



## **Unidad I: Introducción a las bases de datos**

### **1.1 Tecnología de los RDBMS**

#### **1.1.1 ¿Qué es un sistema de gestión de bases de datos?**

Un sistema de administración de bases de datos (DBMS) es un software diseñado para almacenar, recuperar y administrar datos en bases de datos. El sistema DBMS más común es el RDBMS. La forma completa de RDBMS es el sistema de gestión de bases de datos relacionales. Ahora que está claro qué es un sistema de administración de bases de datos, aprendamos sobre la solución de administración de bases de datos relacionales.

#### **1.1.2 Por qué necesitamos una base de datos relacional**

Hoy en día, varias empresas utilizan una arquitectura de base de datos relacional en lugar de archivos planos o bases de datos jerárquicas para el sistema de gestión de bases de datos de su empresa (DBMS). Entonces, ¿cuál es la razón para crear una base de datos relacional? Una base de datos relacional está diseñada específicamente para manejar una amplia gama de formatos de datos y procesar consultas de manera eficiente. ¿Y cómo se organizan los datos en un sistema de base de datos relacional? La respuesta a esto es simple: un sistema de base de datos relacional organiza los datos en tablas que se pueden vincular internamente dependiendo de los datos comunes. Esto permite que un usuario recupere una o más tablas fácilmente con una sola consulta. Por otro lado, el archivo plano almacena datos en una estructura de tabla única, que es menos eficiente y consume más espacio y memoria. Por

tanto, necesitamos una base de datos relacional. Un ejemplo de un sistema de gestión de base de datos relacional podría ser un departamento de producción en una organización que aprovecha este modelo para procesar compras y realizar un seguimiento del inventario.

El sistema de gestión de bases de datos o sistema de gestión relacional más disponible comercialmente y en toda la empresa que se utiliza en la actualidad es el lenguaje de consulta estructurado (SQL) para acceder a la base de datos. Otros sistemas de administración de bases de datos relacionales ampliamente utilizados para empresas incluyen Oracle, MySQL, PostgreSQL y Microsoft SQL Server. Las estructuras RDBMS se utilizan comúnmente para realizar cuatro operaciones básicas: CRUD (crear, leer, actualizar y eliminar), que son fundamentales para respaldar una gestión de datos coherente.

### **1.1.3 ¿Cómo funciona un RDBMS?**

Una base de datos relacional almacena datos en forma de varias tablas. Aquí surge una pregunta clave: ¿cómo funciona una estructura de base de datos y cómo se implementa? Entendamos esto en detalle.

Una estructura de base de datos funciona organizando cada tabla en filas (conocidas como registros o tuplas) y columnas (conocidas como campos o atributos). Las tablas, columnas y filas son los tres componentes principales de una base de datos relacional.

A continuación, se muestra un ejemplo sencillo de una base de datos. La primera columna contiene el ID de la empresa, que es la clave principal. La clave principal se utiliza para identificar cada registro único en una tabla. Cada columna (o atributo) almacena bits de información y la base de datos asigna un ID de empresa único a cada fila.

Un RDBMS generalmente viene con diccionarios de datos y colecciones de metadatos, que son beneficiosos para la gestión de datos. Un diccionario de datos define los objetos de datos de cada usuario dentro de la base de datos. Como resultado, ayuda a los usuarios a identificar todos los objetos que existen en la base de datos y quién puede acceder a ellos.

Por ejemplo, es humanamente imposible recordar todas las tablas almacenadas en una enorme base de datos. Si desea buscar una tabla específica, como Productos, puede utilizar un diccionario de datos para buscar todas las tablas denominadas Productos. Luego, de la lista de tablas resultantes, puede identificar la requerida.

#### **1.1.4 Ventajas de RDBMS**

Las ventajas de un sistema de gestión de bases de datos relacionales ofrecen una visión metódica de los datos, lo que ayuda a las empresas a mejorar sus procesos de toma de decisiones al mejorar diferentes áreas.

a) Seguridad de datos mejorada

Las funciones de autorización y control de acceso del software de bases de datos relacionales admiten el cifrado y descifrado avanzados, lo que permite a los administradores de la base de datos gestionar el acceso a los datos almacenados. Esto ofrece importantes beneficios en términos de seguridad. Además, los operadores pueden modificar el acceso a las tablas de la base de datos e incluso limitar los datos disponibles a otros. Esto convierte a los RDBMS en una solución de almacenamiento de datos ideal para empresas en las que la alta dirección necesita controlar el acceso a los datos para los trabajadores y los clientes.

b) Retener la consistencia de datos

Es más fácil agregar nuevos datos o modificar tablas existentes en un RDBMS mientras se mantiene la coherencia de los datos con el formato existente. Esto se debe principalmente a que un RDBMS es compatible con ACID.

c) Mejor flexibilidad y escalabilidad

Un RDBMS ofrece más flexibilidad a la hora de actualizar los datos, ya que las modificaciones solo deben realizarse una vez. Por ejemplo, actualizar los detalles en la tabla principal actualizará automáticamente los archivos relevantes y le evitará la molestia de cambiar varios archivos uno por uno. La escalabilidad se refiere a la capacidad que posee un

software para adaptarse a las necesidades o las demandas de rendimiento a medida que el número de usuarios y operaciones crece. Generalmente al incrementarse ampliamente las transacciones en un sistema, el funcionamiento de las bases de datos empieza a verse afectado.

d) Fácil mantenimiento

Las bases de datos relacionales se consideran de bajo mantenimiento porque los usuarios pueden probar, regular, corregir y realizar copias de seguridad de los datos rápidamente, ya que la herramienta de automatización en RDBMS ayuda a sistematizar estas tareas.

e) Riesgo reducido de errores

En el software de bases de datos relacionales, puede verificar fácilmente si hay errores en los datos de diferentes registros. Además, como cada elemento de datos se almacena en una sola ubicación, no hay posibilidad de que versiones anteriores difuminen la imagen.

## **1.2 Evolución de los RDBMS**

### **1.2.1 Historia de la base de datos: Línea del tiempo**

El término Base de Datos se usó por primera vez en un simposio celebrado en California, en el año 1963. Su origen se remonta a la antigüedad, cuando ya existían bibliotecas y registros acumulados de hechos y situaciones, de forma escrita y gráfica. Obviamente, por falta de recursos tecnológicos, la búsqueda y recopilación de información era mucho más lenta que hoy día. No había máquinas que ayudaran y pudieran reemplazar el trabajo manual.

Pero mucho ayudaron nuestros antepasados, ya que guardaron información muy valiosa, que sin ella sería muy difícil explicar nuestra existencia desde aquellos tiempos remotos.

Una base de datos se puede comparar a un almacén, en el que se guardan todo tipo de informaciones, clasificándola según la utilización que se le vaya a dar, por ejemplo, listados de cosas, estudiantes, nóminas, mercancías, en fin, lo que queramos dejar guardado o archivado.

Luego, cuando el volumen de datos e informaciones aumentó, hubo la necesidad de mejorar la capacidad de almacenamiento. Fueron apareciendo las primeras computadoras, que facilitaron inmensamente el trabajo de guardar gran cantidad de datos, la informática siempre ha estado ligada al concepto de Base de Datos.

### **1.2.2 Máquina de tarjetas perforadas**

En 1884, los censos se realizaban de forma manual, hasta que Herman Hollerith inventó la máquina automática de perforación de tarjetas, que se usó en el censo de los Estados Unidos, mejorando significativamente el proceso de terminación, de siete años a dos años y medio.

Mientras que, en la década de 1950, se cambia a un sistema de lectura secuencial y ordenada. El inglés Oberlin Smith, con este mecanismo, dio inicio a la automatización de la información

referente a las nóminas, a través de cintas magnéticas, que a su vez respaldaban dicha información.

Lo único malo era que sólo se podía hacer de manera secuencial, es decir, si se quería buscar datos de “José” había que llevar la cinta hasta ese punto y se debía volver al principio, si se quería buscar otra información con otra letra o parámetro.

### **1.2.3 Evolución de la Base de Datos**

Una década después, en 1960, las empresas pudieron adquirir computadoras para facilitar sus gestiones. Las empresas informáticas habían bajado los precios de las mismas, para popularizar el uso de los discos, adelanto muy valioso y útil para esa época, ya que se ubicaba la información de manera directa, sin necesidad de saber la ubicación exacta de los datos. Se convirtió en una opción más rentable para las empresas pequeñas, medianas y grandes.

También se inició la primera generación de bases de datos de red (CODASYL) y las jerárquicas (IMS), que consistían en guardar las estructuras de datos en listas y árboles, además de que permitió crear un estándar en las bases de datos, gracias a los nuevos lenguajes implementados en los sistemas de información.

### **1.2.4 Década del 70**

Hay valiosos aportes, como los de Edgar Frank Codd, científico informático inglés, quien definió el modelo relacional. El multimillonario Lawrence “Larry” Ellison, pudo desarrollar el Relational Software System o sistema de datos ORACLE, aprovechando esa información de Codd. Este consistió en un sistema de administración de Base de Datos relacionados, el cual se destacaba por su estabilidad, escalabilidad, transacciones y multiplataforma.

Codd, además de definir el modelo relacional, publicó una serie de reglas, a través de su artículo “Un modelo relacional de datos para grandes bancos de datos compartidos”, que evaluarían los administradores de sistemas de datos, o sea, desconectaba el almacenamiento de la información física, naciendo así las bases de datos relacionales.

Inicialmente no se usó porque tuvo inconvenientes con el rendimiento, no podía competir con las bases de datos jerárquicas y de redes. Finalmente IBM desarrolló unas técnicas para construir un sistema de bases de datos relacionales eficientes, las cuales llamó System R; por otro lado Ingres se desarrolló en la UBC en los años de 1974 a 1977.

Ingres utilizaba un lenguaje de consulta, llamado QUEL, dando pie a la creación de sistemas como Ingres Corporación, MS SQL Server, Sybase, PACE Wang, y Britton Lee-. Por su parte, el Sistema R utilizó el lenguaje de consulta Secuela, el cual ha contribuido al desarrollo de SQL / DS, DB2, Allbase, Oracle y SQL Non-Stop. En esta década el término Relational Database Management System, o RDBMS, fue ampliamente reconocido. Con esto se abrió paso al nacimiento de la segunda generación de los Sistemas Gestores de Bases de Datos.

### **1.2.5 Años 80': Comercialización de sistemas relacionales**

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 con la misma información, a diferencia de los años anteriores, cuando se diseñaron aplicaciones de procesamiento de transacciones.

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.

Estos sistemas de bases de datos relacionales fueron un éxito comercial, así como el aumento en la venta de ordenadores, estimulando el mercado de bases de datos, lo que provocó una caída importante en la popularidad de las redes y los modelos jerárquicos de bases de datos.

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. La oferta de otros Sistemas Administradores de Bases de Datos Relacionales, como PostgreSQL, MySQL o Firebird aparecieron

posteriormente en la década de 1990. Igualmente se da inicio a las bases de datos que se orientaban a los objetos.

### **1.2.6 Gestores de Bases de Datos**

Hoy día la mayoría de las bases de datos se presentan en formato digital, gracias a los avances tecnológicos en la informática y la electrónica. Esto ofrece un amplio abanico de soluciones al problema de almacenamiento de datos.

Los gestores de bases de datos, Database Management System o DBMS (SGBD) son programas que permiten almacenar y luego acceder a los datos de forma estructurada y rápida. 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.

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:

a) Oracle

Es de los más confiables sistemas de gestión de base de datos relacional, además del más usado. Es propiedad de Oracle Corporation y fue desarrollado en 1977. Se accede



directamente a los objetos, a través del lenguaje de consulta SQL, es muy utilizado en las empresas, con un componente de red que permite la comunicación a través de las redes.

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

#### b) 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.

SQL Server se ejecuta en Transact-SQL, esto es un grupo de programas que pueden añadir características al programa, como tratamiento de errores y excepciones, extracción de datos de la web en forma directa, procesamiento de datos, uso de distintos lenguajes de programación y otros más, que lo hacen un gestor muy completo y competitivo.

Su carácter administrativo es otro valor agregado, tanto en sus funciones y seguridad, como en su flexibilidad.

#### Gestores de base de datos de acceso libre

Dos de los principales y más utilizados gestores de pago, que son de acceso libre (Open Source) son los siguientes:

#### c) 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. Requiere de poca memoria y procesador para su funcionamiento, lo que se traduce en mayor velocidad en sus operaciones. Se usa principalmente para el desarrollo web.

#### d) FireBird

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. Es 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 satisfacerlos en pro de nuestro trabajo.

### **1.2.7 Década 1990**

La investigación en bases de datos se orientó hacia los objetos, con bastante éxito en la gestión de datos complejos. Luego del sacudón de industrias de bases de datos, ahora muchas de las que sobrevivieron venden productos de alta complejidad de bases de datos, a precios muy elevados.

Las herramientas para desarrollar las aplicaciones fueron liberadas, incluyendo el desarrollador de Oracle, PowerBuilder, Visual Basic, como también para la productividad personal, las herramientas como ODBC y Excel / Access, también se desarrollaron.

Con la llegada del Internet la industria de la base de datos creció a un ritmo exponencial. Gran cantidad de usuarios de escritorio comenzaron a usar sistemas cliente-servidor, de bases de datos y acceder a los sistemas informáticos.

Las bases de datos orientadas a los objetos pudieron desarrollarse donde no lo hicieron los datos relacionales de manera eficiente. De esa manera surgieron herramientas como Excel y Access de Microsoft Office y nació la tercera generación de sistemas gestores de bases de datos.

En los 90' también se empezó a modificar la primera publicación de ANSI con respecto al lenguaje SQL, se agregaron nuevas expresiones regulares, triggers, consultas recursivas y

características orientadas a objetos. El boom fue la creación mágica del World Wide Web, a finales de esta década, con el que se facilitaría la consulta a la base de datos.

### **1.2.8 Base de datos en la actualidad**

Hoy día, las compañías que dominan el mercado de las bases de datos son IBM, Microsoft y Oracle. Mientras que en lo que respecta a internet, el dominante es Google, generando gran cantidad de información.

Aun cuando también existe gran cantidad de softwares que permiten crear y manipular bases de datos con gran facilidad, como LINQ de Microsoft, que se asemeja un poco a SQL en cuanto a las consultas nativas en los lenguajes de la plataforma .NET.

El fin primordial del proyecto de LINQ es permitir que el código de Visual Studio sea además, orientado a objetos. Antes de LINQ este manejo de datos externos era mucho más estructurado, más no orientado tanto a los objetos, por esta razón, se preocupa más en facilitar y estandarizar dicho acceso.

Vale acotar que Visual Studio es un entorno de desarrollo integrado para ser usado en sistemas operativos Windows, los que soportan varios lenguajes de programación, como Visual C++, Visual#, Visual J#, ASP.NET y Visual Basic.NET.

Se están desarrollando otras extensiones necesarias, para poder crear aplicaciones, sitios y aplicaciones Web, además de servicios web a todos los entornos que soporte la plataforma .NET, pudiendo interrelacionar aplicaciones entre estaciones de trabajo, páginas web y dispositivos móviles.

Cada día las bases de datos toman tamaños cada vez más grandes y de real importancia en aplicaciones de BigData, Ciencia de Datos, Machine Learning, IA, etc.

## 1.3 Conceptos básicos de las bases de datos

### 1.3.1 Entidad:

Una entidad es una «cosa» u «objeto» en el mundo real que es distinguible de todos los demás objetos. Por ejemplo, cada persona en un desarrollo es una entidad. Una entidad tiene un conjunto de propiedades, y los valores para algún conjunto de propiedades pueden identificar una entidad de forma unívoca. Por ejemplo, el DUI 12345678-9 identifica unívocamente una persona particular en nuestro país. Una entidad puede ser concreta, como una persona o un libro, o puede ser abstracta, como un préstamo, unas vacaciones o un concepto.

### 1.3.2 Conjunto de entidades:

Es un conjunto de entidades del mismo tipo que comparten las mismas propiedades, o atributos. El conjunto de todas las personas que son clientes en un banco dado, por ejemplo, se pueden definir como el conjunto de entidades cliente. Análogamente, el conjunto de entidades préstamo podría representar el conjunto de todos los préstamos concedidos por un banco particular. Las entidades individuales que constituyen un conjunto se llaman la extensión del conjunto de entidades. Así, todos los clientes de un banco son la extensión del conjunto de entidades cliente. Los conjuntos de entidades no son necesariamente disjuntos. Por ejemplo, es posible definir el conjunto de entidades de todos los empleados de un banco (empleado) y el conjunto de entidades de todos los clientes del banco (cliente). Una entidad persona puede ser una entidad empleado, una entidad cliente, ambas cosas, o ninguna. Una entidad se representa mediante un conjunto de atributos.

### 1.3.3 Atributos

Los atributos describen propiedades que posee cada miembro de un conjunto de entidades. La designación de un atributo para un conjunto de entidades expresa que la base de datos almacena información similar concerniente a cada entidad del conjunto de entidades; sin embargo, cada entidad puede tener su propio valor para cada atributo. Posibles atributos del conjunto de entidades cliente son id-cliente, nombre-cliente, calle-cliente y ciudad-cliente. En la vida real, habría más atributos, tales como el número de la calle, el número del portal, la provincia, el código postal, y la comunidad autónoma, pero no se incluyen en el ejemplo simple. Posibles atributos del conjunto de entidades préstamo son número-préstamo e importe.

Cada entidad tiene un valor para cada uno de sus atributos. Por ejemplo, una entidad cliente en concreto puede tener el valor 34546 para id-cliente, el valor “José” para nombre-cliente, el valor “Independencia” para callecliente y el valor “San Salvador” para ciudad-cliente.

### **1.3.4 Relación**

Una relación es una asociación entre diferentes entidades. Por ejemplo, se puede definir una relación que asocie al cliente Juan López con el préstamo P-15. Esta relación especifica que Juan López es un cliente con el préstamo número P-15.

### **1.3.5 Tabla**

Las tablas son objetos de base de datos que contienen todos sus datos. En las tablas, los datos se organizan con arreglo a un formato de filas y columnas, similar al de una hoja de cálculo. Cada fila representa un registro único y cada columna un campo dentro del registro. Por ejemplo, en una tabla que contiene los datos de los empleados de una compañía puede haber una fila para cada empleado y distintas columnas en las que figuren detalles de los mismos, como el número de empleado, el nombre, la dirección, el puesto que ocupa y su número de teléfono particular.

### **1.3.6 Registros**

Son la unidad básica que describe a un objeto o una transacción. Las tablas están formadas por un conjunto de ellos.

### **1.3.7 Campos o Columnas**

El registro está formado por uno o más campos, que son la unidad básica de información, y son una propiedad específica de un objeto o transacción.

Un registro, en el caso de la tabla Clientes, estará formado por el identificador de cliente, el nombre del cliente, y el domicilio del cliente.

### **1.3.8 Dato**

Es la intersección de un campo y un registro. Es toda aquella información que puede almacenarse en una base de datos.

### **1.3.9 Tipo de dato**

Dependiendo de la columna, cada una de ellas tendrá definido un tipo de dato específico que indicará la forma de almacenar, las características y los límites.

Así, hay distintos tipos: los hay numérico, alfanumérico o de tipo cadena, de tipo fecha, etc.

Cada motor de base de datos define distintos tipos de datos, y algunos son específicos de cada una de ellas.

### 1.3.10 Clave primaria

Una clave primaria es una columna o un conjunto de columnas en una tabla cuyos valores identifican de forma exclusiva una fila de la tabla. Una base de datos relacional está diseñada para imponer la exclusividad de las claves primarias permitiendo que haya sólo una fila con un valor de clave primaria específico en una tabla.

### 1.3.11 Clave foránea

Una clave foránea es una columna o un conjunto de columnas en una tabla cuyos valores corresponden a los valores de la clave primaria de otra tabla. Para poder añadir una fila con un valor de clave foránea específico, debe existir una fila en la tabla relacionada con el mismo valor de clave primaria.

## **1.4 Terminología de las bases de datos**

### **1.4.1 Formularios**

Un formulario es un objeto de base de datos que se puede usar para escribir, modificar o mostrar los datos de una tabla o consulta. Los formularios se pueden usar para controlar el acceso a los datos, como qué campos o filas de datos se van a mostrar. Por ejemplo, puede que algunos usuarios necesiten ver sólo algunos de los campos de una tabla que contiene numerosos campos.

Permite definir cómo se agrupan y se ordenan los datos, y usar campos de más de una tabla o consulta siempre y cuando defina con antelación las relaciones entre las tablas y consultas.

### **1.4.2 Consulta (Query)**

En bases de datos, una consulta es el método para acceder a los datos en las bases de datos. Con las consultas se puede modificar, borrar, mostrar y agregar datos en una base de datos. Para esto se utiliza un lenguaje de consultas. El lenguaje de consultas a base de datos más utilizado es el SQL.

### **1.4.3 Informe**

Los informes son resúmenes de la información de la base de datos que puedes consultar en pantalla o mediante el uso de una impresora. Los informes pueden ser una simple lista de los registros de una tabla, o los más complejos que incluyen varias tablas, resumen de totales y tablas o gráficos. Determina qué campos y tablas contienen la información que necesitas, y utilizar el asistente de informes o la ventana de diseño de informes para crear para crearlos.



#### 1.4.4 Macro

Una macro es un conjunto de comandos que se almacena en un lugar especial de Excel de manera que están siempre disponibles cuando los necesites ejecutar.

Por ejemplo, si todas las mañanas creas un reporte de ventas y en ese reporte siempre das el mismo formato a los textos, se podría crear una macro para que lo haga automáticamente por ti. Las macros se utilizan principalmente para eliminar la necesidad de repetir los pasos de aquellas tareas que realizas una y otra vez.

#### 1.4.5 Procedimiento Almacenado

Un procedimiento almacenado (stored procedure en inglés) es un programa (o procedimiento) almacenado físicamente en una base de datos. Su implementación varía de un gestor de bases de datos a otro. La ventaja de un procedimiento almacenado es que al ser ejecutado, en respuesta a una petición de usuario, es ejecutado directamente en el motor de bases de datos, el cual usualmente corre en un servidor separado. Como tal, posee acceso directo a los datos que necesita manipular y sólo necesita enviar sus resultados de regreso al usuario, deshaciéndose de la sobrecarga resultante de comunicar grandes cantidades de datos salientes y entrantes.

#### 1.4.6 Trigger

Un trigger o disparador es un script que se usa en lenguaje de programación SQL, en especial en bases de datos como MySQL o PostgreSQL.

Consiste en una serie de reglas predefinidas que se asocian a una tabla. Estas reglas se aplican a la base de datos cuando se realizan determinadas operaciones en la tabla, por ejemplo, al añadir, actualizar o eliminar registros.

Dicho de otra manera, el trigger desencadena determinadas acciones de forma automática en las tablas de la base de datos cuando se insertan, modifican y se añaden nuevos datos.

### 1.4.7 Índice

Un índice es una estructura diferente dentro de la base de datos; creado con el comando `create index`. Requiere su propio espacio en disco y contiene una copia de los datos de la tabla. Eso significa que un índice es una redundancia. Crear un índice no cambia los datos de la tabla; solamente establece una nueva estructura de datos que hace referencia a la tabla. De hecho, un índice de base de datos se parece mucho a un índice de un libro: ocupa su propio espacio, es redundante y hace referencia a la información actual almacenada en otro lugar.

## 1.5 Modelo de las Bases de Datos

### 1.5.1 Qué es un modelo de base de datos?

Un modelo de base de datos es la estructura lógica que adopta la base de base datos, incluyendo las relaciones y limitaciones que determinan cómo se almacenan y organizan y cómo se accede a los datos. Así mismo, un modelo de base de datos también define qué tipo de operaciones se pueden realizar con los datos, es decir, que también determina cómo se manipulan los mismos, proporcionando también la base sobre la que se diseña el lenguaje de consultas.

En general, prácticamente todos los modelos de base de datos pueden representarse a través de un diagrama de base de datos (en esta entrada veremos algunos).

### 1.5.2 Tipos de modelos de bases de datos

Tal y como ocurre con las bases de datos, existen diferentes tipos de modelos de bases de datos, como vamos a ver en los próximos puntos. Qué modelo elegir para nuestra base de datos dependerá, por un lado, del sistema de gestión de bases de datos que estemos usando, puesto que este debe ser compatible con el modelo de datos.

#### a) Modelo de base de datos relacional

El modelo de base de datos relacional es uno de los más comunes. Este modelo es el que emplean las bases de datos relacionales y ordena los datos en tablas (relaciones) compuestas por columnas y filas.

Cada columna alberga un atributo de la entidad (nombre, dirección, fecha de nacimiento...); a los atributos de una relación se los llama dominio. Escogiendo un atributo en concreto o una combinación de varios tenemos una clave primaria, a la que se puede hacer referencia en otras tablas, en las que será un clave externa.

En cada fila (tupla) se incluyen datos sobre una instancia específica de la entidad (por ejemplo, un cliente específico).

Además, el modelo también representa el tipo de relaciones entre las tablas, que pueden ser uno a uno, uno a muchos o muchos a muchos.

#### b) Modelo jerárquico

Si vamos a emplear una base de datos jerárquica, el modelo de datos que emplearemos será el jerárquico, que se caracteriza por presentar los datos en una estructura de árbol invertido, donde cada registro tiene un único nodo raíz, del que surgen otros nodos (registros); los nodos en un mismo nivel son nodos padre, cada nodo padre tiene el mismo nodo raíz, y puede tener nodos hijos, pero los nodos hijos solo pueden tener un nodo padre. Este modelo se emplea poco actualmente.

En este modelo, los registros de un mismo nivel se clasifican en un orden específico.

#### c) Modelo de red

El modelo en red de base de datos parte del modelo jerárquico, pero aquí se permiten las relaciones de uno a muchos o de muchos a muchos entre registros vinculados, teniendo registros principales múltiples. El modelo se crea a través de conjuntos de registros relacionados; cada uno de estos conjuntos consiste en un registro propietario o principal y uno o más registros miembros o secundarios. Además, un registro puede ser miembro o secundario en diferentes conjuntos. Es decir, que en este modelo se permite que los nodos hijos tengan más de uno nodo padre, de manera que se pueden representar relaciones más complejas.

#### d) Modelo orientado a objetos

El modelo de la base de datos orientada a objetos define la base de datos como una colección de objetos utilizados en la programación orientada a objetos (es decir, que emplear lenguajes

como C++ o Java, por ejemplo). Este modelo de base de datos utiliza tablas también, pero no solo se limita a ellas y permite almacenar información muy detallada sobre cada objeto.

Los objetos se dotan de un conjunto de características propias, que a su vez les diferencian de objetos similares. Los objetos similares pueden agruparse en una clase y cada objeto de esta es una instancia. Las clases intercambian datos entre sí a través métodos (mensajes).

e)      Modelo entidad-relación

El modelo entidad-relación es básicamente el paso previo a uno modelo de bases datos relacional, puesto que se trata de un diagrama elaborado a través de unos elementos básicos y su relación entre ellos.

f)      Modelo de archivo invertido

El modelo de archivo invertido, también llamado de índice invertido, contiene datos que se usan como claves en una tabla de consulta, los valores en la tabla se emplear como punteros a la localización de cada instancia. Actualmente se emplea como índice de las bases modernas de datos, ya que se trata de tablas de consulta en las que se introduce un contenido mínimo.

g)      Modelo plano

El modelo de bases de datos plano, los datos se estructura en dos dimensiones (de hay lo de estructura plana), en la que todos los objetos en una columna concreta tienen valores del mismo tipo y todos los objetos de la misma fila están relacionados entre ellos.

Por ejemplo, en una base de datos que recoja solo el nombre de usuario y la contraseña, cada fila recogerá el nombre y la contraseña correspondiente para cada usuario.

Este modelo tabular de base de datos es el precursor del modelo relacional.

h)      Modelo multidimensional

El modelo de base de datos multidimensional está pensado para la creación de aplicaciones específicas OLAP (procesamiento analítico en línea). Este modelo de base de datos se puede visualizar como un cubo de datos en el que se representan diferentes dimensiones de los datos disponibles; las dimensiones de los cubos se corresponden con la de la tabla y el valor almacenado en cada celda equivale al valor de la métrica.

Este modelo combina el modelo relacional con el jerárquico y permite extraer datos de forma selectiva y eficaz. Sin embargo, tiene un inconveniente, y es que no se puede modificar su estructura, lo que obliga a diseñarlos desde cero cuando hay que introducir cambios.

i)      Modelo semiestructurado

Cuando los datos no encajan en el formato de tablas, filas y columnas, sino que se organizan mediante etiquetas con las que se los puede agrupar y crear jerarquías (como ocurre con los datos de una página web), estamos hablando de datos semiestructurados.

Para ordenar estos datos en una base de datos, podemos recurrir al modelo semiestructurado. En este modelo no hay separación entre los datos y el esquema y resulta útil para describir sistemas o describir interacciones entre bases de datos no que se acogen al mismo esquema.

j)      Modelo de contexto

El modelo de contexto se puede usar cuando se necesita incorporar elementos de otros modelos de bases de datos. Puede adoptar elementos de modelos orientados a objetos, semiestructurados y en red.

k)      Modelo asociativo

En modelo asociativo los datos se dividen en entidad y asociación, de manera que una entidad es todo lo que existe de manera independiente y una asociación es algo que solo existen en relación a algo más. Así, los datos se estructuran en dos grupos:

Elementos, donde cada uno tiene un identificador único, un nombre y un tipo.

Enlaces, donde cada uno tiene un identificador único e identificadores únicos de una fuente, verbo u objeto. La información almacenada está relacionada con la fuente y cada uno de los tres identificadores pueden hacer referencia a un enlace o a un elemento.

#### 1) Modelos de bases de datos NoSQL

También llamadas bases de datos no relacionales, el modelo de base de datos orientado a objetos, pero existen otros:

Modelo de base de datos gráfico, similar al de red, pero más flexible, puesto que permite que cualquier nodo se pueda conectar a cualquier otro.

Modelo multivalor, que nace del modelo relacional, pero en el que los atributos pueden contener una lista de datos en vez de un solo punto de datos.

Modelo de documentos, empleado para almacenar y administrar documentos o datos semiestructurados, en vez de datos atómicos (como hacen las bases relacionales).

## 1.6 Clasificación de las Bases de Datos

Hoy en día hay numerosas BBDD disponibles en función de su estructura, contexto, utilidad y necesidades que se busquen. Veamos a continuación los principales tipos de BBDD que existen.

### 1.6.1 Según la flexibilidad de modificación

La primera clasificación de bases de datos depende de la manera en la que se ordenan los datos.

- a) Bases de datos estáticas: las BBDD estáticas están diseñadas para la lectura de datos. En otras palabras, sólo almacenan y registran los datos. Luego se pueden analizar para comprender su comportamiento a lo largo del tiempo. Son especialmente utilizadas para implementar proyecciones estadísticas y orientar procesos de tomas de decisiones en el ámbito empresarial.
- b) Bases de datos dinámicas: las BBDD dinámicas son, por el contrario, modificables con el paso del tiempo. Así, los datos pueden actualizarse, editarse y eliminarse. Por ejemplo, muchas tiendas varían sus inventarios y los precios de sus productos según la temporada, por lo que una BBDD dinámica es ideal.

### 1.6.2 Según el contenido

La segunda clasificación de bases de datos depende de la prioridad del contenido a analizar.

- a) Bases de datos bibliográficas: Las BBDD bibliográficas son registros que ayudan a clasificar varios campos de datos. Generalmente, dichos campos se pueden consultar de modo separado o conjunto. Un claro ejemplo puede ser la información sobre un libro: autor, año de publicación, editorial, etc.



b) Bases de datos de texto completo: Entre los tipos de bases de datos, las de texto completo son especialmente útiles, puesto que permiten buscar términos específicos, palabras claves y las diversas opciones de una base de datos bibliográfica, además de consultar el texto íntegro almacenado. Son idóneas para trabajos académicos y de investigación.

c) Directorios: Los directorios son BBDD usadas por la mayoría de la gente casi a diario sin darse cuenta. Un ejemplo claro puede ser la agenda de contactos de nuestros teléfonos móviles, donde se almacena multitud de información como:

Nombres y direcciones

Número telefónico y direcciones de email

Datos de facturación, códigos postales

## Ayuda Bibliográfica

<https://ayudaleyprotecciondatos.es/bases-de-datos/modelos/>

<https://www.grapheverywhere.com/tipos-bases-de-datos-clasificacion/>

<https://www.manejandodatos.es/2020/08/conceptos-basicos-de-bases-de-datos/>

<https://www.astera.com/es/type/blog/relational-database-management-system/>

<https://conocelahistoria.com/historia-de-la-base-de-datos/>

<https://aws.amazon.com/es/nosql/>