



Desarrollo de Aplicaciones Informáticas

CICLO FORMATIVO DE GRADO SUPERIOR

FORMACIÓN PROFESIONAL A DISTANCIA

Unidad 2

Organización de un SGBD relacional

MÓDULO

Desarrollo de Aplicaciones en Entornos de Cuarta Generación y con Herramientas CASE



FORMACIÓN PROFESIONAL

Principado de Asturias

Título del Ciclo: DESARROLLO DE APLICACIONES INFORMATICAS

Título del Módulo: DESARROLLO DE APLICACIONES EN ENTORNOS DE CUARTA GENERACIÓN Y CON HERRAMIENTAS CASE

Unidad 2: **Organización de un SGBD relacional**

Dirección: Dirección General de Políticas Educativas, Ordenación Académica y Formación Profesional
Servicio de Formación Profesional Inicial y Aprendizaje Permanente

Dirección de la obra:

Alfonso Careaga Herrera
Antonio Reguera García
Arturo García Fernández
Ascensión Solís Fernández
Juan Carlos Quirós Quirós
Luís M^a Palacio Junquera
Yolanda Álvarez Granda

Coordinador de los contenidos:

Juan Manuel Fernández Gutiérrez

Autores:

Juan Manuel Fernández Gutiérrez
Rodrigo Fernández Martínez

Colección:

Materiales didácticos de aula

Serie:

Formación profesional específica

Edita:

Consejería de Educación y Ciencia

ISBN:

978-84-692-3443-2

Deposito Legal:

AS-3564-2009

Copyright:

2009. Consejería de Educación y Ciencia

Esta publicación tiene fines exclusivamente educativos. Queda prohibida la venta, así como la reproducción total o parcial de sus contenidos sin autorización expresa de los autores y del Copyright

Introducción

Visto el modelo relacional en la unidad de trabajo anterior, nos ocuparemos ahora del estudio de las características y organización de los Sistemas Gestores de Bases de Datos que lo implementan.

La unidad comienza analizando qué características debe poseer un SGBD para que pueda ser calificado como relacional, pasando a continuación al estudio de la organización de estos sistemas.

La organización de un SGBD relacional se estudia viendo las características del SGBD Oracle. Se trata, en primer lugar, la estructura de la base de datos, analizando cómo se organizan y gestionan las estructuras de almacenamiento para dar soporte a los objetos relacionales, y para lograr una protección adecuada de los datos y un rendimiento óptimo en el acceso a los mismos. Después, se describen las estructuras de memoria y los procesos que desarrollan la funcionalidad del SGBD. Finalmente, nos ocupamos del catálogo de la base de datos.

Objetivos

- Identificar las características que definen un SGBD cómo relacional.
- Describir los elementos que componen la estructuras físicas y lógicas de una base de datos.
- Relacionar la estructura de una base de datos con los tres niveles de estructuración ANSI.
- Comprender los mecanismos de gestión del espacio de almacenamiento.
- Identificar los mecanismos que optimizan los procesos de acceso a los datos.
- Conocer los principales procesos y estructuras de memoria de un SGBD Oracle.
- Explicar la estructura del diccionario de datos de Oracle.

Contenidos Generales

1. CARACTERIZACIÓN DE UN SGBD RELACIONAL.....	3
2. EL SGBD ORACLE (ORACLE SERVER).....	4
3. ARQUITECTURA DE LA BASE DE DATOS.	4
4. ESTRUCTURA FÍSICA DE LA BASE DE DATOS.....	5
5. TABLESPACES, BLOQUES DE DATOS, EXTENTS Y SEGMENTOS.....	7
6. OBJETOS DE ESQUEMA.	10
7. ESTRUCTURAS DE MEMORIA Y PROCESOS.....	16
8. EL DICCIONARIO DE DATOS.	19

1. Caracterización de un SGBD Relacional

La caracterización de un sistema gestor de bases de datos como relacional es una cuestión que tiene cierta complejidad. En principio, para que un sistema sea relacional debería implementar con total fidelidad el modelo relacional. La realidad es que, aunque existen en el mercado multitud de productos con la etiqueta de relacional, ninguno cumple la condición anterior. Entonces, habría que tener en cuenta que no todas las características del modelo tienen la misma importancia, con lo que bastaría que un SGBD incorporase en su diseño aquéllas consideradas como fundamentales para poder ser llamado relacional.

En 1982 E. F. Codd estableció que para que un SGBD pueda considerarse 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 restricción, proyección y reunión natural, sin requerir definiciones previas de rutas de acceso físico.

La primera característica, que es fundamental en todo sistema relacional al facilitar la comprensión y el uso de la base de datos por parte del usuario, no basta para definir un sistema como relacional. Son necesarias también unas operaciones que, aunque no sean todas las del álgebra relacional, permiten resolver la mayoría de los problemas prácticos.

Un SGBD que disponga de las tres operaciones anteriores pero que necesite de la definición previa por el usuario de rutas de acceso físico para manejarlas no puede considerarse como relacional, ya que ello significaría atentar contra uno de los principios del modelo relacional: la independencia entre la estructura lógica de los datos y su almacenamiento interno. Así, por ejemplo, no puede considerarse relacional un sistema que implementa la operación de restricción de forma que en la condición solo puedan intervenir columnas sobre las que existe definido un índice.

La existencia de índices, o cualquier otro mecanismo de rutas de acceso físico a los datos, mejora el rendimiento del sistema cuando procesa operaciones de manipulación de la base de datos. Aunque el uso de estos mecanismos no puede imponerse a los usuarios, el propio SGBD relacional los utiliza. Ante cualquier operación planteada por el usuario, la parte del sistema conocida como el **optimizador** se encargará de establecer las rutas más eficientes de acceso a los datos.

Según C. J. Date, un SGBD que sólo cumpla los dos criterios anteriores es un sistema **mínimamente relacional**. Si, además, maneja todas las operaciones del álgebra relacional se puede decir que es **relacionalmente completo**. Y si, también, posee otras características del

modelo relacional, incluyendo los dominios y las dos reglas de integridad, se denomina **totalmente relacional**.

En el resto del capítulo, analizaremos las características de un sistema relacional real: el SGBD Oracle. A través de este estudio veremos cómo se implementan en la práctica muchas de las características del modelo relacional.

2. El SGBD Oracle (Oracle Server).

En el año 1979 Relational Software Inc. presenta el primer SGBD relacional: Oracle V2. Esta versión no soportaba transacciones, pero sí toda la funcionalidad SQL de queries y joins.

En 1983 RSI cambia su nombre definitivo a Oracle Corporation, y lanza Oracle V3, agregando el manejo de transacciones a través de las instrucciones COMMIT y ROLLBACK.

En años posteriores, van surgiendo sucesivas versiones que amplían y potencian las capacidades de este sistema. Paralelamente al desarrollo de estas versiones, aparecen herramientas de análisis, desarrollo y usuario final que hacen uso y complementan la funcionalidad del SGBD Oracle. En la actualidad el SGBD Oracle domina totalmente el mercado de las bases de datos, y las versiones del producto están disponible para una gran variedad de plataformas hardware/software.

Un SGBD Oracle (*Oracle server* en la documentación del producto) consta de: **una base de datos Oracle** y **una instancia Oracle**.

Una **base de datos Oracle** es un conjunto de información tratado como una unidad, y una **instancia Oracle** está formada por un conjunto de procesos y estructuras de memoria compartidas que permiten definir, almacenar y manipular la base de datos, así como controlar el acceso, concurrencia y uso de la información.

3. Arquitectura de la base de datos.

En la documentación del producto, se dice que una base de datos Oracle tiene una **estructura física** y una **estructura lógica**. La **estructura física** está constituida por los ficheros del sistema operativo que dan soporte a los datos. Los *tablespaces* y los

Estructura de una base de datos Oracle		Niveles ANSI
Estructura lógica	Vistas	Externo
	Tablas	Conceptual
	Tablespaces, índices, clusters, etc.	Interno
Estructura física	Ficheros de datos Ficheros de redo log Ficheros de control	

Los niveles ANSI y la estructura de la base de datos Oracle.

objetos de esquema son los elementos que configuran la **estructura lógica**.

Cuando Oracle habla de “estructura lógica”, está incluyendo en el término elementos que corresponden a los niveles externo y conceptual de la arquitectura ANSI, lo que en el capítulo anterior considerábamos estructuras lógicas, pero también elementos que corresponden al nivel interno ANSI. En la figura se muestran los niveles ANSI de la base de datos Oracle, observamos que algunos elementos de la estructura lógica Oracle (tablespaces, índices y *clusters*) están en el nivel interno ANSI.

Los SGBD se apoyan, para gestionar el acceso a los datos, en el sistema operativo sobre el que se ejecutan. En qué grado tiene lugar este apoyo, es decir, qué estructuras son gestionadas con el apoyo del sistema operativo y cuáles directamente por el SGBD, varía de unos sistemas a otros. En el caso de Oracle, la estructura física es gestionada a través de las rutinas del sistema operativo y la estructura lógica es gestionada directamente por el software de Oracle.

El SGBD de Oracle soporta el lenguaje SQL, tanto autocontenido como huésped. El usuario puede plantear operaciones de definición o manipulación de objetos de los niveles externo y conceptual en SQL estándar. Para definir de objetos del nivel interno el SGBD también soporta SQL, aunque, como es lógico, se trata de una extensión del lenguaje propia de Oracle.

4. Estructura física de la base de datos.

La estructura física de una base de datos ORACLE está compuesta por tres tipos de ficheros: **ficheros de datos**, **ficheros redo log** y **ficheros de control**.

Una base de datos Oracle está asociada a uno o más **ficheros de datos**, que contendrán todos los datos de la base. Los datos de las estructuras lógicas, como tablas e índices, se almacenan físicamente en ficheros reservados para la base de datos.

Los **ficheros de datos** tienen las siguientes características:

- Un fichero sólo puede estar asociado a una base de datos.
- El fichero, aunque en el momento su creación para asociarlo a la base de datos tiene un tamaño fijo, puede definirse con características de “*auto-extend*” para hacer que incremente su tamaño en una cierta cantidad cuando se agote el espacio de almacenamiento de la base de datos.
- Uno o más ficheros de una base de datos forman una unidad lógica de almacenamiento denominada **tablespace**.

Durante el funcionamiento normal de la base de datos, la información contenida en un fichero de datos se lee, cuando se necesita, y se almacena en memoria caché del servidor Oracle. Por ejemplo, si un usuario quiere acceder a ciertos datos de una tabla sólo se lee del fichero correspondiente si la información solicitada no está en la caché.

La información nueva, o la modificada, no se graba necesariamente de inmediato en un fichero de datos. Para reducir el número de accesos a disco y, en consecuencia, mejorar el rendimiento, los datos se almacenan en memoria y son grabados todos de una vez, en determinado instante, por el proceso de la instancia Oracle denominado “*Database Writer*”.

Los ficheros de **redo log** desempeñan un papel fundamental en la seguridad de la base de datos. El concepto de fichero de *redo log* corresponde exactamente al de fichero **diario** visto en la unidad anterior, y, por tanto, se trata de un fichero en el que el SGBD graba la información necesaria para la recuperación de la base de datos después de un fallo en el sistema o en los dispositivos de almacenamiento.

Cada uno de los ficheros de *redo log* tiene un tamaño que se determina en el momento de su creación y que no es modificable posteriormente. Una base de datos tiene dos o más ficheros de este tipo siendo utilizados de forma cíclica, es decir, el SGBD empieza a grabar en uno de ellos y cuando lo llena pasa al siguiente, luego al siguiente, y así sucesivamente hasta llenarlos todos, momento en que empieza a grabar nuevamente en el primero.

Una base de datos Oracle puede funcionar en modo *NOARCHIVELOG*, modo por defecto, o en *ARCHIVELOG*. Cuando trabaja en el segundo modo un proceso del SGBD se encarga de salvar el contenido de los ficheros de *redo log* que se han llenado en otro soporte. De esta forma, en una base de datos trabajando en modo *ARCHIVELOG* se habla de ficheros *archived redo log*, u *offline redo log*, refiriéndose a aquellos copiados en un soporte fuera de línea, y ficheros *online redo log*, aquellos en disco sobre los que graba directamente el SGBD. Cuando se trata de recuperar la base de datos, después de un fallo que haya destruido ficheros de la base, son necesarios todos los ficheros *archived redo log* generados desde el último *backup*, además, por supuesto, de dicho *backup*.

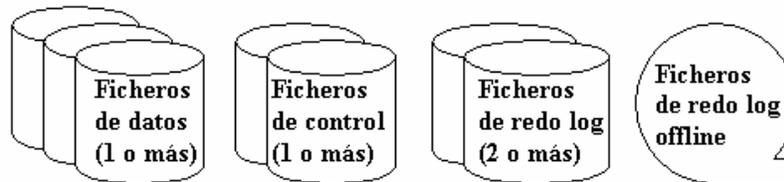
Puesto que la información de *redo log* es crítica en la seguridad de la base de datos, para protegerse contra fallos en los propios ficheros de *redo log* Oracle da la posibilidad de disponer de copias de estos ficheros en diferentes discos.

Cada base de datos Oracle tiene un **fichero de control**. En este fichero se registra la estructura física de la base de datos, y contiene, entre otros, los siguientes datos:

- Nombre de la base de datos.

- 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.

El **fichero de control** es imprescindible en el momento de arranque de la base para localizar los ficheros de datos y de *redo log* a utilizar. También, es imprescindible en el proceso de recuperación en caliente de la base de datos después de un fallo, puesto que, según vimos



Ficheros de una base de datos Oracle.

en la unidad 1, la información sobre el último punto de verificación ejecutado es fundamental para saber qué transacciones deben ser rehechas y cuáles deshechas.

Este fichero, que en la unidad 1 llamábamos fichero de reanque, debe estar disponible en todo momento durante el funcionamiento del SGBD, dado que en él se va a grabar información cada vez que se produzca un *checkpoint* o se modifique la estructura física de la base de datos.

Al igual que en el caso de los ficheros de *redo log*, existe la posibilidad de gestionar varias copias del fichero de control en diferentes dispositivos, con lo que se protege la información vital que contiene contra posibles fallos de disco.

5. Tablespaces, bloques de datos, extents y segmentos.

En una base de datos Oracle el espacio de almacenamiento físico, constituido por los ficheros de datos, está estructurado en una o varias unidades lógicas denominadas *tablespaces*. El SGBD gestiona estas unidades lógicas asignando espacio en ellas a los objetos de esquema como tablas, índices, etc.; para almacenar sus datos.

Toda base de datos Oracle dispone al menos de un *tablespace* de nombre *SYSTEM* en el que están las tablas que soportan el catálogo. Como vimos en el capítulo anterior, el catálogo es el diccionario activo gestionado por el propio SGBD que debe existir en todo sistema relacional.

Un *tablespace* puede estar *online* u *offline*. Normalmente todos los *tablespaces* están *online*, con lo que todos los objetos con espacio asignado en ellos están accesibles. En

momentos determinados, el administrador de la base de datos puede poner *offline* algún *tablespace* para realizar en él tareas de mantenimiento como *backups* o actualizaciones. Mientras un *tablespace* esté *offline* los objetos almacenados en él permanecen inaccesibles para los usuarios y aplicaciones. El *tablespace* *SYSTEM* siempre debe estar *online*, ya que en caso contrario el SGBD no podría funcionar ante la imposibilidad de acceder al catálogo.

Los *tablespaces* facilitan las tareas de administración de la base de datos. Así, por ejemplo, los *tablespaces* permiten al administrador:

- Controlar la asignación de espacio en disco a los distintivos objetos de la base, decidiendo en qué *tablespace* se almacenarán los datos de cada objeto y estableciendo un límite de ocupación de espacio para cada objeto.
- Controlar el uso del espacio de almacenamiento por parte de los usuarios asignándoles límites específicos en cada *tablespace*.
- Controlar la disponibilidad de los datos poniendo determinados *tablespaces online* u *offline*.
- Realizar operaciones parciales de *backup* o *restore* de la base de datos.

El espacio total de almacenamiento de la base de datos será la suma de los espacios de cada *tablespace* y si en algún momento dicho espacio se agota, el administrador debe ampliarlo. El espacio de una base de datos puede ampliarse creando un nuevo *tablespace* o haciendo más grande un *tablespace* ya existente. Un *tablespace* se puede ampliar añadiéndole un nuevo fichero o permitiendo que sus ficheros crezcan dinámicamente cuando se precisa más espacio.

Para el SGBD un *tablespace* es un conjunto de **bloques de datos** o **páginas** en los que va almacenando los datos de los distintos objetos. Un bloque se corresponde con un número determinado de bytes en disco. Todos los bloques de un *tablespace* son del mismo tamaño. El tamaño del bloque de datos del *tablespace* *SYSTEM* se establece en el momento de la creación de la base y debe ser un múltiplo del tamaño del bloque físico manejado por el sistema operativo; su valor puede estar comprendido entre 2KB y 32KB

Cada bloque contiene datos de un solo objeto, y un objeto tendrá, normalmente, repartidos sus datos por varios bloques del mismo *tablespace*. Cuando es preciso almacenar más datos en un objeto y en los bloques que tiene asignados no hay espacio, el SGBD le asigna más bloques. La unidad de asignación de bloques a un objeto es el *extent*.

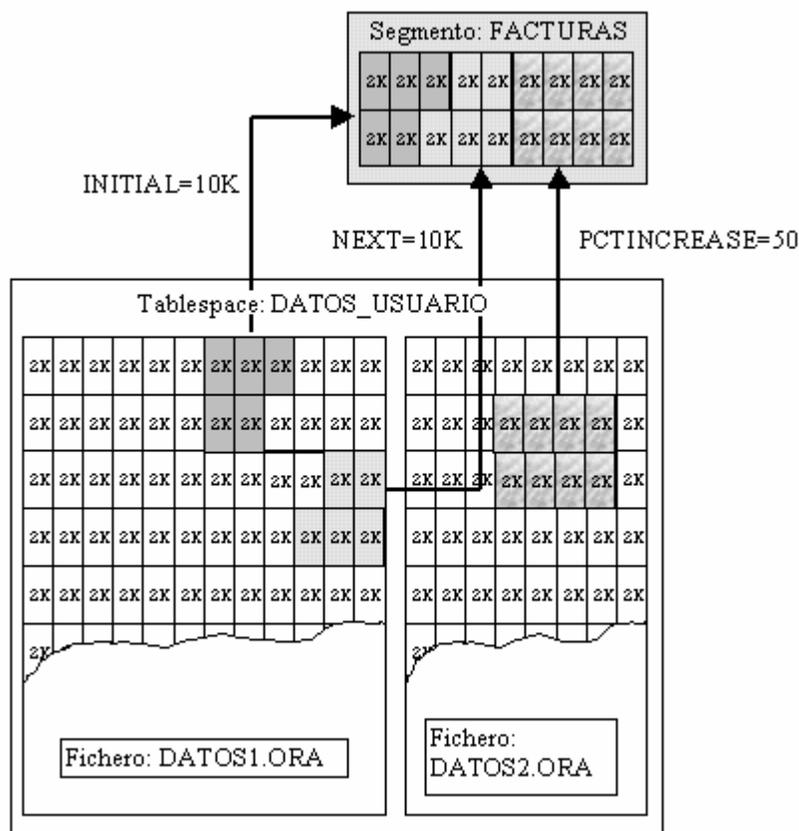
Un *extent* es un número determinado de bloques contiguos que se asignan en una sola operación a un objeto. El tamaño del *extent* varía tanto de un objeto a otro como de una

asignación a otra en el mismo objeto. Este tamaño viene controlado por los parámetros *INITIAL*, *NEXT* y *PCTINCREASE*.

Los valores de los parámetros *INITIAL* y *NEXT* marcan, respectivamente, los tamaños del primer y segundo *extent* asignados a un objeto. El tamaño de cualquier otro *extent* será igual al tamaño del último *extent* asignado al objeto incrementado en un porcentaje igual al valor del parámetro *PCTINCREASE*.

Los valores de los parámetros anteriores se pueden dar al crear cada objeto, aunque, si no se hace así, se aplicarán al objeto los valores dados a los parámetros al crear el *tablespace* en que se va a almacenar.

El conjunto de *extents* asignados a un objeto recibe el nombre de **segmento**. Todos los *extents* de un segmento pertenecen al mismo *tablespace*, aunque pueden pertenecer a distintos



Segmentos, extents y bloques.

ficheros de dicho *tablespace*. Cuando se crea un objeto, por ejemplo una tabla, el SGBD le asigna un segmento con uno o más *extents*. El número de *extents* iniciales de un segmento viene dado por el parámetro *MINEXTENS*, cuyo valor por defecto es 2 para los segmentos de *rollback* y 1 para el resto.

En la figura vemos un ejemplo correspondiente al segmento de la tabla *FACTURAS*. En un determinado instante tiene 18 bloques, asignados mediante 3 *extents* cuyos tamaños fueron marcados por los parámetros *INITIAL*, *NEXT* y *PCTINCREASE*. De

acuerdo con el valor de *PCTINCREASE*, el tercer *extent* debería ser un 50% mayor que el segundo, es decir, de 15K; pero dado que el mínimo espacio que se puede asignar es un bloque, el SGBD redondea siempre los valores por arriba a un número entero de bloques, y por tanto asignó 16K (8 bloques).

Existen cuatro tipos de segmentos: de datos, de índice, temporales y de *rollback*.

Los **segmentos de datos** y los **segmentos de índice** se utilizan, respectivamente, para almacenar los datos de las tablas e índices de la base. A cada índice definido en la base el SGBD le asigna un segmento de índice y a cada tabla o *cluster* le asigna un segmento de datos. Un *cluster* es una estructura que permite almacenar juntas una o más tablas.

Los **segmentos temporales** son empleados por el SGBD para almacenar la información producida en las etapas intermedias del proceso de ciertas sentencias SQL. Por ejemplo, cuando se ejecuta una sentencia SQL de consulta que exige un resultado con las filas ordenadas, el SGBD intentará obtener el resultado mediante el uso de índices, si no es posible, ordenará el resultado de la consulta en memoria, pero si no hay suficiente memoria disponible, utilizará un segmento temporal para realizar la operación.

Los **segmentos de *rollback*** son usados por el SGBD durante la ejecución de las transacciones. En la unidad de trabajo anterior, vimos que una transacción es una secuencia de operaciones que han de ejecutarse de forma atómica, es decir, o se realizan todas o ninguna. Por cada operación de la transacción que se ejecute, el SGBD guarda en un segmento de *rollback* la información necesaria que permita deshacerla más adelante si fuera necesario. La acción de deshacer todas las operaciones ya ejecutadas de una transacción se denomina ***rollback***.

La información almacenada en los segmentos de *rollback* se utiliza también para facilitar lecturas consistentes de los datos involucrados en una transacción. Cuando un usuario o una aplicación consultan datos que han sido modificados por una transacción aun sin concluir, el SGBD suministra los valores previos a la modificación utilizando la información contenida en el correspondiente segmento de *rollback*, con lo que el resultado de la consulta es el mismo que si se hubiese realizado justo antes del comienzo de la transacción. De esta forma, los efectos de una transacción no son visibles hasta su conclusión.

Los segmentos de *rollback* también son vitales en la recuperación la base de datos después de un fallo del sistema.

6. Objetos de esquema.

En una base de datos Oracle se denomina **esquema** a una colección de objetos. Asociado a cada usuario definido en la base de datos existe un esquema con su mismo nombre. Todos los objetos de un esquema son propiedad del usuario asociado a él.

Los **objetos de esquema** son estructuras lógicas que aluden directamente a los datos de la base y que aparecen siempre formando parte de un esquema. En la figura aparecen tipos de objetos de esquema y otros que no pueden formar parte de un esquema. Estudiaremos a continuación las características de algunos de estos objetos.

No existe relación entre los *tablespaces* y los esquemas; objetos en el mismo esquema pueden estar almacenados en distintos *tablespaces*, y un *tablespace* puede contener objetos de diferentes esquemas.

Los datos de un objeto de esquema están almacenados en un solo *tablespace*, pero pueden estar repartidos por los ficheros de dicho *tablespace*.

Objetos de esquema	Objetos sin esquema
<ul style="list-style-type: none">• Tablas• Vistas• Índices• Clusters• Sinónimos• Secuencias	<ul style="list-style-type: none">• Segmentos de <i>rollback</i>• <i>Tablespaces</i>• Usuarios• Roles• <i>Profiles</i>
<ul style="list-style-type: none">• Funciones almacenadas• Procedimientos almacenados• Paquetes• Disparadores• <i>Snapshots</i>• <i>Database links</i>	

Objetos de esquema

□ **Tablas.**

Sabemos que una tabla es la unidad básica de almacenamiento de un sistema relacional.

Cuando se crea una tabla que no forma parte de un *cluster*, el SGBD Oracle le asigna un segmento, en el que posteriormente se almacenarán sus datos. El segmento asignado pertenecerá al *tablespace* indicado en la sentencia de creación de la tabla, aunque, si no se indica nada, pertenecerá al *tablespace* por defecto del usuario que da la orden de creación. Todo usuario definido en una base de datos tiene asignado un *tablespace* por defecto en el que se almacenarán los objetos creados por él cuando no se explicita otro en las correspondientes sentencias de creación.

Como ya se dijo, el segmento asignado a una tabla tendrá inicialmente un número de *extents* igual al valor dado al parámetro *MINEXTENTS*. Este parámetro tiene un valor en cada *tablespace*, que se asigna al crearlo, pero también puede asignársele un valor específico para una tabla en el momento de crearla. El valor de tabla prevalece sobre el de *tablespace*.

El SGBD Oracle almacena las filas de cada tabla en los bloques del segmento asignado; en cada bloque se almacena una o más filas, dependiendo de su tamaño. Cada fila queda identificada por un número, único dentro de la base de datos, que se denomina *ROWID* y que corresponde a su dirección física.

El formato de *ROWID* cambió a partir de la versión 8 de Oracle; en la documentación se utiliza el término ***ROWID extendido(extended rowid)*** para referirse al formato actual, y ***ROWID restringido(restricted rowid)*** cuando se habla del utilizado en la versión 7 y anteriores. En lo que sigue nos referiremos siempre al ***ROWID extendido***.

ROWID	DENOMINACION
AAAM7IAAEAAAAGkAAA	TECLADO LOGITECH A23
AAAM7IAAEAAAAGkAAB	RATON LOGITECH X22
AAAM7IAAEAAAAGkAAC	MONITOR SAMSUNG 15"
AAAM7IAAEAAAAGkAAD	MONITOR SAMSUNG 17"
AAAM7IAAEAAAAGkAAE	MONITOR SAMSUNF 19"
AAAM7IAAEAAAAGkAAF	MONITOR SAMSUNG 21"

Formato del ROWID de una fila.

EL *ROWID* de cada fila no se almacena físicamente como las columnas de la tabla, pero su valor debe almacenarse como un dato más en ciertos casos como en el de los índices o el de una fila almacenada en varios trozos. Un *ROWID* extendido almacenado ocupa 10 bytes, y se guarda codificado en base 64.

Un *ROWID* está formado por 18 caracteres con el siguiente significado:

- El **número de objeto**, los seis primeros caracteres, que identifica el segmento dentro de todos los de la base de datos.
- El **número de fichero de datos**, los tres caracteres que siguen, que identifica al fichero en el que se encuentra el bloque de datos que contiene la fila. Este número es relativo al *tablespace*.
- El **número de bloque**, los seis caracteres siguientes, que identifica al bloque de datos que contienen la fila dentro de todos los bloques del fichero.
- El **número de fila**, los últimos tres caracteres, que identifican la fila dentro del bloque.

Oracle permite consultar el *ROWID* de cualquier fila mediante lo que se denomina la **pseudocolumna ROWID**. Si en una consulta a cualquier tabla se incluye esta pseudocolumna, se obtiene el *ROWID* de cada fila recuperada. Si sobre la tabla ALMACEN realizamos la consulta SQL siguiente.

```
SELECT ROWID, DENOMINACION FROM ALMACEN
```

obtendríamos el resultado mostrado en la figura.

El *ROWID* de una fila se mantiene invariable durante el tiempo de existencia de la misma.

Al almacenar una fila, el SGBD graba los valores de cada columna en el mismo orden en que se definieron las columnas al crear la tabla, excepto en el caso de columnas de tipo LONG que van siempre en último lugar. El valor de cada columna va precedido de un campo donde se indica la longitud de la columna en bytes. Oracle soporta columnas de longitud fija y de longitud variable. Las columnas de longitud variable tienen siempre el tamaño justo para almacenar el dato que contengan, por lo que dicho tamaño puede variar cuando se actualiza el dato. Los nulos no se almacenan, de forma que las columnas que contienen nulos solo llevan el campo de longitud con un valor de cero. Si una o más columnas con nulos ocupan los últimos lugares de la fila ni siquiera se almacena la longitud. Cada fila va precedida de un campo de cabecera con información sobre el formato de la fila.

Cuando el tamaño de una fila es tal que no cabe en un bloque el SGBD la parte en dos o más trozos que almacena en distintos bloques. Cada trozo se identifica por su *ROWID* y los distintos trozos quedan lógicamente unidos al almacenar en la cabecera de cada uno el *ROWID* del siguiente.

Al modificar una fila que contenga columnas de longitud variable o con nulos, su tamaño puede aumentar. Si el espacio libre en el bloque no es suficiente para acomodar su nuevo tamaño, la fila se traslada, aunque, para que su *ROWID* no cambie, Oracle dejará en el bloque original un trozo de la fila con un enlace a la nueva dirección física.

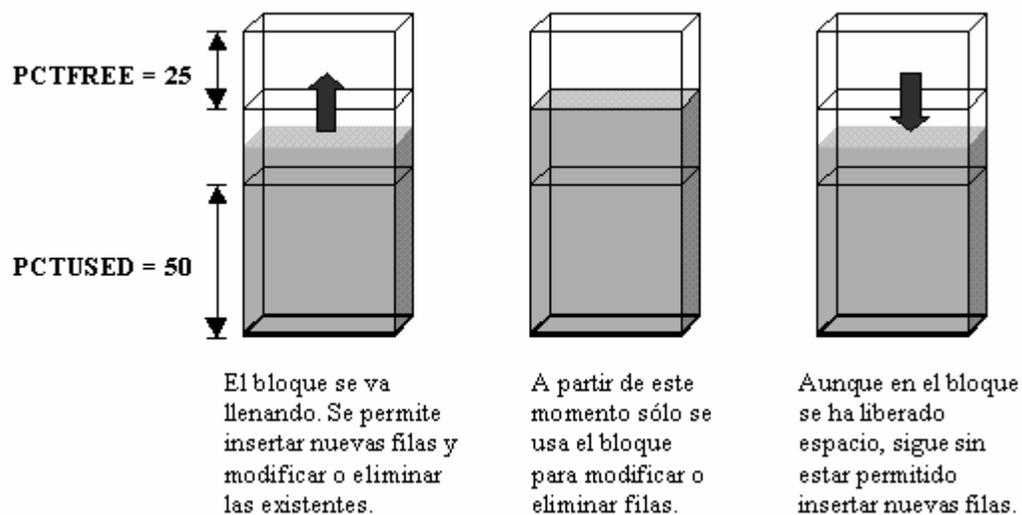
Cuando una fila fue trasladada a otro bloque, el rendimiento del sistema en los accesos a esa fila decrece ya que para localizarla es preciso leer dos bloques; el original, para saber la nueva dirección de la fila, y el nuevo. Y esta doble lectura puede, en algunos casos, necesitar dos accesos a disco. Para evitar en lo posible este problema, ORACLE permite configurar en cada bloque una reserva de espacio utilizable sólo para operaciones de actualización.

En una base de datos Oracle existen dos parámetros, asociados a toda tabla, índice o *cluster*, denominados *PCTFREE* y *PCTUSED*. Estos parámetros tienen unos valores específicos para cada tabla, índice o *cluster*, que se asignan en el momento de creación del correspondiente objeto, y que, también, pueden ser modificados posteriormente.

El valor de *PCTFREE* marca el porcentaje de espacio del bloque que se reservará para permitir realizar modificaciones en las filas del bloque. Mientras el porcentaje de bloque libre sea superior al valor de este parámetro el SGBD utilizará el espacio libre del bloque tanto para insertar nuevos registros como para modificar los ya existentes. En el momento en el que el porcentaje de espacio libre en el bloque sea igual o inferior al valor de *PCTFREE* ya no se insertarán nuevas filas en él, utilizándose sólo para acomodar los posibles aumentos de tamaño de las filas que se modifiquen. Posteriormente, como consecuencia de borrado de filas o modificaciones que acorten la longitud de la fila, puede aumentar el espacio libre del bloque, pero aunque éste vuelva a ser superior a *PCTFREE*, no se permitirán inserciones hasta que el porcentaje de bloque ocupado sea igual o inferior al valor de *PCTUSED*. En la figura tenemos un ejemplo de uso del espacio de un bloque de acuerdo con los valores de estos parámetros.

□ **Índices.**

Los índices son estructuras opcionales asociadas a tablas o *clusters*, que permiten acelerar el acceso a los datos de estos objetos. La existencia o inexistencia de un índice es totalmente transparente a los usuarios y a las aplicaciones que hacen uso de los datos, pero el



Significado de los parámetros *PCTFREE* y *PCTUSED*.

SGBD utiliza siempre los índices existentes.

Los índices son lógicamente y físicamente independientes de los datos de la tabla a la que se asocian. Se puede crear o eliminar en cualquier momento un índice sin afectar a la tabla base ni a otros índices. Los índices, como estructuras independientes que son, necesitan espacio de almacenamiento.

Un índice puede estar definido sobre una o más columnas de una tabla. Una tabla puede tener más de un índice. Por cada fila de la tabla el índice almacenará el valor de la columna o columnas indexadas junto al *ROWID* de la fila. De esta forma, el SGBD leerá el índice para, a

partir de valor de la columna o columnas indexadas, encontrar la dirección física de la fila a la que se desea acceder.

Así como el rendimiento de las consultas a una tabla puede aumentar por la presencia de índices, el rendimiento de las operaciones de modificación, eliminación e inserción de filas puede disminuir debido a que el SGBD debe actualizar los índices después de ejecutar cualquiera de ellas. Por tanto, se debe limitar la definición de índices a aquellos casos en que realmente se consiga una mejora en el rendimiento de las operaciones de consulta. Así, es adecuado definir índices en las columnas cuyos valores se utilizarán para buscar filas específicas de la tabla o para acceder en orden a dichas filas. El SGBD Oracle crea automáticamente un índice asociado a cada columna definida como clave primaria o alternativa.

El SGBD asigna un segmento a cada índice para almacenar sus datos. El *tablespace* del segmento, así como la asignación de extents, se rige por los mismos mecanismos vistos al tratar las tablas.

□ **Clusters.**

Un *cluster* es un método alternativo de almacenar tablas con columnas comunes y que se usan juntas con frecuencia. Cuando se crea un *cluster* el SGBD le asigna un segmento y las tablas del *cluster* comparten los bloques de este segmento para almacenar sus datos.

Las columnas comunes de las tablas del *cluster* constituyen la **clave del cluster** (*cluster key*, en inglés). El SGBD almacena físicamente juntas, en el mismo bloque, las filas de las tablas con el mismo valor de clave del *cluster*. Con esto se consigue mejorar el tiempo de acceso a los datos cuando se realizan reuniones naturales de las tablas del *cluster* por las columnas que constituyen la clave, ya que las filas relacionadas están almacenadas juntas. También, cada valor de clave se almacena una sola vez, independientemente del número de filas que contengan dicho valor, lo que lleva a un ahorro de espacio de almacenamiento.

Como ocurría con los índices, la existencia de *clusters* es transparente a los usuarios y a las aplicaciones.

Cuando se crea un *cluster* el SGBD le asigna un segmento del *tablespace* por defecto asociado al usuario que lo crea o del *tablespace* indicado en la sentencia de creación. La asignación de *extents* y el uso de los bloques funciona exactamente igual a lo visto para las tablas.

Para poder hacer uso de un *cluster* el SGBD Oracle obliga a definir un índice sobre su clave.

7. Estructuras de memoria y procesos.

Al arrancar un SGBD Oracle, u *Oracle Server*, en cualquier máquina se asigna un área de memoria denominada *System Global Area (SGA)* y se inician varios procesos conocidos como procesos *background (background processes)*. El conjunto de los procesos *background* y la SGA constituye una **instancia Oracle**. La memoria y los procesos de una instancia gestionan los datos de la base y facilitan el acceso de los usuarios a los mismos.

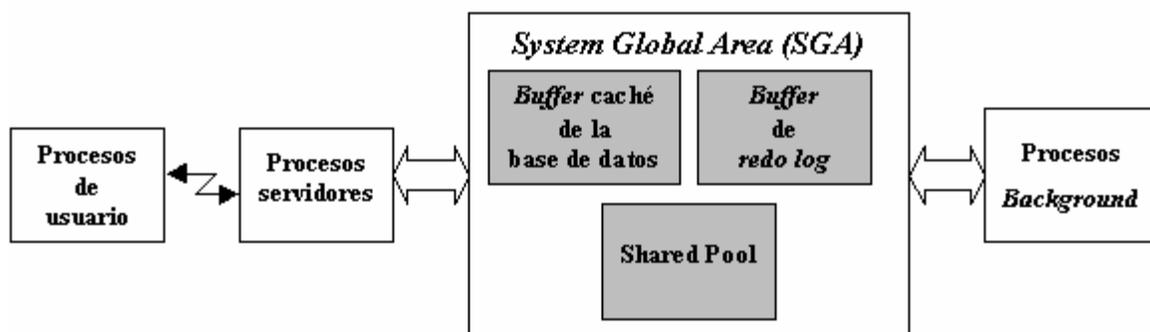
Es posible la existencia de varias instancias ejecutándose en la misma máquina. En este caso, cada instancia accede a su propia base de datos física, es decir, cada una utiliza un conjunto distinto de ficheros de datos, control y *redo log*. También, en sistemas que trabajan en paralelo, es posible la existencia de múltiples instancias accediendo a una sola base de datos.

Además de los procesos *background*, existen otro tipo de procesos Oracle denominados **procesos servidores (server processes)**. Oracle crea procesos servidores para atender las peticiones de los procesos de usuario, es decir, de los procesos que ejecutan el código de los programas de aplicación. Cualquier operación que desee realizar un proceso de usuario sobre los datos de la base será gestionada por un proceso servidor. Cuando el SGBD arranca un proceso servidor le asigna un buffer en memoria, denominado *Program Global Area (PGA)*, para contener los datos y la información de control.

Cuando el proceso de usuario y el proceso servidor que lo atiende residen en distintas máquinas de una red es necesaria la ejecución de SQL*Net tanto en la máquina servidor como en la cliente. SQL*Net es el *software* de Oracle que actúa de interfaz con el protocolo de comunicación usado por las red.

□ System Global Area (SGA).

La SGA es un área compartida de la memoria del ordenador en el que se ejecuta el SGBD, que contiene datos e información de control de una instancia. Cada instancia tiene su



Estructuras de memoria y procesos.

propia SGA.

Los usuarios conectados a un *Oracle server* comparten la información contenida en las distintas estructuras que componen la SGA. En la figura tenemos representadas las tres subdivisiones de la SGA: el *buffer* caché de la base de datos, el *buffer* de *redo log* y el *shared pool*.

Para mejorar la velocidad de acceso a los datos de la base el SGBD gestiona un sistema de caché, manteniendo en memoria, en el **buffer del caché de la base de datos**, aquellos bloques de datos más recientemente usados. Al estar estos bloques en memoria disminuye la necesidad de accesos a disco, ya que los bloques sobre los que se ha realizado recientemente alguna operación son los que con mayor probabilidad intentarán acceder las próximas operaciones. Algunos de estos bloques pueden contener datos modificados que aun no han sido grabados en disco.

El *buffer* de *redo log* mantiene información sobre los cambios realizados a la base de datos. Los procesos servidores se encargan de grabar la información de *redo log* en este *buffer* y un proceso *background* grabará el contenido del *buffer* en un fichero de *redo log*. Como sabemos, esta información es fundamental en la gestión de las transacciones y en la recuperación de la base de datos ante fallos del sistema.

El **shared Pool** es la parte de la SGA que mantiene otras estructuras como las **áreas de SQL compartidas** y el **caché del diccionario**.

Para ejecutar cada sentencia SQL de una aplicación de usuario el SGBD necesita un **área SQL compartida**. Esta área contendrá la sentencia compilada y su plan de ejecución. Una misma área SQL compartida es utilizada por múltiples aplicaciones que ejecutan la misma sentencia, con lo que además de ahorrar memoria se mejora el tiempo de ejecución de las sentencias, ya que no es preciso realizar la compilación y la optimización de cada una de ellas. Por cada ejecución de una de estas sentencias que comparten la misma área SQL se mantiene un **área SQL privada** que contiene la información propia de la ejecución concreta. El área SQL privada se mantiene en el *Program Global Area* (PGA), aunque, en ciertos casos, puede estar la SGA. Las aplicaciones pueden manejar las áreas SQL privadas mediante la definición de **cursores**. Como estudiaremos en una unidad posterior, un **cursor** permite controlar la ejecución de una sentencia SQL desde un programa de aplicación.

El **caché del diccionario** es la estructura del *shared pool* que contiene los bloques de datos del catálogo de la base más recientemente accedidos. Como sabemos, el catálogo o diccionario activo de la base de datos es accedido continuamente por el SGBD para ejecutar las sentencias SQL, por lo que la existencia de un caché mejora el rendimiento del sistema.

□ **Los procesos background.**

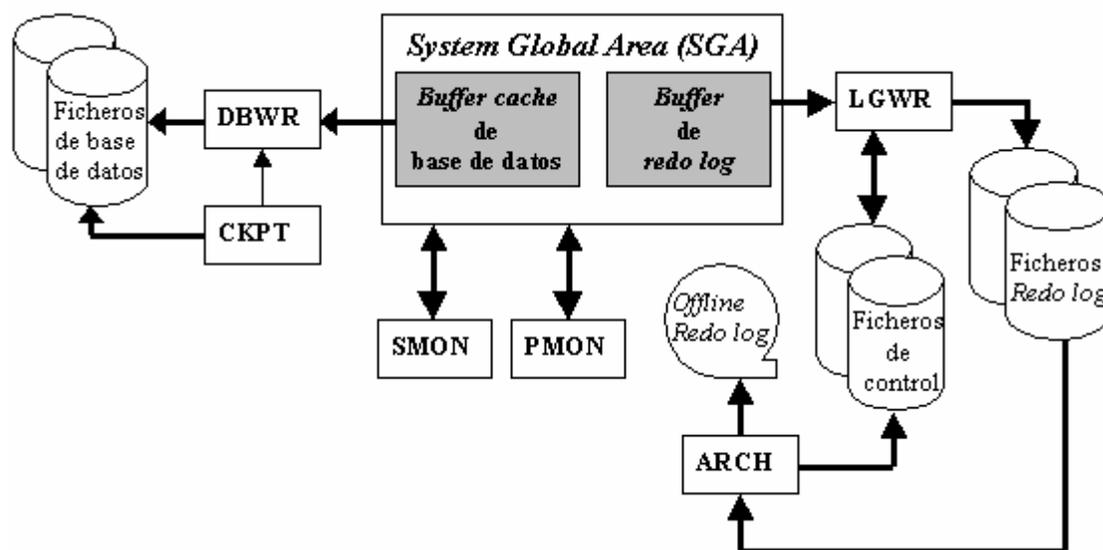
Nada más arrancar una instancia, Oracle crea un conjunto de procesos *background*. Estos procesos se ejecutan en segundo plano (en *background*) mientras existe la instancia, realizando tareas de entrada/salida de datos y de monitorización de otros procesos para conseguir una mejora en el rendimiento y fiabilidad del sistema. Los principales procesos *background* son los siguientes:

El **Database Writer (DBWR)** escribe en los ficheros de datos los bloques modificados del *buffer* caché de la base de datos. El DBWR está optimizado para minimizar las operaciones de escritura en disco. En general, escribe en el disco cuando se precisa introducir nuevos nuevos bloques en el caché y no hay espacio. Los bloques del *buffer* caché que hace más tiempo que no fueron accedidos son los primeros en ser grabados en disco.

El **log Writer (LGWR)** graba la información de *redo log* contenida en *buffer* de *redo log* en el fichero de *redo log* activo. Las escrituras en disco tienen lugar cuando termina una transacción o cuando se llena el *buffer*.

El **Checkpoint (CKPT)** genera, en instantes determinados, un punto de verificación (*checkpoint*) haciendo que el DBWR grabe en los ficheros de datos todos los bloques modificados y no grabados. También, registra en los ficheros de datos y de control el *checkpoint*. CKPT es opcional; si no existe, su labor es realizada por LGWR.

El **Archiver (ARCH)** se ocupa de copiar los ficheros de *redo log* en un dispositivo de archivo. La copia se produce cuando un fichero se ha llenado y deja de estar activo. Este proceso sólo existe en aquellas bases de datos funcionando en modo *ARCHIVELOG*.



Los procesos background.

El **System Monitor (SMON)** realiza la recuperación de la base de datos al arrancar la instancia. Si es necesario, al comenzar la ejecución de la instancia, SMON realiza la recuperación de la base de datos a partir de la información sobre el último punto de verificación registrado en el fichero de control y de la información de *redo log*. Este proceso se encarga también de desasignar segmentos temporales que ya no están en uso, así como de reunir *extents* no usados para conseguir que el espacio libre dentro de los *tablespaces* sea contiguo y más fácil de asignar.

El **Process Monitor (PMON)** se ocupa de la recuperación de procesos cuando se produce el fallo de un proceso de usuario. PMON es responsable de limpiar el caché y de liberar los recursos que el proceso estuviera usando.

8. El 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.

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.

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.

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 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.

La mayoría de las vistas del catálogo aparecen en conjuntos de tres que contienen similar información y que se diferencian entre sí por el prefijo con que se nombran. Por ejemplo, las vistas `USER_TABLES`, `ALL_TABLES` y `DBA_TABLES` contienen 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.

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 Database Reference*.

Vistas	Descripción
DICTIONARY	Descripción de las tablas 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.
DBA, ALL, USER_VIEWS	Definiciones de vistas.
DBA, ALL, USER_TAB_COLUMNS	Descripción de las columnas de tablas, vistas y <i>clusters</i> .
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.
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 <i>clusters</i> .
DBA, ALL, USER_IND_COLUMNS	Columnas de los índices definidos.
DBA, ALL, USER_SEQUENCES	Información sobre secuencias.

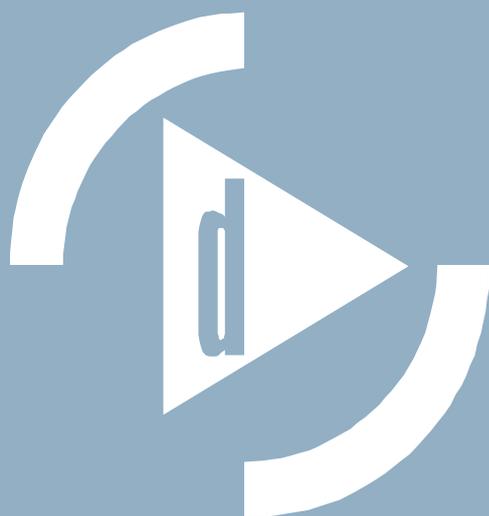
DBA, ALL, USER_SYNONYMS	Información sobre sinónimos existentes en la base.
DBA_CLUSTERS, USER_CLUSTERS	Descripción de los <i>clusters</i> definidos.
DBA_CLU_COLUMNS, USER_CLU_COLUMNS	Correspondencias entre columnas de la clave del <i>cluster</i> y columnas de las tablas agrupadas en el <i>cluster</i> .
DBA_DATA_FILES	Información sobre los ficheros de datos.
DBA, USER_TABLESPACES	Descripción de los <i>tablespaces</i> .
DBA, USER_SEGMENTS	Información sobre segmentos.
DBA_ROLLBACK_SEGS	Descripción de los segmentos de <i>rollback</i> .
DBA, USER_EXTENTS	<i>Extents</i> de los segmentos.
DBA, USER_FREE_SPACE	<i>Extents</i> no asignados de los <i>tablespaces</i> .

El catálogo contiene también otro tipo de tablas conocido como **tablas de rendimiento dinámico** (*Dynamic Performance Tables*). Estas tablas son continuamente actualizadas por SGBD Oracle durante su funcionamiento, almacenando en ellas información relativa al rendimiento del sistema.

Realmente, las tablas de rendimiento dinámico no son auténticas tablas aunque pueden ser consultadas como si lo fueran. Sobre ellas hay definidas vistas cuyo nombre empieza siempre por el prefijo **V_**\$. Los usuarios acceden a estas vistas a través de sinónimos públicos que comienzan siempre con **V\$**.

Desarrollo de Aplicaciones Informáticas

materiales didácticos de aula



UNIÓN EUROPEA
Fondo Social Europeo



Gobierno del Principado de Asturias
Consejería de Educación y Ciencia



FORMACIÓN PROFESIONAL
Principado de Asturias