

En la Figura 3.2 se muestran, para el caso concreto de Oracle, las distintas estructuras lógicas y físicas empleadas en cada uno de los niveles generales de la arquitectura de un SGBD [13]. Estas estructuras tienen la finalidad de mejorar el rendimiento del sistema en el acceso a los datos. A lo largo del capítulo iremos explicando en profundidad cada una de ellas.

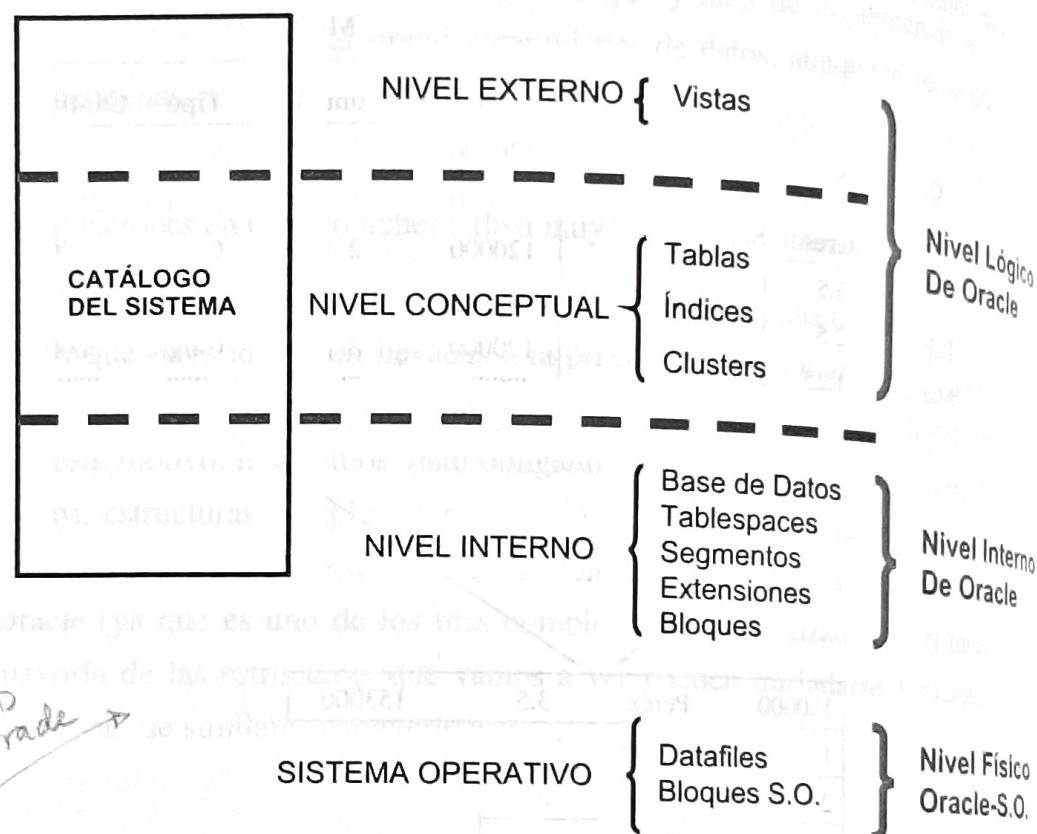


FIGURA 3.2 Estructura por niveles en Oracle.

### 3.2. EL DICCIONARIO DE DATOS O CATÁLOGO

Como ya sabemos, los SGBD deben almacenar y mantener gran cantidad de información para garantizar su propio funcionamiento (al margen de los datos de usuarios y aplicaciones); tal es el caso, por ejemplo, de las correspondencias entre niveles, los usuarios y sus privilegios, las definiciones de tablas e índices... Para ello, los sistemas utilizan un repositorio en el que toda esta información es organizada y jerarquizada convenientemente y que se denomina **diccionario de datos o catálogo**.

El catálogo es uno de los elementos más importantes de una BD y consiste en un conjunto de tablas y vistas orientadas, fundamentalmente, a la consulta, que ofrece información acerca de la BD a la que está asociado (en términos más técnicos se dice que contiene *metainformación*). Concretando un poco más, el diccionario de datos contiene:

- Definiciones de objetos. Entre los objetos que se contemplan se encuentran:
  - *Tablas*: Para las que se almacenan los atributos y sus correspondientes tipos de datos, las restricciones de integridad, propietario, fecha de creación...
  - *Vistas*: Para las que se almacena su nombre, consulta que permite construirla, propietario...
  - *Índices*: Para los que se almacena nombre, tabla y atributos sobre los que está construido, tipo...

La lista anterior no pretende ser exhaustiva, puesto que son objetos también procedimientos, funciones, disparadores y clústers.

- Espacios reservados para cada uno de los objetos mencionados y estructuras del nivel interno empleadas en su almacenamiento.
- Restricciones de integridad asociadas tanto a los atributos como a las tablas.
- Nombres de los usuarios, contraseñas, fecha de alta, cuotas de disco...
- Definición de privilegios y roles, mediante los cuales se puede restringir el uso que hacen los usuarios de los recursos del sistema. En el Capítulo 4 hablaremos ampliamente de este tema.
- Información de auditoría. Para el seguimiento de los usuarios y para la mejora continua del rendimiento del sistema, en las tablas de auditoría del diccionario se registran todas las operaciones realizadas, a qué objetos han afectado, la hora, qué usuario la ha llevado a cabo, etc. Dado el consumo en tiempo y recursos de este seguimiento, lo normal es que pueda habilitarse opcionalmente.

- Información general de la BD, como la ubicación de los datos en los tablespaces<sup>1</sup>, nombre y ubicación de los ficheros de control, copias de seguridad realizadas, estadísticas, etc.

El catálogo de la BD, al estar estructurado en tablas y vistas, se consulta como si de cualquier otra BD se tratara. Todas sus tablas y vistas están almacenadas en un compartimento específico denominado *tablespace SYSTEM*. El diccionario no es útil solo para el propio SGBD sino que constituye una herramienta esencial para todos los usuarios del sistema, desde los usuarios menos especializados hasta el administrador (sys), pasando por los programadores de aplicaciones, ya que suministra cualquier tipo de información sobre el sistema de forma selectiva y organizada. El acceso al diccionario se lleva a cabo a través de sentencias estándar de consulta (SELECT en el caso de SQL), ya que se trata de tablas de solo lectura.

◊ *El diccionario de la BD se organiza según dos tipos de estructuras:*

- **Tablas base:** Son las tablas que almacenan toda la información sobre los datos y la estructura de la BD que tienen asociada. Sólo el propio SGBD y el administrador (sys) pueden escribir en dichas tablas y la mayor parte de su contenido es información de acceso restringido a cada tipo de usuario.
- **Vistas:** Sirven para resumir y visualizar cómodamente la información contenida en las tablas base del diccionario. Estas vistas organizan la información de dichas tablas y la convierten en información útil y más comprensible (nombres de tablas, de usuarios, de privilegios, etc.) llevando a cabo las reuniones y selecciones necesarias para conseguirlo. La mayoría de los usuarios solo tienen acceso a las vistas.

En Oracle, cada una de las vistas disponibles tiene tres versiones, que se distinguen por su prefijo y por la cantidad de información que ofrecen.

*Los prefijos utilizados son los siguientes:*

- **USER:** Las vistas que llevan este prefijo, ofrecen información que se refiere

<sup>1</sup>El tablespace es una estructura lógica de almacenamiento que describiremos ampliamente en el Apartado 3.3.1.

a los objetos que son propiedad del usuario que realiza la consulta. Por ejemplo, para obtener información sobre los objetos de tipo tabla, realizaríamos la siguiente consulta al catálogo:

```
SELECT * FROM user_tables;
```

- **ALL**: Las vistas que llevan este prefijo, ofrecen información sobre todos los objetos accesibles para el usuario que hace la consulta, sea o no el propietario. La consulta correspondiente sería:

```
SELECT * FROM all_tables;
```

- **DBA**: Las vistas que llevan este prefijo, ofrecen todas las tuplas y todos los atributos de la tabla base subyacente. Se necesitan privilegios especiales para su acceso y la consulta a realizar sería:

```
SELECT * FROM dba_tables;
```

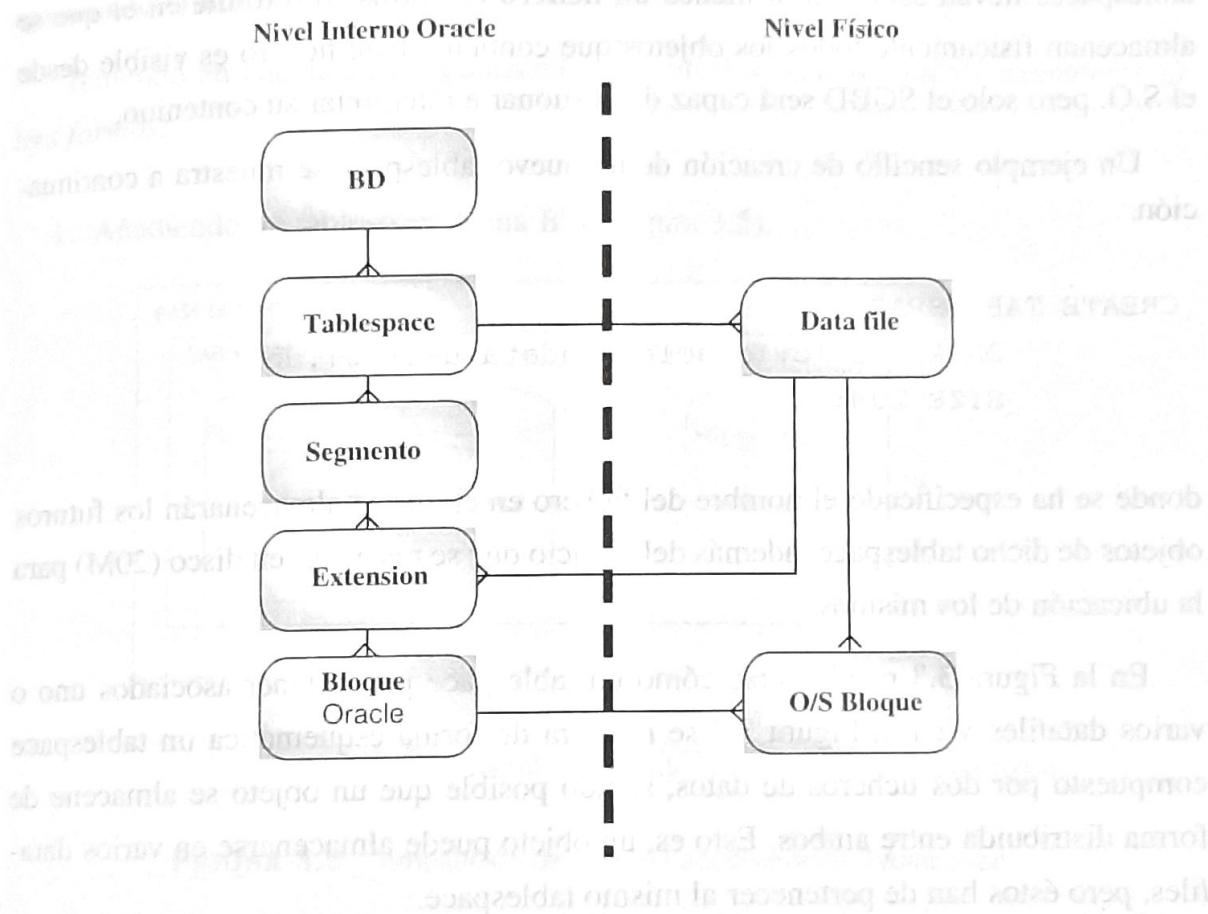
El propietario de todas las tablas y vistas del diccionario es un usuario especial (SYS en Oracle) y, por tanto, ningún otro usuario deberá estar autorizado a modificar, borrar o insertar información en dichas tablas, ya que se puede comprometer peligrosamente la integridad de toda la BD. De hecho, el SGBD es el único que debería escribir en él.

◊ *El diccionario tiene tres utilidades básicas:*

1. Organizar toda la información de usuarios, objetos y esquemas de almacenamiento.
2. Mantener siempre actualizada dicha información. De hecho, el SGBD lo modifica cada vez que se ejecuta una sentencia del DDL y los cambios son reflejados y percibidos por los usuarios de inmediato.
3. Permitir acceso de solo lectura a cualquier usuario autorizado para obtener información de la BD. En este sentido, el catálogo sirve como repositorio de información a todos los niveles: externo, conceptual e interno.

Por ejemplo, si un usuario user1 crea una tabla emp, se añadirán en el diccionario las tuplas necesarias para reflejar la existencia de esta nueva tabla: su nombre, su propietario, las columnas y sus tipos, estructuras de almacenamiento donde se alojará, restricciones de integridad, etc. A continuación mostramos parte de los esquemas

En la Figura 3.3 puede apreciarse la correspondencia que existe entre las estructuras empleadas en el nivel interno y las empleadas a nivel físico (S.O. y dispositivos de almacenamiento).



**FIGURA 3.3** Correspondencia interna-física en Oracle.

Pasamos a describir cada una de estas estructuras con mayor detalle.

### 3.3.1. TABLESPACES Y FICHEROS DE DATOS

Los **tablespaces**<sup>4</sup> o espacios de tablas son estructuras lógicas del nivel interno que nos permiten organizar y almacenar objetos de la BD relacionados entre sí (por su contenido, su función, su propietario...), en diferentes compartimentos. El concepto de tablespace es, en cierto modo, similar al de directorio en una unidad de

<sup>4</sup>A lo largo del capítulo utilizaremos el término en inglés.

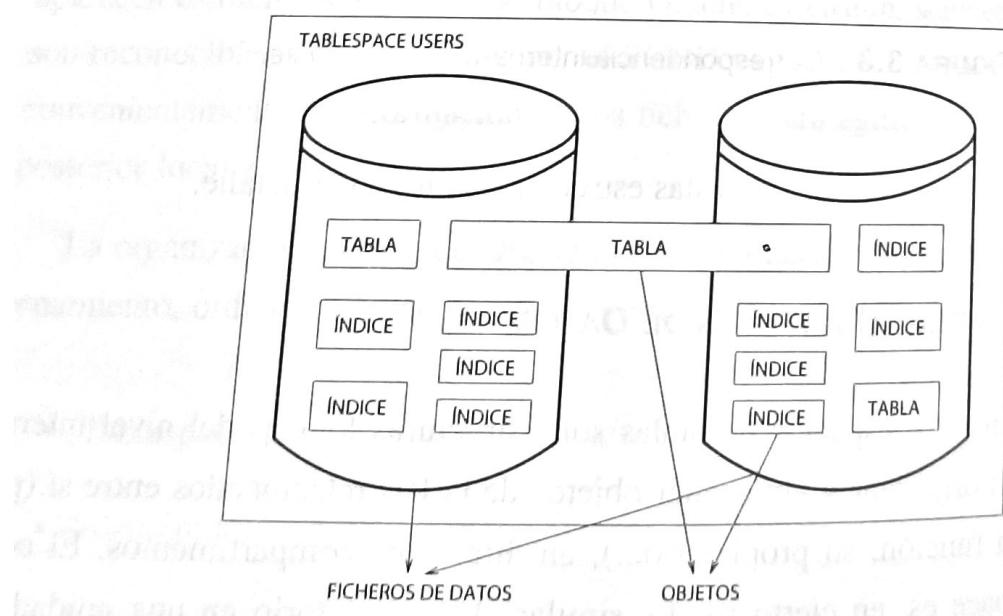
almacenamiento, con la salvedad de que el tablespace no es visible desde el S.O. En este sentido, en una BD suele haber siempre varios tablespaces, de la misma manera que fraccionamos nuestras unidades en varios directorios. Sin embargo, todos los tablespaces llevan asociado al menos un **fichero de datos** o **datafile** en el que se almacenan físicamente todos los objetos que contiene. Este fichero es visible desde el S.O. pero solo el SGBD será capaz de gestionar e interpretar su contenido.

Un ejemplo sencillo de creación de un nuevo tablespace se muestra a continuación:

```
CREATE TABLESPACE users
  DATAFILE 'c:\oracle\oradata\users01.dbf'
  SIZE 20M;
```

donde se ha especificado el nombre del fichero en el que se almacenarán los futuros objetos de dicho tablespace, además del espacio que se reservará en disco (20M) para la ubicación de los mismos.

En la Figura 3.3 puede verse cómo un tablespace puede tener asociados uno o varios datafiles y en la Figura 3.4 se muestra de forma esquemática un tablespace compuesto por dos ficheros de datos, siendo posible que un objeto se almacene de forma distribuida entre ambos. Esto es, un objeto puede almacenarse en varios datafiles, pero éstos han de pertenecer al mismo tablespace.

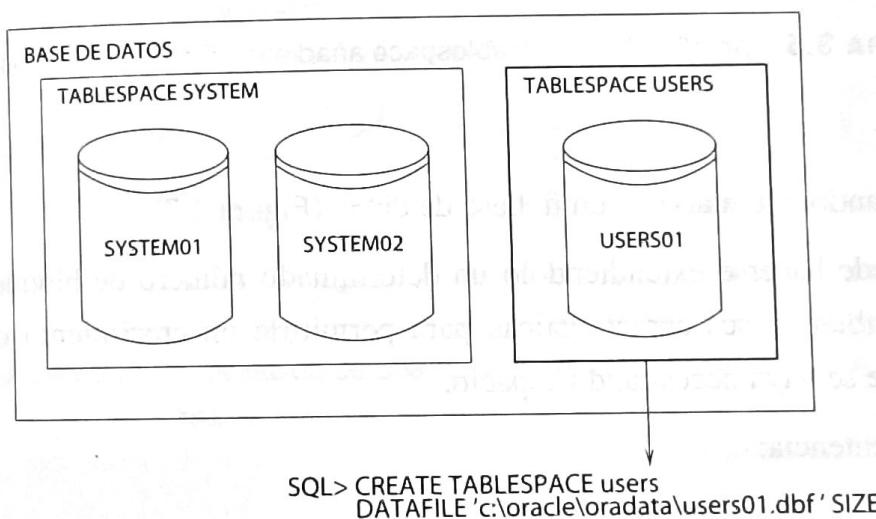


**FIGURA 3.4** Organización de tablespaces y ficheros de datos.

Una BD consiste, pues, en una o más unidades lógicas de almacenamiento llamadas tablespaces, que se traducirán en uno o más ficheros de datos o datafiles, que son estructuras físicas compatibles con las del sistema operativo sobre el que se esté ejecutando el SGBD.

Teniendo en cuenta esta organización por niveles, *una BD puede extenderse de tres formas:*

1. Añadiendo un tablespace a una BD (Figura 3.5).



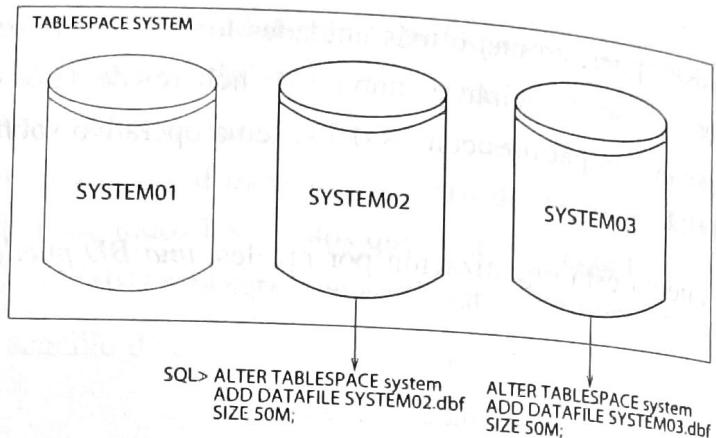
**FIGURA 3.5** Ampliación de una BD añadiendo un tablespace.

Con esta operación se incrementa el tamaño total de la BD. Esta opción es útil cuando la estructura global de la BD deba ser ampliada para dar cabida a un nuevo grupo de objetos.

2. Añadiendo un fichero de datos a un tablespace existente (Figura 3.6).

Con esta operación se incrementa la cantidad de espacio en disco reservada para el correspondiente tablespace. Esta opción se recomienda cuando la unidad en la que estamos guardando nuestros datos está llegando al límite de su capacidad. La sentencia que nos permite llevar a cabo esta operación es la siguiente:

```
ALTER TABLESPACE system ADD DATAFILE  
'c:\oracle\oradata\system02.dbf' SIZE 10M;
```



**FIGURA 3.6** Ampliación de un tablespace añadiendo ficheros de datos.

### 3. Aumentando el tamaño de un fichero de datos (Figura 3.7).

Esto puede hacerse extendiéndolo un determinado número de bloques fijo, o bien cambiando sus características para permitirle un crecimiento dinámico conforme se vaya necesitando espacio.

Con la sentencia:

```
ALTER DATABASE DATAFILE
  'c:\oracle\oradata\users01.dbf' RESIZE 15M;
```

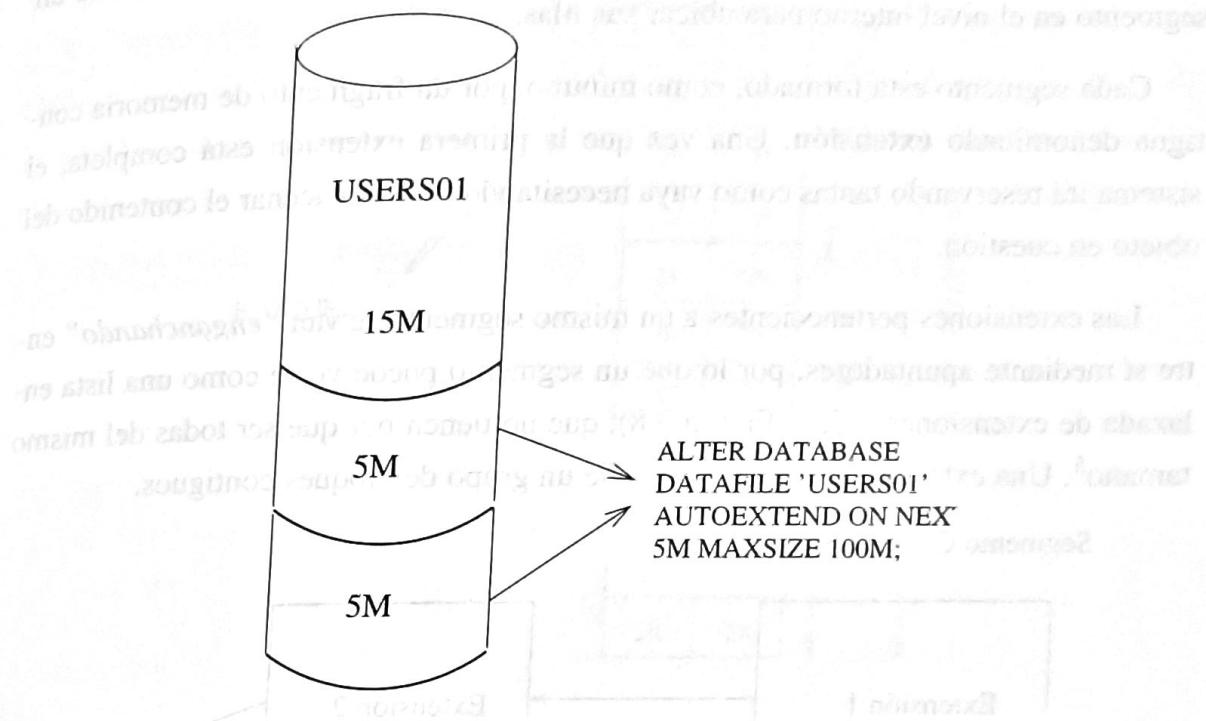
ampliamos el espacio reservado inicialmente al tamaño especificado en la cláusula **RESIZE**, mientras que con la sentencia:

```
ALTER DATABASE DATAFILE
  'c:\oracle\oradata\users01.dbf' AUTOEXTEND ON
    NEXT 5M MAXSIZE 100M;
```

estamos permitiendo el crecimiento dinámico del fichero (**AUTOEXTEND ON**) mediante sucesivos incrementos de 5M (**NEXT**) hasta alcanzar un tamaño máximo de 100M (**MAXSIZE**).

El tamaño de un tablespace es la suma de los tamaños de los ficheros de datos (datafiles) que lo constituyen. El tamaño de la BD es la suma de los tablespaces que la constituyen.

### TABLESPACE USERS



**FIGURA 3.7** Ampliación de una BD añadiendo bloques a uno de sus ficheros.

En la mayoría de los SGBD relacionales existe un tablespace especial (en Oracle llamado **SYSTEM**) que contiene, como mínimo, todas las tablas del diccionario de la BD. Este tablespace se crea automáticamente cuando se instala la BD. Se recomienda crear al menos un tablespace adicional para almacenar los datos de los usuarios y no mezclar información de distinta naturaleza.

Dado que los tablespaces contienen datos de diferentes objetos, éstos se dividen en unidades lógicas de almacenamiento de menor tamaño llamadas *segmentos*, con el fin de separarlos convenientemente y facilitar así su recuperación. Lo vemos en la siguiente sección.

#### 3.3.2. SEGMENTOS, EXTENSIONES Y BLOQUES

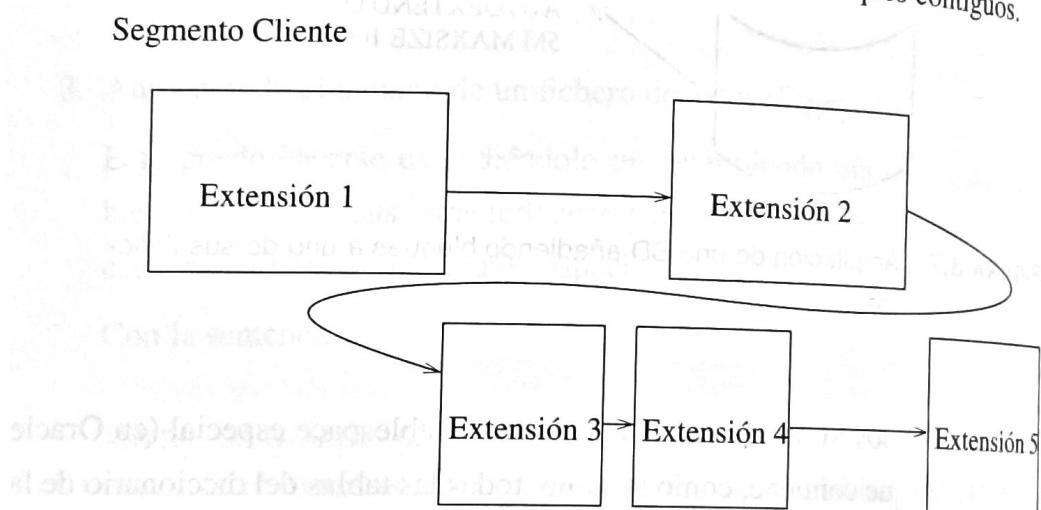
Éstos son los siguientes niveles de almacenamiento según la jerarquía que hemos visto en la Figura 3.3. Los **segmentos** almacenan datos de un mismo objeto (tabla, índice...) ayudando así a clasificar y ordenar la información contenida en un tablespace y permitiendo optimizar tanto su almacenamiento como su posterior localización.

Por ejemplo, cada vez que se crea una tabla desde el nivel conceptual, se crea un segmento en el nivel interno para ubicar sus filas.

Cada segmento está formado, como mínimo, por un fragmento de memoria contigua denominado **extensión**. Una vez que la primera extensión está completa, el sistema irá reservando tantas como vaya necesitando para almacenar el contenido del objeto en cuestión.

Las extensiones pertenecientes a un mismo segmento se van “enganchando” entre sí mediante apuntadores, por lo que un segmento puede verse como una lista enlazada de extensiones (véase Figura 3.8), que no tienen por qué ser todas del mismo tamaño<sup>5</sup>. Una extensión está compuesta de un grupo de bloques contiguos.

Segmento Cliente

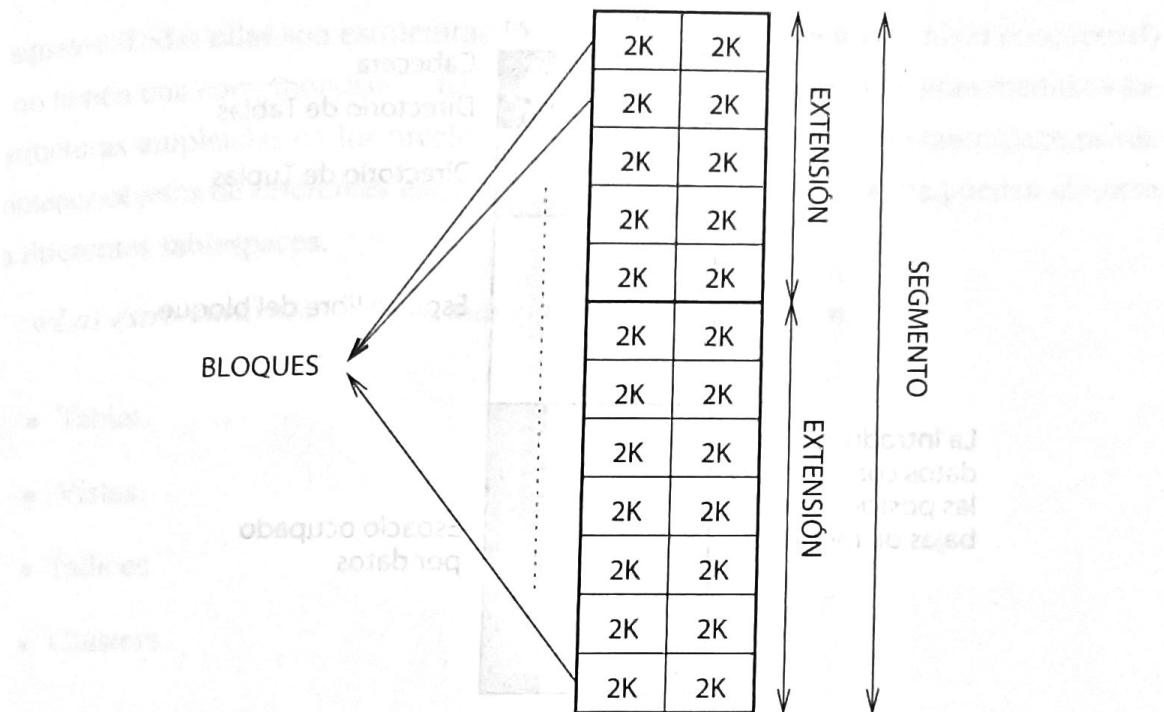


**FIGURA 3.8** Estructura de un segmento.

Por tanto, un segmento se compone de extensiones, y éstas se componen a su vez de **bloques**. El bloque es la unidad lógica de almacenamiento más pequeña y, por tanto, el nivel más bajo al que el SGBD puede administrar sus datos. Debe dimensionarse con un tamaño múltiplo del tamaño de bloque del S.O. para optimizar las operaciones de lectura/escritura en disco. La relación entre segmentos, extensiones y bloques puede verse gráficamente en la Figura 3.9.

En Oracle se distinguen cuatro tipos básicos de segmentos:

<sup>5</sup>En las sentencias *create tablespace* y *create table* existen cláusulas que permiten especificar los tamaños de las extensiones inicial y siguientes.



**FIGURA 3.9** Organización de segmentos, extensiones y bloques.

1. *Segmentos de datos.* Se generan durante un `create table` y contienen filas de tabla.
2. *Segmentos de índices.* Se generan durante un `create index` y contienen entradas de índice.
3. *Segmentos temporales.* Se generan durante operaciones como: `order by`, `group by`, `union`, `intersect`, `minus...` para las que se necesita espacio intermedio donde almacenar resultados parciales. Estos segmentos se destruyen automáticamente cuando el resultado final ya ha sido calculado.
4. *Segmentos de rollback.* Sirven para almacenar los valores antiguos de los datos antes de ser modificados por alguna operación, con el fin de poder recuperar la BD en caso de fallo o deshacer una transacción (rollback) no finalizada. Más detalles sobre el uso de este tipo de segmentos se estudiarán en el Capítulo 4.

Los bloques tienen una estructura idéntica independientemente de que contengan datos de una tabla, de un índice o del diccionario. Dicha estructura puede verse en la Figura 3.10 y consta de las siguientes partes:

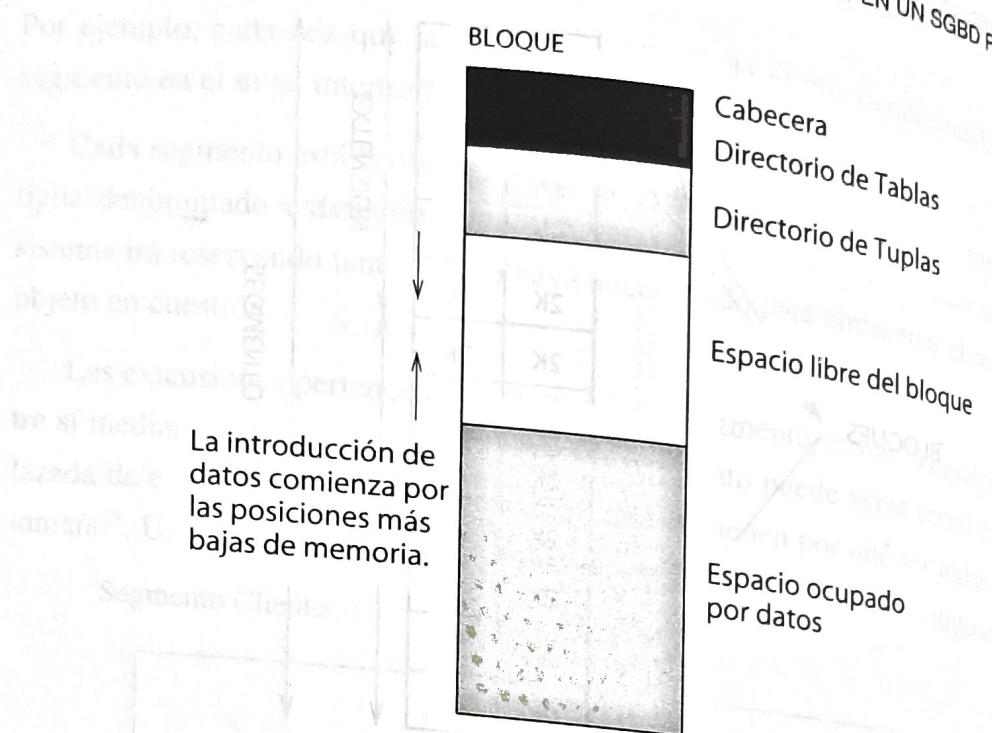


FIGURA 3.10 Estructura interna de un bloque.

- **Cabecera:** Contiene información general del bloque, como su dirección y el tipo de segmento al que pertenece (de datos, de índice, temporal o de rollback).
- **Directorio de Tablas:** Esta zona del bloque guarda información de las tablas que tienen filas en este bloque.
- **Directorio de Tuplas:** Esta zona del bloque guarda información acerca de las filas almacenadas en el bloque, incluyendo su dirección.
- **Zona de Datos:** Donde se almacena la información de la BD propiamente dicha. Se comienza a ocupar desde las posiciones más bajas de memoria.
- **Espacio Libre:** Se va ocupando, conforme se va necesitando, con información de la cabecera o con información de datos, indistintamente.

## ESTRUCTURA LÓGICA DE UN SGBD RELACIONAL

En todos los sistemas de BD cada usuario tiene asociado un esquema propio consistente en una colección de objetos: tablas, vistas, índices, clústers, procedimientos, etc.