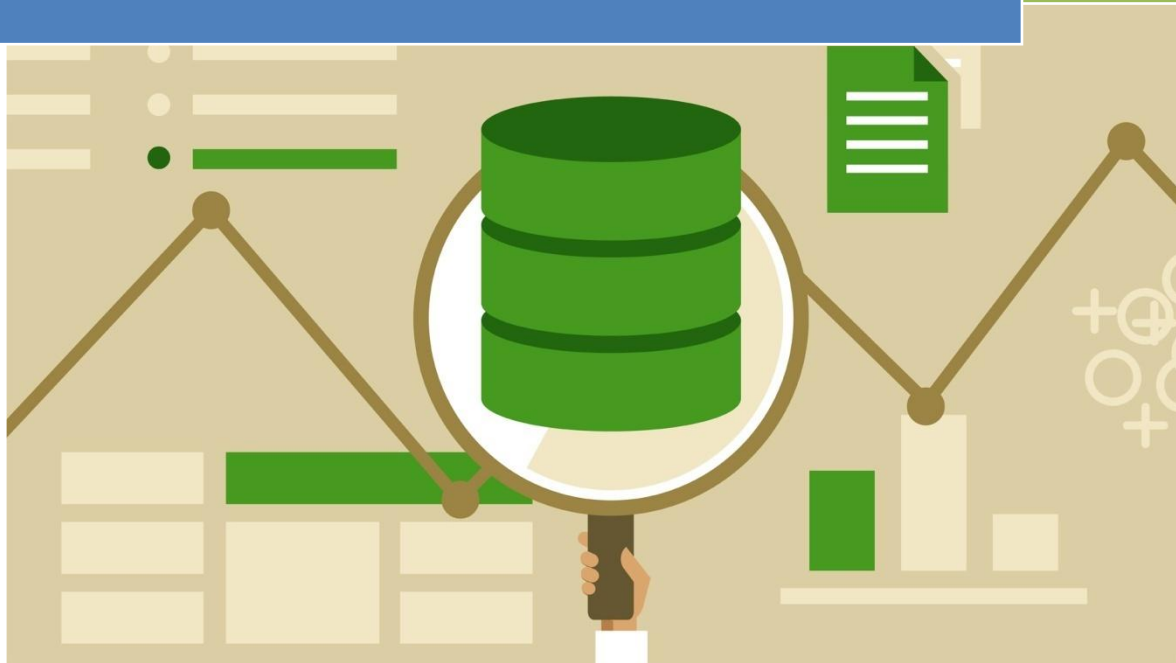


Base de Datos

2. Modelos de Base de Datos



Curso 2021-2022

Tabla de contenido

1. ARQUITECTURA DE LOS SGBD	3
1.1. ESQUEMAS Y NIVELES	3
2. LOS MODELOS DE BASES DE DATOS MÁS COMUNES.....	5
2.1. MODELO JERÁRQUICO.....	5
2.2. MODELO EN RED	6
2.3. MODELO RELACIONAL.....	7
2.4. EL PARADIGMA DE LA ORIENTACIÓN A OBJETOS	8
2.5. NUEVOS MODELOS DE BASES DE DATOS	9
3. BASES DE DATOS DISTRIBUIDAS	10
3.1. ARQUITECTURAS DE SISTEMAS DE BASES DE DATOS: CENTRALIZADAS, DESCENTRALIZADAS, CLIENTE-SERVIDOR	10
3.1.1. <i>Arquitecturas centralizadas y cliente-servidor</i>	11
3.1.2. <i>Arquitecturas descentralizadas</i>	12
3.1.3. <i>Ventajas e inconvenientes de la distribución de BD</i>	14

Las bases de datos (BD) representan informáticamente la parte del mundo real de nuestro interés, que previamente hemos conceptualizado, mediante unos procesos de observación y de abstracción.

Por tanto, podemos afirmar que las BD son modelos de la realidad. Pero no debemos confundir esta característica de las BD con el que se entiende por modelo de datos (o modelo de base de datos). La estructura concreta de cada BD está construida a partir del modelo de datos respectivo, elegido por el diseñador en función de las necesidades y de las herramientas disponibles.

Los **modelos de datos** son unos conjuntos de herramientas lógicas para describir los datos, sus interrelaciones, su significado y las restricciones a aplicar para garantizar su coherencia.

Todos los modelos de BD, en general, proporcionan tres tipos de herramientas:

- **Estructuras de datos** . Elementos con los que se construyen las BD, tales como tablas, árboles, etc.
- **Reglas de integridad**. Restricciones que los datos deberán respetar, como por ejemplo tipo de dato, dominios, llaves, etc.
- **Operaciones a realizar con los datos** . Altas, bajas, modificaciones y consultas, como mínimo.

1. Arquitectura de los SGBD

En 1975, el comité ANSI / X3 / SPARC propuso una arquitectura para los SGBD estructurada en tres niveles de abstracción (interno, conceptual y externo), que resulta muy útil para separar los programas de aplicación de la BD considerada desde un punto de vista físico.

SGBD

Acrónimo de Sistema Gestor de Bases de Datos. Es un software especializado en la gestión de BD bases de datos (entendidas, éstas, como un conjunto estructurado de información).

Por otra parte, también resulta interesante examinar la arquitectura de los SGBD desde un punto de vista funcional, ya que conociendo los diferentes componentes, y los flujos de datos y de control podemos entender el funcionamiento de los SGBD, y estaremos en mejores condiciones de utilizarlos de una manera óptima.

1.1. Esquemas y niveles

Para gestionar las BD, los SGBD deben conocer su estructura (es decir, las entidades, los atributos y las interrelaciones que contiene, etc.). Los SGBD necesitan disponer de una descripción de las BD que deben gestionar. Esta definición de la estructura recibe el nombre de esquema de la BD, y debe estar constantemente al alcance del SGBD para que éste pueda cumplir sus funciones.

De acuerdo con el estándar ANSI / X3 / SPARC, debería haber tres niveles de esquemas:

Comité ANSI3SPARC

Es un grupo de estudio de la Standard Planning and Requirements Comomittee (SPARC) del ANSI (American National Standards Institute), dentro del comité X3, que se ocupa de ordenadores y de informática.

- En el **nivel externo** se sitúan las diferentes visiones lógicas que los procesos usuarios (programas de aplicación y usuarios directos) tienen de las partes de la BD que utilizan. Estas visiones se denominan *esquemas externos* .
- En el **nivel conceptual** hay una sola descripción lógica básica, única y global, que llamamos *esquema conceptual* , y que sirve de referencia para el resto de esquemas.
- En el **nivel físico** hay una única descripción física, que llamamos *esquema interno* .

La figura 1 representa la relación, por una parte, entre el nivel físico y el esquema interno; y, de otra parte, entre el nivel lógico y el esquema conceptual, y los diferentes esquemas externos (o vistas).

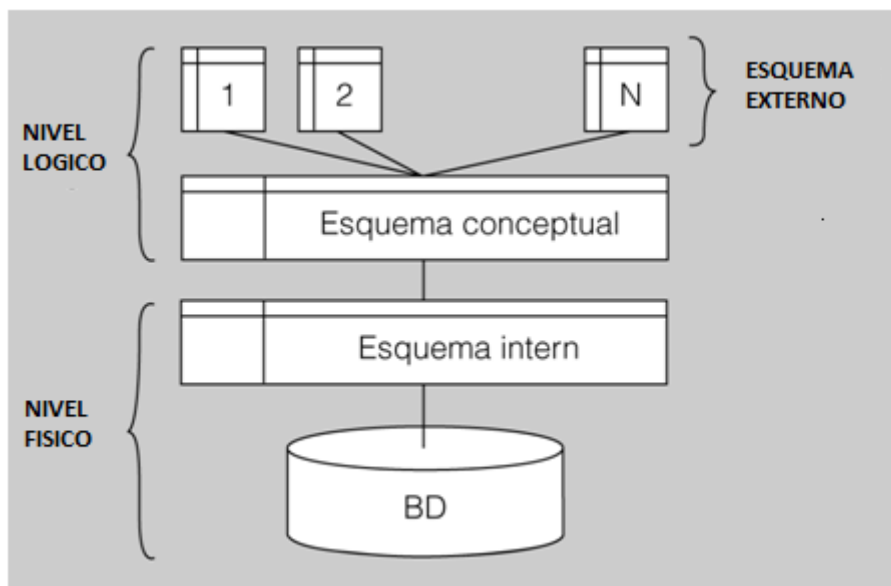


Figura 1 Esquemas y Niveles

Al definir un esquema externo, sólo se incluirán los atributos y entidades que interesen, los podremos renombrar, podremos definir datos derivados, podremos definir una entidad de tal manera que las aplicaciones que utilizan este esquema externo creen que son dos, o podremos definir combinaciones de entidades que en parezcan una sola, etc.

En el esquema conceptual, se describirán las entidades tipo, sus atributos, las interrelaciones y también las restricciones o reglas de integridad.

El esquema interno o físico contendrá la descripción de la organización física de la BD: caminos de acceso (índices, *hasHing* , apuntadores ...), codificación de los datos, gestión del espacio, tamaño de la página, etc.

2. Los modelos de bases de datos más comunes

Los modelos de datos más utilizados a lo largo del tiempo han sido los siguientes, expuestos en orden de aparición:

- * Jerárquico
- * En red
- * Relacional
- * Relacional con objetos / orientado a objetos

Actualmente, hay algunas nuevas tendencias incipientes, gracias al fenómeno Internet, aunque la información en las empresas sigue utilizando los modelos clásicos de BD.

2.1. Modelo jerárquico

Las BD jerárquicas se concibieron al principio de los años sesenta, y aún se utilizan gracias al buen rendimiento y la mejor estabilidad que proporcionan con grandes volúmenes de información.

Las **BD jerárquicas** almacenan la información en una estructura jerárquica que podemos imaginar con una forma de árbol invertido, donde cada nodo padre puede tener diferentes hijos. El nodo superior, que no tiene padre, se conoce como *raíz*. Y los nodos que no tienen hijos se llaman *hojas*.

Los datos se almacenan en forma de registros. Cada registro tiene una tupla de campos. Un conjunto de registros con los mismos campos forma un archivo.

Pero las BD jerárquicas no ofrecen una perspectiva lógica por encima de la física. Las relaciones entre los datos establecen siempre a nivel físico, mediante direccionamientos físicos en el medio de almacenamiento utilizado (es decir, indicando sectores y pistas, en el caso más habitual de los discos duros).

Así pues, las interrelaciones establecen mediante punteros entre registros. Cualquier registro contiene la dirección física de su registro padre en el medio de almacenamiento utilizado. Esta circunstancia proporciona un rendimiento muy bueno: el acceso desde un registro a otro es prácticamente inmediato.

Pero, si bien el modelo jerárquico optimiza las consultas de datos desde los nodos hacia el nodo raíz, con las consultas en sentido inverso se produce el fenómeno contrario, ya que entonces hay que hacer un recorrido secuencial de todos los registros de la BD.

Otras limitaciones de este modelo consisten en su incapacidad para evitar la redundancia (las repeticiones indeseadas de los datos) o para garantizar la integridad referencial (ya que los registros pueden quedar "huérfanos" en borrarse el nodo padre respectivo). La responsabilidad para evitar estos problemas queda en manos de las aplicaciones externas a la BD.

La figura 2 muestra la estructura de nodos interrelacionados de una BD jerárquica.

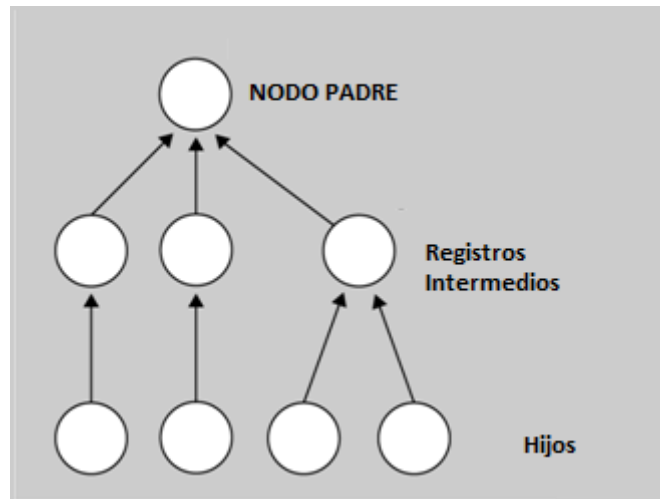


Figura 2 Estructura de una BD jerárquica

Dos de los gestores de BD jerárquicas que tienen mayor implantación son los siguientes:

- IMS (*information management system*), de la multinacional norteamericana IBM.
- Adabas (*adaptable database system*), de la empresa alemana Software AG.

2.2. Modelo en red

Al principio de los años setenta, en el mercado, fueron surgiendo BD que seguían un **modelo en red**, similar al modelo jerárquico, con registros interrelacionados mediante una estructura en forma de árbol invertido, pero más flexible, ya que permitía que los nodos tuvieran más de un solo padre.

La [figura3](#) muestra la estructura de nodos interrelacionados de una BD en red.

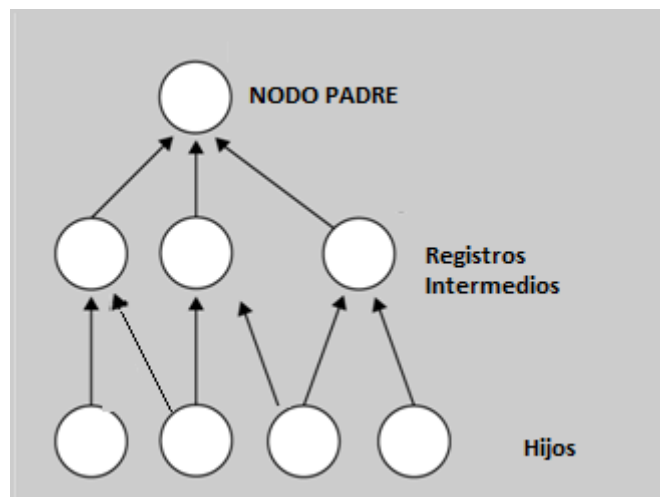


Figura 3 Estructura de una BD en red

El modelo en red supuso una mejora respecto al modelo jerárquico, porque permitía controlar de manera más eficiente el problema de la redundancia de datos.

A pesar de estas ventajas, el modelo en red no ha tenido tanta fortuna como su predecesor, debido a la complejidad que conlleva la administración de las BD que la adoptan.

El consorcio de la industria de las tecnologías de la información CODASYL (acrónimo de *Conference on Data Systems Languages*) propuso un estándar que siguieron la mayoría de fabricantes.

Uno de los gestores de BD en red más conocidos, y que sigue el estándar CODASYL, es la IDMS (*integrated database management system*), de Computer Associates.

2.3. Modelo Relacional

Modelo Relacional

Fue propuesto formalmente por E.F. Codd, 1970, en su trabajo *A Relational Model of Data for Large Shared Data Banks* ('Un modelo relacional de datos para grandes bancos de datos compartidos').

El **modelo relacional** se basa en la lógica de predicados y en la teoría de conjuntos (áreas de la lógica y de las matemáticas). Actualmente, es el sistema más ampliamente utilizado para modelar datos.

A partir de los años ochenta, se empezaron a comercializar gran cantidad de BD que aplicaban este modelo.

Los datos se estructuran en representaciones tabulares, llamadas **tablas**, que representan entidades tipo del mundo conceptual, y que están formadas por filas y columnas. Las columnas forman los **campos**, que implementan los atributos, es decir, las características que nos interesan de las entidades. Y las filas son los **registros**, que implementan las entidades instancia, constituidas por los conjuntos de los valores que presentan los campos correspondientes a cada instancia.

En los modelos de datos jerárquicos y en red los datos estructuraban gracias a dos elementos: los registros y las interrelaciones. Pero el modelo relacional sólo consta de un elemento: las **relaciones** o **tablas**.

La figura 4 muestra la estructura de tablas correspondiente a una BD relacional.



Figura 4 Estructura de una BD relacional

Las interrelaciones deben implementarse utilizando las tablas: cuando es necesario se deben añadir, a las tablas, uno o más campos que actúen como lo llamaremos clave foránea y que, por tanto, "apunten" al campo o campos referenciados de otra mesa, los cuales han de formar su clave primaria. Al coincidir los valores de los campos de la clave primaria y de la clave foránea, se establece la interrelación entre los registros.

El modelo relacional conlleva ciertas ventajas respecto al modelo jerárquico y al modelo en red:

- Proporciona herramientas para evitar la duplicidad de registros, mediante claves primarias y foráneas que permiten interrelacionar las tablas.
- Vela por la integridad referencial: al eliminarse un registro o al modificarse su valor, o bien no permite hacerlo si hay registros interrelacionados en otras tablas, o bien se borran o se modifican en cascada los registros interrelacionados, en función de qué orientación hayamos seguido en administrar la BD.
- Al no tener importancia a la ubicación de los datos, favorece la comprensibilidad. De hecho, el modelo relacional, como tal se limita al nivel lógico, y deja de lado el nivel físico. Por este motivo, se dice que el modelo relacional posibilita la independencia física de los datos.

2.4. El paradigma de la orientación a objetos

El **modelo de datos relacionales con objetos** es una extensión del modelo relacional en sentido estricto.

Este nuevo modelo admite la posibilidad de que los tipos de datos sean, además de los tradicionales, **tipos abstractos de datos (TAD)**. Con esta particularidad, se acercan los sistemas de BD relacionales al paradigma de la **programación orientada a objetos** (Object Oriented Programming o **OOP** en inglés).

TAD

Un tipo abstracto de datos (TAD) es un concepto que define los datos junto con sus operaciones asociadas.

Los modelos estrictamente orientados a objetos definen las BD en términos de objetos, de sus propiedades y, lo que es más innovador, de sus operaciones. Los objetos con una misma estructura y comportamiento pertenecen a una clase, y las clases se organizan en jerarquías. Las operaciones de cada clase se especifican en términos de procedimientos predefinidos, llamados *métodos*.

Oracle

Algunos SGBD presentes en el mercado desde hace mucho tiempo, van extendiendo el modelo relacional para incorporar conceptos relativos a la orientación a objetos. Este es el caso de la conocida firma Oracle, la cual está trabajando en esta línea desde la versión 8 de su producto.

La figura .5 muestra la estructura de clases, subclases e instancias, de una BD orientada a objetos.

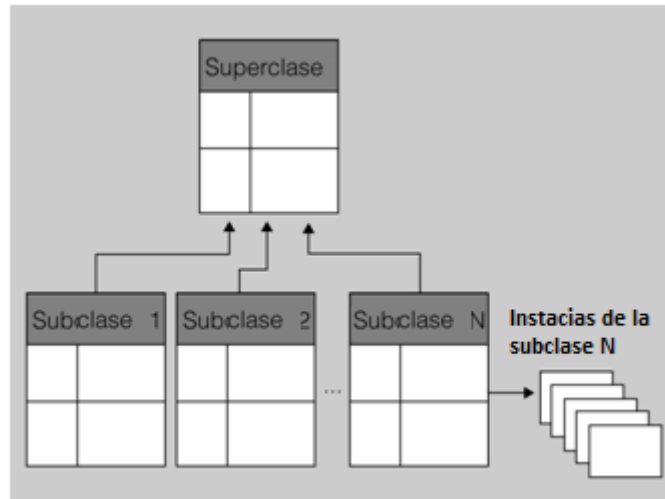


Figura 5 Estructura de una BD orientada a objetos

2.5. Nuevos modelos de bases de datos

Actualmente, hay organizaciones que gestionan grandes cantidades de datos y que estos datos los deben consultar habitualmente y necesitan que estas consultas sean rápidas. Típicamente esta necesidad se creó en empresas que necesitaban consultar datos para la toma de decisiones.

Se puede pensar, por ejemplo, en una multinacional que dispone de BD relacionales donde almacena todas sus facturas, todas las líneas de facturas, que se han ido haciendo a lo largo de 15 años de historia. Disponer de un resumen global de la evolución de la facturación de los últimos 10 años, puede suponer, en esta BD relacional la consulta de millones de registros. Se puede intuir, pues, que resultará un proceso costoso obtener esta información.

El modelo relacional, pues, muchas veces no da una respuesta suficientemente eficiente para gestionar este tipo de información y por ello comenzaron a aparecer los sistemas Datawarehouse. Los Datawarehouse son almacenes de datos que integran herramientas para extraer transformar y cargar información llamada de inteligencia empresarial así como información de metadatos.

En un almacén de datos lo que se quiere es contener datos que son necesarios o útiles para una organización, es decir, que se utiliza como un repositorio de datos para posteriormente transformarlos en información útil para el usuario.

Un almacén de datos debe entregar la información correcta a la gente indicada en el momento óptimo y en el formato adecuado.

3. Bases de datos distribuidas

BD

BD es el acrónimo de *base de datos*, entendida como conjunto estructurado de datos almacenados que permiten obtener información.

Uno de los sectores informáticos donde más se está evolucionando últimamente, integrando el desarrollo tecnológico con la innovación metodológica, es el relativo a los sistemas distribuidos de información.

Con esta última expresión queremos hacer referencia a la utilización de datos almacenados en diferentes ubicaciones, a veces muy distantes entre sí, pero que al mismo tiempo están conectadas, mediante una red de comunicaciones.

Esta tendencia se hace patente en el uso habitual de Internet por parte de cualquiera de nosotros, pero también hay numerosas empresas e instituciones que necesitan que los sistemas informáticos (y por tanto, también las BD) se adapten cada vez más a su estructura geográfica o funcional.

Un caso concreto de estos sistemas lo constituyen las bases de datos distribuidas. Es importante conocer las diferentes arquitecturas aplicadas a los sistemas de bases de datos, en general. Podemos distinguir entre las arquitecturas centralizadas (incluyendo los sistemas cliente-servidor) y las descentralizadas, en el que se debe tener en cuenta el funcionamiento de los sistemas paralelos.

También es importante conocer las diferentes metodologías para distribuir BD, en función de los objetivos planteados en la fase de diseño, y también de si la estrategia empleada tiene un carácter ascendente o descendente. Se deben tener en cuenta, sin embargo, las dos consecuencias más problemáticas de la distribución de BD: la duplicación y la fragmentación de los datos, donde debemos diferenciar entre fragmentaciones horizontales, verticales y mixtas.

No debemos olvidar tampoco las problemáticas específicas que representan para las BD distribuidas tanto las transacciones como la concurrencia, ni los diferentes protocolos con los que se da respuesta a estas eventualidades.

3.1. Arquitecturas de sistemas de bases de datos: centralizadas, descentralizadas, cliente-servidor

La arquitectura de todo sistema de BD está muy condicionada por las características del sistema informático sobre el que se ejecuta, y en especial por los siguientes aspectos:

- La conexión en red de diferentes computadoras.
- El procesamiento paralelo de consultas dentro de una misma computadora.
- La distribución de los datos en diferentes computadoras, incluso alejadas entre sí.

Estas innovaciones tecnológicas han permitido, respectivamente, el desarrollo, a partir de los iniciales sistemas totalmente centralizados en una sola computadora, de

diferentes arquitecturas de sistemas de BD más evolucionadas, y que permiten dar respuesta a una gran variedad de necesidades de los usuarios y de las organizaciones en que estos trabajan, tales como:

- Sistemas cliente-servidor.
- Sistemas paralelos.
- Sistemas distribuidos.

3.1.1. Arquitecturas centralizadas y cliente-servidor

Inicialmente los sistemas de BD eran de tipo estrictamente centralizado, en el sentido que se ejecutaban sobre un único sistema informático, sin necesidad de interactuar con ningún otro.

Pero el abaratamiento de los ordenadores personales y el desarrollo vertiginoso de sus capacidades, junto con la implantación de las redes y de Internet, ha hecho evolucionar los sistemas centralizados hacia arquitecturas de tipo cliente-servidor.

Sistemas centralizados en una sola computadora

Al hablar de sistemas de BD centralizados, se hace referencia tanto a los pequeños sistemas monousuarios que se ejecutan en un único ordenador personal, como a los grandes sistemas multiusuario de alto rendimiento.

Concurrencia

Se habla de concurrencia cuando varios procesos se ejecutan paralelamente, y, en este caso, hacen uso de los mismos datos.

En los sistemas monousuarios, los sistemas de BD centralizados son pequeños sistemas de BD pensados para las tareas que pueda hacer un solo usuario con una estación de trabajo. Estos sistemas no siempre ofrecen todas las posibilidades que siempre debe garantizar cualquier sistema de BD multiusuario, por modesto que sea, como por ejemplo el control automático de la concurrencia.

En los sistemas multiusuarios, en cambio, se trata de grandes sistemas que pueden dar servicio a un gran número de usuarios. Estos sólo disponen para interactuar con el sistema de terminales sin capacidad propia para almacenar datos ni tampoco para procesar consultas, ya que de la realización de estas tareas se encarga el único sistema centralizado existente.

Sistemas cliente-servidor

Los sistemas cliente-servidor tienen sus funcionalidades repartidas entre el sistema servidor central y múltiples sistemas clientes que le envían peticiones.

ODBC y JDBC

ODBC y JDBC son los acrónimos de open database connectivity y Java database connectivity, respectivamente.

Se trata de dos de los protocolos más utilizados para la interconexión de las aplicaciones de BD.

Gradualmente, los antiguos terminales de los sistemas centralizados han sido sustituidos por ordenadores personales, igualmente conectados a los subsistemas centrales. Como consecuencia de ello, prácticamente todos los sistemas centralizados actúan hoy en día como sistemas servidores que satisfacen las peticiones que les envían los respectivos sistemas clientes.

Las arquitecturas basadas en servidores de datos se han implantado especialmente en las BD orientadas a objetos.

Actualmente, los estándares ODBC y JDBC permiten que todo cliente que utilice cualquiera de los dos, se pueda conectar a cualquier servidor que proporcione la interfaz respectiva.

Podemos distinguir dos tipologías de sistemas servidores:

Las arquitecturas basadas en servidores de consultas son las que tienen más implantación.

Servidores de datos. Se utilizan en redes de área local en el que se llega a una alta velocidad de conexión entre los clientes y el servidor, siempre que las estaciones de trabajo sean comparables al servidor en cuanto a la capacidad de procesamiento. En entornos así definidos, puede tener sentido enviar los datos a los clientes, hacer allí todas las tareas de procesamiento de estos datos, y finalmente reenviar, si es necesario, los resultados al servidor.

Servidores de consultas. Proporcionan una interfaz, mediante la cual los clientes los envían peticiones para que resuelvan consultas, y los devuelvan los resultados obtenidos. Así pues, las transacciones se ejecutan sobre el servidor, pero los datos resultantes se visualizan en el cliente del que provenía la petición, en su caso.

3.1.2. Arquitecturas descentralizadas

Memoria principal vs. memoria persistente

Entendemos por memoria principal la memoria volátil, habitualmente la RAM. Entendemos por memoria persistente aquella que no es volátil. Habitualmente en forma de disco u otros dispositivos de almacenamiento externo.

La descentralización de las arquitecturas de BD puede consistir en el reparto de la carga de trabajo entre diferentes componentes físicos del sistema (básicamente en cuanto a procesadores, memoria principal y memoria persistente) comunicados entre sí mediante una red de interconexión.

Pero una arquitectura descentralizada también puede consistir en la distribución de la misma BD en diferentes computadoras, siguiendo la metodología más adecuada para alcanzar el objetivo propuesto.

SISTEMAS PARALELOS

El objetivo principal de los sistemas paralelos consiste en aumentar la velocidad de procesamiento y de E/S mediante la utilización en paralelo de UCP, memoria y discos duros.

E/S y I/O son los acrónimos de entrada y salida, y del original en inglés input / output.

Los sistemas paralelos son muy útiles (o incluso son imprescindibles) en el trabajo cotidiano con BD muy grandes (del orden de terabytes), o que tienen que procesar muchas transacciones (del orden de miles por segundo) , ya que normalmente los sistemas centralizados y los sistemas clienteservidor no tienen suficiente capacidad para dar respuesta a este tipo de necesidades.

UCP y CPU son los acrónimos de unidad central de proceso, y del original en inglés central processing unit.

Hoy en día muchos ordenadores de gama alta ofrecen un cierto grado de paralelismo, dado que incorporan dos o cuatro procesadores. Pero hay computadoras paralelas que soportan cientos de procesadores y discos duros.

Ahora bien, una de las características que permite evaluar la utilidad de un sistema paralelo de BD es su ampliabilidad, la cual debe garantizar el funcionamiento ulterior del sistema a una velocidad aceptable, aunque crezca el tamaño de la BD o el número de transacciones.

Por otra parte, hay diferentes modelos de arquitecturas paralelas de BD. Algunos de los modelos más importantes son:

- **Memoria compartida.** Todos los procesadores comparten una memoria común, habitualmente mediante un bus. Las arquitecturas de memoria compartida deben dotar cada procesador de mucha memoria caché, para evitar los accesos a la memoria compartida siempre que ello sea posible. Ahora bien, el límite razonable de procesadores trabajando en paralelo debe, justamente, por el coste que representa el mantenimiento de la coherencia de la caché.
- **Discos compartidos.** Todos los procesadores comparten un conjunto de discos común, mediante una red de interconexión. Este modelo permite el trabajo en paralelo de un número de procesadores mayor que con el modelo de memoria compartida, pero como contrapartida la comunicación entre ellos es más lenta, ya que utilizan una red en lugar de un bus.
- **Sin compartición.** Los procesadores no comparten ni memoria ni discos. Este modelo tiene unas potencialidades de ampliación aún mayores que el de compartición de discos, pero en cambio los costes de comunicación son superiores a los del modelo mencionado.
- **Jerárquico.** Combinación de las características de los modelos anteriores. Esta arquitectura se estructura en diferentes niveles. Al nivel más alto, los nodos, conectados mediante una red de interconexión, no comparten ni memoria ni discos. Cada uno de estos nodos puede ser,

internamente, un sistema de memoria compartida entre diferentes procesadores. O bien cada uno de estos nodos puede ser, internamente, un sistema de discos compartidos que incluya, en un tercer nivel, un sistema de memoria compartida.

SISTEMAS DISTRIBUIDOS

Una **BD distribuida** está formada por un conjunto de BD parcialmente independientes, almacenadas en diferentes computadoras, que comparten un esquema común y que coordinan el procesamiento de las transacciones que acceden a datos remotos.

Los procesadores de las diferentes computadoras que forman un sistema distribuido se comunican entre sí mediante una red de comunicaciones, pero no comparten ni memoria ni discos. Cada una de estas computadoras constituye, por tanto, un **nodo del sistema distribuido**.

Normalmente, los nodos de las BD distribuidas se encuentran en lugares geográficamente distantes, se administran parcialmente de manera independiente y tienen una interconexión fuerza lenta entre ellos.

SGBD

SGBD es el acrónimo de **sistema gestor de bases de datos**. Se trata de un software especializado en la gestión y el almacenamiento de BD.

Normalmente, todo SGBD actual es capaz de trabajar con otro idéntico. Pero esto no siempre es tan fácil entre SGBD de diferentes fabricantes. En función de esta eventualidad, se distingue entre dos tipos de sistemas distribuidos de BD:

- **Sistemas homogéneos:** son sistemas fuertemente acoplados, en la que todos los nodos utilizan el mismo SGBD o, en el peor de los casos, diferentes SGBD del mismo fabricante.
- **Sistemas heterogéneos:** los SGBD que utilizan los nodos son diferentes, y, por tanto, suelen ser más difíciles de acoplar.

El acoplamiento es el grado de interacción y dependencia que tienen dos partes de un sistema.

En todo caso, los usuarios de los sistemas distribuidos de BD no deben conocer los detalles de almacenamiento de los datos que utilice, como su ubicación concreta o su organización. De esta característica se llama **transparencia**. Evidentemente, los administradores de la BD sí deberán ser conscientes de estos aspectos.

3.1.3. Ventajas e inconvenientes de la distribución de BD

Hay buenas razones para implementar BD distribuidas, como la compartición de la información, la disponibilidad de los datos o la agilización del procesamiento de algunas consultas:

- **Compartición de la información y autonomía local**. Una ventaja de compartir los datos mediante la distribución consiste en que desde cada

nodo se puede controlar, hasta cierto punto (de hecho, hasta donde permita el administrador global de la BD), la administración de los datos almacenados localmente. Esta característica se conoce como *autonomía local*.

- **Fiabilidad y disponibilidad** . Si se produce un fallo en algún nodo de un sistema distribuido o en las comunicaciones con este, es posible que los otros nodos puedan continuar trabajando, si los datos que contiene el nodo caído o incomunicado están repetidas en otros nodos del sistema.
- **Agilización del procesamiento de consultas** . Cuando una consulta necesita acceder a datos almacenados en diferentes nodos, puede ser posible dividir la consulta en diferentes subconsultas que se ejecuten en los nodos respectivos.

Pero el uso de BD distribuidas también tiene sus puntos débiles, como por ejemplo el incremento de los costes en desarrollo de software y el aumento de posibilidad de errores, y el tiempo adicional a añadir al tiempo de procesamiento:

- **Incremento en los costos de desarrollo de software** . Es más difícil estructurar un sistema distribuido de BD, y también son más complicadas las aplicaciones que deben trabajar con este sistema, que si se trata de un sistema de BD centralizado.
- **Más posibilidad de errores**. Como los diferentes nodos del sistema distribuido operan en paralelo, es más difícil garantizar la corrección de los algoritmos.
- **Tiempo extra que hay que añadir al tiempo de procesamiento**. El intercambio de mensajes y los cálculos necesarios para garantizar la integridad de los datos distribuidas entre todos los nodos conllevan un agregado extra de tiempo, inexistente en los sistemas centralizados.