INSTITUTO TECNOLOGICO METROPOLITANO ITM MEDELLIN

INGENIERIA EN SISTEMAS DE INFORMACIÒN

BASE DE DATOS AVANZADA

Tal vez la normalización no sea normal

Por

Cristian Camilo Montoya

Jhony Ochoa

Tal vez la normalización no sea normal

Uno de los elementos con los que estamos luchando ahora en Stack Overflow es cómo mantener niveles de rendimiento casi instantáneos en una base de datos relacional a medida que aumenta la cantidad de datos. Más específicamente, cómo escalar nuestro sistema de etiquetado. Los principios tradicionales de diseño de bases de datos te dicen que las bases de datos bien diseñadas siempre están normalizadas, pero no estoy tan seguro.

Dare Obasanjo tuvo una excelente publicación Cuándo no normalizar su base de datos SQL en la que proporciona un esquema de base de datos de muestra para un sitio de redes sociales genérico. Esto es lo que parecería si lo diseñáramos de la manera normalizada aceptada:

La normalización sin duda cumple en términos de limitar la duplicación. Cada entidad está representada una vez, y solo una vez, por lo que casi no hay riesgo de inconsistencias en los datos. Pero este diseño también requiere la friolera de seis combinaciones para recuperar la información de un solo usuario.

Esto funciona: las consultas ahora son increíblemente simples (selección \* de los usuarios), y probablemente también son increíblemente rápidas. Pero tendrá un montón de huecos vacíos en sus datos, junto con una gran cantidad de matrices de campos con nombres incómodos.

¿Y todos esos problemas molestos de integridad de datos que la base de datos usó para usted? Esos son todos tus trabajos ahora. Felicitaciones por su degradación! Ambas soluciones tienen sus pros y sus contras.

¿cuál es mejor: una base de datos normalizada o una base de datos desnormalizada?

¡Pregunta capciosa! ¡La respuesta es que no importa! Hasta que tengas millones y millones de filas de datos, eso es. Todo es rápido para los pequeños. Incluso una PC modesta para los estándares actuales, digamos una caja de doble núcleo con 4 gigabytes de memoria, le brindará un rendimiento casi idéntico en cualquier caso para cualquier cosa que no sea la base de datos más grande.

calcular los datos en las escrituras y evitar las uniones mediante el desarrollo de un conjunto de tablas desnormalizadas.

temamos al espectro de la desnormalización. Los datos duplicados y los problemas de sincronización a menudo son exagerados y son relativamente fáciles de solucionar con los trabajos cron. Los discos y la memoria son baratos y se abaratan cada nanosegundo. Mida el rendimiento en su sistema y decida qué funciona, sin predisposiciones ni sesgos.

Como dice el viejo adagio, normaliza hasta que duela, desnormaliza hasta que funcione.

https://blog.codinghorror.com/maybe-normalizing-isnt-normal/