

## PERFORMANȚA BAZELOR DE DATE: SQL VS NOSQL

**Eutalia PISTRUI**

*Departamentul Ingineria Software și Automatică, grupa TI-215, Facultatea CIM,  
Universitatea Tehnică a Moldovei, Chișinău, Republica Moldova*

Autorul corespondent: Eutalia Pistrui, [eutalia.pistrui@isa.utm.md](mailto:eutalia.pistrui@isa.utm.md)

Îndrumător/coordonator științific: **Dorian SARANCIUC**, lector universitar,  
Departamentul Ingineria Software și Automatică

**Rezumat.** Datorită progreselor din tehnologia informației din ultimele decenii organizațiile au înregistrat o creștere semnificativă a cantității de date pe care le generează și le gestionează. Un aspect semnificativ al acestei evoluții este selectarea și utilizarea adecvată a bazei de date pentru a garanta stocarea, gestionarea și recuperarea eficientă a datelor. Bazele de date SQL, care acordă prioritate consistenței și integrității sunt potrivite pentru aplicațiile cu cerințe exigente de gestionare a datelor. Cu toate acestea, bazele de date NoSQL, care acordă prioritate scalabilității și disponibilității, sunt cele mai bune alegeri pentru scenariile care necesită gestionarea eficientă a volumelor mari de date, fiind, de asemenea, flexibile la schimbările frecvente ale structurii informațiilor. În acest context, articolul analizează și compară performanțele între bazele de date SQL și cele NoSQL, evidențiind diferențele semnificative în structură, modelul de date, operațiile CRUD, scalabilitate, disponibilitate, consistență și toleranța la partiționare.

**Cuvinte cheie:** baze de date SQL (Structured Query Language), baze de date NoSQL (Not Only SQL), scalabilitate, operațiile CRUD (Create, Read, Update, Delete), structură, model de date, performanțe.

### Introducere

Pe măsură ce cantitatea de date generate în era digitală crește exponențial, organizațiile au fost nevoite să selecteze sistemul de gestionare a bazelor de date potrivit pentru a asigura o funcționare eficientă și flexibilitate operațională. Acest proces de selecție devine mai complex odată cu diversificarea ofertei de baze de date, cu un accent deosebit pe două categorii majore: bazele de date SQL și cele NoSQL.

Sistemele de gestionare a datelor au depins pe scară largă de bazele de date SQL, care utilizează un model relațional bine definit, caracterizat de tabele structurate și un limbaj standardizat de interogare. Cu toate acestea, odată cu progresul tehnologic și diversificarea datelor, au apărut soluțiile NoSQL, care pun accentul pe flexibilitate și scalabilitate. Acestea se adaptează mai eficient la cerințele aplicațiilor moderne, reprezentând o alternativă semnificativă în contextul evoluției continue a tehnologiei. În acest context, prezentul articol își propune să ofere o analiză detaliată a diferențelor dintre cele două tipuri de baze de date, cu accent pe aspecte precum structura și modelul de date, performanța în operațiunile CRUD, scalabilitatea, disponibilitatea și adaptabilitatea la situații de partiționare.

### Structura și modelul de date

Bazele de date SQL, cunoscute pentru modelul lor relațional, adoptă o structură clar definită în care datele sunt organizate în tabele cu coloane predefinite și relații între ele. Această structură strictă oferă beneficii în ceea ce privește consistența datelor și capacitatea de a efectua operațiuni complexe de interogare.

Pe de altă parte, bazele de date NoSQL ne fiind relaționale, nu stochează exclusiv date în rânduri și tabele, având o abordare mai flexibilă. Acestea permit utilizarea unor modele de date variate:

- documente (Document-oriented databases) - care utilizează documente pentru a ține și a codifica datele în formate standard, inclusiv XML (eXtensible Markup Language), YAML (YAML Ain't Markup Language), JSON (JavaScript Object Notation) și BSON (Binary JSON).
- chei-valoare (Key-Value stores) - care utilizează un array asociativ (cunoscut și sub numele de dicționar sau hartă) ca model de date, acest model reprezintă datele ca o colecție de perechi cheie-valoare.
- coloane (Column-family stores) - unde datele sunt stocate în celule grupate într-un număr virtual nelimitat de coloane în loc de rânduri.
- grafuri (Graph databases) - care reprezintă datele pe un grafic care arată cum diferite seturi de date se relaționează între ele, Neo4j, RedisGraph (un modul grafic încorporat în Redis) și OrientDB sunt exemple de baze de date grafice [1].

Această flexibilitate permite dezvoltatorilor să modeleze datele în funcție de cerințe specifice ale aplicației, eliminând restricțiile impuse de structura tabelară.

Comparativ cu bazele de date SQL, care pot impune o anumită rigiditate în structura datelor, bazele de date NoSQL se remarcă prin abilitatea de a se adapta rapid la modificările în modelele de date, oferind un grad mai înalt de libertate dezvoltatorilor în procesul de proiectare a schemelor.

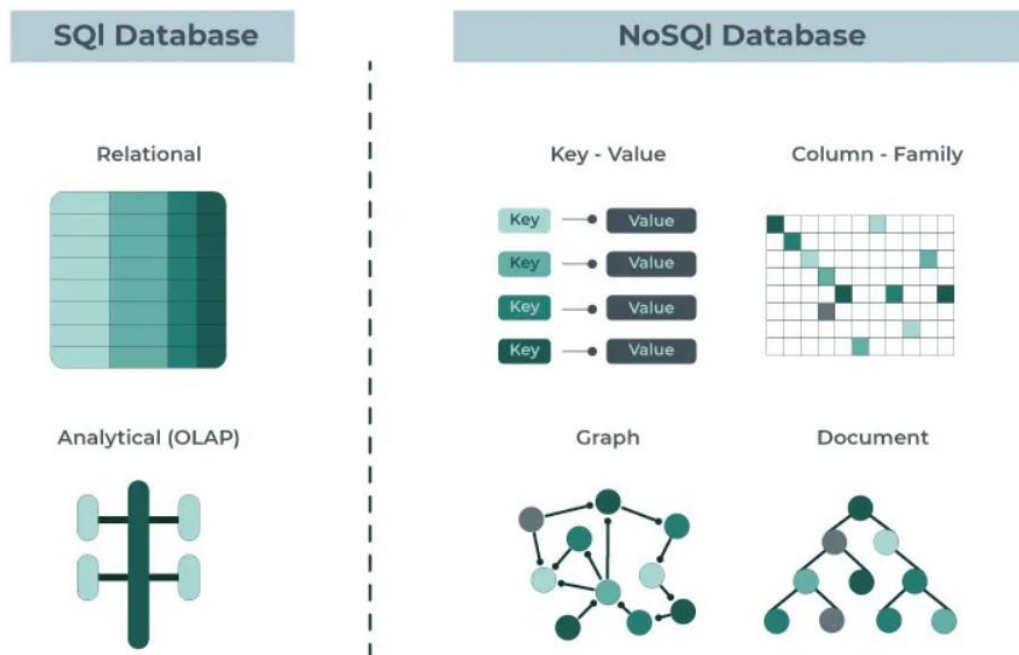


Figura1. Structura datelor SQL vs NoSQL [2]

### Performanța în operațiunile CRUD

Operațiunile CRUD reprezintă o funcționalitate fundamentală a oricărui sistem de gestionare a bazelor de date și sunt criteriile substanțiale pentru compararea performanței bazelor de date SQL și NoSQL.

Bazele de date SQL excelează la operațiuni de citire datorită structurii lor relaționale. Interogările complexe pot fi efectuate eficient, iar optimizarea interogărilor asigură o recuperare rapidă a datelor. Cu toate acestea, atunci când se efectuează operațiuni de scriere precum Crearea și Actualizarea, bazele de date SQL pot întâmpina unele dificultăți. Implementarea modificărilor în bazele de date SQL necesită adesea actualizări asincrone de index și verificări ale integrității referențiale, care pot afecta performanța în scenarii cu volum mare.

În schimb, bazele de date NoSQL se remarcă în ceea ce privește performanța operațiunilor de scriere, în special Create și Update. Cu o abordare flexibilă și structuri de date mai simple, acestea pot gestiona eficient un volum mare de inserări și actualizări simultane. Bazele de date NoSQL precum MongoDB sau Cassandra folosesc frecvent tehnici precum partiționarea și replicarea pentru a asigura o distribuție uniformă a încărcării și pentru a îmbunătăți performanța de scriere [3].

Este important de subliniat faptul că eficacitatea operațiunilor CRUD poate fluctua în funcție de contextul de utilizare. În cazurile în care aplicația se concentrează în principal pe extragerea de informații, o bază de date SQL poate fi alegerea potrivită. În schimb, pentru aplicațiile care vizează scrieri și actualizări frecvente, bazele de date NoSQL pot oferi avantaje semnificative de performanță.

### **Consistență și toleranța la partiționare**

Bazele de date SQL au tendința de a pune un accent mai mare pe consistență, deoarece aderă la principiul ACID (Atomicity, Consistency, Isolation, Durability). Consecvența asigură că toate nodurile sau serverele din sistem împărtășesc o perspectivă unificată și consecventă a datelor la un moment dat. Cu toate acestea, această abordare poate implica așteptarea confirmării actualizărilor de la toate nodurile, ceea ce poate afecta performanța, în special în mediile distribuite. Toleranța la partiții este adesea o problemă delicată pentru bazele de date SQL. În cazul unei partiții, în care comunicarea între noduri sau servere este întreruptă, sistemele distribuite trebuie să ia decizii cu privire la modul de a naviga în această situație fără a compromite consistența.

Bazele de date NoSQL sunt renumite pentru capacitatea lor de a oferi o mai mare flexibilitate atunci când vine vorba de echilibrarea consistenței, disponibilității și toleranței partițiilor. Acest lucru este în conformitate cu principiile prezentate de Teorema CAP. Aceste baze de date pun de obicei un accent mai mare pe asigurarea disponibilității și toleranței partițiilor, ceea ce le permite să continue să funcționeze chiar dacă există posibile întreruperi în comunicarea între diferite noduri.

Alegerea dintre bazele de date SQL și cele NoSQL în ceea ce privește consistența și toleranța la partiționare depinde în mare măsură de natura aplicației. Dacă o consistență strictă este esențială și sistemul se desfășoară într-un mediu cu o conectivitate stabilă între noduri, o bază de date SQL poate fi preferată. În schimb, dacă aplicația poate tolera o consistență eventuală și are nevoie de o reziliență sporită în condiții de partiționare, o bază de date NoSQL ar putea fi o alegere mai potrivită.

### **Scalabilitate și disponibilitate**

Scalabilitatea și disponibilitatea sunt două caracteristici fundamentale în arhitectura sistemului de baze de date, iar diferențele dintre bazele de date SQL și NoSQL devin evidente în aceste aspecte.

Bazele de date SQL adoptă frecvent o arhitectură monolitică, ceea ce poate face procedura de scalare orizontală mai complicată. Scalare orizontală reprezintă adăugarea de noi noduri sau servere pentru a dispersa volumul de lucru și a extinde capacitatea sistemului. În cazul bazelor de date relaționale, acest curs de acțiune poate fi complicat din cauza gestionării asocierilor dintre tabele și a menținerii coerenței în datele distribuite. Manipularea informațiilor sensibile și serviciile financiare sunt domenii în care bazele de date SQL sunt foarte apreciate datorită proprietăților lor de înaltă consistență și integritate a datelor.

Bazele de date NoSQL sunt concepute pentru a oferi o scalabilitate orizontală mai ușoară. Aceste baze de date folosesc adesea modele de date distribuite care pot gestiona eficient cantități mari de date prin adăugarea de noduri sau clustere. Replicarea și sharding sunt tehnici comune pentru creșterea capacității de stocare și îmbunătățirea performanței. Ceea ce ține de disponibilitate, bazele de date NoSQL sunt construite pentru a face față defecțiunilor hardware sau ale unui singur nod. Acestea oferă disponibilitate continuă chiar și în cazul în care o parte a sistemului se defectează.

Alegerea dintre bazele de date SQL și NoSQL în ceea ce privește scalabilitatea și disponibilitatea depinde de cerințele specifice ale aplicației. Dacă o aplicație necesită scalabilitate ușoară și poate accepta o eventuală consistență, o bază de date NoSQL poate fi o alegere potrivită. În schimb, dacă consistența și integritatea datelor sunt o prioritate, iar cerințele de scalabilitate sunt mai mici, o bază de date SQL poate fi mai potrivită.

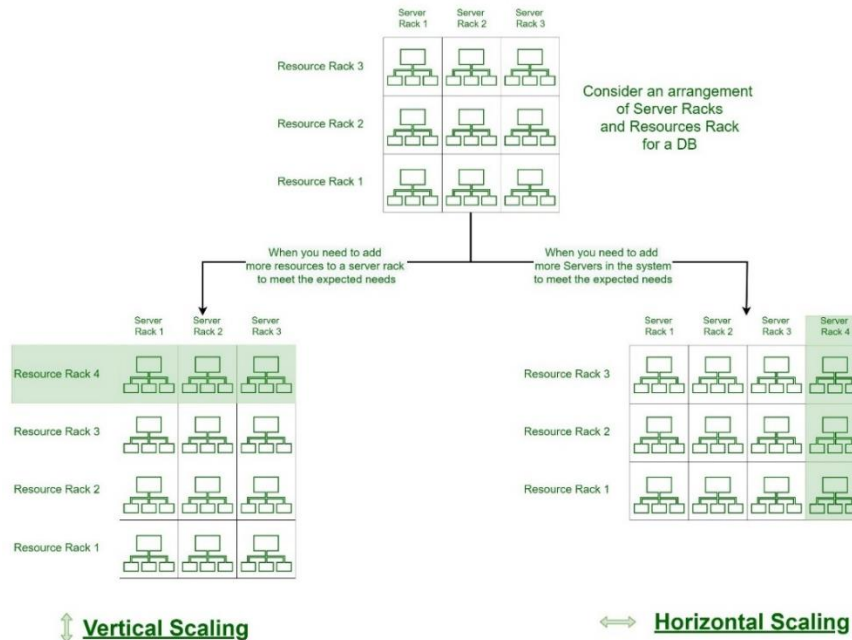


Figura 2. Scalabilitate verticală și orizontală [4]

### Concluzii

După o analiză aprofundată a performanței dintre bazele de date SQL și NoSQL, devine evident faptul că alegerea dintre cele două depinde în mare măsură de cerințele specifice ale aplicației și ale contextului de implementare. Fiecare dintre aceste abordări are beneficii și compromisuri distincte, iar decizia finală trebuie luată pe baza priorităților specifice ale proiectului. Bazele de date SQL sunt o soluție de încredere pentru tratarea domeniilor sensibile la date datorită consistenței și integrității lor. În schimb, bazele de date NoSQL se concentrează pe scalabilitate și disponibilitate, fiind cele mai potrivite pentru aplicațiile care necesită flexibilitate în manipularea unor cantități mari de date.

### Bibliografie:

- [1] Explore key differences between SQL and NoSQL databases and learn which type of database is best for various use cases. [online] [accesat 14.01.2024]. Disponibil: <https://www.ibm.com/blog/sql-vs-nosql/>
- [2] When To Use SQL Databases Vs. NoSQL Databases: Making The Right Decision. [online] [accesat 14.01.2024]. Disponibil: <https://expeed.com/when-to-use-sql-databases-vs-nosql-databases-making-the-right-decision/>
- [3] Pros and Cons of Using SQL vs NoSQL Databases [online] [accesat 14.01.2024]. Disponibil: <https://www.codingninjas.com/studio/library/pros-and-cons-of-using-sql-vs-nosql-databases>
- [4] Horizontal and Vertical Scaling In Databases [online] [accesat 14.01.2024]. Disponibil: <https://www.geeksforgeeks.org/horizontal-and-vertical-scaling-in-databases/>