Baze de date

Universitatea “Transilvania” din Brasov

Lect.dr. Costel Aldea

costel.aldea@gmail.com

BigData (1) - volume, variety, velocity

1) Characteristics of Data in Big Data Environments
2) Dataset Types in Big Data Environments
3) Machine Learning Types
4) Fundamental Analysis and Analytics
5) Business Intelligence & Big Data
6) Data Visualization & Big Data
7) Big Data Analysis Lifecycle (from business case evaluation to data analysis and visualization)
8) A/B Testing, Correlation
9) Regression, Heat Maps

BigData (2)

10) Time Series Analysis
11) Network Analysis
12) Spatial Data Analysis
13) Classification, Clustering
14) Outlier Detection
15) Filtering (including collaborative filtering & content-based filtering)
16) Natural Language Processing
17) Sentiment Analysis, Text Analytics
18) File Systems & Distributed File Systems, NoSQL
19) Processing Workloads, Clusters

BigData (3)

20) Cloud Computing & Big Data

21) Big Data Storage Terminologies (including sharding, replication, CAP theorem, ACID, BASE)

22) Big Data Storage Requirements

23) On-Disk Storage (including distributed file system – databases)

24) Introduction to NoSQL – NewSQL

25) NoSQL Rationale – Characteristics

26) NoSQL Database Types (including key-value, document, column-family and graph databases)

27) Big Data Processing Requirements

28) Big Data Processing (including batch mode and realtime mode)

29) MapReduce Explained (including map, combine, partition, shuffle and sort, and reduce)

BigData (4)

30) Big Data pipelines, its stages and the design process involved in developing Big Data processing solutions
31) Bulk Synchronous Parallel (BSP) processing engine
32) BSP vs. MapReduce
33) Big Data with Extract-Load-Transform (ELT)
34) Big Data Solutions (including Characteristics, Design Considerations & Design Process)
35) In-Memory Storage Devices, In-Memory Data Grids & In-Memory Databases
36) Event Processing (CEP)
37) Read-Through, Read-Ahead, Write-Through & Write-Behind Integration Approaches
38) Polyglot Persistence (including Explanation, Issues & Recommendations)

NoSQL

- Not only SQL

- DB-Engines Ranking

  http://db-engines.com/en/ranking

Prezentare generala

- O baza de date NoSQL (initial referindu-se la “non SQL” sau “non relationala”) ofera un mecanism de stocare si recuperare a datelor care a fost construit cu alte scopuri decat relatiile tabelare folosite in bazele de date relationale.
- Astfel de baze de date au existat inca din anii 1960, dar acestea nu au obtinut “porecla” NoSQL pana in secolul 21, aceasta fiind declansata de nevoile Web 2.0 ale unor companii precum Facebook, Google sau Amazon.

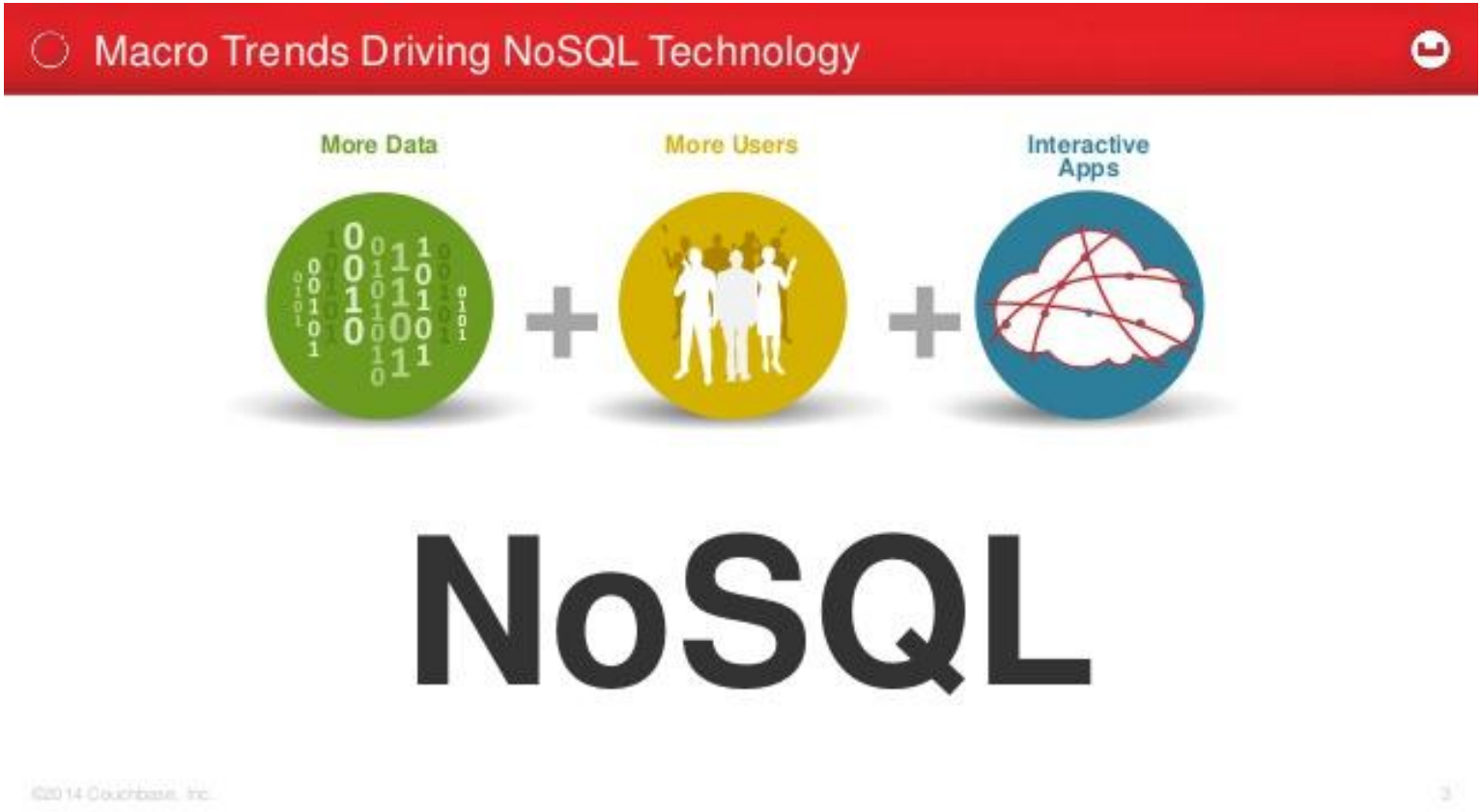
Prezentare generala

- Motivatiile pentru aceasta abordare include simplitatea a design-ului, o scalare orizontala mai simpla (care era o problema la bazele de date relationale) si un control mai bun.
- Structurile de date folosite de bazele de date NoSQL (ex. Valori cheie, coloana mare, grafic sau document) sunt specificate de cele folosite ca prestabilite in bazele de date relationale, facand unele operatii mai rapide in NoSQL.

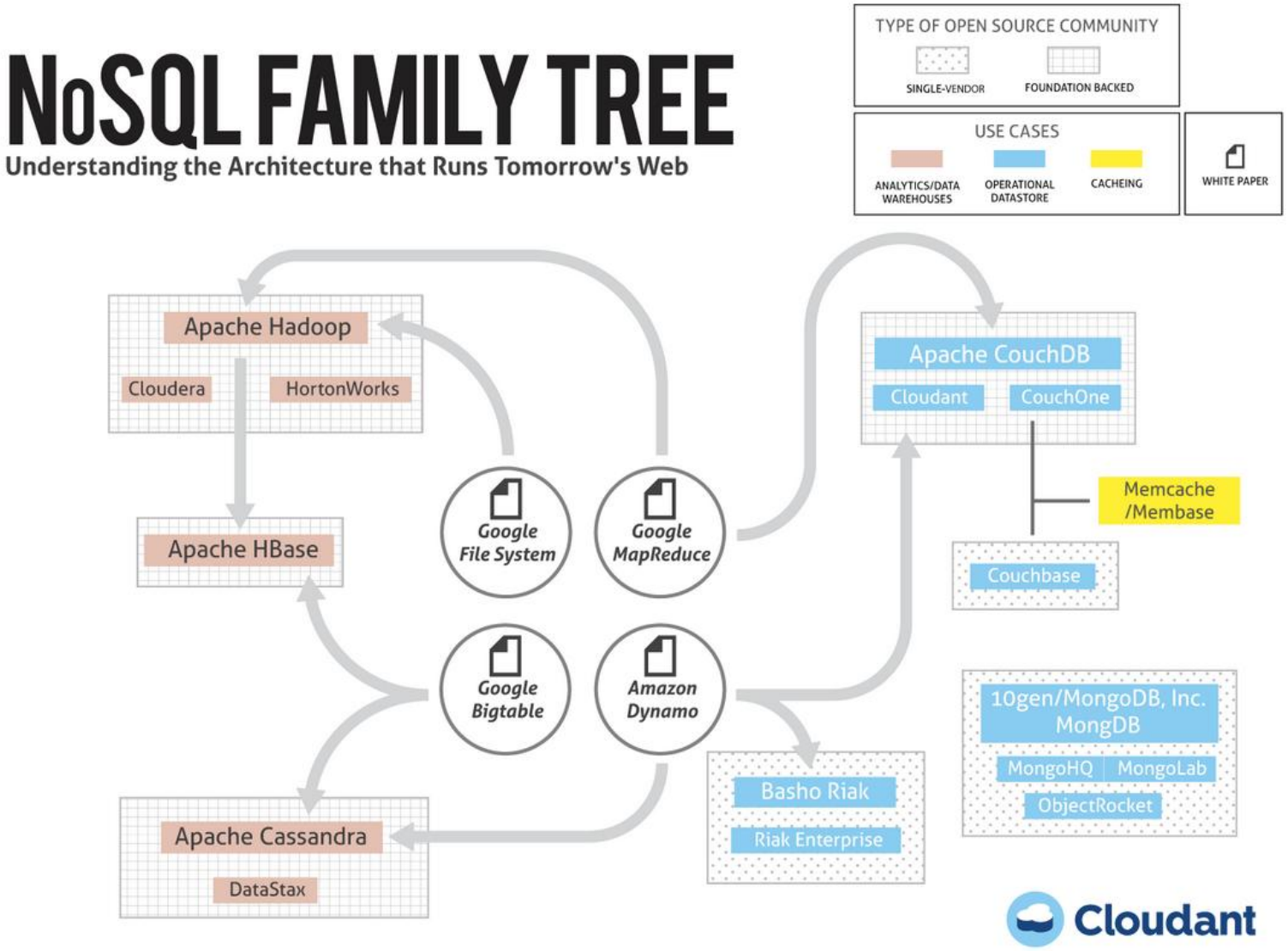
Prezentara generala

- Adecvarea particulara a unui NoSQL dat depinde de problema pe care trebuie sa o rezolve.
- Uneori structurile de date folosite de bazele de date NoSQL sunt vazute ca “mai flexibile” decat tabelele bazelor de date relationale.

Tehnologia NoSQL



Evolutia NoSQL



Utilitate

- Bazele de date NoSQL sunt tot mai mult folosite in aplicatii de intreprindere mari sau in aplicatii de tipul real-time web.
- Sistemele NoSQL sunt adesea denumite si “Not Only SQL” pentru a se arata ca ele pot utiliza si limbaje SQL.
- Multe mecanisme de stocare NoSQL au compromis consistenta in favoarea disponibilitatii, portionand toleranta si viteza.

Utilitate

- Barierele marii “adoptari” a NoSQL includ si folosirea unui limbaj de query mai redus, fata de SQL, unde lipsea abilitatea de a face JOIN-uri ad-hoc intre tabele, lipsa de interfete standardizate si investitiile mare din trecut facute in actualele baze de date relationale.
- Sistemele NoSQL ofera concepte precum logare “write-ahead” pentru a evita pierderea de date.

N(ot) O(nly) SQL

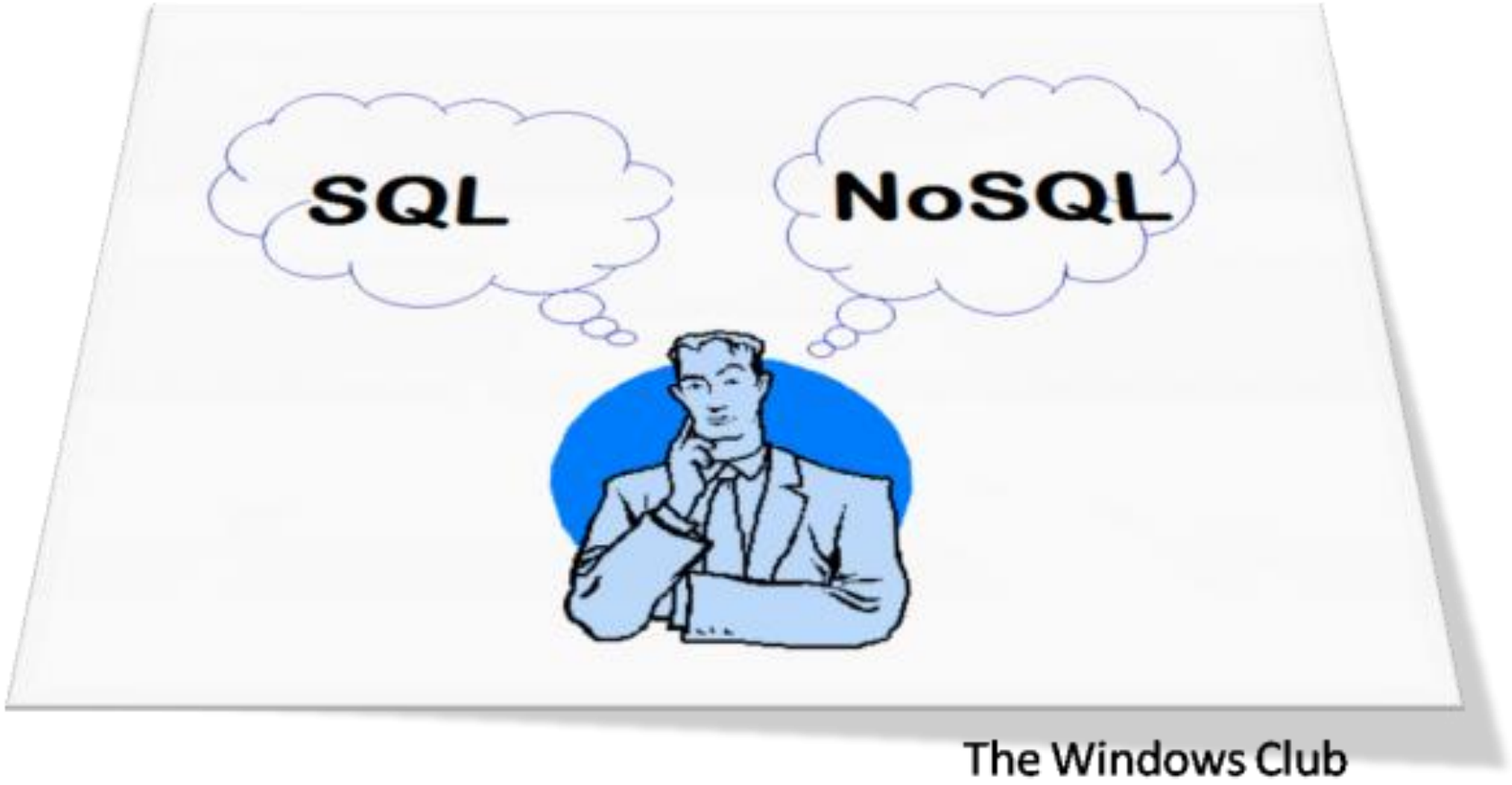
De-a lungul timpului

- Carlo Strozzi
  - Termenul de NoSQL a fost folosit prima data de Carlo Strozzi in 1998 pentru a isi numi varianta usoara: “Strozzi NoSQL open-source relational database”.
  - Acesta a sugerat ca acest nou sistem sa se numeasca “NoREL”, deoarece porneste de la modelul relational.

De-a lungul timpului

- Johan Oskarsson
  - Acesta a introdus termenul NoSQL in 2009, in site-ul Last.fm unde a organizat un eveniment pentru a discuta despre “bazele de data distribuite, non relationale” care sunt open-source.
  - Majoritatea sistemele NoSQL de inceput nu au oferit consistenta, izolare si durabilitate.

Vechi sau nou?



Tipuri de exemple NoSQL

- Au fost multe incercari de a clasifica bazele de date NoSQL, fiecare cu diferite categorii si subcategorii, unele chiar suprapunandu-se.
- O clasificare de baza ar fi:
  - Column: Accumulo, Cassandra, Druid, Hbase, Vertica
  - Document: Apache CouchDB, Clusterpoint, Couchbase, DocumentDB, HyperDex
  - Key-value: Aerospike, Dynamo, FoundationDB
  - Graph: Allegro, InfiniteGraph, MarkLogic, Neo4J
  - Multi-model: Alchemy Database, AragoDB, CortexDB

O clasificare detaliata

Type	Examples of this type
Key-Value Cache	Coherence, eXtreme Scale, GigaSpaces, GemFire, Hazelcast, Infinispan, JBoss Cache, Memcached, Repcached, Terracotta, Velocity
Key-Value Store	Flare, Keyspace, RAMCloud, SchemaFree, Hyperdex, Aerospike
Key-Value Store (Eventually-Consistent)	DovetailDB, Oracle NoSQL Database, Dynamo, Riak, Dynomite, MotionDb, Voldemort, SubRecord
Key-Value Store (Ordered)	Actord, FoundationDB, Lightcloud, LMDB, Luxio, MemcacheDB, NMDB, Scalaris, TokyoTyrant
Data-Structures Server	Redis
Tuple Store	Apache River, Coord, GigaSpaces
Object Database	DB4O, Objectivity/DB, Perst, Shoal, ZopeDB
Document Store	Clusterpoint, Couchbase, CouchDB, DocumentDB, Lotus Notes, MarkLogic, MongoDB, Qizx, RethinkDB, XML-databases
Wide Column Store	BigTable, Cassandra, Druid, HBase, Hypertable, KAI, KDI, OpenNeptune, Qbase

Caracteristici generale

- memorarea unor volume mari de date.
- nu există o structură fixă a datelor.
- între date se pot stabili legături (prin referinţe la date memorate în alte baze de date) .
- aceleaşi date pot să fie memorate pe mai multe servere (partajare şi replicare) .
- la interogare nu se folosesc operaţii de join (mari consumatoare de timp) .
- sunt soluţii foarte bune pentru cazuri particulare.

Care sunt situatiile in care se recomanda folosirea unei solutii NoSQL?

- baza de date tradițională nu mai poate fi scalată la un preț acceptabil;
- baza de date a fost deja denormalizată pentru a îmbunătăți performanțele;
- stocați cantități foarte mari de text și/sau imagini;
- generați foarte multe informații temporare cum ar fi: coșuri de cumpărături, chestionare incomplete, istorice de navigare, personalizări, etc;

- aveți nevoie să rulați interogări de date care nu implică doar simple relații ierarhice. de exemplu: "toți oamenii dintr-o rețea socială care nu au cumpărat anul acesta o carte dar au legătură cu o persoană care a cumpărat";
- tranzacțiile nu trebuie să fie perfect consistente; de exemplu un buton de "like", dacă tranzacția eșuează nu este nici o problemă, utilizatorul cel mai probabil va mai apăsa o dată butonul.

Dezavantaje

- nu există standarde (cum există standardul SQL la bazele de date relaţionale) .
- nu se asigură consistenţa bazei de date (de către sistemul de gestiune).
- nu există metode performante pentru protecţia datelor.
- modelele propuse sunt la primele versiuni.
- există posibilităţi limitate de interogare.
- aproape toate sistemele apărute sunt open-source.
- există relativ puţini dezvoltatori software pentru NoSQL.

Clasificarea modelelor de memorare - NoSQL

1. Colecţii de perechi cheie-valoare: Amazon Dynamo, Redis, Membase, MemcacheDB, Scalaris, Tokyo Cabinet, Voldemort, Riak.
2. BigTable (column database): Google Bigtable, Cassandra (Facebook), Hadoop/HBase, HyperTable, Amazon SimpleDB.
3. Graf: Neo4j, InfiniteGraph, InfoGrid, GraghBase, HyperGraphDB .
4. Colecţii de documente: MongoDB, Couchbase, CouchDB, Terrastore, RavenDB .

1. Colecţii de perechi cheie-valoare

- Baza de date este formată dintr-o colecţie de perechi (cheie, valoare), asemănătoare colecţiilor (tabelelor) asociative din unele limbaje de programare.
- Cheia şi valoarea sunt şiruri de caractere, iar cheile sunt distincte (se folosesc pentru identificare).
- Memorarea bazei de date se poate face: prin utilizarea unui "hash table" sau sub forma unei variante de B-arbore.

Operatii NoSQL

- Operaţiile permise în această bază de date sunt:
  - adăugarea unei perechi la colecţie
  - eliminarea unei perechi din colecţie
  - modificarea valorii dintr-o pereche existentă
  - consultarea valorii pentru o cheie dată.

Valoarea este un şir de caractere sau o dată binară şi nu este interpretată de sistem (este analizată de client)

2. Modelul BigTable

- este un mod de memorare propus şi folosit de Google.
- Se recomandă pentru tabele de dimensiune mare, cu multe elemente nedefinite.
- In modelul BigTable se memorează consecutiv valorile nenule din fiecare coloană) => mod de memorare orientat coloană.

3. Modelul graf

- Un graf orientat se poate construi cu acest triplet:

  (identificator_entitate, nume_atribut, valoare_atribut) .

- Vârfurile grafului pot să fie de două tipuri: identificatori de entităţi (rerpezentate ca o elipsă), sau constante (rerpezentate sub forma de dreptunghi).
- Valoarea unui atribut poate să fie un identificator de entitate, şi atunci este de primul tip.

4. Colecţii de documente

- o bază de date conţine diverse colecţii de documente (obiecte), analog tabelelor dintr-o bază de date relaţionala.
- Intr-o colecţie se grupează documentele utile într-o interogare.
- Intre colecţii diferite nu se pot efectua operaţii de join.
- baza de date şi colecţiile din baza de date se identifică printr-un nume.
- un document (obiect) nu are o structură stabilită
- un document are un identificator unic in colecţia de date, prin campul cu denumirea "_id".

Exemplu document

- caracterele (din denumiri, din valori) sunt case-sensitive .
- pentru gestiunea unei baze de date şi a unei colecţii există mai multe metode.
- Exemplu de document dintr-o bază de date de acest tip:

```
{
    "nume": "Pop",
     "prenume": "Ion",
    "contract_studiu": ["MIC0002","MIH0002", "MMP0003",
    "MID0004"],
    "adresa": { "localitatea": "Cluj-Napoca",
                    "strada": "Kogalniceanu", "numarul": 1 }
    "email": “abc@info.unitbv.ro"
}
```

Cassandra

- este un sistem de stocare cheie-valoare structurat, scalabil, consistent şi distribuit.

- este consistentă deoarece sistemul de stocare garantează că dacă se execută update-uri asupra unui obiect toate accesele vor întoarce valoarea ultimului update.

Cassandra

- Modelul de date – poate fi descris ca nişte hash-map-uri imbricate.
- Hash-map-urile stochează datele printr-o cheie unică folosită pentru a regăsi datele.
- perechile cheie-valoare nu sunt stocate ca două valori individuale ci cuplate într-o clasă numită column.
- îşi structurează modelul de date în spaţii de chei, familii de coloane, coloane şi supercoloane.
- Un spaţiu de chei este un nume care grupează familiile de coloane şi poate fi comparată cu shema unei singure baze de date în perspectiva SQL

Bibliografie

- https://www.books-express.ro/blog/nosql-avantaje-si-dezavantaje/
- http://www.rusu.coneural.org/teaching/MLR5027/2014.BD.Curs.14.pdf
- http://documents.tips/documents/referat-bd.html
- https://en.wikipedia.org/wiki/NoSQL
- http://db-engines.com/en/ranking