

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**DISEÑO E IMPLEMENTACIÓN
DE BASE DE DATOS**

Tema

**EVOLUCIÓN DE LOS SISTEMAS DE BASE DE
DATOS, ARQUITECTURA DE LOS SISTEMAS
DE BASES DE DATOS. IMPORTANCIA DE LAS
BASES DE DATOS.**

Evolución de los Sistemas de Gestión de Bases de Datos



Las bases de datos han estado en uso desde los primeros días de las computadoras electrónicas. A diferencia de los sistemas modernos, que se pueden aplicar a datos y necesidades muy diferentes, la mayor parte de los sistemas originales estaban enfocados a bases de datos específicas y pensados para ganar velocidad a costa de perder flexibilidad. Los SGBD originales sólo estaban a disposición de las grandes organizaciones que podían disponer de las complejas computadoras necesarias.

- **Sistemas de navegación (1960)**

Según las computadoras fueron ganando velocidad y capacidad, aparecieron sistemas de bases de datos de propósito general; a mediados de 1960 ya había algunos sistemas en uso. Apareció el interés en obtener un estándar y Charles Bachman —autor de uno de los primeros SGBD, el Integrated Data Store (IDS)— fundó el Database Task Group dentro de CODASYL, el grupo responsable de la creación y estandarización de COBOL. En 1971 publicaron su estándar, que pasó a ser conocido como la «aproximación CODASYL», y en breve aparecieron algunos productos basados en esta línea.

La estrategia de CODASYL estaba basada en la navegación manual por un conjunto de datos enlazados en red. Cuando se arrancaba la base de datos, el programa devolvía un enlace al primer registro de la base de datos, el cual a su vez contenía punteros a otros datos. Para encontrar un registro concreto el programador debía ir siguiendo punteros hasta llegar al registro buscado.

Para responder a preguntas simples como «buscar todas las personas en Japón» el programa debía recorrer todos los datos para escoger los registros correctos. No existían los conceptos «buscar» ni «encontrar», algo que sería inaceptable hoy en día, pero que en los tiempos en que los datos se guardaban en cintas no era viable llevarlos a la práctica.

Se encontraron soluciones a muchos de esos problemas. El fabricante Prime creó un SGBD ajustado a CODASYL basado en árboles binarios que atajaba la navegación de registro en registro proveyendo caminos alternativos de acceso. También aportaba un

lenguaje de interrogación muy claro. De hecho, no hay razón para no poder aplicar los conceptos de normalización a bases de datos CODASYL, pero en último término CODASYL resultaba muy complejo y requería de mucho esfuerzo y práctica para producir una aplicación útil.

IBM también tenía su SGBD propio en 1968, conocido como IMS. Se trataba de un software desarrollado para el programa Apolo sobre System/360. IMS tenía conceptos similares a CODASYL, pero usaba una jerarquía estricta de ordenación de los datos, frente a la estructura en red de CODASYL. Ambos conceptos fueron englobados posteriormente en el concepto de bases de datos de navegación debido al modo de acceso a los datos, de hecho, Bachman recibió al premio Turing en 1973 por su ponencia «El programador como navegador».

- **Sistemas relacionales (1970)**

Edgar Codd trabajaba en IBM, en una de esas oficinas periféricas que estaba dedicada principalmente al desarrollo de discos duros. Estaba descontento con el modelo de navegación CODASYL, principalmente con la falta de operación de búsqueda. En 1970 escribió algunos artículos en los que perfilaba una nueva aproximación que culminó en el documento «A Relational Model of Data for Large Shared Data Banks».

En este artículo descubrió un nuevo sistema para almacenar y trabajar con grandes bases de datos. En vez de almacenar registros de tipo arbitrario en una lista encadenada como en CODASYL, la idea de Codd era usar una «tabla» de registros de tamaño fijo. Una lista encadenada tiene muy poca eficiencia al almacenar datos dispersos donde algunos de los datos de un registro pueden dejarse en blanco. El modelo relacional resuelve esto dividiendo los datos en una serie de tablas —o relaciones— normalizadas, en las que los elementos optativos han sido extraídos de la tabla principal para que ocupen espacio sólo si lo necesitan. En este modelo relacional los registros relacionados se enlazan con una «clave».

Un uso común de las bases de datos puede mantener una agenda de usuarios, su nombre, información de acceso, dirección y teléfono. En la solución de navegación todos esos datos estarían localizados en un solo registro, y las características no usadas simplemente no estarían en la base de datos. En la solución relacional, los datos estarían normalizados en una tabla de usuario, una de teléfono y una de dirección, en la que serían añadidos registros si tuviéramos que incorporar teléfono y dirección.

Reconciliar toda la información es la clave de este sistema. En el modelo relacional, una parte de la información se usa como clave, identificando de manera biunívoca un registro concreto. Cuando se recopila información acerca de un usuario, se accederá a la información de las tablas optativas buscando mediante esa clave. Por ejemplo, si el nombre de usuario es único, la dirección y número de teléfono de ese usuario será guardada con el nombre de usuario como clave. La recopilación de esta información en un solo registro es algo para lo que los lenguajes tradicionales no están pensados.

Así como el enfoque de navegación requiere programas que realicen bucles para recolectar registros, el enfoque relacional también los requerirá. La solución de Codd para los necesarios bucles se basa en un lenguaje orientado a conjuntos, una sugerencia que más tarde cristalizaría en el ubicuo SQL. Planteó el uso de una rama del álgebra

llamada cálculo de tuplas, y demostró que con ella se podrían realizar todas las operaciones típicas sobre una base de datos, además de extraer conjuntos de datos de una forma sencilla.

El artículo de Codd cayó en manos de Michael Stonebraker en Berkeley. Comenzó un proyecto llamado INGRES con fondos asignados a un proyecto de base de datos geográfica programada por los estudiantes. Comenzando en 1973, INGRES produjo sus primeras versiones de prueba que estuvieron listas para uso general en 1979. INGRES era muy similar a System R de IBM en varios aspectos, incluyendo un lenguaje para acceso a los datos, conocido como QUEL. Con el paso del tiempo, INGRES adoptó el estándar SQL.

IBM realizó una implementación de prueba del modelo relacional —PRTV— y una de producción —Business System 12— ambas descontinuadas. Honeywell escribió MRDS para Multics, y aparecen también dos nuevas implementaciones: Alphora Dataphor y Rel. La mayoría de las demás implementaciones de SGBD llamados relativos son en realidad SGBD SQL.

En la década de 1970, la Universidad de Michigan comenzó el desarrollo del MICRO Information Management System basado en el modelo teórico de datos de D. L. Childs. Micro fue utilizado para gestionar gran cantidad de datos en el Departamento de Trabajo del gobierno de EUA. Corría en mainframe usando Michigan Terminal System. Estuvo en producción hasta 1998.

- **Sistemas SQL (finales década de los 70)**

IBM comenzó a trabajar a principios de 1970 en un prototipo lejanamente basado en los conceptos de Codd llamándolo System R. La primera versión estuvo lista en 1974 o 1975, y comenzó así el trabajo en sistemas multitable, en los que los datos podían disgregarse de modo que toda la información de un registro (alguna de la cual es opcional) no tiene que estar almacenada en un único trozo grande. Las versiones multiusuario siguientes fueron probadas por los usuarios en 1978 y 1979, tiempo por el que un lenguaje SQL había sido estandarizado. Las ideas de Codd se revelaron como operativas y superiores a las de CODASYL, lanzando a IBM al desarrollo de una verdadera versión de producción de System R, conocido como SQL/DS, y posteriormente como Database 2 (DB2).

Muchos de los técnicos de INGRES estaban seguros del éxito comercial del sistema, y formaron sus propias compañías para comercializar el desarrollo, pero con una interfaz SQL. Sybase, Informix, NonStop SQL y la misma INGRES se vendían como derivados del INGRES original en los años 1980. Incluso el SQL Server de Microsoft está basado en Sybase, y por consiguiente en INGRES. Sólo Larry Ellison —el fundador de Oracle— comenzó un nuevo camino basado en el artículo de IBM sobre System R, y aventajó a IBM sacando al mercado su primera versión en 1978, llamado «un modelo relacional de datos para grandes bancos de datos compartidos».

Stonebraker aplicó las lecciones de INGRES al desarrollo de una nueva base de datos —Postgres— conocida ahora como PostgreSQL. PostgreSQL se utiliza para muchas aplicaciones críticas (los registros de dominios .org y .info lo usan para su almacenamiento primario, así como grandes compañías e instituciones financieras).

En Suecia, el artículo de Codd generó la base de datos Mimer SQL en la universidad de Uppsala. En 1984 este proyecto se consolidó en una compañía independiente. A principios de 1980, Mimer introdujo la gestión de transacciones para dar robustez a las aplicaciones, una idea que fue recogida en muchos otros SGBD.

- **Sistemas orientados a objetos (1980)**

Durante la década de 1980 el auge de la programación orientada a objetos influyó en el modo de manejar la información de las bases de datos. Programadores y diseñadores comenzaron a tratar los datos en las bases de datos como objetos. Esto quiere decir que, si los datos de una persona están en la base de datos, los atributos de la persona como dirección, teléfono y edad se consideran que pertenecen a la persona, no son datos extraños. Esto permite establecer relaciones entre objetos y atributos, más que entre campos individuales.

Otro gran foco de atención durante la década fue el incremento de velocidad y fiabilidad en el acceso. En 1989, dos profesores de la Universidad de Wisconsin publicaron un artículo en una conferencia ACM en el que exponían sus métodos para mejorar las prestaciones de las bases de datos. La idea consistía en replicar la información importante —y más solicitada— en una base de datos temporal de pequeño tamaño con enlaces a la base de datos principal. Esto implicaba que se podía buscar mucho más rápido en la base de datos pequeña que en la grande. Su mejora de prestaciones llevó a la introducción de la indexación, incorporado en la totalidad de los SGBD.

- **Sistemas NoSQL (2000)**

El siglo XXI trajo una nueva tendencia en las bases de datos: el NoSQL. Esta tendencia introducía una línea no relacional significativamente diferentes de las clásicas. No requieren por lo general esquemas fijos, evitan las operaciones join almacenando datos desnormalizados y están diseñadas para escalar horizontalmente. La mayor parte de ellas pueden clasificarse como almacenes clave-valor o bases de datos orientadas a documentos.

Recientemente ha habido una gran demanda de bases de datos distribuidas con tolerancia a particiones, pero de acuerdo con el teorema CAP no es posible conseguir un sistema distribuido que simultáneamente proporcione consistencia, disponibilidad y tolerancia al particionado. Un sistema distribuido puede satisfacer sólo dos de las tres restricciones a la vez. Por dicha razón muchas de las bases de datos NoSQL usan la llamada consistencia eventual para proporcionar disponibilidad y tolerancia al particionado, con un nivel máximo de consistencia de datos.

Entre las aplicaciones más populares encontramos MongoDB, MemcacheDB, Redis, CouchDB, Hazelcast, Apache Cassandra y HBase, todas ellas de código abierto.

- **Sistemas XML (2010)**

Las Bases de Datos XML forman un subconjunto de las Bases de Datos NoSQL. Todas ellas usan el formato de almacenamiento XML, que está abierto, legible por humanos y máquinas y ampliamente usado para interoperabilidad.

En esta categoría encontramos: BaseX, eXist, MarkLogic Server, MonetDB/XQuery, Sedna.



Bibliografía:

<https://bbddrafaelcampanero.wordpress.com/2016/09/29/evolucion-de-los-sistemas-de-gestion-de-bases-de-datos/>

