Differences Between SQL and NoSQL

By: Edward Welborn

Date: 06-08-2019

Link to Final Spark Page: <https://spark.adobe.com/video/AnTFpx71t1ofc>

# Relational Vs. Non-Relational (or SQL Vs. NoSQL)

The major differences between an SQL database and a NoSQL database

The key major difference between SQL and NoSQL is pretty significant. SQL databases are a relational database, while NoSQL databases are not. This means, that SQL databases strictly cling to relations between child and parent tables. NoSQL is document or distributed storage that does not require any type of table to function.

NoSQL is dynamic for document or unstructured data, and SQL is a well-designed structured database that are pre-defined.

# Three NoSQL Twitter Features

Discuss three (3) specific features, from Twitter, that could be in NoSQL. A feature is something like a tweet, like, comment, follow, reply, add emotion, etc.

The original tweet will be the one to feed the rest of the features of in a NoSQL environment. Being that NoSQL is a document driven database instead of a relational database, the parent document feeds to the children and back. Therefore, each feature of comment, reply, and tweet would generate a specific document that would be send back and forth to the parent, being the original tweet would be the parent.

# Twitter Features How To

How would each of the 3 features in Twitter be used (Focus on how it's used with NoSQL)

The feature of Tweeting could be a NoSQL document, being the parent document, every tweet would be a document, that is fed to the children

The feature of the Comment would be a child document in NoSQL, being fed from the parent, and feed back to the parent to the website and application.

This feature of replying to the original tweet would also be a child of the parent tweet, feeding back and forth between the two documents which will populate the site and application.

# Twitter Features Pros/Cons

What are the pros and cons of each of the 3 features (This should try to be NoSQL related, but does not have to be)

The major con for NoSQL especially with twitter is a lack of a universal SQL query language. NoSQL does have a query type of language, but in order to move over to NoSQL from a SQL language, the entire codebase will have to be rewritten.

# One Relational Facebook Feature

Discuss one (1) specific feature, from Facebook, that could be in a Relational (SQL) Database. A feature is something like a post, like, comment, follow, reply, add emotion, login process, etc.

The login process for Facebook, could be a relational database. It would have to connect to the backend server for authentication, so that the application knows it is a valid login or not.

# Facebook Feature How To

How would the feature in Facebook be used (Focus on how it's used with SQL)

The login feature would post the login username and password, then after that information is entered would contact the backend server to make sure the credentials are correct in the database.

# Facebook Feature Pros/Cons

What are the pros and cons of the feature (This should try to be SQL related, but does not have to be)

# Four NoSQL Database Types

Discuss four (4) types of NoSQL Databases, with descriptions of each.

GIRISH KUMAR, (N.D), gives us in his article these basic 4 types of NoSQL databases.

1. **Key-Value Store** – It has a Big Hash Table of keys & values {Example- Riak, Amazon S3 (Dynamo)}

**2. Document-based** **Store- It**stores documents made up of tagged elements. {Example- CouchDB}

1. **Column-based Store-**Each storage block contains data from only one column, {Example- HBase, Cassandra}
2. **Graph-based**-A network database that uses edges and nodes to represent and store data. {Example- Neo4J}

# Five NoSQL Databases

Discuss five (5) different NoSQL Databases, with descriptions about each one, and what each is used for.

Akram Hussain, (2014), gives five examples in his well-written article:

Apache Cassandra is one of the few database systems that can process data in real time and generate high performance and maintain high availability. Cassandra is perfect for ‘mission critical’ big data projects, as Cassandra offers ‘no single point of failure’ if a data node goes down.

The added benefit in using MongoDB is that it provides high performance, high availability, and easy scalability for large sets of unstructured data in JSON-like files. MongoDB is the ultimate opposite to the popular MySQL. MySQL data has to be read in rows and columns, which has its own set of benefits with smaller sets of data.

Neo4j is the frontrunner of the graph-based model. As a graph database, it manages and queries highly connected data reliably and efficiently. The data collected from sites and applications are initially stored in nodes that are then represented as graphs.

OrientDB is an open source NoSQL hybrid graph-document database that was developed to combine the flexibility of a document database with the complexity of a graph database

# Hadoop includes a database known as HBase, which runs on top of HDFS and is a distributed, column-oriented data store. HBase is also better known as a distributed storage system for Hadoop nodes, which are then used to run analytics with the use of MapReduce V2, also known as Yarn.

# Five NoSQL Databases Pros/Cons

What are the Pros and Cons of each product. (This could be price, platform differences, etc)

According to Anmol Sarna(2018), the major cons of using Apache Cassandra are:

There is no support for ACID Properties, there is no support for Aggregates, Cassandra has a high latency when making a high volume of calls to the database, there is no join or subquery support, a workaround is possible but could lead to other issues, data duplication could be an issue, reads from the database can be significantly slower, since Cassandra is optimized more for reading data, than writing data, and memory allocation and leaks can be prevalent in the JVM memory management modules.

The Dataflair team, give a very detailed pros and cons of using the MongoDB, Joins not Supported

MongoDB doesn’t support joins like a relational database. Yet one can use joins functionality by adding by coding it manually. But it may slow execution and affect performance.

MongoDB has a high memory usage, since the database stores key names for each key:value pair.

Also, due to not having the functionality of joins, there is the possibility of data redundancy. Which may result in the higher usage of memory, and possibility of latency.

MongoDB has a limited data size for its documents. The maximum you can have in any document is 16mb.

MongoDB also has limited nesting capabilities, you are not able to perform nesting of documents more than one hundred levels.

According to [www.slant.co](http://www.slant.co), orientDB has only a few limitations at this point, it is not well known so finding documentation, limitations, code snippets or general support may be hard to come by. Due to the fact that orienDB is a hybrid database, the learning curve is much higher than most NoSQL or SQL database engines. When dong bulk inserts in this database it is common to receive an out of memory error, so single inserts are optimal with this database engine.

KnowledgeHut(2016) wrote a very interesting article on the disadvantages of the Hadoop database engine. The lack of preventive measures, security measures are disabled by default, so the database administrator would need extra special care to make sure the data is secure in this environment. Hadoop is optimized for large data, so it has an especially hard time at manipulating small data. Since Hadoop is java reliant, it has been proven that java is a risky endeavor and poses a risk factor since cyber hackers can easily exploit the framework for nefarious uses.

Neo4J as well as other graphDB engines have several inherent problems, such as it proves to be inferior to transactional relations, such as accounting. Basic queries seem harder to do, such as summing, max, and simple calculations. You have to learn a new query language, called cipher, which doesn’t have a lot of support so the learning curve is a lot higher than most other database engines.

# Weather App (Two NoSQL Solutions)

Discuss two (2) NoSQL databases and how they could be used for a “weather app.”

Joab Jackson of pcworld.com(2014) reported that Weather Channel, found that it had to switch to a NoSQL MongoDB data store in order to more quickly develop apps and add features to its range of Internet-based weather information services. MongoDB provided them the best combination of developmental ease as well as fast response times.

A graphDB such as Neo4J would also do well for parts of a weather application due to its fast response time for reading and writing data, while simultaneously performing large queries.

# References

N/A, (N.D.), retrieved from <https://www.agiratech.com/the-key-differences-between-sql-and-nosql-database/>

GIRISH KUMAR, (N.D), retrieved from <https://www.3pillarglobal.com/insights/exploring-the-different-types-of-nosql-databases>

Akram Hussain, (October 31, 2014), retrieved from <https://hub.packtpub.com/top-5-nosql-databases/>

Anmol Sarna (August 17, 2018), retrieved from <https://blog.knoldus.com/is-apache-cassandra-really-the-database-you-need/>

DATAFLAIR TEAM (PUBLISHED APRIL 17, 2018 · UPDATED SEPTEMBER 14, 2018), <https://data-flair.training/blogs/advantages-of-mongodb/>

N/A, (N.D.), retrieved from <https://www.slant.co/options/3700/~orientdb-review>

KnowledgeHut, (24th Nov, 2016), retrieved from <https://www.knowledgehut.com/blog/big-data/top-pros-and-cons-of-hadoop>

Joab Jackson (OCTOBER 14, 2014), retrieved from <https://www.pcworld.com/article/2833772/the-weather-channel-forecasts-heavy-nosql-ahead.html>

Mingxi Wu (May. 16, 2019) retrieved from <https://dzone.com/articles/what-are-the-pros-and-cons-of-using-a-graph-databa>