

Our Biggest Sale of the Year
All eBooks & Videos*

*Exclusions apply

\$5SALE



Home ▸ Data News ▸ Different types of NoSQL databases and when to use them

Data News Databases News Insights

Different types of NoSQL databases and when to use them

By Richard Gall · September 10, 2019 · 11:01 am · 1750 · 0

7 min read

Why NoSQL databases?

The popularity of NoSQL databases over the last decade or so has been driven by an explosion of data. Before what's commonly described as 'the big data revolution', relational databases were the norm – these are databases that contain structured data. Structured data can only be structured if it is based on an existing schema that defines the relationships (hence relational) between the data inside the database.

However, with the vast quantities of data that are now available to just about every business with an internet connection, relational databases simply aren't equipped to handle the complexity and scale of large datasets.

Why *not* SQL databases?

This is for a couple of reasons. The defined schemas that are a necessary component of every relational database will not only undermine the richness and integrity of the data you're working with, relational databases are also hard to scale. Relational databases can only scale vertically, not horizontally. That's fine to a certain extent, but when you start getting high volumes of data – such as when millions of people use a web application, for example – things get really slow and you need more processing power. You can do this by upgrading your hardware, but that isn't really sustainable. By scaling out, as you can with NoSQL databases, you can use a distributed network of computers to handle data. That gives you more speed and more flexibility.

This isn't to say that relational and SQL databases have had their day. They still fulfil many use cases. The only difference is that NoSQL can offer a level of far greater power and control for data intensive use cases. Indeed, using a NoSQL database when SQL will do is only going to add more complexity to something that just doesn't need it.

Take your team to the next level with **Packt for Teams**



- [Seven NoSQL Databases in a Week](#)

Different types of NoSQL databases and when to use them

Our Biggest Sale of the Year

\$5SALE

All eBooks & Videos*

*Exclusions apply

Take your team to the next level with **Packt for Teams**



[Find Out More](#)

Must Read in Cloud & Networking



Cloud & Networking News

10 tech startups for 2020 that will help the world build...

Richard Gall · December 30, 2019 · 2:00 pm

0

The Datadog IPO in September marked an important moment for the tech industry. This wasn't just

USE THEM

So, now we've looked at why NoSQL databases have grown in popularity in recent years, let's dig into some of the different options available. There are a huge number of NoSQL databases out there – some of them open source, some premium products – many of them built for very different purposes.

Broadly speaking there are 4 different models of NoSQL databases:

- Key-Value pair-based databases
- Column-based databases
- Document-oriented databases
- Graph databases

Let's take a look at these four models, how they're different from one another, and some examples of the product options in each.

Key Value pair-based NoSQL database management systems

Key/Value pair based NoSQL databases store data in, as you might expect, pairs of keys and values. Data is stored with a matching key – keys have no relation or structure (so, keys could be height, age, hair color, for example).

When should you use a key/value pair-based NoSQL DBMS?

Key/value pair based NoSQL databases are the most basic type of NoSQL database. They're useful for storing fairly basic information, like details about a customer.

Which key/value pair-based DBMS should you use?

There are a number of different key/value pair databases. The most popular is Redis. Redis is incredibly fast and very flexible in terms of the languages and tools it can be used with. It can be used for a wide variety of purposes – one of the reasons high-profile organizations use it, including Verizon, Atlassian, and Samsung. It's also open source with enterprise options available for users with significant requirements.

- [Redis 4.x Cookbook](#)

Other than Redis, other options include Memcached and Ehcache. As well as those, there are a number of other multi-model options (which will crop up later, no doubt) such as Amazon DynamoDB, Microsoft's Cosmos DB, and OrientDB.

- [Hands-On Amazon DynamoDB for Developers \[Video\]](#)
- [RDS PostgreSQL and DynamoDB CRUD: AWS with Python and Boto3 \[Video\]](#)

Column-based NoSQL database management systems

Column-based databases separate data into discrete columns. Instead of using rows – whereby the row ID is the main key – column-based database systems flip things around to make the data the main key.

By using columns you can gain much greater speed when querying data. Although it's true that querying a whole row of data would take longer in a column-based DBMS, the use cases for column based databases mean you probably won't be doing this. Instead you'll be querying a specific part of the data rather than the whole row.

When should you use a column-based NoSQL DBMS?

Column-based systems are most appropriate for big data and instances where data is relatively simple and consistent (they don't particularly handle volatility that well).

Which column-based NoSQL DBMS should you use?

The most popular column-based DBMS is Cassandra. The software prides itself on its performance, boasting 100% availability thanks to lacking a single point of failure, and offering impressive scalability at a good price. Cassandra's popularity speaks for itself – Cassandra is [used by 40% of the Fortune 100](#).

- [Mastering Apache Cassandra 3.x – Third Edition](#)
- [Learn Apache Cassandra in Just 2 Hours \[Video\]](#)

because the company was the fourth tech startup...



Beyond Kubernetes: Key skills for infrastructure and ops engineers in 2020

December 20, 2019 - 7:48 am



New for 2020 in operations and infrastructure engineering

December 19, 2019 - 11:36 am

Must Read in Data



Artificial Intelligence News

Emmanuel Tsukerman on why a malware solution must include a machine...

Savia Lobo - December 30, 2019 - 6:00 am

0

Machine learning is indeed the tech of present times! Security, which is a growing concern for many organizations today and machine learning is one...



Understand Quickbooks online/desktop, online security, use cases, and more with Crystalynn...

December 27, 2019 - 5:30 am



Key skills for data professionals to learn in 2020

December 20, 2019 - 7:08 am

Interviews



There are other options available, such as HBase and Cosmos DB.

- [HBase High Performance Cookbook](#)

Document-oriented NoSQL database management systems

Document-oriented NoSQL systems are very similar to key/value pair database management systems. The only difference is that the value that is paired with a key is stored as a document. Each document is self-contained, which means no schema is required – giving a significant degree of flexibility over the data you have.

For software developers, this is essential – it's for this reason that document-oriented databases such as MongoDB and CouchDB are useful components of the [full-stack](#) development tool chain.

Some search platforms such as [ElasticSearch](#) use mechanisms similar to standard document-oriented systems – so they could be considered part of the same family of database management systems.

When should you use a document-oriented DBMS?

Document-oriented databases can help power many different types of websites and applications – from stores to content systems. However, the flexibility of document-oriented systems means they are not built for complex queries.

Which document-oriented DBMS should you use?

The leader in this space is, MongoDB. With an amazing 40 million downloads (and apparently 30,000 more every single day), it's clear that MongoDB is a cornerstone of the NoSQL database revolution.

- [MongoDB 4 Quick Start Guide](#)
- [MongoDB Administrator's Guide](#)
- [MongoDB Cookbook – Second Edition](#)

There are other options as well as MongoDB – these include CouchDB, CouchBase, DynamoDB and Cosmos DB.

- [Learning Azure Cosmos DB](#)
- [Guide to NoSQL with Azure Cosmos DB](#)

Graph-based NoSQL database management systems

The final type of NoSQL database is graph-based. The notable distinction about graph-based NoSQL databases is that they contain the relationships between different data. Subsequently, graph databases look quite different to any of the other databases above – they store data as nodes, with the 'edges' of the nodes describing their relationship to other nodes.

Graph databases, compared to relational databases, are multidimensional in nature. They display not just basic relationships between tables and data, but more complex and multifaceted ones.

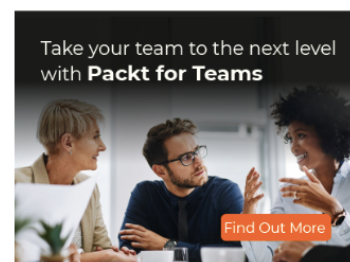
When should you use a graph database?

Because graph databases contain the relationships between a set of data (customers, products, price etc.) they can be used to build and model networks. This makes graph databases extremely useful for applications ranging from [fraud detection](#) to smart homes to search.

Which graph database should you use?

The world's most popular graph database is Neo4j. It's purpose built for data sets that contain strong relationships and connections. Widely used in the industry in companies such as eBay and Walmart, it has established its reputation as one of the world's best NoSQL database products.

Back in 2015 Packt's Data Scientist demonstrated how he used Neo4j to build a graph application. [Read more.](#)



Must Read in Cloud & Networking



- [Learning Neo4j 3.x \[Video\]](#)
- [Exploring Graph Algorithms with Neo4j \[Video\]](#)

NoSQL databases are the future – but know when to use the right one for the job

Although NoSQL databases will remain a fixture in the engineering world, SQL databases will always be around. This is an important point – when it comes to databases, using the right tool for the job is essential.

It's a valuable exercise to explore a range of options and get to know how they work – sometimes the difference might just be a personal preference about usability. And that's fine – you need to be productive after all. But what's ultimately most essential is having a clear sense of what you're trying to accomplish, and choosing the database based on your fundamental needs.



Richard Gall

Co-editor of the Packt Hub. Interested in politics, tech culture, and how software and business are changing each other.



10 tech startups for 2020 that will help the world build...

Richard Gall · December 30, 2019 · 2:00 pm

The Datadog IPO in September marked an important moment for the tech industry. This wasn't just because the company was the fourth tech startup...



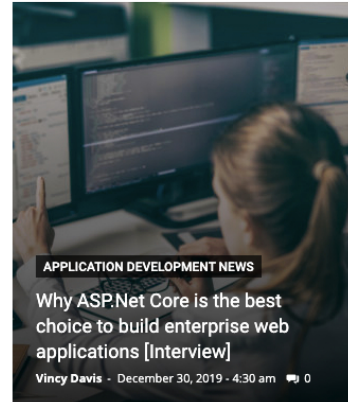
Beyond Kubernetes: Key skills for infrastructure and ops engineers in 2020

December 20, 2019 · 7:48 am



New for 2020 in operations and infrastructure engineering

December 19, 2019 · 11:36 am



APPLICATION DEVELOPMENT NEWS

Why ASP.NET Core is the best choice to build enterprise web applications [Interview]

Vincy Davis · December 30, 2019 · 4:30 am

EDITOR PICKS



10 tech startups for 2020 that will help the world build...

December 30, 2019 · 2:00 pm



Emmanuel Tsukerman on why a malware solution must include a machine...

December 30, 2019 · 6:00 am



Why ASP.NET Core is the best choice to build enterprise web...

December 30, 2019 · 4:30 am

POPULAR POSTS



Customizing Elgg Themes

October 27, 2009 · 12:00 am



Basics of Jupyter Notebook and Python

October 11, 2015 · 12:00 am



How to use arrays, lists, and dictionaries in Unity for 3D...

May 16, 2018 · 9:00 pm

POPULAR CATEGORY

Tutorials	6365
News	3482
Data News	2689
Web Development News	2240
Programming News	1723
Cloud & Networking News	1229
Insights	1130
Artificial Intelligence News	1036
CMS & E-Commerce News	842

ABOUT US



Technology news, analysis, and tutorials from Packt. Stay up to date with what's important in software engineering today.

Contact us: hub@packtpub.com

FOLLOW US

