



# Unidad 11

**CREACIÓN Y MANIPULACIÓN DE OTROS  
OBJETOS DE LA BASE DE DATOS: VISTAS,  
SINÓNIMOS, USUARIOS,  
ROLES, PERFILES, ÍNDICES Y SECUENCIAS**



Bases de datos

2011-12



# 1. INTRODUCCIÓN

- Un SGBDR debería ser aquél que implementase, con total fidelidad el modelo relacional, pero, en la actualidad, no existe ningún SGBD con esta característica.
- Aunque deseable que un sistema relacional incorpore en su diseño todos los aspectos del modelo relacional, hay que tener en cuenta que no todos ellos tienen la misma importancia.





# 1. INTRODUCCIÓN

- En 1982, E. F Codd publicó un artículo en el que establecía que para que un SGBD pueda considerar relacional, debe poseer las dos características siguientes:
  - El usuario debe percibir las bases de datos como tablas y nada más que como tablas.
  - El SGBD debe manejar las operaciones de selección, proyección y reunión natural, sin requerir definiciones previas de rutas de acceso físico es decir debe haber independencia entre la estructura lógica de los datos y su almacenamiento interno.





# 1. INTRODUCCIÓN

- En 1985 E. R Codd establece, en un artículo publicado en *Computerworld*, doce reglas para determinar el grado en que un sistema es relacional.





# 1. INTRODUCCIÓN

- La revisión y ampliación por Codd tanto de estas doce reglas, como de las ideas técnicas en que se basan, culminan con la publicación en 1990 de la versión 2 del modelo relacional (*Relational Model Version 2 o RMIV2*).





## 12 Reglas de Codd

### 1. La regla de información.

Toda la información se presentará en la BD sólo como valores en posiciones de columna dentro de filas de las tablas.

### 2. La regla de acceso garantizado.

Debe ser posible obtener la dirección de cada valor escalar individual en la BD, mediante la especificación del nombre de la tabla que lo contiene, el nombre de la columna en que se encuentra y el valor de la clave primaria de la fila donde está.

### 3. El manejo sistemático de los valores nulos.

El SGBDR ha de soportar valores nulos para representar la ausencia de información de forma automática e independiente del tipo de dato.





## 12 Reglas de Codd

### **4. Catálogo activo on-line basado en el modelo relacional.**

La descripción de la BD, diccionario de datos, se representa a nivel lógico como datos ordinarios, de forma que un usuario autorizado pueda utilizar el mismo lenguaje relacional para manejar sus datos y consultar el catálogo.





## 12 Reglas de Codd

### **5. La regla del sublenguaje de datos completo.**

El sistema que debe manejar, por lo menos, un lenguaje relacional con las siguientes características:

- a) Que tenga una sintaxis lineal.
- b) Que se pueda usar tanto de forma interactiva como huésped.
- c) Que maneje operaciones de definición de datos (incluyendo definición de vistas), de manipulación de datos, restricciones de seguridad e integridad y operaciones de gestión de transacciones.







## 12 Reglas de Codd

### **6. La regla de actualización de vistas.**

El sistema debe poder actualizar las vistas actualizables en teoría.

### **7. Inserción, modificación y eliminación de alto nivel.**

El sistema debe manejar operadores de inserción, modificación y eliminación de todo un conjunto a la vez.

### **8. Independencia física de los datos.**

La forma en que se almacenen los datos o de los métodos de acceso, no debe influir en su manipulación lógica.





## 12 Reglas de Codd

### **9. Independencia lógica de los datos.**

El añadir, eliminar o modificar objetos de la BD no debe repercutir en los programas y/o usuarios que acceden a esos objetos.

### **10. Independencia de la integridad.**

Las restricciones de integridad deben especificarse por separado de los programas y almacenarse en el catálogo, no en los programas. Debe ser posible alterar esas restricciones sin afectar innecesariamente a las aplicaciones ya existentes.





## 12 Reglas de Codd

### **11. Independencia de la distribución.**

Las aplicaciones ya existentes deberán seguir funcionando sin problema cuando se introduzca una versión distribuida del SGBDR o cuando se redistribuyan los datos.

### **12. La regla de no subversión.**

Si el sistema ofrece una interfaz de bajo nivel (registro a registro), esa interfaz no podrá ser utilizada para subvertir el sistema saltándose, por ejemplo, una restricción, relacional de seguridad o integridad.





## 2. EL SGDB ORACLE

- La empresa Oracle cuenta con 30 años de historia.
- A mediados de los 70 Larry Ellison se dio cuenta que el modelo relacional todavía no se había llevado a la práctica, por lo que fundó la compañía Relational Systems con la idea de desarrollar y comercializar tecnología basada en el modelo relacional.
- En 1979 obtuvieron el sistema SQL RDMS, 1º producto que más se aproximaba a un SGBD fundamentado en el modelo relacional. Por aquel entonces adoptaron el nombre de ORACLE.
- En 1982 el SGBD de Oracle fue declarado el mejor producto del mercado

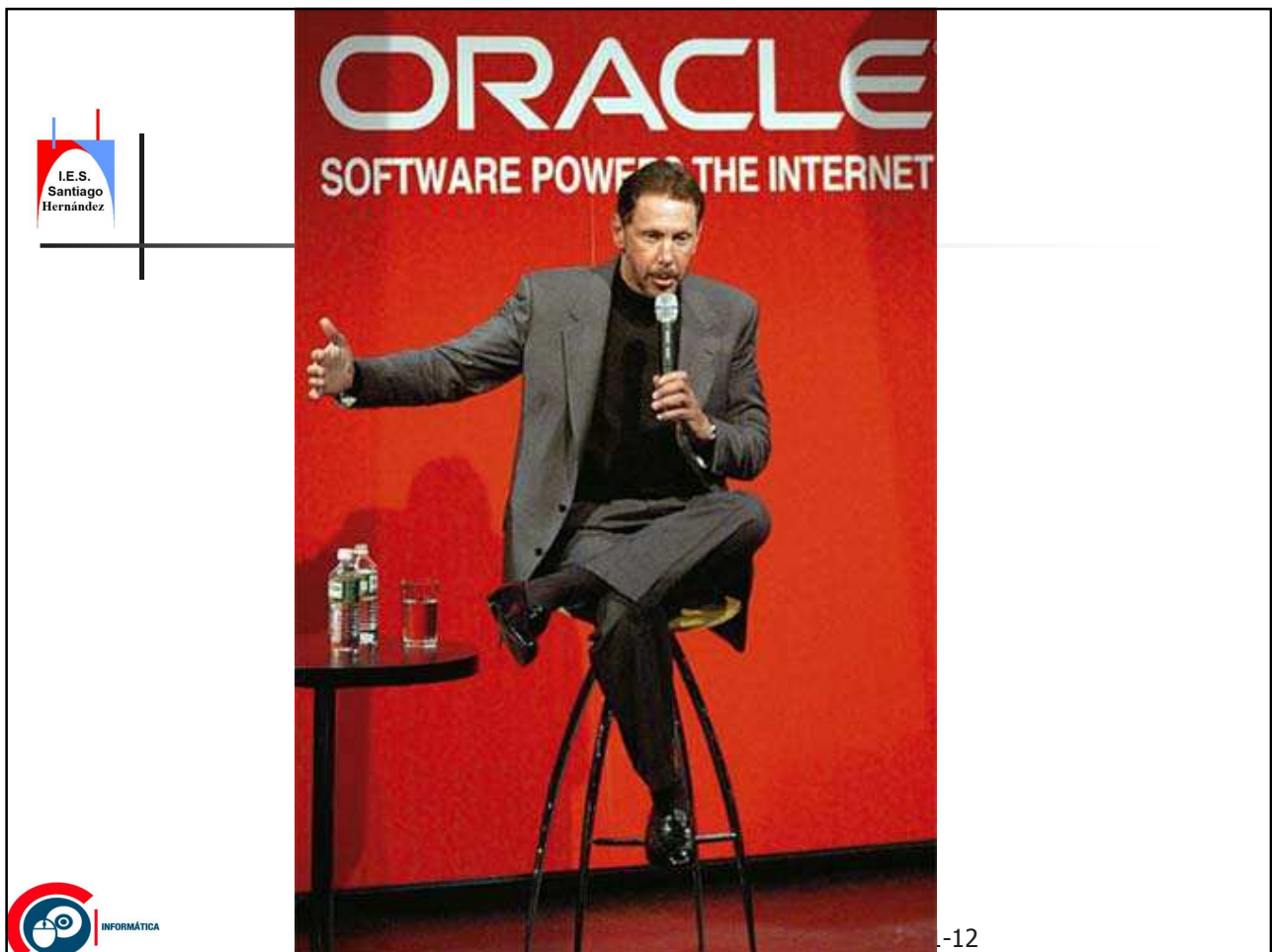




## 2. EL SGDB ORACLE

- En 1986 lanzó la primera BD cliente/servidor.,
- En 1988 introdujo un lenguaje de programación den sus SGBD PL/SQL
- En 1997 proyectó la 1ª BD que se fundamenta en tecnología Web.
- Actualmente se comercializa la versión 11g basada en tecnología Grid Computing







## ORACLE 11g

- La tarea de administración de una BD es bastante compleja por lo que se suele encomendar incluso a varias personas.
- El DBA debe ser un usuario muy experimentado, sus funciones son:

Instalar Oracle, diseñar y crear una BD, arrancar y detener la BD, crear y controlar usuarios, conceder privilegios, gestionar el espacio, hacer copias de seguridad y recuperar la BD.





## ¿Qué es Oracle 11g?

- Es la nueva suite de varios productos software de la compañía Oracle, basados todos en la tecnología **Grid Computing o computación grid**.
- **Grid** es una nueva arquitectura que permite utilizar de forma coordinada todo tipo de recursos (entre ellos cómputo, almacenamiento y aplicaciones específicas)
- En este sentido es una nueva forma de computación distribuida, en la cual los recursos pueden ser heterogéneos (diferentes arquitecturas, supercomputadores, clusters...) y se encuentran conectados mediante redes.
- El término *grid* se refiere a una infraestructura que permite la integración y el uso colectivo de ordenadores de alto rendimiento, redes y bases de datos







## 2. EL SGDB ORACLE

- En definitiva el sistema grid computing consiste en utilizar los ordenadores conectados a una red para repartir las tareas a realizar entre todos ellos, aprovechando sus tiempos muertos y haciendo uso por tanto de toda su capacidad de procesamiento.





## 2. EL SGDB ORACLE

### ■ Herramientas importantes:

- SQL\*Plus proporciona acceso a la BD en modo texto. Determinados comandos, como por ejemplo para la generación de informes son propios de esta herramienta y no funcionan sobre otras.
- iSQL\*Plus: similar a SQL\*Plus pero utilizando un entorno gráfico más amigable del tipo navegador web. Ya no está disponible en Oracle11g, hay que utilizar SQLdeveloper
- Asistente de configuración: permite mantener, eliminar y crear una BD, admite el uso de plantillas o la creación de BD basadas en otras anteriores.





## INSTANCIA ORACLE

- En Oracle se introduce un concepto nuevo de BD, la instancia.
- Una instancia es un conjunto de procesos y estructuras de memoria que Oracle usa para permitir el acceso a una determinada BD. Una instancia suele tener una BD asociada pero solo una.





## INSTANCIA ORACLE

- Las instancias asociadas a las BD de Oracle son en la versión 10g, servicios de Windows.
- Así, durante la instalación hemos creado una BD y se ha generado un servicio de Windows que puede iniciarse manual o automáticamente al arrancar Windows.
- **LISTENER:** proceso que está al tanto de los intentos de conexión que se hacen en Oracle con el fin de permitirlos o denegarlos, en Windows es un servicio que se arrancará manual o automáticamente. Así pues la 1ª condición para acceder a una BD oracle será comprobar que el listener está en funcionamiento. En Oracle 11g este servicio se aconseja manejarlo desde el EM. A veces será necesario reiniciarlo o actualizarlo. Por defecto el puerto del listener es el 1521















## INSTANCIA ORACLE

- El siguiente servicio correspondiente a la instancia que queremos arrancar será el servicio correspondiente a la BD que queremos activar. Recuerda que puedes tener varias BD.
- El nombre de este servicio es OracleServiceXXX siendo XXX el nombre de la BD



 Notificación de sucesos del sistema	Registra su...	Iniciado	Automático	Sistema local
 OracleDBConsoleoracle2		Iniciado	Automático	Sistema local
 OracleDBConsoleoracle3		Iniciado	Automático	Sistema local
 OracleJobSchedulerORACLE2			Deshabilitado	Sistema local
 OracleJobSchedulerORACLE3			Deshabilitado	Sistema local
 OracleOraDb10g_home2iSQL*Plus	iSQL*Plus ...	Iniciado	Automático	Sistema local
 OracleOraDb10g_home2TNSListener		Iniciado	Automático	Sistema local
 OracleServiceORACLE2		Iniciado	Automático	Sistema local
 OracleServiceORACLE3		Iniciado	Automático	Sistema local
 Plus and Plus	Habilita us...	Iniciado	Automático	Sistema local

Conviene valorar la posibilidad de arrancar dichos servicios manual o automáticamente dada la cantidad de recursos consume. Esto variará en función del uso que vaya darse al SGDB.



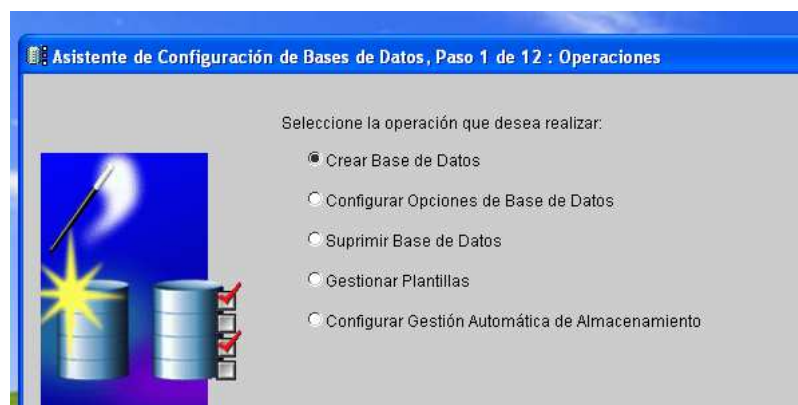
- Una vez arrancado el servidor, accederemos desde el cliente mediante SQL\*Plus, SQL developer, isqlplus,...





### 3. Creación y gestión de BD

- Oracle 10g dispone de un asistente que nos permitirá crear, suprimir o modificar BD o gestionar plantillas de BD







## Gestionar plantillas

- Mediante esta opción, se pueden confeccionar plantillas que después facilitará el proceso de creación de una BD.
- Una plantilla son el conjunto de todas las características que se fijan durante la creación de la BD.
- Podemos crear plantillas a partir de BD existentes, en este caso disponemos de dos opciones





- Incluir solo la estructura (tablespaces, ficheros físicos,... pero no los datos de usuario)
- Incluir los datos de usuario contenidos en los ficheros, con lo cual se crea una copia de la BD, tanto estructura como datos.





## 2. Arquitectura de ORACLE

- Un SGBDR debe ser capaz de manejar la información a varios niveles. Según la arquitectura ANSI, los datos deben poder ser estructurados a nivel externo, a nivel conceptual y a nivel interno. Los dos primeros niveles implican que el sistema debe ser capaz de manejar las estructuras que establece el modelo relacional.
- El tercer nivel, el interno, obliga a que el sistema maneje las estructuras adecuadas tanto para el almacenamiento real de los datos, como para lograr un rendimiento adecuado en las operaciones de manipulación de dichos datos.





## 2. Arquitectura de ORACLE

En los manuales de Oracle se dice que una BD tiene una **estructura física** y otra **lógica**. La estructura física está constituida por los ficheros del SO que dan soporte a los datos. Los *tablespaces* y los objetos de esquema configuran la estructura lógica.

Estructura de una base de datos Oracle		Niveles ANSI
Estructura lógica	Vistas	Externo
	Tablas	Conceptual
	Índices, Tablespaces, Segmentos, Bloques Oracle	Interno
Estructura física	Ficheros de datos Ficheros de redo log Ficheros de control Bloques del SO	





## 2.1 Componentes de la BD

- Los componentes de una BD son:
  - Archivos de datos.
  - Archivos de diario.
  - Archivos de control.





## Archivos de datos (database files): Tablespaces y registros de rehacer

- Contienen toda la información de la BD: datos de usuario y datos de sistema. Antes de introducir datos en la BD, es necesario crear un espacio para las tablas (*tablespaces*) y después crear las tablas, dentro de las cuales se introducirán los datos.
- Es conveniente tener un tablespaces para cada aplicación y uno para almacenar los datos de la aplicación de almacén, otro para los datos de la aplicación de nóminas.





## Tablespaces

- Cada *tablespace* consta de uno o más archivos en disco. Un archivo de datos sólo puede pertenecer a un único tablespace. Oracle dispone de cuatro que se crean al instalarlo:





## Tablespaces

- ❖ **SYSTEM.** Almacena toda la información que Oracle necesita para gestionarse el mismo como el diccionario de datos. Se almacena en el archivo SYSTEM01.DBF
- ❖ **USERS.** Contiene información personal de los usuarios. Es el lugar que nos deja el DBA para hacer pruebas, se almacena en el fichero USERS01.DBF
- ❖ **TEMP.** Almacena tablas temporales, para gestionar sus transacciones. TEMP01.DBF
- ❖ **UNDOTBS.** Donde se guarda la información de deshacer. Se utiliza para almacenar la imagen anterior de los datos antes de permitir actualizaciones, lo que permite recuperar los datos cuando no se completa una transacción. UNDOTBS01.DBF







## Registros de rehacer o redo\_log, el registro de transacciones

- Son archivos de datos donde Oracle registra todas las transacciones o modificaciones (INSERT, UPDATE Y DELETE) que se producen. Esto permite recuperar la BD si hay problemas.





## Registros de rehacer o redo\_log, el registro de transacciones

- Una BD tiene 2 o más ficheros de este tipo, siendo utilizados de forma cíclica, ie, el SGBD empieza a grabar en uno de ellos y cuando lo llena pasa al siguiente, luego al siguiente, y así sucesivamente. Estos ficheros son: RBDO01.DBF, REDO02.DBF, REDO03.DBF,...
- Un registro de Redo\_Log contiene: identificación de la transacción, dirección de bloque, número de fila, número de columna y valor anterior y nuevo del dato modificado.





## Archivos de control

- Un fichero de control contiene información sobre los ficheros asociados con una BD Oracle. Todas las modificaciones importantes que se hagan en la estructura de la BD se registran en el fichero de control. Estos ficheros de control mantienen la integridad de la BD. Se recomienda tener dos archivos de control por si uno se estropea. Los ficheros de control se llaman CONTROL01.CTL, CONTROL02.CTL y CONTROL03.CTL y contienen la siguiente información:
  - Información de arranque y parada
  - Nombre, fecha y hora de creación de la base de datos.
  - Nombres y localizaciones de los ficheros de datos y de redo log.
  - Información sobre puntos de verificación.(puntos de control que se dan cuando se llena el redo-log, cuando se detiene la BD)





### 3. Diccionario de datos

- Todo SGBD está construido alrededor de un diccionario de datos. En el caso de los sistemas relacionales, este diccionario es, en realidad, un diccionario/directorio gestionado por el propio SGBD, que recibe el nombre de catálogo.
- El catálogo no sólo sirve al SGBD, sino, también, a los usuarios, que lo pueden consultar utilizando el mismo lenguaje con el que consultan el resto de la base de datos.





### 3. Diccionario de datos

- Oracle implementa el catálogo mediante un conjunto de tablas y vistas. Las tablas almacenan la información sobre la base de datos y en ellas sólo escribe el SGBD.
- Los usuarios no suelen acceder directamente a ellas, porque la mayoría de sus datos están almacenados en un formato críptico.
- Las vistas permiten a los usuarios consultar la información en un formato más adecuado a sus necesidades. Los usuarios de una base de datos Oracle consultan el catálogo utilizando el lenguaje SQL.





### 3. Diccionario de datos

- Los datos del catálogo son necesarios para el funcionamiento del SGBD.
- Cuando se ejecuta una operación de manipulación de datos, el SGBD accederá a las tablas del catálogo para, por ejemplo, comprobar que el objeto existe y que el usuario tiene privilegios suficientes para realizar la operación.
- Durante la ejecución de una operación de definición de datos, el SGBD actualizará las tablas del catálogo para reflejar la existencia de un nuevo objeto o las modificaciones realizadas a un objeto ya existente.





### 3. Diccionario de datos

- Todas las tablas y vistas están definidas en el esquema del usuario **SYS**.
- Existen sinónimos públicos definidos sobre muchas de las vistas para facilitar el acceso a los usuarios. Cuando existe un sinónimo público sobre un objeto de esquema, éste puede ser referenciado por cualquier usuario a través de dicho sinónimo, sin necesidad de hacer referencia al esquema en el que está definido.
- El acceso a estas vistas, como a todos los objetos de la base de datos, está controlado por el sistema de privilegios o autorizaciones, por lo que los usuarios sólo podrán consultar aquellas vistas sobre las que tengan los privilegios adecuados.





## 4. Vistas

- Una vista es una tabla sin contenido, totalmente virtual, que devuelve las filas como resultado de ejecutar una consulta SQL.







## 4. Vistas

- La diferencia con una consulta ejecutada directamente es que, mientras cada sentencia SQL tiene que pasar por un proceso de compilación, la vista es una consulta cuya definición ha sido almacenada previamente y que ya ha sido compilada, siendo por tanto el tiempo de ejecución bastante menor





## 4. Vistas

- Además un usuario puede no tener acceso a la información de varias tablas pero sí tener acceso a la vista que consulta esas tablas, proporcionando así, un acceso controlado solo a determinadas filas y columnas de esas tablas.
- Además, se pueden crear vistas para que los usuarios no expertos puedan acceder de forma fácil a la información obtenida a través de una sentencia select compleja.





## Vistas: sintaxis

Para crear una vista:

```
CREATE [OR REPLACE] VIEW nombre_vista [lista_columnas] AS  
sentencia_select
```

Para eliminar una vista:

```
DROP VIEW nombre_vista
```





## Ejemplo

```
CREATE VIEW vistaPedidos (num, cliente, total) AS  
SELECT num, nombre_cli, SUM(cantidad*preciouni)  
FROM clientes c, pedidos p, detalles_pedidos d  
WHERE c.cod_cli=p.cod_cli and p.cod_cli=d.cod_cli  
GROUP BY num, nombre_cli;
```





## Ejemplo

- Cuando queramos hacer consultas sobre esta vista, utilizaremos la sintaxis habitual
- `SELECT * FROM vistaPedidos`





## Vistas del catálogo

- La mayoría de las vistas del catálogo aparecen en conjuntos de tres que contienen información similar y que se diferencian entre sí por el prefijo con el que se nombran.





## Vistas del catálogo

- Por ejemplo, las vistas USER-TABLES, ALL-TABLES y DBA- TABLES contienen toda información sobre tablas definidas en la base de datos. La diferencia básica entre ellas está en que si un usuario consulta la primera, obtiene información sobre las tablas definidas en su esquema, si consulta la segunda, la información obtenida correspondería a todas las tablas accesibles por él, es decir, a todas aquellas sobre las que tiene privilegios, y, al consultar la tercera, obtendría descripciones de todas las tablas de la base de datos.





## Tipos de vistas

- **USER\_**
- Con este tipo de vistas, se pueden ver los objetos que pertenecen al usuario con el que estamos conectados exclusivamente. Por ejemplo, en *user\_tables*, veremos las tablas del usuario actual







## Tipos de vistas

- **ALL\_**
- Usando este tipo de vistas, podremos ver los objetos a los que el usuario actual tiene acceso, tanto los propios como vía permisos sobre los mismos.





## Tipos de vistas

- **DBA\_**
- Con estas vistas, se pueden ver todos los objetos de la base de datos. Es decir, si tenemos la vista *dba\_tables*, servirá para ver todas las tablas que se encuentran contenidas en nuestra instancia. Para poder ver este tipo de vistas, se requiere tener privilegios suficientes.





## Vistas del catálogo

- En general, la diferencia entre los tres tipos de vistas es la que se deduce del ejemplo anterior, es decir, las vistas con prefijo USER devuelven información referente al esquema del usuario, las de prefijo ALL se refieren a los objetos a los que el usuario puede acceder y las DBA tratan de todos los objetos de la base. Además, las columnas de los tres tipos de vistas no suelen ser exactamente las mismas. Así, las de prefijo USER suelen omitir la columna OWNER (propietario), dado que se da por supuesto que el propietario es el usuario que realiza la consulta. También, algunas vistas con prefijo DBA suelen contener columnas adicionales con información útil para el administrador.
- A continuación, damos una lista de las vistas del catálogo que consideramos más interesantes. Si se desea obtener más información sobre estas vistas, o sobre todas las vistas del catálogo, se debe consultar el manual *Oracle Server Reference*.





## Vistas del catálogo

- **DICTIONARY** Descripción de las tabla y vistas que componen el catálogo
- **DICT** Sinónimo de DICTIONARY
- **DBA-CATALOG, ALL-CATALOG** Información sobre tablas, vistas, sinónimos y secuencias de la base.
- **DBA, ALL, USER-OBJECTS** Información sobre objetos de la base de datos.
- **DBA, ALL, USER-TABLES** Descripción de las tablas de la base de datos.





## Vistas del catálogo

- DBA, ALL, USER-VIEWS Definiciones de vistas.
- DBA, ALL, USER-TAB-COLUMNS Descripción de las columnas de tablas, vistas y *clusters*.
- DBA, ALL, USER-CONSTRAINTS Definición de restricciones definidas sobre tablas.
- DBA, ALL, USER, CONS\_COLUMNS Columnas que forman parte de definiciones de restricciones.
- TABLE-PRIVILEGES, ALL-TAB-PRIVS, USER-TAB-PRIVS, DBA\_TAB-PRIVS Información de privilegios sobre objetos de la base de datos.





## Vistas del catálogo

- COLUMN-PRIVILEGES, ALL-COL-PRIVS,
- USER-COL-PRIVS, DBA\_COL-PRIVS    Información de privilegios sobre columnas.
- USER-SYS-PRIVS, DBA SYS\_PRIVS,    Privilegios del sistema concedidos a usuarios y roles.
- DBA, ALL, USER USERS    Información sobre usuarios definidos en la base de datos.
- DBA, ALL, USER-INDEXES    Descripción de índices definidos sobre tablas *y clusters*.
- DBA, ALL, USER-IND-COLUMNS    Columnas de los índices definidos.
- DBA, ALL, USER-SEQUENCES    Información sobre secuencias.
- DBA, ALL, USER-SYNONYMS    Información sobre sinónimos existentes en la base.





## Vistas del catálogo

- DBA-CLUSTERS, USER-CLUSTERS Descripción de los *clusters* definidos.
- DBA-CLU-COLUMNS,
- USER-CLU-COLUMNS Correspondencias entre columnas de la clave del *cluster* y columnas de las tablas agrupadas en el *cluster*.
- DBA-DATA-FILES Información sobre los ficheros de datos.
- DBA, USER-TABLESPACES Descripción de los *tablespaces*.
- DBA, USER\_SEGMENTS Información sobre segmentos,
- DBA-ROLLBACK-SEGS Descripción de los segmentos de *rollback*.
- DBA, USER\_EXTENTS *Extents* de los segmentos.
- DBA, USER FREE-SPACE *Extents* no asignados de los *tablespaces*.





## 5. GESTIÓN DE LA SEGURIDAD

- o La gestión de la seguridad en Oracle tiene que ver con la gestión de usuarios y concesión o retirada de privilegios. El DBA es el responsable de asignar estos permisos o privilegios a los usuarios.







## 5. GESTIÓN DE LA SEGURIDAD

- Podemos clasificar la seguridad de la BD en dos categorías, seguridad del sistema y de los datos:
  - **Seguridad del sistema:** incluye los mecanismos que controlan el acceso y uso de la BD a nivel de sistema, pe cada vez que un usuario se conecta a la BD, los mecanismos de seguridad comprobarán que éste está autorizado.
  - **Seguridad de los datos:** incluye los mecanismos que controlan el acceso y uso de la BD a nivel de objetos, pe cada vez que un usuario acceda a un objeto, los mecanismos de seguridad comprobarán que éste está autorizado para acceder a ese objeto y qué tipo de operaciones está autorizado a hacer sobre él.





## 5.1 Usuarios

- Un usuario es un nombre definido en la BD que se puede conectar a ella y acceder a determinados objetos según defina el DBA.
- Para acceder a la BD el usuario lo hará a través de SQL\*Plus, SQL developer,...
- Cada usuario tiene asociado un esquema con su mismo nombre y que consta de la colección de todos los objetos (tablas, vistas, funciones,...) de ese usuario. Por defecto cada usuario tiene acceso a todos los objetos de su esquema y puede acceder a los objetos de otro usuario siempre que se le hayan concedido permisos.





## 5.1.1 Creación de usuarios

- Al instalar una BD se crean automáticamente dos usuarios con privilegios DBA: SYS Y SYSTEM
- SYS con clave CHANGE\_ON\_INSTALL y el usuario SYSTEM con clave SYSTEM, ( hay otros usuarios como SCOTT/TIGER y HR/HR pero no son DBA)
- SYS es el propietario de las tablas del diccionario de datos y solamente él puede modificarlas. Es oracle el que maneja las tablas del usuario SYS, ningún usuario aunque sea administrador puede modificarlas. El diccionario de datos está en el tablespace SYSTEM. Los usuarios tienen acceso de solo lectura al diccionario de datos. Contiene objetos de la BD, nombres de usuario, derechos autorizaciones, restricciones, información del espacio libre/ocupado,...





## 5.1.1 Creación de usuarios

- El usuario SYSTEM es creado por Oracle para realizar las tareas de administración de la BD.
- No es conveniente crear tablas de usuario en el esquema de SYSTEM.
- Para crear otros usuarios es preciso conectarse como SYSTEM ya que éste posee el correspondiente privilegio de creación de usuarios
- Al instalar Oracle, el administrador de la BD ha de crearse un usuario para sí mismo con los derechos de administrador y realizar todas las tareas de administración con este nombre de usuario. Para que un usuario pueda a su vez crear usuarios se necesita el privilegio CREATE USER.





## Formato para crear un usuario. Desde SYSTEM / MANAGER

**CREATE USER nombre\_usuario**

**IDENTIFIED (BY password | EXTERNALLY)**

[ DEFAULT TABLESPACE tablespace]-- por defecto USERS

[ TEMPORARY TABLESPACE tablespace] -por defecto TEMP

[ QUOTA (integer (K|M) | UNLIMITED ) ON tablespace) -> por defecto UNLIMITED

[ PASSWORD EXPIRE] -> permite al usuario cambiar de clave al conectarse la primera vez

[ACCOUNT (LOCK | UNLOCK)] -> permite bloquear y desbloquear la cuenta del usuario

[ PROFILE ( perfil | DEFAULT)]; -> se utiliza para controlar el uso de los recursos





- **DEFAULT TABLESPACE:** asigna a un usuario el tablespace por defecto para almacenar sus objetos. Si se omite esta cláusula, por defecto es USERS
- **TEMPORARY TABLESPACE:** es el nombre del tablespace para trabajos temporales, si se omite por defecto es TEMP





- **QUOTA** asigna un espacio en Kbytes o en Megabytes en el espacio\_tabla asignado. **Si no se especifica esta cláusula el usuario no puede crear objetos en el espacio\_tabla.** Para conceder recursos ilimitados a un usuario se le debe dar el privilegio UNLIMITED TABLESPACE con la orden GRANT.
- **PROFILE** perfil sirve para asignar un perfil determinado. Un perfil limita el número de sesiones concurrentes de un usuario, limita el tiempo de uso de la CPU, tiempo de una sesión, desconecta al usuario si sobrepasa el tiempo máximo de uso,.... Por defecto asigna el perfil por omisión al usuario creado





## Nota

- El hecho de crear un usuario no quiere decir que éste tenga privilegios para conectarse a la BD, hay que dárselos explícitamente.







## Ejemplo

```
CREATE USER milagros  
IDENTIFIED BY milagros  
DEFAULT TABLESPACE trabajo  
TEMPORARY TABLESPACE trabajo  
QUOTA 500K on trabajo
```





## Vistas con información de usuarios

- Para obtener información de todos los usuarios creados en la BD existen las vistas:
  - USER\_USERS con información del usuario actual.
  - ALL\_USERS con información acerca de todos los usuarios creados en la BD





## 5.1.2 Modificación de usuarios

Las opciones de create user se cambian con

**ALTER USER nombreusuario**

**IDENTIFIED BY clave\_acceso**

[DEFAULT TABLESPACE espacio\_tabla]

[TEMPORARY TABLESPACE espacio\_tabla].

[QUOTA {entero {K|M} | UNLIMITED} ON espacio\_tabla]

[PROFILE perfil];

Un usuario puede cambiarse su clave de acceso y necesita el privilegio ALTER USER para cambiarse el tablespace, la cuota o el perfil.





## 5.1.3 Borrado de usuarios

- La orden **DROP USER usuario [CASCADE]** borra el usuario y todos sus objetos.
- Para borrar un usuario que tenga objetos creados necesitamos la opción [CASCADE]. Así mismo necesitamos el privilegio DROP USER para poder borrar un usuario.
- Ej: `SELECT owner, table_name from dba_tables where owner = 'PEDRO'`; nos da las tablas de Pedro.
- `DROP USER Pedro CASCADE` borra a Pedro y además sus objetos (tablas,...).





## 5.2 PRIVILEGIOS

- Un privilegio es la capacidad de un usuario dentro de la base de datos a realizar determinadas operaciones o a acceder a determinados objetos de otros usuarios
- Existen aproximadamente 128 privilegios del sistema. La palabra clave ANY en los privilegios significa que los usuarios cuentan con el privilegio en cada esquema
- El comando GRANT agrega un privilegio a un usuario o grupo de usuarios
- El comando REVOKE elimina los privilegios
- Cuando se crea un usuario, es necesario darle privilegios para que pueda hacer algo. Oracle ofrece varios roles(conjunto de privilegios) Entre ellos tenemos: CONNECT, RESOURCE, DBA, EXP\_FULL\_DATABASE y IMP\_FULL\_DATABASE





Rol	Privilegios
CONNECT	Alter session, create cluster, create database link, create sequence, create session, create synonym, create table y create view.
RESOURCE	Create cluster, create procedure, create table, create sequence y create trigger.
DBA	Posee todos los privilegios del sistema





- Hay dos tipos de privilegios que podemos definir en la BD, los de sistema y los de objetos





## Privilegios sobre los objetos.

- Estos privilegios nos permiten acceder y realizar cambios en los datos de otros usuarios y son los que se ofrecen en la tabla siguiente:







Privilegio sobre objetos	Tabla	Vista	Secuencia	<u>Procedure</u>	Sentencias SQL permitidas con cada privilegio
ALTER	X		X		ALTER
DELETE	X	X			DELETE
EXECUTE				X	EXECUTE
INDEX	X				CREATE INDEX ON
INSERT	X	X			INSERT INTO
REFERENCES	X				CREATE o ALTER TABLE
SELECT	X	X	X		SELECT
UPDATE	X	X			UPDATE





## Privilegios sobre los objetos.

La orden para dar privilegios es GRANT y su formato es :

**GRANT { priv\_objeto, [priv\_objeto], .... | ALL  
[PRIVILEGES]}**

**[(columna, [columna],....)]**

**ON [usuario.]objeto**

**TO {usuario | rol | PUBLIC} [,{usuario | rol | PUBLIC}...]**

**[WITH GRANT OPTION];**





## Nota

ON designa los objetos donde se conceden privilegios.

TO identifica usuarios o roles a los que se concede privilegios.

ALL concede todos los privilegios sobre los objetos.

WITH GRANT OPTION concede al usuario receptor el privilegio de concederlo/s a otros usuarios.

PUBLIC incluye todos los usuarios actuales y futuros.





## Ejemplos

- Con la orden GRANT podemos conceder privilegios insert, update o references sobre determinadas columnas de una tabla.

### Ejemplo

- GRANT SELECT, INSERT ON TABLA1 TO FRANCISCO
- GRANT ALL ON TABLA2 TO JUAN
- GRANT UPDATE(TEMPERATURA) ON TABLA1 TO JUAN





## Privilegios del sistema

- Los privilegios del sistema son los que dan derecho a ejecutar un tipo de comando SQL o a realizar alguna acción sobre objetos de un tipo especificado.
- De los 80 privilegios existentes éstos son algunos de ellos:





## Privilegios de sistema

INDEX	CREATE ANY INDEX ALTER ANY INDEX DROP ANY INDEX	
PRIVILEGE	GRANT ANY PRIVILEGE	Conceder cualquier privilegio del sistema.
PROCEDURE	CREATE PROCEDURE CREATE ANY PROCEDUR ALTER ANY PROCEDURE DROP ANY PROCEDURE EXECUTE ANY PROCEDURE	Crear proced. <u>almacenados</u> , funciones y paquetes en esquema propio. Crear proced. <u>almacenados</u> , funciones y paquetes en cualquier esquema.
PROFILE	CREATE,ALTER,DROP	
ROLE	CREATE, ALTER ANY, DROP, GRANT ANY	
ROLLBACK_SEGMENT	CREATE,ALTER,DROP	





SEQUENCE

CREATE, ALTER ANY,  
DROP ANY, SELECT ANY

SESSION

CREATE, ALTER;  
RESTRICTED

SYNONYM

CREATE, CREATE  
PUBLIC, DROP PUBLIC,  
CREATE ANY, DROP ANY

TABLE

CREATE  
CREATE ANY  
ALTER ANY  
DROP ANY  
LOCK ANY  
SELECT ANY TABLE  
INSERT ANY TABLE  
UPDATE ANY TABLE  
DELETE ANY TABLE

TABLESPACE

CREATE, ALTER,  
MANAGE, DROP,  
UNLIMITED



TRIGGER	CREATE, CREATE ANY, ALTER ANY, DROP ANY,
USER	CREATE, ALTER, DROP
VIEW	CREATE, CREATE ANY, DROP ANY

■ Ver lista privilegios del sistema







- Si el privilegio lleva la palabra ANY se entiende que el privilegio se concede sobre cualquier esquema y si no la lleva solamente sobre el propio.
- El formato de la orden GRANT para asignar privilegios del sistema es:

**GRANT {privilegio | rol} [, {privilegio | rol}, ...]**

**TO {usuario | rol | PUBLIC}**  
**[, {usuario | rol | PUBLIC} ] ....**

**[WITH ADMIN OPTION];**

Con with admin option, permite que el receptor del privilegio o rol pueda conceder esos mismos privilegios a otros usuarios o roles.





## Ejemplos

- CREATE USER PEDRO IDENTIFIED BY PEDRO DEFAULT TABLESPACE USER\_DATA;
- GRANT CREATE SESSION TO PEDRO; para que Pedro cree sesión.
- GRANT CONNECT TO PEDRO; Se concede a Pedro el rol CONNECT que conlleva los siguientes privilegios (alter session, create cluster, create database link, create sequence, create session, create synonym, create table y create view
- GRANT DBA TO PEDRO; JUAN;
- GRANT DROP USER TO MILAGROS WITH ADMIN OPTION; Borrar usuarios y poder conceder este privilegio a otros usuarios.
- GRANT SELECT ANY TABLE TO PUBLIC; Cualquier usuario puede hacer select en cualquier tabla.





## Vistas sobre privilegios

- DBA\_SYS\_PRIVS vista con todos los privilegios de sistema de los usuarios de la BD.
- USER\_SYS\_PRIVS vista con los privilegios del sistema asignados al usuario.
- USER\_TAB\_PRIVS concesiones sobre objetos que son propiedad del usuario, concedidos o recibidos por éste.





## Retirada de privilegios

- Para retirar privilegios o roles a usuarios o para retirar privilegios concedidos a los roles la orden es

**REVOKE {priv\_objeto [,priv\_objeto].... | ALL  
[PRIVILEGES]}**

**ON [usuario.]objeto**

**FROM {usuario | rol | PUBLIC} [,{usuario | rol | PUBLIC}]  
...;**





Para retirar privilegios de sistema o roles a usuarios o para retirar privilegios a roles, el formato es el siguiente:

**REVOKE (system\_priv | role) [, (system\_priv | role) ] ...  
FROM (user | role | PUBLIC) [, (user | role | PUBLIC) ]**





## Ejemplos

**REVOKE UPDATE ON TABLE1 FROM PEDRO;  
REVOKE ALL ON TABLA1 FROM FRANCISCO;  
REVOKE SELECT ANY TABLE FROM PUBLIC;  
REVOKE DBA FROM PEDRO;**





## 6. ROLES

- Los roles son grupos específicos de privilegios relacionados que se otorgan a los usuarios o a otros roles. Están diseñados para facilitar la gestión de los privilegios en la base de datos





- Pueden estar formados tanto por privilegios del sistema como por objetos
- Se pueden activar o desactivar para cada usuario al que se le haya otorgado el rol
- Pueden necesitar una contraseña para activarse
- El nombre de cada rol debe ser único entre los nombres de usuario y nombre de rol existentes
- No lo posee nadie, no están en ningún esquema
- Sus descripciones se almacenan en el diccionario de datos.







## CREACIÓN DE ROLES

- Para crear un rol se necesita el privilegio de sistema CREATE ROLE.
- **CREATE ROLE role [NOT IDENTIFIED | IDENTIFIED BY password )**

Identified by password indica que el usuario que desee usar los privilegios tiene que introducir la clave de acceso en la orden SET ROLE para activar el rol





## Conceder privilegios a los roles

**GRANT role [, role] ... TO (user | role | PUBLIC)  
[, (user | role | PUBLIC)] ... [WITH ADMIN  
OPTION];**

CREATE ROLE ACCESO;

GRANT SELECT, INSERT ON EMPLE TO ACCESO;

GRANT CREATE SESSION TO ACCESO;

Una vez creado el rol y asignarle privilegios podemos conceder roles a los usuarios de la misma forma que les concedemos permisos

GRANT ACCESO TO USUARIO1

Podemos posteriormente añadir más privilegios a un rol

GRANT SELECT ON DEPART TO ACCESO





## Nota

- Supongamos que el usuario1 anterior tiene tb permisos para crear vistas. También puede hacer select sobre la tabla emple ya que tiene el rol acceso, pero no podrá crear vistas sobre emple ya que el privilegio create view es sobre objetos de su propiedad.





## Supresión de privilegios en los roles

**REVOKE role [, role]... FROM (user | role | PUBLIC) [, (user | role | PUBLIC) ]...;**

Ejemplo: retiramos el privilegio create table al rol acceso  
**REVOKE CREATE TABLE FROM ACCESO**





## Supresión de un rol

**DROP ROLE rol;**

Ejemplo: borramos el rol acceso

DROP ROLE ACCESO





## Establecer un rol por defecto

- Con la orden CREATE USER no se puede asignar un rol por defecto, hay que hacerlo con la orden ALTER USER
- ALTER USER usuario DEFAULT ROLE rol  
(antes tiene que ser concedido el rol al usuario)





## Vistas sobre roles en el Catálogo

- Todos los roles: DBA\_ROLES
- Privilegios asignados a usuarios y roles: DBA\_ROLE\_PRIVS.
- Privilegios del sistema asignados a usuarios y roles: DBA\_ROLE\_PRIVS.
- Roles asignados al usuario: USER\_ROLE\_PRIVS
- Roles activos para el usuario: SESSION\_ROLES
- Roles asignados a otros roles: ROLE\_ROLE\_PRIVS
- Privilegios sobre tablas aplicados a roles: ROLE\_TAB\_PRIVS
- Privilegios del sistema aplicados a roles: ROLE\_SYS\_PRIVS
- Concesiones sobre objetos del usuario: USER\_TAB\_PRIVS.







## 7. Perfiles

- Un perfil es un conjunto de límites a los recursos de la BD, que se usa para poner límites al uso de recursos de la BD por parte de los usuarios, ej.: tiempo de conexión, ....
- Por omisión se asigna el perfil DEFAULT cuando se da de alta al usuario. Este perfil supone recursos ilimitados (UNLIMITED).
- Los recursos son :





	
SESSIONS_PER_USER	Número de sesiones <u>multiple</u> s concurrentes permitidas por nombre de usuario
CPU_PER_SESSION (1)	Tiempo máximo de CPU por sesión en centésimas de seg.
CPU_PER_CALL	Tiempo máximo de CPU por llamada en centésimas de seg.
CONNECT_TIME(1)	Tiempo de conexión en minutos.
IDLE_TIME	Tiempo de inactividad antes de que el usuario sea desconectado
LOGICAL_READS_PER_SESSION(1)	Bloques de datos leídos en una sesión.
LOGICAL_READS_PER_CALL	Bloques de datos leídos por llamada.
PRIVATE_SGA (1)	Bytes enteros de espacio en SGA.
COMPOSITE_LIMIT	Coste total de recursos en una sesión expresado en unidades de servicio calculado sumando (1).
FAILED_LOGIN_ATTEMPS	Nº de intentos de acceso fallidos consecutivos que producirán el bloqueo de la cuenta
PASSWORD_LOCK_TIEM	Nº de <u>días</u> que quedará bloqueada una cuenta si se sobrepasa el parámetro <u>FAILED_LOGIN_ATTEMP</u>
	
Bases de datos 2011-12	



Para activar el uso de perfiles en el sistema, el administrador deberá ejecutar la orden

`ALTER SYSTEM SET RESOURCE_LIMIT=TRUE`

Con FALSE desactivamos la utilización de perfiles





## Crear el perfil

```
CREATE PROFILE nombreperfil LIMIT  
recurso { Entero [K | M] | UNLIMITED |  
DEFAULT}  
[,recurso {    }]
```

Con UNLIMITED se entiende que no hay límite y  
DEFAULT coge el límite para el perfil DEFAULT.





## Asignar perfil

- Para asignar un perfil a un usuario se puede utilizar la orden  
**ALTER USER usuario  
PROFILE nombreperfil**
- O bien al crear el usuario podemos asignarle el perfil





## Modificar perfil

- Se hace con la orden ALTER PROFILE y el mismo formato que la orden CREATE PROFILE





## Borrar un perfil

DROP PROFILE perfil [CASCADE]

Si algún usuario tiene asignado el perfil,  
es necesario utilizar la opción CASCADE.





# VISTAS

## ■ DBA\_PROFILES





## Pasos para crear un perfil

- Crear el perfil. `CREATE PROFILE`
- Crear el usuario. `CREATE USER` asignándole el perfil o bien con la orden `ALTER USER`
- Conceder al usuario como mínimo el rol `CONNECT`.  
`GRANT CONNECT`
- Activar el uso de perfiles en el sistema, el administrador ha de ejecutar  
`ALTER SYSTEM SET RESOURCE_LIMIT = TRUE;`
- A partir de ahora ya se puede conectar el nuevo usuario utilizando el nuevo perfil







## 8. TABLESPACES

Un tablespace es una unidad lógica de almacenamiento de datos. Está constituido por uno o mas datafiles (ficheros de datos), que son los ficheros físicos que ocupan el espacio en el disco duro. Contiene las tablas relacionadas con una aplicación. Es conveniente colocar todas las tablas de una misma aplicación en un mismo tablespace.





## TABLESPACES

- Cuando se instala ORACLE se crean varios tablespaces:
- USERS : Contiene información personal de los usuarios.
- SYSTEM : contiene información propia de ORACLE.
- UNDOTBS1: ORACLE guarda información para deshacer cambios en BD.
- TEMP: ORACLE almacena sus tablas temporales.





## Sintaxis

```
CREATE TABLESPACE nombretablespace  
DATAFILE 'nombrearchivo' [SIZE entero[K|M] ] [REUSE]  
[AUTOEXTEND {OFF|ON clausulas}]  
[, 'nombrearchivo' [SIZE entero[K|M] ] [REUSE] ...  
[DEFAULT STORAGE  
  ( INITIAL          tamaño  
    NEXT              tamaño  
    MINEXTENTS        tamaño  
    MAXEXTENTS        tamaño  
    PCTINCREASE       valor  
  )]  
[ONLINE|OFFLINE];
```





## TABLESPACES

- DATAFILE son el/los archivos de que consta el tablespace.
- SIZE entero: especifica el tamaño del tablespace puede venir en K o M
- DEFAULT STORAGE define el almacenamiento por omisión para todos los objetos que se creen en este tablespace.
- REUSE usa el archivo si ya existe o lo crea si no existe.
- Con ONLINE el tablespace está disponible (opción por defecto) y con OFFLINE no lo está.
- AUTOEXTEND: activa o desactiva el crecimiento automático de los archivos de datos del tablespace. Ejemplo:

AUTOEXTEND ON NEXT entero MAXSIZE {UNLIMITED|entero {K|M}}





En el siguiente ejemplo creamos un tablespace de 15M llamado trabajo, el tamaño inicial para el objeto que se cree en el tablespace es de 10K y el tamaño de la siguiente extensión del objeto, también es 10K. Cada extensión subsiguiente será un 25% mayor que la anterior.

```
CREATE TABLESPACE trabajo DATAFILE  
'trabaj1.ora' size 10M, 'trabaj2.ora' size 5 M  
DEFAULT STORAGE (INITIAL 10K NEXT 10K  
PCTINCREASE 25);
```





## Vistas

- La vista(utilizable sólo por DBA) DBA\_DATA\_FILES muestra los archivos de cada tablespace.
- La vista USER\_FREE\_SPACE muestra las extensiones libres a las que puede acceder el usuario y da la información siguiente:
- Nombre del tablespace, FILE\_ID: Número de identificación interna del archivo, BLOCK\_ID: Identificación del primer bloque libre, BYTES: Número de bytes libres, BLOCKS : Número de bloques libres, RELATIVE\_FNO: Número relativo del fichero en la primera extensión del bloque.
- La vista DBA\_FREE\_SPACE (sólo DBA) muestra extensiones libres en todos los tablespaces.
- La vista DBA\_TABLESPACES (sólo DBA) describe los tablespaces.
- La vista DBA\_TS\_QUOTAS(solo DBA) describe los bytes utilizados por usuario en cada tablespace.





## Para modificar un tablespace :

### **ALTER TABLESPACE nombretablespace**

```
{
[ADD DATAFILE 'nombreadarchivo' [SIZE entero[K|M] ] [REUSE] [AUTOEXTEND ON NEXT
    entero {K|M} MAXSIZE {UNLIMITED | entero.{K|M}.| OFF]
[ADD DATAFILE 'nombreadarchivo' [SIZE entero[K|M] ] [REUSE] [AUTOEXTEND ON
    ...|OFF]...
]
[RENAME DATAFILE 'archivo' [, 'archivo']... TO 'archivo' [, 'archivo' [, 'archivo'] ]
DEFAULT STORAGE
(
    INITIAL          tamaño
    NEXT             tamaño
    MINEXTENTS       tamaño
    MAXEXTENTS       tamaño
    PCTINCREASE      valor
)]
[ONLINE|OFFLINE];
```





## Ejemplos

- ALTER TABLESPACE TRABAJO OFFLINE.
- ALTER TABLESPACE TRABAJO ONLINE.
- ALTER TABLESPACE TRABAJO ADD  
DATAFILE 'trabaj3.ora' size 6 M;







## Borrado de tablespaces

- **DROP TABLESPACE nombretablespace;** borra tablespaces vacíos.
- DROP TABLESPACE nombretablespace INCLUDING CONTENTS; borra cualquier tablespace.
- Es recomendable desactivar un tablespace antes de borrarlo.
- Una vez borrado el tablespace debemos borrar los archivos físicos con órdenes del sistema operativo.





## 9. Otros objetos

- 9.1 Secuencias
- 9.2 Índices





## 9.1 Secuencias

- Una secuencia es un objeto de la BD que sirve para generar números enteros únicos, es muy útil para generar automáticamente claves primarias. Para crear una secuencia en el esquema propio es necesario tener el privilegio CREATE SEQUENCE





## 9.1 Secuencias

- Es una rutina interna de la base de datos la que realiza la función de generar un número distinto cada vez.
- Las secuencias se almacenan independientemente de la tabla, por lo que la misma secuencia se puede utilizar para diversas tablas.





## Formato

```
CREATE SEQUENCE secuencia  
[INCREMENT BY n]  
[START WITH n]  
[{MAXVALUE n | NOMAXVALUE}]  
[{MINVALUE n | NOMINVALUE}]  
[{CYCLE | NOCYCLE}]
```





## Nota

- Secuencia. Es el nombre que se le da al objeto de secuencia
- INCREMENT BY. Indica cuánto se incrementa la secuencia cada vez que se usa. Por defecto se incrementa de uno en uno
- START WITH. Indica el valor inicial de la secuencia (por defecto 1)
- MAXVALUE. Máximo valor que puede tomar la secuencia. Si no se toma NOMAXVALUE que permite llegar hasta el 1027
- MINVALUE. Mínimo valor que puede tomar la secuencia. Por defecto -1026
- CYCLE. Hace que la secuencia vuelva a empezar si se ha llegado al máximo valor.





**Ejemplo:**

```
CREATE SEQUENCE numeroPlanta  
INCREMENT 100  
STARTS WITH 100  
MAXVALUE 2000;
```





## Vistas de secuencias

- La vista del diccionario de datos de Oracle `USER_SEQUENCES` muestra la lista de secuencias actuales. La columna `LAST_NUMBER` muestra cual será el siguiente número de secuencia disponible.







## Uso de secuencias

- Los métodos NEXTVAL y CURRVAL se utilizan para obtener el siguiente número y el valor actual de la secuencia respectivamente.
- Ejemplo  
`SELECT numeroPlanta.NEXTVAL FROM DUAL;`





## Uso de secuencias

- Realmente NEXTVAL incrementa la secuencia y devuelve el valor actual.
- CURRVAL devuelve el valor de la secuencia, pero sin incrementar la misma.





## Uso de secuencias

Ambas funciones pueden ser utilizadas en:

- Una consulta SELECT que no lleve DISTINCT, ni grupos, ni sea parte de una vista, ni sea subconsulta de otro SELECT, UPDATE o DELETE
- Una subconsulta SELECT en una instrucción INSERT
- La cláusula VALUES de la instrucción INSERT
- La cláusula SET de la instrucción UPDATE
- No se puede utilizar (y siempre hay tentaciones para ello) como valor para la cláusula DEFAULT de un campo de tabla.





## Uso de secuencias

Su uso más habitual es como apoyo al comando INSERT (en Oracle):

```
INSERT INTO plantas(num, uso)
VALUES(numeroPlanta.NEXTVAL, 'Suites');
```





## Modificar secuencias

Se pueden modificar las secuencias, pero la modificación sólo puede afectar a los futuros valores de la secuencia.

```
ALTER SEQUENCE secuencia  
[INCREMENT BY n]  
[START WITH n]  
[{MAXVALUE n | NOMAXVALUE}]  
[{MINVALUE n | NOMINVALUE}]  
[{CYCLE | NOCYCLE}]
```





## Borrado de secuencias

- Lo hace el comando DROP SEQUENCE seguido del nombre de la secuencia a borrar.





## Vistas de secuencias

- En Oracle se hace mediante la vista **USER\_SEQUENCES**, permite observar la lista de secuencias del usuario.





## 9.2 Índices

- Los índices son objetos que forman parte del esquema que hacen que las bases de datos aceleren las operaciones de consulta y ordenación sobre los campos a los que el índice hace referencia.
- Se almacenan aparte de la tabla a la que hace referencia, lo que permite crearlos y borrarlos en cualquier momento.







## 9.2 Índices

- Lo que realizan es una lista ordenada por la que Oracle puede acceder para facilitar la búsqueda de los datos. Cada vez que se añade un nuevo registro, los índices involucrados se actualizan a fin de que su información esté al día. De ahí que cuantos más índices haya, más le cuesta a Oracle añadir registros, pero más rápidas se realizan las instrucciones de consulta.
- La mayoría de los índices se crean de manera implícita, como consecuencia de las restricciones PRIMARY KEY, UNIQUE y FOREIGN KEY. Estos son índices obligatorios, por lo que los crea el propio SGBD.





## CREACIÓN DE ÍNDICES

- Aparte de los índices obligatorios comentados anteriormente, se pueden crear índices de forma explícita. Éstos se crean para aquellos campos sobre los cuales se realizarán búsquedas e instrucciones de ordenación frecuente.





## SINTAXIS

Sintaxis:

```
CREATE INDEX nombre ON tabla (columna1 [,columna2...])
```

Ejemplo:

```
CREATE INDEX nombre_completo ON clientes (apellido1, apellido2, nombre);
```

El ejemplo crea un índice para los campos apellido1, apellido2 y nombre. Esto no es lo mismo que crear un índice para cada campo, este índice es efectivo cuando se buscan u ordenan clientes usando los tres campos (apellido1, apellido2, nombre) a la vez.





## NOTAS

Se aconseja crear índices en campos que:

- Contengan una gran cantidad de valores.
- Contengan una gran cantidad de nulos.
- Sean parte habitual de cláusulas WHERE, GROUP BY u ORDER BY.
- Sean parte de listados de consultas de grandes tablas sobre las que casi siempre se muestran como mucho un 4% de su contenido.





## NOTAS

No se aconseja crear índices en campos que:

- Pertenezcan a tablas pequeñas
- No se usen a menudo en las consultas
- Pertenecen a tablas cuyas consultas muestran menos de un 4% del total de registros
- Pertenecen a tablas que se actualizan frecuentemente
- Se utilizan en expresiones





## Vistas sobre índices

- Para ver la lista de índices en Oracle se utiliza la vista **USER\_INDEXES**.
- Mientras que la vista **USER\_IND\_COLUMNS** Muestra la lista de columnas que son utilizadas por índices.





## Borrado de índices

- La instrucción **DROP INDEX** seguida del nombre del índice permite eliminar el índice en cuestión.

