

Resumen de Grafos

Capítulo 1: Los grafos son el futuro

Los grafos son una nueva tecnología que se puede utilizar para mejorar la infraestructura de bases de datos existente. Los grafos permiten un mejor rendimiento al manejar las consultas de relaciones a medida que aumenta el número y profundidad de las relaciones. Además, la estructura y el esquema de un modelo de datos en grafo son flexibles y pueden adaptarse a medida que cambian las soluciones y la industria, lo que permite a los equipos de TI y arquitectos de datos avanzar a la velocidad del negocio. El desarrollo con grafos también se alinea perfectamente con las prácticas actuales de desarrollo ágil y basado en pruebas, lo que permite que las aplicaciones respaldadas por grafos evolucionen junto con los requisitos comerciales cambiantes.

Los grafos son más intuitivos de entender que los sistemas de gestión de bases de datos relacionales y están compuestos por nodos y relaciones. Cada nodo representa una entidad y cada relación representa cómo se asocian dos nodos. Twitter es un ejemplo perfecto de una base de datos en grafo que conecta a millones de usuarios activos mensuales. Las bases de datos en grafo priorizan las relaciones, lo que significa que las conexiones de datos son más fáciles de establecer y los modelos de datos son más simples y expresivos que los producidos por bases de datos relacionales o almacenamientos NoSQL. Es importante entender las dos propiedades clave de las tecnologías de bases de datos en grafo: el almacenamiento de grafo nativo que está diseñado para guardar y manejar grafos, y el motor de procesamiento de grafo nativo que apunta los nodos entre ellos en la base de datos.

Capítulo 2: Porque las relaciones entre datos importan.

La ironía de las bases de datos relacionales (RDBMS). Aunque estas bases de datos están diseñadas para codificar formas y estructuras tabulares, son inflexibles en su esquema, lo que hace difícil mantenerse al día con las cambiantes necesidades empresariales y, por lo tanto, suelen estar llenas de tablas de JOIN grandes y complejas que afectan el rendimiento y dificultan el desarrollo futuro.

Aunque las bases de datos NoSQL pueden ofrecer cierta ventaja en términos de rendimiento, también tienen problemas para manejar las relaciones de datos adecuadamente, ya que su construcción desconectada dificulta el aprovechamiento de las relaciones de datos. Por lo tanto, las bases de datos de grafos pueden ser la solución. Al centrar las relaciones de datos en su modelo, son más flexibles, permitiendo la adición de nuevos nodos y relaciones sin comprometer la red existente y con resultados más eficientes en términos de velocidad de consulta.

Capítulo 3: Modelado de datos básico.

La modelización de datos es un proceso de abstracción que parte de las necesidades de negocio y usuarios para mapear una estructura de almacenamiento y organización de datos. Aunque en los sistemas de gestión de bases de datos relacionales (RDBMS) este proceso es complejo, ya que es necesario crear primero un modelo lógico para luego forzar esa estructura en un modelo físico tabular, lo que resulta en una base de datos

que se aleja del boceto original. En cambio, la modelización para una base de datos gráfica es mucho más sencilla, ya que el diagrama de la estructura de la base de datos es en sí mismo un gráfico. En lugar de adaptar el modelo lógico a una tabla compleja, se enriquece el modelo inicial según las necesidades de negocio y usuario.

Es importante destacar que la modelización de datos no es una actividad única que se realiza al principio del desarrollo de una aplicación. Los sistemas cambian constantemente y las necesidades de negocio evolucionan, por lo que el modelo de datos debe adaptarse para mantener la integridad de los datos. Los RDBMS no son adecuados para cambios rápidos debido a su esquema rígido y complejo proceso de modelización, lo que hace que una base de datos gráfica sea una mejor opción para adaptar el modelo de datos a un futuro incierto y mantener el rendimiento.

Capítulo 4: Fallos en modelado de datos que evitar

El Capítulo 4 habla sobre los errores comunes en la modelización de datos y cómo evitarlos. En particular, se analiza un ejemplo de modelo de detección de fraudes en las comunicaciones por correo electrónico. Se comienza con un modelo estrella que representa a los usuarios, sus actividades y alias conocidos. Sin embargo, este modelo resulta insuficiente ya que no permite correlacionar las relaciones entre EMAILED, CC y BCC. La solución para este problema es agregar nodos a nuestro modelo que representen cada uno de los correos electrónicos intercambiados y añadir nuevas relaciones para hacer seguimiento de quién escribió el correo y a quién se envió en CC y BCC. Así se logra un modelo más robusto que nos permite explorar el comportamiento sospechoso y realizar un seguimiento de la información crítica en caso de fraudes o ciberseguridad.

A pesar de que este modelo mejorado es más robusto, aún no está completo. No se pueden rastrear las respuestas o reenvíos de los correos electrónicos intercambiados. La solución no es simplemente agregar relaciones de REPLIED_TO y FORWARDED en el modelo existente, sino que se requiere una solución más elaborada. Se deben agregar dos etiquetas, "Correo electrónico" y "Respuesta", para representar los dos roles de una respuesta de correo electrónico. Con esto, se pueden usar las mismas relaciones TO, CC y BCC para mapear si la respuesta se envió al remitente original, a todos los destinatarios o a un subconjunto de los destinatarios. También se puede hacer referencia al correo electrónico original con una relación REPLY_TO. Con esta solución, se logra un modelo de datos más completo y efectivo para la detección de fraudes en las comunicaciones por correo electrónico.