

Graph NoSQL (Not only SQL) database

Graph NoSQL (Not only SQL) database - este o tehnologie care gestionează datele, proiectată să se ocupe de seturi foarte mari de date structurate, semi-structurate sau nestructurate.

Graph databases fac parte din NoSQL databases create pentru a răspunde limitelor bazelor de date relaționale existente.

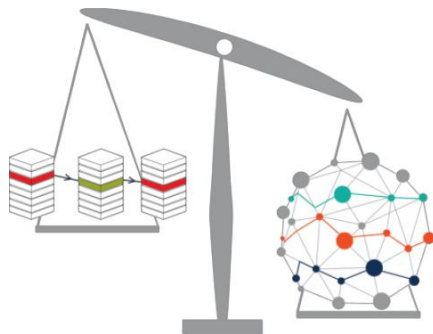
NoSQL Graph Database Vs. Relational Database

Abordarea tradițională a gestionării datelor, baza de date relațională, au fost dezvoltate în anii 1970 în scopul de a ajuta întreprinderile să stocheze informații structurate. Baza de date relațională are nevoie de o schemă a sa care trebuie să fie definită înainte de adăugarea oricăror informații noi.

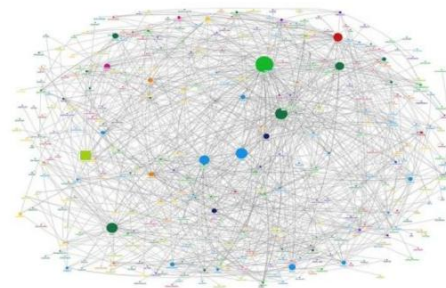
Astăzi, totuși, datele mobile, sociale și Internet-ul obiectelor (IoT) sunt peste tot, și conțin date nestructurate în timp real, care se încarcă pe minut. În afară de gestionarea unei cantități masive de date de orice natură, NoSQL Graph database nu are nevoie de schema sa redefinită înainte de adăugarea noilor date.

Acest lucru face ca baza de date a grafului(Graph database) să fie mult mai flexibilă, dinamică și mai ieftină în integrarea noilor surse de date decât bazele de date relaționale.

În comparație cu viteza moderată a datelor dintr-una sau mai multe locații ale bazelor de date relaționale, NoSQL Graph Databases sunt capabile să stocheze, să recupereze, să integreze și să analizeze date de mare viteză provenite din mai multe locații.



NoSQL Graph Database Vs. Relational Database



A complex graph database.

Azure Cosmos DB- este un serviciu de baze de date Microsoft bazat pe gestionarea datelor la scară planetară , lansat în mai 2017. Acesta se bazează și pe extinderea anterioară a Azure **DocumentDB** , lansat în 2014. Nu are schemă și este clasificat în general ca NoSQL database.

Cu opțiunea curentă recomandată de tip de colecție împărțită, Cosmos DB este ajustabil dinamic pe trei dimensiuni:

- **Tranzitată.** Dezvoltatorii păstrează disponibilitatea serviciului în funcție de sarcina variabilă a aplicației. Cosmos DB va mări resursele (memorie, procesor, partiții, replici etc.) pentru a atinge acea viteză de transfer solicitată, menținându-se peste 99.99% din concurența în domeniu pentru citirile sub 10 ms și pentru scrierea sub 15 ms. Transmiterea este specificată în unități de solicitare (RU) pe secundă.
- **Spațiu.** În mod similar, dezvoltatorii pot specifica de cât de mult spațiu vor avea nevoie. Atât spațiul cât și viteza de tranziție afectează în mod direct cât de mult este încărcat utilizatorul, dar poate fi reglat dinamic pentru a gestiona sarcina maximă și pentru a economisi costurile atunci când este încărcat mai puțin.
- **Consistența.** Cosmos DB oferă patru nivele de consistență: puternic, limitat-scară, sesiune și eventual. Cu cât nivelul este mai la stânga în această listă, cu atât este mai mare consistența și costul RU(Request Units) care reduce în mod substanțial capacitatea disponibilă pentru aceeași setare a RU. Consistența nivelului sesiunii este implicită.

Partiționarea

Cosmos DB a adăugat funcționalitatea automată de partiționare în 2016, odată cu introducerea colecțiilor partajate. Colecția va cuprinde mai multe partiții fizice cu documente distribuite de o cheie de partiție furnizată de apelant. Programul decide automat numărul de partiții necesare pentru a răspândi datele în funcție de mărimea și necesarul de transfer. Când programul decide să adauge (sau să elimine) partiții, datele rămân disponibile în timp ce sunt reechilibrate pe noile partiții (sau cele rămase).

Indexare automată

În mod prestabilit, fiecare câmp din fiecare document este indexat automat, oferind în general performanțe bune fără a se adapta tipurilor de interogare specifice. Aceste valori implicite pot fi modificate prin setarea unei politici de indexare care poate varia pentru fiecare câmp în parte.

JavaScript

Un motor JavaScript este încorporat în Cosmos DB, și permite funcționalități suplimentare:

- **Proceduri stocate.** Funcții care cuprind un set complex de operații și logică arbitrar într-o tranzacție compatibilă cu proprietățile ACID(Atomicity, Consistency, Isolation, and Durability). Ele sunt izolate de modificările efectuate în timp ce procedura stocată este executată și fie toate operațiile de scriere reușesc, fie toate eșuează, lăsând baza de date într-o stare consecventă. Procedurile stocate sunt executate într-o singură partiție.
- **Triggere.** Funcțiile care se execută înainte sau după anumite operații specifice (cum ar fi de exemplu inserarea unui document) care pot fie să modifice operația, fie să o anuleze. Triggerele se execută numai la cerere și nu rulează garantat.
- **Funcții definite de utilizator (UDF).** Funcții care pot fi apelate și amplifică limbajul de interogare SQL compensând suportul SQL limitat.