

Clase . Bases de datos relacionales

Bases de datos relacionales

Concepto

Las bases de datos relacionales se basan en el modelo relacional y usan un conjunto de tablas para representar tanto los datos como las relaciones entre ellos

El modelo relacional significa que las estructuras lógicas de datos, las tablas de datos, vistas e índices están separadas de las estructuras físicas de almacenamiento. Esta separación significa que los administradores de bases de datos pueden administrar el almacenamiento físico de datos sin afectar el acceso a esos datos como una estructura lógica. Por ejemplo, cambiar el nombre de un archivo de base de datos no cambia el nombre de las tablas almacenadas en él.

La distinción entre lógica y física también se aplica a las operaciones de la base de datos, que son acciones claramente definidas que permiten a las aplicaciones manipular los datos y las estructuras de la base de datos. Las operaciones lógicas permiten que una aplicación especifique el contenido que necesita, mientras que las operaciones físicas determinan cómo se debe acceder a esos datos y luego realizan la tarea.

Para garantizar que los datos sean siempre precisos y accesibles, las bases de datos relacionales siguen ciertas reglas de integridad. Por ejemplo, una regla de integridad puede especificar que no se permiten filas duplicadas en una tabla, para eliminar la posibilidad de que ingrese información errónea en la base de datos.

Modelo relacional

Este modelo se basa en un conjunto de tablas. El usuario del sistema de bases de datos puede consultar estas tablas, insertar tuplas nuevas, borrar tuplas y actualizar (modificar) las tuplas.

[Escriba aquí]

Gino Luciano Rojo

En los primeros años de las bases de datos, cada aplicación almacenaba datos en su propia estructura única. Cuando los desarrolladores querían crear aplicaciones para usar esos datos, tenían que saber mucho sobre la estructura de datos particular para encontrar los datos que necesitaban. Estas estructuras de datos eran ineficientes, difíciles de mantener y difíciles de optimizar para ofrecer un buen rendimiento de la aplicación. El modelo de base de datos relacional se diseñó para resolver el problema de varias estructuras de datos arbitrarias.

El modelo relacional proporcionó una forma estándar de representar y consultar datos que cualquier aplicación podría utilizar. Desde el principio, los desarrolladores reconocieron que la principal fortaleza del modelo de base de datos relacional estaba en el uso de tablas, que eran una forma intuitiva, eficiente y flexible de almacenar y acceder a información estructurada.

Con el tiempo, cuando los desarrolladores comenzaron a utilizar el lenguaje de consulta estructurado (SQL) para escribir y consultar datos en una base de datos, surgió otra fortaleza del modelo relacional. Durante muchos años, se utilizó ampliamente el SQL como lenguaje para consultas de bases de datos. El SQL, que se basa en el álgebra relacional, proporciona un lenguaje matemático internamente consistente que facilita la mejora del rendimiento de todas las consultas de la base de datos. En comparación, otros enfoques deben definir consultas individuales.

El modelo relacional es el mejor para mantener la consistencia de los datos en todas las aplicaciones y copias de la base de datos (denominadas instancias). Por ejemplo, cuando un cliente deposita dinero en un cajero automático y, luego, mira el saldo de la cuenta en un teléfono móvil, el cliente espera ver que ese depósito se refleje inmediatamente en un saldo de cuenta actualizado. Las bases de datos relacionales se destacan en este tipo de consistencia de datos, lo que garantiza que múltiples instancias de una base de datos tengan los mismos datos todo el tiempo.

Es difícil para otros tipos de bases de datos mantener este nivel de coherencia oportuna con grandes cantidades de datos. Algunas bases de datos recientes, como NoSQL, solo pueden proveer “consistencia eventual.” Bajo este principio, cuando la base de datos se escala o cuando varios usuarios acceden a los mismos datos al mismo tiempo, los datos necesitan algo de tiempo para “ponerse al día.” La consistencia eventual es aceptable para algunos usos, como para mantener listados en un catálogo de productos, pero para operaciones comerciales críticas como transacciones de un carrito de compras, la base de datos relacional sigue siendo el estándar de oro.

Tablas

También llamada entidad y en algunos textos relación. Es una estructura compuesta por tuplas (filas, registros) y atributos (campos, columnas o atributos).

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.

El número de tablas de una base de datos en sql se limita solo por el número de objetos admitidos en una base (2.147.483.647). Una tabla definida por el usuario estándar puede tener hasta 1.024 columnas. El número de filas de la tabla sólo está limitado por la capacidad de almacenamiento del servidor (espacio en KBs).

Se puede asignar propiedades a la tabla y a cada columna de la tabla para controlar los datos admitidos y otras propiedades. Por ejemplo, crear restricciones en una columna para no permitir valores nulos o para proporcionar un valor predeterminado si no se especifica un valor, o puede asignar una restricción de clave en la tabla que exige la unicidad o definir una relación entre las tablas.

Ventajas del modelo relacional

Favorece el proceso de normalización.

La normalización es el proceso de organizar datos en una base de datos. Esto incluye la creación de tablas y el establecimiento de relaciones entre esas tablas de acuerdo con las reglas diseñadas tanto para proteger los datos como para que la base de datos sea más flexible mediante la eliminación de la redundancia y la dependencia incoherente.

Los datos redundantes desperdician espacio en disco y crean problemas de mantenimiento. Si es necesario cambiar los datos que se encuentran en más de un lugar, los datos deben cambiarse exactamente de la misma forma en todas las ubicaciones. El cambio de dirección de un cliente es mucho más fácil de implementar si los datos se almacenan solo en la tabla clientes y en ninguna otra parte de la base de datos.

El objetivo es generar un conjunto de esquemas de relaciones que permita almacenar información sin redundancias innecesarias, pero que también permita recuperar la información con facilidad. El enfoque es diseñar esquemas que se hallen en la forma normal adecuada.

[Escriba aquí]

Gino Luciano Rojo

Para determinar si un esquema de relación se halla en una de las formas normales deseadas, hace falta información adicional sobre la empresa real que se está modelando con la base de datos.

Permite implementar el lenguaje SQL.

SQL es un lenguaje de computación para trabajar con conjuntos de datos y las relaciones entre ellos. Los programas de bases de datos relacionales, como SQL Server y Microsoft Office Access, usan SQL para trabajar con datos. A diferencia de muchos lenguajes de computación, SQL no es difícil de leer y entender, incluso para un usuario inexperto. Al igual que muchos lenguajes de computación, SQL es un estándar internacional reconocido por organismos de estándares como ISO y ANSI.

SQL se usa para describir conjuntos de datos que pueden ayudarle a responder preguntas. Al usar SQL, debe usar la sintaxis correcta. La sintaxis es el conjunto de reglas mediante las que se combinan correctamente los elementos de un idioma. La sintaxis SQL se basa en la sintaxis del idioma inglés y usa muchos de los mismos elementos que la sintaxis de Visual Basic para Aplicaciones (VBA).

Permite implementar integridad referencial.

Las reglas de integridad referencial evitan que los usuarios de bases de datos accidentalmente eliminen asignaciones entre columnas relacionadas.

Por ejemplo, tiene una base de datos denominada Libros con dos tablas denominadas Títulos y Autores. Sólo para este ejemplo, cada título sólo tiene un autor, pero cada autor puede tener varios títulos, creándose lo que se denomina una relación uno-a-muchos. La tabla de "un" lado de la relación se denomina tabla principal y la de "muchos" lados, tabla secundaria.

Con el fin de recabar toda la información de cada libro y su autor, una consulta usa claves para que coincida cada libro de la tabla Títulos con su autor de la tabla Autores. La tabla principal (Autores) debe tener una clave principal que identifique exclusivamente cada fila. La tabla Títulos debe tener una columna que contenga el IdDeAutor de cada título. Esta columna se denomina clave externa. La clave principal identifica a cada autor, mientras que la externa, los libros que ha escrito cada autor.

Si un usuario cambia el IdDeAutor del autor de la tabla principal, sin actualizar el IdDeAutor de la tabla Títulos, los libros de esta tabla ya no tienen ningún autor válido.

En vez de tener que cambiar el IdDeAutor en ambas tablas, puede establecer reglas de integridad referencial para que se realice la acción deseada. Puede elegir entre tener el modelo de base de datos para que emprenda acciones específicas cuando se cambie o elimine la clave en la tabla principal.

[Escriba aquí]

Tipos de relaciones

Uno de los objetivos de un buen diseño de base de datos es eliminar la redundancia de los datos (datos duplicados). Para conseguirlo, conviene desglosar los datos en muchas tablas basadas en temas para que cada hecho esté representado solo una vez (proceso de normalización). Después, se debe proveer a la base de datos de un medio para recopilar de nuevo la información desglosada (esto se realiza colocando campos comunes en tablas que estén relacionadas). Sin embargo, para realizar este paso correctamente, primero debe conocer las relaciones existentes entre las tablas y, a continuación, especificar dichas relaciones en la base de datos.

Tras crear una tabla para cada tema en la base de datos, es preciso proporcionar a la base de datos un medio para recopilar de nuevo esa información cuando sea necesario. Para ello, se colocan campos comunes en las tablas que estén relacionadas y se definen las relaciones entre las tablas. De ese modo, se pueden crear consultas, formularios e informes que muestren a la vez la información de varias tablas.

Una relación uno a muchos.

Planteamos como ejemplo una base de datos de seguimiento de pedidos que incluya una tabla Clientes y una tabla Pedidos. Un cliente puede realizar cualquier número de pedidos. Por cada cliente representado en la tabla Clientes puede haber representados muchos pedidos en la tabla Pedidos. Por lo tanto, la relación entre la tabla Clientes y la tabla Pedidos es una relación de uno a muchos.

Para representar una relación uno a varios en el diseño de la base de datos, se debe tomar la clave principal del lado "uno" de la relación y agréguela como campo o campos adicionales a la tabla en el lado "muchos" de la relación. En este caso, por ejemplo, agregaría un nuevo campo (el campo Id. de la tabla Clientes) a la tabla Pedidos y lo denominaría Id. de cliente. Así, la base de datos puede usar el número de identificador del cliente de la tabla Pedidos para dar con al cliente correcto de cada producto.

Una relación muchos a muchos

Centrémonos ahora en la relación entre una tabla Productos y una tabla Pedidos. Un solo pedido puede incluir varios productos. Por otra parte, un único producto puede aparecer en muchos pedidos. Por tanto, por cada registro de la tabla Pedidos puede haber varios registros en la tabla Productos. Además, por cada registro de la tabla Productos puede haber varios registros en la tabla Pedidos. Esta relación se denomina relación de varios a varios. Se debe tener en cuenta que para detectar las relaciones de varios a varios existentes entre las tablas, es importante que considere ambas partes de la relación.

[Escriba aquí]

Gino Luciano Rojo

Una buena práctica para simplificar una relación de varios a varios, se debe crear una tercera tabla, a menudo denominada tabla de unión, que divide la relación de muchos a muchos en dos relaciones uno a muchos . Debe insertar la clave principal de cada una de las dos tablas en la tercera. Como resultado, la tercera tabla registra cada ocurrencia, o instancia, de la relación. Por ejemplo, la tabla Pedidos y la tabla Productos tienen una relación muchos a muchos que se define mediante la creación de dos relaciones uno a muchos con la tabla Detalles de pedidos. Un pedido puede incluir muchos productos, y cada producto puede aparecer en muchos pedidos.

Una relación uno a uno

En una relación uno a uno, cada registro de la primera tabla sólo puede tener un registro coincidente en la segunda tabla y viceversa. Esta relación no es común porque la mayor parte de las veces la información relacionada de este modo se almacena en la misma tabla. Se puede usar la relación uno a uno para dividir una tabla con muchos campos, para aislar parte de una tabla por razones de seguridad o para almacenar información que solo se aplica a un subconjunto de la tabla principal. Cuando se identifica esta relación, ambas tablas deben compartir un campo común.

Claves

Clave primaria

También llamada llave primaria o primary key, hace que el registro sea unívoco y obligatoriamente no nulo. Este es el principal componente de la integridad referencial.

La clave primaria debe escogerse de manera que los valores de sus atributos no se modifiquen nunca, o muy rara vez. Por ejemplo, el campo domicilio de una persona no debe formar parte de la clave primaria, ya que es probable que se modifique. Por otra parte, está garantizado que los números de la seguridad social no cambian nunca. Los identificadores exclusivos generados por las empresas no suelen cambiar, salvo si se produce una fusión entre dos de ellas; en ese caso, puede que el mismo identificador haya sido emitido por ambas empresas, y puede ser necesaria una reasignación de identificadores para garantizar que sean únicos.

Clave foránea

También llamada foreign key, clave secundaria o clave externa, puede ser o no una clave primaria dentro de la tabla, su característica es que es el punto de enlace con otra tabla donde esta es primary key.

[Escriba aquí]

Una clave externa (FK) es una columna o combinación de columnas que se usa para establecer y aplicar un vínculo entre los datos de dos tablas a fin de controlar los datos que se pueden almacenar una tabla de clave externa. En una referencia de clave externa, se crea un vínculo entre dos tablas cuando las columnas de una de ellas hacen referencia a las columnas de la otra que contienen el valor de clave principal. Esta columna se convierte en una clave externa para la segunda tabla.

Las columnas de clave externa suelen usarse en los criterios de combinación cuando los datos de las tablas relacionadas se combinan en consultas mediante la correspondencia de la columna o columnas de la restricción de clave externa de una tabla y la columna o columnas de la clave única o principal de la otra.

Clave índice

Es un campo que facilita la búsqueda dentro de una tabla, generalmente son campos primary key.

Un índice permite al Motor de base de datos buscar con rapidez datos relacionados en la tabla de clave externa.

Un índice contiene claves generadas a partir de una o varias columnas de la tabla o la vista. Dichas claves están almacenadas en una estructura (árbol b) que permite que SQL Server busque de forma rápida y eficiente la fila o filas asociadas a los valores de cada clave.

Clave concatenada

Ayuda a encontrar la singularidad en una tabla combinando dos campos ya que no hay una llave primaria.

Esta se consigue concatenando dos o más columnas dentro de la tabla, de esta forma se consigue la característica que tiene una clave primaria, ya que hace que cada uno de los registros no se repitan y sean únicos

Se dice que es una clave conceptual, ya que esta no se puede definir dentro de la estructura lógica de la tabla, por medio de lenguaje de definición, sino que se define al momento de usar las tablas en una consulta SQL.

Clave candidata

Cuando existe más de una clave primaria dentro de la tabla, por ejemplo legajo y dni. Estas columnas se le denominan claves candidatas. Es importante identificar que una tabla sólo

[Escriba aquí]

puede tener una clave primaria, por ende las candidatas son las columnas de las cuales se debe elegir una para establecer la integridad referencial.

Modelo entidad-relación

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consiste en un conjunto de objetos básicos, denominados entidades, y de las relaciones entre esos objetos. Una entidad es una “cosa” u “objeto” del mundo real que es distinguible de otros objetos.

Las entidades se describen en las bases de datos mediante un conjunto de atributos. Por ejemplo, los atributos número_cuenta y saldo pueden describir una cuenta concreta de un banco y constituyen atributos de la entidad cuenta.

Análogamente, los atributos nombre_cliente, calle_cliente y ciudad_cliente pueden describir una entidad cliente. Se usa un atributo extra, id_cliente, para identificar unívocamente a los clientes (dado que es posible que haya dos clientes con el mismo nombre, calle y ciudad). Se debe asignar un identificador de cliente único a cada cliente.

En Estados Unidos, muchas empresas usan el número de la seguridad social de cada persona (un número único que el Gobierno de Estados Unidos asigna a cada persona) como identificador de cliente. Una relación es una asociación entre varias entidades. Por ejemplo, la relación impositor asocia un cliente con cada cuenta que tiene. El conjunto de todas las entidades del mismo tipo, y el conjunto de todas las relaciones del mismo tipo se denominan, respectivamente, conjunto de entidades y conjunto de relaciones.

La estructura lógica general (esquema) de la base de datos se puede expresar gráficamente mediante un diagrama E-R, que está constituido por los siguientes componentes:

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan atributos.
- Rombo, que representan conjuntos de relaciones entre miembros de varios conjuntos de entidades.
- Líneas, que unen los atributos con los conjuntos de entidades entre sí, y también los conjuntos de entidades con las relaciones.

Cada componente se etiqueta con la entidad o relación que representa.

[Escriba aquí]

Gino Luciano Rojo