¿Por qué SQL está superando a NoSQL y qué significa esto para el futuro de los datos?

Una nueva esperanza

Dos jóvenes científicos informáticos se dieron cuenta de que “ gran parte del éxito de la industria de la computación depende del desarrollo de una clase de usuarios que no sea informática capacitada especialistas "Querían un lenguaje de consulta que fuera tan fácil de leer como el inglés, y que también abarcara la administración y manipulación de la base de datos.

A principios de la década de 1970, donde nació la base de datos relacional, los lenguajes de consulta se basaban en una lógica matemática y una notación complejas. Donald Chamberlin y Raymond Boyce impresionaron a dos nuevos doctores por el modelo de datos relacionales, pero vieron que el lenguaje de consulta sería un importante cuello de botella para la adopción de este nuevo modelo de base de datos. Se propusieron diseñar un nuevo lenguaje de consulta que sería (en sus propias palabras): " más accesible para los usuarios sin capacitación formal en matemáticas o programación de computadoras ".

El resultado fue SQL, que se introdujo por primera vez en el mundo en 1974. En las próximas décadas, el SQL demostraría ser inmensamente popular. Como las bases de datos relacionales se hicieron cargo de la industria del software, el SQL se convirtió en el lenguaje preeminente para interactuar con una base de datos, y se convirtió en la lengua franca para Ecosistema abarrotado y competitivo.

NoSQL contraataca

A medida que Internet seguía creciendo y creciendo, la comunidad de software descubrió que las bases de datos relacionales de esa época no podían manejar esta nueva carga. Hubo una perturbación en la fuerza, como si un millón de bases de datos gritaran y se sobrecargaran de repente.

Luego, nuevos gigantes de Internet lograron avances, y desarrollaron sus propios sistemas no relacionales distribuidos para ayudar con este nuevo ataque de datos: MapReduce y Bigtable por Google, y Dynamo en Amazon. Estos documentos llevaron a más bases de datos no relacionales, incluyendo Hadoop, Cassandra y MongoDB. Debido a que estos eran sistemas nuevos, en gran parte escritos desde cero, también evitaban el SQL, lo que llevó al surgimiento del movimiento NoSQL.

La comunidad de desarrolladores de software se comió NoSQL, abrazándolo mucho más ampliamente de lo que pretendían Google / Amazon. Es fácil entender por qué: NoSQL era nuevo y brillante; Prometió escala y poder; Parecía el camino rápido hacia el éxito de la ingeniería. Pero entonces empezaron a aparecer los problemas.

Los desarrolladores pronto descubrieron que no tener SQL era en realidad bastante limitante. Cada base de datos NoSQL ofrecía su propio lenguaje de consulta único, lo que significaba: más idiomas para aprender (y para enseñar a sus compañeros de trabajo); mayor dificultad para conectar estas bases de datos a aplicaciones, lo que lleva a toneladas de código de pegamento frágil; la falta de un ecosistema de terceros, que obliga a las empresas a desarrollar sus propias herramientas operativas y de visualización.

Retorno del SQL

Un día nos dimos cuenta de que construir nuestro propio lenguaje de consulta no tenía sentido. Que la clave era abrazar SQL. Y esa fue una de las mejores decisiones de diseño que hemos tomado. Inmediatamente se abrió un mundo completamente nuevo. Hoy, aunque solo tenemos una base de datos de 5 meses, nuestros usuarios pueden usarnos en producción y obtener todo tipo de cosas maravillosas de la caja: herramientas de visualización (Tableau), conectores para ORM comunes, una variedad de herramientas y opciones de copia de seguridad , una gran cantidad de tutoriales y explicaciones de sintaxis en línea, etc.