ASSIGNMENT 4

Report submitted to Dakota State University in partial fulfillment
of the requirements for the course of

INFS 774: Big Data Analytics
Summer 2018

June 10, 2018

By
Yuchi Hu

Course Instructor:

Dr. Jun Liu

The relational database has been the dominant type of database in the software industry for over 20 years, providing "mechanisms to store data persistently, concurrency control, transactions, mostly standard interfaces and mechanisms to integrate application data, reporting." (Sadalage, 2014). Relational databases organize data into tables with rows and columns; each row in a table can be identified by a unique key. SQL (Structured Query Language) is the primary language used by relational databases to query and maintain data. Despite the popularity and success of relational databases, application developers have been frustrated with the "impedance mismatch between the relational model and the in-memory data structures." (Sadalage & Fowler, 2012a). The exponential increase in the amount of data has also led to a need for alternatives to relational databases. Computer clusters are needed to store, process, and analyze large volumes of data, and relational databases do not run efficiently on clusters. A new type of databases, called NoSQL databases, is such an alternative to relational databases and has been gaining popularity in the last decade.

NoSQL databases can also be called aggregate-oriented databases because they use an aggregate data model that runs on clusters instead of the relational model used by relational databases. An aggregate is a collection of data that we interact with as a unit, and aggregates make it easier for the database to manage data storage over clusters (Sadalage, 2014). Key-value, document, and column-family databases can all be considered aggregate-oriented databases. Sadalage (2014) describes the key features of NoSQL, or aggregate-oriented, databases as:

- They do not use the relational model.
- They run well on clusters.
- They are mostly open-source.

- They are built for the 21st century web estates.
- They are schema-less.

NoSQL databases make data distribution easier by moving aggregates, which contain all related data. There are two prominent forms of data distribution models in NoSQL databases: sharding and replication. Sharding distributes different data across multiple servers, so each server acts as the single source for a subset of data; replication copies data across multiple servers, so each bit of data can be found in multiple places (Sadalage & Fowler, 2012a).

The ability of NoSQL databases to store and process larger datasets and carry out tasks more efficiently than relational databases does not, however, render relational databases obsolete. NoSQL can stand for "Not Only SQL", which means database solutions should be chosen based on the nature of the problem, so relational databases can still be more suitable for some problems than NoSQL databases, particularly when we are dealing with smaller data. Furthermore, there is a huge list of NoSQL databases, and the ability to choose which NoSQL database to use allows us to tackle many different kinds of problems. This leads us to the term "polyglot persistence", which is the idea that different applications should use different data storage technologies based on the varying data storage needs. Polyglot persistence can also occur within a single application since different parts of an application's data store have different access characteristics (Sadalage & Fowler, 2012b).

NoSQL databases can be categorized into four types: key-value, document, column-family, and graph.

- Key-value databases are optimized for read-heavy application workloads (such as social networking, gaming, media sharing, and Q&A portals) or compute-intensive

workloads (such as a recommendation engine) ("What Is NoSQL?" n.d.). Examples include Riak, Redis, Memcached, Berkeley DB, upscaledb, Amazon DynamoDB, Project Voldemort, and Couchbase (Sadalage, 2014).

- Document databases store semistructured data as documents, typically in JSON or XML format. The schema for each NoSQL document can vary, which provides more flexibility in organizing and storing application data and reducing storage required for optional values ("What Is NoSQL?" n.d.). Examples include MongoDB, CouchDB, Terrastore, OrientDB, and RavenDB (Sadalage, 2014).
- Column-family databases are optimized for reading and writing columns of data as opposed to rows of data. Column-oriented storage for database tables drastically reduces the overall disk I/O requirements and reduces the amount of data needed to load from disk ("What Is NoSQL?" n.d.). Examples include Cassandra, HBase, Hypertable, and Amazon DynamoDB (Sadalage, 2014).
- Graph databases store vertices and directed links called edges and can be built on both SQL and NoSQL databases. Vertices and edges can each have properties associated with them ("What Is NoSQL?" n.d.). Examples include Neo4J, Infinite Graph, OrientDB, and FlockDB (Sadalage, 2014).

Why use a NoSQL database? The many types of NoSQL databases give us great flexibility in picking a database that's suitable for a particular application's needs. NoSQL databases can handle larger data volumes, have lower latency, and are easier to scale than relational databases. The rise of NoSQL databases does not mean they are replacing relational databases altogether: relational databases are more established, provide better support, and virtually all use SQL. As previously mentioned, the concept of polyglot

persistence means that data storage technologies should be chosen based on the data storage needs of the applications. In short, choose the right tool for the job!

References

Sadalage, P. (2014). *NoSQL Databases: An Overview*. Retrieved from https://www.thoughtworks.com/insights/blog/nosql-databases-overview

Sadalage, P. & Fowler, M. (2012a). *NoSQL Distilled*. Boston, MA: Addison-Wesley.

Sadalage, P. & Fowler, M. (2012b). *Introductory Infodeck to NoSQL Databases*. Retrieved from https://martinfowler.com/articles/nosql-intro-original.pdf

What Is NoSQL? (n.d.). Retrieved from https://aws.amazon.com/nosql/