

# NOUA GENERAȚIE DE BAZE DE DATE NoSQL

Elena-Geanina Ularu

ularugeanina@yahoo.com

Florina Puican

puicanflorina@yahoo.com

Institutul de Studii Doctorale București

**Rezumat:** Bazele de date tradiționale sunt puse la încercare din ce în ce mai mult de noile tipuri de aplicații care le folosesc. Aceste tipuri de aplicații utilizează de regulă o cantitate mare de date complexe. Dacă în trecut pentru o mare perioadă de timp bazele de date relaționale esențiale pentru aplicații web erau MySQL, astăzi aceste baze de date relaționale întâmpină multe dificultăți în lucru cu cantități 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. Dezvoltarea NoSQL și NewSQL amenință monopolul MySQL.

În acest articol vom analiza caracteristicile bazelor de date NoSQL, arătând care sunt avantajele și dezavantajele acestora față de modelul relațional tradițional și cum pot fi folosite pentru a aduce un plus organizației. De asemenea articolul de față conține o scurtă analiză a funcționalităților bazei de date NoSQL, MongoDB, pentru a exemplifica practic ușurința trecerii de la SQL la NoSQL și simplitatea dezvoltării de aplicații în acest tip de baze de date.

**Cuvinte cheie:** Baze de date, NoSQL, SQL, model relațional, open-source.

**Abstract:** Traditional relational database management systems are being challenged by the increasing requirements of new application types. This kind of applications usually need to work with large amounts of complex type of data. If for a long time open source relational databases like MySQL represented the basis for Web applications nowadays this well known RDBMS faces a lot of challenges from the alternative database providers. The market where MySQL activates has been opened to cloud database providers. These databases are called NoSQL- Not only SQL and they represent non relational databases. The development of NoSQL and NewSQL and the new features of this type of databases are threatening the MySQL monopoly.

In this paper we are analyzing the characteristics of NoSQL databases, stressing the advantages and disadvantages compared to the relational model and showing how can these databases be used to help the organization. In addition, this paper presents a short analysis on the features of MongoDB, in order to give a practical example on the switch between SQL and NoSQL and the ease in development with the latest kind of database.

**Keywords:** Databases, NoSQL, SQL, relational model, open-source.

## 1. Introducere

Orice organizație colecționează cantități mari de date constând în informații necesare pentru buna desfășurare a activității de business. Aceste informații sunt stocate pentru a putea fi folosite în analize viitoare, precum previziuni și analize de business, cu scopul de a aduce un avantaj competitiv. În mod tradițional, organizațiile stochează aceste date în baze de date relaționale pentru a putea avea acces la ele și pentru a putea desfășura analizele de business necesare. Cu toate acestea, în ultima vreme mulți dintre dezvoltatori au început să implementeze și să lucreze cu baze de date non relaționale numite NoSQL-Not Only SQL.

Diversele baze de date NoSQL existente azi pe piață prezintă diferite abordări. Ceea ce au în comun este faptul că nu sunt relaționale. 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 2.0, precum Amazon and Google, din nevoia de a procesa cantități mari de date. Acestea au dezvoltat Dynamo și Big Table ce stau la baza multor alte baze de date NoSQL existente acum pe piață.

### La început

Limbajul SQL a fost inventat în 1970 la IBM. Acest limbaj a fost dezvoltat pentru a lucra cu date structurate conform modelului relațional al lui Edgar F. Codd și reprezintă limbajul standard pentru bazele de date relaționale. SQL permite construirea de interogări pentru analiza unor cantități mari de date structurate și contribuie la buna desfășurare a activităților economice ale unei companii. Un exemplu concret este reprezentat de obținerea de indicatori de performanță pentru

fiecare departament în parte.

Deși în prezent nu există competitori reali pentru SQL, problema se schimbă în cazul aplicațiilor web. În acest caz, nu există o multitudine de asocieri inner și outer pentru calcule complexe ci mai degrabă se folosește o gândire orientată obiect, îndeosebi datorită MVC (Model–View–Controller). Pentru a transforma aceste modele orientate obiect în baze de date relaționale au loc diverse procese de normalizare ce rezultă în ierarhii complexe de tabele și îndepartează întregul model de principiile modelării orientate obiect. Faptul că limbajul SQL permite interogări dinamice asupra unor seturi de date complexe este nefolositor prin folosirea unei baze de date SQL doar pentru stocarea persistentă a datelor orientate obiect.

Aici intervin bazele de date NoSQL. Acestea permit dezvoltatorului de aplicații să stocheze date care nu au o schemă predefinită. Carlo Strozzi a folosit prima dată termenul de NoSQL în 1998, reprezentând numele bazei sale de date relaționale open-source fără interfață SQL. Termenul a fost reintrodus în 2009 de către Eric Evans în cadrul unui eveniment cu tema baze de date distribuite open-source. De această dată termenul nu a fost folosit pentru a defini un întreg sistem, ci a fost folosit pentru a marca un pas în evoluția de la baze de date relaționale către baze de date cu performanțe sporite. Din acest moment s-au dezvoltat un număr vast de baze de date non-relaționale. [1]

O definiție a bazelor de date NoSQL este dată de site-ul nosql-database.org. Acesta caracterizează bazele de date NoSQL ca noua generație de baze de date ce îndeplinesc următoarele condiții: nu sunt relaționale, sunt distribuite, open-source și se caracterizează prin scalabilitate orizontală. Alte caracteristici ce trebuie menționate sunt lipsa unei scheme pentru a modela baza de date, prezintă suport pentru replicare, API, nu respectă în întregime ACID, stochează o cantitate mare de date.[2]

Unul din motivele apariției NoSQL constă în nevoia aplicațiilor web de a manipula cantități mari de date a pentru a putea rămâne competitive. Cantitatea de informație digitală la nivel mondial este măsurată în exabytes. Conform unui studiu realizat de Universitatea Southern California cantitatea de date adăugată în 2006 a fost de 161 de exabytes. Doar un an mai târziu, în 2007 capacitatea totală s-a ridicat la 295 de exabytes, o creștere cu aproape 200%. Altfel spus, există o cantitate mare de informație în lume și aceasta crește exponențial. De aici survine și nevoia de baze de date web ce suportă cantități mari de date.[3]

O bază de date NoSQL ignoră principiile RDBMS și nu stochează date folosind tabele ci folosind chei de identificare. Datele pot fi regăsite în funcție de cheile asignate. Părerea noastră este că acest tip de baze de date evadează din rigorile relaționale prin lipsa unei scheme, lipsa necesității de normalizare a datelor și de stocare a relațiilor dintre tabele aducând astfel performanțe sporite aplicațiilor care le folosesc. De asemenea acest tip de baze de date îmbunătățesc și răspunsul la schimbări de-a lungul timpului. Într-un sistem relațional nu există flexibilitatea necesară pentru a asimila modificări în modelul de date. Faptul că bazele de date NoSQL nu au o schemă de date fixă face aceste baze de date să fie mult mai flexibile și adaptabile la schimbări de model în cursul anilor.

Conform definiției date de Rick Cattell, bazele de date NoSQL prezintă șase trăsături de bază:

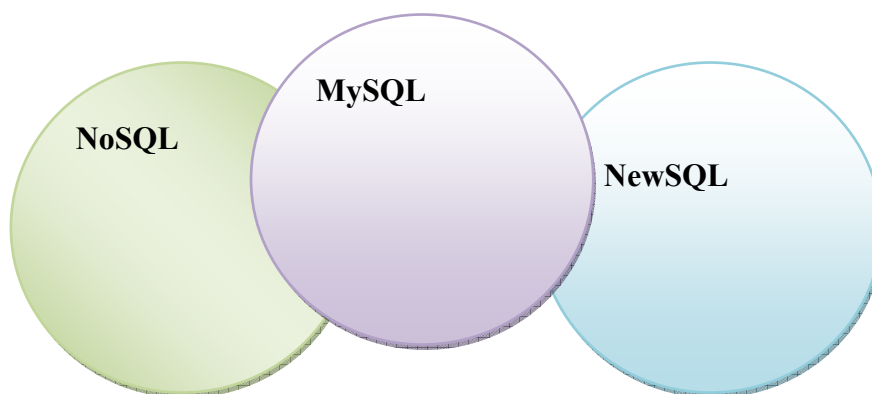
1. Abilitatea de a scala orizontal pe mai multe servere;
2. Abilitatea de a replica și distribui datele pe mai multe servere;
3. CLI (call level interface) caracterizat prin simplitate (în contrast cu SQL binding);
4. Un model concurențial mai slab decât modelul relațional (ACID);
5. Utilizarea eficientă a indexării distribuite și a RAM pentru o stocare eficientă;
6. Abilitatea de a adăuga dinamic noi atribute la înregistrările existente.[4]

O altă caracteristică importantă a sistemelor NoSQL este arhitectura „shared nothing” prin care fiecare nod/server este independent, nici unul din ele nu partajează memorie sau spațiu. Datorită acestei caracteristici pot fi efectuate un număr mare de operații de citire/scriere pe

secundă. O simplă operație poartă numele de OLTP (online transaction processing) și este de asemenea comună și în cadrul aplicațiilor web moderne.

Conform unui studiu realizat de 451 Group Research intitulat MySQL vs. NoSQL and NewSQL între 2009 și 2011 s-a înregistrat o scădere în utilizarea MySQL de la 82% la 73%. Studiul a fost efectuat asupra unui eșantion compus din 347 de utilizatori de baze de date open-source. 49% din respondenți au abandonat soluțiile MySQL pentru a migra la soluții NoSQL. Astfel se poate observa amenințarea directă pe care o presupune NoSQL asupra MySQL.[5]

Conform aceluiași studiu MySQL este direct amenințat de apariția NoSQL și NewSQL, acesta din urmă reprezentând un nou set de baze de date relaționale care încearcă să adauge la modelul relațional performanțele și funcționalitățile bazelor de date NoSQL.



**Figura 1. Amenințarea NoSQL și NewSQL asupra MySQL [5]**

Având la dispoziție toate aceste informații despre conceptul NoSQL putem formula o definiție proprie a ceea ce considerăm NoSQL. Astfel mișcarea NoSQL reprezintă o încercare de a depăși limitările modelului relațional și un pas de trecere către NewSQL și anume relațional plus extra funcționalități NoSQL.

## **2. Avantajele folosirii sistemelor NoSQL**

În bazele de date relaționale (MySQL, MSSQL, PostgreSQL) optimizările se fac prin definițiile index. Modelul de date pune accent pe normalizarea, abstractizarea și integritatea datelor. Baze de date NoSQL au apărut ca noua generație de baze de date ce permite cereri la noi nivele și fac față nevoilor tehnologiilor din ziua de azi, prin arhitecturi construite pe cloud și sisteme distribuite. Bazele de date non-relaționale sunt distribuite și memorează date care nu respectă garanțiile ACID.

Utilizatorii nu se mai mulțumesc doar cu rolul de a folosi bazele de date, ci doresc și să producă conținut și astfel s-a produs o restructurare a modalității de stocare și folosire a datelor de pe Web. Din acest punct de vedere, bazele de date relaționale nu mai prezintă eficiență din perspectiva timpului, în special în contextul aplicațiilor mari.

Printre primele aplicații mari, ce au ridicat problema scalabilității, se număra aplicația Web Facebook. Această rețea de socializare presupune un număr impresionant de utilizatori, la nivelul milioanei. Prin adoptarea bazelor de date NoSQL scalarea masivă a adus un beneficiu enorm.

Unul din primele sisteme NoSQL pentru managementul datelor a fost proiectat inițial de Facebook și se numește „Cassandra”. Acest sistem distribuit permite stocarea masivă de date și este bazat pe structura BigTable. Această structură oferă performanță prin compresie, iar ca singur dezavantaj este faptul că nu poate fi folosit în afara companiei. A fost implementată cu sistemul de fișiere Google.

Avantajele folosirii sistemelor NoSQL constau în portabilitate, iar serverele acestora utilizează modelul DHT (Distributed Hash Table), astfel încât manipularea datelor obiect se realizează prin furnizarea de către utilizator a cheii primare a obiectului. DHT (Distributed Hash Table) fiind accesibil prin intermediul API Hashtable. Funcția hash este un algoritm matematic ce stă la baza arhitecturii NoSQL. Acest algoritm are la intrare o lungime variabilă și produce o lungime fixă la ieșire. Fiecare cheie este partiționată, iar valoarea hash rezultantă este folosită pentru a se trimite la un server de baze de date NoSQL. Înregistrarea este memorată pentru căutări ulterioare. NoSQL permite accesul la date doar într-un model îngust de acces predefinit. Astfel modelul de acces la date este unul fiabil. Printre sistemele NoSQL din ziua de azi pot fi menționate cele open-source, precum Hbase, MongoDB, CouchDB, GTM.[6]

Bazele de date NoSQL au atât modele mai simple decât cele SQL, dar și modele mult mai complicate decât acestea. Ceea ce se dorește însă este obținerea unei flexibilități mult mai mari. Procesarea se face online, la o performanță înaltă și este scalabilă orizontal. În cazul în care este necesară păstrarea mai multor copii ale datelor sesiune de către un utilizator și nu este necesară supunerea tuturor controalelor de consistență asupra acestora, specifice bazelor de date relaționale, atunci este eficientă doar păstrarea acestora în memorie prin sisteme NoSQL. [4]

Sistemele de baze de date NoSQL permit și un transfer mare de date, în comparație cu RDBMS-urile. Ca de exemplu, Google este capabil să proceseze date la nivelul zecilor petabyte într-o zi prin abordarea MapReduce.

Sistemul de baze de date NoSQL este proiectat astfel încât să nu fie dependent de componentele de hardware. Orice mașină poate fi îndepărtată sau adăugată fără eforturi operaționale semnificative. O caracteristică importantă a bazelor de date NoSQL este scalarea orizontală, ce presupune construirea pe loc și nu vertical. Scalarea orizontală este utilizată atunci când există capacitatea de a rula mai multe instanțe pe servere simultan. Este utilizată în momentul în care necesitățile aplicațiilor constau în trafic foarte mare și conectarea la un număr mare de evenimente. Soluțiile NoSQL pot fi ușor de scalat orizontal. [7]

Bazele de date NoSQL permit ca PC-urile să fie avantajos extinse, fără complexitatea și costul “sharding”, care ar implica partiționarea unor baze de date în mai multe tabele pentru a rula pe grupuri mai mari sau grile.

Unul din avantajele substanțiale aduse de sistemul de management NoSQL este procesarea distribuită a datelor, ce se realizează și prin intermediul MapReduce. În NoSQL se lucrează în medii distribuite, concentrându-se pe dinamica mecanismelor de interogare. Deși nu este un substitut al lipsei interogărilor și al capacităților de întreținere, procesarea distribuită a datelor este un ghid către NoSQL.

Modelele de interogare în NoSQL se bazează pe căutarea unor chei primare sau a unui câmp ID și pe lipsa unei interogări pe alte domenii. Bazele de date MongoDB și CouchDB permit interogări mai avansate, cum ar fi cele static predefinite pe nodurile bazei de date. Proiectarea bazelor de date NoSQL a fost făcută cu caracteristici dinamice de interogare în favoarea obținerii performanței și a scalabilității. [4]

Bazele de date NoSQL Companion reprezintă o abordare în care atributele căutate sunt copiate într-un SQL sau text. Capacitățile de interogare ale acestei baze de date sunt utilizate pentru a prelua cheile primare de sortare în seturi de date, prin care baza de date NoSQL va fi mai târziu accesată. A se vedea Fig. 2:

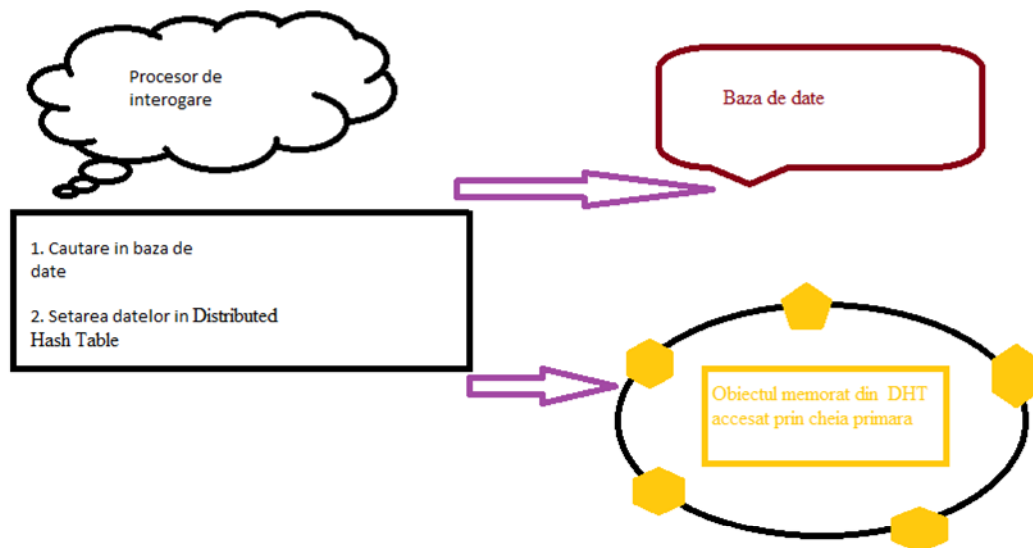


Figura 2

Mai există și alte modalități de căutare în NoSQL, precum cele Scatter / Gather, unde se permite interogarea și indexarea în cadrul unui server de baze de date. În acest caz procesorul de interogare poate trimite interogări către nodurile bazei de date, unde interogările sunt executate local. Rezultatele de la bazele de date ale tuturor serverelor sunt trimise înapoi la procesorul de interogare. A se vedea Fig.3:

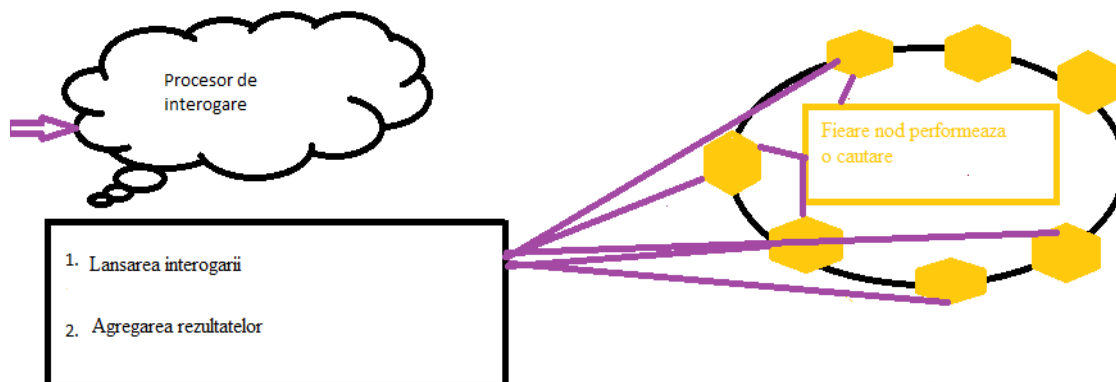


Figura 3

Datorită avantajelor menționate, sistemele NoSQL sunt în ascensiune și necesare proiectelor web complexe. Pentru a fi adoptate la o scară cât mai largă, acestea trebuie să fie:

1. open-source
2. facil de utilizat
3. suficiente și consistente

NoSQL-ul completează bazele de date relaționale, se caracterizează prin lipsa unei scheme fixe și aduce în plus scalarea ca soluție eficientă pentru seturile de date de dimensiuni mari.

### 3. Dezavantajele folosirii managementului bazelor de date NoSQL

Este știut faptul că în ansamblul sistemelor, modele alese joacă un rol vital. Datorită faptului că în sistemele NoSQL nu există o autoritate calificată pentru a defini un singur, bine-definit model, folosirea unui model non-normalizat poate conduce la suprapunerea obiectelor de date. La nivel de sistem este de asemenea important să se țină cont de modelele de concurență și de limitele serviciilor de date alese.

Din punctul de vedere al arhitecturii bazelor de date NoSQL există probleme legate de interfețe și interoperabilitate. Modelul DHT (Distributed Hash Table) conține interfețele sale încă standardizate, însă nu conține o semantică intrinsecă pentru indicii. Interoperabilitatea este importantă în momentul în care datele trebuie accesate de mai multe servicii, moment în care se poate pierde din performanță.

Spre deosebire de bazele de date relaționale care s-au consacrat ca fiind stabile, sistemele NoSQL apar pe piață ca o alternativă ce poate fi pusă în aplicare doar cu precauție, deoarece încă nu au ajuns la un nivel de maturitate cel puțin egal cu RDBMS-urile. De asemenea, în cazul apariției unor probleme, suportul sistemelor NoSQL este încă limitat, acestea fiind în mare proiecte open-source, iar companiile care oferă suport pentru bazele de date NoSQL nu oferă credibilitatea companiilor globale. [8]

Sistemele NoSQL se potrivesc foarte bine și în tehnologia Cloud, care se bazează pe virtualizare. Există și un punct slab al virtualizării ce ține de performanța I/O, limitările CPU-ului și ale memoriei fiind de altfel în strânsă legătură. Bazele de date NoSQL folosesc proporția cea mai mare pe memoria de disc, aceasta fiind locația principală de scriere, însă datorită scalării orizontale sunt capabile să memoreze datele eficient.

Bazele de date NoSQL prezintă un dezavantaj din punctul de vedere al administrării, pentru că necesită un anumit efort pentru a fi menținute și cunoștințe solide pentru instalare. Persoanele calificate pentru bazele de date NoSQL sunt mai puține decât cele cu experiență pentru RDBMS. De aceea preluarea lor de către firme trebuie să fie făcută în mod corespunzător, întrucât beneficiile reale aduse de acestea pot fi însoțite uneori și de anumite probleme.

#### **4. MongoDB**

Pentru a putea exemplifica practic bazele de date NoSQL am ales să analizăm baza de date NoSQL MongoDB. În opinia noastră această bază de date este ușor de folosit pentru utilizatorii de RDBMS-uri. MongoDB lucrează cu date nestructurate și organizează aceste date în format document. Implementarea acestei baze de date este mai ușoară decât un RDBMS deoarece ea urmărește modelul cheie valoare și nu are nevoie de o schemă predefinită a datelor. Conceptele acesteia pornesc de la concepte tradiționale, de aceea înțelegerea filosofiei acestei baze de date este ceva ușor de realizat. Prezentul articol urmărește atât prezentarea generală a bazei de date cât și instalarea și utilizarea ei.

MongoDB este o bază de date open-source NoSQL scrisă în C++. Aceasta poate conține mai multe baze de date, colecții și indecși. În unele cazuri (baze de date și colecții) aceste obiecte pot fi create implicit. Odată create, ele se găsesc în catalogul sistemului db.systems.collection, db.system.indexes. Colecțiile conțin documente (BSON). Aceste documente conțin la rândul lor mai multe câmpuri. În MongoDB nu există câmpuri predefinite spre deosebire de bazele de date relaționale, unde există coloanele care sunt definite în momentul în care tabelele sunt create. Nu există schemă pentru câmpurile dintr-un document, acestea precum și tipurile lor pot varia. Astfel nu există operația de „alter table” pentru adăugare de coloane. În practică este obișnuit ca o colecție să aibă o structură omogenă, deși nu este o cerință, colecțiile putând avea structuri diferite. Această flexibilitate presupune ușurință în migrarea și modificarea imaginii de ansamblu asupra datelor.

Pentru a înțelege această bază de date NoSQL o să facem o scurtă paralelă cu MySQL:

Termeni MySQL	Concepte MongoDB
Baza de date	Baza de date
Tabelă	Colecție
Index	Index
Rând	Document BSON
Coloană	Câmp BSON
Join	Încapsulare și legătură
Cheie primară	Câmpul cheie unică <code>_id</code>
Group by	Agregare
ServerDB = mysqld	ServerDB = mongod
Client = mysql	Client = mongo

Figura 4. MySQL vs NoSQL

Caracteristicile MongoDB:

- Stocarea datelor sub formă de documente - Baza de date MongoDB stochează obiecte (documente). Aceste documente reduc nevoia de join;
- Prezintă support pentru indexare – Indexarea pe fiecare din attribute se face în modul tradițional (RDBMS) asupra cheilor de regăsire ale documentelor;
- Disponibilitate - Disponibilitatea datelor este asigurată printr-un proces automat de failover;
- Auto-Sharding - Shardingul sau partiționarea datelor pe orizontală se face automat. Citirile și scrierile sunt distribuite pe partiții. Lipsa joinurilor face ca interogările distribuite să fie rapide;
- Limbajul de interogare - Limbajul de interogare este îmbunătățit și păstrează principii Sql și C++;
- Modificări rapide - MongoDB suportă operații de actualizare atomice cât și pe cele tradiționale. Operațiile de mai jos demonstrează flexibilitatea limbajului NoSQL:

```
$inc - { $inc : { camp : valoare } } – incementeaza campul "camp" cu numarul dat ca valoare
$set - { $set : { camp : valoare } } – campul „camp” ia valoarea data
$push - { $push : { camp : valoare } } – se adauga valoarea data campului specificat
$saddToSet - { $addToSet : { camp : valoare } } – adaugarea valorii intr-un vector
$pop - { $pop : { camp : 1 } } – sterge ultimul element dintr-un vector
$pop - { $pop : { camp : -1 } } – sterge primul element dintr-un vector
$pull - { $pull : { camp : _valoare } } – sterge toate valorile egale cu "valoare" din campul "camp"
```

- Map Reduce – MongoDB folosește map/reduce pentru agregare și procesare batch;
- GridFS – Poate stoca fișiere de dimensiuni mari fără a complica stiva de date;
- Suport – Există o cantitate mare de informații online – tutoriale, cărți forumuri ce servesc pentru a asimila rapid lucrul cu MongoDB. [9]

Instalarea MongoDB este extrem de simplă. Această bază de date poate fi instalată pe sisteme de operare variate, incluzând Windows, Linux, Ubuntu, Debian și OS X. Mediul de dezvoltare în cazul acestui articol a fost reprezentat de MongoDB în Windows 7 Professional 32 bit. Sistemele 32 bit sunt folosite de obicei pentru testare, deoarece instalările MongoDB pe un astfel de sistem permit o capacitate maximă de stocare de 2G. De menționat este faptul că nu se permite instalarea pe sistemele Windows XP. De asemenea distribuția Mongo include fișierul bin/mongo ce reprezintă un shell interactiv scris în JavaScript și folosit în aceeași manieră în care este folosit SQL Plus pentru baze de date Oracle pentru comenzi în linie de comandă. Shellul este folosit pentru verificări teste și pentru funcțiile administrative.

### Modelul de date Mongo

Un sistem MongoDB conține un set de baze de date. O bază de date conține un set de colecții. O colecție conține un set de documente. Un document conține un set de câmpuri. Un câmp conține o pereche cheie-valoare. Cheia este un nume (string). O valoare poate fi orice tip de dată: string, integer, float, un document sau un vector de documente etc.

Pentru a demonstra funcționalitățile MongoDB vom construi o aplicație utilizând shellul MongoDB. Shell-ul interactiv este inclus în distribuția MongoDB. Pentru a începe lucrul cu acest shell trebuie apelat fișierul ./bin/mongo din locația unde a fost instalată baza de date. Dacă se rulează fără nici un parametru, atunci conectarea se va face la o bază de date numită test ce rulează pe mașina locală.

Când shell-ul este lansat în execuție, se verifică dacă în directorul home există fișierul .mongorc.js. Dacă acesta este găsit, atunci conținutul lui este interpretat și rulat înainte de a afișa promptul. Acest fișier permite modificarea unor variabile. Instrucțiunile folosite pentru utilizarea shellului sunt:

```
help //ajutor pentru setările bazei de date
db.help() // afișează metodele disponibile la nivelul bazei de date
db.mycollection.help() //metode la nivelul colecțiilor
```

Pentru a putea schimba baza de date în care se lucrează există următoarele instrucțiuni:

```
show dbs // afișează bazele de date existente pe server
use db_name // schimbă baza de date în lucru
show collections // afișează lista colecțiilor din baza de date curentă
```

Aplicația prezentată în acest articol va presupune înregistrarea comentariilor angajaților unei multinaționale ce lucrează în diferite locații referitoare la cum consideră aceștia mediul de muncă în locația specifică.

Pas 1. Utilizatorii se logează într-o anumită locație. Vom crea baza de date cu locații.

```
> locatii = { denumire : "Bucuresti", adresa : "Calea Dorobanti", codpostal : "11123"}
{
  "denumire" : "Bucuresti",
  "adresa" : "Calea Dorobanti",
  "codpostal" : "11123"
```

Figura 5. Creare locații



Adaugăm comentariile și alte date.

```
> locatie1 = { denumire : "Bucuresti", adresa : "Calea Dorobanti", codpostal : "11123", comentarii : ["mediu placut","seriozitate"], coordonate1L : [30.9,78.3], detaliicomentarii : [ { angajat : "Ionescu Alin", detaliu : "mediu de lucru placut"}, {angajat: "Dobrita Ion", detaliu : "mediu de lucru serios"}]
{
  "denumire" : "Bucuresti",
  "adresa" : "Calea Dorobanti",
  "codpostal" : "11123",
  "comentarii" : [
    "mediu placut",
    "seriozitate"
  ],
  "coordonate1L" : [
    30.9,
    78.3
  ],
  "detaliicomentarii" : [
    {
      "angajat" : "Ionescu Alin",
      "detaliu" : "mediu de lucru placut"
    },
    {
      "angajat" : "Dobrita Ion",
      "detaliu" : "mediu de lucru serios"
    }
  ]
}
>
```

Figura 6. Adaugare detalii

Instrucțiunile de inserare și regăsire sunt:

```
Db.db.insert(locatie1);
```

```
Db.db.find();
```

Ulterior putem adăuga detalii în campul de detalii astfel:

```
db.baza_mongo.update( {denumire : "Bucuresti"}, { $push : { "detaliicomentarii"
{ "angajat" : "Cristescu Marian" , "detaliu" : "mediu de lucru placut" } } });
```

Acesta este o scurtă aplicație pentru a demonstra ușurința și flexibilitatea pe care o are orice utilizator în folosirea bazelor de date NoSQL. Acestea sunt optime pentru a manipula cantități mari de date, de preferat în cazul în care natura datelor de manipulat nu se încadrează în standardele modelului relațional. Ele se caracterizează prin abilitatea de stocare și regăsire într-o mare cantitate de date, fără a avea relații între elemente.

## 5. Concluzii

Potrivit definiției formulate în acest articol bazele de date NoSQL reprezintă o trecere către baze de date superioare ce vor integra flexibilitatea și performanțele lor actuale cu modelul relațional. 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.

Baza de date prezentată în acest articol, MongoDB este o bază de date ușor de înțeles și de manipulat. Aceasta este ideală atât pentru proiecte mici, de test, cât și pentru proiecte ce implică un volum mare de date. În opinia noastră MongoDB este o bază de date ce va fi folosită din ce în ce mai mult pe viitor datorită tendinței actuale înclinată către aplicații Web.

În acest articol am realizat o prezentare a generației noi de baze de date NoSQL, pe care le considerăm un pas important în evoluția bazelor de date, accentuând importanța acestora.

## BIBLIOGRAFIE

1. **SEEGER, M.:** Key-Value stores: a practical overview, Computer Science and Media Ultra-Large-Sites SS09 Stuttgart, Germany, 21 Septembrie 2009.
2. <http://nosql-database.org/>
3. **WU, SUZANNE:** How Much Information IS There in the World. University of Southern California, 10 Februarie 2011.
4. **CATTELL, R.:** Scalable SQL and NoSQL Data Stores, SIGMON Record, Decembrie 2010, Vol 39, Nr 4.
5. **ASLETT, M.:** “MySQL vs. NoSQL and NewSQL- survey results”, 22 May 2012, 451 Research Group.
6. **SHALOM, N.:** The Common Principles Behind The NoSQL Alternatives, December 2009,– Blog post of 2009-12-15. [http://natishalom.typepad.com/nati\\_shaloms\\_blog/2009/12/the-common-principlesbehind-the-nosql-alternatives.html](http://natishalom.typepad.com/nati_shaloms_blog/2009/12/the-common-principlesbehind-the-nosql-alternatives.html).
7. **WELSH, M., CULLER, D.; BREWER, E.:** SEDA: an architecture for well conditioned, scalable internet services. In: Proceedings of the eighteenth ACM Symposium on Operating Systems Principles. New York, NY, USA : ACM, 2001 (SOSP '01), p. 230–243.
8. **STONEBRAKER, M.:** The “NoSQL” Discussion has Nothing to Do With SQL. November 2009. – Blog post of 2009-11-04. <http://cacm.acm.org/blogs/blog-cacm/50678-the-nosql-discussion-has-nothing-to-do-withsql/fulltext>
9. <http://docs.mongodb.org/>