Big Data Management and NoSQL Databases

Simona-Vasilica Oprea
The Bucharest University of Economic Studies, Romania
simona.oprea@csie.ase.ro

Adela Bára
Academy of Romanian Scientists, Romania
bara.adela@ie.ase.ro

Niculae Oprea
“Politehnica” University of Bucharest, Romania
niculae.oprea@transelectrica.ro

Abstract

In this paper, we aim to showcase the general context, characteristics of NoSQL, data management using NoSQL database and an exemplification in MongoDB. Moreover, organization and management of large volumes of data (Big Data), the trend in the field of advanced database systems and business intelligence will be depicted in this research. The large volumes of data in various fields of activity require adequate management in order to gain a competitive advantage or to achieve the quality of decisions. Various fields generate large volumes of data, such as: consumerism and home devices, smart infrastructure, security and surveillance, healthcare system, transportation, retail and industrial sectors, others.

Key words: database, NoSQL, big data, relational DB
J.E.L. classification: C55, C80, C82

1. Introduction

The evolution of data management spans from files (in the 60s) to databases. In the 80s, relational databases emerged, in the 90s, object-oriented databases also emerged (Rashid and Sawyer, 2005). After the 2010s, a new category of databases, namely non-relational databases or NoSQL (Not only SQL) databases have taken the floor (Chen and Lee, 2019). Autonomous databases also emerged (Curino et al., 2013). They all coexist and have interesting particularities that make them suitable for certain applications (Schuler and Kessleman, 2019).

This evolution convinced companies to try one of the latest databases (Bjeladinovic, Marjanovic and Babarogic, 2020), (Bjeladinovic, 2018), but there were cases in which the decision makers went back to relational databases (https://www.theguardian.com/info/2018/nov/30/bye-bye-mongo-hello-postgres).

This paper is structured in several sections. In the next section, a brief literature review is provided. Research methodology regarding data management is presented in section 3, whereas in section 4, the main findings in MongoDB and a comparison of SQL and NoSQL are presented. The conclusion is drawn in section 5.

2. Literature review

Several research papers approached the database evolution in time (Schuler and Kessleman, 2021), (Domínguez et al., 2008), (Cleve et al., 2015), (Hillenbrand et al., 2022). Recent trends in database technology were investigated in (Lieponienė, 2021). Various social and research networks, internet of things (IoT), e-commerce require reliable data management. The challenges of these networks are relate to the volume of data they generate, the speed and variety of data that has to be
stored and processed to extract useful information and obtain competitive advantages. (Lieponienė, 2021) compared relations and non-relational DB, considering alternatives such as multi-model DB (Lu and Holubová, 2019) and cloud DB. Big data management in the cloud was analyzed in (Hameurlain and Morvan, 2016). The use of Big Data Analytics in healthcare is investigated in (Batko and Ślęzak, 2022). Its use spread to numerous fields, such as: manufacturing (Li, Chen and Shang, 2022), government (Long et al., 2021), energy (Zhou, Fu and Yang, 2016), etc. The applicability of multi-model databases for accessible indoor navigation was also underlined in (Simon-Nagy, Fleiner and Varkonyi-Koczy, 2020).

3. Research methodology - Data management

A relational database (DB) represents the traditional method of organizing data into homogeneous collections of structured data. They are organized according to an abstract model that describes how data is represented and accessed, using mathematical formalizations. The main characteristic elements of the data model are as follows: Data structure, Operators, Integrity Constraints. One example of schema is depicted in Figure 1.

*Figure no. 1. Example of relational schema*

![Example of relational schema](image)

*Source: Authors’ contribution*

The advantages and disadvantages of relational DB are depicted in Table 1.

*Table no. 1 Advantages and disadvantages of relational DB*

<table>
<thead>
<tr>
<th>The advantages of relational DB are:</th>
<th>The disadvantages of relational DB are:</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Unitary, coherent and structured organization of data according to a previously defined normalized scheme (on-write scheme);</td>
<td>✓ Loss of some useful semantic information through normalization;</td>
</tr>
<tr>
<td>✓ The data protection ensured by the SGBD aims at: data security and integrity;</td>
<td>✓ Relational operations, even optimized ones, are expensive in terms of computing resources;</td>
</tr>
<tr>
<td>✓ Data shareability;</td>
<td>✓ Relational languages are limited.</td>
</tr>
<tr>
<td>✓ Competitive access;</td>
<td></td>
</tr>
<tr>
<td>✓ Data independence from applications;</td>
<td></td>
</tr>
<tr>
<td>✓ Optimizing access to data;</td>
<td></td>
</tr>
<tr>
<td>✓ Relational languages are standardized (SQL standard).</td>
<td></td>
</tr>
</tbody>
</table>

*Source: Authors’ contribution*
They implement ACID properties: Atomicity, Consistency, Isolation and Durability.

Big data aims to integrate, organize, process and store large volumes of data in a short time. It represents a suite of technologies used for the collection, processing, storage and analysis of data characterized by large volume, variety, velocity (speed), veracity, as described in Figure 2. The 10 Vs of Big Data (https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx) refers to more Vs that include visualization and value.

The management of large volumes of data includes data management-oriented technologies that allow the integration, processing and analysis of large volumes of data in a short time. Its purpose is to extract useful information from data and gain competitive advantage. The multiple data sources include applications and the corresponding data from log files, sensors and real-time data, web and text sources or data that reflects activity on social networks or forums, articles, etc. Some examples are included in Figure 3.

**Figure no. 2. Characteristics of big data**

![Figure 2. Characteristics of big data](Source: Authors' contribution)

**Figure no. 3. Examples of multiple big data sources**

![Figure 3. Examples of multiple big data sources](Source: Oracle)

Since the 80s, data was traditionally stored in relational DBs and data warehouses. However, relational DBs do not have the functionality to organize large volumes of data and processing requirements related to volume, variety, veracity and speed of data collection require non-relational DB or Not only SQL (NoSQL).

**NoSQL databases - advantages**

- They do not have a fixed scheme (structure) which gives flexibility to data collections (on-read scheme);
- They allow the storage and processing of semi-structured or unstructured data;
- The lack of the need to normalize data and store relations between tables brings increased performance to applications;
✓ Provides support for replication, distribution, API (Application Programming Interface);
✓ The ability to scale horizontally or replicate and distribute data on multiple servers;
✓ Allow dynamic modification of the structure of data collections.

NoSQL databases - disadvantages
✓ Implements a competitive model that is weaker than the relational model (ACID);
✓ Lack of standardization of the language;
✓ Reduced management facilities;
✓ Higher dynamic of versions;
✓ Partially implements the CAP theorem (Consistency, Availability, Partition tolerance).

NoSQL databases implements CAP Theorem that refers to:
✓ Consistency refers to the fact that all users of the system must access the same data, all the time;
✓ Availability is a guarantee that every request will be answered;
✓ Partition tolerance is the property of the system to function in case certain nodes in the system are no longer functional.

According to the CAP theorem, a DB cannot simultaneously satisfy all 3 constraints, but it can excel at any 2 of them: CA, CP or AP (graphically depicted in Figure 4).

Figure no. 4. CAP theorem and its combination of 2 constraints: CA, CP or AP

Source: Authors’ contribution

The three properties are depicted graphically in Figure 5.

Figure no. 5. CAP theorem: Consistency, Availability and Partitioning

<table>
<thead>
<tr>
<th>Consistency</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Consistency" /></td>
<td><img src="image2" alt="Availability" /></td>
</tr>
</tbody>
</table>

469
Therefore, there are databases that better implement CP, whereas others better implement AP. Two examples of databases are MongoDB that is better at Partitioning and Cassandra that is better at Availability (as in Figure 6).

![CP – MongoDB, AP - Cassandra](source)

Data management in NoSQL DB includes the following methods of data organization, by: Columns; Key-value; Document; Graph. On columns, the data is stored in columns (Suárez-Cabal et al., 2023), (Khourdifi, Bahaj and Elalami, 2018) as in Figure 7. The following databases are included in this category: HBase, Cassandra, Hypertable, Accumulo, Amazon Simple DB, Cloudata, Cloudera.

![Row-store vs. column store](source)

On key-value, values are identified by a unique key (Puangsaijai and Puntheeranurak, 2017). The data model is simple, scalable, easy to manipulate, without a predefined schema (El Alami, Bahaj and Khourdifi, 2018). Some examples include: Redis, Riak, Oracle NoSQL Database, Tarantool, Tokyo Cabinet, etc. Documents are semi-structured data and can be stored in different forms: JSON, BSON, XML, etc. Documents are stored in the DB by a unique key to identify that document (Blanco et al., 2022). Couchbase and MongoDB are the most popular systems that implement document storage. Performance evaluation of NoSQL document DB, such as Couchbase, CouchDB, and MongoDB is analyzed in (Carvalho, Sá and Bernardino, 2023).
Graphs use nodes, edges/arcs or relationships between them (links) and node properties (Angles and Gutierrez, 2008). Graph-based DBs do not use indexes to retrieve a particular value in the DB, but instead use relationships between nodes to retrieve adjacent elements. Examples of systems implementing graph-based storage: Allegro, Neo4J, OrientDB, Virtuoso. An efficient graph database model is provided in (Vyawahare, Karde and Thakare, 2019). Additionally, an example of application that reply on graph DB is provided by The Oracle of Bacon that provides interesting movie links (http://oracleofbacon.org/movie-links.php). A graph example is presented in Figure 8, while a summary of DB examples is provided in Table 2.

**Figure no. 8. Graph example**

![Graph example](image)

Source: Authors’ contribution

**Table no. 2 Summary of DB examples**

<table>
<thead>
<tr>
<th>No.</th>
<th>DB type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Columns</td>
<td>HBase, Cassandra</td>
</tr>
<tr>
<td>2</td>
<td>Key-value</td>
<td>Oracle NoSQL Database, Riak, Redis</td>
</tr>
<tr>
<td>3</td>
<td>Document</td>
<td>CouchDB, Couchbase, and MongoDB</td>
</tr>
<tr>
<td>4</td>
<td>Graph</td>
<td>Allegro, Neo4J, OrientDB, Virtuoso</td>
</tr>
<tr>
<td>5</td>
<td>Multi-model</td>
<td>ArangoDB</td>
</tr>
</tbody>
</table>

Source: Authors’ contribution

The DBs are ranked according to their popularity. At the writing of this paper, Oracle and MongoDB are on top for relational and non-relational DB (https://db-engines.com/en/ranking), followed by MySQL and Redis.

4. Findings

MongoDB Atlas is an environment that allows the creation of a non-relational DB (Figure 9 – Cluster0). Based on the connection string, a user can connect from an application (Figure 10).
Using the connection string, a user connects via NoSQL Booster for Mongo to Cluster0 (as in Figure 11).
In Figure 12, in NoSQL Booster for MongoDB, one node is primary and two are secondary. They ensure data replication in MongoDB.

Figure no. 12. Cluster nodes

MongoDB includes methods that correspond to SQL statements. For instance, the SQL SELECT statement can be compared with MongoDB find() method (as in Table 3).

Table no. 3. SQL SELECT compared with MongoDB find() method

<table>
<thead>
<tr>
<th>SQL SELECT</th>
<th>MongoDB find()</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT * FROM clienti_daune</td>
<td>db.clienti_daune.find()</td>
</tr>
<tr>
<td>SELECT _id, MARCA, MODEL, PRET_MANOPERA FROM clienti_daune</td>
<td>db.clienti_daune.find().projection([MARCA:1, MODEL:1, PRET_MANOPERA:1])</td>
</tr>
<tr>
<td>SELECT MARCA, MODEL, PRET_MANOPERA FROM clienti_daune</td>
<td>db.clienti_daune.find().projection([MARCA:1, MODEL:1, PRET_MANOPERA:1])._id:0 //exclude indexul _id</td>
</tr>
<tr>
<td>SELECT MARCA, MODEL, TARA_FURNIZOR FROM clienti_daune WHERE TARA_FURNIZOR = 'UK'</td>
<td>db.clienti_daune.find({ TARA_FURNIZOR: &quot;UK&quot; }).where(MARCA:1, TARA_FURNIZOR:1,_id:0)</td>
</tr>
<tr>
<td>SELECT MARCA, MODEL, TARA_FURNIZOR FROM clienti_daune WHERE TARA_FURNIZOR in ('UK', 'US') ORDER BY TARA_FURNIZOR DESC</td>
<td>db.clienti_daune.find({ TARA_FURNIZOR: { $in: [&quot;UK&quot;, &quot;US&quot;] } }).sort(TARA_FURNIZOR:-1)</td>
</tr>
</tbody>
</table>

Source: Authors’ contribution
Pipelines as in (Figure 13) can be set to filter and group by various variables the data set.

5. Conclusions

A brief evolution of data management and databases was emphasized in this paper along with a comparison between relational and non-relational DB. The NoSQL databases allow the flexible organization of large volumes of data using extensible data schemes (groups of columns, documents, graphs, key-value) compared to modeling data in fixed relational schemes.

They are designed for horizontal scaling through multi-node data distribution models.

It supports different interfaces for data access and advanced analysis as exemplified in MongoDB.

6. Acknowledgement

This paper was supported by Academy of Romanian Scientists, Ilfov 3, 050044 Bucharest, Romania, project title: “Soluții informatice pentru analiza impactului rețelelor de social media asupra instrumentelor investiționale cu grad ridicat de risc: cryptomonede și bursă”.

7. References