END IF;
 36
 37     -- Check if the password contains at least one letter, one digit and one
 38     -- punctuation mark.
 39     -- 1. Check for the digit
 40     isdigit:=FALSE;
 41     m := length(password);
 42     FOR i IN 1..10 LOOP
 43        FOR j IN 1..m LOOP
 44           IF substr(password,j,1) = substr(digitarray,i,1) THEN
 45              isdigit:=TRUE;
 46               GOTO findchar;
 47           END IF;
 48        END LOOP;
 49     END LOOP;
 50     IF isdigit = FALSE THEN
 51        raise_application_error(-20003, 'Password should contain at least one
             digit, one character and one punctuation');
 52     END IF;
 53     -- 2. Check for the character
 54     <<findchar>>
 55     ischar:=FALSE;
 56     FOR i IN 1..length(chararray) LOOP
 57        FOR j IN 1..m LOOP
 58           IF substr(password,j,1) = substr(chararray,i,1) THEN
 59              ischar:=TRUE;
 60               GOTO findpunct;
 61           END IF;
 62        END LOOP;
 63     END LOOP;
```

```
 64     IF ischar = FALSE THEN
 65        raise_application_error(-20003, 'Password should contain at least one \
 66                digi, one character and one punctuation');
 67     END IF;
 68     -- 3. Check for the punctuation
 69     <<findpunct>>
 70     ispunct:=FALSE;
 71     FOR i IN 1..length(punctarray) LOOP
 72        FOR j IN 1..m LOOP
 73           IF substr(password,j,1) = substr(punctarray,i,1) THEN
 74              ispunct:=TRUE;
 75               GOTO endsearch;
 76           END IF;
 77        END LOOP;
 78     END LOOP;
 79     IF ispunct = FALSE THEN
 80        raise_application_error(-20003, 'Password should contain at least one \
 81                digi, one character and one punctuation');
 82     END IF;
 83
 84     <<endsearch>>
 85     -- Check if the password differs from the previous password by at least
 86     -- 3 letters
 87     IF old_password IS NOT NULL THEN
 88       differ := length(old_password) - length(password);
 89
 90       IF abs(differ) < 3 THEN
 91         IF length(password) < length(old_password) THEN
 92           m := length(password);
 93         ELSE
 94           m := length(old_password);
 95         END IF;
 96
 97         differ := abs(differ);
 98         FOR i IN 1..m LOOP
 99           IF substr(password,i,1) != substr(old_password,i,1) THEN
100             differ := differ + 1;
101           END IF;
102         END LOOP;
103
104         IF differ < 3 THEN
105           raise_application_error(-20004, 'Password should differ by at \
106           least 3 characters');
107         END IF;
108       END IF;
109     END IF;
110     -- Everything is fine; return TRUE ;
111     RETURN(TRUE);
112   END;
113   /                    [Enter]

Function created.
```

Ahora crearemos un perfil utilizando la función.

```
SQL> create profile pf_func limit
  2  password_verify_function verify_function;    [Enter]

Profile created.
```

Ahora intentemos crear un nuevo usuario, cuya contraseña sea igual al nombre de usuario.

```
SQL> create user operador
  2  identified by operador
  3  profile pf_func;                [Enter]
create user operador
*
ERROR at line 1:
ORA-28003: password verification for the specified password failed
ORA-20001: Password same as or similar to user
```

Un usuario con contraseña simple.

```
SQL> create user operador
  2  identified by oracle
  3  profile pf_func;                [Enter]
create user operador
*
ERROR at line 1:
ORA-28003: password verification for the specified password failed
ORA-20002: Password too simple
```

Un usuario que cumpla con las reglas.

```
SQL> create user operador
  2  identified by audaz#97
  3  profile pf_func;                [Enter]

User created.
```

Verificar Información sobre Usuarios y Perfiles

Usuarios y su Información Asociada

Para eso contamos con tres vistas: DBA_USERS, ALL_USERS, USER_USERS.

Ejemplo 10

```
SQL> select username, profile, account_status from dba_users;  [Enter]

USERNAME                PROFILE                ACCOUNT_STATUS
----------------------- ---------------------- ------------------------------
SYS                     DEFAULT                OPEN
SYSTEM                  DEFAULT                OPEN
DBSNMP                  DEFAULT                OPEN
GCORONEL                DEFAULT                OPEN
EGCC                    DEFAULT                LOCKED(TIMED)
WMSYS                   DEFAULT                EXPIRED & LOCKED
ORDPLUGINS              DEFAULT                EXPIRED & LOCKED
CTXSYS                  DEFAULT                EXPIRED & LOCKED
ANONYMOUS               DEFAULT                EXPIRED & LOCKED
PM                      DEFAULT                EXPIRED & LOCKED
OE                      DEFAULT                EXPIRED & LOCKED
```

Tablaspaces y Quotas

La vista que debemos utilizar es: DBA_TS_QUOTAS y USER_TS_QUOTAS.

Ejemplo 11

```
SQL> alter user egcc
  2  default tablespace ts_alm
  3  quota 500K on ts_alm;           [Enter]

User altered.

SQL> select * from dba_ts_quotas;    [Enter]

TABLESPACE_NAME USERNAME       BYTES  MAX_BYTES   BLOCKS  MAX_BLOCKS
--------------- ------------- ------ ---------- -------- -----------
TS_ALM          EGCC               0     516096        0          63
```

Perfiles y Limites asignados

En este caso la vista que debemos utilizar es: DBA_PROFILES.

```
SQL> select * from dba_profiles order by 1;       [Enter]

PROFILE         RESOURCE_NAME                  RESOURCE LIMIT
--------------- ------------------------------ -------- --------------
DEFAULT         COMPOSITE_LIMIT                KERNEL   UNLIMITED
DEFAULT         FAILED_LOGIN_ATTEMPTS          PASSWORD UNLIMITED
DEFAULT         PASSWORD_LIFE_TIME             PASSWORD UNLIMITED
DEFAULT         PASSWORD_REUSE_TIME            PASSWORD UNLIMITED
DEFAULT         PASSWORD_REUSE_MAX             PASSWORD UNLIMITED
DEFAULT         PASSWORD_VERIFY_FUNCTION       PASSWORD NULL
DEFAULT         PASSWORD_LOCK_TIME             PASSWORD UNLIMITED
DEFAULT         PASSWORD_GRACE_TIME            PASSWORD UNLIMITED
DEFAULT         PRIVATE_SGA                    KERNEL   UNLIMITED
DEFAULT         CONNECT_TIME                   KERNEL   UNLIMITED
DEFAULT         IDLE_TIME                      KERNEL   UNLIMITED
DEFAULT         LOGICAL_READS_PER_CALL         KERNEL   UNLIMITED
DEFAULT         LOGICAL_READS_PER_SESSION      KERNEL   UNLIMITED
DEFAULT         CPU_PER_CALL                   KERNEL   UNLIMITED
DEFAULT         CPU_PER_SESSION                KERNEL   UNLIMITED
DEFAULT         SESSIONS_PER_USER              KERNEL   UNLIMITED
PF_CPU          COMPOSITE_LIMIT                KERNEL   DEFAULT
PF_CPU          PRIVATE_SGA                    KERNEL   DEFAULT
PF_CPU          CONNECT_TIME                   KERNEL   DEFAULT
PF_CPU          PASSWORD_GRACE_TIME            PASSWORD DEFAULT
PF_CPU          IDLE_TIME                      KERNEL   DEFAULT
PF_CPU          PASSWORD_LOCK_TIME             PASSWORD 1
PF_CPU          LOGICAL_READS_PER_CALL         KERNEL   DEFAULT
PF_CPU          PASSWORD_VERIFY_FUNCTION       PASSWORD DEFAULT
PF_CPU          LOGICAL_READS_PER_SESSION      KERNEL   DEFAULT
PF_CPU          FAILED_LOGIN_ATTEMPTS          PASSWORD 3
PF_CPU          PASSWORD_LIFE_TIME             PASSWORD DEFAULT
PF_CPU          PASSWORD_REUSE_TIME            PASSWORD 1
PF_CPU          PASSWORD_REUSE_MAX             PASSWORD UNLIMITED
PF_CPU          CPU_PER_CALL                   KERNEL   DEFAULT
PF_CPU          CPU_PER_SESSION                KERNEL   1
PF_CPU          SESSIONS_PER_USER              KERNEL   DEFAULT
PF_FUNC         COMPOSITE_LIMIT                KERNEL   DEFAULT
PF_FUNC         PRIVATE_SGA                    KERNEL   DEFAULT
PF_FUNC         CONNECT_TIME                   KERNEL   DEFAULT
PF_FUNC         PASSWORD_GRACE_TIME            PASSWORD DEFAULT
PF_FUNC         IDLE_TIME                      KERNEL   DEFAULT
PF_FUNC         PASSWORD_LOCK_TIME             PASSWORD DEFAULT
PF_FUNC         LOGICAL_READS_PER_CALL         KERNEL   DEFAULT
PF_FUNC         PASSWORD_VERIFY_FUNCTION       PASSWORD VERIFY_FUNCTION
PF_FUNC         LOGICAL_READS_PER_SESSION      KERNEL   DEFAULT
PF_FUNC         CPU_PER_SESSION                KERNEL   DEFAULT
PF_FUNC         PASSWORD_LIFE_TIME             PASSWORD DEFAULT
PF_FUNC         PASSWORD_REUSE_TIME            PASSWORD DEFAULT
PF_FUNC         PASSWORD_REUSE_MAX             PASSWORD DEFAULT
PF_FUNC         CPU_PER_CALL                   KERNEL   DEFAULT
PF_FUNC         SESSIONS_PER_USER              KERNEL   DEFAULT
PF_FUNC         FAILED_LOGIN_ATTEMPTS          PASSWORD DEFAULT
```

También contamos con otras vistas que pueden resultar útiles: USER_PASSWORD_LIMITS, USER_RESOURCE_LIMITS, V$SESSION, etc.

Página en Blanco

Lección 12

Privilegios y Roles

Contenido

- Entendiendo Privilegios y Roles 164
  - Privilegios del Sistema 164
  - Privilegios de Objetos 165
  - Roles de Usuarios 167
- Administración de Roles 169
  - Creación de Roles 169
  - Eliminando Roles 170
- Concediendo Privilegios y Roles 171
  - Concediendo Privilegios del Sistema 172
  - Concediendo Privilegios de Objetos 173
- Revocando Privilegios y Roles 176
  - Revocando Privilegios del Sistema 177
  - Revocando Privilegios de Objetos 178
- Verificar Información sobre Privilegios y Roles 180
  - Privilegios del sistema 180
  - Roles 181
  - Privilegios de Objetos 181
  - Privilegios Actuales en el Dominio de una sesión 182

Entendiendo Privilegios y Roles

Un privilegio es un derecho para ejecutar un particular tipo de instrucción SQL, o un derecho para acceder a un objeto de otro usuario.

Los roles, son creados por los usuarios (normalmente administradores) y se usan para agrupar privilegios y otros roles. Estos son un medio para asignar múltiples privilegios y roles a los usuarios.

Privilegios del Sistema

Son más de 100 privilegios del sistema. Cada privilegio del sistema le permite a un usuario realizar una operación particular de la base de datos o clase de operación de la base de datos.

Acceso a los objetos en el esquema SYS

Usuarios con los privilegios explícitos o aquéllos que se conectan con los privilegios administrativos (SYSDBA) pueden acceder a los objetos del esquema SYS. Otros necesitan que se les habilite el acceso concediéndoles cualquiera de los siguientes roles:

Rol	Descripción
SELECT_CATALOG_ROLE	Este rol puede concederse a los usuarios para permitirles los privilegios SELECT en todas las vistas de diccionario de datos.
EXECUTE_CATALOG_ROLE	Este rol puede concederse a los usuarios para permitirles los privilegios EXECUTE sobre paquetes y procedimientos del diccionario de datos.
DELETE_CATALOG_ROLE	Este rol puede concederse a los usuarios para permitirles a usuarios eliminar registros de la tabla del sistema de auditoria (AUD$).

Ejemplo 1

Si intentamos consultar la vista DBA_ROLES con el usuarios **scott** obtendríamos un mensaje de error.

```
SQL> conn scott/tiger              [Enter]
Connected.

SQL> select * from dba_roles;      [Enter]
select * from dba_roles
              *
ERROR at line 1:
ORA-00942: table or view does not exist
```

Pero si le concedemos el rol SELECT_CATALOG_ROLE, si lo podríamos hacer.

```
SQL> conn / as sysdba                      [Enter]
Connected.

SQL> grant SELECT_CATALOG_ROLE to scott;   [Enter]

Grant succeeded.

SQL> conn scott/tiger                      [Enter]
Connected.

SQL> select * from dba_roles;              [Enter]

ROLE                           PASSWORD
------------------------------ --------
CONNECT                        NO
RESOURCE                       NO
DBA                            NO
SELECT_CATALOG_ROLE            NO
EXECUTE_CATALOG_ROLE           NO
DELETE_CATALOG_ROLE            NO
```

Privilegios de Objetos

Cada tipo de objeto tiene diferentes privilegios asociados con él.

Podemos especificar ALL [PRIVILEGES] para conceder o revocar todos los privilegios disponibles para un objeto. ALL no es un privilegio; más bien, es un atajo, o una manera de conceder o revocar todos los privilegios del objeto con una palabra en la INSTRUCCIÓN GRANT y REVOKE. Si todos los privilegios del objeto se conceden con el atajo ALL, es posible revocarse en forma individual.

Igualmente, si todos los privilegios se concedieron individualmente, pueden ser revocados especificando ALL.

Ejemplo 2

Si queremos que el usuario **claudia** pueda consultar la tabla dept de scott, debe tener el privilegio **SELECT** sobre esa tabla.

```
SQL> conn / as sysdba                      [Enter]
Connected.

SQL> create user claudia
  2  identified by model;                  [Enter]

User created.

SQL> grant create session to claudia;      [Enter]

Grant succeeded.

SQL> conn scott/tiger                      [Enter]
Connected.

SQL> grant select on dept to claudia;      [Enter]

Grant succeeded.

SQL> conn claudia/model                    [Enter]
Connected.

SQL> select * from scott.dept;             [Enter]

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
```

Roles de Usuarios

Un **rol** agrupa varios privilegios y roles, para que puedan concederse y revocarse simultáneamente de los usuarios.

Oracle provee algunos roles predefinidos para ayudar en la administración de la base de datos. Estos roles, se definen automáticamente para las bases de datos de Oracle cuando ejecutamos scripts estándares que son parte de creación de la base de datos.

Algunos de estos roles predefinidos se muestran en el siguiente cuadro.

ROL	PRIVILEGIOS DEL SISTEMA
CONNECT	CREATE SESSION
RESOURCE	CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE
DBA	Todos los privilegios del sistema con la cláusula WITH ADMIN OPTION.

La forma como se han definido estos privilegios podemos consultarlo en el archivo SQL.BSQ.

Ejemplo 3

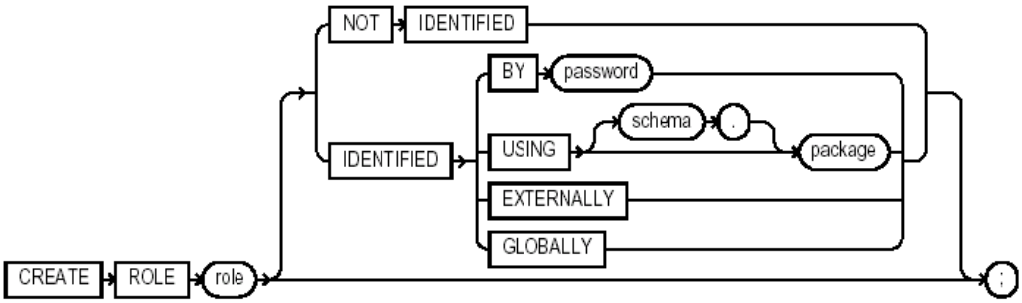
Comprobaremos en este ejemplo el rol CONNECT y RESOURCE.

```
SQL> conn / as sysdba                                   [Enter]
Connected.

SQL> create user sergio
  2  identified by chino
  3  default tablespace users
  4  quota 500K on users;                                [Enter]

User created.

SQL> grant connect, resource to sergio;                 [Enter]

Grant succeeded.

SQL> conn sergio/chino                                  [Enter]
Connected.

SQL> create table equipos(
  2     id number primary key,
  3     nombre varchar2(30)
  4  );                                                  [Enter]

Table created.

SQL> insert into equipos values(1,'Alianza Lima');      [Enter]

1 row created.

SQL> commit;                                            [Enter]

Commit complete.

SQL> select * from equipos;                             [Enter]

        ID NOMBRE
---------- ------------------------------
         1 Alianza Lima
```

Administración de Roles

Creación de Roles

Sintaxis



Ejemplo 4

```
SQL> conn / as sysdba;                [Enter]
Connected.

SQL> create role rl_demo;             [Enter]

Role created.

SQL> create role rl_digitador
  2  identified by admin;             [Enter]

Role created.
```

Eliminando Roles

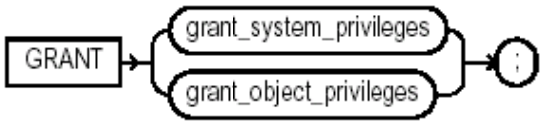
Sintaxis

Ejemplo 5

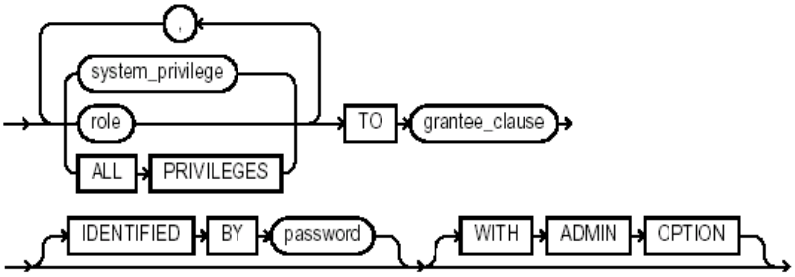
```
SQL> conn / as sysdba;        [Enter]
Connected.

SQL> drop role rl_demo;       [Enter]

Role dropped.
```

Concediendo Privilegios y Roles

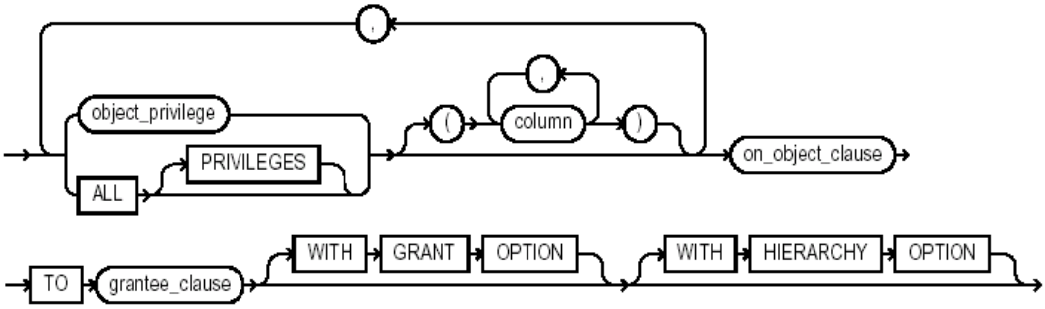
Sintaxis General



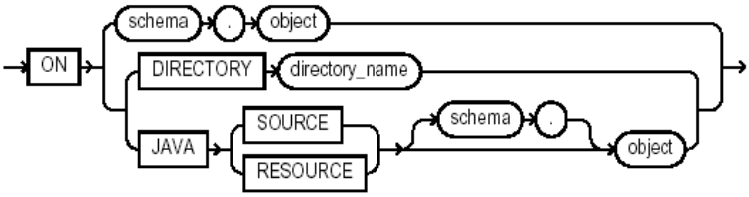
grant_system_privileges::=



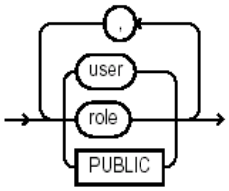
grant_object_privileges::=



on_object_clause::=



grantee_clause::=



Concediendo Privilegios del Sistema

Ejemplo 6

Si queremos que todos los usuarios a los que se le concede el rol rl_digitador, pueda iniciar sesión en la base de datos, debemos concederle el privilegio **CREATE SESSION** al rol.

```
SQL> conn / as sysdba                          [Enter]
Connected.

SQL> grant create session to rl_digitador;     [Enter]

Grant succeeded.
```

También podemos conceder el privilegio directamente al usuario.

```
SQL> create user user01
  2  identified by user01;                [Enter]

User created.

SQL> grant create session to user01;      [Enter]

Grant succeeded.
```

También es posible crear un nuevo usuario.

```
SQL> grant create session
  2  to newuser identified by newuser;    [Enter]

Grant succeeded.
```

Si queremos que el usuario o rol pueda conceder a otros usuarios el privilegio que esta recibiendo, debemos utilizar la cláusula ADMIN OPTION. En el siguiente script estamos creando el usuario **useradmin**, y le esta concediendo el privilegio **CREATE SESSION**, y él también puede conceder este privilegio a otros usuarios.

```
SQL> grant create session
  2  to useradmin identified by admin
  3  with admin option;                   [Enter]

Grant succeeded.

SQL> conn useradmin/admin                 [Enter]
Connected.

SQL> grant create session to user01;      [Enter]

Grant succeeded.
```

Concediendo Privilegios de Objetos

Ejemplo 7

Si queremos que el usuario **newuser** pueda consultar la tabla **dept** de **scott** debe tener el privilegio **SELECT** sobre la tabla.

```
SQL> conn scott/tiger                      [Enter]
Connected.

SQL> grant select on dept to newuser;      [Enter]

Grant succeeded.

SQL> conn newuser/newuser                  [Enter]
Connected.

SQL> select * from scott.dept;             [Enter]

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
```

Si queremos que tenga todos los privilegios sobre una tabla debe usar la palabra **ALL**.

```
SQL> conn scott/tiger                          [Enter]
Connected.

SQL> grant all on emp to newuser;              [Enter]

Grant succeeded.

SQL> conn newuser/newuser                      [Enter]
Connected.

SQL> insert into scott.emp(empno,ename)
  2  values(5555,'Gustavo');                   [Enter]

1 row created.

SQL> select empno, ename from scott.emp;       [Enter]

     EMPNO ENAME
---------- ----------
      7876 ADAMS
      7900 JAMES
      7902 FORD
      7934 MILLER
      5555 Gustavo
```

También podemos conceder el privilegio a un rol, de tal manera que a los usuarios se le concede el rol, y automáticamente tendrá los privilegios del rol.

```
SQL> conn scott/tiger                      [Enter]
Connected.

SQL> grant all on emp to rl_digitador;     [Enter]
Grant succeeded.

SQL> revoke all on emp from newuser;       [Enter]
Revoke succeeded.

SQL> conn / as sysdba                      [Enter]
Connected.

SQL> grant rl_digitador to newuser;        [Enter]
Grant succeeded.
```

Aun cuando **newuser** no tiene privilegios directamente sobre la tabla **scott.emp**, si puede realizar todo tipo de operaciones sobre la tabla por que tiene los privilegios concedidos al rol **rl_digitador**.

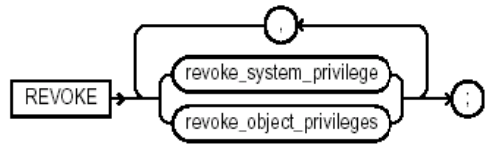
```
SQL> conn newuser/newuser                    [Enter]
Connected.

SQL> select empno, ename from scott.emp;     [Enter]

     EMPNO ENAME
---------- ----------
      7788 SCOTT
      7900 JAMES
      7902 FORD
      7934 MILLER
      5555 Gustavo
```

Los privilegios también se pueden conceder sobre columnas específicas, solo para las instrucciones insert y update.

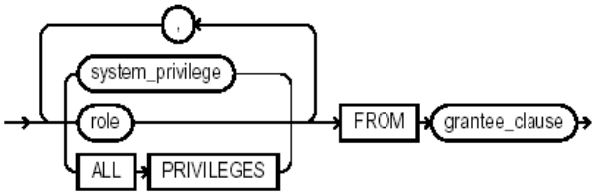
```
SQL> conn / as sysdba                                    [Enter]
Connected.

SQL> grant insert(empno,ename) on scott.emp to sergio;   [Enter]
Grant succeeded.

SQL> conn sergio/chino                                   [Enter]
Connected.

SQL> insert into scott.emp(empno,ename,sal)
  2  values(6666,'Sergio',5000);                         [Enter]
insert into scott.emp(empno,ename,sal)
                   *
ERROR at line 1:
ORA-01031: insufficient privileges


SQL> insert into scott.emp(empno,ename)
  2  values(6666,'Sergio');                              [Enter]

1 row created.
```

Revocando Privilegios y Roles

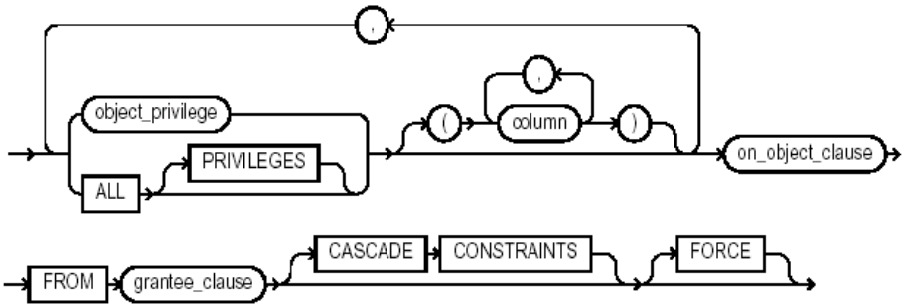
Sintaxis General



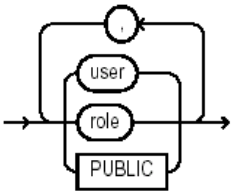
revoke_system_privileges::=



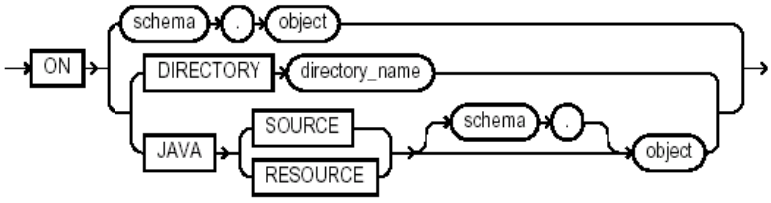
revoke_object_privileges::=



grantee_clause::=



on_object_clause::=



Revocando Privilegios del Sistema

Ejemplo 8

Primero consultemos los privilegios del sistema que han sido asignados a roles u usuarios.

```
SQL> conn / as sysdba;                        [Enter]
Connected.

SQL> select * from dba_sys_privs
  2  where privilege = 'CREATE SESSION'
  3  order by 1;                              [Enter]

GRANTEE                        PRIVILEGE                                ADM
------------------------------ ---------------------------------------- ---
CLAUDIA                        CREATE SESSION                           NO
NEWUSER                        CREATE SESSION                           NO
RL_DIGITADOR                   CREATE SESSION                           NO
USERADMIN                      CREATE SESSION                           YES
USER01                         CREATE SESSION                           NO
```

Ahora debemos revocar el privilegio **CREATE SESSION** al usuario **user01**.

```
SQL> revoke create session from user01;   [Enter]
Revoke succeeded.

SQL> conn user01/user01                   [Enter]
ERROR:
ORA-01045: user USER01 lacks CREATE SESSION privilege; logon denied

Warning: You are no longer connected to ORACLE.
```

Revocando Privilegios de Objetos

Ejemplo 9

Primero veremos los privilegios que se han asignado a los usuarios.

```
SQL> conn / as sysdba                          [Enter]
Connected.

SQL> select table_name, grantee, privilege
  2  from dba_tab_privs
  3  where owner = 'SCOTT';                    [Enter]

TABLE_NAME      GRANTEE                        PRIVILEGE
--------------- ------------------------------ --------------------
DEPT            CLAUDIA                        SELECT
DEPT            NEWUSER                        SELECT
EMP             RL_DIGITADOR                   ALTER
EMP             RL_DIGITADOR                   DELETE
EMP             RL_DIGITADOR                   INSERT
EMP             RL_DIGITADOR                   SELECT
EMP             RL_DIGITADOR                   UPDATE
EMP             RL_DIGITADOR                   ON COMMIT REFRESH
EMP             RL_DIGITADOR                   QUERY REWRITE
EMP             RL_DIGITADOR                   DEBUG
EMP             RL_DIGITADOR                   FLASHBACK
```

Revoquemos el privilegio **SELECT** a **Claudia** de la tabla **DEPT**.

```
SQL> revoke select on scott.dept from claudia;    [Enter]
Revoke succeeded.

SQL> conn claudia/model                           [Enter]
Connected.

SQL> select * from scott.dept;                    [Enter]
select * from scott.dept
                      *
ERROR at line 1:
ORA-00942: table or view does not exist
```

Ejemplo 10

Si revocamos el privilegio de un rol, automáticamente los usuarios que tienen este rol se verán afectados.

```
SQL> conn newuser/newuser                                [Enter]
Connected.

SQL> select ename from scott.emp;                        [Enter]

ENAME
----------
ADAMS
JAMES
FORD
MILLER
Gustavo
Sergio

SQL> conn / as sysdba                                    [Enter]
Connected.

SQL> revoke select on scott.emp from rl_digitador;       [Enter]
Revoke succeeded.

SQL> conn newuser/newuser                                [Enter]
Connected.

SQL> select ename from scott.emp;                        [Enter]
select ename from scott.emp
                        *
ERROR at line 1:
ORA-01031: insufficient privileges
```

Verificar Información sobre Privilegios y Roles

Privilegios del sistema

Ejemplo 11

Listar todos lo privilegios del sistema.

```
SQL> conn / as sysdba                                    [Enter]
Connected.

SQL> select name from system_privilege_map order by 1;    [Enter]

NAME
----------------------------------------
. . .
ALTER SESSION
ALTER SYSTEM
ALTER TABLESPACE
ALTER USER
ANALYZE ANY
ANALYZE ANY DICTIONARY
AUDIT ANY
AUDIT SYSTEM
BACKUP ANY TABLE
BECOME USER
CHANGE NOTIFICATION
COMMENT ANY TABLE
CREATE ANY CLUSTER
. . .
```

Listar los privilegios del sistema concedidos.

```
SQL> select * from dba_sys_privs order by 1;   [Enter]

GRANTEE                        PRIVILEGE                           ADM
------------------------------ ----------------------------------- ---
CLAUDIA                        CREATE SESSION                      NO
CONNECT                        CREATE VIEW                         NO
CONNECT                        CREATE TABLE                        NO
CONNECT                        ALTER SESSION                       NO
CONNECT                        CREATE SEQUENCE                     NO
CONNECT                        CREATE DATABASE LINK                NO
CONNECT                        CREATE SYNONYM                      NO
CONNECT                        CREATE SESSION                      NO
CONNECT                        CREATE CLUSTER                      NO
GCORONEL                       UNLIMITED TABLESPACE                NO
NEWUSER                        CREATE SESSION                      NO
RESOURCE                       CREATE TYPE                         NO
RESOURCE                       CREATE OPERATOR                     NO
RESOURCE                       CREATE INDEXTYPE                    NO
```

También podemos usar USER_SYS_PRIVS.

Roles

Ejemplo 12

Listar los roles asignados a usuarios.

```
SQL> select * from dba_role_privs order by 1;     [Enter]

GRANTEE                        GRANTED_ROLE                   ADM DEF
------------------------------ ------------------------------ --- ---
GCORONEL                       CONNECT                        NO  YES
GCORONEL                       RESOURCE                       NO  YES
NEWUSER                        RL_DIGITADOR                   NO  YES
SCOTT                          CONNECT                        NO  YES
SCOTT                          SELECT_CATALOG_ROLE            NO  YES
SCOTT                          RESOURCE                       NO  YES
SERGIO                         CONNECT                        NO  YES
```

También podemos usar la vista USER_ROLE_PRIVS.

Privilegios de Objetos

Ejemplo 13

Consultar quienes tienen privilegios sobre los objetos de scott.

```
SQL> select table_name, privilege, grantee
  2  from dba_tab_privs
  3  where owner = 'SCOTT';                  [Enter]

TABLE_NAME PRIVILEGE                           GRANTEE
---------- ----------------------------------- ---------------
DEPT       SELECT                              NEWUSER
EMP        ALTER                               RL_DIGITADOR
EMP        DELETE                              RL_DIGITADOR
EMP        INSERT                              RL_DIGITADOR
EMP        UPDATE                              RL_DIGITADOR
EMP        ON COMMIT REFRESH                   RL_DIGITADOR
EMP        QUERY REWRITE                       RL_DIGITADOR
EMP        DEBUG                               RL_DIGITADOR
EMP        FLASHBACK                           RL_DIGITADOR
```

También podemos usar la vista USER_TAB_PRIVS.

Para consultar las columnas especificas sobre las que se concedieron privilegios.

```
SQL> select table_name, column_name, privilege, grantee
  2  from dba_col_privs
  3  where owner = 'SCOTT';       [Enter]

TABLE_NAME COLUMN_NAME     PRIVILEGE       GRANTEE
---------- --------------- --------------- ---------------
EMP        EMPNO           INSERT          SERGIO
EMP        ENAME           INSERT          SERGIO
```

También podemos usar la vista USER_COL_PRIVS.

Privilegios Actuales en el Dominio de una sesión

Ejemplo 14

Vamos a consultar los privilegios de **sergio**, para eso necesitamos conectarnos como scott.

```
SQL> conn sergio/chino    [Enter]
Connected.
```

Consultemos sus roles.

```
SQL> select * from session_roles;      [Enter]

ROLE
------------------------------
CONNECT
```

Ahora consultemos sus privilegios del sistema.

```
SQL> select * from session_privs;    [Enter]

PRIVILEGE
--------------------
CREATE SESSION
ALTER SESSION
CREATE TABLE
CREATE CLUSTER
CREATE SYNONYM
CREATE VIEW
CREATE SEQUENCE
CREATE DATABASE LINK
```