NoSql vs Relational database [closed]

Asked 14 years, 1 month ago Modified 1 year, 8 months ago

Viewed 117k times

162

Closed. This question is opinion-based. It is not currently accepting answers.

Want to improve this question? Update the question so it can be answered with facts and citations by editing this post.

Closed last year.

The community reviewed whether to reopen this question last year and left it closed:

> Original close reason(s) were not resolved

Improve this question

Recently ***NoSQL*** has gained immense popularity.

What are the advantages of ***NoSQL*** over traditional ***RDBMS***?

database database-design nosql relational-database rdbms

Share

Improve this question

Follow

edited Mar 11, 2016 at 23:23

Levent Divilioglu

11.6k ● 5 ● 60 ● 110

asked Nov 12, 2010 at 0:57

user496949

85.9k ● 149 ● 313 ● 432

2 Possible duplicate of What is NoSQL, how does it work, and what benefits does it provide? – Trevor Boyd Smith Feb 20, 2017 at 15:32

9 Answers

Sorted by: Highest score (default)

125

Not all data is relational. For those situations, NoSQL can be helpful.

With that said, NoSQL stands for "Not Only SQL". It's not intended to knock SQL or supplant it.

SQL has several very big advantages:

1. Strong mathematical basis.
2. Declarative syntax.
3. A well-known language in Structured Query Language (SQL).

Those haven't gone away.

It's a mistake to think about this as an either/or argument. NoSQL is an alternative that people need to consider when it fits, that's all.

Documents can be stored in non-relational databases, like CouchDB.

Maybe reading this will help.

Share Improve this answer Follow

edited Nov 12, 2010 at 1:12

answered Nov 12, 2010 at 0:59

duffymo

308k ● 46 ● 374 ● 565

11 Could you give some examples of non-relational data? – user496949 Nov 12, 2010 at 1:02

8 Documents and images can be stored inside RDBMS too like SQL Server and Oracle? Then why NoSQL? – user496949 Nov 12, 2010 at 1:18

3 Semi-structured data is one such class. It contains XML, Emails, JSON, etc. See the wikipedia page on it. The general rule is that the structure is there, but is loosely defined and dynamically extensible (the latter tend to class with the relational model - and while it is not impossible to model, it is definitely cumbersome). Another class is "natural data": A Novel, An Image, both with no meta-data attached. – I GIVE CRAP ANSWERS Nov 12, 2010 at 1:21

2 Well, you can't do `SELECT blob FROM images WHERE blob CONTAINS('red car')`. So while you can store the data raw in the database, you can't search it without attaching metadata. Full-text-search modules in RDBMS systems bridges some of the semi-structural gap.
– I GIVE CRAP ANSWERS Nov 12, 2010 at 1:24

1 user496949, you keep posing this as either/or. No one said you can't store documents in a SQL database. And if you don't want to use NoSQL, don't. – duffymo Nov 12, 2010 at 2:18

102

The history seem to look like this:

1. Google needs a storage layer for their inverted search index. They figure a traditional RDBMS is not going to cut it. So they implement a NoSQL data store, BigTable on top of their GFS file system. The major part is that thousands of cheap commodity hardware machines provides the speed and the redundancy.
2. Everyone else realizes what Google just did.
3. Brewers CAP theorem is proven. All RDBMS systems of use are CA systems. People begin playing with CP and AP systems as well. K/V stores are vastly simpler, so they are the primary vehicle for the research.
4. Software-as-a-service systems in general do not provide an SQL-like store. Hence, people get more interested in the NoSQL type stores.

I think much of the take-off can be related to this history. Scaling Google took some new ideas at Google and everyone else follows suit because this is the only solution they know to the scaling problem right now. Hence, you are willing to rework everything around the distributed database idea of Google because it is the only way to scale beyond a certain size.

C - Consistency
A - Availability
P - Partition tolerance
K/V - Key/Value

Share Improve this answer

Follow

edited Jun 20, 2020 at 9:12

Community Bot

1 ● 1

answered Nov 12, 2010 at 1:39

I GIVE CRAP ANSWERS

18.8k ● 3 ● 44 ● 47

3 Look up the CAP Theorem on Wikipedia. CA and CP stems from there. K/V is short for Key/Value, a (distributed) finite mapping from keys into values. – I GIVE CRAP ANSWERS Dec 10, 2012 at 16:25

37

NoSQL is better than RDBMS because of the following reasons/properities of NoSQL

1. It supports semi-structured data and volatile data

2. It does not have schema
3. Read/Write throughput is very high
4. Horizontal scalability can be achieved easily
5. Will support Bigdata in volumes of Terra Bytes & Peta Bytes
6. Provides good support for Analytic tools on top of Bigdata
7. Can be hosted in cheaper hardware machines
8. In-memory caching option is available to increase the performance of queries
9. Faster development life cycles for developers

EDIT:

To answer "why RDBMS cannot scale", please take a look at RDBMS Overheads pdf written by Stavros Harizopoulos,Daniel J. Abadi,Samuel Madden and Michael Stonebraker

RDBMS's have challenges in handling huge data volumes of Terabytes & Peta bytes. Even if you have Redundant Array of Independent/Inexpensive Disks (RAID) & data shredding, it does not scale well for huge volume of data. You require very expensive hardware.

Source for below 4 points (Stavros Harizopoulos, Daniel J. Abadi, Samuel Madden , Michael Stonebraker from Report)

> ***Logging****:* Assembling log records and tracking down all changes in database structures slows performance.

> ***Locking***: Traditional two-phase locking poses a sizeable overhead since all accesses to database structures are governed by a separate entity, the Lock Manager.

> ***Latching***: In a multi-threaded database, many data structures have to be latched before they can be accessed. Removing this feature and going to a single-threaded approach has a noticeable performance impact.

> ***Buffer management***: A main memory database system does not need to access pages through a buffer pool, eliminating a level of indirection on every record access.

This does not mean that we have to use NoSQL over SQL.

Still, RDBMS is better than NoSQL for the following reasons/properties of RDBMS

1. Transactions with **ACID** properties - Atomicity, Consistency, Isolation & Durability

2. **Adherence to Strong Schema** of data being written/read
3. **Real time query management** (in case of data size < 10 Tera bytes)
4. Execution of complex queries involving **join & group by clauses**

We have to use RDBMS (SQL) and NoSQL (Not only SQL) depending on the business case & requirements

Share Improve this answer Follow

edited Mar 31, 2023 at 18:29

answered Aug 12, 2015 at 14:44

Ravindra babu

38.9k ● 11 ● 256 ● 219

5 It's worth noting that some NoSQL databases support ACID transactions. – Dave Cassel May 14, 2018 at 17:12

17

NOSQL has no special advantages over the relational database model. NOSQL does address certain limitations of current SQL DBMSs but it doesn't imply any fundamentally new capabilities over previous data models.

NOSQL means only no *SQL* (or "not only SQL") but that doesn't mean the same as no *relational*. A relational database in principle would make a very good NOSQL

solution - it's just that none of the current set of NOSQL products uses the relational model.

Share Improve this answer

Follow

edited May 23, 2013 at 3:17

Teodor Talov

1,943 ● 2 ● 25 ● 39

answered Nov 12, 2010 at 6:45

nvogel

25.5k ● 3 ● 46 ● 87

4 It seems that at the recent O'Reilly Strata Conference, Mark Madsen has coined a new interpretation of "NoSQL" in his history of databases in no-tation to supersede "Not Only SQL". It is now: "No, SQL" ;-) – Lukas Eder Dec 14, 2013 at 20:11

4

RDBMS focus more on relationship and *NoSQL* focus more on storage.

You can consider using *NoSQL* when your *RDBMS* reaches bottlenecks. *NoSQL* makes *RDBMS* more flexible.

Share Improve this answer

Follow

edited May 29, 2017 at 17:13

Ravindra babu

38.9k ● 11 ● 256 ● 219

answered Dec 17, 2016 at 21:39

suiwenfeng

1,993 ● 2 ● 25 ● 32

4

The biggest advantage of NoSQL over RDBMS is **Scalability**.
NoSQL databases can easily scale-out to many nodes, but for RDBMS it is very hard.
Scalability not only gives you more storage space but also much higher performance since many hosts work at the same time.

Share Improve this answer

Follow

edited Sep 1, 2019 at 21:57

Ahmed Nabil
18.9k ● 12 ● 68 ● 95

answered Oct 11, 2017 at 19:19

Jon
81 ● 4

3

If you need to process huge amount of data with high performance

OR

If data model is not predetermined

then

NoSQL database is a better choice.

Share Improve this answer

Follow

edited May 17, 2017 at 15:16

Ravindra babu
38.9k ● 11 ● 256 ● 219

answered May 24, 2015 at 16:51

Chris
1,742 ● 2 ● 17 ● 21

3

Just adding to all the information given above

NoSql Advantages:

1) NoSQL is good if you want to be production ready fast due to its support for schema-less and object oriented architecture.

2) NoSql db's are eventually consistent which in simple language means they will not provide any lock on the data(documents) as in case of RDBMS and what does it mean is latest snapshot of data is always available and thus increase the latency of your application.

3) It uses MVCC (Multi view concurrency control) strategy for maintaining and creating snapshot of data(documents).

4) If you want to have indexed data you can create view which will automatically index the data by the view definition you provide.

NoSql Disadvantages:

1) Its definitely not suitable for big heavy transactional applications as it is eventually consistent and does not support ACID properties.

2) Also it creates multiple snapshots (revisions) of your data (documents) as it uses MVCC methodology for concurrency control, as a result of which space get consumed faster than before which makes compaction and hence reindexing more frequent and it will slow down your application response as the data and transaction in your application grows. To counter that you can horizontally scale the nodes but then again it will be higher cost as compare sql database.

Share Improve this answer

Follow

edited Nov 8, 2018 at 8:11

Community Bot

1 • 1

answered Oct 31, 2018 at 10:00

Manvendra Jina

117 • 5

2

From mongodb.com:

> NoSQL databases differ from older, relational technology in four main areas:
>
> **Data models**: A NoSQL database lets you build an application without having to define the schema first unlike relational databases which make you define your schema before you can add any data to the system. No predefined schema makes NoSQL databases much easier to update as your data and requirements change.

> **Data structure**: Relational databases were built in an era where data was fairly structured and clearly defined by their relationships. NoSQL databases are designed to handle unstructured data (e.g., texts, social media posts, video, email) which makes up much of the data that exists today.
>
> **Scaling**: It's much cheaper to scale a NoSQL database than a relational database because you can add capacity by scaling out over cheap, commodity servers. Relational databases, on the other hand, require a single server to host your entire database. To scale, you need to buy a bigger, more expensive server.
>
> **Development model**: NoSQL databases are open source whereas relational databases typically are closed source with licensing fees baked into the use of their software. With NoSQL, you can get started on a project without any heavy investments in software fees upfront.

Share Improve this answer

Follow

edited Nov 22, 2018 at 3:42

Pang

10.1k ● 146 ● 85 ● 124

answered May 12, 2018 at 18:20

deadpool

326 ● 1 ● 5 ● 12