

# VIITORUL BAZELOR DE DATE CU NOSQL

ȚARANOVICI Vasile

Universitatea Tehnică a Moldovei

**Abstract:** În articol sunt expuse bazele de date de tip non-relațional (NoSQL), avantajele și dezavantajele, utilizarea și compararea cu bazele de date SQL. NoSQL recunoscut ca Not Only SQL reprezintă un pas curajos spre dezvoltarea bazelor de date prin implementarea unor noi tipuri de date (Cheie-valoare, Document, Coloană și Graf) care ușurează considerabil stocarea datelor cât și viteza de scriere/citire. În articol este abordat topicul „utilizarea bazelor de date NoSQL”, nivelul de popularitate și motivele utilizării NoSQL în unele industrii.

**Cuvinte cheie:** baze de date relaționale, NoSQL, MongoDB, SQL vs NoSQL, utilizare NoSQL, performanța NoSQL, consistența datelor, open source.

## 1. Introducere

Dacă în trecut pentru o mare perioadă de timp bazele de date relaționale esențiale pentru aplicațiile web erau MySQL, astăzi aceste baze de date relaționale întâmpină multe dificultăți în lucrul cu volume mari de date. Pe piața în care activează MySQL au pătruns furnizorii de soluții de baze de date cloud. Aceste baze de date cloud poartă numele de NoSQL- Not only SQL și sunt baze de date non relaționale.

### 1.1. Problemele bazelor de date relaționale

- Operații mult prea complexe: select, update, delete;
- Nu suportă mix-ul de date nestructurate;
- Nu sunt scalabile la operarea cu volume mari de date;
- Nu permit ca datele să fie salvate oriunde și oricând, fără a efectua o verificare.

### 1.2. Conceptul NoSQL

Termenul NoSQL a fost inventat de Eric Evans, un angajat al companiei Rackspace. Numele a fost o încercare de a descrie apariția unui număr tot mai mare de BD non-relaționale, distribuite, ce stochează date care nu tot timpul oferă garanții ACID (Atomicitate, Consistență, Izolare și Durabilitate).

NoSQL sunt baze de date care oferă un mecanism de stocare și recuperare a datelor, modelate în alte mijloace decât relațiile tabelare folosite în bazele de date relaționale.

## 2. NoSQL versus SQL

### 2.1. Tipuri de date

- Cheie-Valoare (Redis, Couchbase Server)

Este unul din cele mai simple concepte NoSQL. În acest tip, fiecare element este înregistrat cu o cheie. Utilizatorul poate căuta sau șterge datele utilizând valoarea cheii respective.

- Document (MongoDB, CouchDB)

În acest tip datele sunt stocate într-un document cu o cheie unică. Documentul poate conține multe perechi de valori-chei, tablouri de chei sau perechi de formulare ierarhice.

Acest tip este proiectat pentru stocarea, obținerea și moderarea informației orientată pe documente (semi-structurată).

- Coloană (Cassandra, HBase)

În modul Coloană, stocăm datele ca secțiuni a coloanelor de date (nu pe rânduri ca în SQL). De obicei acest tip oferă performanțe foarte mari și o arhitectură scalabilă eficientă.

- Graf (Neo4J, HyperGraphDB)

Aceste baze sunt proiectate pentru date ale căror relații pot fi reprezentate ca grafuri și au elemente ce sunt interconectate cu un număr nedeterminat de relații între ele.

### 2.2. Definierea structurii. Scheme

SQL: În SQL, structura și tipurile de date trebuie predefinite înainte de a introduce datele. Dacă se dorește modificarea tipului de date, baza de date va fi afectată

În NoSQL nu apare obligațiunea definirii structurii. Date diverse pot fi stocate împreună cu datele existente fără a afecta baza de date.

### 2.3. Dezvoltare și consistență

SQL are un nivel foarte înalt de consistență iar NoSQL are câteva sisteme care sunt foarte inconsistente (MongoDB) și câteva sisteme consistente (Cassandra).

În SQL, câteva sisteme de gestiune a bazelor de date sunt open-source (Postgres, MySQL) însă sunt și closed-source, cum ar fi SQL Server și Oracle. NoSQL este complet open-source ceea ce oferă acces liber mai multor dezvoltatori.

### 2.4. Scalarea

SQL funcționează după principiul scalării verticale. În acest caz, se adaugă resurse în aceeași unitate logică pentru a mări capacitatea ei. De exemplu: adăugarea unui CPU, memorie externă sau operativă.

NoSQL operează cu scalarea pe orizontală (în afară). În această metodă, se adaugă noduri noi către sistem (servere) astfel încât toată prelucrarea datelor va fi distribuită pe mai multe servere (figura 1).

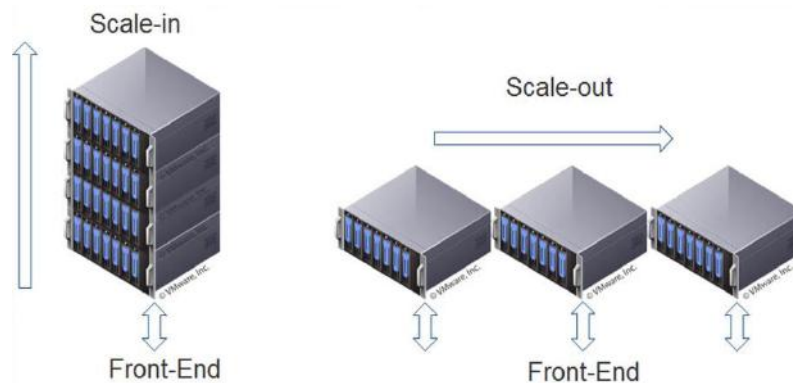


Fig. 1. Sclarea SQL vs NoSQL

## 3. Utilizarea NoSQL

Big Data în timp real - abilitatea de a extrage informații din datele operaționale în timp real - este esențială pentru o întreprindere agilă. Aceasta sporește eficiența operațională, reduce costurile și crește veniturile, permiând acționarea imediată pe datele curente.

Vizualizarea Client 360 ° - pe măsură ce crește numărul de produse și servicii, canale, mărci și unități de afaceri, modelul de date fixe al bazelor de date relaționale obligă întreprinderile să fragmenteze datele clienților, deoarece diferite aplicații funcționează cu date diferite ale clienților. Tipul de date document din NoSQL utilizează un model de date flexibil, care permite mai multor aplicații să acceseze aceleași date ale clienților, precum și să adauge attribute noi fără a afecta alte aplicații.

Aplicații mobile - cu aproape două miliarde de utilizatori de smartphone-uri, aplicațiile mobile se confruntă cu provocări de scalabilitate în ceea ce privește creșterea și volumul. De exemplu, nu este neobișnuit ca jocurile mobile să ajungă la zeci de milioane de utilizatori într-o lună de lucruri. Cu o bază de date distribuită, scalabilă, aplicațiile mobile pot începe cu o mică implementare și se extind ca baza de utilizatori.

Internetul Lucrurilor - aproximativ 20 de miliarde de dispozitive sunt conectate la Internet - totul, de la telefoane inteligente și tablete până la electrocasnice și sisteme instalate în mașini, spitale și depozite. Volumul, viteza și varietatea datelor generate de mașină sunt în creștere odată cu proliferarea telemetriei digitale, care este semi-structurată și continuă (figura 2).

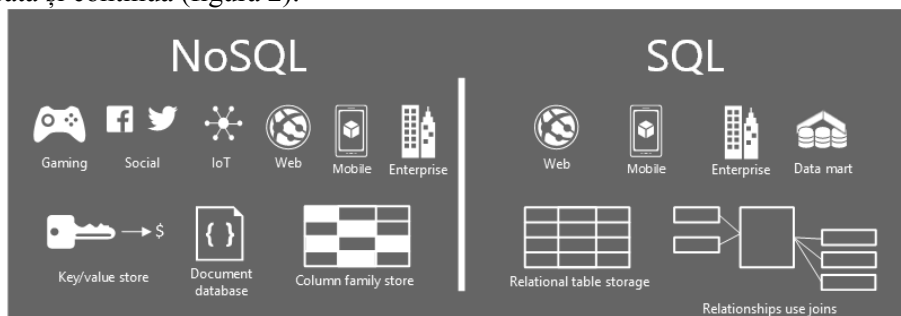


Fig. 2. Utilizarea NoSQL vs SQL

Detectarea fraudelor - pentru organizațiile de servicii financiare, detectarea fraudei este esențială pentru reducerea pierderilor de profit, minimizarea expunerii financiare și respectarea reglementărilor. În timp ce bazele de date relaționale se străduiesc să îndeplinească această cerință de latență redusă, bazele de date NoSQL elastice sunt scalabile și pot furniza în mod fiabil performanța necesară.

#### **4. Puncte forte și puncte slabe ale NoSQL**

##### **4.1 Puncte forte**

- Sunt mai deschise și flexibile și permit adaptarea la specificațiile unui proiect mai ușor decât în modelul relațional;
- Modificările pot fi efectuate fără a opri baza de date;
- Posibilitatea de a crește în numărul de dispozitive (sc. orizontală);
- Pot opera cu dispozitive ce nu consumă multe resurse;
- Optimizarea interogărilor este proiectată pentru seturi foarte mari de date.

##### **4.2 Puncte slabe**

- Concentrare îngustă: NoSQL este structurat mai mult pentru stocarea rapidă a seturilor mari de date, însă îi lipsește funcționalitatea. La capitolul moderarea tranzacțiilor, bazele relaționale sunt mai eficiente;
- NoSQL este open-source. Nu există un standard de încredere;
- Management de date: Scopul Big Data este de a face lucrul cu multe date cât mai simplu posibil. Management-ul de date în NoSQL este mult mai complex decât într-o bază relațională;
- Backup: la acest capitol, NoSQL este foarte slab, în special MongoDB care nu are o metodă consistentă de a face backup la date;
- Documente de dimensiuni mari: BD ca MongoDB și CouchDB stochează datele în format JSON și aceste documente tind să devină foarte mari.

#### **Concluzii**

1. Odată cu apariția bazelor de date NoSQL, dezvoltatorii au oportunitatea de a beneficia de mai multă agilitate în modelul de date abordat. De asemenea, aceste baze de date constituie modelul optim pentru aplicațiile web. De aceea cunoașterea caracteristicilor lor este foarte importantă, în special înainte de a migra la o astfel de soluție.

2. NoSQL reprezintă o trecere către baze de date superioare ce vor integra flexibilitatea și performanțele lor actuale cu modelul relațional.

3. NoSQL nu înlocuiește BD relaționale. Este dezvoltat pentru a înregistra un număr foarte mare de date ce nu pot fi înscrise de BD relaționale. În unele cazuri este utilizat alături de SQL.

4. NoSQL este o tehnologie în continuă creștere și are nevoie de multe îmbunătățiri la capitolul documentație, suport, backup și management-ul datelor înregistrate.

#### **Bibliografie**

1. *Introduction to NoSQL Databases*. [Resursă electronică]. - Regim de acces: <http://www.c-sharpcorner.com/UploadFile/f0b2ed/introduction-of-nosql-database/>
2. *SQL vs NoSQL*. [Resursă electronică]. - Regim de acces: <https://www.sitepoint.com/sql-vs-nosql-choose/>
3. *NoSQL vs SQL. The key differences*. [Resursă electronică]. - Regim de acces: <https://blog.pandorafms.org/nosql-vs-sql-key-differences/>
4. *10 use cases where NoSQL will outperform SQL*. [Resursă electronică]. - Regim de acces: <https://www.networkworld.com/article/2999856/big-data-business-intelligence/10-use-cases-where-nosql-will-outperform-sql.html>
5. *Java tools and technologies landscape for 2014 (Rebellabs)*. [Resursă electronică]. - Regim de acces: <https://zeroturnaround.com/rebellabs/java-tools-and-technologies-landscape-for-2014/>