



# A Comparison of SQL and NoSQL Databases

Slides from: Keith W. Hare  
Metadata Open Forum

More reading: <http://martinfowler.com/articles/nosqlKeyPoints.html>



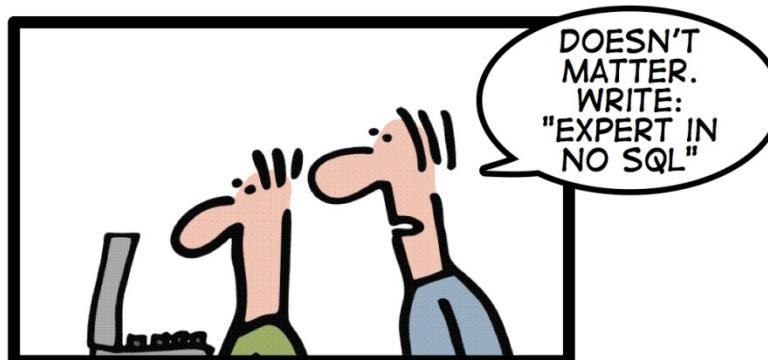
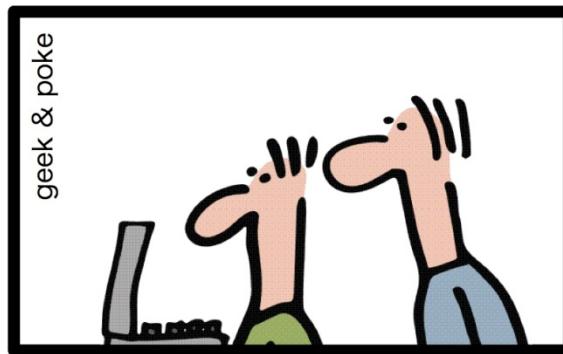
# Abstract

**NoSQL databases (either no-SQL or Not Only SQL)** are currently a hot topic in some parts of computing. In fact, one website lists over a hundred different NoSQL databases.

This presentation reviews the features common to the NoSQL databases and compares those features to the features and capabilities of SQL databases.

# BIG DATA!

# HOW TO WRITE A CV



# +

## SQL Characteristics

- Data stored in rows, columns and tables
- Relationships represented by data
- Data Manipulation Language
- Data Definition Language
- Transactions
- Abstraction from physical layer



# SQL Physical Layer Abstraction

- Applications specify what, not how
- Query optimization engine
- Physical layer can change without modifying applications
  - Create indexes to support queries
  - In Memory databases



# Data Manipulation Language (DML)

- Data manipulated with Select, Insert, Update, & Delete statements
  - Select T1.Column1, T2.Column2 ...  
From Table1, Table2 ...  
Where T1.Column1 = T2.Column1 ...
- Data Aggregation
- Compound statements
- Functions and Procedures
- Explicit transaction control



# Data Definition Language

- Schema defined at the start
- Create Table (Column1 Datatype1, Column2 Datatype 2, ...)
- Constraints to define and enforce relationships
  - Primary Key
  - Foreign Key
  - Etc.
- Triggers to respond to Insert, Update , & Delete
- Stored Modules
- Alter ...
- Drop ...
- Security and Access Control



# Transactions – ACID Properties

- Atomic – All of the work in a transaction completes (commit) or none of it completes
- Consistent – A transaction transforms the database from one consistent state to another consistent state. Consistency is defined in terms of constraints.
- Isolated – The results of any changes made during a transaction are not visible until the transaction has committed.
- Durable – The results of a committed transaction survive failures



# NewSQL: real-time analytics

- 1) SQL as the primary mechanism for application interaction
- 2) ACID support for transactions
- 3) A non-locking concurrency control mechanism so real-time reads will not conflict with writes, and thereby cause them to stall.
- 4) An architecture providing much higher per-node performance than available from the traditional "elephants"
- 5) A scale-out, shared-nothing architecture, capable of running on a large number of nodes without bottlenecking



# NoSQL Definition

From [www.nosql-database.org](http://www.nosql-database.org):

**Next Generation Databases mostly addressing some of the points: being non-relational, distributed, open-source and horizontal scalable. The original intention has been modern web-scale databases. The movement began early 2009 and is growing rapidly. Often more characteristics apply as: schema-free, easy replication support, simple API, eventually consistent / BASE (not ACID), a huge data amount, and more.**



# NoSQL Products/Projects

<http://www.nosql-database.org/> lists 122

## NoSQL Databases

- Cassandra
- CouchDB
- Hadoop & Hbase
- MongoDB
- StupidDB
- Etc.



# NoSQL Products/Projects

<http://www.nosql-database.org/> lists 122 Databases

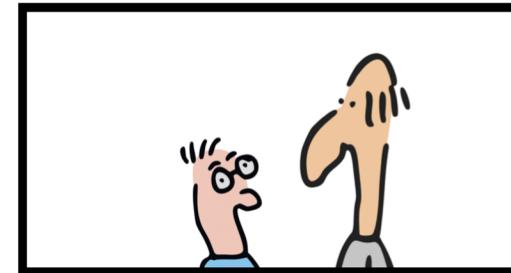
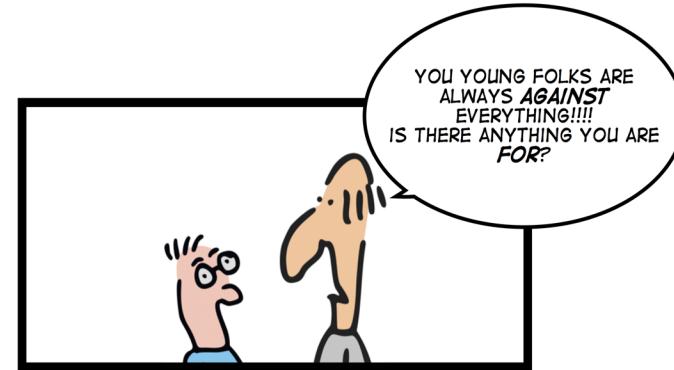
■ Cassandra

■ CouchDB

■ Hadoop & Hbase

■ MongoDB

■ StupidDB





# NoSQL Distinguishing Characteristics

- Large data volumes
  - Google's "big data"
- Scalable replication and distribution
  - Potentially thousands of machines
  - Potentially distributed around the world
- Queries need to return answers quickly
- Mostly query, few updates
- Asynchronous Inserts & Updates
- Schema-less
- ACID transaction properties are not needed – BASE
- CAP Theorem
- Open source development



# BASE Transactions

- Acronym contrived to be the opposite of ACID
  - Basically Available,
  - Soft state,
  - Eventually Consistent
- Characteristics
  - Weak consistency – stale data OK
  - Availability first
  - Best effort
  - Approximate answers OK
  - Aggressive (optimistic)
  - Simpler and faster



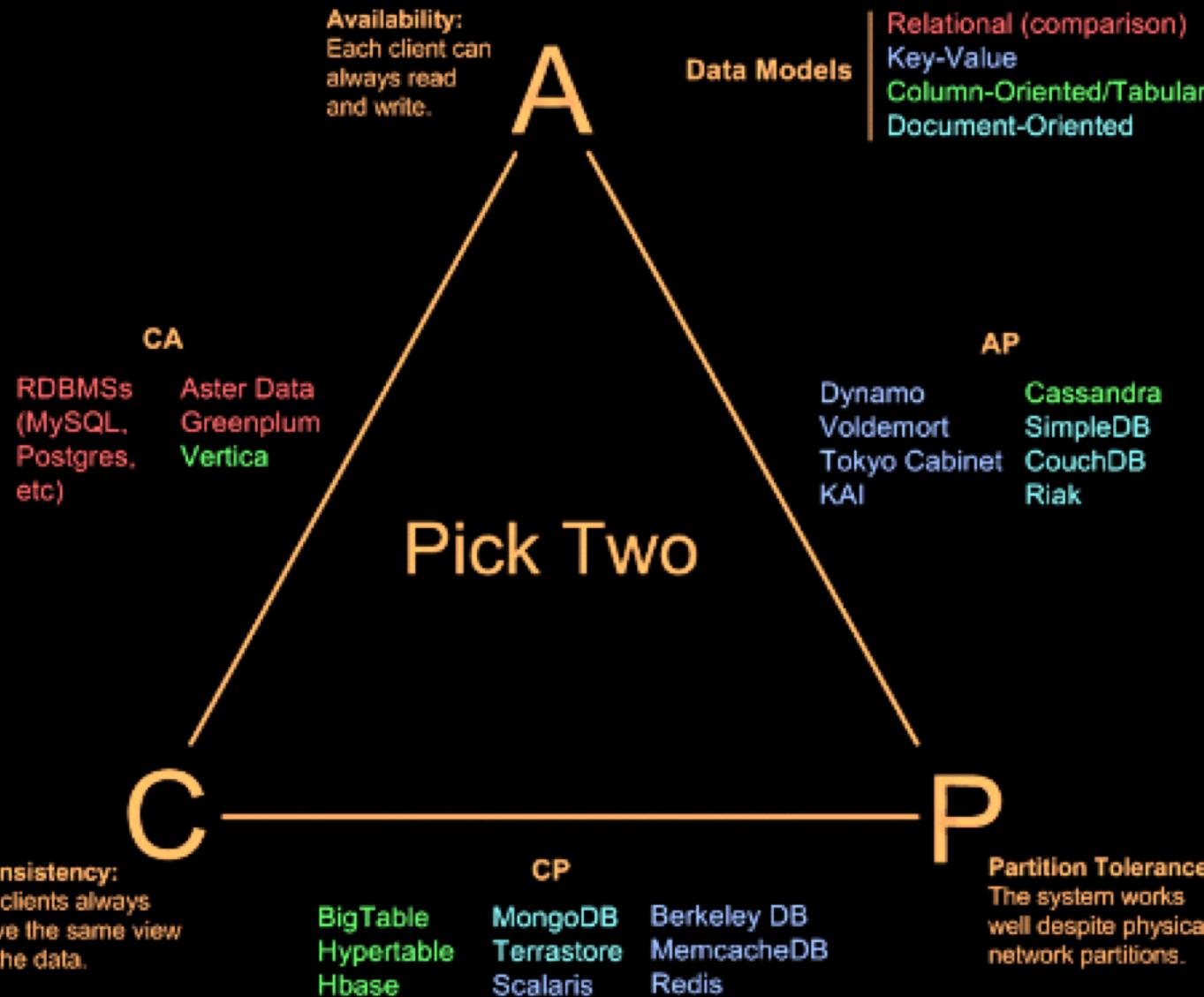
## Brewer's CAP Theorem

**A distributed system can support only two of the following characteristics:**

- **Consistency**
- **Availability**
- **Partition tolerance**

# Visual Guide to NoSQL Systems

+





# NoSQL Database Types

**Discussing NoSQL databases is complicated because there are a variety of types:**

- **Column Store – Each storage block contains data from only one column**
- **Document Store – stores documents made up of tagged elements**
- **Key-Value Store – Hash table of keys**



# NoSQL Example: Column Store

- Each storage block contains data from only one column
- Example: Hadoop/Hbase
  - <http://hadoop.apache.org/>
  - Yahoo, Facebook
- Example: Ingres VectorWise
  - Column Store integrated with an SQL database
  - <http://www.ingres.com/products/vectorwise>



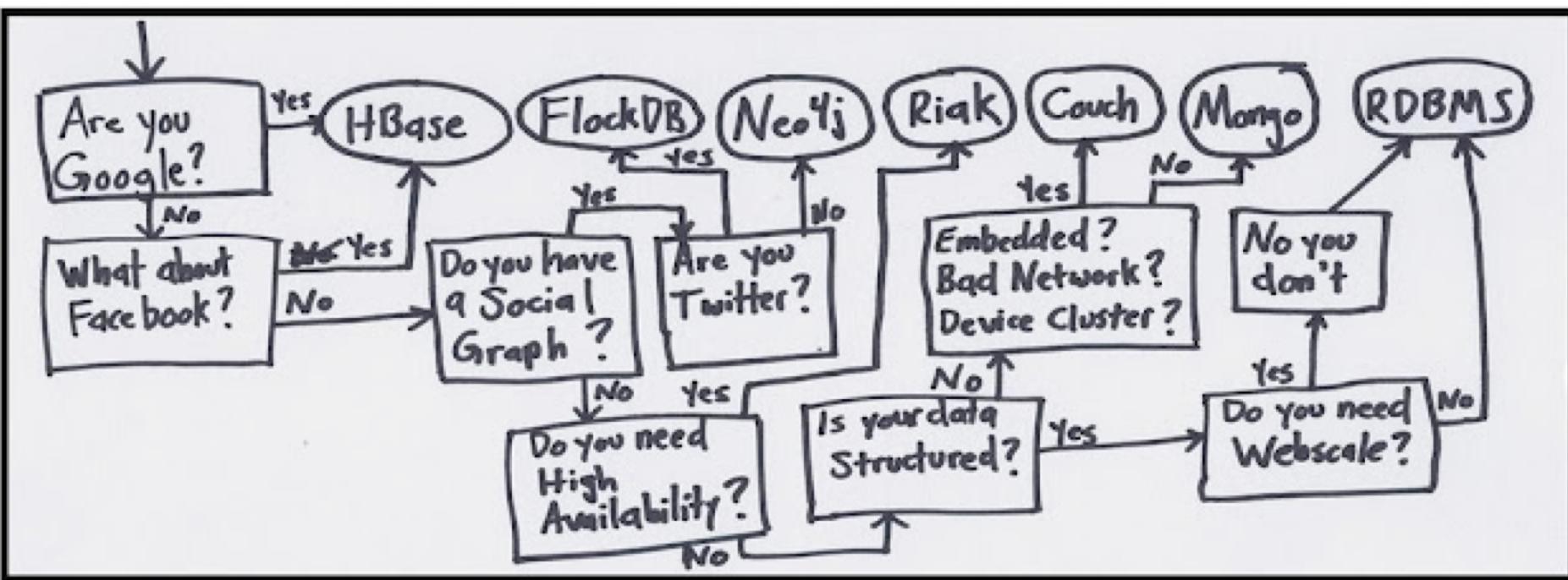
## Column Store Comments

- More efficient than row (or document) store if:
  - Multiple row/record/documents are inserted at the same time so updates of column blocks can be aggregated
  - Retrievals access only some of the columns in a row/record/document



## Other Non-SQL Databases

- XML Databases
- Graph Databases
- Codasyl Databases
- Object Oriented Databases
- Etc...
- Will not address these today





Key-Value



Ordered Key-Value



Big Table



Document, Full-Text Search



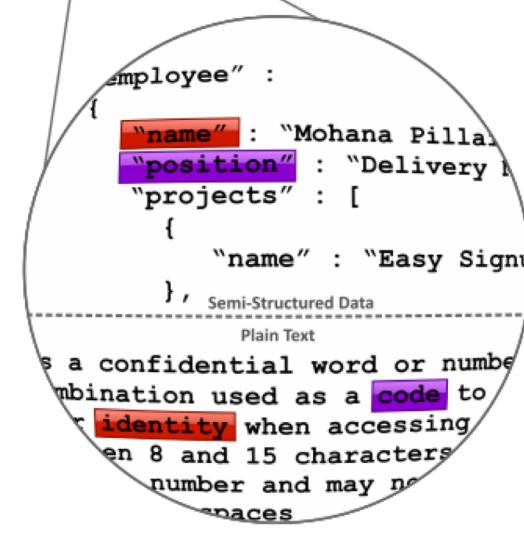
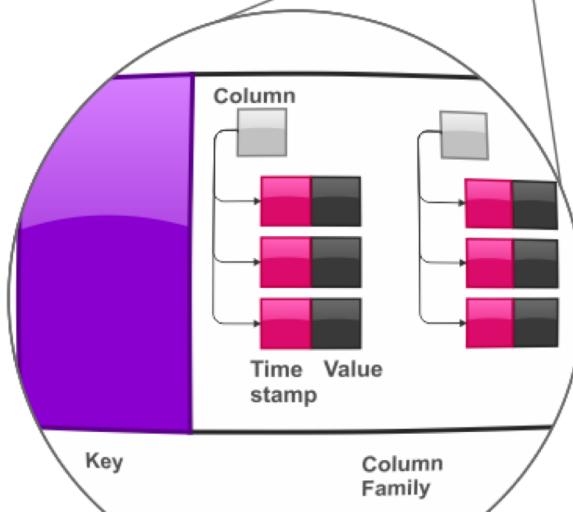
Graph



SQL



Key Value





# Storing and Modifying Data

- Syntax varies
  - HTML
  - Java Script
  - Etc.
- Asynchronous – Inserts and updates do not wait for confirmation
- Versioned
- Optimistic Concurrency



# Retrieving Data

- **Syntax Varies**

- No set-based query language
- Procedural program languages such as Java, C, etc.

- Application specifies retrieval path

- No query optimizer

- Quick answer is important

- May not be a single “right” answer



## Open Source

- Small upfront software costs
- Suitable for large scale distribution on commodity hardware



# NoSQL Summary

- NoSQL databases reject:
  - Overhead of ACID transactions
  - “Complexity” of SQL
  - Burden of up-front schema design
  - Declarative query expression
  - Yesterday’s technology
- Programmer responsible for
  - Step-by-step procedural language
  - Navigating access path



# Summary

## ■ SQL Databases

- Predefined Schema
- Standard definition and interface language
- Tight consistency
- Well defined semantics

## ■ NoSQL Database

- No predefined Schema
- Per-product definition and interface language
- Getting an answer quickly is more important than getting a correct answer



# Web References

- “NoSQL -- Your Ultimate Guide to the Non - Relational Universe!”  
<http://nosql-database.org/links.html>
- “NoSQL (RDBMS)”  
<http://en.wikipedia.org/wiki/NoSQL>
- PODC Keynote, July 19, 2000. *Towards Robust. Distributed Systems*. Dr. Eric A. Brewer. Professor, UC Berkeley. Co-Founder & Chief Scientist, Inktomi.  
[www.eecs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf](http://www.eecs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf)
- “Brewer's CAP Theorem” posted by Julian Browne, January 11, 2009. <http://www.julianbrowne.com/article/viewer/brewers-cap-theorem>
- “How to write a CV” Geek & Poke Cartoon  
<http://geekandpoke.typepad.com/geekandpoke/2011/01/nosql.html>



# Web References

- “Exploring CouchDB: A document-oriented database for Web applications”, Joe Lennon, Software developer, Core International.  
<http://www.ibm.com/developerworksopensource/library/os-couchdb/index.html>
- “Graph Databases, NOSQL and Neo4j” Posted by Peter Neubauer on May 12, 2010 at:  
<http://www.infoq.com/articles/graph-nosql-neo4j>
- “Cassandra vs MongoDB vs CouchDB vs Redis vs Riak vs HBase comparison”, Kristof Kovacs.  
<http://kkovacs.eu/cassandra-vs-mongodb-vs-couchdb-vs-redis>
- “Distinguishing Two Major Types of Column-Stores”  
Posted by Daniel Abadi on March 29, 2010  
[http://dbmsmusings.blogspot.com/2010/03/distinguishing-two-major-types-of\\_29.html](http://dbmsmusings.blogspot.com/2010/03/distinguishing-two-major-types-of_29.html)



# Web References

- “MapReduce: Simplified Data Processing on Large Clusters”, Jeffrey Dean and Sanjay Ghemawat, December 2004.  
<http://labs.google.com/papers/mapreduce.html>
- “Scalable SQL”, ACM Queue, Michael Rys, April 19, 2011  
<http://queue.acm.org/detail.cfm?id=1971597>
- “a practical guide to noSQL”, Posted by Denise Miura on March 17, 2011 at  
<http://blogs.marklogic.com/2011/03/17/a-practical-guide-to-nosql/>