

Diseño de bases de datos relacionales

2023

Bravo Ainara, Bravo Iara



Laboratorio IV

Contenido

Descripción de la arquitectura, las características y sus funcionalidades	3
Elementos principales de una base de datos relacional	4
Características de una base de datos relacional	5
Ventajas y desventajas de una base de datos relacional	6
Comparación contra los otros tipos de datos, cuándo conviene usarla y cuándo no	7
Bases de datos jerárquicas	9
Bases de datos transaccionales	9
Manejadores o gestores de bases de datos relacionales	10
Instalación del motor de la BD	12

Descripción de la arquitectura, las características y sus funcionalidades

Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas.

Una base de datos relacional es un conjunto de tablas (o relaciones) formadas por filas (registros) y columnas (campos); así, cada registro (cada fila) tiene una ID única, denominada clave y las columnas de la tabla contienen los atributos de los datos. Cada registro tiene normalmente un valor para cada atributo, lo que simplifica la creación de relaciones entre los puntos de datos.

Las bases de datos relacionales son útiles para cualquier necesidad de información en la que los puntos de datos se relacionan entre sí y también se deben administrar de forma coherente, segura y basada en reglas. Esto es lo que hace que las bases de datos relacionales sean más populares para negocios y empresas. Cuando las empresas quieren extraer conclusiones de sus propios datos, se basan en bases de datos relacionales para generar análisis útiles. Muchos de los informes que generan las empresas para realizar un seguimiento del inventario, las finanzas, las ventas o realizar proyecciones futuras se crean mediante bases de datos relacionales.

La metodología de diseño de bases de datos relacionales se ha consolidado a lo largo de los años satisfaciendo las propiedades de generalidad (independencia de la plataforma hardware/software), calidad del producto (precisión, completitud y eficacia) y facilidad de uso. Consta de las siguientes etapas:

1. Diseño conceptual

Su objetivo es definir las entidades y las relaciones entre ellas de manera abstracta, sin enfocarse en un modelo lógico en concreto. La herramienta a usar sería el modelo entidad - relación y el resultado que se obtiene es el diagrama entidad-relación.

2. Diseño lógico

Su objetivo es definir el esquema de la base de datos según el modelo que implementa el SGBD. La herramienta a usar sería el modelo lógico que implemente el sistema de gestión de bases de datos objetivo, pero es independiente de los aspectos físicos. Una de las técnicas que se emplea para verificar la calidad del esquema lógico es la normalización. El resultado de esta etapa es un esquema relacional basado en un modelo relacional.

3. Diseño físico

Su objetivo es generar el esquema físico de la base de datos en el modelo de datos que implementa el SGBD. Esto abarca la definición sobre el SGBD de las tablas con sus campos, la imposición de todas las restricciones de integridad y la definición de índices. La herramienta a utilizar sería el modelo físico de datos (organización de archivos e índices para el SGBD considerado), y el resultado que se consigue es el esquema físico de la base de datos.

Elementos principales de una base de datos relacional

En una base de datos relacional, cada fila en una tabla es un registro con una ID única. Las columnas contienen los atributos de los datos y cada registro suele tener un valor para cada atributo.

1. Relaciones

La estructura de datos del modelo relacional es la relación. Definiremos a una relación como una tabla con columnas y filas. En el modelo relacional, las relaciones se utilizan para almacenar información sobre los objetos que se representan en la base de datos.

En un SGBD relacional hay dos tipos de relaciones:

- **Relaciones base.** Son aquellas que almacenan datos y su implementación es llamada "tabla".
- **Relaciones derivadas.** Son aquellas que no almacenan datos, pero son calculadas al aplicar operaciones, relaciones y su implementación es llamada "consulta" o "vista". Son convenientes por expresar información de varias relaciones actuando como si fuera una sola tabla. Algunas no son determinadas por los usuarios, sino que son inherentemente definidas por ser la BD relacional.

2. Restricciones

- Restricciones semánticas: permite a los usuarios incorporar restricciones personales a los datos
 - Clave primaria: Hace que los atributos marcados como clave primaria no puedan repetir valores.
 - Unicidad: Impide que los valores de los atributos marcados de esa forma, puedan repetirse.
 - Obligatoriedad: Prohíbe que el atributo marcado de esta forma no tenga ningún valor.
 - Integridad referencial: Prohíbe colocar valores en una clave externa que no estén reflejados en la tabla donde ese atributo es clave primaria.

- Regla de validación: Condición que debe de cumplir un dato concreto para que sea actualizado.
- Restricciones inherentes: no son determinadas por los usuarios, sino que son definidas por el hecho de que la base de datos sea relacional.
 - No puede haber dos tuplas iguales
 - El orden de las tuplas/atributos no importa
 - Cada atributo sólo puede tomar un valor en el dominio en el que está inscrito

3. Dominios

Una poderosa característica del modelo relacional son los dominios. Estos se definen como un conjunto de valores legales de uno o varios atributos. Mientras que un atributo es el nombre de una columna de una relación. Cada atributo de una base de datos relacional se define sobre un dominio, pudiendo haber varios atributos definidos sobre el mismo dominio. Por lo tanto, un dominio restringe los valores del atributo.

Existen distintos tipos de dominios como pueden ser enteros, cadenas de texto, fecha, no procedurales, etc.

4. Claves

- **Clave primaria o única.** Aquella clave candidata que se escoge para identificar sus tuplas de modo único. Ya que una relación no tiene tuplas duplicadas, siempre hay una clave candidata y, por lo tanto, la relación siempre tiene clave primaria.
- **Clave foránea.** Aquel atributo de una relación cuyos valores coinciden con los valores de la clave primaria de alguna otra relación (puede ser la misma). Las claves foráneas representan relaciones entre datos.
- **Clave índice.** Surgen a partir de la necesidad de tener un acceso más rápido a los datos. Generalmente no se consideran parte de la base de datos, pues se trata de un detalle agregado.

5. Procedimientos almacenados

Consiste en un código ejecutable que se asocia y almacena con la base de datos. Usualmente recogen y personalizan operaciones en común, como insertar un registro dentro de una tabla, recopilar información estadística, o encapsular cálculos complejos. No forman parte del modelo relacional.

Características de una base de datos relacional

Las bases de datos relacionales tienen lo que se denomina un modo de coherencia o integridad basado en cuatro criterios: atomicidad, coherencia, aislamiento y durabilidad (ACID).

1. Atomicidad: Esta propiedad indica que, para que una transacción se dé por completada, deben haberse realizado todas sus partes o ninguna de ellas. Si una parte de la transacción falla, el sistema debe ser capaz de hacer que el resto de operaciones fallen, por lo que la base de datos no sufrirá ningún cambio indeseado.
2. Consistencia: se basa en la premisa que afirma que una transacción debe llevar al sistema de un estado válido a otro que también lo sea. Cabe resaltar que la validez de las operaciones está determinada por su seguimiento o no de las reglas establecidas para garantizar la fiabilidad de la base de datos.
3. Aislamiento: se refiere a la manera y el momento en el que los cambios resultantes de una operación se harán visibles para las demás operaciones concurrentes. Es decir, la realización de una operación no debería afectar a las otras, debido a que cada una de las transacciones debe ser ejecutada en aislamiento total, sin importar si se llevan a cabo de manera simultánea.
4. Durabilidad: hace referencia a la propiedad que garantiza que, una vez se haya llevado a cabo una determinada operación (aquellas transacciones que tuvieron un commit), estas tengan la capacidad de persistir y no puedan ser deshechas incluso si el sistema falla o se presentan eventos como errores o caídas o pérdida de alimentación eléctrica, entre otros.

Ventajas y desventajas de una base de datos relacional

El modelo de datos de la base de datos relacional se implementa y gestiona más fácilmente que otros modelos. Las grandes cantidades de información que las empresas quieren almacenar a largo plazo se organizan sin problemas.

El modelo relacional fija una normativa que tiene el fin de evitar duplicaciones. Si las reglas se aplican, los sistemas relacionales facilitan un almacenamiento de datos libre de redundancias, puesto que solo es necesario editar los datos una única vez.

Las bases de datos relacionales normalizadas permiten almacenar datos sin contradicciones, contribuyendo así a la consistencia de los datos. Asimismo, los sistemas relacionales presentan funciones con las cuales se definen y se controlan automáticamente las condiciones de integridad. Aquellas transacciones que ponen en peligro la consistencia de los datos se bloquean.

El sistema de base de datos relacional se apoya en un procesamiento orientado a conjuntos que subdivide cada entidad en valores mínimos, permitiendo conectar entidades diferentes por medio del contenido.

Para la realización de consultas a bases de datos relacionales se ha consolidado el lenguaje SQL, que ha sido estandarizado por la ISO y la IEC. El propósito de tal estandarización es que las aplicaciones puedan desarrollarse y ejecutarse con independencia del SGBD en que se utilicen.

Por otro lado, no siempre es posible integrar cualquier tipo de dato en el formato fijo de las tablas aun cuando estén interconectadas. Los datos abstractos o no estructurados que surgen en relación con las aplicaciones multimedia y las soluciones de big data no pueden representarse en el modelo relacional.

Al mismo tiempo el principio de base de los sistemas relacionales que consiste en almacenar la información en tablas separadas conduce inevitablemente a su segmentación, debido a que este diseño deriva en complejas consultas que abarcan varias tablas, de modo que el elevado número de segmentos resultantes acostumbra a reflejarse negativamente en el rendimiento.

Por último el modelo relacional plantea elevados requisitos en cuanto a la consistencia de datos que van en detrimento de la velocidad de escritura en las transacciones.

Comparación contra los otros tipos de datos, cuándo conviene usarla y cuándo no

Base de datos no relacional

Si conocemos la información que necesitamos registrar, podremos diseñar la estructura de la base de datos relacional y las tablas que necesitaremos para poder relacionarlas, de manera sencilla y rápida para acceder a los datos que queramos consultar.

Una base de datos no relacional no tiene un identificador que sirva para relacionar un conjunto de datos con otros. Para realizar consultas se emplean lenguajes propios como JSON, CQL o GQL. Normalmente se usa cuando la

información se organiza mediante documentos o cuando no tenemos un esquema exacto de lo que vamos a almacenar.

Después de esta explicación, podemos decir que ambos tipos de bases de datos son útiles y dependerá del tipo de aplicación que deseamos realizar la elección de una u otra base de datos.

Así, si queremos desarrollar una aplicación de tipo contable, de inventario o de información de clientes, es probable que el modelo relacional se adapte mejor. En este tipo de aplicaciones, normalmente habrá más de una tabla que tenga relación con el resto, por lo que una base de datos relacional será más útil y podrá representar mejor nuestra aplicación.

Si por el contrario nuestra aplicación necesita de un sistema en el que los registros que vaya a almacenar no necesitan relacionarse con otros, usaremos una base de datos no relacional. Estas serán muy útiles para guardar grandes volúmenes de datos que sean de un solo tipo de esquema, como por ejemplo puede ser una base de datos en la que solo queramos almacenar libros o películas, ya que almacenaremos cada documento y no lo relacionaremos con ningún otro dato, solamente le daremos una serie de atributos que se podrán consultar pero que no tendrán relación alguna con otra tabla.

Base de datos orientada a objetos

Las bases de datos relacionales se basan en la relación entre la información y diferentes factores. En la práctica, puedes verlo como una tabla, donde cada columna representa un tipo de información asociada con una fila. Una base de datos orientada a objetos se organiza en forma de diferentes objetos, que contienen archivos e información agrupados, así como procedimientos para leerlos y ejecutarlos

En una base de datos relacional, cada información individual se almacena como un archivo separado, independiente de los demás, y no hay soporte para el almacenamiento persistente de objetos con una estructura compleja. La atención se centra en datos más directos. Las bases de datos orientadas a objetos usan identificadores para etiquetar cada objeto, junto con técnicas de indexación para ubicar páginas en el disco. Como resultado, el sistema puede admitir objetos estructurados de forma más compleja.

Una base de datos orientada a objetos, como ya lo hemos mencionado, usa identificadores de objetos, que son como etiquetas. Al realizar una búsqueda de una de estas etiquetas, el sistema trae el conjunto de archivos e información que contiene. Y, dentro de ese objeto, puede haber otros tipos de jerarquías y relaciones internas.

El modelo relacional, a su vez, hace uso de dos claves relacionales. La primaria, como ya se mencionó, es la base, generalmente representada en la primera columna de la tabla, que identifica cada fila. Luego se usa una clave externa, que está relacionada con la primaria. Juntas sirven como una coordenada para localizar la información buscada.

Asimismo, pueden existir claves secundarias o terciarias, que actúan como filtros de búsqueda. De esta forma, es posible utilizar otras claves y separar aquellas que tienen información específica en un área determinada o aquellas que tienen un área vacía.

Una base de datos orientada a objetos, que tiene menos restricciones estructurales, puede contener mucha más información y en mayor variedad. Esto da como resultado un sistema de datos más flexible que se enfrenta a múltiples formatos de archivo, por ejemplo.

Una base de datos relacional, por otro lado, contiene un nivel más bajo de complejidad, lo que da como resultado un volumen de información relativamente menor.

Las bases de datos orientadas a objetos tienen una amplia variedad de permisos y restricciones según el sistema, por lo que es imposible generalizar. Por otro lado, las bases de datos relacionales tienen claves, integridad referencial e integridad de entidad impuestas como parte de su método.

En las bases de datos orientadas a objetos, la manipulación se incorpora directamente al lenguaje de programación utilizado, por ejemplo, C#.

Una base de datos relacional, en cambio, incluye su propio lenguaje para manipular su información, como SQL, QBE o QUEL. Esto reduce un poco la barrera para usar este sistema de manera más eficiente.

El primero es capaz de manejar solo un tipo de información a la vez, mientras que el segundo puede abarcar múltiples tipos de datos simultáneamente .

Bases de datos jerárquicas

La principal diferencia entre las bases de datos jerárquicas y las bases de datos relacionales es que las bases de datos jerárquicas almacenan datos en forma de árbol con nodos parentales y secundarios, mientras que las bases de datos relacionales almacenan datos en tablas con filas y columnas como entidades y atributos. Una base de datos jerárquica contiene datos duplicados, pero las bases de datos relacionales no.

Bases de datos transaccionales

Las diferencias más notorias son que para las bases de datos transaccionales la redundancia y duplicación de información no es un problema como para las bases de datos relacionales. Otro aspecto importante que podemos analizar es que las bases de datos relacionales se basan en la organización de la información en partes pequeñas que se integran mediante identificadores a diferencia de las bases de datos transaccionales, no tienen un identificador que sirva para relacionar dos o más conjuntos de datos. Además, son más robustas, es decir, tienen mayor capacidad de almacenamiento.

Manejadores o gestores de bases de datos relacionales

Existe un tipo de software exclusivamente dedicado a tratar bases de datos relacionales, conocido como sistema de gestión de bases de datos relacionales, también llamados manejadores o gestores de las BDR.

ORACLE

Es un sistema de gestión de base de datos relacional desarrollado por Oracle Corporation. Oracle surge en 1977 bajo el nombre SDL (Software Development Laboratories). Su creador, Larry Ellison, estuvo en el comité que definió SQL. Como software de base de datos, Oracle optimiza la gestión y seguridad de los conjuntos de datos creando esquemas estructurados a los que solo pueden acceder administradores autorizados. Su licenciamiento varía entre los \$57.312 dólares y \$79.892 dólares. Sus principales características son el soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma. Las principales empresas que usan Oracle son Netflix, LinkedIn, eBay, MIT, etc.

PostgreSQL

Es un sistema de gestión de bases de datos de código abierto y de clase empresarial. Es decir que se puede utilizar, modificar e implementar según las necesidades de cada aplicación. La primera versión formal de PostgreSQL fue en 1997. Permite diseñar tablas relacionales y campos JSON para no relacionales. Almacena texto, imágenes, sonidos y video. Escala a tantos servidores de bases de datos como el usuario desee. Crea funciones personalizadas usando C, C++, Java o PL/SQL. Es compatible con Linux, Mac OS y Windows. PostgreSQL es más capaz de lidiar con circunstancias inusuales de bases de datos y procesamiento de datos de gran volumen. Las principales empresas que usan PostgreSQL son IKEA, BMW, UBER, Netflix o Instagram. Su licenciamiento es gratuito.

MariaDB

Es un sistema de gestión de bases de datos de código abierto disponible en el mercado que actúa como interfaz SQL para que los datos puedan ser fácilmente accesibles. Su lanzamiento inicial fue en 2009. Grandes proyectos como Wikipedia y Google han migrado hacia ella. Se trata de un servicio de manejo de bases de datos que cuenta con licencia GPL. Creado por el mismo desarrollador de MySQL cuenta con un mayor rendimiento y funcionalidades que MySQL. Su licenciamiento varía en los \$500 dolares.

MySQL

Es el sistema de gestión de bases de datos relacionales más usado en el mundo de código abierto y administrado por Oracle.. Tiene una sencilla curva de aprendizaje. El SGBD MySQL fue desarrollado en 1995. Posee doble licencia: Open Source y comercial con soporte. Compatible con Linux, Mac Os y Windows. Es adecuado tanto para proyectos grandes como pequeños. Utilizada por empresas como Facebook, Twitter, LinkedIn, Yahoo!, Amazon Web Services. Hay diferentes tipos de licenciamiento, gratuito o pago (entre \$2000 dolares a \$10000 dolares).

SQL Server

Es un motor de base de datos relacional desarrollado por Microsoft que incluye mejoras al lenguaje SQL y soluciones para empresas. Su lanzamiento oficial fue en 1989. Funciona en Windows, Linux, Docker y Kubernetes. Integra herramientas de BI para no recurrir a soluciones de terceros. Tiene versiones gratuitas con todo lo necesario para proyectos y también versiones pagas que varían entre los \$230 dólares y \$15123 dólares. Puede ser instalado localmente o ser ejecutado en la nube de Azure. A diferencia de otros SGBD incluye un cliente gráfico para la administración. Facebook, Uber, Netflix, Amazon o Airbnb son algunos de los nombres de grandes empresas que utilizan a SQL Server como su gestor de bases de datos.

SQLite

Es un motor de base de datos SQL transaccional de código abierto, ligero, autónomo, de configuración simple y sin servidor. Apareció en 2000 de la mano de su creador D. Richard Hip. SQLite hace uso del lenguaje SQL y para el análisis de datos más complejos se puede combinar con scripts en Python. Compatible con Windows, Linux y Mac Os. Al estar integrado en todos los teléfonos se usa para almacenamiento interno de apps. Su código es de dominio público y gratuito. Almacena información persistente de forma sencilla. No admite un gran volumen de información. No obstante muchas empresas lo utilizan para sus aplicaciones de

escritorio como Adobe Photoshop Elements, Mozilla Firefox, Skype, Opera, The New Yorker, etc.

Instalación del motor de la BD

- <https://youtu.be/FaPO3GRAPV8>