

Bases de Datos – Antecedentes – Definición

El término bases de datos fue escuchado por primera vez en un simposio celebrado en California en 1963.

En una primera aproximación, se puede decir que una base de datos es un conjunto de información relacionada que se encuentra agrupada o estructurada.

Desde el punto de vista informático, una base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Por su parte, un sistema de Gestión de Bases de datos es un tipo de software muy específico dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan; o lo que es lo mismo, una agrupación de programas que sirven para definir, construir y manipular una base de datos, permitiendo así almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

Actualmente, las bases de datos están teniendo un impacto decisivo sobre el creciente uso de las computadoras.

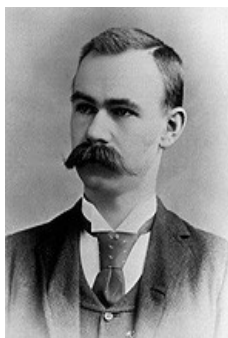
Pero para poder entender más profundamente una base de datos cabe entender su historia.

Orígenes

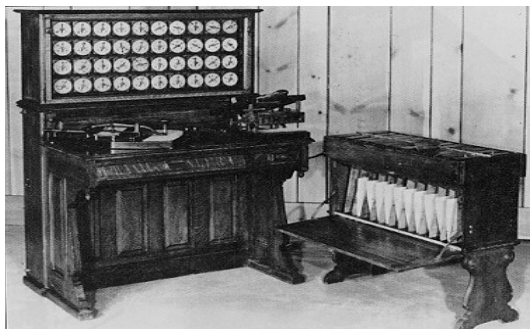
Los orígenes de las bases de datos se remontan a la Antigüedad donde ya existían bibliotecas y toda clase de registros. Además también se utilizaban para recoger información sobre las cosechas y censos. Sin embargo, su búsqueda era lenta y poco eficaz y no se contaba con la ayuda de máquinas que pudiesen reemplazar el trabajo manual.

Posteriormente, el uso de las bases de datos se desarrolló a partir de las necesidades de almacenar grandes cantidades de información o datos. Sobre todo, desde la aparición de las primeras computadoras, el concepto de bases de datos ha estado siempre ligado a la informática.

En 1884 Herman Hollerith creó la máquina automática de tarjetas perforadas, siendo nombrado así el primer ingeniero estadístico de la historia. En esta época, los censos se realizaban de forma manual.



Ante esta situación, Hollerith comenzó a trabajar en el diseño de una máquina tabuladora o censadora, basada en tarjetas perforadas.



Posteriormente, en la década de los cincuenta se da origen a las cintas magnéticas, para automatizar la información y hacer respaldos. Esto sirvió para suplir las necesidades de información de las nuevas industrias. Y a través de este mecanismo se empezaron a automatizar información, con la desventaja de que solo se podía hacer de forma secuencial.



Década de 1960

Posteriormente en la época de los sesenta, las computadoras bajaron los precios para que las compañías privadas las pudiesen adquirir; dando paso a que se popularizara el uso de los discos, cosa que fue un adelanto muy efectivo en la época, debido a que a partir de este soporte se podía consultar la información directamente, sin tener que saber la ubicación exacta de los datos.

En esta misma época se dio inicio a las primeras generaciones de bases de datos de red y las bases de datos jerárquicas, ya que era posible guardar estructuras de datos en listas y arboles.

Otro de los principales logros de los años sesenta fue la alianza de IBM y American Airlines para desarrollar SABRE, un sistema operativo que manejaba las reservas de vuelos, transacciones e informaciones sobre los pasajeros de la compañía American Airlines.

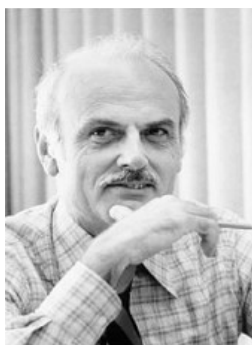
Y, posteriormente, en esta misma década, se llevó a cabo el desarrollo del IDS desarrollado por Charles Bachman (que formaba parte de la CODASYL) supuso la creación de un nuevo tipo de sistema de bases de datos conocido como modelo en red que permitió la creación de un standard en los sistemas de bases de datos gracias a la creación de nuevos lenguajes de sistemas de información.

CODASYL (Conference on Data Systems Languages) era un consorcio de industrias informáticas que tenían como objetivo la regularización de un lenguaje de programación estándar que pudiera ser utilizado en multitud de ordenadores.

Los miembros de este consorcio pertenecían a industrias e instituciones gubernamentales relacionadas con el proceso de datos, cuya principal meta era promover un análisis, diseño e implementación de los sistemas de datos más efectivos; y aunque trabajaron en varios lenguajes de programación como COBOL, nunca llegaron a establecer un estándar fijo, proceso que se llevó a cabo por ANSI.

Década de 1970

Por lo que respecta a la década de los setenta, Edgar Frank Codd, científico informático inglés conocido por sus aportaciones a la teoría de bases de datos relacionales, definió el modelo relacional a la par que publicó una serie de reglas para los sistemas de datos relacionales a través de su artículo "Un modelo relacional de datos para grandes bancos de datos compartidos".



Este hecho dio paso al nacimiento de la segunda generación de los Sistemas Gestores de Bases de Datos.

Como consecuencia de esto, durante la década de 1970, Lawrence J. Ellison, más conocido como Larry Ellison, a partir del trabajo de Edgar F. Codd sobre los sistemas de bases de datos relacionales, desarrolló el Relational Software System, o lo que es lo mismo, lo que actualmente se conoce como Oracle Corporation, desarrollando así un sistema de gestión de bases de datos relacional con el mismo nombre que dicha compañía.



Posteriormente en la época de los ochenta también se desarrollará el SQL (Structured Query Language) o lo que es lo mismo un lenguaje de consultas o lenguaje declarativo de acceso a bases de datos relacionales que permite efectuar consultas con el fin de recuperar información de interés de una base de datos y hacer cambios sobre la base de datos de forma sencilla; además de analiza grandes cantidades de información y permitir especificar diversos tipos de operaciones frente a la misma información, a diferencia de las bases de datos de los años ochenta que se diseñaron para aplicaciones de procesamiento de transacciones.

Pero cabe destacar que ORACLE es considerado como uno de los sistemas de bases de datos más completos que existen en el mundo, y aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace relativamente poco, actualmente sufre la

competencia del SQL Server de la compañía Microsoft y de la oferta de otros Sistemas Administradores de Bases de Datos Relacionales con licencia libre como es el caso de PostgreSQL, MySQL o Firebird que aparecerían posteriormente en la década de 1990.

Década de 1980

Por su parte, a principios de los años ochenta comenzó el auge de la comercialización de los sistemas relacionales, y SQL comenzó a ser el estándar de la industria, ya que las bases de datos relacionales con su sistema de tablas (compuesta por filas y columnas) pudieron competir con las bases jerárquicas y de red, como consecuencia de que su nivel de programación era sencillo y su nivel de programación era relativamente bajo.



Década años 1990

En la década de 1990 la investigación en bases de datos giró en torno a las bases de datos orientadas a objetos. Las cuales han tenido bastante éxito a la hora de gestionar datos complejos en los campos donde las bases de datos relacionales no han podido desarrollarse de forma eficiente.

Definición

Las bases de datos son sistemas estructurados de forma lógica para la administración electrónica de datos que, con ayuda de un sistema de gestión de bases de datos (database management system, DBMS), regulan las pertenencias y los derechos de acceso y guardan la información, añadiéndola al repositorio que contienen. La mayoría de bases de datos solo pueden abrirse, editarse y consultarse con aplicaciones específicas.

¿Por qué son necesarias las bases de datos?

Los primeros modelos que se desarrollaron fueron las bases de datos en red y jerárquicas, si bien pronto demostraron ser demasiado simples y estar limitadas técnicamente. IBM fue la empresa que revolucionó el sector, con el desarrollo del modelo relacional de base de datos en los años setenta, con mucho el más potente, que pronto encontró un campo de cultivo favorable en el mundo laboral. Los productos que más éxito tuvieron en este momento, fueron el lenguaje de consultas a bases de datos SQL de Oracle y los sucesores de IBM, SQL/DS y DB2.

Hasta bien entrados los años 2000, cuando algunos proyectos de código libre insuflaron algo de aire fresco al sector, el mercado del software de base de datos estuvo gobernado por los pesos pesados. Entre los sistemas libres más populares se cuentan MySQL y PostgreSQL. La tendencia iniciada en 2001 hacia los sistemas NoSQL también contribuyó a la debilitación de la posición de los sistemas de bases de datos de los grandes fabricantes.

Hoy, los sistemas de bases de datos son imprescindibles en numerosos campos. Cualquier tipo de software concebido para las empresas se basa en robustas bases de datos con un gran número de opciones y herramientas para los administradores del sistema. La seguridad de los

datos, además, ha ido ganando importancia con el tiempo, y es que en las bases de datos electrónicas se almacenan y cifran contraseñas, datos personales e incluso divisa digital.

El sistema financiero moderno, no es más que una red de bases de datos, en la cual la mayor parte de las cuantías monetarias solo existen como unidades electrónicas de información, cuya protección, por medio de bases de datos seguras es una de las tareas principales de las instituciones financieras. Aunque no solo por esto son cruciales las bases de datos electrónicas para la civilización moderna.

Funciones y condiciones de un sistema de gestión de base de datos (SGBD)

Un término muy extendido para describir las funciones y los requisitos de las transacciones en un database management system es el de ACID, acrónimo de atomicity, consistency, isolation y durability (atomicidad, consistencia, aislamiento, durabilidad). Estos cuatro parámetros, cubren los requisitos más importantes de un SGBD (ACID compliant):

Atomicidad designa a la propiedad "todo o nada" de los gestores de bases de datos: para que una consulta sea válida y la transacción se complete correctamente se ha de llevar a cabo en el orden correcto de pasos.

La consistencia (o coherencia) se da cuando al finalizar una transacción, la base de datos sigue siendo estable, lo que requiere la supervisión continua de todas las transacciones.

El aislamiento es la condición que garantiza que las transacciones no se obstaculicen unas a otras, algo que normalmente se logra con ciertas funciones de bloqueo que aíslan los datos que participan en una transacción.

La durabilidad significa que en un SGBD todos los datos se guardan a largo plazo incluso tras concluir una transacción y también, o especialmente, en el caso de fallos del sistema o caídas del SGBD. Para esta condición, son esenciales los registros de transacción, que protocolizan todos los procesos que tienen lugar en el SGBD.

A continuación detallamos una forma diferente de clasificar las funciones y los requisitos de un sistema de gestión de bases de datos:

Función/condición	Significado
Almacenar datos	Las bases de datos almacenan textos, documentos, contraseñas, etc., en formato electrónico, a los que puede accederse mediante consultas.
Editar datos	Según de qué permisos se disponga, la mayoría de bases de datos permiten editar in situ los datos que salvaguardan.
Borrar datos	Los registros de las bases de datos pueden borrarse por completo, sin dejar espacios en blanco. En algunos casos los datos que se han borrado pueden restablecerse, pero en otros, se eliminan definitivamente.
Gestionar los metadatos	Normalmente, la información se guarda con metadatos o metaetiquetas que mantienen el orden dentro de la base de datos y hacen posible la función de búsqueda. Los metadatos también suelen

	utilizarse para regular los permisos.
Seguridad de los datos	<p>La gestión de datos comprende cuatro operaciones fundamentales: crear (create), leer/recuperar (read/retrieve), actualizar (update) y borrar (delete). Este concepto, conocido por su acrónimo CRUD, constituye la base de la gestión de datos.</p> <p>Las bases de datos han de ser seguras para evitar que sujetos no autorizados puedan acceder a la información que guardan. Además de un solvente método de cifrado, para mantener la seguridad de los datos es esencial poner esmero en su administración, sobre todo su administrador principal. La seguridad de los datos implica tomar las precauciones técnicas necesarias para impedir la manipulación o la pérdida de datos.</p>
Integridad de los datos	<p>La integridad de los datos significa que los datos han de cumplir con ciertas reglas para asegurar su corrección y definir la lógica de negocio del banco de datos. Solo así, puede asegurarse que la base de datos, al completo, funciona de forma constante y coherente. En los modelos relacionales se dan cuatro de estas reglas: integridad de campo, integridad de entidad, integridad referencial y consistencia lógica.</p>
Función multiusuario	<p>Las aplicaciones de base de datos permiten acceder a las bases de datos desde diferentes dispositivos. El reparto de permisos y la seguridad de los datos son elementales en el uso multiusuario. También constituye un reto, mantener la consistencia de los datos sin dificultar el rendimiento, cuando varios usuarios leen y escriben a la vez.</p>
Optimizar las consultas	<p>Técnicamente, una base de datos ha de poder procesar las consultas de la mejor manera posible para garantizar una buena performance. Si utiliza demasiadas rutas diferentes para solucionar una consulta, el rendimiento global del sistema se verá perjudicado.</p>
Triggers y stored procedures	<p>Estos dos procedimientos son miniaplicaciones guardadas en los SGBD que se activan con ciertos eventos. Con ellos se pretende, entre otras cosas, mejorar la integridad de los datos. Los</p>

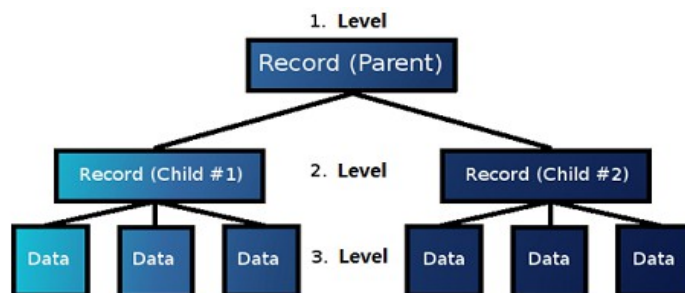
	disparadores (triggers) y los procedimientos almacenados (stored procedures) son procesos típicos de las bases de datos relacionales. Los segundos contribuyen a la seguridad del sistema si los usuarios solo ejecutan las acciones con procedimientos predefinidos.
Transparencia del sistema	La transparencia del sistema es relevante, sobre todo, en los sistemas distribuidos; privando al usuario de la distribución y la implementación de los datos, la utilización de una base de datos distribuida se asemeja al de una centralizada. Los procesos que corren en segundo plano se muestran u ocultan en diversos niveles de transparencia. La función principal es, no obstante, simplificar su uso todo lo posible.

Evolución de los modelos de bases de datos

Las diferencias entre los modelos de bases de datos más habituales es resultado de la evolución técnica de la transmisión electrónica de datos, que no solo perseguía la eficiencia y la manejabilidad, sino también, el empoderamiento de los fabricantes más renombrados.

Modelo jerárquico de base de datos

Este es el modelo más antiguo, hoy superado en gran medida por el modelo relacional (entre otros), si bien recientemente su empleo ha ido creciendo. XML utiliza este sistema para guardar datos y algunas compañías de seguros y bancos recurren a las bases de datos jerárquicas sobre todo en las aplicaciones más antiguas de base de datos. El sistema de base de datos jerárquico más conocido es IMS/DB de IBM.



En el modelo jerárquico de base de datos las dependencias son estrictas

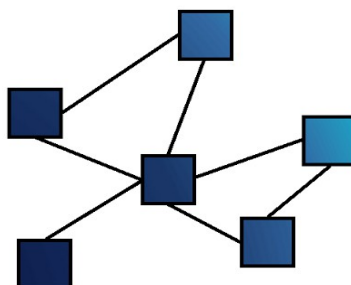
En las bases de datos jerárquicas las dependencias son inequívocas. Cada registro tiene solo un precedente (Parent-Child Relationships, PCR) a excepción de la raíz (root), constituyendo un esquema en árbol como el de arriba. Mientras que cada nodo "hijo", solo puede tener un nodo "padre", los "padres" pueden tener tantos "hijos" como quieran. Dado el estricto ordenamiento jerárquico, los niveles sin relación directa, no interactúan entre sí y conectar dos árboles diferentes tampoco es fácil. Por todo esto, las estructuras de base de datos jerárquicas son

extremadamente inflexibles, pero muy claras.

Los registros con hijos se llaman records y los que no tienen se llaman hojas y son los que suelen contener los documentos. Los records sirven para clasificar las hojas. Las consultas a una base de datos jerárquica alcanzan a las hojas partiendo desde la raíz y pasando por los distintos records.

Base de datos en red

El modelo en red se desarrolló casi de forma simultánea al relacional, aunque con el tiempo sería superado por la competencia. A diferencia del modelo jerárquico, aquí los registros o records no revelan relaciones padre-hijo estrictas, sino que cada registro puede tener múltiples precedentes, lo que le da la estructura en red de su nombre. Para acceder a un registro tampoco hay, por eso mismo, un camino único e invariable.



En el modelo de base de datos en red no gobierna ninguna jerarquía fija y, por lo tanto, son varios los caminos que llevan a un mismo destino

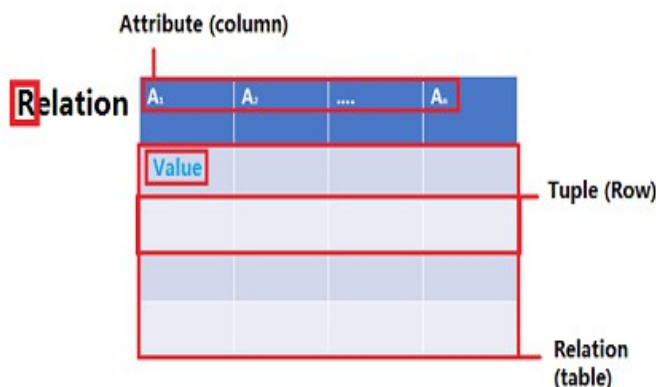
Al registro situado en el centro de la imagen puede accederse en teoría desde los otros cinco, y accediendo a él, puede accederse a otros cinco registros. En el modelo en red también pueden definirse dependencias: el registro situado más arriba no está conectado directamente con el de más a la derecha, de modo que para llegar a él ha de pasar por el del centro, que puede aceptar o denegar el paso. Podría entonces establecer contacto con el de arriba a la izquierda. En el modelo en red, los registros pueden añadirse o eliminarse sin que la estructura global se vea afectada.

Hoy el modelo de base de datos en red se utiliza, sobre todo, en los grandes ordenadores. En otros campos se sigue confiando en el modelo jerárquico (clientes de IBM, sobre todo) o se ha dado el paso hacia el modelo relacional, mucho más flexible y fácil de utilizar. Algunos modelos conocidos de base de datos en red son el UDS de Siemens y el DMS de Sperry Univac. Con el tiempo, ambos fabricantes han desarrollado también interesantes formas mixtas entre el modelo en red y el relacional aunque sin lograr arrancar del todo. Con todo, aún hoy pueden encontrarse aspectos de estos intentos en el SQL de Siemens. La base de datos orientada a grafos, por su estructura reticular, es considerada la evolución moderna del modelo en red.

Modelo de base de datos relacional

El modelo que goza de más popularidad a día de hoy es el relacional, aunque tampoco queda libre de crítica. Su correspondiente sistema de gestión es más conocido como SGBDR (RDBMS en inglés) y como lenguaje utiliza normalmente SQL. Este modelo basado en tablas, gira en torno al concepto de relación, un término bien definido en matemáticas y que aquí se utiliza como sinónimo de tabla. Para formular las relaciones se utiliza álgebra relacional, con cuya ayuda puede obtenerse la información de estas relaciones. Este es el principio que fundamenta

el lenguaje SQL.



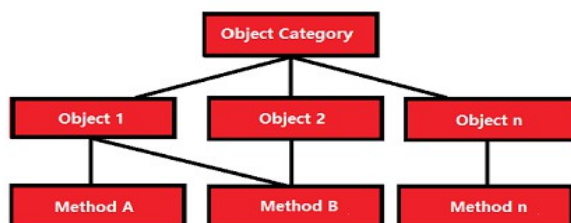
El modelo relacional se basa en tablas y se ha consolidado como estándar de bases de datos

El modelo relacional trabaja con tablas independientes que determinan la localización de los datos y sus conexiones. Estos datos conforman un registro (en la imagen, una fila o "tupla") y se guardan en columnas como atributos (en la imagen, de A_1 a A_n). La relación es lo que resulta de los atributos interrelacionados. Para identificar inequívocamente un registro es elemental la clave primaria, que normalmente se define como el primer atributo (A_1) y que no puede cambiarse. Dicho de otra manera, esta clave primaria o ID, define la posición exacta del registro con todos los atributos.

(Nota: pueden ampliar viendo este video <https://www.youtube.com/watch?v=CBVp8sbo1w0>)

Modelo de base de datos orientado a objetos

Las bases de datos de objetos no nacen hasta finales de 1980 y hasta hoy, solo han encontrado una escasa aplicación. Estas bases de datos, disponibles también en formato open source, suelen utilizarse en plataformas Java y .NET. La más conocida es db4o, que destaca, sobre todo, por un escaso uso de la memoria. Las bases de datos de objetos acostumbran a trabajar con el lenguaje OQL, muy similar a SQL.



El modelo orientado a objetos prevé el almacenamiento de los datos junto a sus métodos de acceso

En el modelo orientado a objetos, los datos se guardan en un objeto junto con sus funciones (métodos) y los atributos que los describen más en profundidad. En un sistema de gestión de bases de datos de objetos, son los métodos, depositados en el objeto junto con los datos, los que definen cómo se accede al objeto.

Los objetos pueden ser complejos y estar compuestos por múltiples tipos de datos, son únicos

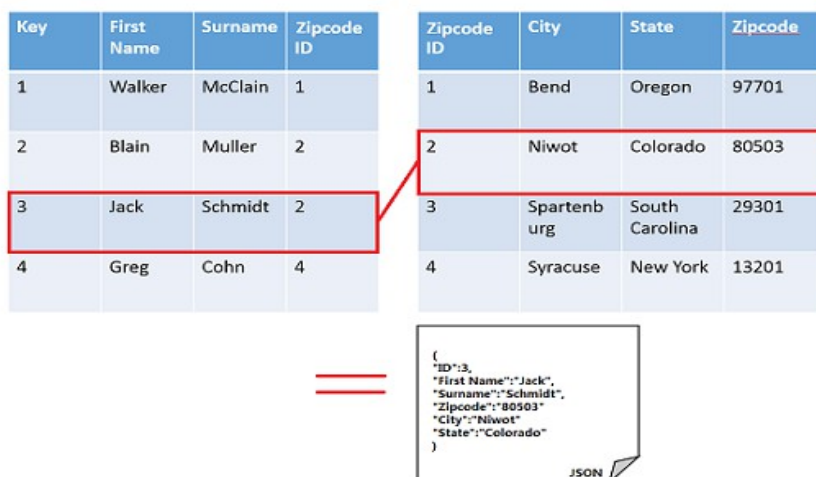
dentro del sistema de base de datos y se identifican con un identificador de objeto (OID en inglés) único. Como puede verse en la figura de arriba, los objetos se agrupan en clases (object category), dando como resultado una jerarquía de clases. Pese a la aparente similitud con el modelo jerárquico, aquí predomina el paradigma orientado a objetos y no existe ninguna relación padre-hijo fija. Aún así, a través de la clase puede definirse el método para el acceso.

Las ventajas de las bases de datos orientadas a objetos destacan, sobre todo, en problemas con tipos de datos complejos. Estas bases de datos trabajan, en su mayor parte, de forma autónoma sin recurrir a la normalización y a la correspondencia de ID, permitiendo así almacenar los objetos nuevos de forma relativamente simple y fluida. Sin embargo, las consultas son mucho más ágiles en un sistema de base de datos relacional. La escasa popularidad de los sistemas orientados a objetos resulta en una insuficiente compatibilidad con muchas de las aplicaciones de base de datos que se usan habitualmente.

Modelo de base de datos orientado a documentos

En este modelo, los documentos son la unidad básica para el almacenamiento de datos. Estas unidades son las que estructuran los datos y no deben confundirse con los documentos de los programas de procesamiento de texto. Aquí, los datos se guardan en los llamados pares clave-valor, comprendiendo así, una "clave" y un "valor". Como no están definidos ni la estructura ni el número de pares, los documentos que integran una base de datos orientada a documentos pueden resultar muy dispares entre sí. Cada documento es una unidad cerrada en sí misma y establecer relaciones entre documentos no resulta fácil, pero en este modelo no es necesario.

(Nota: En los últimos años, y gracias al éxito de NoSQL, las bases de datos documentales han experimentado un gran auge, sobre todo, por su buena escalabilidad. Un ejemplo para este tipo de sistema de base de datos es MongoDB.)



En el modelo de base de datos orientado a documentos, los datos se guardan en documentos individuales y no en tablas como en el modelo relacional

En el modelo relacional (arriba representado con las dos tablas), varias relaciones (tablas) se conectan entre sí para seleccionar un registro común. En el modelo documental, un único documento basta para guardar toda la información. Aquí no se está obligado a utilizar un determinado esquema porque, mientras se use siempre el mismo lenguaje de base de datos, este modelo está conceptualmente libre de esquemas.

Una idea fundamental de las bases de datos documentales es que los datos que guardan

relación entre sí siempre se guardan juntos en un lugar (en el documento). Mientras que las bases de datos relacionales suelen representar y mostrar la información relacionada conectando varias tablas, en el modelo que nos ocupa es suficiente con consultar un solo documento. Esto reduce el número de procedimientos necesarios para consultar la base de datos.

Estos sistemas son especialmente interesantes para las aplicaciones web, puesto que permiten guardar formularios HTML completos. Fue sobre todo con el avance de la web 2.0 cuando estas bases de datos vieron aumentar su popularidad. Con todo, es necesario remarcar que entre los diversos sistemas basados en documentos se dan diferencias notables, desde la sintaxis hasta la estructura interna, por lo que no todas las bases de datos orientadas a documentos son apropiadas para cualquier escenario. Es debido a estas diferencias por lo que hoy disponemos de algunos sistemas de bases de datos orientados a documentos de la reputación de Lotus Notes, Amazon SimpleDB, MongoDB, CouchDB, Riak, ThruDB, OrientDB, etc.

Bases de datos: modelos y características

Modelo de base de datos	Desarrollo	Ventajas	Inconvenientes	Ámbitos de aplicación	Marcas
Jerárquico	Década de 1960	Acceso de lectura muy rápido, estructura clara, técnicamente simple	Estructura fija en árbol que no permite conexiones entre árboles	Bancos, compañías de seguros, sistemas operativos	IMS/DB
En red	Principios de la década de 1970	Admite varias formas de acceder a un registro, sin jerarquía estricta	Poor overview with larger databases. En bases de datos más grandes no se tiene una vista general	Grandes ordenadores	UDS (Siemens), DMS (Sperry Univac)
Relacional	1970	Simple, flexible creation and editing, easily expandable, fast commissioning, lively and competitive. Creación y edición fácil y flexible, fácil de ampliar, rápida puesta en marcha, contexto de competencia muy dinámica	Inmanejable con cantidades grandes de datos, segmentación deficiente, atributos de clave artificiales, interfaz de programación externa, no refleja bien las propiedades y la conducta de los objetos	Control de gestión (controlling), facturación, sistemas de control de inventario, sistemas de gestión de contenido, etc.	MySQL, PostgreSQL, Oracle, SQLite, DB2, Ingres, MariaDB, Microsoft Access
Orientado a objetos	Final de la década de 1980	Best support of object-oriented programming languages, storage of multimedia content. Soporta mejor los lenguajes de programación orientados a objetos, permite almacenar contenido multimedia	Increasingly poorer performance with large data volumes, few compatible interfaces. El rendimiento empeora con grandes volúmenes de datos, pocas interfaces compatibles	Inventario (museos, comercio minorista)	db4o
Orientado a documentos	1980s Década de 1980	Los datos relacionados se guardan de forma centralizada en documentos independientes, estructura libre, concepción multimedia	El trabajo de organización es relativamente alto, a menudo requiere conocimientos de programación	Aplicaciones web, buscadores, bases de datos de texto	Lotus Notes, Amazon SimpleDB, MongoDB, CouchDB, Riak, ThruDB, OrientDB