SQL Vs NoSQL

By: Mulan Lau

Date: May 12, 2019

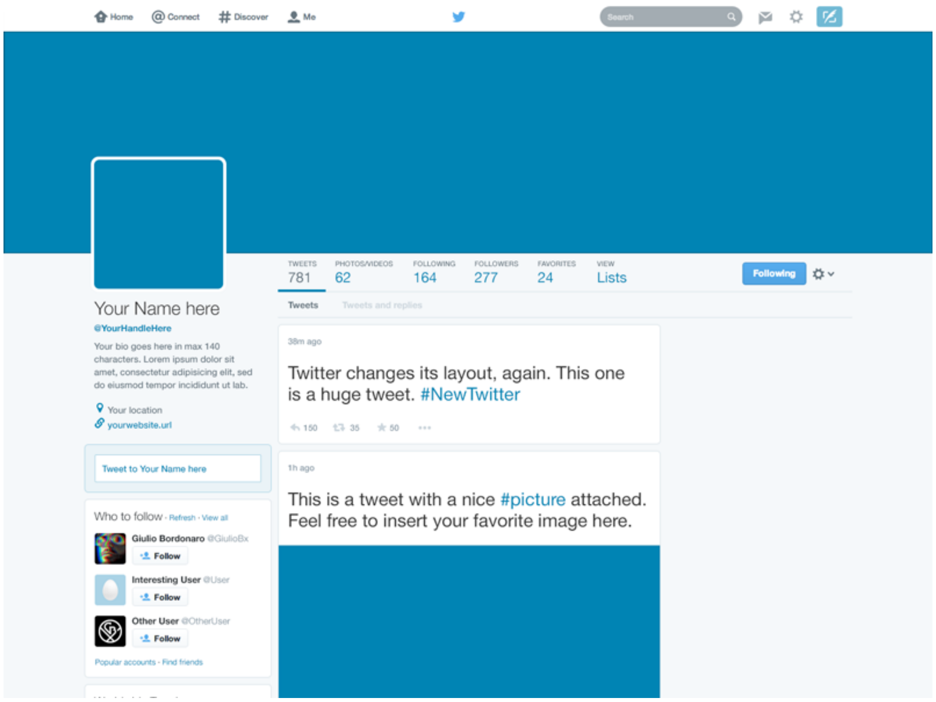
Final Spark Page:
https://spark.adobe.com/page/7vLQgf5PQ1EPh/

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

The three main differences between SQL and NoSQL are language, scalability, and structure. SQL uses SQL which is a structured query language, whereas NoSQL databases can be created with or without any structure to the data. When it comes to scalability the two are quite different also. SQL adds work to the server, singular, the database is hosted on when the database increases in size. NoSQL on the other hand, increases the number of servers the database is hosted on when the database expands. Structurally, SQL databases have a table structure versus NoSQL databases which can have a variety of structures including graph-based, document-based, and key-value pairs. Additionally, SQL databases are bound by structured schemas created before data entry whereas NoSQL databases have highly dynamic schemas that allow the user to add structure before or after data entry. Lastly, SQL databases use SQL commands to manipulate data while NoSQL databases use object-oriented APIs. (MongoDB, 2019b; Xplenty, 2017)

Three NoSQL Twitter Features

Three Twitter features that could be NoSQL features are a tweet, profile, and reply.



Your bio goes here in max 140 characters. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut lab.

38m ago

Twitter changes its layout, again. This one is a huge tweet. #NewTwitter

1h ago

This is a tweet with a nice #picture attached. Feel free to insert your favorite image here.

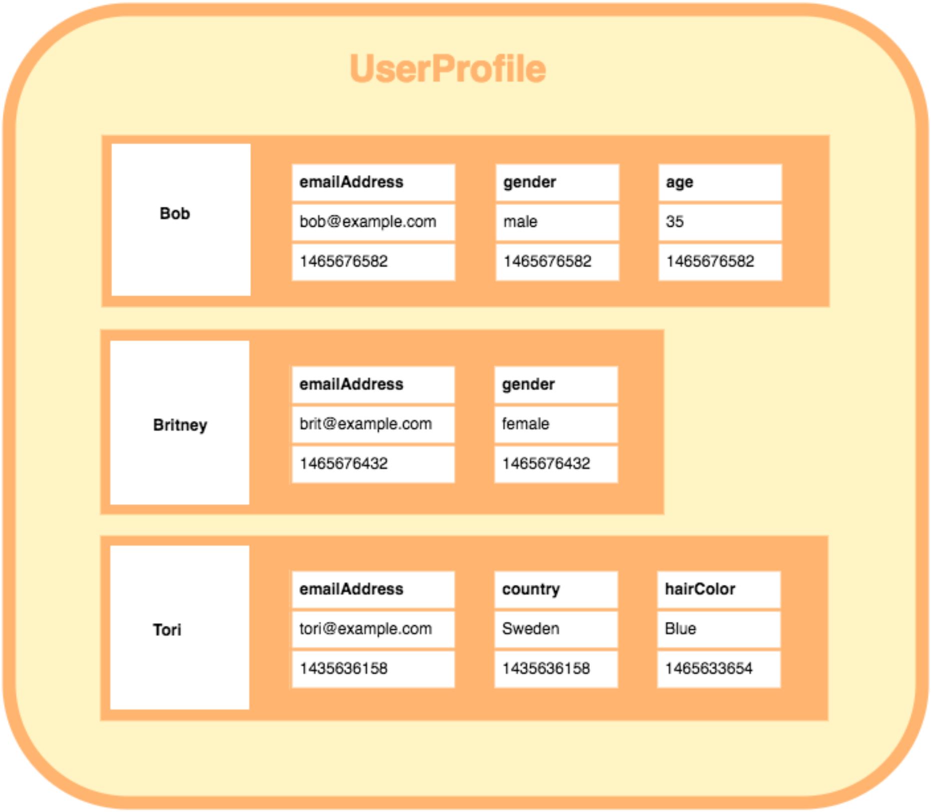
(Bordonado, 2014)

Twitter Features How To

- Tweet: A tweet could be created using the document-based schema in NoSQL. Each tweet being the equivalent of a JSON object with the contents of the tweet belonging to various fields within the object. For example, if the user were to tweet a phrase like "At the beach." and include a picture of the user at the beach then, the text portion would be the value of a text field and the picture would be the value of an image field. This structure would allow each tweet to have as many or as little fields as necessary based on the contents of the tweet.
- Profile: Twitter users are required to create a profile in order to leave comments, follow, and interact in general with the other functions of Twitter. A profile could be created using the wide-column based NoSQL structure. The row key could be the user id and each column could refer to a field present in the profile such as profile picture, name, bio, etc. The user would fill out the desired fields and only those values that were entered would appear in that users row.
- Reply: A reply could also use a wide-column based schema. Each reply would have a unique row key which could be the reply id and columns for the username, post id, like count, reply id (if the user was replying to a reply), and text. Then, only the values provided would be shown in each

row. For example, if the reply was to a post by a user and had some text, but no likes, then the row for this reply would have the username value, the post id value, and the text value.

- Section References: (AWS, n.d.; Bhardwaj, 2016; Knight, 2019)



Bob	emailAddress	gender	age
	bob@example.com	male	35
	1465676582	1465676582	1465676582

Britney	emailAddress	gender
	brit@example.com	female
	1465676432	1465676432

Tori	emailAddress	country	hairColor
	tori@example.com	Sweden	Blue
	1435636158	1435636158	1465633654

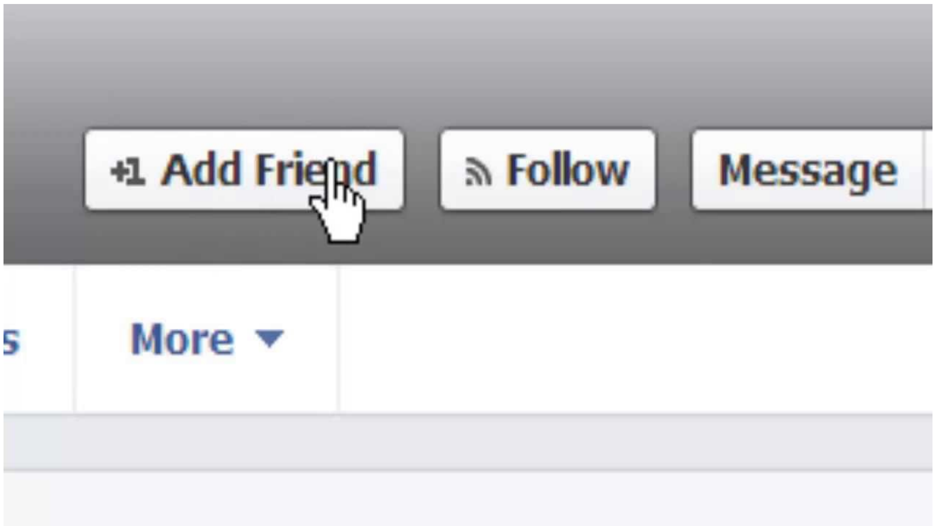
(I., 2016)

Twitter Features Pros/Cons

- Tweet: The pros of structuring a tweet using the document-based schema are that the tweet could potentially contain any number of individual fields and values for images, text, or other content. The cons are that every tweet would require certain fields such as username for the user posting the tweet and date for the date and time the tweet was posted. This is a potential con as it might cause a bit of repetition, but it could end up not being a factor depending on the way the program getting the information and inputting it into the database is written.
- Profile: The pros of using the wide-column schema for profiles are that each field has already been defined and will only appear with a value if there is a value to show. This can save space as every profile won't have every column if every column doesn't contain a value. The cons of using this schema are that the database creator would need to ensure that all the possible fields have a corresponding column. However, as every user has access to the same possible fields for input, this will probably save time in the long run.
- Reply: The pros of using a wide-column schema for replies are that every reply has many fields that are the same with only the values varying, such as username, post id, and date, and a reply does not need to have a value for every available column. The cons are that if the user is able to add content to the reply that does not already

have a column then the data may be misplaced or lost. The solution may be to ensure that all possible inputs have a column or to change the schema to a document-based one which would allow for additional fields and values for individual replies should the reply need it.

- Section References: (AWS, n.d.; Bhardwaj, 2016; Knight, 2019)



(Arabic, 2013)

One Relational Facebook Feature

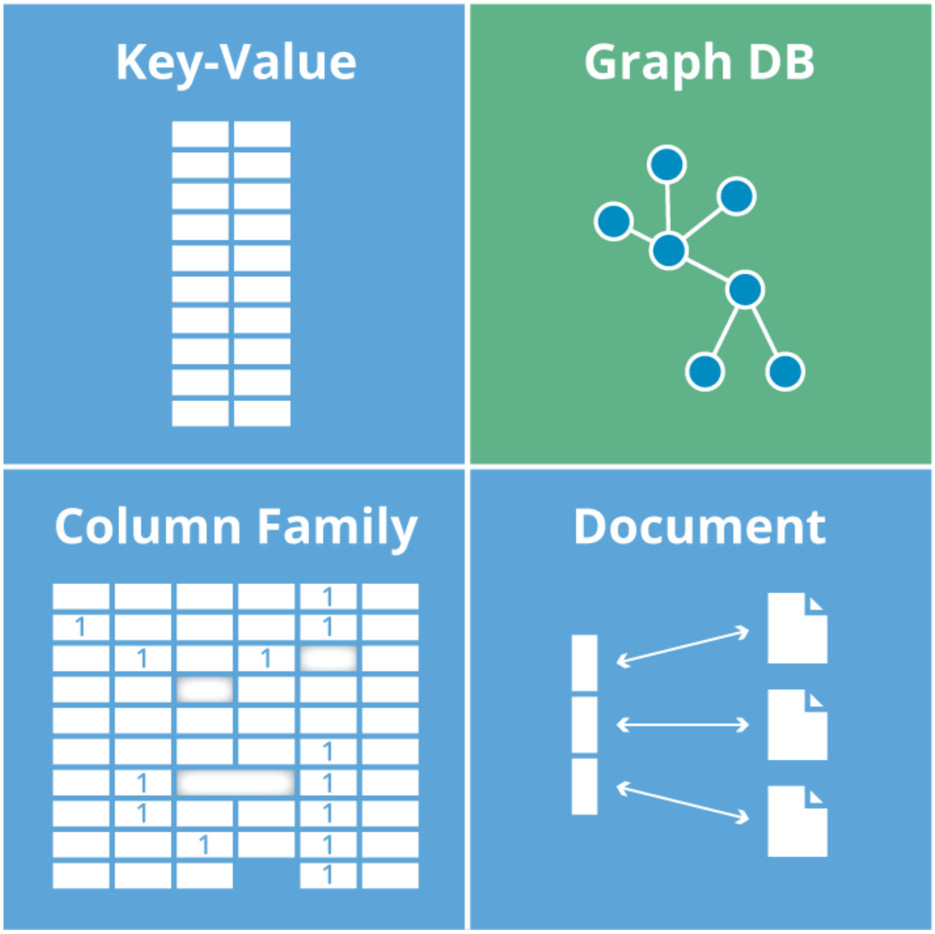
One Facebook feature that could be relational is the add friend feature.

Facebook Feature How To

The add friend feature could use a relational database with multiple tables. The button itself would have the table of add_friend. This table would have columns for user id, friend id, status, and date sent. The user id would be the unique id of the user pressing the button, the friend id would be the unique id of the friend the user wants to add, the status column would be a foreign key to a status table, and the date sent column would contain the date and time that the button was pressed. The status table would contain the various possible statuses for the request, pending, added, or denied. Once the status for the request changed to either added or denied, the request could be taken down from the add friend table. If the status changed to added, then the friend id could be added to a friends table that contains columns for the user id, indicating the user sending the requests, and the friend id. This friends table would contain all the users friends and allow people to find friends by user as well. (Facebook, n.d.)

Facebook Feature Pros/Cons

The pros for the add friend feature being created this way, as explained above, are that the data generated by the feature is stored in an organized manner. The cons are that if the program pushing/pulling data for the database isn't written well there is a potential for data loss.



(Sasaki, 2018)

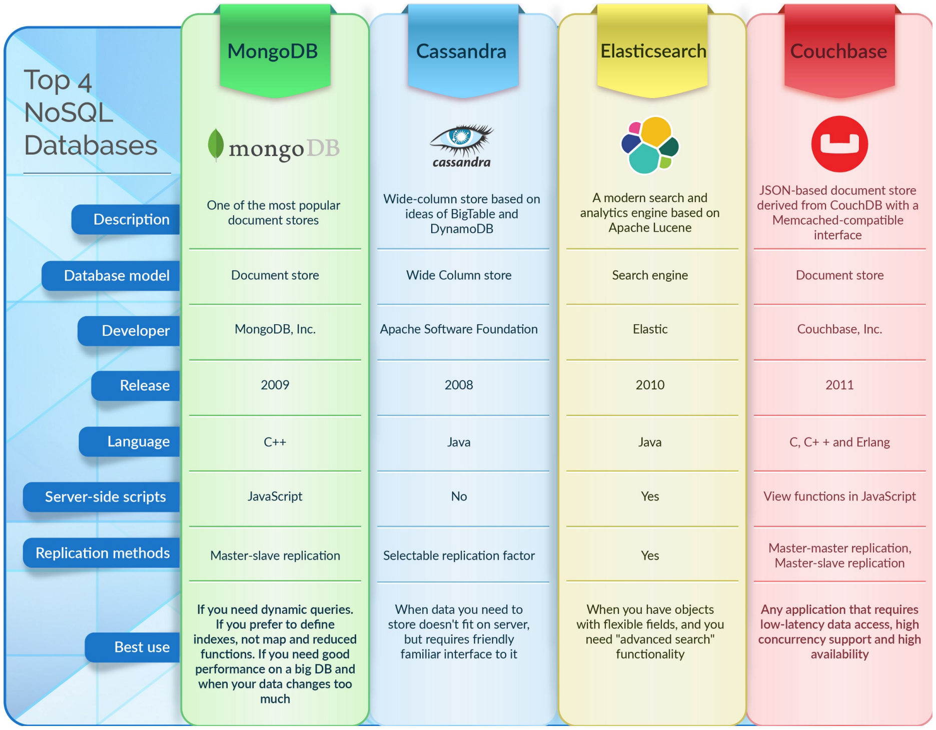
Four NoSQL Database Types

- Wide-Column aka Column Store: This database type uses a keyspace containing column families which contain rows containing more columns. The column families act like tables do in a relational database and can be used to group rows by general type. Each row can contain multiple columns, however a column will only appear in a row if the column has a value.

- Key-Value: Key-Value databases are very simple and act similar to dictionaries. The key is used to access the value. Due to the simplicity of key-value databases, they tend to have high performance speeds.
- Graph: Graph databases are databases built of nodes of data and the relationships between those nodes. They are ideal for things like social networking because of the emphasis placed on the relationships between nodes.
- Document-Based: Document-Based databases use document objects similar to JSON objects which contain all the data needed for each document within the document itself. This makes querying data quicker and easier because each document object is independent and self-sufficient in containing all of its necessary data.
- Section References: (I., 2016; Namistyuk, n.d.; Sasaki, 2018; MongoDB, 2019a)

Five NoSQL Databases

- MongoDB: MongoDB is a free to use document-based database.
- Amazon DynamoDB: DynamoDB is an Amazon hosted key-value and document-based database.
- Apache Cassandra: Cassandra is a Facebook developed column-based database.
- Riak: Riak is an open source key-value database.
- Neo4j: Neo4j is an open source graph-based database.



Top 4 NoSQL Databases	MongoDB	Cassandra	Elasticsearch	Couchbase
Description	One of the most popular document stores	Wide-column store based on ideas of BigTable and DynamoDB	A modern search and analytics engine based on Apache Lucene	JSON-based document store derived from CouchDB with a Memcached-compatible interface
Database model	Document store	Wide Column store	Search engine	Document store
Developer	MongoDB, Inc.	Apache Software Foundation	Elastic	Couchbase, Inc.
Release	2009	2008	2010	2011
Language	C++	Java	Java	C, C+ + and Erlang
Server-side scripts	JavaScript	No	Yes	View functions in JavaScript
Replication methods	Master-slave replication	Selectable replication factor	Yes	Master-master replication, Master-slave replication
Best use	If you need dynamic queries. If you prefer to define indexes, not map and reduced functions. If you need good performance on a big DB and when your data changes too much	When data you need to store doesn't fit on server, but requires friendly familiar interface to it	When you have objects with flexible fields, and you need "advanced search" functionality	Any application that requires low-latency data access, high concurrency support and high availability

(DashMagazine, 2017)

Five NoSQL Databases Pros/Cons

- MongoDB: The pros of MongoDB are that it's free to use, has extensive documentation, has a large community, is a distributed database, is very flexible, and has an aggregate pipeline for analyzing data. The cons of MongoDB are that it doesn't directly support data joins and it lacks clearly defined relationships between pieces of data, referential integrity.
- Amazon DynamoDB: The pros of DynamoDB are that it is hosted by a trusted, durable company, Amazon, it's highly flexible, scalable, serverless, and it maintains high performance regardless of scale. The cons are that a hot key - frequently accessed key - in DynamoDB is not what many developers consider a hot key. In other words, the access rate may not be very high for other databases, but this same rate could be high enough to cause problems with the DynamoDB database. Additionally, this can greatly limit the scalability of the DynamoDB.
- Apache Cassandra: The pros of Cassandra are that it's linearly scalable, high performance, flexible, and designed to work efficiently on cheap hardware. The cons are that data joins are not supported, it lacks referential integrity, and options for pulling data are limited.
- Riak: The pros of Riak are that it's flexible, high performance, provides APIs for CRUD (create, read, update, delete) operations, includes search operations, scalable, and resilient in the case of a

server failure. The cons are that it has less support than larger databases such as MongoDB and Cassandra and it can be expensive to run.

- Neo4j: The pros of Neo4j are that Neo4j has a large active community, support, high performance, scalability, and is easy to learn. The cons are that you need to pay for the Enterprise version to use Neo4j for education or business and only the paid version supports database scaling.
- Section References: (Brazeal, 2017; McNulty & Camilleri, 2017; MongoDB, n.d.; Neo4j, n.d.; Noworyta, 2018; Pults, Carter, & Condon, 2017; Soloshchuk, 2016; Stern, 2018; Stojkovic, 2012; Tutorialspoint.com, n.d.)

Weather App (Two NoSQL Solutions)

Two NoSQL databases that could be used for a weather app are MongoDB and Riak. MongoDB could be used to store the details for the weather in specific places and Riak could be used to store and access the specific places. For example, the various weather conditions in Honolulu could be stored in a document object in the MongoDB database and the various cities with weather conditions including Honolulu could be stored in the Riak database. The Riak database would use a given key to access the value which would be the city. Then, the city would access its self contained data, the weather conditions.

References:

- Cover Image: Raza, M. (2018, August 29). Introduction to Database Management Systems (DBMS) [Digital image]. Retrieved May 11, 2019, from https://www.bmc.com/blogs/dbms-database-management-systems/
- Arabic, T. (2013, June 29). How To (Hide/Unhide) "Add Friend" For Your Facebook Account? [Digital image]. Retrieved May 13, 2019, from https://www.youtube.com/watch?v=UVt0YwqWjM0
- AWS. (n.d.). What is NoSQL? | Nonrelational Databases, Flexible Schema Data Models | AWS. Retrieved May 12, 2019, from https://aws.amazon.com/nosql/
- Bhardwaj, A. (2016, October). NoSQL Database Overview. Retrieved May 12, 2019, from https://rdbmsknowledge.blogspot.com/2016/10/nosql-database-overview.html
- Bordonado, G. (2014, February 12). Freebie: Twitter 2014 GUI PSD (New profile template) [Digital image]. Retrieved May 12, 2019, from https://dribbble.com/shots/1420640-Freebie-

Twitter-2014-GUI-PSD-New-profile-template#shot-description

- Brazeal, F. (2017, November 18). Why Amazon DynamoDB isn't for everyone. Retrieved May 13, 2019, from https://read.acloud.guru/why-amazon-dynamodb-isnt-for-everyone-and-how-to-decide-when-it-s-for-you-aefc52ea9476
- DashMagazine. (2017, June 7). Top 4 NoSQL DataBases [Infographic] [Digital image]. Retrieved May 13, 2019, from https://hackernoon.com/top-4-nosql-databases-infographic-b6acc389befc
- Facebook. (n.d.). Facebook. Retrieved May 13, 2019, from https://www.facebook.com/
- I. (2016, June 23). What is a Column Store Database? [Digital image]. Retrieved May 12, 2019, from https://database.guide/what-is-a-column-store-database/
- I. (2016, June 23). What is a Column Store Database? Retrieved May 12, 2019, from https://database.guide/what-is-a-column-store-database/
- Knight, M. (2019, March 15). What is a Wide Column Database? Retrieved May 12, 2019, from https://www.dataversity.net/wide-column-database/#
- McNulty, S., & Camilleri, S. C. (2017, September). What are the disadvantages of MongoDB? Retrieved May 13, 2019, from

https://www.quora.com/What-are-the-disadvantages-of-MongoDB

- MongoDB. (2019, April 1). Document Databases. Retrieved May 13, 2019, from https://www.mongodb.com/document-databases
- MongoDB. (n.d.). What Is MongoDB? Retrieved May 13, 2019, from https://www.mongodb.com/what-is-mongodb
- MongoDB. (2019, April 1). What is NoSQL? Retrieved May 12, 2019, from https://www.mongodb.com/nosql-inline
- Namistyuk, T. (n.d.). What is a Key-Value Store? Retrieved May 13, 2019, from https://www.aerospike.com/learn/what-is-a-key-value-store/
- Neo4j. (n.d.). Top 10 Reasons for Choosing Neo4j for Your Graph Database. Retrieved May 13, 2019, from https://neo4j.com/top-ten-reasons/
- Noworyta, T. (2018, July 26). The pros and cons of using MongoDB. Retrieved May 13, 2019, from https://medium.com/selleo/the-pros-and-cons-of-using-mongodb-8e7601857a8e
- Pults, D., Carter, K., & Condon, S. (2017, January 13). Riak vs. Cassandra – A Brief Comparison. Retrieved May 13, 2019, from https://riak.com/posts/business/riak-vs-cassandra-an-updated-brief-comparison/
- Sasaki, B. M. (2018, July 17). Graph Databases for Beginners: Why Graph Technology Is the Future.

Retrieved May 13, 2019, from https://neo4j.com/blog/why-graph-databases-are-the-future/

- Sasaki, B. M. (2018, October 26). Graph Databases for Beginners: Why We Need NoSQL Databases [Digital Image]. Retrieved May 13, 2019, from https://neo4j.com/blog/why-nosql-databases/?ref=blog
- Soloshchuk, V. (2016, July 28). Neo4j Community Edition: Worth Using? Retrieved May 13, 2019, from https://www.linkedin.com/pulse/neo4j-community-edition-worth-using-vasyl-soloshchuk/
- Stern, A. (2018, November 25). Review: "Riak is highly available, highly scalable, and has a fast search index." Retrieved May 13, 2019, from https://www.trustradius.com/products/riak/reviews
- Stojkovic, S. (2012, September). Cassandra: Pros & Cons! Retrieved May 13, 2019, from https://marsmedia.info/en/blog/cassandra-pros-cons-and-model
- Tutorialspoint.com. (n.d.). Cassandra Introduction. Retrieved May 13, 2019, from https://www.tutorialspoint.com/cassandra/cassandra_introduction.htm
- Xplenty. (2017, September 28). The SQL vs NoSQL Difference: MySQL vs MongoDB. Retrieved May 11, 2019, from https://medium.com/xplenty-blog/the-sql-vs-nosql-difference-mysql-vs-mongodb-32c9980e67b2