

Sistemas y Aplicaciones  
Informáticas

Tema 36. Sistemas Gestores de  
Bases de Datos. Funciones.  
Componentes. Arquitectura de  
Referencia y Operacionales. Tipos de  
Sistemas.

<b>1. ÁMBITO DE DOCENCIA.</b> .....	<b>3</b>
<b>2. SISTEMAS GESTORES DE BASES DE DATOS.</b> .....	<b>3</b>
2.1. SISTEMAS DE INFORMACIÓN. ....	3
2.1.1. <i>Concepto. Componentes. Tipos.</i> .....	3
2.1.2. <i>Sistemas de ficheros. Características. Inconvenientes.</i> .....	3
2.2. BASES DE DATOS. ....	3
2.2.1. <i>Concepto. Características.</i> .....	3
2.2.2. <i>Ventajas. Funciones.</i> .....	4
2.2.3. <i>Modelos de datos. Esquemas. Diseño de bases de datos.</i> .....	4
2.3. SISTEMAS GESTORES DE BASES DE DATOS.....	5
2.3.1. <i>Concepto. Ventajas.</i> .....	5
2.3.2. <i>Lenguajes de definición y manipulación de datos. Funcionamiento.</i> .....	5
<b>3. FUNCIONES.</b> .....	<b>6</b>
<b>4. COMPONENTES.</b> .....	<b>7</b>
<b>5. ARQUITECTURA ANSI/SPARC.</b> .....	<b>8</b>
5.1. NIVELES. FUNCIONES DE TRADUCCIÓN. FASES DE LA ARQUITECTURA.....	8
5.2. CORRESPONDENCIAS ENTRE NIVELES. INDEPENDENCIA DE DATOS. ....	9
<b>6. ARQUITECTURAS OPERACIONALES.</b> .....	<b>9</b>
<b>7. TIPOS DE SISTEMAS.</b> .....	<b>10</b>

## 1. **Ámbito de docencia.**

- Sistemas informáticos monousuario y multiusuario (ASI 1).
- Sistemas informáticos multiusuario y en red (DAI 1).
- Instalación y mantenimiento de equipos y sistemas informáticos (ESI 1).

## 2. **Sistemas gestores de bases de datos.**

### 2.1. **Sistemas de información.**

#### 2.1.1. **Concepto. Componentes. Tipos.**

- Un sistema de información es un conjunto de elementos que gestiona la información de una determinada organización. Sus componentes son:
  - \* *Datos*. Información relevante que almacena y gestiona el sistema de información.
  - \* *Hardware*. Equipamiento físico que se utiliza para gestionar los datos.
  - \* *Software*. Aplicaciones que permiten el funcionamiento adecuado del sistema.
  - \* *Recursos humanos*. Personal que maneja el sistema de información.
- Existen dos tipos fundamentales de sistemas de información:
  - \* *Orientados al proceso*. En estos sistemas de información se crean diversas aplicaciones (software) para gestionar diferentes aspectos del sistema. Cada aplicación realiza unas determinadas operaciones, y almacena y utiliza sus propios datos.
  - \* *Orientados a los datos*. En esos sistemas los datos se almacenan en una única estructura lógica que es utilizable por las aplicaciones. A través de esa estructura se accede a los datos que son comunes a todas las aplicaciones.

#### 2.1.2. **Sistemas de ficheros. Características. Inconvenientes.**

- Los sistemas de ficheros son sistemas de información orientados al proceso que tienen las siguientes características:
  - \* Los ficheros se diseñan para una determinada aplicación.
  - \* Los datos se encuentran almacenados en soportes de almacenamiento secundario, mientras su descripción está separada de los mismos, formando parte de los programas.
  - \* No hay control sobre el acceso y la manipulación de los datos más allá de lo impuesto por los programas de aplicación.
- Los inconvenientes que se plantean son los siguientes:
  - \* Hay una ocupación inútil de memoria secundaria.
  - \* Suele aparecer un cierto grado de inconsistencia y duplicación en la información.
  - \* Falta de flexibilidad del sistema de ficheros para adaptarse a las nuevas necesidades.
  - \* Existe cierta dificultad para compartir información.

### 2.2. **Bases de datos.**

#### 2.2.1. **Concepto. Características.**

- Una base de datos es un conjunto estructurado de datos relacionados entre sí que reside en soportes de almacenamiento secundario de acceso directo.
- Las características que las diferencian de los sistemas de ficheros son las siguientes:
  - \* Además de los datos, se almacenan las relaciones entre ellos y sus restricciones semánticas.

- \* No debe existir redundancia lógica, aunque se admite redundancia física por eficiencia.
- \* Las bases de datos han de atender a múltiples usuarios y diferentes aplicaciones.
- \* Existe independencia tanto física como lógica entre datos y tratamientos.
- \* La definición y la descripción de los datos están integradas con los mismos datos.
- \* Incorporan procedimientos de actualización y recuperación que mantienen la integridad, seguridad y confidencialidad de los datos.

### 2.2.2. Ventajas. Funciones.

- Las ventajas de las bases de datos son las siguientes:
  - \* Reducción de la redundancia de una misma información para uso de distintas aplicaciones.
  - \* Mantenimiento de la consistencia de la información, evitando que exista información discrepante sobre un mismo y único hecho.
  - \* Compartir los mismos datos entre distintos usuarios y aplicaciones, gestionando el acceso concurrente de todas ellas a la información.
  - \* Distribución de los recursos existentes, en capacidad de almacenamiento y de procesamiento, entre las necesidades de los distintos usuarios y aplicaciones.
- Las principales funciones que desarrolla una base de datos son las siguientes:
  - \* Crear nuevas estructuras de datos que permitan el almacenamiento de nuevos datos, así como de las interrelaciones adecuadas entre los mismos.
  - \* Insertar nuevos datos sobre las estructuras ya creadas, al igual que la inserción de interrelaciones entre los datos introducidos en el sistema.
  - \* Extraer selectivamente la información mediante un lenguaje de consulta.
  - \* Actualizar o modificar las estructuras de datos y los contenidos de la base de datos.
  - \* Eliminar datos existentes en la base de datos manteniendo siempre su integridad.

### 2.2.3. Modelos de datos. Esquemas. Diseño de bases de datos.

- Un modelo de datos es una colección de conceptos para la descripción de los datos, las relaciones entre ellos y las restricciones que deben cumplir. Un esquema es una descripción de una base de datos mediante un modelo de datos. Los modelos de datos se clasifican en:
  - \* *Modelos conceptuales*. Describen los datos con un alto nivel de abstracción, utilizando entidades (concepto del mundo real), atributos (propiedad de interés de una entidad) y relaciones (interacción entre dos o más entidades). Son independientes de la base de datos a utilizar. Por ejemplo, el modelo entidad/relación y el modelo orientado a objetos.
  - \* *Modelos lógicos*. Representan los datos valiéndose de estructuras de registros de varios tipos, formados por un número determinado de campos. Son dependientes de la base de datos a utilizar. Por ejemplo, el modelo relacional, el modelo de red y el modelo jerárquico.
  - \* *Modelos físicos*. Los modelos físicos describen cómo se almacenan los datos en cuanto al formato de los registros, la estructura de los ficheros y los métodos de acceso utilizados.
- El diseño de bases de datos se estructura en tres pasos:
  - \* *Diseño conceptual*. Recibe como entrada la especificación de requerimientos y su resultado es el esquema conceptual, que es una descripción de alto nivel de la estructura de la base de datos mediante un modelo conceptual, y que es independiente del SGBD que se utilice.

- \* *Diseño lógico.* Recibe como entrada el esquema conceptual y da como resultado un esquema lógico, que es una descripción de la estructura de la base de datos mediante un modelo lógico, y que puede ser procesado por el SGBD que se utilice.
- \* *Diseño físico.* Recibe como entrada el esquema lógico y da como resultado un esquema físico, que es una descripción mediante un modelo físico de las estructuras de almacenamiento y de los métodos usados para tener un acceso efectivo a los datos.

### **2.3. Sistemas gestores de bases de datos.**

#### **2.3.1. Concepto. Ventajas.**

- Un sistema gestor de base de datos (SGBD) es un conjunto coordinado de programas, procedimientos y lenguajes que permite describir, recuperar y manipular la información almacenada en la base de datos, manteniendo su integridad, confidencialidad y seguridad.
- Las ventajas de los sistemas gestores de bases de datos son las siguientes:
  - \* Independencia de la representación de la información respecto a las aplicaciones que la utilizan. De esta forma, es posible modificar la estructura de almacenamiento de la información sin afectar a las aplicaciones que los utilizan, y que distintas aplicaciones utilicen distintas vistas de los datos.
  - \* Garantizar la seguridad de la información, controlando el acceso y la manipulación de la información por las distintas aplicaciones y usuarios.
  - \* Mejorar la integridad de los datos, expresada en restricciones que no pueden violarse. Estas restricciones se pueden aplicar tanto a los datos como a sus relaciones.
  - \* Control de la concurrencia para evitar la pérdida de información o la integridad de los datos.
  - \* Mejora en los servicios de copias de seguridad y de recuperación ante fallos.

#### **2.3.2. Lenguajes de definición y manipulación de datos. Funcionamiento.**

- Los sistemas gestores de bases de datos proporcionan un lenguaje para la definición de los datos (DDL), sus relaciones, sus condiciones de acceso e integridad, el control de vistas de usuarios y la especificación de las características físicas de la base de datos.
- También proporcionan un lenguaje para la manipulación de datos (DML) y un soporte para gestionar las peticiones del usuario: consulta, modificación, inserción y borrado de datos.
- La comunicación entre procesos de usuario, SGBD y sistema operativo consta de lo siguiente:
  - \* El proceso de usuario llama al SGBD indicando la porción de la base de datos a tratar.
  - \* El SGBD traduce la llamada a términos del esquema lógico de la base de datos. Accede al esquema lógico comprobando derechos de acceso y obtiene el esquema físico.
  - \* El SGBD traduce la llamada a los métodos de acceso del sistema operativo.
  - \* El sistema operativo accede a los datos tras traducir las órdenes dadas por el SGBD.
  - \* Los datos pasan del disco a una memoria intermedia o buffer, en la que se almacenan temporalmente, y de ésta al área de trabajo del usuario del proceso del usuario.
  - \* El SGBD devuelve indicadores al área de comunicaciones del proceso de usuario en los que manifiesta si ha habido errores o advertencias a tener en cuenta. Si las indicaciones son satisfactorias, los datos del área de trabajo serán utilizables por el proceso de usuario.

### 3. Funciones.

- **Acceso a los datos.** Proporciona a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos, ocultándoles la estructura física interna (la organización de los ficheros y las estructuras de almacenamiento).
- **Diccionario de datos.** Mantiene un catálogo accesible por los usuarios en el que se almacenan las descripciones de los datos. Este catálogo es lo que se denomina diccionario de datos y contiene información que describe los datos de la base de datos (metadatos).
- **Estados consistentes.** Se garantiza que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna. Una transacción es una secuencia atómica de acciones en una base de datos ejecutadas por un usuario. Cada transacción que parte de un estado consistente, si se ejecuta completamente, debe dejar la base de datos en otro estado consistente. El sistema gestor de base de datos asegura que la ejecución de transacciones de forma concurrente es igual a alguna ejecución en serie de esas transacciones.
- **Control de la concurrencia.** Proporciona un mecanismo que asegura que la base de datos se actualiza correctamente cuando varios usuarios la están actualizando concurrentemente, eliminando la posibilidad de interferencias o conflictos entre diferentes acciones.
- **Control de acceso.** Garantiza la seguridad de la información controlando el acceso a la misma únicamente a los usuarios autorizados. La protección debe ser contra accesos no autorizados, tanto intencionados como accidentales.
- **Recuperación de fallos.** Dispone de un sistema de recuperación de la base de datos en caso de que el sistema falle en medio de una transacción, con el fin de devolverla a un estado consistente. Este fallo puede ser a causa de un error del hardware o del software, o que el usuario detecte un error durante la transacción y la aborta antes de que finalice.
- **Comunicación por red.** Debe ser capaz de integrarse con algún software de comunicación, debido a que los usuarios pueden conectarse de forma remota, por lo que la comunicación con la máquina que alberga al sistema gestor de base de datos se debe hacer a través de una red.
- **Integridad de la información.** Mantiene la integridad de los datos expresada mediante restricciones, que son una serie de reglas que la base de datos no puede violar cuando se realizan operaciones de inserción, modificación o borrado. La integridad de la base de datos requiere la validez y consistencia de los datos almacenados.
- **Herramientas de administración.** Proporciona una serie de herramientas que permiten administrar la base de datos de modo efectivo, entre las cuales se encuentran las dedicadas a:
  - \* La creación y especificación de los datos y de la estructura de la base de datos.
  - \* La administración y creación de la estructura física en las unidades de almacenamiento.
  - \* La manipulación de los datos de las bases de datos.
  - \* La recuperación en caso de desastre y la creación de copias de seguridad.
  - \* La gestión de la comunicación de la base de datos.
  - \* La monitorización del uso y del funcionamiento de la base de datos.
  - \* El análisis estadístico para examinar las prestaciones o las estadísticas de utilización.

## 4. Componentes.

- Un sistema gestor de bases de datos se divide en módulos que se encargan de las responsabilidades del sistema. Algunas de estas funciones las proporciona el sistema operativo.
- Los componentes funcionales de un sistema gestor de base de datos se pueden dividir en:
  - \* *Procesador de consultas.* Es el componente principal, transforma las consultas en un conjunto de instrucciones de bajo nivel que se dirigen al gestor de la base de datos:
    - **Compilador de DML (Data Manipulation Language).** Traduce las instrucciones de DML a instrucciones de bajo nivel que entiende el motor de evaluación de consultas. Además, intenta transformar las peticiones del usuario en otras equivalentes pero más eficientes, encontrando así una buena estrategia para ejecutar la consulta.
    - **Precompilador de DML embebido.** Convierte las instrucciones de DML embebidas en un programa de aplicación en llamadas a procedimientos normales en el lenguaje anfitrión. Interactúa con el compilador de DML para generar el código apropiado.
    - **Intérprete de DDL (Data Definition Language).** Interpreta las instrucciones de DDL y las registra en un conjunto de tablas que contienen metadatos (catálogo).
    - **Motor de evaluación de consultas.** Ejecuta las instrucciones de bajo nivel generadas por el compilador de DML.
  - \* *Gestor de la base de datos.* Proporciona la interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación. Acepta consultas y examina los esquemas externo y conceptual para determinar qué registros se requieren para satisfacer la petición. Entonces realiza una llamada al gestor de ficheros para ejecutar la petición:
    - **Gestor de autorización e integridad.** Comprueba que se satisfagan las restricciones de integridad y la autorización de los usuarios para acceder a los datos.
    - **Gestor de transacciones.** Asegura que la base de datos quede en un estado consistente a pesar de los fallos del sistema, y que las ejecuciones de transacciones concurrentes ocurran sin conflictos.
    - **Gestor de ficheros.** Maneja los ficheros en disco en donde se almacena la base de datos utilizando los métodos de acceso del sistema operativo que se encargan de leer o escribir los datos en el buffer del sistema. Este gestor establece y mantiene la lista de estructuras e índices definidos en el esquema interno.
    - **Gestor de buffers.** Es responsable de traer los datos del disco de almacenamiento a memoria principal, y decidir qué datos tratar en la memoria caché.
    - **Gestor del diccionario de datos.** Controla los accesos al diccionario de datos y se encarga de mantenerlo.
    - **Gestor de recuperación.** Este módulo garantiza que la base de datos permanece en un estado consistente en caso de que se produzca algún fallo.
    - **Planificador.** Este módulo es el responsable de asegurar que las operaciones que se realizan concurrentemente sobre la base de datos tienen lugar sin conflictos.
    - **Optimizador de consultas.** Este módulo determina la estrategia óptima para la ejecución de las consultas.

- Los **componentes físicos** de un sistema gestor de base de datos son los siguientes:
    - \* *Ficheros de datos*. Almacenan la base de datos en sí.
    - \* *Diccionario de datos*. Almacena metadatos acerca de la estructura de la base de datos.
    - \* *Índices*. Proporcionan acceso rápido a elementos de datos que tienen valores particulares.
    - \* *Datos estadísticos*. Almacenan información estadística sobre los datos en la base de datos.
- El procesador de consultas usa esta información para ejecutar una consulta.

## 5. Arquitectura ANSI/SPARC.

### 5.1. Niveles. Funciones de traducción. Fases de la arquitectura.

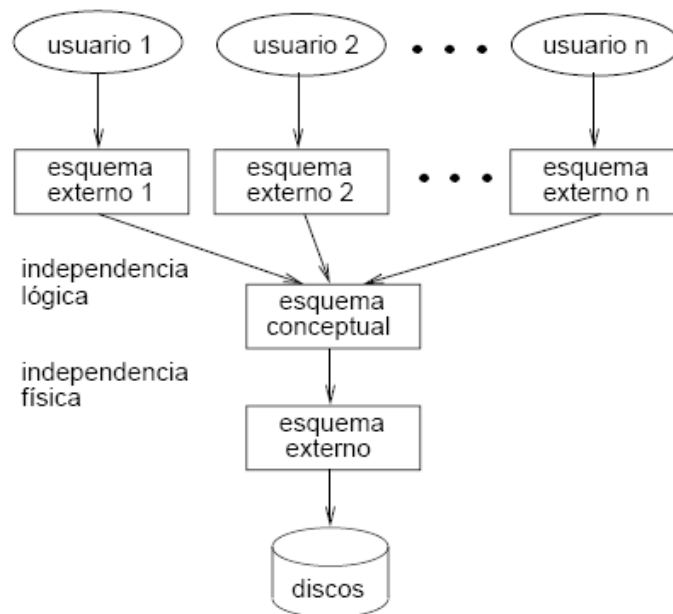
- En 1975 el comité ANSI/SPARC propuso una arquitectura cuyo objetivo es el de separar los programas de aplicación de la base de datos física. Para ello describe los datos bajo tres niveles de abstracción diferentes:
  - \* *Esquema externo (estructura lógica de usuario)*. Es la visión que cada usuario particular tiene de la base de datos. En él deberán encontrarse sólo aquellos datos y relaciones que necesite cada usuario. Habrá tantos esquemas externos como exijan las diferentes aplicaciones, aunque un mismo esquema externo podrá ser utilizado por varias aplicaciones.
  - \* *Esquema conceptual (estructura lógica global)*. Es la visión del administrador y responde al enfoque del conjunto de la realidad representada en la base de datos. Deben incluirse la descripción de entidades, atributos, relaciones, operaciones de los usuarios y restricciones de integridad y confidencialidad, ocultando los detalles de las estructuras de almacenamiento.
  - \* *Esquema interno (estructura física)*. Es la forma en que se organizan los datos en el almacenamiento físico. En este esquema se describen los ficheros y los índices utilizados, así como los métodos de acceso. Se encuentra en el paso anterior a los aspectos físicos como pista o cilindro, y es independiente de los dispositivos de almacenamiento, que son tratados por el gestor de ficheros.
- Entre estos niveles existen unos interfaces que realizan funciones de traducción:
  - \* *Función de traducción externo/conceptual*. Define la correspondencia entre cada una de las vistas externas y la única vista conceptual (diferentes tipos de datos, diferentes nombres de campos, múltiples registros conceptuales fundidos en un único registro externo).
  - \* *Función de traducción conceptual/interno*. Establece cómo se almacena a nivel interno los registros y campos conceptuales. Si se modifica el almacenamiento de los datos, sólo es necesario modificar la aplicación de correspondencia, y no la vista conceptual.
- La arquitectura completa está dividida en dos fases:
  - \* *Definición de datos*. La creación de la base de datos comienza con la elaboración del esquema conceptual, que se procesa utilizando una herramienta CASE que lo convierte en los metadatos. Utilizando esta información, se construyen los esquemas interno y externo.
  - \* *Manipulación de datos*. El usuario puede realizar operaciones sobre la base de datos. Esta petición es transformada por el transformador externo/conceptual que obtiene el esquema correspondiente ayudándose también de los metadatos. El resultado lo convierte otro transformador en el esquema interno usando también la información de los metadatos.



Finalmente del esquema interno se pasa a los datos usando el último transformador que también accede a los metadatos y de ahí se accede a los datos. Para que los datos se devuelvan al usuario en formato adecuado para él se tiene que hacer el proceso contrario.

### 5.2. Correspondencias entre niveles. Independencia de datos.

- En un SGBD basado en esta arquitectura, cada grupo de usuarios hace referencia exclusivamente a su propio nivel externo. Por tanto el SGBD debe transformar cualquier petición en términos de nivel externo a una petición expresada en términos de nivel conceptual, y después a una petición de nivel interno, que se procesará sobre la base de datos almacenada. El proceso de transformar peticiones y resultados de un nivel a otro se denomina correspondencia.
- La independencia de datos es la capacidad para modificar el esquema de un nivel del sistema sin tener que modificar el esquema del nivel inmediato superior. Esto significa:
  - \* *Independencia física de los datos.* Aunque el nivel físico cambie, el nivel conceptual no debe verse afectado. En la práctica esto significa que aunque se añadan o cambien discos u otro hardware, o se modifique el sistema operativo u otros cambios relacionados con la física de la base de datos, el nivel conceptual permanece invariable.
  - \* *Independencia lógica de los datos.* Significa que aunque se modifique el esquema conceptual, los esquemas externos y los programas de aplicación no serán afectados.



## 6. Arquitecturas operacionales.

- **Estructura cliente-servidor.** Estructura clásica, la base de datos y su SGBD están en un servidor al cual acceden los clientes. El cliente posee software que permite al usuario enviar instrucciones al SGBD en el servidor y recibir los resultados de estas instrucciones. Para ello el software cliente y el servidor deben utilizar software de comunicaciones en red.
- **Sistemas distribuidos.** Ocurre cuando los clientes acceden a datos situados en más de un servidor. También se conoce esta estructura como base de datos distribuida. El cliente no sabe si los datos están en uno o más servidores, ya que el resultado es el mismo independientemente de

dónde se almacenan los datos. En esta estructura hay un servidor de aplicaciones que es el que recibe las peticiones, y el encargado de traducirlas a los distintos servidores de datos para obtener los resultados. Los sistemas homogéneos utilizan el mismo SGBD en múltiples sitios. Los sistemas heterogéneos dotan de cierta autonomía local a los SGBD participantes.

- **Cliente/servidor Web/servidor de datos.** El cliente se conecta a un servidor mediante un navegador web y desde las páginas de éste ejecuta las consultas. El servidor web traduce esta consulta al servidor (o servidores) de datos.

## 7. Tipos de sistemas.

- **Relacionales.** Los datos y las relaciones existentes entre los datos se representan mediante tablas, cada una con un conjunto de columnas y un nombre único. La base de datos es percibida a nivel lógico (externo y conceptual) por el usuario como un conjunto de tablas, ya que a nivel físico pueden estar implementadas mediante distintas estructuras de almacenamiento. Sólo es necesario especificar que datos se han de obtener.
- **De red.** Los datos se presentan como colecciones de registros y las relaciones entre los datos se representan mediante conjuntos, que son punteros en la implementación física. Los registros se organizan como un grafo: los registros son los nodos y los arcos son los conjuntos. El SGBD de red más popular es el sistema IDMS. Es necesario especificar cómo deben obtenerse los datos.
- **Jerárquicos.** Son un tipo de modelo de red con algunas restricciones. Los datos se almacenan en estructuras lógicas denominadas segmentos que se relacionan entre sí mediante arcos. Hay una serie de nodos que contendrán atributos y que se relacionarán con nodos hijos, de forma que puede haber más de un hijo para el mismo padre, pero un hijo sólo tiene un padre. Por tanto una base de datos jerárquica puede representarse mediante un árbol. El SGBD jerárquico más importante es el sistema IMS. Es necesario especificar cómo deben obtenerse los datos.
- **Orientados a objetos.** Define una base de datos en términos de objetos, sus propiedades y sus operaciones. Los objetos con la misma estructura y comportamiento pertenecen a una clase, y las clases se organizan en jerarquías o grafos acíclicos. Las operaciones de cada clase se especifican en términos de procedimientos predefinidos denominados métodos. Su modelo conceptual se suele diseñar en UML y el lógico actualmente en ODMG (Object Data Management Group).
- **Objeto-relacionales.** Tratan de ser un híbrido entre el modelo relacional y el orientado a objetos. En las bases de datos objeto-relacionales se intenta conseguir una compatibilidad relacional dando la posibilidad de integrar mejoras de la orientación a objetos. Se basan en el estándar SQL 99, que añade a las bases relacionales la posibilidad de almacenar procedimientos de usuario, triggers, tipos definidos por el usuario, consultas recursivas, etc. Las últimas versiones de la mayoría de las grandes bases de datos relacionales (Oracle, SQL Server, Informix, ...) son objeto-relacionales.