

2. Historia y evolución de los SGBD II (Resumen)

5. Década de los 80 – SGBD Relacionales.

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 debido a que su nivel de programación era sencillo y relativamente bajo.

Los SGBD de los años setenta eran demasiado complejos e inflexibles, y sólo los podía utilizar un personal muy cualificado.

Durante la década de los 80 aparecen y se extienden muy rápidamente los ordenadores personales, también surgen software para equipos monousuario con los cuales es muy fácil utilizar y crear conjunto de datos, los cuales se denominan **personal data bases**.

En la década de los años 80', se desarrolló el SQL (Structured Query Language), un lenguaje de consultas que permite consultar, valga la redundancia, con el fin de recuperar información de una base de datos y a su vez, hacer cambios sobre esa misma base, de forma sencilla. Permitía analizar gran cantidad de información y especificar varios tipos de operaciones.

SQL comenzó a ser el modelo estándar de las industrias, con su base de datos bajo un sistema de tablas (filas y columnas), pudo competir con las bases jerárquicas y de redes, ya que su nivel de programación era sencillo y el nivel era relativamente bajo.

El ORACLE está considerado como uno de los sistemas de bases de datos más completos del mundo, su dominio en el mercado fue casi total hasta muchos años después, pero esto cambió con la aparición del SQL Server de Microsoft.

6. Década de los 90: Distribución, C/S y 4GL

A finales de los ochenta y principios de los noventa, las empresas se han encontrado con el hecho de que sus departamentos han ido comprando ordenadores departamentales y personales, y han ido haciendo aplicaciones con BD. El resultado ha sido que en el seno de la empresa hay numerosas BD y varios SGBD de diferentes tipos o proveedores.

Los SGDB actuales permiten que un programa pueda trabajar con diferentes BD como si se tratase de una sola, esto se lo conoce como bases de datos distribuida.

Esta distribución ideal se consigue cuando las diferentes BD son soportadas por una misma marca de SGBD, es decir, cuando hay homogeneidad.

En la actualidad, gracias principalmente a la estandarización del lenguaje SQL, los SGBD de marcas diferentes pueden darse servicio unos a otros y colaborar para dar servicio a un programa de aplicación.

Además de esta distribución "impuesta", al querer tratar de forma integrada distintas BD preexistentes, también se puede hacer una distribución "deseada", diseñando una BD distribuida físicamente, y con ciertas partes replicadas en diferentes sistemas.

Las razones básicas por las que interesa esta distribución son las siguientes:

Disponibilidad

La disponibilidad de un sistema con una BD distribuida puede ser más alta, porque si queda fuera de servicio uno de los sistemas, los demás seguirán funcionando.

Coste

Una BD distribuida puede reducir el coste. En el caso de un sistema centralizado, todos los equipos usuarios, que pueden estar distribuidos por distintas y lejanas áreas geográficas, están conectados al sistema central por medio de líneas de comunicación. El coste total de las comunicaciones se puede reducir haciendo que un usuario tenga más cerca los datos que utiliza con mayor frecuencia; por ejemplo, en un ordenador de su propia oficina o, incluso, en su ordenador personal.

Para esta distribución de datos se utiliza la arquitectura cliente/servidor.

Un programa de aplicación que un usuario ejecuta en su PC pide ciertos datos de una BD donde, a su vez, se ejecuta el SGBD relacional que la gestiona. El programa de aplicación es el cliente y el SGBD es el servidor.

La facilidad para disponer de distribución de datos no es la única razón, ni siquiera la básica, del gran éxito de los entornos C/S en los años noventa. Tal vez el motivo fundamental ha sido la flexibilidad para construir y hacer crecer la configuración informática global de la empresa, así como de hacer modificaciones en ella, mediante hardware y software muy estándar y barato.

El éxito de las BD ha llevado a la aparición de los Fourth Generation Languages (4GL), lenguajes muy fáciles y potentes, especializados en el desarrollo de aplicaciones fundamentadas en BD.

7. Tendencias actuales

La orientación a objetos (OO) encuentra el éxito al final de los años ochenta, en el desarrollo de software básico, en las aplicaciones de ingeniería industrial y en la construcción de interfaces gráficas con los usuarios, ha hecho que durante la década de los noventa se extendiese en prácticamente todos los campos de la informática.

La utilización de lenguajes como C++ o Java requiere que los SGBD relacionales se adapten a ellos con interfaces adecuadas.

La rápida adopción de la web a los SI hace que los SGBD incorporen recursos para ser servidores de páginas web, como por ejemplo la inclusión de SQL en guiones HTML, SQL incorporado en Java, etc.

Durante estos últimos años se ha empezado a extender un tipo de aplicación de las BD denominado Data Warehouse, o almacén de datos, que también produce algunos cambios en los SGBD relacionales del mercado.

Los datos de este gran almacén, el Data Warehouse, se obtienen por una replicación más o menos elaborada de las que hay en las BD que se utilizan en el trabajo cotidiano de la empresa. Estos almacenes de datos se utilizan exclusivamente para hacer consultas, de forma especial para que lleven a cabo estudios los analistas financieros, los analistas de mercado, etc.

Actualmente, los SGBD se adaptan a este tipo de aplicación, incorporando, por ejemplo, herramientas como las siguientes:

La creación y el mantenimiento de réplicas, con una cierta elaboración de los datos.

La consolidación de datos de orígenes diferentes.

La creación de estructuras físicas que soporten eficientemente el análisis multidimensional.

Gestores de Bases de Datos

Las aplicaciones más usadas son para gestiones de empresas e instituciones públicas, así como en entornos científicos, para almacenar la información experimental.

Una base de datos es un sistema compuesto por un conjunto de datos, los cuales están almacenados en discos, a los que se accede directamente y un conjunto de programas que regulen o manejen ese conjunto de datos.

Mientras que un sistema de Gestión de Bases de Datos es un software que sirve de interfaz entre la base de datos, el usuario y las aplicaciones que se utilizan.

Los mejores gestores de base de datos

El principal lenguaje de base de datos y el más utilizado desde que se conoce la programación de gestión, es el Structured Query Language (SQL). Este, de consulta estructurada, facilita el acceso a la gestión de las bases de datos relaciones, lo que permite realizar tareas en ellas y realizar consultas, que sirvan para obtener, agregar, eliminar o modificar información.

Para el desarrollo de este lenguaje hay que utilizar un gestor de base de datos, de los que hay muchos, unos de acceso libre y otros de pago. Veamos cuáles son, primeramente, los gestores de base de datos de pago:

Oracle

*El **sistema** de gestión de base de datos relacional más utilizado en el mercado.*

Es propiedad de Oracle Corporation y fue desarrollado en 1977.

Su versatilidad le facilita ejecutarse en casi todas las plataformas existentes, Windows, Unix, Linux, MAC OS, entre otros.

SQL Server

En competencia directa a Oracle, está SQL Server de Microsoft. Los dos ocupan gran parte del mercado en el sector de base de datos. Son muy parecidos en algunas de sus características y funciones, aunque tienen sus marcadas diferencias.

Gestores de base de datos de acceso libre

MySQL

Este es de simple instalación y actúa de lado del cliente o servidor, es de código abierto y tiene licencia comercial disponible. Pertenece a Oracle Corporation y gestiona las bases de datos relacionales, con funciones multiusuario y es el más usado dentro del software libre.

Fire Bird

De gran potencia y muy sencillo a la vez, este sistema de gestión de base de datos relacional SQL, es uno de los mejores gestores Open Source (Código abierto) o libres. Es compatible con Windows y Linux.

Tiene buen soporte para los procedimientos almacenados, las transacciones compatibles con ACID y con los métodos de acceso múltiple como Nativo, Python, .NET, etc...

Como vemos, son múltiples las posibilidades que tenemos de acceso a gestores de base de datos, tanto adquiriendo licencias de pago como acudiendo a software libre. En función de los gustos, formas de trabajar y necesidades de cada uno, seguro encontraremos distintos gestores de base de datos que pueden satisfacernos en pro de nuestro trabajo.