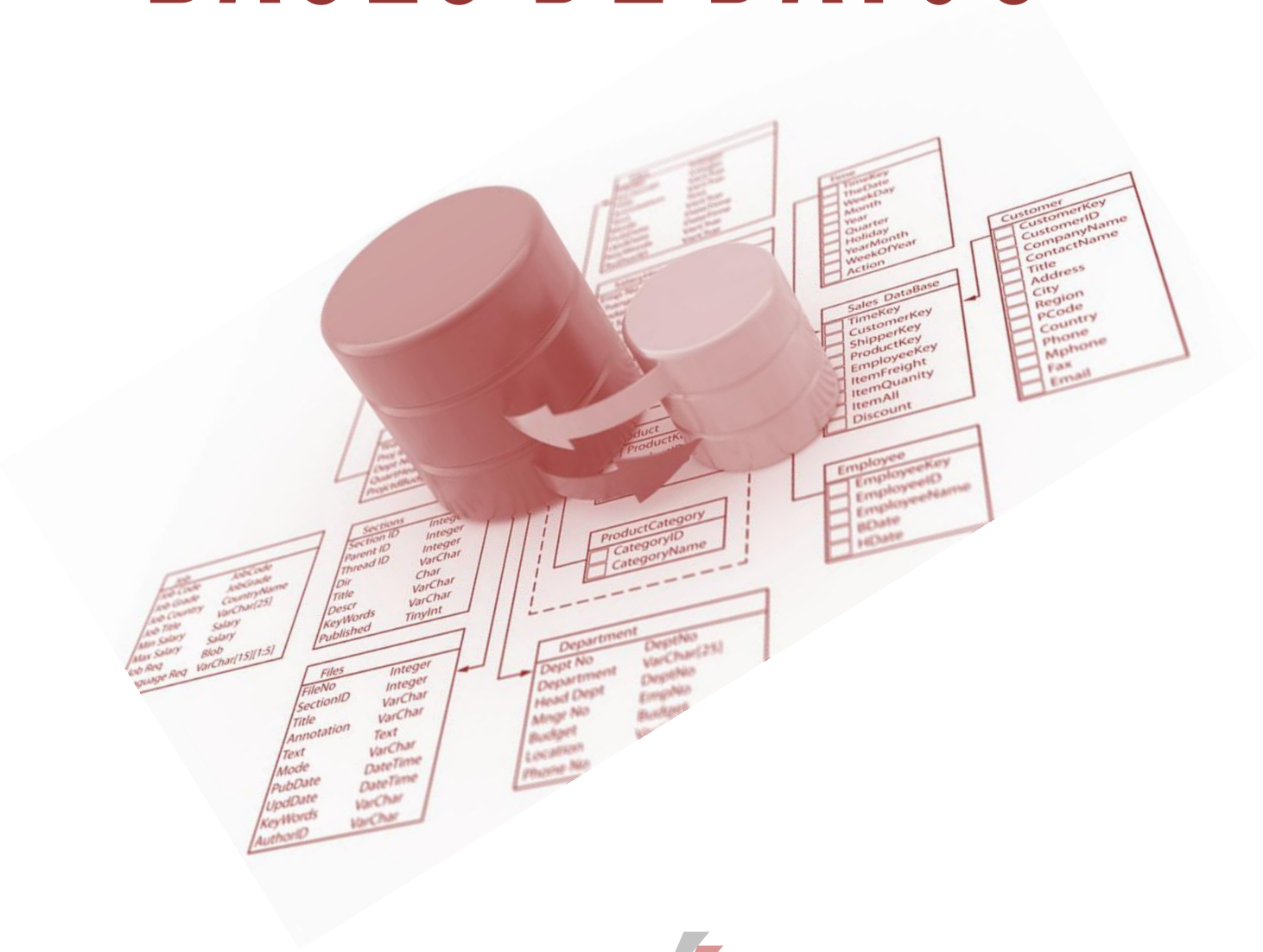
BASES DE DATOS



GESTIÓN DE SEGURIDAD

1 Introducción

La seguridad en la Base de Datos se refiere fundamentalmente a autorizar y desautorizar acciones de los usuarios sobre los objetos de la base de datos.

Los controles de acceso regulan el acceso de los usuarios a los objetos de la base de datos a través de la asignación de privilegios.

Un **privilegio** es un permiso para acceder a un objeto de una manera determinada.

Un **usuario** es un nombre definido en la base de datos que se puede conectar y acceder a los objetos.

Un **schema** es un conjunto nominado de objetos asociados con un usuario particular.

El **esquema** de una base de datos (en inglés, database schema) describe la estructura de una base de datos. En una base de datos relacional, el esquema define sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla.

El esquema es generalmente almacenado en un diccionario de datos. Aunque generalmente el esquema es definido en un lenguaje de base de datos, el término se usa a menudo para referirse a una representación gráfica de la estructura de base de datos.

En MySQL, físicamente, un esquema es sinónimo de una base de datos. Algunos productos de base de datos hacer una distinción. Por ejemplo, en la base de datos de Oracle el producto, un esquema representa sólo una parte de una base de datos: las tablas y otros objetos de propiedad de un único usuario.

Cuando se crea un usuario se crea el schema correspondiente con el mismo nombre. Una vez que un usuario se conecta con la base de datos, por omisión tiene acceso a todos los objetos de su schema.

Cuando se crea un nuevo usuario o se cambia alguno existente, el administrador de seguridad debe tomar una serie de decisiones relacionadas con la seguridad del usuario y su dominio, como por ejemplo:

- Si la autenticación del usuario va a ser realizada por la base de datos, el sistema operativo, o el servicio de autenticación de la red.
- Asignar los valores default del usuario y sus tablespaces.
- Una lista, si existe, de tablespaces accesibles por el usuario, y las cuotas (límites) que el usuario tiene para cada tablespace.
- El perfil (profile) de recursos del sistema que el usuario tiene disponible.
- Los privilegios y roles que el usuario posee para acceder a los objetos que de la base de datos.

2 Tipos de Usuarios

Muchas personas participan en el diseño, uso y mantenimiento de una base de datos grande.

- **Administradores**: Si consideramos a la base de datos y al SGBD como recursos del sistema de base de datos, debemos considerar a una persona que administre dichos recursos. El Administrador de la base de datos (DBA, en inglés) es quién se encarga de autorizar el acceso a la base de datos, de coordinar y vigilar su empleo, y de adquirir los recursos necesarios de software y hardware. El DBA es la persona responsable cuando surgen problemas como violaciones a la seguridad o una respuesta lenta del sistema.
- **Diseñadores de bases de datos**: se encargan de identificar los datos que se almacenarán en la base de datos y de elegir las estructuras apropiadas para representar y almacenar dichos datos.
- **Usuarios finales**: son las personas que necesitan tener acceso a la base de datos para consultarla, actualizarla y generar informes; la base de datos existe primordialmente para que ellos la usen.
- **Analistas y Programadores de Aplicaciones**: los analistas determinan los requerimientos de los usuarios finales y desarrollan especificaciones para transacciones programadas que satisfagan dichos requerimientos. Los programadores implementan estas especificaciones en forma de programas y luego prueban, depuran, documentan y mantienen esas transacciones programadas.

3 Creación de Usuarios

El comando **CREATE USER** nos va a permitir crear usuarios y asignarles una contraseña con el parámetro **IDENTIFIED BY**. Con el comando **GRANT** le asignaremos posteriormente los privilegios.

CREATE USER user [IDENTIFIED BY [PASSWORD] password]

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

Sin embargo, antes de crear un usuario debemos de ver las especificaciones para cada SGBD, pues hay variaciones de uno a otro. Por ejemplo:

MySQL

mysql> CREATE USER juan IDENTIFIED BY juanpassword;

ORACLE

SQL > CREATE USER alba IDENTIFIED BY albapassword;

SQL SERVER

CREATE USER [YourDomainYourDbUser] FROM LOGIN [YourDomainYourUser]
En este caso, el LOGIN deberá haber sido creado previamente.

PostgreSQL

CREATE USER alberto PASSWORD albertopassword;

4 Privilegios y Roles

Un privilegio es un derecho para ejecutar un tipo particular de sentencia SQL ó para acceder a un objeto de otro usuario.

Un usuario puede recibir los privilegios de dos maneras:

- Explícitamente.
- Se asignan privilegios a un rol (un conjunto nominado de privilegios) y luego se asignan estos roles a uno o más usuarios.

El objetivo de los roles es permitir una mejor administración de los privilegios, por lo general, se deberían garantizar privilegios a los roles y no a los usuarios individuales.

Los modos disponibles para asignar o desasignar privilegios y roles son:

- Emplear las utilidades de las distintas herramientas para la gestión de bases de datos.
- Los comandos SQL **GRANT** y **REVOKE**.

Los usuarios con la opción de ADMIN OPTION o con GRANT ANY PRIVILEGE pueden asignar o quitar privilegios del sistema a otros usuarios.

Cualquier usuario con el privilegio del sistema GRANT ANY ROLE puede asignar o anular cualquier rol a los usuarios de la base de datos. Cualquier usuario con la opción ADMIN OPTION también puede asignar o anular roles a los usuarios.

5 Privilegios

Un privilegio es la capacidad de un usuario dentro de la base de datos a realizar determinadas operaciones o acceder a determinados objetos de otros usuarios.

Pueden ser de varios tipos:

- **Privilegios del sistema**

Un privilegio del sistema es el derecho a realizar una acción particular, o realizar una acción sobre cualquier objeto de un tipo particular.

Existen alrededor de 60 privilegios del sistema distintos.

- **Privilegios sobre los objetos del schema**

Un privilegio sobre un objeto del schema es un derecho para efectuar una acción particular sobre una tabla, vista, secuencia, procedimiento, función o paquete.

Algunos otros objetos como clusters, índices, triggers y vínculos de base de datos (database links) no se asocian a privilegios sobre los objetos, sino que son controlados por medio de privilegios del sistema.

Un objeto del schema y su sinónimo son equivalentes en cuanto a los privilegios.

Un usuario posee automáticamente todos los privilegios sobre los objetos que se encuentran en su schema.

- Privilegios de objetos sobre tablas]

Los privilegios de objetos sobre tablas pueden asignarse al nivel de las operaciones de DML o DDL que se quieran hacer.

Para las operaciones de DML se pueden asignar los privilegios de DELETE, INSERT, SELECT y UPDATE. Los privilegios de INSERT y UPDATE se pueden restringir inclusive a nivel de columna de una tabla.

Cuando se inserta una fila con valores solo para las columnas asignadas las otras columnas reciben el valor NULL o los valores default definidos para esa columna.

Para las operaciones del DDL se pueden asignar los privilegios de ALTER, INDEX y REFERENCES.

El privilegio REFERENCES habilita a quien se le asigna este privilegio que pueda usar la tabla (sus columnas) como clave primaria de cualquier clave foránea que necesite definir sobre sus propias tablas.

- Privilegios de objetos sobre vistas

Los privilegios de objetos sobre vistas permiten a los usuarios llevar a cabo varias operaciones DML que afectan a las tablas bases de las cuales las vistas son derivadas.

Los privilegios de objetos sobre tablas se pueden aplicar de manera similar sobre las vistas.

Para crear una vista se debe haber garantizado los siguientes privilegios:

- CREATE VIEW o CREATE ANY VIEW (privilegio de sistema) directamente o a través de un rol.
- Explícitamente los privilegios de objeto SELECT, INSERT, UPDATE o DELETE sobre las tablas base que corresponden a las vistas, o los privilegios SELECT ANY TABLE, INSERT ANY TABLE, UPDATE ANY TABLE ó DELETE ANY TABLE (privilegios del sistema).
- Para poder garantizar a otros usuarios el acceso a las vistas se deben tener el privilegio sobre los objetos base con GRANT OPTION o el privilegio del sistema ADMIN OPTION.

Para usar una vista se necesita tener privilegios sobre la vista únicamente. No se requiere tener privilegios sobre los objetos base que soportan a las vistas.

Las vistas agregan dos niveles adicionales de seguridad sobre las tablas base: a nivel de columna y a nivel de valores de la tabla: una vista puede proveer el acceso sólo a algunas columnas de la tabla, o a determinadas filas, filtradas por determinados valores en cada columna.

- Privilegios de objeto sobre procedimientos

El único privilegio de objetos para procedimientos (incluyendo funciones y paquetes) es el de EXECUTE. Con esto se puede agregar otro nivel de seguridad a la base de datos, ya que un usuario solo requiere el privilegio de ejecutar un procedimiento, pero no necesita los privilegios de acceder a las bases que el

procedimiento accede. De esta manera se fuerza que los usuarios se accedan a las tablas base sólo a través los procedimientos, cuyo funcionamiento está debidamente probado.

5.1 Privilegios sobre los objetos en Oracle SQL

Nos permiten acceder y realizar cambios en los datos de otros usuarios. Por ejemplo: el privilegio de consultar la tabla de otro usuario es un privilegio sobre objetos.

```
GRANT {PRIV_OBJETO [, PRIV_OBJETO]… | ALL [PRIVILEGES]}
 [(COL [,COL]…)]
ON [USUARIO] OBJETO
TO {USUARIO | ROL | PUBLIC} [,{USUARIO | ROL | PUBLIC}…]
[WITH GRANT OPTION];
```

ON= Especifica el objeto sobre el que se dan los privilegios.

TO= Identifica a los usuarios o roles a los que se conceden los privilegios.

ALL= Concede todos los privilegios sobre el objeto especificado.

WITCH GRANT OPTION= Permite que el receptor del privilegio o rol se lo asigne a otros usuarios o roles.

PUBLIC= Asigna los privilegios a todos los usuarios actuales y futuros: El propósito principal del grupo PUBLIC es garantizar el acceso a determinados objetos a todos los usuarios de la base de datos.

Por ejemplo:

SQL > GRANT ALL ON tabla_alumnos TO david

Siendo tabla_alumnos una tabla de nuestra base de datos y david un usuario de esta, hemos asignado mediante GRANT ALL, todos los permisos al usuario david sobre esta tabla.

Si queremos asignar sólo uno de estos permisos utilizamos la misma sentencia pero con el permiso que queramos otorgar.

SQL > GRANT SELECT ON tabla_alumnos TO david

SQL > GRANT SELECT, INSERT ON tabla_alumnos TO david

5.2 Privilegios de sistema en Oracle SQL

Dan derecho a ejecutar un tipo de comando SQL o a realzar alguna acción sobre objetos de un tipo especificado. Por ejemplo, el privilegio para crear TABLESPACES es un privilegio de sistema.

```
GRANT {PRIVILEGIO | ROL} [, {PRIVILEGIO | ROL}, …]
TO {USUARIO | ROL | PUBLIC} [,{USUARIO | ROL | PUBLIC}]
[WITH ADMIN OPTION];
```

WITH ADMIN OPTION= Permite que el receptor del privilegio o rol pueda conceder esos mismos privilegios a otros usuarios o roles.

5.3 Retirada de privilegios de objetos a los usuarios

Si queremos quitar un privilegio a uno de estos objetos haremos lo mismo que con GRANT pero utilizando la sentencia **REVOKE**.

REVOKE {PRIV_OBJETO [,PRIV_OBJETO]... | ALL [PRIVILEGES]}
ON [USUARIO.]OBJETO
FROM {USUARIO | ROL | PUBLIC} [, {USUARIO | ROL | PUBLIC}]...;

Por ejemplo:
SQL > REVOKE ALL ON tabla_usuarios FROM david

Retirada de privilegios de sistema o roles a los usuarios

REVOKE {PRIV_SISTEMA | ROL} [,{PRIV_SISTEMA | ROL}]...
FROM {USUARIO | ROL | PUBLIC} [,{USUARIO | ROL | PUBLIC}]...;

6 Roles

En general se emplean para asignar los privilegios relacionados con los usuarios finales de las aplicaciones de un sistema o para asignar roles a otros roles.

Los roles de la base de datos tienen la siguiente funcionalidad:

- Un rol puede tener privilegios del sistema y privilegios de objetos del schema.
- Un rol se puede asignar a otros roles (excepto a sí mismo directa o indirectamente).
- A cualquier usuario de la base de datos se le puede asignar cualquier rol.
- Un rol asignado a un usuario se puede habilitar o inhabilitar en cualquier momento.
- Un rol garantizado indirectamente puede ser explícitamente habilitado o inhabilitado al usuario.

En una base de datos cada rol debe ser único.

6.1 Roles en Oracle SQL

Son un conjunto de privilegios agrupados que se crean con el formato:

CREATE ROLE NOMBREROL [IFENTIFIED BY CONTRASEÑA];

NOTA: Un rol puede decidir el acceso de un usuario a un objeto, pero no puede permitir la creación de objetos.

Supresión de privilegios en los roles

REVOKE NOMBREPRIVILEGIO ON NOMBRETABLA FROM NOMBREROL;

REVOKE NOMBREPRIVILEGIO FROM NOMBREROL;

Supresión de un rol

DROP ROLE NOMBREROL;

Establecer un rol por defecto

ALTER USER NOMBREUSUARIO
DEFAULT {[ROLE NOMBRE_ROL] | [NONE]};

NONE= Hace que el usuario no tenga rol por defecto.

7 Gestión de tablespaces

Una de las tareas habituales en la administración de una base de datos Oracle es la de crear un nuevo tablespace para contener nuevos objetos como tablas, índices, etc.

Un **tablespace** es una unidad lógica de almacenamiento de datos representada físicamente por uno o más archivos de datos. Se recomienda no mezclar datos de diferentes aplicaciones en un mismo tablespace.

7.1 Crear un tablespace

CREATE TABLESPACE NOMBRETABLESPACE
DATAFILE 'NOMBREARCHIVO' [SIZE ENTERO [K | M] [REUSE]
[DEFAULT STORAGE
(INITIAL TAMAÑO
MINEXTENTS TAMAÑO
MAXEXTENTS TAMAÑO
PCTINCREASE VALOR
)]
[ONLINE | OFFLINE];

REUSE= Reutiliza el archivo si ya existe o lo crea si no existe.

DEFAULT STORAGE= Define el almacenamiento por omisión para todos los objetos que se creen en este espacio de la tabla. Fija la cantidad de espacio si no se especifica en la sentencia CREATE TABLE.

7.2 Modificación de tablespaces

```
ALTER TABLESPACE NOMBRETABLESPACE
{[ADD DATAFILE 'NOMBREARCHIVO' [SIZE ENTERO [K | M] [REUSE]
 [AUTOEXTEND ON… | OFF]
 ]
 [RENAME DATAFILE 'ARCHIVO' [, 'ARCHIVO']…
 TO 'ARCHIVO' [, 'ARCHIVO']]
 [DEFAULT STORAGE CLAUSULAS_ALMACENAMIENTO]
 [ONLINE | OFFLINE]
 };
```

ADD_DATAFILE= Añade al tablespace uno o varios archivos.

AUTOEXTEND= Activa o desactiva el crecimiento automático de los archivos de datos del tablespace. Cuando un tablespace se llena podemos usar esta opción para que el tamaño del archivo o archivos de datos asociados crezca automáticamente.

AUTOEXTEND OFF: desactiva el crecimiento automático.

RENAME_DATAFILE= Cambia el nombre de un archivo existente del tablespace. Este cambio se tiene que hacer desde el sistema operativo y, después, ejecutar la orden SQL.

7.3 Borrado de tablespaces

```
DROP TABLESPACE NOMBRETABLESPACE
[INCLUDING CONTENTS];
```

INCLUDING CONTENTS= Permite borrar un tablespace que tenga datos. Sin esta opción solo se puede suprimir un tablespace vacío.

Se recomienda poner el tablespace offline antes de borrarlo para asegurarse de que no haya sentencias SQL que estén accediendo a datos del tablesapace, en cuyo caso no sería posible borrarlo.

Cuando se borra un tablespace los archivos asociados no se borran del sistema operativo, por lo que tendremos que borrarlos de forma manual.

Ejemplo

Se creará un nuevo tablespace para ubicar los objetos de una nueva aplicación llamada **CONTA**. Antes de crearlo tenemos que realizar un análisis de los requerimientos de espacio y ubicación. Por ejemplo: el tablespace se debe llamar CONTA_01, vamos a necesitar 300MB para nuestro nuevo tablespace y el datafile lo vamos a ubicar en /u03/oradata/PRUEBA001/dat, tenemos que verificar que en filesystem /u03 tenemos ese espacio libre.

La instrucción para crear un tablespace es CREATE TABLESPACE Nombretablespace seguido de una serie de opciones, en este ejemplo vamos a ver la forma más habitual de crear un tablespace.

```
CREATE TABLESPACE CONTA
DATAFILE '/u03/oradata/PRUEBA001/dat/CONTA_01.dbf' SIZE 300M
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;
```

Nuestra instrucción CREATE TABLESPACE creará el tablespace CONTA con el datafile CONTA_10.dbf de 300MB y dejamos que Oracle se encargue de gestionar automáticamente los extents de los objetos que se creen en el tablespace.

Fuentes

http://www.desarrolloweb.com/
http://cursoanpefernandoortiz.blogspot.com.es/
https://es.wikibooks.org/
Y otros.

Índice de contenidos