UNIVERSITATEA "ALEXANDRU IOAN CUZA"
DIN IAȘI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ

Baze de date de tip graf

Student:
Marius-George Olaru

Coordonator științific:
Lect. Dr. Cristian Frăsinaru

SESIUNEA: IULIE 2018

UNIVERSITATEA "ALEXANDRU IOAN CUZA"
DIN IAȘI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ

Baze de date de tip graf

Student:
Marius-George Olaru

Coordonator științific:
Lect. Dr. Cristian Frăsinaru

SESIUNEA: IULIE 2018

Declaratie privind drepturile de autor

Eu, Marius-George Olaru, declar că lucrarea intitulată, "Baze de date de tip graf" este scrisă de mine și nu a mai fost prezentată niciodată la o altă facultate sau instituție de învățământ superior din țară sau din străinătate. De asemenea, declar că toate sursele utilizate sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

- toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și dețin referință precisă a sursei;
- reformularea în cuvinte proprii a textelor scrise de către alți autori deține referință precisă;
- codul sursă, imaginile etc. preluate din proiecte open-source sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;
- rezumarea ideilor altor autori precizează referință precisă la textul original.

Student: _____

Data: _____

Semnatură:

Declaratie de consimtamant

Prin prezență declar că sunt de acord ca Lucrarea de licență cu titlul "Baze de date de tip graf", codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică de la Universitatea "Alexandru Ioan Cuza" din Iași , să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Student: _____

Data: _____

Semnatură:

Cuprins

Introducere

Sistemele de baze de date sunt o componentă esenţială a vieţii de zi cu zi în societatea modernă. În cursul unei zile, majoritatea persoanelor desfăşoară activităţi care implică interacţiunea cu o bază de date: depunerea sau extragerea unor sume de bani din bancă, rezervarea biletelor de tren sau avion etc. Bazele de date pot avea dimensiuni (număr de înregistrări) extrem de variate, de la câteva zeci de înregistrări (de exemplu, baza de date pentru o agendă cu numere de telefon) sau pot ajunge la zeci de milioane de înregistrări (de exemplu, baza de date pentru plata taxelor şi a impozitelor). În sensul cel mai larg, o bază de date (database) este o colecţie de date corelate din punct de vedere logic, care reflectă un anumit aspect al lumii reale. Criteriul principal de clasificare a bazelor de date îl reprezintă modelul conceptual utilizat în descrierea structurii datelor. Aparținând criteriului bazelor de date nerelaționale, Neo4J este o bază de date fondată pe teoria grafurilor, fiind o soluție optimizată pentru a modela și interoga volume mari de date strâns relaționate, reprezentabile prin structuri de tip graf.

Am ales să folosesc acest sistem de gestiune a bazei de date în contextul unei aplicații Web , care poate fi descrisă ca o rețea socială în miniatură. Principalul motiv pentru care am luat această decizie este acela că o reprezentare sub formă de graf a datelor se pliază foarte "intuitiv" în acest context, astfel încât entitățile din baza de date sunt reprezentate prin noduri, iar relațiile dintre acestea prin muchiile dintre noduri.

Pentru realizarea aplicației se vor folosi tehnologii actuale, ce se află atât în componenta de back-end a aplicației , cât și pe partea de client.

Scopul lucrării prezente este de a folosi pentru stocarea datelor soluția oferită de Neo4J (fiind cea mai cunoscută implementare de bază de date de tip graf), astfel studiind această abordare de reprezentare a datelor și de a putea compara cu alte abordări (spre exemplu bazele de date relaționale) din următoarele perspective: performanță , scalabilitate, flexibilitate, complexitate.

Deși folosirea bazelor de date relaționale se pretează pe majoritatea situațiilor, în acest caz, folosirea unei baze de date de tip graf, respectiv Neo4J, în contextul în care avem de memorat cantități mari de noduri și relații între acestea, și apoi de parcurs prin interogări, este o soluție extrem de potrivită.

Contributii

În cadrul acestui proiect am avut contribuții pe următoarele planuri:

- Pe plan teoretic:
  - Definirea unei structuri a bazei de date , prin crearea nodurilor și a muchiilor dintre ele , care să pună cât mai mult în valoare folosirea acestui tip de reprezentare a datelor;
  - Studierea limbajului folosit pentru executarea scripturilor asupra bazei de date Neo4J, Cypher;
  - Compararea performanței realizate de baza de date pentru aducerea datelor în serviciu, apoi urmând să fie afișate clientului.
- Pe plan practic:
  - Identificarea cerințelor pentru a implementa o astfel de aplicație;
  - Analizarea soluțiilor din punct de vedere al implementării modelului propus;
  - Implementarea aplicației pe partea de client, pentru a demonstra contextul folosit, și anume cel al unei rețele sociale în miniatură;
  - Integrarea aplicației cu baza de date pusă la dispoziție de Neo4J, și implementarea scripturilor Cypher necesare funcționării corecte a modelului.

Contextul actual

În prezent, bazele de date tradiţionale sunt puse la încercare din ce în ce mai mult de noile tipuri de aplicaţii care le folosesc. Aceste tipuri de aplicaţii utilizează de regulă o cantitate mare de date complexe. Dacă în trecut pentru o mare perioadă de timp bazele de date relaţionale esenţiale pentru aplicaţii web erau MySQL, astăzi aceste baze de date relaţionale întâmpină multe dificultăţi în lucru cu cantităţi mari de date. Pe piaţa în care activează MySQL au pătruns furnizorii de soluţii de baze de date cloud. Aceste baze de date cloud poartă numele de NoSQL - Not only SQL şi sunt baze de date non relaţionale. Dezvoltarea NoSQL şi NewSQL ameninţă monopolul MySQL.

Diversele baze de date NoSQL existente azi pe piaţă prezintă diferite abordari. Ceea ce au în comun este faptul că nu sunt relaţionale. Principalul avantaj este acela că permit lucrul eficient cu date nestructurate precum e-mail, multimedia, procesoare de text. În prezent există multe companii care au dezvoltat propriile baze de date NoSQL. Cele mai populare sunt cele dezvoltate de către companiile mari Web, precum Amazon şi Google, din nevoia de a procesa cantităţi mari de date. Acestea au dezvoltat Dynamo şi Big Table ce stau la baza multor alte baze de date NoSQL existente acum pe piaţă.

Unul din motivele apariţiei NoSQL constă în nevoia aplicaţiilor web de a manipula cantităţi mari de date a pentru a putea rămane competitive. Cantitatea de informaţie digitală la nivel mondial este măsurată în exabytes. Conform unui studiu realizat de Universitatea Southern California cantitatea de date adaugată în 2006 a fost de 161 de exabytes. Doar un an mai tarziu, în 2007 capacitatea totală s-a ridicat la 295 de exabytes, reprezentând o creștere substanțială. Altfel spus, există o cantitate mare de informaţie în lume şi aceasta creşte exponenţial. De aici survine şi nevoia de baze de date web ce suportă cantităţi mari de date.

Conform unui studiu realizat de 451 Group Research intitulat MySQL vs. NoSQL and NewSQL între 2009 şi 2011 s-a înregistrat o scădere în utilizarea MySQL de la 82% la 73%. Studiul a fost efectuat asupra unui eşantion compus din 347 de utilizatori de baze de date opensource. 49% din respondenţi au abandonat soluţiile MySQL pentru a migra la soluţii NoSQL. Astfel se poate observa ameninţarea directă pe care o presupune NoSQL asupra MySQL.

Baze de date de tip graf

O bază de date de tip graf este o bază de date care folosește grafuri ca structura de stocare a datelor. Nodurile (entitățile) și muchiile (relațiile dintre acestea) sunt principalele proprietăți care definesc o astfel de modelare a bazei de date. Muchiile dintre noduri permit acestora să poată fi conectate direct, astfel încât în multe situații datele pot fi preluate cu o operație simplă.

1 Introducere în "Baze de date NoSQL"

Bazele de date nerelationale (NoSQL) au apărut din necesitatea unei simplificari a designului și a sclabilitatii pe orizontală și îmbunătățirii timpui de răspuns. În unele situații bazele de date nerelationale sunt mai rapide decât cele relaționale (mai ales la dimensiuni foarte mari ăla bazelor de date sau în aplicațiile Web real-time.

Tipuri de baze de date nerelationale:

1. Coloana : este asemănător cu modelul relațional dar datele sunt păstrate în memorie pe coloane și nu pe linii ceea ce permite anumite operații de compresie a datelor și îmbunătățește performanță în anumite situații de utilizare și spațiul de memorie folosit;

2. Document : conceptul general este de Document în care se păstrează codificat datele că și XML, JSON, etc. iar documentele au o cheie unică și pot fi asociate cu înregistrări iar ele se păstrează în colecții care se pot asocia cu tabele;

3. Cheie-valoare : se bazează pe conceptul de dicționar (map) prin care se asociază fiecărei cheie o anumită valoare;

4. Graf : se aplică pentru acele modele în care relațiile dintre elemente sunt multiple spre exemplu rețele de socializare, rute de transport;

5. Multi-model : înglobează mai multe modele de baze de date nerelationale intenționând să ofere și acele proprietăți pe care doar bazele de date relaționale le ofereau.

Există actualmente o serie de sisteme de gestiune care se deosebesc de cele clasic relaționale prin una sau ambele din caracteristicile de mai jos:

- folosirea modelului relațional al datelor;
- folosirea limbajului de interogare SQL.

Astfel de sisteme se caracterizeaza in plus prin unele din elementele de mai jos:

- sunt proiectate pentru medii distribuite sau paralele;
- sunt folosite pentru gestiunea documentelor și mai putin a datelor atomice;
- oferă garanții mai slabe de consistentă;
- unele sunt proiectate pentru a oferi consistentă la citire, specifică în sistemele clasice;
- în cazul arhitecturilor distribuite datele sunt prezente redundant în mai multe noduri. În felul acesta este realizată scalabilitatea și protecția împotriva căderilor accidentale ale unor noduri;

Bazele de date NoSQL oferă o scalabilitate nelimitată cu performanțe importante, împărțind nodurile în toată baza de date, indiferent cât de mare ar fi aceasta. Deși bazele de date tradiționale bazate pe RDBMS încă domină piața, bazele NoSQL își fac loc in general în medii care necesită o procesare rapidă și de mare viteză. Termenul NoSQL nu este unul nou, fiind utilizat încă de la sfârșitul anilor 1990, cu unele modele care stau la baza acestuia dezvoltate chiar mai devreme. NoSQL a intrat pe piață la un nivel redus la mijlocul anilor 2000, urmând ca deja în 2010 NoSQL sa fie utilizat pentru aplicații critice, necesitând deja garanții de performanță. În prezent, NoSQL este utilizat pentru gestionarea bazelor de date ale unora din cele mai mare magazine de date din lume pentru aplicații precum rețele sociale, analiza datelor de la senzori și analiza pieței de valori.

2 Scurt istoric Neo4J

Neo4J reprezintă un sistem de management al unei baze de date de tip graf dezvoltat de compania Neo4J și este catalogat ca cel mai popular reprezentant al acestui tip de reprezentare a datelor. Acest sistem a pornit ca un concept în anul 2000 când s-a început dezvoltarea primului prototip Neo4J, ajungându-se că în 2002 să fie dezvoltată prima versiune de Neo4J, iar în 2003 acest sistem să fie lansat în producție, și să fie disponibil 24/7. Abia în 2010 este lansată versiunea 1.0 , pentru ca în 2016 să fie lansată versiunea 3.0. De asemenea, în 2017 Neo4J lansează "Graph Platform", un tool extrem de util pentru a vizualiza și a analiza date, și o să vedem în paginile următoare importanța acestuia.

Neo4J este implementat în Java și accesibil prin limbajul specific, dezvoltat de companie, Cypher.

Datorită eficienței integrării modelului pus la dispoziție de Neo4J, acesta a ajuns să fie utilizat de giganți din domeniu precum: Microsoft , eBay , IBM , LinkedIn etc.

3 Modelarea datelor interconectate

Neo4j este o baza de date open-source fondată pe teoria grafurilor, fiind o soluție optimizată pentru a modela și interoga volume mari de date strâns relaționate, reprezentabile prin structuri de tip graf. Dinamismul,creșterea volumului datelor, precum și evolutia continuă a procesării informațiilora impus ieșirea din spațiul bazelor de date relaționale tradiționale și orientarea spre soluții NoSQL. O caracteristică unică a acestora este gradul ridicat de adaptibilitate la modelele reale de date.

Atât în cazul bazelor de date relaționale cât și în cazul unor soluții NoSQL (non-graph), procesul de modelare/design trece prin două faze:

- definirea conceptelor , a entităților și a interacțiunii dintre ele - model logic/real;
- materializarea modelului logic într-un model fizic/abstract.

De cele mai multe ori modelul logic este foarte diferit de modelul fizic. În cadrul unei organizații software în prima fază poate participa orice echipă nu neapărat tehnică(management / sales) pentru o mai bună definire a cerințelor sau conceptelor.În cea de a doua fază însă are loc abstractizarea modelului logic în funcție de opțiunea de stocare. Astfel gradul de înțelegere al modelului logic scade odată cu creșterea complexității datelor.

Marele avantaj al bazelor de date de tip graf și implicit utilizarea Neo4j este că modelul logic este același cu modelul fizic. Având acest mod de reprezentare uniformă sau astfel spus o reprezentare "human readable" ce oferă un mare grad de flexibilitate , adaptibilitate și expresivitate în modelarea datelor reale.

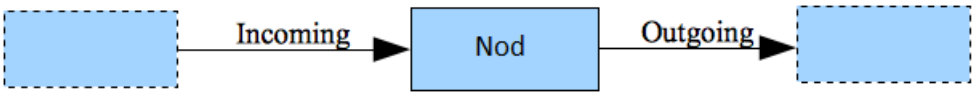
4 Reprezentarea datelor în Neo4J

În Neo4j datele sunt reprezentate prin noduri și relații. Atât nodurile cât și relațiile pot fi avea proprietăți. Relațiile au un rol foarte important în cadrul bazei de date de tip graf pentru că traversarea grafului și implicit manipularea datelor se realizează prin intermediul lor.

O relație implică întotdeauna două noduri, are o direcție și un tip identificat unic printr-un nume.

Relativ la un nod relațiile se pot clasifica în două tipuri:

- incoming
- outgoing



Atât proprietățile unui nod cât și cele ale unei relații pot fi indexate pentru îmbunătățirea performanței de traversare a grafului , similar cu indexarea coloanelor în bazele de date tradiționale.

Forțând o comparație cu bazele de date tradiționale, vă puteți imagina un nod ca o înregistrare dintr-un tabel, iar o relație ca o înregistrare dintr-un tabel de legătură sau o pereche de coloane din același tabel în cazul unei reprezentări tip denormalizat.

5 Limbajul de interogare Cypher

Neo4j are propriul limbaj de interogare a datelor organizate în structuri de graf. Este folosit conceptul de "Traversal" prin intermediul căruia se navighează în graf, se identifică drumurile și implicit se selecteză nodurile pentru rezultatul unei interogări.

Limbajul Cypher este un limbaj de interogare declarativ fiind foarte intuitiv și "human readable", putând fi înțeles cu ușurință chiar și de o persoană non-tehnică. Unele cuvinte cheie sunt inspirate din SQL cum ar fi: where, order by , limit, skip (echivalentul offset).

Limbajul este alcătuind din următoarele clauze :

- START - punctul de intrare în graf. Orice interogare în graf are cel puțin un nod de start;
- MATCH - șablonul pentru căutarea nodurilor și care este legat de nodul de start;
- WHERE - condițiile de filtrare a nodurilor / relațiilor;
- RETURN -rezultatul interogării;
- CREATE -creează noduri sau relații;
- DELETE -șterge noduri sau relații;
- SET -setează proprietăți noduri sau relații;
- FOREACH -update pe liste de noduri;
- WITH -împarte interogarea în mai multe părți distincte.