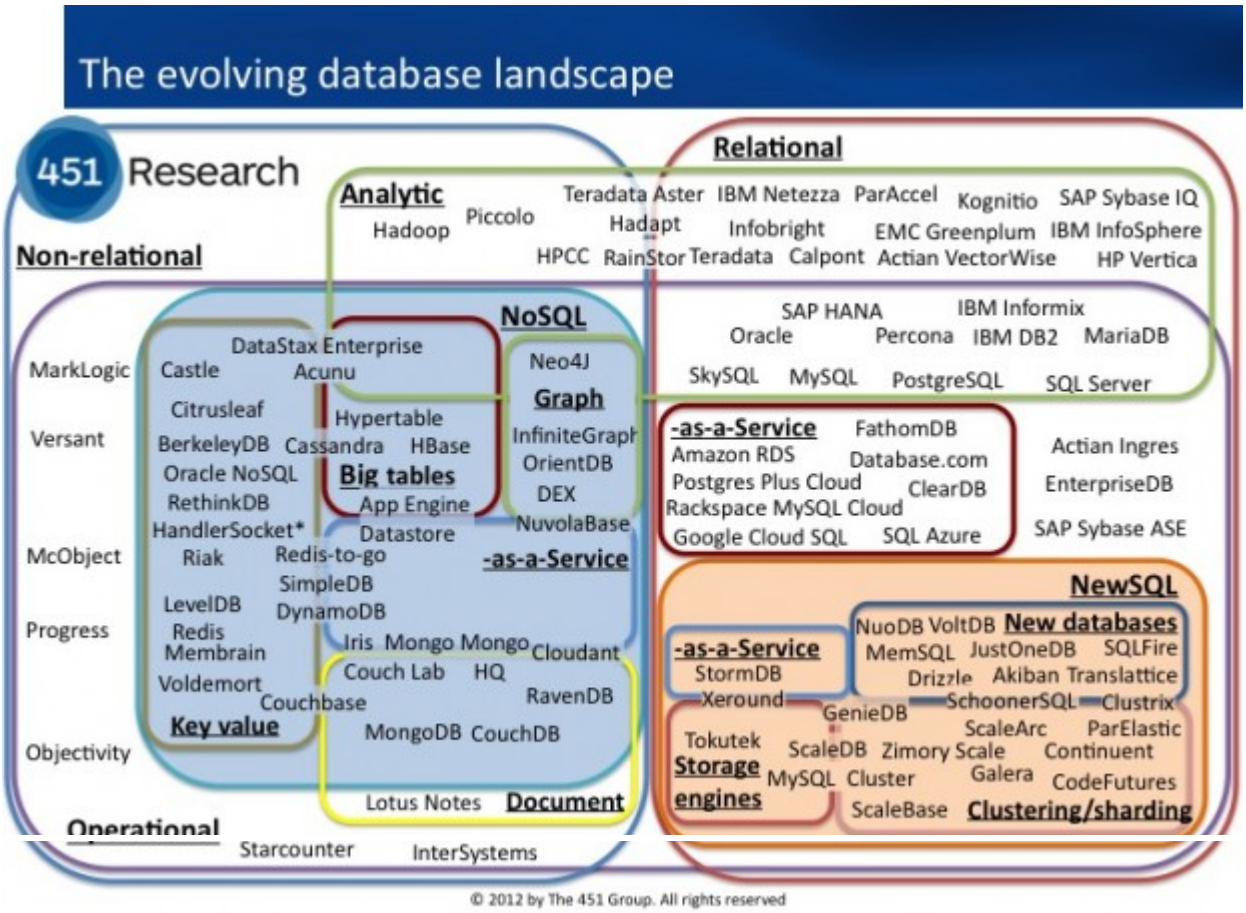
DATACONOMY

DATA SCIENCE 101 · UNDERSTANDING BIG DATA

SQL VS. NOSQL- WHAT YOU NEED TO KNOW

EILEEN MCNULTY · JULY 1, 2014



'SQL is outdated'. 'RDBMS can no longer meet businesses' data management needs'. 'New database technologies like NoSQL are the solution for today's enterprises'. We hear statements like these alot, both inside and outside the database technologies industry. But are they accurate? Is SQL a thing of the past, and are NoSQL solutions the way forward?

In this article, we'll outline the differences between SQL and NoSQL, the vast array of differences within NoSQL technologies themselves, and discuss if Relational Database Management Systems really are a thing of the past.

SQL VS. NOSQL; AN OVERVIEW

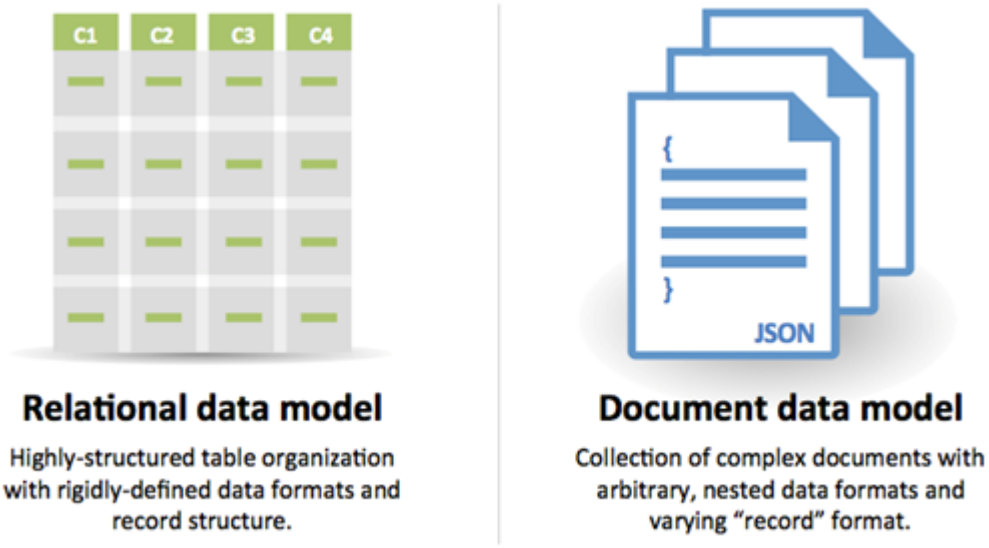
	SQL	NoSQL
Data storage	Stored in a relational model, with rows and columns. Rows contain all of the information about one specific entry/entity, and columns are all the separate data points; for example, you might have a row about a specific car, in which the columns are 'Make', 'Model', 'Colour' and so on.	The term "NoSQL" encompasses a host of databases, each with different data storage models. The main ones are: document, graph, key-value and columnar. More on the distinctions between them below.
Schemas and Flexibility	Each record conforms to fixed schema, meaning the columns must be decided and locked before data entry and each row must contain data for each column. This can be amended, but it involves altering the whole database and going offline.	Schemas are dynamic. Information can be added on the fly, and each 'row' (or equivalent) doesn't have to contain data for each 'column'.
Scalability	Scaling is vertical. In essence, more data means a bigger server, which can get very expensive. It is possible to scale an RDBMS across multiple servers, but this is a difficult and time-consuming process.	Scaling is horizontal, meaning across servers. These multiple servers can be cheap commodity hardware or cloud instances, making it alot more cost-effective than vertical scaling. Many NoSQL technologies also distribute data across servers automatically.
ACID Compliancy (Atomicity, Consistency, Isolation, Durability)	The vast majority of relational databases are ACID compliant.	Varies between technologies, but many NoSQL solutions sacrifice ACID compliancy for performance and scalability

THE MANY FACES OF NOSQL

Having heard the term “NoSQL”, you could be forgiven for thinking all technologies under this umbrella have the same data model. In fact, NoSQL refers to a whole host of technologies, which store and process data in different ways. Some of the main ways include:

Document Databases

This image from Document Database solution CouchDB sums up the distinction between RDBMS and Document Databases pretty well:



Instead of storing data in rows and columns in a table, data is stored in documents, and these documents are grouped together in collections. Each document can have a completely different structure. Document databases include the aforementioned CouchDB and MongoDB.

Key-Value Stores

Data is stored in an associative array of key-value pairs. The key is an attribute name, which is linked to a value. Well-known key value stores include Redis, Voldemort (developed by LinkedIn) and Dynamo (developed by Amazon).

Graph Databases

Used for data whose relations are represented well in a graph. Data is stored in graph structures with nodes (entities), properties (information about the entities) and lines (connections between the entities). Examples of this type of database include Neo4J and InfiniteGraph.

Columnar (or Wide-Column) Databases

Instead of 'tables', in columnar databases you have column families, which are containers for rows. Unlike RDBMS, you don't need to know all of the columns up front, each row doesn't have to have the same number of columns. Columnar databases are best suited to analysing huge datasets- big names include Cassandra and HBase.

SQL VS. NOSQL- WHICH TO USE?

The idea that SQL and NoSQL are in direct opposition and competition with each other is flawed one, not in the least because many companies opt to use them concurrently. As with all of the technologies I've previously discussed, there really isn't a 'one-system-fits-all' approach; choosing the right technology hinges on the use case. If your data needs are changing rapidly, you need high throughput to handle viral growth, or your data is growing fast and you need to be able to scale out quickly and efficiently, maybe NoSQL is for you. But if the data you have isn't changing in structure and you're experiencing moderate, manageable growth, your needs may be best met by SQL technologies. Certainly, SQL is not dead yet.

(Featured image source: InfoQ)

Eileen McNulty-Holmes – Editor

Eileen has five years' experience in journalism and editing for a range of online publications. She has a degree in English Literature from the University of Exeter, and is particularly interested in big data's application in humanities. She is a native of Shropshire, United Kingdom.

Email: eileen@dataconomy.com

TAGS: Cassandra couchbase Dynamo HBase InfiniteGraph mongoDB Neo4J Redis Voldemort Weekly Newsletter

---

PREVIOUS POST

IBM'S CHEF WATSON: USING DATA TO DELIGHT YOUR TASTEBUDS

NEXT POST

BIG DATA PROVING TO BE A REAL CHALLENGE FOR DATA SCIENTISTS

THE AUTHOR