

# NoSQL Yes, But YesCQL, No?

**Berlin Buzzwords  
June 7, 2011**

**Eric Evans**  
**[eevans@rackspace.com](mailto:eevans@rackspace.com)**  
**@jericevans**  
**<http://blog.sym-link.com>**



# Cassandra Query Language

- Structured query language for Apache Cassandra.
- CQL for short (pronounced /si:kwəl/).
- SQL alike (best effort).
- An alternative to the existing API, not a replacement (not yet).
- Available for use in Cassandra 0.8.0.



**Wait, aren't you the guy...?**





WIKIPEDIA  
The Free Encyclopedia

- Main page
- Contents
- Featured content
- Current events
- Random article
- Donate to Wikipedia

- Interaction
  - Help
  - About Wikipedia
  - Community portal
  - Recent changes
  - Contact Wikipedia

- Toolbox
- Print/export

- Languages
  - Deutsch
  - Español
  - Français
  - Italiano
  - Português
  - Русский
  - Türkçe

Article Discussion

Read Edit View history Search

## NoSQL

From Wikipedia, the free encyclopedia  
(Redirected from Nosql)

*This article is about the class of database management systems. For the specific relational database management software, see [NoSQL \(RDBMS\)](#).*

In computing, **NoSQL** is a broad class of database management systems that differ from classic relational database management systems (RDBMSes) in some significant ways: they may not require fixed table schemas, and usually avoid join operations and typically scale horizontally. Academics and papers typically refer to these database systems as **storage**,<sup>[1][2][3][4]</sup> a term that would include classic relational databases as a subset.

### Contents [hide]

- 1 History
- 2 Architecture
- 3 Taxonomy
  - 3.1 Document store
  - 3.2 Graph
  - 3.3 Key-value store
    - 3.3.1 Eventually-consistent key-value store
    - 3.3.2 Hierarchical key-value store
  - 3.4 Hosted services
    - 3.4.1 Key-value cache in RAM
    - 3.4.2 Key-value stores implementing the Paxos algorithm
    - 3.4.3 Key-value stores on disk
  - 3.5 Multivalued databases
  - 3.6 Object database
    - 3.6.1 Ordered key-value store



## History

Carlo Strozzi used the term "NoSQL" to describe a database that "departs from the relational model altogether".

Eric Evans, a Rackspace employee, reinforced the term in a 2005 article describing an open-source relational database that did not expose an SQL interface.<sup>[5]</sup> (Strozzi suggests that, in fact, it should have been called more appropriately 'NoREL', or something to that effect.)<sup>[6]</sup>

The term was popularized in early 2009 when Johan Oskarsson of Last.fm wanted to organize an event to discuss open-source databases. The event was titled "NoSQL: The new attributes of classic relational databases".

The "no:sql(east)" conference 2009 was a significant influence on the NoSQL debate. Its self-conception was "a conference of non-relational data stores", and thus, the most common interpretation of "NoSQL" is "non-relational", although NoSQL is not meant as anti-RDBMS, but rather as a complement to relational databases, and Graph Databases.<sup>[citation needed]</sup>

Architecture



**So, is this a troll?**



**Gonna Give You Up.**



**Never.**



**Naw.**  
**(not a troll, honest)**



**But, why?**





**Because the API sucks.**



# “Thrift sucks, ergo the API sucks”

- Generated code (C++ compiler).
- Loads of languages, but varying levels of support.
  - PHP anyone?
- Upstream alternating between extremes of combativeness and apathy.
  - Patches ignored, (or refused).
  - Loads of (serious )bugs ignored for long periods.
  - Infrequent releases.



# “Avro Does Not Suck, so...”

1. Avro
2. Something, something, something
3. Profit!



**And the API *still* sucks.**



# Brass Tacks

- Unstable
  - Too tightly coupled to internal APIs
- Too difficult to use
  - Very little abstraction (forces clients to abstract)
  - Poor mental fit for query/data models



# Back to the drawing board

- RPC (Thrift, Avro, Protobuf, etc)
- REST
- Query language
- Etc, etc



# Back to the drawing board

- RPC (Thrift, Avro, Protobuf, etc)
  - Easy to implement
  - Performant
- REST
- Query language
- Etc, etc



# Back to the drawing board

- RPC (Thrift, Avro, Protobuf, etc)
  - Easy to implement
  - Performant
- REST
  - Little need for client abstraction
- Query language
- Etc, etc





# Back to the drawing board

- RPC (Thrift, Avro, Protobuf, etc)
  - Easy to implement
  - Performant
- REST
  - Little need for client abstraction
- Query language
  - Little need for client abstraction
  - Reads well; What you see is what you get
  - The Devil we all know
- Etc, etc



# Grok This

```
firstname = Column(name="firstname", value="Eric", timestamp=time)
firstcsc = ColumnOrSuperColumn(column=firstname)
lastname = Column(name="lastname", value="Evans", timestamp=time)
lastcsc = ColumnOrSuperColumn(column=lastname)

mutations = []
mutations.append(Mutation(column_or_supercolumn=firstcsc))
mutations.append(Mutation(column_or_supercolumn=lastcsc))

client.batch_mutate(mutation_map={"eevans": {"table": mutations}},
                    consistency_level=ConsistencyLevel.ONE)
```



# What about this?

```
UPDATE table  
SET firstname=Eric, lastname=Evans  
WHERE KEY=eevans
```



# Grok This

```
parent = ColumnParent(column_family="table")
colnames = ["firstname", "lastname"]
predicate = SlicePredicate(column_names=colnames)
row = client.get_slice(key="eevans",
                      column_parent=parent,
                      predicate=predicate,
                      consistency_level=CL.ONE)
```



# And this?

```
SELECT firstname, lastname  
FROM table  
WHERE KEY = eevans
```



# Official(?) Drivers

- Java (JDBC)
- Python (DBAPI2)
- Node.js
- Twisted
- PHP (coming soon?)
- Ruby (coming soon?)



# More Info

- Docs ([doc/cql/CQL.html](http://doc/cql/CQL.html))
- [http://www.datastax.com/docs/0.8/api/using\\_cql](http://www.datastax.com/docs/0.8/api/using_cql)
- <http://caqel.deadcafe.org> (live demo!)
- cqlsh (interactive shell shipped w/ Python driver)



