Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Licenciatura en Ingeniería de Sistemas de Información

Sistemas de Bases de Datos II

**Informe General**

**INVESTIGACIONES REALIZADAS POR LOS SEIS GRUPOS**

Profesor:

Henry Lezcano

Integrantes:

Estudiantes del grupo 1IF131

Fecha:

11/18/2020

**Contenido**

[Grupo #1: Paola Márquez, Milagros Aguilar, Anel Atencio y Reynaldo Rojas 3](#_Toc56537262)

[Grupo #2: Karen Cabrera, Medardo Logreira y Sebastian Zamora 13](#_Toc56537263)

[Grupo #3: David Cedeño, Gabriel Grimaldo, Simón Billeke y Daniel Rodríguez 20](#_Toc56537264)

[Grupo #4: Stefanie Arosemena, Emanol González, José Quintero y Andreina Gómez 30](#_Toc56537278)

[Grupo #5: Milagros Campos, Elionays Rosas, Guadalupe Castillo y Alejandro Urriola 37](#_Toc56537279)

[Grupo #6: Rodolfo Marciscano, Julio Chami y Ariel Moreno 41](#_Toc56537280)

[Bibliografías 44](#_Toc56537281)

# Grupo #1: Paola Márquez, Milagros Aguilar, Anel Atencio y Reynaldo Rojas

**ADMINISTRACIÓN Y AUTENTIFICACIÓN DE USUARIOS**

Todo acceso a una base de datos requiere conectar mediante un usuario y contraseña. Dicho usuario dará derecho a utilizar ciertos objetos de la base de datos, pero tendrá restringido (salvo que se trate de un superadministrador) el uso de otros.

A los usuarios se les asigna una serie de privilegios que son los que dan permiso de uso a ciertos objetos. Estos privilegios suelen agruparse en lo que se conoce como roles, que permiten estructurar mejor los permisos que se conceden a los usuarios. El perfil del usuario será el conjunto de permisos y restricciones que se aplican a dicho usuario.

Por ello cuando un usuario conecta debe probar que es quien dice ser (normalmente mediante una contraseña), es decir se autentifica. Por otro lado esta autentificación dará lugar a unos privilegios (unos derechos) y unas restricciones.

**Creación de Usuarios**

Una de las más básicas tareas de un administrador de base de datos es identificar los usuarios. Cada usuario que conecta en la base de datos debe de tener una cuenta. En las cuentas compartidas son difíciles de aplicar una auditoria.

Para crear un usuario utilizamos la sentencia **CREATE USER**. Cuando creas una cuenta como mínimo tienes que asignar un único nombre (username) y una contraseña para poder autenticarse.

Para cambiar alguno de los atributos que se le ha añadido al usuario creado se utiliza la sentencia **ALTER USER**.

A los usuarios de Oracle se les puede asignar la configuración referida a:

* **Nombre de usuario**. No puede repetirse y como máximo debe tener 30 caracteres que sólo podrán contener letras del alfabeto inglés, números, el signo dólar y el signo de guión bajo (**\_**). Además, el nombre no puede comenzar con un número.
* **Configuración física**. Se refiere al espacio asociado al usuario para almacenar sus datos (lo que Oracle llama **tablespace**) y la cuota (límite de almacenamiento) que se le asigna a dicho usuario y mediante la que se establece el espacio máximo que el usuario puede gastar en el tablespace.
* **Perfil asociado**. El perfil del usuario indica los recursos y configuración que tomará el usuario al sistema
* **Privilegios y roles**. Permiten especificar los permisos que posee el usuario.
* **Estado de la cuenta de usuario**:
  + **Abierta**. El usuario puede conectar y realizar sus acciones habituales
  + **Bloqueada**. El usuario no podrá conectar mientras siga en estado bloqueado. El bloqueo lo realiza el DBA:
  + **ALTER USER** usuario **ACCOUNT LOCK**
  + **Expirada**. La cuenta agotó el tiempo máximo asignado a ella. Para salir de este estado, el usuario/a debe resetear su contraseña de usuario.
  + **Expirada y bloqueada**.
  + **Expirada en periodo de gracia**. Está en los últimos momentos de uso antes de pasar a estado de expirada.

**Autenticación Oracle**

Cuando uno se conecta con una instancia de una base de datos la cuenta de usuario debe de estar autenticada. ORACLE provee tres métodos de autenticación para nuestra cuenta de usuario.

* **Autenticación mediante password**

Cuando un usuario conecta con una base de datos verifica que este usuario y la contraseña introducida almacenada en la base de datos, sea correcta. Las contraseñas se guardan encriptadas en la base de datos (en el data dictionary).



* **Autenticación externa**

Cuando un usuario conecta con la base de datos se verifica que el nombre de usuario es el mismo que el nombre de usuario del sistema operativo para permitir la validación.

No se almacenan las cuentas en la base de datos de ninguna forma. Estas cuentas están siempre referidas con OPS$. A partir de la versión 10g puedes configurar **OS\_AUTHENT\_PREFIX** en el spfile.



Mediante **IDENTIFIED EXTERNALLY** decimos a la base de datos que nuestra cuenta es externa y tiene que ser validada con el sistema operativo.

* **Autenticación global**

Cuando un usuario se conecta con la base de datos se verifica globalmente cuando la información pasa por una opción avanzada de seguridad (**ADVANCED SECURITY OPTION**) para la autenticación tal como Kerberos, RADIUS …

Para las cuentas globales no se almacena tampoco nada en la base de datos.



Mediante **IDENTIFIED GLOBALLY** decimos a la base de datos que nuestra cuenta se autentica globalmente, mediante otra opción de seguridad avanzada.

**Borrado De Un Usuario**

Para borrar un usuario utilizamos la sentencia **DROP USER**, podemos opcionalmente incluir la opción **CASCADE**, se utiliza para borrar recursivamente los objetos del usuario que se pretende borrar.



**Cuentas de usuario bloqueadas y no bloqueadas**

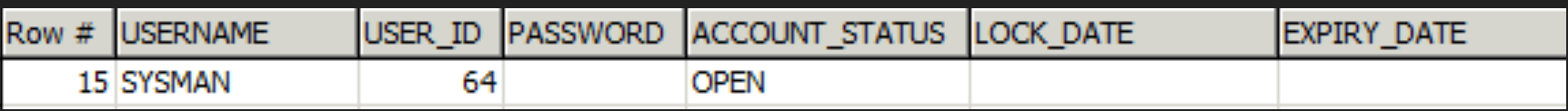
**Cuentas bloqueadas por límite de intentos de inicio de sesión**

Si se está trabajando con una aplicación vinculada a una base de datos y utilizando las capacidades de esta, si tiene soporte para gestionar usuarios, es posible encontrarse con el problema de que un usuario llegue al máximo número de intentos de inicio de sesión y que, en consecuencia, se bloquee su cuenta.

En Oracle para ver la lista de los usuarios que se tienen y sus diferentes propiedades, como si está bloqueado o no, se ejecuta este comando en el editor de SQL:

**SQL**> **SELECT** \* **FROM** dba\_users;

Este comando permitirá observar una tabla como la siguiente, con la información de todos los usuarios que se encuentren en la base de datos:



Cuando uno de estos usuarios llega al máximo número de intentos de acceso y se bloquea, se actualizan el ACCOUNT\_STATUS y el LOCK\_DATE, y por ende, el usuario no podrá entrar a la aplicación.

Los usuarios que tienen bloqueado el acceso a la interfaz de olvido de contraseña debido a un exceso de intentos fallidos de inicio de sesión no pueden iniciar la sesión en dicha interfaz hasta que un administrador desbloquee la cuenta, hasta que el usuario bloqueado (o un usuario con capacidades adecuadas) cambie o reinicialice la contraseña del usuario), o hasta que caduque el bloqueo.

Si se define un valor de tiempo de espera de bloqueo en la directiva de cuentas de *Identity Manager* [1], el bloqueo de una cuenta acabará caducando.

**Cuentas bloqueadas por administrador de base de datos**

Para denegar temporalmente el acceso a la base de datos a una cuenta de usuario en particular, es posible bloquear la cuenta de usuario. Si el usuario intenta conectarse, la base de datos muestra un mensaje de error y no permite la conexión. En este caso, es posible desbloquear la cuenta de usuario cuando se desee volver a permitir el acceso a la base de datos para ese usuario.

Para bloquear una cuenta de usuario se utiliza el siguiente comando:

**SQL**> **ALTER** **USER** username **ACCOUNT** **LOCK**;

Un administrador puede desbloquear una cuenta si tiene control administrativo sobre la organización a la que está afiliado el usuario y posee la capacidad *Desbloquear usuario*.

Un administrador con capacidades adecuadas puede aplicar las operaciones siguientes al estado de bloqueo de un usuario.

* Actualizar (incluido el reabastecimiento de recursos)
* Cambiar o reinicializar la contraseña
* Inhabilitar o habilitar
* Renombrar
* Desbloquear

**Para desbloquear un usuario**

Mediante la tabla **dba\_users** podemos ver los usuarios bloqueados filtrando por los que el **atributo lock\_date no sea null:**



Una vez identificado el usuario bloqueado, lo podremos desbloquear mediante **account unlock**:



Podemos ver como el **account\_status** cambia de “**LOCKED(TIMED)**” a “**OPEN**“:



**Usuarios expirados**

Por defecto, en ORACLE 11g, si los usuarios se crean bajo el ROLE Default, estarán sujetos a una política en la cual la contraseña caducará a los 180 días.

Esto es bueno puesto que te obliga a cambiar la contraseña. El problema es cuando uno desconoce esta política o se le olvida cambiar la contraseña en ese tiempo y el usuario, pasado ese periodo, se bloquea con sus correspondientes consecuencias para la aplicación que lo usa.

Para poder ver si en nuestra base de datos tenemos activa esta política ejecutamos la siguiente consulta:



El resultado que obtenemos en este caso el siguiente:



Si queremos verificar a qué profile está asociado nuestro usuario, lo podemos hacer con la siguiente consulta:



Para cambiar esta política en el ROLE DEFAULT y establecerla a indefinida permitiéndonos así a nosotros cambiar la contraseña del usuario cuando creamos conveniente ejecutaremos la siguiente consulta:



En caso de que nuestro usuario haya sido bloqueado debido a que ya pasaron los 180 días desde su creación o cambio de contraseña, podremos desbloquearlo y cambiar la contraseña de la siguiente forma:



**Usuarios predeterminados de la Base de Datos**

|  |  |  |
| --- | --- | --- |
| **USUARIOS PREDETERMINADOS** | | |
| **Usuario** | **Contraseña** | **Propósito** |
| SYS | CHANGE\_ON\_INSTALL INTERNAL | Usuario utilizada para realizar tareas de administración de la base de datos. |
| SYSTEM | MANAGER | Otro usuario utilizada para realizar tareas de administración de bases de datos. |
| ANONYMOUS | ANONYMOUS | Habilita el acceso HTTP a Oracle XML DB. |
| APEX\_PUBLIC\_USER |  | Es un usuario con privilegios mínimos utilizada para la configuración de Oracle Application Express con Oracle Application Express Listener o Oracle HTTP Server y mod\_plsql. |
| FLOWS\_FILES |  | Es el usuario propietario de los archivos cargados de Oracle Application Express. |
| APEX\_040000 |  | Es el usuario que posee el esquema y los metadatos de Oracle Application Express. |
| OUTLN | OUTLN | Usuario que respalda la estabilidad del plan. La estabilidad del plan le permite mantener los mismos planes de ejecución para las mismas sentencias SQL. OUTLN actúa como una función para administrar de forma centralizada los metadatos asociados con los esquemas almacenados. |
| DIP | DIP | Usuario genérico DIP para procesar eventos propagados por DIP. Esta cuenta sería utilizada por todas las aplicaciones que utilizan el servicio de aprovisionamiento DIP al conectarse a la base de datos. |
| ORACLE\_OCM | ORACLE\_OCM | Este usuario contiene la instrumentación para la recopilación de configuraciones que usa Oracle Configuration Manager. |
| XS$NULL |  | Es un usuario interno que representa la ausencia de un usuario en una sesión. Dado que XS $ NULL no es un usuario, solo la instancia de Oracle Database puede acceder a esta cuenta. |
| MDSYS | MDSYS | Es el usuario administrador de Oracle Spatial and Graph y Oracle Multimedia Locator. |
| CTXSYS | CTXSYS | Es el usuario utilizado para administrar Oracle Text. Oracle Text le permite crear aplicaciones de consulta de texto y aplicaciones de clasificación de documentos. Proporciona indexación, búsqueda de palabras y temas, y capacidades de visualización de texto. |
| DBSNMP | DBSNMP | Es el usuario utilizada por el componente de Agente de administración de Oracle Enterprise Manager para monitorear y administrar la base de datos. |
| XDB | CHANGE\_ON\_INSTALL | Usuario utilizada para almacenar datos y metadatos de Oracle XML DB. Oracle XML DB proporciona almacenamiento y recuperación XML de alto rendimiento para datos de Oracle Database. |
| APPQOSSYS |  | Usuario utilizado para almacenar y administrar todos los datos y metadatos requeridos por Oracle Quality of Service Management. |
| HR | HR | Usuario propietario del esquema de recursos humanos incluido en los esquemas de muestra de Oracle. |

# Grupo #2: Karen Cabrera, Medardo Logreira y Sebastian Zamora

**ADMINISTRACIÓN DE PRIVILEGIOS Y FUNCIONES:**

Oracle pone al alcance del DBA varios niveles de seguridad:

* Seguridad de cuentas para la validación de usuarios.
* Seguridad en el acceso a los objetos de la base de datos.
* Seguridad a nivel de sistema para la gestión de privilegios globales.

**2.2 Seguridad de Objetos**

El acceso a los objetos de la BD se realiza vía privilegios. Estos permiten que determinados comandos sean utilizados contra determinados objetos de la BD. Esto se especifica con el comando GRANT, *conceder*. Los privilegios se pueden agrupar formando lo que se conoce por roles. La utilización de los roles simplifica la administración de los privilegios cuando tenemos muchos usuarios. Los roles pueden ser protegidos con *passwords*, y pueden activarse y desactivarse dinámicamente, con lo que constituyen una capa más de seguridad en el sistema.

**Privilegios de Sistema y de Objetos**

En Oracle existen dos tipos de privilegios de usuario.

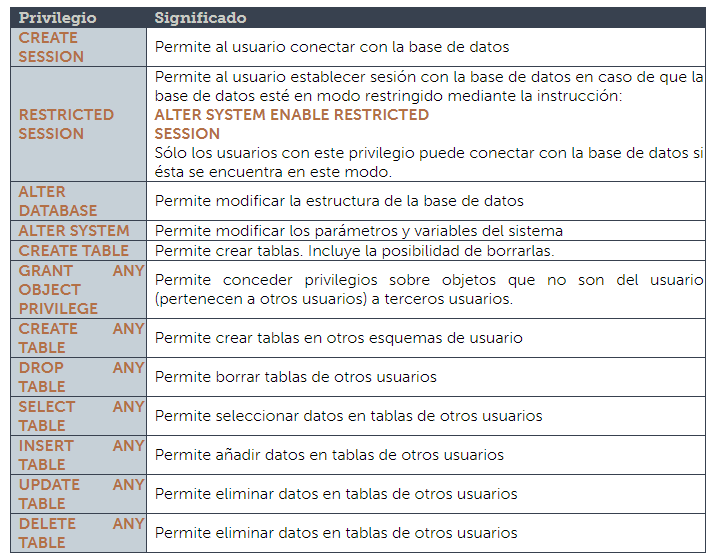
**2.2.1System:** Que permite al usuario hacer ciertas tareas sobre la BD, como por ejemplo crear un Tablespace. Estos permisos son otorgados por el administrador o por alguien que haya recibido el permiso para administrar ese tipo de privilegio. Existen como 100 tipos distintos de privilegios de este tipo.

En general los permisos de sistema, permiten ejecutar comandos del tipo DDL (Data definition Language), como CREATE, ALTER y DROP o del tipo DML (Data Manipulation Language). Oracle 10g tiene mas de 170 privilegios de sistema los cuales pueden ser vistos consultando la vista: SYSTEM\_PRIVILEGE\_MAP

Entre todos los privilegios de sistema que existen, hay dos que son los importantes: SYSDBA y SYSOPER. Estos son dados a otros usuarios que serán administradores de base de datos.

Para otorgar varios permisos a la vez, se hace de la siguiente manera:

SQL> GRANT CREATE USER, ALTER USER, DROP USER TO ahernandez;



**2.2.2 Object:** Este tipo de permiso le permite al usuario realizar ciertas acciones en objetos de la BD, como una Tabla, Vista, un Procedure o Función, etc. Si a un usuario no se le dan estos permisos sólo puede acceder a sus propios objetos (véase USER\_OBJECTS). Este tipo de permisos los da el owner o dueño del objeto, el administrador o alguien que haya recibido este permiso explícitamente (con Grant Option).

Por ejemplo, para otorgar permisos a una tabla Ventas para un usuario particular:

SQL> GRANT SELECT,INSERT,UPDATE, ON analista.venta TO jperez;

Adicionalmente, podemos restringir los DML a una columna de la tabla mencionada. Si quisieramos que este usuario pueda dar permisos sobre la tabla Factura a otros usuarios, utilizamos la cláusula WITH GRANT OPTION. Ejemplo:

SQL> GRANT SELECT,INSERT,UPDATE,DELETE ON venta TO mgarcia WITH GRANT OPTION;

---------------------------------------------------------------

SYSDBA. Con capacidad de parar e iniciar (instrucciones SHUTDOWN y STARTUP) la instancia de base de datos; modificar la base de datos (ALTER DATABASE), crear y borrar bases de datos (CREATE y DROP DATABASE), Crear el archivo de parámetros (CREATE SPFILE), cambiar el modo de archivado de la base de datos, recuperar la base de datos y además incluye el privilegio de sistema RESTRICTED SESSION. En la práctica sus capacidades son las asociadas al usuario SYS.

SYSOPER. Permite lo mismo que el anterior salvo: crear y borrar la base de datos y recuperar en todas las formas la base de datos (hay modos de recuperación que requieren el privilegio anterior).

La vista V$PWFILE\_USERS nos permite examinar a los usuarios administrativos.

<https://youtu.be/f4LJg4qEp40>

**Conceder Privilegios**

Se usa con la instrucción GRANT que funciona así:

***GRANT privilegio1 [,privilegio2[,…]] TO usuario***

***[WITH ADMIN OPTION];***

La opción WITH ADMIN OPTION permite que el usuario al que se le concede el privilegio puede conceder dicho privilegio a otros usuarios. Es, por tanto, una opción a utilizar con cautela.

**Ejemplo:**

**GRANT CREATE SESSION, ALTER SESSION, CREATE TABLE,**

**CREATE VIEW, CREATE SYNONYM, CREATE SEQUENCE,**

**CREATE TRIGGER, CREATE PROCEDURE, CREATE TYPE**

**TO Hlezcano;**

**Revocar Privilegios**

Retira privilegios concedidos a un usuario. Se realiza con la instrucción REVOKE que funciona de esta forma:

REVOKE privilegio1 [,privilegio2 [,…]] FROM usuario;

Al revocar los privilegios, las acciones llevadas a cabo con ellos (borrar, modificar,) no se anulan.

**CREACIÓN DE ROLES**

Los roles se crean usando esta sintaxis

**CREATE ROLE rol [NOT IDENTIFIED |**

**IDENTIFIED {BY password | EXTERNALLY | GLOBALLY | USING package}];**

Por defecto un rol no requiere identificación.

**MODIFICACIÓN DE ROLES**

Disponemos de la instrucción ALTER ROLE permite modificar la configuración del rol. Tiene las mismas opciones que CREATE ROLE y sólo se usa si deseamos establecer un nuevo método para autentificarnos.

**ASIGNAR Y RETIRAR PRIVILEGIOS A ROLES**

Se realiza con la instrucción GRANT y se usa igual que cuando establecemos permisos a los usuarios, en la sintaxis de los comandos GRANT y REVOKE vistas anteriormente, simplemente se indicaría un nombre de rol en lugar de un nombre de usuario. Por ejemplo, si deseamos asignar los privilegios CREATE TABLE y CONNECT a un rol llamado rol1. Se haría:

GRANT CREATE TABLE, CONNECT TO rol1;

De la misma forma, podemos quitar privilegios asignados a un rol mediante el comandol REVOKE:

REVOKE CREATE TABLE FROM rol1;

[5.6.11]asignar roles a usuarios

La sintaxis completa para asignar roles a un usuario es:

GRANT rol1 [,rol2 [,…]]

TO {usuario|rol|PUBLIC [,{usuario|rol|PUBLIC} […] }

[WITH ADMIN OPTION]

Al igual que en las instrucciones anteriores, PUBLIC asigna el rol a todos los usuarios y WITH ADMIN OPTION permite al usuario al que se le concede el rol, conceder él dicho rol a otros usuarios/as.

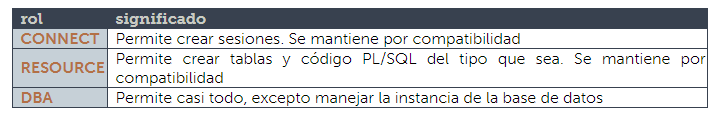
[5.6.12]roles por defecto

Los usuarios tienen una serie de roles por defecto, estos son aquellos roles que van unidos al usuario, de modo que en cuanto un usuario lanza una sesión, los privilegios que contienen sus roles por defecto, comienzan a funcionar.

Cuando asignamos un rol mediante el comando GRANT, este pasa a ser un rol por defecto.

**ROLES PREDEFINIDOS**

Oracle dispone de una serie de roles predefinidos que se pueden asignar a los usuarios. Hay más de cincuenta roles predefinidos. Los clásicos son:



**ACTIVAR Y DESACTIVAR ROLES**

La activación (y también la desactivación) de un rol se realiza mediante SET ROLE (sólo podemos activar y desactivar roles que el usuario tenga asignados mediante la instrucción GRANT). Su sintaxis es:

**SET ROLE**

**{ rol1 [IDENTIFIED BY contraseña]**

**[,rol2 [IDENTIFIED BY contraseña] [,…]]**

**| ALL [EXCEPT rol1 [,rol2 [,…]]]**

**| NONE**

**};**

Las posibilidades son:

* Indicar una lista de roles que serán los que se activen (se usa cuando se habían desactivado)
* Indicar ALL para activar todos los roles, excepto aquellos que se indiquen en la cláusula EXCEPT que quedarán sin activar.
* NONE desactiva todos los roles (incluido el rol por defecto). Sólo quedarán activados los privilegios individuales marcados explícitamente.

La activación y desactivación sólo sirve para la sesión actual, en la siguiente sesión volverán a estar activados sólo los roles por defecto.

[5.6.15]asignar a un usuario un rol por defecto

Cuando se crea un usuario mediante CREATE USER, no disponemos de la posibilidad de asignar un rol por defecto. De hecho, se le asigna automáticamente la opción ALL que hace que todos los roles que se le asignen en el futuro (mediante GRANT) pasarán a ser roles por defecto.

Por ello la instrucción que administra los roles por defecto es ALTER USER:

ALTER USER usuario

DEFAULT ROLE {rol1 [, rol2 [,…]| ALL [EXCEPT rol1 [,rol2[…]] |NONE ]};

La opción ALL coloca a todos los roles como roles por defecto, EXCEPT especifica una lista de roles que no serán colocados como roles por defecto. NONE hace que no haya ningún rol por defecto. Finalmente podemos simplemente especificar la lista de roles que quedarán como roles por defecto.

**BORRAR ROLES**

Lo hace la instrucción DROP ROLE, seguida del rol a borrar. Desde ese momento a los usuarios a los que se habían asignado el rol se les revoca.

# Grupo #3: David Cedeño, Gabriel Grimaldo, Simón Billeke y Daniel Rodríguez

# Límites de Recurso de la Base de Datos

## ¿Qué ocurre cuando se alcanzan los límites de recursos de base de datos?

### **CPU de Proceso**

Cuando la utilización de CPU de proceso de base de datos es elevada, la latencia de las consultas aumenta y puede agotar el tiempo de espera. En estas condiciones, el servicio puede poner en cola las consultas y se proporcionan recursos para la ejecución a medida que el recurso esté disponible.

### **Almacenamiento**

Cuando el espacio de la base de datos alcanza el límite de tamaño máximo, las inserciones y actualizaciones de la base de datos que aumentan el tamaño de los datos producen un error y los clientes reciben un mensaje de error.

### **Memoria**

A diferencia de otros recursos (CPU, nodos de trabajo, almacenamiento), el hecho de alcanzar el límite de memoria no afecta negativamente al rendimiento de las consultas y no provoca errores. En efecto un mayor uso de memoria suele mejorar el rendimiento de las consultas debido a lecturas más rápidas de la memoria, en lugar de lecturas más lentas del almacenamiento.

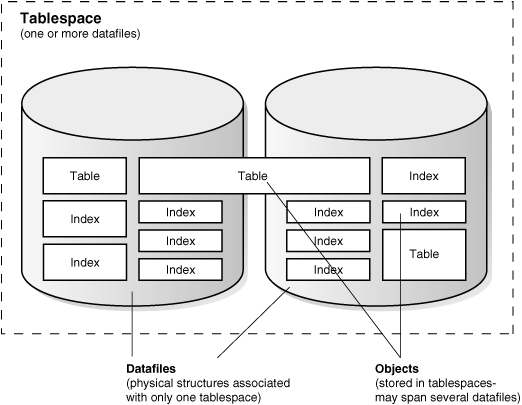
En raras ocasiones, una carga de trabajo suficientemente exigente puede producir una condición de memoria insuficiente, lo que provoca errores de memoria insuficiente.

## **Cuotas de Espacio de Tablas**

Oracle almacena datos de forma lógica en espacios de tabla y físicamente en archivos de datos asociados con el espacio de tabla correspondiente.

* La estructura física se corresponde a los ficheros del sistema operativo: de datos (datafiles), de redo log y de control (controlfiles).
* La estructura lógica está formada por los tablespace y los objetos de un esquema de BD (tablas, vistas, índices,...).

Cada BD estará formada por uno o más tablespaces. Cada tablespace se corresponde con uno o más ficheros de datos. Objetos: tablas, vistas, índices asociados a una tabla, clústeres, etc.



## **¿Qué es un tablespace?**

Una base de datos se divide en una o más unidades de almacenamiento lógicas llamadas tablespace (espacios de tabla). Estos se dividen en unidades lógicas de almacenamiento denominadas segmentos, que además se dividen en extensiones. Las extensiones son una colección de bloques contiguos.

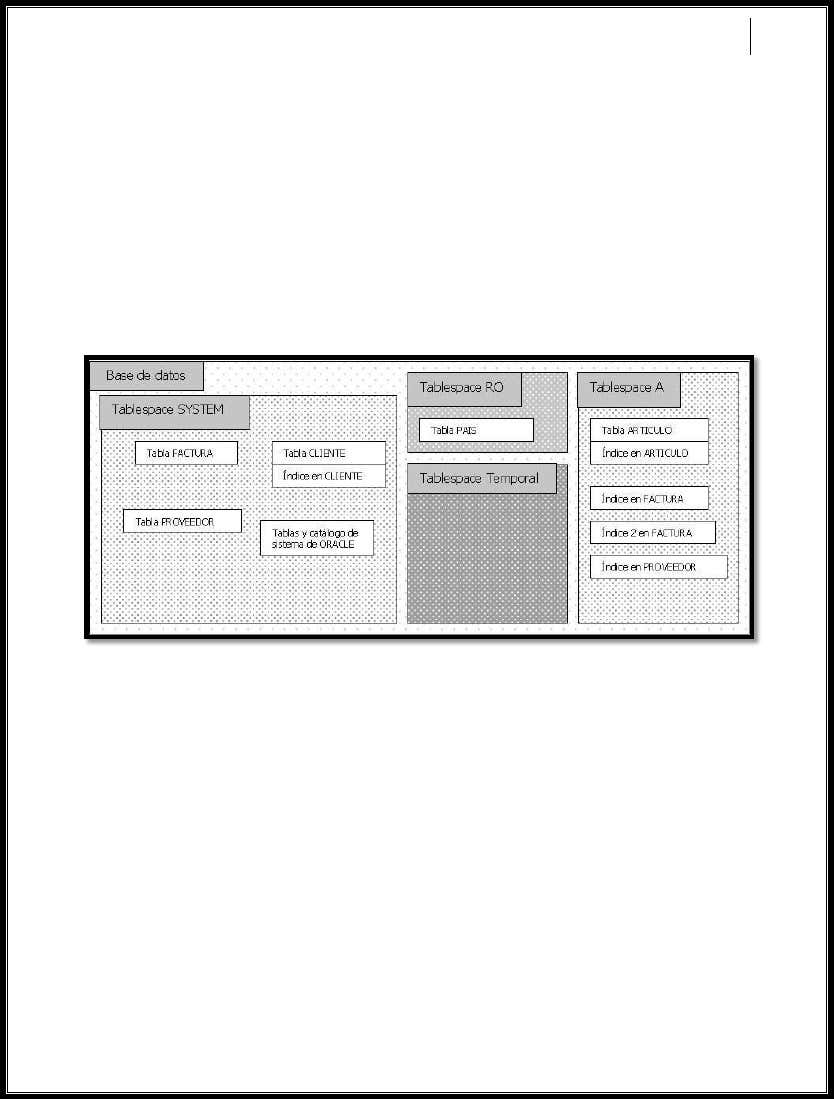
Un tablespace (espacio de tabla) no es un fichero físico en el disco, simplemente es el nombre que tiene un conjunto de propiedades de almacenamiento que se aplican a los objetos (tablas, secuencias…) que se van a crear en la base de datos bajo el tablespace indicado (tablas, secuencias…).

Un objeto en base de datos debe estar almacenado obligatoriamente dentro de un tablespace.

Las propiedades de estos son:

* Localización de los ficheros de datos.
* Especificación de máximas cuotas de consumo de disco.
* Control de la disponibilidad de los datos (en línea o fuera de línea).
* Backup de datos.

Cuando un objeto se crea dentro de un cierto tablespace, este objeto adquiere todas las propiedades antes descritas del tablespace utilizado.



En este esquema podemos ver que, por ejemplo, la tabla ARTÍCULO se almacena dentro del tablespace A, y que por lo tanto tendrá todas las propiedades del tablespace A que pueden ser:

* Sus ficheros de datos están en$ORACLE\_HOME/datos/datos\_tablespace\_A.
* Los objetos no pueden ocupar más de 10Mb de espacio de base de datos.
* En cualquier momento se puede poner fuera de línea todos los objetosde un cierto tablespace. -Se pueden hacer copiar de seguridad sólo deciertos tablespaces.

Si nos fijamos, se puede apreciar que es posible tener una tabla en un tablespace, y los índices de esa tabla en otro. Esto es debido a que los índices no son más que objetos independientes dentro de la base de datos, como lo son las tablas. Y al ser objetos independientes, pueden ir en tablespaces independientes.

El tablespace SYSTEM es uno de los que se crear por defecto en todas las bases de datos Oracle. En él se almacenan todos los datos de sistema, el catálogo y todo el código fuente y compilado de procedimientos PL/SQL.

También es posible utilizar el mismo tablespace para guardar datos de usuario. En el esquema también vemos que hay un tablespace Temporal (en gris oscuro).

Este representa las propiedades que tendrán los objetos que la base de datos cree temporalmente para sus cálculos internos (normalmente para ordenaciones y agrupaciones).

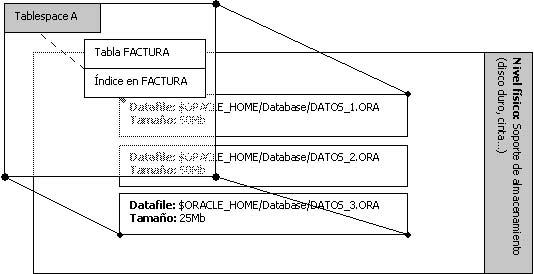
Su creación difiere en una de sus cláusulas de creación. El tablespace RO (Read Only) (en gris claro) difiere de los demás en que es de solo lectura (Read Only), y que por lo tanto todos los objetos en él contenidos pueden recibir órdenes de consulta de datos, pero no de modificación de datos.

Estos pueden residir en soportes de sólo lectura, como pueden ser CDROMs, DVDs, etc.

Cuando se crea un tablespace, éste se crea de lectura/escritura. Después se puede modificar para que sea de solo lectura. Un tablespace puede estar en línea o fuera de ella (Online o Offline), esto es que todos los objetos contenidos en él están a disposición de los usuarios o están inhabilitados para restringir su uso. Cualquier objeto almacenado dentro de un tablespace no podrá ser accedido si este está fuera de línea.

## **¿Qué es un Datafile?**

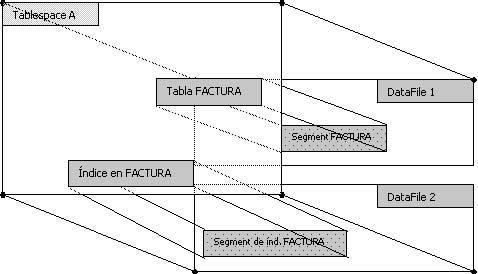
Un datafile es la representación física de un tablespace. Son los "ficheros de datos" donde se almacena la información físicamente. Un datafile puede tener cualquier nombre y extensión (siempre dentro de las limitaciones del sistema operativo), y puede estar localizado en cualquier directorio del disco duro, aunque su localización típica suele ser $ORACLE\_HOME/Database. Un datafile tiene un tamaño predefinido en su creación (por ejemplo 100Mb) y este puede ser alterado en cualquier momento. Cuando creemos un datafile, este ocupará tanto espacio en disco como hayamos indicado en su creación, aunque internamente esté vacío. Oracle hace esto para reservar espacio continuo en disco y evitar así la fragmentación.



En el esquema podemos ver como el “Tablespace A” está compuesto (físicamente) por tres datafiles (DATOS\_1.ORA, DATOS\_2.ORA y DATOS\_3.ORA). Estos tres datafiles son los ficheros físicos que soportan los objetos contenidos dentro del tablespace A. Aunque siempre se dice que los objetos están dentro del tablespace, en realidad las tablas están dentro del datafile, pero tienen las propiedades asociadas al tablespace.

## **¿Qué es un Segment?**

Un segment es aquel espacio reservado por la base de datos, dentro de un datafile, para ser utilizado por un solo objeto. Así una tabla (o cualquier otro objeto) está dentro de su segmento, y nunca podrá salir de él, ya que, si la tabla crece, el segmento también crece con ella. Físicamente, todo objeto en base de datos no es más que un segmento (segmento, trozo, sección) dentro de un datafile. Se puede decir que, un segmento es a un objeto de base de datos, lo que un datafile a un tablespace: el segmento es la representación física del objeto en base de datos.



Esta sería una representación de los segmentos como objetos en Oracle. En la que podemos ver cómo el espacio que realmente se ocupa dentro del datafile es el segment y que cada segmento pertenece a un objeto.

Los tipos de segmentos principales son:

* Segmentos de TABLE: aquellos que contienen tablas
* Segmentos de INDEX: aquellos que contienen índices
* Segmentos de ROLLBACK: aquellos se usan para almacenar información de la transacción activa.
* Segmentos TEMPORALES: aquellos que se usan para realizar operaciones temporales que no pueden realizarse en memoria, tales como ordenaciones o agrupaciones de conjuntos grandes de datos

## **Asignación de un espacio de tabla predeterminado del usuario.**

Cada usuario debe tener un espacio de tabla por defecto. Cuando se crea un objeto esquema en el esquema del usuario y la declaración DDL no especifica un espacio de tablas para contener el objeto, Oracle Database almacena el objeto en el espacio de tablas de usuario por defecto.

La configuración predeterminada para los espacios de tabla por defecto de todos los usuarios es el espacio de tablas SYSTEM. Si un usuario no crea objetos, y no tiene privilegios para hacerlo, entonces este ajuste por defecto estaría bien.

Sin embargo, si existe la posibilidad de crear cualquier tipo de objeto de un usuario, debe asignar específicamente al usuario un espacio de tabla predeterminado, como el espacio de tabla USERS. El uso de un espacio de tabla que no sea SYSTEM reduce la contención entre los objetos del diccionario de datos y objetos de usuario para los mismos archivos de datos. En general, no se debe almacenar datos de usuario en el tablespace SYSTEM.

Al separar los datos de usuario de los datos del sistema, se reduce la probabilidad de problemas con el espacio de tablas SYSTEM, que puede en algunos casos provocar que toda la base de datos deje de funcional. Este espacio de tabla permanente por defecto no es utilizado por los usuarios del sistema, es decir, SYS, SYSTEM, y OUTLN, cuyo valor predeterminado tablas permanentes es SYSTEM. Un espacio de tablas designado como el espacio de tablas permanentes por defecto no se puede quitar.

Para lograr este objetivo, primero debe designar a otro espacio de tabla como el espacio de tabla por defecto permanente. Usted puede utilizar la sentencia ALTER TABLESPACE SQL para modificar el tablespace permanente por defecto por otro espacio de tablas. Tenga en cuenta que esto afectará a todos los usuarios u objetos creados después de la instrucción ALTER DDL. Cuando se especifica el espacio de tabla por defecto para un usuario, especifique también una cuota en ese espacio de tablas.

## **Asignación de una cuota de espacio de tabla para el usuario.**

Podemos asignar a cada usuario una cuota de espacio de tabla para cualquier espacio de tabla (excepto un espacio de tabla temporal). La asignación de una cuota se logra de la forma siguiente:

* Los usuarios con privilegios al crear cierto tipo de objetos pueden crear estos objetos en la tablespace específica.
* Oracle Database limita la cantidad de espacio que se puede asignar para el almacenamiento de objetos de usuario en el espacio de tabla especificando el monto de la cuota.

De forma predeterminada, el usuario no dispone de cuota en cualquier espacio de tablas en la base de datos. Si el usuario tiene el privilegio para crear un objeto de esquema, debe asignar una cuota para permitir al usuario crear objetos. Como mínimo, asignar a los usuarios una cuota para el espacio de tabla por defecto, y las cuotas adicionales para otros espacios de tabla en la que ellos puedan crear objetos. La siguiente sentencia CREATE USER asigna las siguientes cuotas para lostest\_ts y data\_ts espacios de tabla:

CREATE USER jward

IDENTIFIED BY password

DEFAULT TABLESPACE data\_ts

QUOTA 100M ON test\_ts

QUOTA 500K ON data\_ts

TEMPORARY TABLESPACE temp\_ts

PROFILE clerk;

Puede asignar a un usuario o bien cuotas individuales de una determinada cantidad de espacio en disco en cada espacio de tabla o una cantidad ilimitada de espacio de disco en todos los espacios de tabla. Las cuotas específicas impiden que los objetos de un usuario puedan utilizar demasiado espacio en la base de datos.

## La restricción de los límites de cuota para objetos de usuario en un espacio de tabla.

Se puede restringir los límites de cuota para los objetos de usuario en un espacio de tabla mediante la instrucción SQL ALTER USER, para cambiar la cuota actual del usuario a cero. Después de que se le asigna una cuota de cero, los objetos del usuario en el espacio de tablas se mantienen, y el usuario todavía puede crear nuevos objetos, pero los objetos existentes no se asignarán a cualquier nuevo espacio.

Por ejemplo, no se puede insertar datos en una de las tablas existentes de este usuario. La operación fallará enviando el mensaje ORA-1536 error de espacio.

## **Perfiles de límite de recursos:**

Un profile o perfil de usuario es un conjunto de límites de recursos a los que se le asigna un nombre. Los perfiles pueden ser asignados a diferentes usuarios o grupos de usuarios para controlar y limitar el uso de recursos del sistema. Para que estos perfiles funcionen según los valores que vamos asignando, debemos tener presente que el parámetro RESOURCE\_LIMIT debe estar configurado en TRUE.

Para cambiar este parámetro lo hacemos con la instrucción:

ALTER SYSTEM RESOURCE\_LIMIT =TRUE;

Luego del cambio es necesario parar la base de datos y luego levantar nuevamente para que asuma el nuevo parámetro.

La definición de profiles deberá contemplar los siguientes aspectos:

* Nombre: Se indicará nombre del perfil creado.
* Sesión por usuario: Se indicará el número máximo de sesiones que puede abrir un usuario.
* CPU por sesión: Se expresará el máximo de centésimas de segundo de procesador por llamada.
* Tiempo de Conexión: Se indicará el número máximo de minuto de duración de una sesión.
* Tiempo ocioso: Se indicará el número máximo de tiempo ocioso de una sesión.
* Lecturas por sesión: Se indicará el número máximo de lecturas de bloques de datos por sesión.
* Límite compuesto: Se expresará el coste total de recursos por sesión.
* SGA privada: Se expresará el número máximo de octetos de espacio privado de SGA por sesión.

Todo usuario que es creado en la base de datos Oracle debe ser asignado a algún profile, si esto no se realiza el usuario tendrá el profile asignado por default al momento de la creación.



# Grupo #4: Stefanie Arosemena, Emanol González, José Quintero y Andreina Gómez

**Control de acceso**

La definición más generalizada de un sistema de control de acceso hace referencia al mecanismo que en función de la identificación ya autentificada permite acceder a datos o recursos. Básicamente encontramos sistemas de controles de acceso en múltiples formas y para diversas aplicaciones.

Empezamos definiendo este término porque es primordial el tenerlo claro con la explicación que daremos a continuación. Como punto primordial una base de datos también necesita de estos controles de acceso ya que no puede permitir que cualquier tipo de persona acceda a los datos e información de una empresa. Una de las formas de control de acceso que tienen estas base de datos son los perfiles.

**Administración de cuentas de usuario.**

La administración de cuentas de usuario y grupos es una parte esencial de la administración de sistemas dentro de una organización. Pero para hacer esto efectivamente, un buen administrador de sistemas primero debe entender lo que son las cuentas de usuario y los grupos y cómo funcionan.

La razón principal para las cuentas de usuario es verificar la identidad de cada individuo utilizando un computador. Una razón secundaria (pero aún importante) es la de permitir la utilización personalizada de recursos y privilegios de acceso.

Los recursos incluyen archivos, directorios y dispositivos. El control de acceso a estos dispositivos forma una gran parte de la rutina diaria de un administrador de sistemas; a menudo el acceso a un recurso es controlado por grupos. Los grupos son construcciones lógicas que se pueden utilizar para enlazar a usuarios para un propósito común. Por ejemplo, si una organización tiene varios administradores de sistemas, todos ellos se pueden colocar en un grupo administrador de sistema. Luego se le pueden dar permisos al grupo para acceder a recursos claves del sistema. De esta forma, los grupos pueden ser una herramienta poderosa para la administración de recursos y acceso.

**Perfil de Usuario**

Un sistema software permite a los usuarios definir, crear, mantener y controlar el acceso a la base de datos. El SGBD es el software que interactúa con los programas de aplicación del usuario y con la base de datos. Normalmente, un SGBD proporciona la siguiente funcionalidad:

* Permite a los usuarios definir la base de datos, usualmente mediante un lenguaje de definición de datos (DDL, Data Definition Language). El DDL permite a los usuarios especificar las estructuras y tipos de datos y las restricciones aplicables a los datos que hay que almacenar en la base de datos.
* Permite a los usuarios insertar, actualizar, borrar y extraer datos de la base de datos, usualmente mediante un lenguaje de manipulación de datos (DML, Data Manipulation Language). Al disponer de un repositorio centralizado para todos los datos y las descripciones de los datos, el lenguaje DML puede proporcionar un mecanismo general de consulta de esos datos, denominado lenguaje de consulta.

La existencia de un lenguaje de consulta resuelve el problema de los sistemas basados en archivos en los que el usuario tenía que trabajar con un conjunto fijo de consultas, o bien en los que existía una proliferación de programas que provocaban graves problemas de gestión del software. El lenguaje de consulta más común es el lenguaje SQL (Structured Query Language, lenguaje estructurado dé consulta), que es ahora tanto el estándar formal como el estándar defacto para los SGBD relacionales.

Ademas el SGBD debe proporcionar un acceso controlado a la base de datos. Por ejemplo, puede proporcionar:

* Un sistema de seguridad, que evita que los usuarios no autorizados accedan a la base de datos
* un sistema de integridad, que mantiene la coherencia de los datos almacenados
* un sistema de control de concurrencia que permite el acceso compartido a la base de datos
* un sistema de control de recuperación, que restaura la base de datos a un estado previo coherente después de cada fallo hardware o software

**Diferencia entre Perfil y Rol en una BD Oracle**

Aunque a primera vista los términos de rol y perfil parezcan semejantes en realidad estos 2 difieren en su aplicación técnica a la hora que nos adentramos mas en lo que son las bases de datos. Pero en resumidas cuentas podemos decir que:

* Un perfil es un conjunto de límites a los recursos de la base de datos.
* Un rol es conjunto de privilegios agrupados.

Podemos decir que un perfil es aquel que administra una serie de recursos de las bases de datos mientras que el rol es aquel brinda los permisos de accesos a ciertos perfiles para que estos no puedan a acceder a todos los recursos de la base de datos sino a los necesarios.

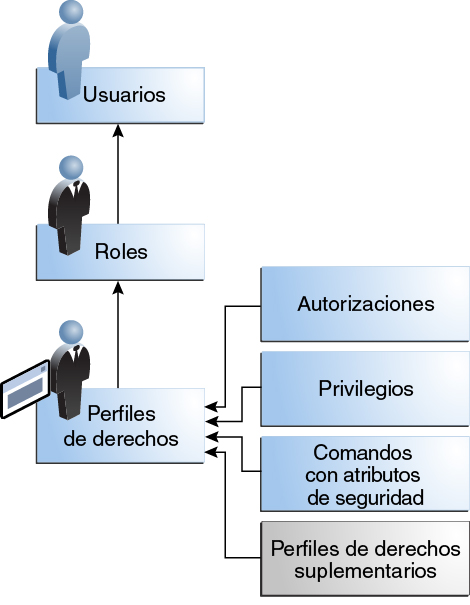


Ilustración 1: Jerarquía de Acceso a una Base de datos

Los limites que puede tener un perfil van desde:

* Tiempo de inactividad en la base de datos
* Cantidad de intentos fallidos de conexión por contraseña
* Cantidad de sesiones del usuario
* Tiempo para cambiar alguna contraseña
* Máxima cantidad de bytes de espacio privado reservado en la SGA.
* Máximo numero de bloques de base de datos leídos por sesión.

**Tipos de Perfiles**

En una base datos podemos decir que existen dos tipos de perfiles que son:

* El perfil Default
* El perfil que usted define

El perfil que nosotros definimos puede contar o no con una serie de parámetros que nosotros deberemos de definir dependiendo del caso que se presente.

**Perfil Default (Predeterminado)**

Hay un perfil llamado DEFAULT que se aplica automáticamente a todos los usuarios y que les da recursos ilimitados sobre la base de datos. Para limitar el número de recursos se debe de activar (poniéndola el valor TRUE) la variable de sistema RESOURCE\_LIMIT (que por defecto está a FALSE). Para activarlo escribiríamos lo siguiente:



Ilustración 2: Limitación de recursos

**Parámetros de los Perfiles**

Para que los perfiles funcionen, el parámetro de la base de datos resource\_limit, deberá tener el valor a true como ya habíamos mostrado anteriormente, pero si desea verificar que esta esté en valor true pondrá lo siguiente:

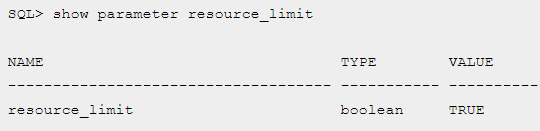


Ilustración 3: Verificar los límites de recursos

Ahora volviendo a los parámetros, existen 2 tipos de parámetros que se pueden aplicar a los perfiles que son:

* **Perfiles de manejo de contraseñas**: Gestionan el funcionamiento de las contraseñas para el usuario.
* **Perfiles relacionados con el uso de recursos**: Establecen el máximo o mínimo uso de recursos de la base de datos por parte del usuario.

Tabla 1: Variables relacionadas a contraseñas

|  |  |
| --- | --- |
| Variable de perfil | Significado |
| FAILED\_LOGIN\_ATTEMPTS | Número consecutivo de errores en las contraseñas antes de bloquear la cuenta. Por defecto son 10 |
| PASSWORD\_LOCK\_TIME | Número de días hasta que se bloquea una cuenta si se supera el límite de intentos al meter una contraseña. Por defecto es uno |
| PASSWORD\_LIFE\_TIME | Números de días que tiene vigencia una contraseña. Por defecto es 180 |
| PASSWORD\_GRACE\_TIME | Días que la contraseña se la concede un periodo extra de gracia tras consumir su tiempo de vida. Por defecto es 7 |
| PASSWORD\_REUSE\_TIME | Número de días que una contraseña puede ser reutilizada |
| PASSWORD\_VERIFY\_FUNCTION | Función a la que se invoca cuando se modifica una contraseña con el fin de verificar su validez en base a las reglas de complejidad que deseemos |

Tabla 2: Variables relacionadas al uso de los recursos

|  |  |
| --- | --- |
| Variable de perfil | Significado |
| SESSIONS\_PER\_USER | Número de conexiones de usuario concurrentes que se permiten. |
| CPU\_PER\_SESSION | Límite de tiempo (en centésimas de segundo) que se permite a un usuario utilizar la CPU antes de ser echado del sistema. De esa forma se evitan peligros de rendimiento |
| CPU\_PER\_CALL | Como la anterior pero referida a cada proceso |
| PRIVATE\_SGA | Para conexiones en instalaciones de servidor compartido, número de KB que puede consumir cada sesión en la zona de memoria compartida (SGA) |
| CONNECT\_TIME | Minutos como máximo que se permite a una sesión |
| IDLE\_TIME | Minutos máximos de inactividad de una sesión |
| LOGICAL\_READS\_PER\_SESSION | Máximo número de bloques leídos en una sesión |
| LOGICAL\_READS\_PER\_CALL | Máximo número de bloques leídos por un proceso |
| COMPOSITE\_LIMIT | Máximo número de recursos consumidos por una sesión. Es la media ponderada de varios parámetros anteriores |

**Creación de perfiles**

Para crear un perfil con los parámetros que nosotros deseemos deberemos de seguir la siguiente sintaxis:



Ilustración 4: Sintaxis de creación de Perfil

Los parámetros para especificar son los que aparecen en las tablas anteriores. A cada parámetro se le indica un valor, o bien la palabra DEFAULT si deseamos que tome su valor por defecto, o bien UNLIMITED para indicar que el parámetro tomará un valor de infinito.

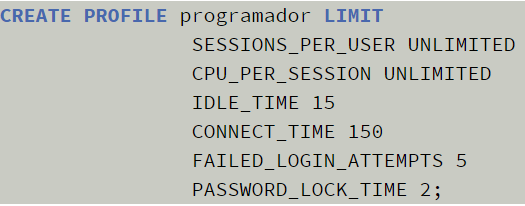


Ilustración 5: Ejemplo de creación de Perfil

Además, también podemos modificar estos perfiles con el comando “Alter profile” que sigue la misma sintaxis que el comando “Create profile”. Y si deseamos borrar algún perfil también podemos hacerlo con el comando “Drop profile”. En este caso es DROP PROFILE seguida del nombre del perfil a eliminar. Se puede usar la palabra CASCADE para eliminar todas las restricciones que impidan borrar el perfil.



Ilustración 6: Eliminar un Perfil

Asignación de un perfil a un usuario

Cada usuario tiene un solo perfil. La instrucción de creación de usuarios (Create User) dispone de la cláusula “Profile” para indicar el perfil que se asigna a ese usuario.

Si lo que deseamos es asignar un perfil a un usuario después de haberle creado, disponemos de la instrucción “Alter User” con la que podemos indicar el perfil.



Ilustración 7: Asignación de un perfil

# Grupo #5: Milagros Campos, Elionays Rosas, Guadalupe Castillo y Alejandro Urriola

**Respaldo y Recuperación de una base de datos**



En general, copia de seguridad y recuperación se refieren a las diversas estrategias y procedimientos involucrados en la protección de tu base de datos contra la pérdida de información, y la reconstrucción después de cualquier tipo de pérdida.

Una copia de seguridad es una copia tu base de datos que se puede usar para reconstruirla. Se pueden dividir en copias físicas y copias lógicas.

**Copias de seguridad físicas de base de datos**

Las copias de seguridad físicas son respaldos de los archivos físicos utilizados para almacenar y recuperar tu información, como archivos de datos, archivos de control y registros. En última instancia, cada copia de seguridad física es una copia de archivos que almacenan información de la base de datos en otra ubicación, ya sea en un disco o en algún almacenamiento fuera de línea como la cinta.

**Copias de seguridad lógicas de base de datos**

Las copias de seguridad lógicas contienen datos lógicos (por ejemplo, tablas o procedimientos almacenados) exportados desde una base de datos y almacenados en un archivo binario, para luego volver a importarlos a una base de datos utilizando la utilidad de importación correspondiente.

En la técnica de respaldo lógico, las utilidades importar / exportar se utilizan para crear el respaldo de la base de datos. Una copia de seguridad lógica realiza una copia de seguridad del contenido. Se puede usar una copia de seguridad lógica para restaurar la base de datos a la última copia de seguridad. Sin embargo, a diferencia de la copia de seguridad física, esta no se debe usar para crear una copia de seguridad del sistema operativo porque la restauración con este enfoque permitiría corregir los archivos de datos dañados. Por lo tanto, en estas situaciones, se deben preferir las copias de seguridad físicas.

Las copias de seguridad físicas son la base de cualquier estrategia de copia de seguridad y recuperación. Las copias de seguridad lógicas son un complemento útil para las copias de seguridad físicas en muchas circunstancias, pero no son una protección suficiente contra la pérdida de datos sin copias de seguridad físicas.

A menos que se especifique lo contrario, el término “copia de seguridad” tal como se utiliza en la documentación de copia de seguridad y recuperación se refiere a copias de seguridad físicas, y hacer una copia de seguridad de una parte o la totalidad de su base de datos es tomar algún tipo de copia de seguridad física. El enfoque en el conjunto de documentación de copia de seguridad y recuperación estará casi exclusivamente en las copias de seguridad físicas.

**¿Qué significa importar y exportar una base de datos?**

Importar una base de datos es un proceso para restaurar una base de datos con una copia de seguridad/archivo de volcado de memoria (dump de memoria) mientras que exportar una base de datos es el proceso de crear la copia de seguridad/volcado de memoria de una base de datos.



Exportar un dato de base de datos significa guardar una base de datos de origen en un archivo, que luego puede utilizarse para su almacenamiento o distribución.

Importar un dato de base de datos significa restaurar los datos de ese archivo a una base de datos de destino. Se puede importar una base de datos al mismo servidor de base de datos o a otro. La única restricción es que las bases de datos de origen y de destino deben ser del mismo tipo, por ejemplo, MySQL.

**¿Por qué necesitas tener copias de seguridad?**

Si bien hay varios tipos de problemas que pueden detener el funcionamiento normal de una base de datos o afectar las operaciones de importar / exportar, solo dos requieren intervención de expertos y recuperación de medios: fallas y errores del usuario.

Los errores del usuario ocurren cuando, ya sea debido a un error en la lógica de la aplicación o un error manual, los datos en tu base de datos se cambian o se eliminan incorrectamente. La pérdida de información debido a un error del usuario incluye pasos en falso tales como descartar tablas importantes o eliminar o cambiar el contenido de una tabla. Si bien la capacitación de los usuarios y la administración cuidadosa de los privilegios pueden evitar la mayoría de los errores de los usuarios, tu [estrategia de respaldo](https://serman.com/blog-recuperacion-datos/quiero-respaldar-mis-datos/) determina qué tan bien recupera los datos perdidos cuando el error del usuario causa la pérdida de datos.

Una falla de medios es la falla de una lectura o escritura de un archivo de disco requerido para ejecutar la base de datos, debido a un problema físico con el disco, como un bloqueo de la cabeza. Cualquier archivo de una base de datos puede ser vulnerable a una falla de medios.

La técnica de recuperación adecuada después de una falla de medios depende de los archivos afectados y de los tipos de respaldo disponibles.

# Grupo #6: Rodolfo Marciscano, Julio Chami y Ariel Moreno

**Desarrollo de Base de Datos en la Web y el Control de Acceso.**

En un desarrollo web, las bases de datos nos permiten actuar como almacén de datos y extraer el contenido de la web y disponerlo fácilmente, tal y como el desarrollador requiere en su proyecto web o el internauta necesita visualizarlo.

Aunque mucha gente puede que no sea consciente, la base de datos es na parte fundamental del rendimiento de una web, además de otros factores.

Por este motivo, es muy importante que hagamos un diseño inicial de la misma óptimo y orientado al desarrollo web que tenemos pensado implementar.

En la fase inicial de diseño de la base de datos, es necesario determinar la finalidad de la misma, organizar y categorizar la información a contener según las claves, definir las relaciones entre tablas y elementos que conformarán la misma y normalizarla de cara a facilitar la integración de la base de datos con otros sistemas.

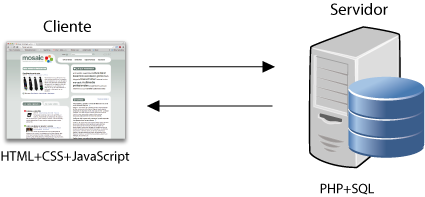
Así mismo, es fundamental dotar a la base de datos de un hardware adecuadamente dimensionado a las necesidades de consulta que va a tener dicha base de datos.

En una aplicación web, como un gestor de contenidos o una tienda en línea o cualquier otra web que haga uso de bases de datos, podemos encontrar hasta cuatro tipos de desarrollo:

Desarrollo de HTML + CSS

Desarrollo JavaScript

Desarrollo en PHP

Consultas SQL

Esquema de ejecución de las diferentes partes de una aplicación web

De las cuatro, las dos primeras se ejecutan en el navegador y las dos segundas, en el servidor, aproximadamente de la siguiente manera:

En el servidor, un archivo .php contiene el programa PHP con las instrucciones SQL para acceder a los datos, con el código HTML y CSS necesario para poderlos visualizar correctamente y con el JavaScript necesario para interactuar con el usuario.

El servidor ejecuta las instrucciones PHP y las consultas SQL, obtiene los datos y envía al cliente un archivo con el HTML, el CSS, el JavaScript y los datos obtenidos.

El navegador presenta los datos al usuario y ejecuta las instrucciones JavaScript.

El navegador envía al servidor los datos que proporciona el usuario.

El servidor recibe esa información y la guarda en la base de datos.

En la práctica, PHP, SQL y HTML suelen ir en un archivo, mientras que el CSS y el JavaScript van en archivos aparte, con lo que el navegador recibe un archivo HTML con los datos, un archivo CSS con el diseño y un archivo JavaScript con el programa que se va a ejecutar.

En general, las diferentes tecnologías que podemos encontrar en una aplicación web requieren de varias personas con diferentes competencias. Volviendo a la base de datos, su diseño no es una tarea trivial. Dependiendo de lo bien o mal hecha que esté, el número de transacciones (consultas o modificaciones) que deberán hacerse puede variar de modo considerable. En aplicaciones sencillas o para sitios con poco movimiento, la importancia puede ser relativa, pero en aplicaciones complejas y, en especial, en sitios con muchas visitas, reducir el número de consultas a la base de datos puede reducir considerablemente el coste en hardware del sitio. Y es que, en sitios web con muchas visitas, la importancia de la base de datos es tal que suele situarse en un servidor aparte del servidor de la aplicación.

# Bibliografías

**Grupo #1:**

[1] J. Sánchez. (s.f.). *Administración de usuarios en Oracle Database.* JorgeSanchez.net.Sitio web: https://jorgesanchez.net/manuales/abd/control-usuarios-oracle.html

[2] Informe ejecutivo de Oracle. (junio 2008). *Introducción a Oracle Identity Management.* Oracle. Sitio web: https://www.oracle.com/technetwork/es/documentation/317540-esa.pdf

[3] Oracle Corporation. (2010). *Desbloqueo de cuentas de usuario.* Oracle. Sitio web: https://docs.oracle.com/cd/E19957-01/821-0062/byaee/index.html

[4] Informaticadas. (febrero 2014). *Usuarios bloqueados en Oracle.* Sitio web: http://informaticadasdecadadia.blogspot.com/2014/02/usuarios-bloqueados-en-oracle.html

[5] Oracle Help Center. (s.f.). *Locking and Unlocking User Accounts.* Oracle. Sitio web: https://docs.oracle.com/database/121/ADMQS/GUID-6A8D4A59-6DB2-4662-BA4C-05B914D31B4F.htm#ADMQS12042

[6] Oracle. (s.f.). *Oracle9i Default Accounts and Passwords.* Sitio web: https://docs.oracle.com/cd/A97630\_01/win.920/a95490/username.htm

[7] Oracle Help Center. (s.f.). *Oracle Database System Privileges Accounts and Passwords.* Oracle. Sitio web: https://docs.oracle.com/en/database/oracle/oracle-database/12.2/hpdbi/oracle-database-system-privileges-accounts-and-passwords.html#GUID-7513171C-1055-48BB-8C79-B27EECC9B7E9

[8] Oracle Help Center. (s.f.). *Database 2 Day + Security Guide.* Oracle. Sitio web: https://docs.oracle.com/cd/E11882\_01/server.112/e10575/tdpsg\_user\_accounts.htm#TDPSG20000

[9] Admin. (diciembre, 2010). Administración de Usuarios en Oracle. Sitio web: https://orasite.com/tutoriales/administracion/administracion-de-usuarios-en-oracle

[10] Admin. (marzo, 2014). ORA-28001: la contraseña ha expirado. Sitio web: https://orasite.com/tutoriales/administracion/contrasena-expirado-expired-password

[11] Systemadmin. (octubre, 2013). Desbloquear usuario en ORACLE. Sitio web: http://systemadmin.es/2013/10/desbloquear-usuario-en-oracle

**Grupo #2:**

1. Recuperado el 5 de noviembre de 2020, Guía de administración de dominios de E/S de Oracle: <https://docs.oracle.com/cd/E58626_01/E69681/html/E62192/z40015671104978.html>
2. Recuperado el 5 de noviembre de 2020, Administración de usuarios en Oracle Database <https://jorgesanchez.net/manuales/abd/control-usuarios-oracle.html#h26>
3. ORACLE: Seguridad - Mayo de 1998. Jesús Vegas – recuperado el 5 de noviembre de 2020 <https://www.infor.uva.es/~jvegas/cursos/bd/oraseg/oraseg.html>
4. Video: SQL Tutorial - How to grant System and Object Privileges in Oracle Database <https://youtu.be/f4LJg4qEp40>

**Grupo #3:**

1. Docs.microsoft.com. 2020. *Límites De Recursos Para Los Servidores Lógicos En Azure - Azure SQL Database*. [online] Available at: <https://docs.microsoft.com/es-es/azure/azure-sql/database/resource-limits-logical-server#what-happens-when-database-resource-limits-are-reached> [Accessed 10 November 2020].
2. Docs.oracle.com. 2020. *Tablespaces, Datafiles, And Control Files*. [online] Available at: <https://docs.oracle.com/cd/B19306\_01/server.102/b14220/physical.htm> [Accessed 10 November 2020].
3. Oracle., C., 2020. *Como Crear Profile De Usuario En Base De Datos Oracle. - Oracleros.Com*. [online] oracleros.com. Available at: <https://oracleros.com/como-crear-profile-de-usuario-en-base-de-datos-oracle/> [Accessed 10 November 2020].
4. Gómez Mendoza, Angélica (2004). Bitacora. Consultado el 18 de noviembre del 2013. Recuperado de: <http://www.estructurayprogramacion.com/>
5. Serios Gómez, Rodrigo (2008). Ciencias y tecnologias. Consultado el 17 de noviembre de 2013. Disponible en: <http://www.jorgesanchez.net/bd/abd/abd1.pdf>

**Grupo #4:**

1. Acuña, C. C. (14 de Noviembre de 2020). *Blogspot.com.* Obtenido de Administracion de Base de Datos: http://carloscruza.blogspot.com/2009/10/privilegios-roles-y-perfiles-en-oracle.html
2. Gamiño, H. E. (10 de Noviembre de 2020). *Desempeño en Oracle.* Obtenido de http://hugoracle.blogspot.com/2008/09/roles-en-oracle.html
3. OLVERA, O. O. (10 de Noviembre de 2020). *Orlando Olguin.* Obtenido de Worpress: https://orlandoolguin.wordpress.com/2009/09/20/manejo-de-perfiles/
4. Orasite. (15 de Noviembre de 2020). *Administración de usuarios en Oracle.* Obtenido de https://orasite.com/tutoriales/administracion/administracion-de-usuarios-en-oracle
5. Parada, V. (12 de 11 de 2020). *Parada.* Obtenido de http://www.parada.cl/memoria/doc\_3\_4.html#:~:text=3.4%20Acceso%20a%20Bases%20de%20Datos&text=Una%20nueva%20forma%20de%20accesar,habitual%20para%20acceder%20ese%20servicio.
6. Sanchez, J. (13 de 11 de 2020). Obtenido de JORGESANCHEZ.NET: https://jorgesanchez.net/manuales/abd/control-usuarios-oracle.html
7. TECNOSeguro. (13 de 11 de 2020). *TECNOSeguro.* Obtenido de https://www.tecnoseguro.com/faqs/control-de-acceso/que-es-un-control-de-acceso#:~:text=La%20definici%C3%B3n%20m%C3%A1s%20generalizada%20de,formas%20y%20para%20diversas%20aplicaciones.
8. Vegas, J. (14 de Noviembre de 2020). *ORACLE: Seguridad.* Obtenido de https://www.infor.uva.es/~jvegas/cursos/bd/oraseg/oraseg.html#1.3
9. Yar. (12 de Noviembre de 2020). *Firmware.* Obtenido de http://comunidad.fware.pro/general/usuario-perfil-y-rol-en-oracle/

**Grupo #5:**

1. *Exporting and Importing Database Dumps*. Docs.plesk.com. (2020). Retrieved 15 November 2020, from <https://docs.plesk.com/en-US/onyx/customer-guide/website-databases/exporting-and-importing-database-dumps.69538/#:~:text=In%20Plesk%2C%20to%20export%20a,file%20to%20a%20destination%20database>.
2. *Respaldo y recuperación de base de datos*. Serman.com. (2020). Retrieved 15 November 2020, from <https://serman.com/blog-recuperacion-datos/respaldo-y-recuperacion-de-base-de-datos/>.
3. *Cómo importar y exportar bases de datos vía phpMyAdmin*. Pickaweb. (2020). Retrieved 15 November 2020, from <https://www.pickaweb.es/ayuda/como-importar-y-exportar-bases-de-datos-via-phpmyadmin/#:~:text=Importar%20una%20base%20de%20datos,de%20una%20base%20de%20datos>.
4. Tools, T. (2020). *Exportar e Importar Base de Dato de ORACLE y SQL DEVELOPER* [Video].

**Grupo #6:**

1. Carlos Casado Martinez  
   Universidad Oberta de Catalunya  
   <http://multimedia.uoc.edu/blogs/fem/es/las-aplicaciones-web-y-las-bases-de-datos/>
2. Carlos Yañez  
   Centro de Formación Profesional a Distancia de España (CEAC)  
   https://www.ceac.es/blog/la-gestion-de-las-bases-de-datos-en-el-desarrollo-web