Baltic J. Modern Computing, Vol. 8 (2020), No. 4, 551-559
https://doi.org/10.22364/bjmc.2020.8.4.06

Recent Trends in Database Technology

Jurgita LIEPONIENĖ

Panevėžys University of Applied Sciences, Laisvės 23, LT-35200 Panevėžys, Lithuania

jurgita.lieponiene@panko.lt

Abstract. Rapid development of information technologies puts forward new requirements to data storage, processing and management. Social networks, web 2.0, web 3.0 technologies, internet of things, cloud computing require new data management technologies. As an alternative to relational databases NoSQL and multi-model database systems become more and more popular. This paper briefly discusses the past achievements of database research and development, compares relational and NoSQL databases, analyses multi-model and cloud-based databases.

Keywords: relational databases, NoSQL, multi-model database, cloud-based database

Introduction

Relational database management systems technology has been successfully applied to many application domains (Shalini and Dhamodharan, 2016). Relational databases have proved to be an effective solution for data management requirements in large and small organizations. This database approach is satisfying the needs of businesses dealing with static, query intensive data sets if these data sets are relatively small in nature (Salehnia, 2015).

But digital world is growing very fast and become more complex in the volume, variety, velocity in nature. According to the analysis of IBM, 95 percent of the data of the world is generated in the last few years and still generation of data is continued at a rate of 2.5 quintillion of bytes data every data (Padmapriya et. al., 2019). Large data volumes are daily generated from heterogeneous sources and this is due to many technological trends, including the internet of things, social networks, web 2.0, web 3.0 technologies, the proliferation of the cloud computing as well as the spread of smart devices. Continuously growing amount of data has inspired seeking more and more efficient database solutions for storing and manipulating data. The enterprise's database is desired to be accessed as fast as possible. Traditional database technologies have limited storage capacity, rigid management tools and are expensive (Oussous et. al., 2018). They lack of scalability, flexibility and performance needed for big data. Big data management requires significant resources, new methods and powerful technologies.

The objective of the research: to examine the recent trends in database technology, compare relational and NoSQL databases and their application areas, analyze multi-model and cloud-based databases.

The rest of this paper is structured as follows. Section 2 gives an overview of database technologies according their popularity. Section 3 presents comparative analysis of relational and NoSQL database and discuss about their application areas. Section 4 presents multi-model and cloud-based databases. Finally, Section 6 presents conclusions.

The Review of Database Technology

The popularity of database management systems shows recent trends in database technology. DB-Engines has been created and is maintained by Austrian IT consulting company – solid IT. DB-Engines ranks database management systems based on popularity, and the rankings are updated once a month. The ranking data is based on different metrics: number of mentions of the system on websites, general interest in the system, frequency of technical discussions about the system, number of job offers, in which the system is mentioned, number of profiles in professional networks, in which the system is mentioned, relevance in social networks (WEB, a).

In March of 2020 DB-Engines ranked 354 database management systems. Table 1 shows the list of 10 most popular database management systems marked by DB-Engines. The statistics provided in the table 1 illustrates that multi-model database systems indeed seem to be an actual trend in database technology: multi-model database management systems dominate among the top 10 database management systems. Multi-model databases support more than just a single data model. According to world’s leading research and advisory company Gartner (WEB, b) soon all leading database management systems will offer multiple data models in a unified platform.

Oracle, the undisputed leader of relational DBMS enhanced its product with document, graph and RDF capabilities, making it a multi-model DBMS (WEB, a). In the version 8.0 of MySQL a new document store model was introduced, which allows JSON documents to be stored in collections and managed using CRUD operations (WEB, a). Microsoft SQL Server, PostgreSQL, IBM DB2 and many other RDBMS similarly extended their functionalities with multi-model capabilities. The same trend is also seen in NoSQL systems. Redis, the most popular key-value store extended its application scenarios with downloadable modules into a multi-model DBMS (WEB, a). Multi-model databases are getting more popular every year because of their ability to support multiple data models and using one backend for all of them.

DB-Engines lists 354 different database management systems, which are classified according to their database model. Multi-model databases belong to more than one category. Below given chart shows the popularity of each category. The chart data show that relational database model remains the most popular database model (74.8 perc.). Document databases are the most popular database model among NoSQL databases (9.2 perc.).

Table 1. The Most Popular Database Management Systems (DB-Engines, March 2020)

Rank	Database Management Systems	Database Model	Score
1.	Oracle	Relational, Multi-model (*Document store, Graph DBMS, RDF store*)	1340.64
2.	MySQL	Relational, Multi-model (*Document store*)	1259.73
3.	Microsoft SQL Server	Relational, Multi-model (*Document store, Graph DBMS*)	1097.86
4.	PostgreSQL	Relational, Multi-model (*Document store*)	513.92
5.	MongoDB	Document, Multi-model (*Search engine*)	437.61
6.	IBM DB2	Relational, Multi-model (*Document store, RDF store*)	162.56
7.	Elasticsearch	Search engine, Multi-model ()	149.17
8.	Redis	Key-value, Multi-model (*Document store*)	147.58
9.	Microsoft Access	Relational	125.14
10.	SQLite	Relational	121.95

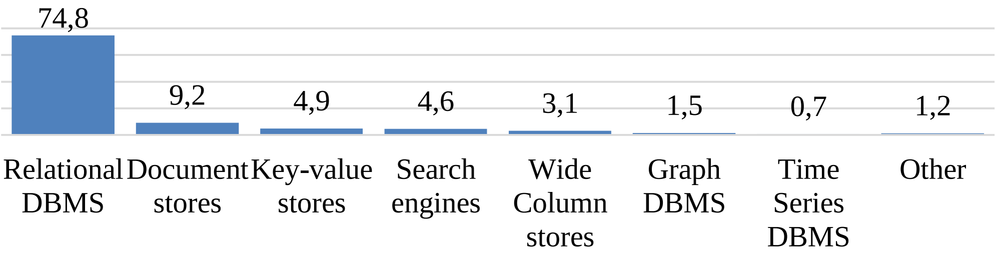

Figure 1. Database popularity according database model in percent (DB-Engines, March 2020)

Another actual trend in database technology is cloud databases. According to world’s leading research and advisory company Gartner the future of the database management systems market is cloud and 75 percent of databases will be cloud-based in three years (WEB, c). A cloud database management system is a database that typically runs on a cloud computing platform, such as Amazon EC2, GoGrid, Salesforce and Rackspace (Muntjir and Junedul, 2017). Among of the databases available on the Cloud, some are relational, and some use a NoSQL or multi-model data model. Most popular cloud-based databases according to DB-Engines presented in table2.

To summarize, relational database model is still very popular. NoSQL databases are probably not here to replace them, but to be used to solve different requirements. The trends in database management systems market show that, in the near future, all leading database management systems will offer multiple data models in a unified platform and will be cloud-based.

Table 2. The Most Popular Cloud-based Database Management Systems (DB-Engines, March 2020)

Rank	Database Management Systems	Database Model	Score
1.	Amazon DynamoDB	Multi-model (*Document store, Key value store*)	62.51
2.	Microsoft Azure SQL Database	Relational, Multi-model (*Document store*)	35.44
3.	Couchbase	Document, Multi-model (*Document store, Key-value store*)	32.08
4.	Microsoft Azure Cosmos DB	Multi-model (*Document store, Graph DBMS, Key-value store, Wide column store*)	31.63
5.	Google BigQuery	Relational	27.76

Comparative Analysis of Relational and NoSQL Databases

Aim of this comparative analysis is to investigate NoSQL databases in comparison with relational database approach in terms of data model, data structure, scheme, scalability, query language, transactions. Below provided table represents the generalized results of the comparative analysis of relational and NoSQL databases.

Table 3. Comparative Analysis of Relational and NoSQL databases

No	Criteria	Relation Databases	NoSQL
1.	Data model	Relation data model	Depends on database. The most common NoSQL database systems can be organized into four categories: document databases, column databases, key-value stores, and graph databases.
2.	Data structure	Database structures data into tables and rows (records).	Database stores data in different data structures. Data structures depends on database.
3.	Scheme	Fixed Scheme. Each record conforms to fixed scheme.	Flexible Scheme
4.	Scalability	Scaling is vertical. More data means a bigger server.	Scaling is horizontal, meaning across servers.
5.	Consis-tency	In relational database Consistency means that all users see the same version of data after the transaction. Relational database provides better consistency than no-relational database.	In NoSQL database „Eventual Consistency“ means that there is no guarantee for reads and writes after the transaction for all entities in the database will be immediately consistent.
6.	Query language	SQL	Query language depends on database.
7.	Trans-actions	Most relational databases are ACID compliant.	NoSQL databases follow BASE (Basically Available, Soft State, Eventual consistency) principles.

A relational database is based on the relational model of data, which organizes data into one or more tables of rows and columns, with a unique key for each row. The relational data model was invented in 1970 by E. F. Codd. Relational databases are also known as SQL databases because of their adoption of ANSI-standard SQL for data manipulation and querying.

A very important notion in database management system is the concept of transaction. In a simplified view, a transaction is a set of commands that perform a given action and takes the database from a consistent state to another consistent state. M. Shalini and S. Dhamodharan emphasize that relational database transactions possess qualities that ensure the reliability of transactions. These qualities of transactions are called ACID qualities: atomicity, consistency, isolation, durability. Atomicity is an ability of database to ensure that all the actions of the transaction or none of them would be executed. Consistency is the keeping of the permanent state of the database before the start of the transaction and after its finish. Isolation is a possibility to separate an executed transaction from other processes. Durability is the ensuring that if a message concerning successfully executed transaction was received, the results of transaction cannot disappear in themselves.

Relational databases are best-suited where transaction integrity and data consistency is most important. They are tried and tested for applications which need to satisfy rigid ACID principles. Typically, relational databases are used for online transaction processing systems like finance and accounting, payroll, HR, fleet management, reservations, inventory, sales, order processing, CRM or trading support systems, Data warehouses or data marts.

According to Pore and Pawar (2015) relational databases having the variety of limitations due to constant growth of stored and analyzed data, e.g. the restrictions on scalability and storage, and efficiently losing of query as the volume of data is very large, and the storing and managing of larger databases become challenging. Relational databases are oriented towards data integrity, so the distribution of this model is made owing to data accessibility. Strict ensuring of data integrity does not allow systems to be easily distributed, and this means that automatic system scalability and accessibility become a very complicated and expensive task of the infrastructure. Relational database can be scaled just vertically, because entire database has to be hosted in a single server. This is necessary in order to ensure reliability and continuous availability of data (Sharma, 2019). Amazon, Facebook, and LinkedIn are among the first companies to discover the serious limitations of SQL database technology for supporting big data and big user's requirements (Pore and Pawar, 2015).

NoSQL database technologies entered the market in 1998. They were designed to address performance and scalability requirements of web-based application which cannot be addressed by traditional relational databases. NoSQL is best considered with the acronym NOSQL – Not Only SQL – which more accurately represents an approach that combines non-relational databases with the use of relational ones (Chen and Lee, 2019). The most common NoSQL database systems can be organized into four categories: document databases, column databases, key-value stores, and graph databases (Faraj et. al., 2014).

The various problems faced in relational databases are overcome by NoSQL databases. The techniques which NoSQL provides are related to high scalability, availability and fault tolerance (Sharma, 2019). NoSQL databases were designed to scale horizontally. Instead of increasing power of one single server we just need to add more server instances to get expected power (Sharma, 2019). The relational databases strongly follow the ACID (Atomicity, Consistency, Isolation, and Durability) properties while the NoSQL databases follow BASE (Basically Available, Soft State, Eventual consistency) principles (Chandra, 2015). BASE model helps to provide flexibility to the NoSQL data so that the data might be structured properly (Sharma, 2019). The database is basically available, means that, if some part of the database are currently unavailable, other part of the database continue the function. Soft state refers to state change without input which is required for eventual consistency. Soft-state means that the state of the system cannot be guaranteed – different instances of the same data might have different content as it may not have yet captured recent updates in another part of the cluster. Eventually consistent means that modified data may not become consistent across the entire system immediately but will become consistent with time (Batra et. al., 2015). So, the data will be consistent in the future.

The NoSQL databases emerge with different security issues (Zaki, 2017). The main focus of NoSQL databases is handling the new data sets, with less priority on security. NoSQL databases do not provide any feature of embedding security in the database itself. Developers need to impose security in the middleware (Zaki, 2017).
NoSQL is a database technology driven by Cloud Computing, the Web, Big Data and the Big Users. **NoSQL now leads the way for the popular internet companies such as LinkedIn, Google, Amazon, and Facebook – to overcome the drawbacks of the 40-year-old relational database management systems** (Shetty and Chidimar, 2016)**.**

Multi-model and Cloud-based Databases

Polyglot persistence is a technology which is used to store data in multiple data storage technologies, which is chosen based upon the way data is used in applications/software. Polyglot Persistence has some disadvantages such as increase in complexity and no support for maintaining data consistency. In polyglot persistence different databases are connected to every other database engine, which will require taking care of the different database engines

The multi-model databases are the way to solve the polyglot persistence problem and offer advantages of data modeling and managing without disadvantages of using different database technologies for each kind of data. The multi-model databases provide all models (a particular subset) in one database with unified query language and API for all supported models. These databases can be understood as databases which can store data in different formats (relations, documents, graphs, objects, etc.) under one management system (Płuciennik and Zgorzałek, 2017). Multi-model databases are getting more popular every year because of their ability to support multiple data models and using one backend for all of them.

The first main wave of multi-model databases has appeared soon after the beginning of the new millennium with the emergence of XML data (Lu and Holubov, 2019). The relational database management systems were extended towards XML, usually via the SQL/XML standard or its variation**.** They were transformed to so-called XML-enabled databases. According L. Jiaheng and I. Holubov (2019) the second wave can be observed after 2010 with the arrival of the era of big data. The XML-enabled databases were often extended towards the data models used in NoSQL databases and combined their original data format with other formats.

The main advantage of a multi-model database is a single system for all required models. Developers have just one database to maintain, just one adapter for their application and also they have to learn details just about one database (such as query syntax, architecture or topology). The next advantage of a multi-model database is

the cost of scaling up. In the case when an application requires scaling of databases, developers can scale just the multi-model database technology instead of scaling all databases separately (scale a relational database, a document database, etc.).
Multi-model databases can be classified according their architecture that defines how databases support different models. Multi-model databases with complex engine architecture transform all supported data types to a single core model. Their engine has to pre-process all operations for the core model. Multi-model databases with layer architecture support different models via different layers on top of an engine. Data are stored in the relevant model an each data model has its own component which communicates with the engine.

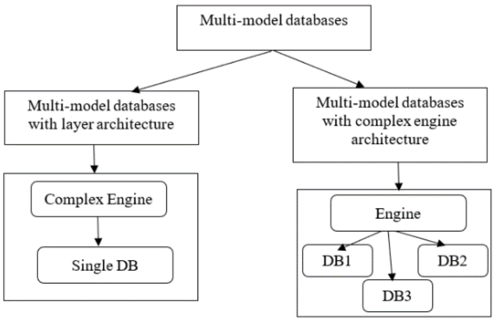
Multi-model databases
Multi-model databases with layer architecture
Multi-model databases with complex engine architecture
Complex Engine
Single DB
Engine
DB1
DB2
DB3

Fig 2. Classification of multi-model databases

A cloud-based database is defined as the database that typically runs on cloud computing platform. There are two types of cloud-based database models: **traditional** and **database as a service** (DBaaS).

In the case of traditional cloud-based database model, an organization purchases virtual machine space from a cloud services provider, and the database is deployed to the cloud. The organization is responsible for oversight and database management. This model allows users to install and maintain their own databases. The users run databases on the cloud independently, using a virtual machine image. In this case a cloud database can be a conventional database such as SQL Server and MySQL and this traditional database can be installed, configured and maintained by the user himself on a cloud server

In the case of database as a service (DBaaS), service operator is responsible for database management system provisioning, configuration, scaling, performance tuning, backup, privacy, and access control. Users purchase access to a database service, maintained by a cloud database provider. Many corporations and e-commerce companies have already adopted database as a service and they are currently reaping full benefits from this service (Mutisya and Kahonge, 2017). Cloud-based databases can support relational, NoSQL and multi-model database models.

Conclusions

The analysis of the results of the researches conducted by international companies shows that relational model heads the top ten of the most popular database models.

The comparative analysis of relational and NoSQL databases showed these systems differ in data model used, flexibility of schemes, supported database standards, used queries language, features of transactions, data distribution opportunities. The various problems faced in relational databases are overcome by NoSQL databases. The techniques which NoSQL provides are related to high scalability, availability and fault tolerance.

Multi-model databases are new trend for the database management system, which uses a single platform to manage data stored in different models. Compared to the polyglot persistence technology that employs separate data stores to satisfy various use cases, multi-model database is considered as the next generation of data management system incorporating flexibility, scalability, and consistency.

The growing popularity of cloud-based databases is marking the beginning of new era of databases. Cloud-based databases can support relational, NoSQL and multi-model databases models.

References

Batra, A., Pahwa, A., Sharma, D. (2015). Mongo DB - NoSql Database: A Survey. *Advances in Computer Science and Information Technology (ACSIT)*, 2(5), 495-499.

Chandra, D. (2015). BASE analysis of NoSQL database. *Future Generation Computer Systems*, 52, 13-21.

Chen, J., Lee, W. (2019). An Introduction of NoSQL Databases Based on Their Categories and Application Industries. *Algorithms*, 12(5), 106-111.

Faraj, A., Rashid, B., Shareef, T. (2014). Comparative Study of Relational and Non-relations database performances using ORACLE and MONGODB systems. *International Journal of Computer Engineering and Technology (IJCET)*, 5(11), 11-22.

Lu, J., Holubov, I. (2019). Multi-model Databases: A New Journey to Handle the Variety of Data. *ACM Computing Surveys*, 0(0), 1-38.

Muntjir, M., Junedul, M. (2017). Cloud Database Infrastructure: Database System Transference in Cloud Computing Management and Security. *International Journal of Computer Trends and Technology*, 47(1), 15-28.

Mutisya, M., Kahonge, A. (2017). Utilization of Cloud Database on Criminal Background Check. *International J. Advanced Research in Computer Science & Technology*, 5(4), 46-51.

Oussous, A., Benjelloun, F., Lahcen, A., Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University – Computer and Information Sciences*, 30, 431-448.

Padmapriya, N., Parteeban, N., Kamal, N., Suresh, A., Arun, S. (2019). Enhanced Cyber Security for Big Data Challenges. *International Journal of Innovative Technology and Exploring Engineering*, 8(10), 3478-3481.

Płuciennik, E., Zgorzałek, K. (2017). The Multi-model Databases – A Review. *Communications in Computer and Information Science*, 716, 141-152.

Pore, S., Pawar, S. (2015). Comparative Study of SQL & NoSQL Databases. *International Journal of Advanced Research in Computer Engineering & Technology*, 4(5), 1747-1753.

Salehnia, A. (2015). Comparisons of Relational Databases with Big Data: a Teaching Approach. *Semantic Scholar*, 1-8.

Shalini, M., Dhamodharan, S. (2016). Performance and Scaling Comparison Study of RDBMS and NoSQL (MongoDB). *An International Journal of Advanced Computer Technology*, 3 (11), 1270-1275.

Sharma, Y. (2019). Case Study of Traditional RDBMS and NOSQL database system. *International Journal of Research Granthaalayah*, 7, 351-359.

Shetty, D., Chidimar, S. (2016). Comparative Study of SQL and NoSQL Databases to evaluate their suitability for Big Data Application. *International Journal of Computer Science and Information Technology Research*, 4(2), 314-318.

Zaki, A. (2017). Nosql Databases: New Millennium Database for Big data, Big Users, Cloud Computing and its Security Challenges. International Journal of Research in Engineering and Technology, 403-409.

WEB (a). *DB-Engines Ranking*, https://db-engines.com/en/ranking.

WEB (b). *Gartner. Magic Quadrant for Operational Database Management Systems*, https://www.gartner.com/doc/3467318/ magic-quadrant-operational-database-management.

WEB (c). Feinberg, D., Andrian, M., Ronthal, A. *The Future of the DBMS Market Is Cloud*, https://www.gartner.com.

Received October 21, 2020, accepted November 16, 2020