CIB Session 12th
NoSQL Databases Structures

By: Shahab Safaee & Morteza Zahedi
Software Engineering PhD
Email: safaee.shx@gmail.com , morteza.zahedi.a@gmail.com


cibtrc.ir


cibtrc


cibtrc

Agenda

- What is NoSQL?
- BASE Transactions
- NoSQL Types
- Redis System Properties
- Hbase System Properties
- Cassandra System Properties
- MongoDB System Properties
- Jeo4j System Properties
- Some of Important Statistics
- NoSQL vs. SQL Summery

What is NoSQL?

- Stands for **N**ot **O**nly **SQL**
  - Term was redefined by Eric Evans after Carlo Strozzi.
- Class of non-relational data storage systems
- Usually do not require a fixed table schema nor do they use the concept of joins
- All NoSQL offerings relax one or more of the ACID properties (Based on the CAP theorem)

NoSQL Definition

From www.nosql-database.org:

- Next Generation Databases mostly addressing some of the points:
  - being **non-relational**
  - **distributed**
  - **open-source**
  - **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.

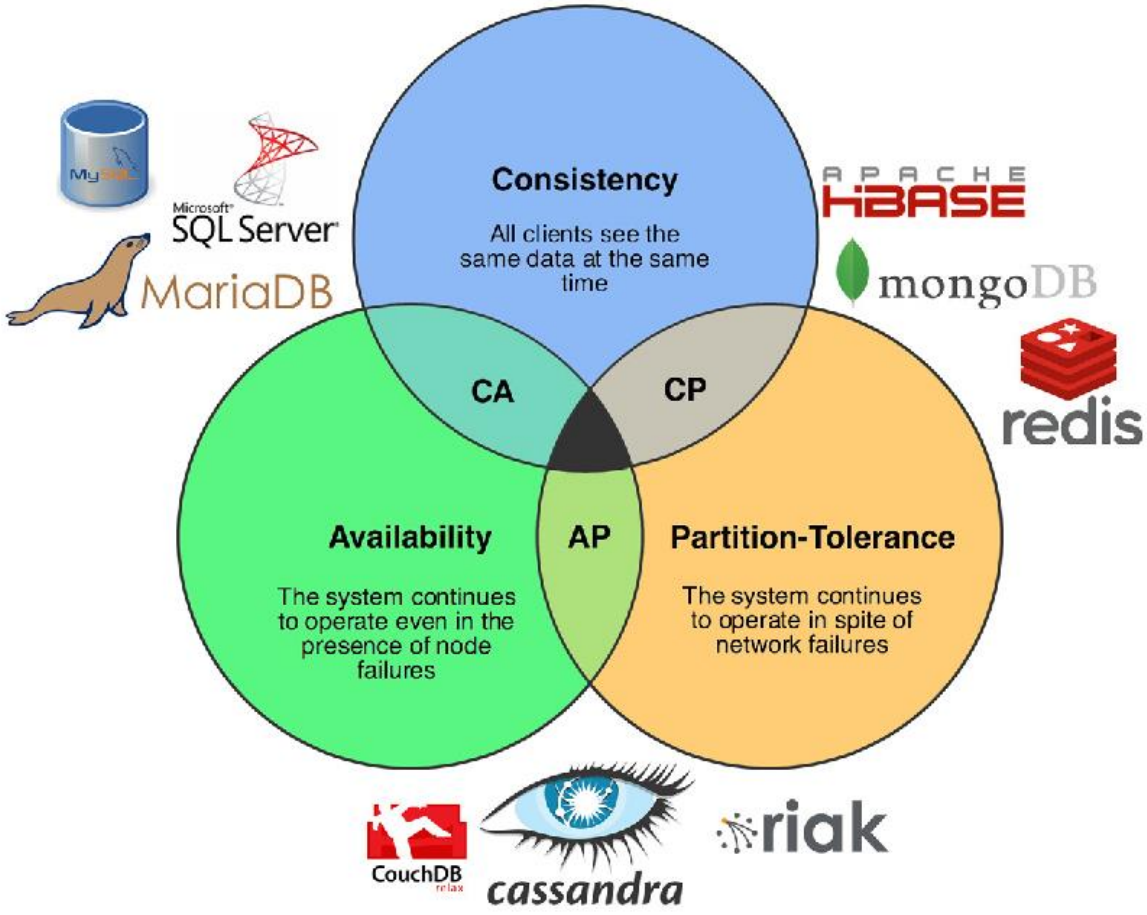
BASE Transactions

- Acronym contrived to be the opposite of ACID
  - **B**asically **A**vailable
    - The database appears to work most of the time (Replication and Sharding Mechanisms).
    - Part of system failure is OK, But total system failure is not Ok.
  - **S**oft state
    - Consistency guaranty with Application Developer.
  - **E**ventually Consistent
    - In ACID, Enforce Consistency and guarantee Consistency of Transaction.
    - In Eventually Consistency, Currently accept transaction then in next time Consistence.
      - No Guarantee Consistency in any time, but Guarantee in next time.
    - Guaranties consistency only at undefended future time.
- Characteristics
  - Weak consistency
  - Availability first
  - Optimistic
  - Simpler and faster

BASE vs ACID

- ACID:
  - Strong Consistency
  - Less Availability
  - Pessimistic Concurrency
  - Complex
- BASE
  - Availability is the most important thing.
  - Weaker consistency (Eventual)
  - Simple and Fast
  - Optimistic

CAP Theorem with ACID and BASE Visualized



CAP theorem with databases that "choose" CA, CP and AP

NoSQL Types

No SQL database are classified into four types:

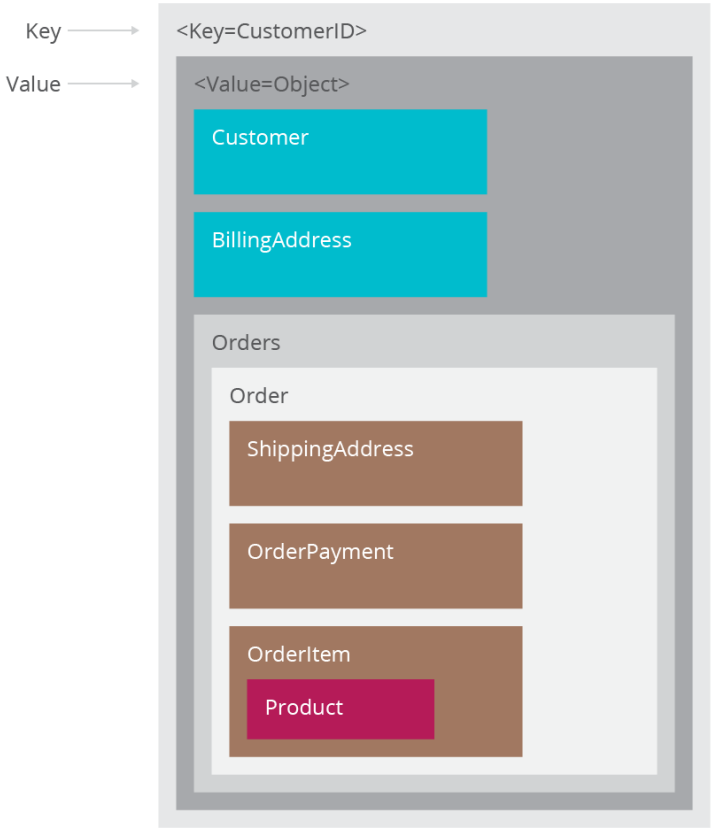
- Key Value pair based
- Column based
- Document based
- Graph based

Motivated by column storage model, eventual consistency, distributed cache system

According to the way of data storage, this type of databases can be classified into 3 categories

- Temporary
- permanent
- hybrid

Key Value Pair Based



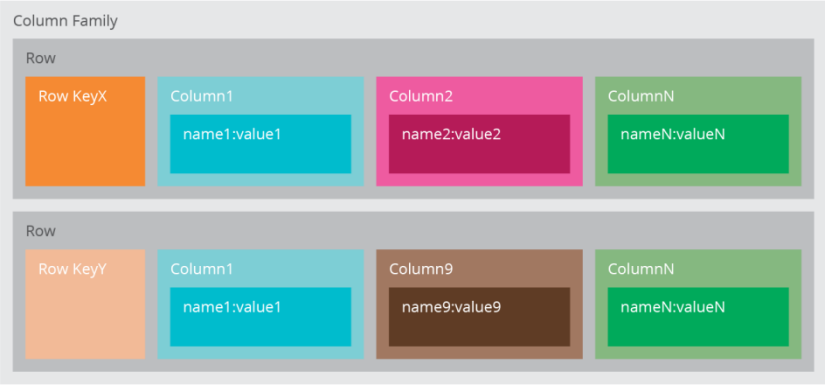
- Designed for processing dictionary. Dictionaries contain a collection of records having fields containing data.
- Records are stored and retrieved using a key that uniquely identifies the record, and is used to quickly find the data with in the database.

Example:

CouchDB, Oracle NoSQL Database, Riak etc.

- We use it for storing session information, user profiles, shopping cart data, Telecom directories.
- We would avoid it when we need to query data having relationships between entities.

Column based



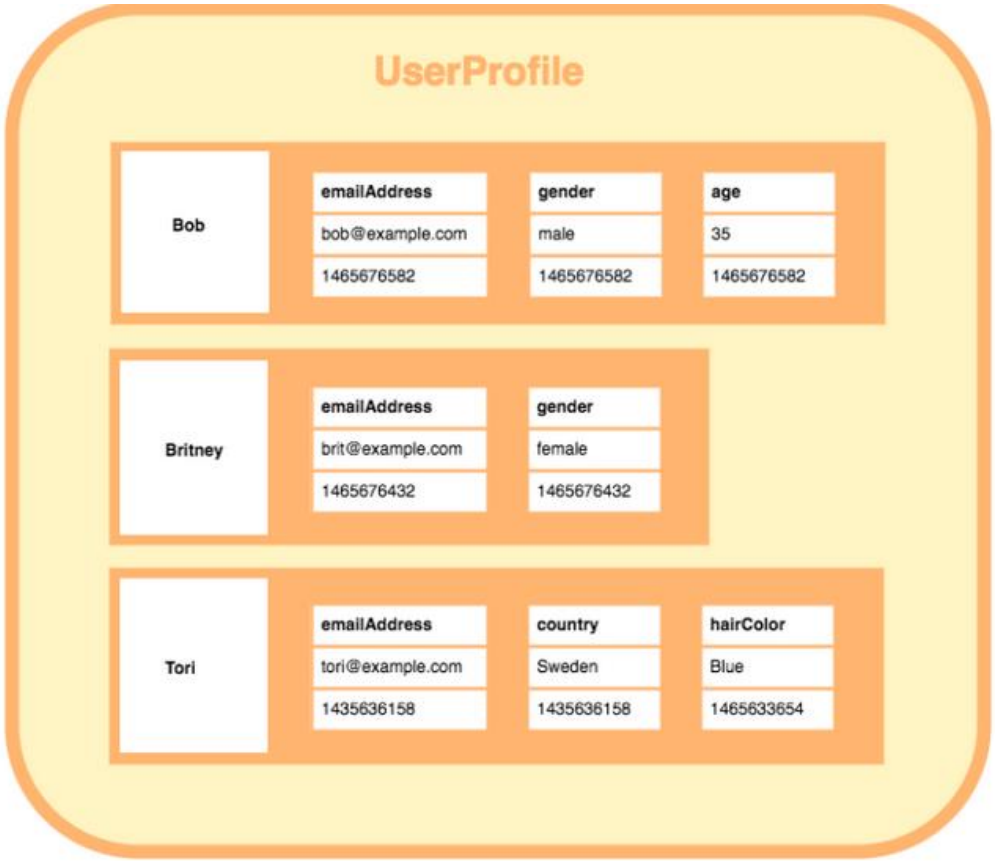
- It store data as Column families containing rows that have many columns associated with a row key. Each row can have different columns.
- Column families are groups of related data that is accessed together.

Example:

Cassandra, HBase, Hypertable, and Amazon DynamoDB.

- We use it for content management systems, blogging platforms, log aggregation.
- We would avoid it for systems that are in early development, changing query patterns.

Column based (Example)



the column-oriented database can be used for the storage of the batch program to update the massive amount of data.

Benefits of Column Store Databases

- **Compression**. Column stores are very efficient at data compression and/or partitioning.
- **Aggregation** queries. Due to their structure, columnar databases perform particularly well with aggregation queries (such as SUM, COUNT, AVG, etc).
- **Scalability**. Columnar databases are very scalable. They are well suited to massively parallel processing (MPP), which involves having data spread across a large cluster of machines – often thousands of machines.
- **Fast to load and query**. Columnar stores can be loaded extremely fast. A billion row table could be loaded within a few seconds. You can start querying and analysing almost immediately.

Document Based

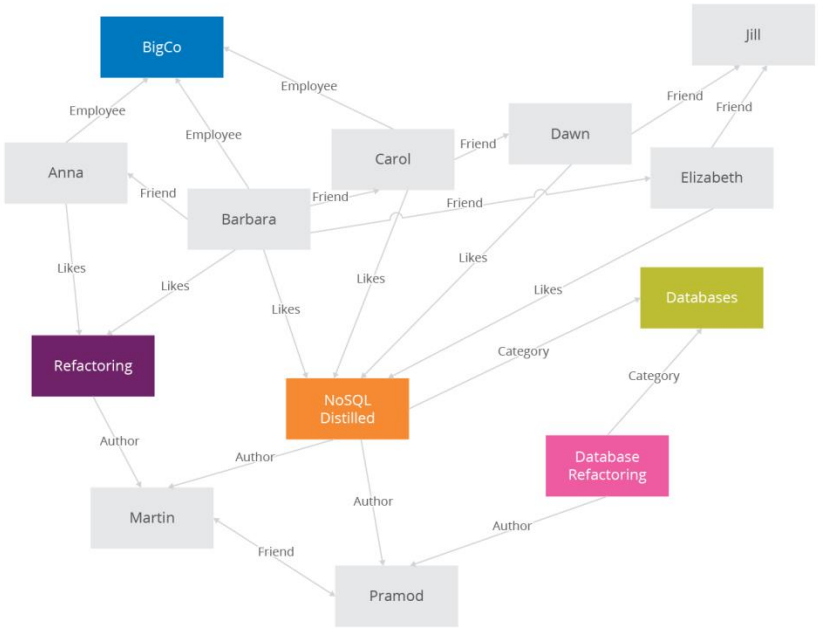
- The database stores and retrieves documents. It stores documents in the value part of the key-value store.
- Self-describing, hierarchical tree data structures consisting of maps, collections, and scalar values.

Example:

LotusNotes, MongoDB, CouchDB, OrientDB, RavenDB.

- We use it for content management systems, blogging platforms, web analytics, real-time analytics, e-commerce applications.
- We would avoid it for systems that need complex transactions spanning multiple operations or queries against varying aggregate structures.

Graph Based



- Store entities and relationships between these entities as nodes and edges of a graph respectively. Entities have properties.
- Traversing the relationships is very fast as relationship between nodes is not calculated at query time but is actually persisted as a relationship.

Example:

Neo4J, Infinite Graph, OrientDB, FlockDB.

- It is well suited for connected data, such as social networks, spatial data, routing information for goods and supply.

Top 10 of NoSQL DB with Data Models

	Data Model	Query API
Cassandra	Columnfamily	Thrift
CouchDB	Document	map/reduce views
HBase	Columnfamily	Thrift, REST
MongoDB	Document	Cursor
Neo4J	Graph	Graph
Redis	Collection	Collection
Riak	Key/value	REST
Scalaris	Key/value	get/put
Tokyo Cabinet	Key/value	get/put
Voldemort	Key/value	get/put

Common Advantages

- Cheap, easy to implement (open source)
- Data are replicated to multiple nodes (therefore identical and fault-tolerant) and can be partitioned
  - Down nodes easily replaced
  - No single point of failure
- Easy to distribute
- Don't require a schema
- Can scale up and down
- Relax the data consistency requirement (CAP)

What is not provided by NoSQL

- Joins
- Group by
- ACID transactions
- SQL
- Integration with applications that are based on SQL

Some Statistics

- Facebook Search
- MySQL > 50 GB Data
  - Writes Average : ~300 ms
  - Reads Average : ~350 ms
- Rewritten with Cassandra > 50 GB Data
  - Writes Average : 0.12 ms
  - Reads Average : 15 ms

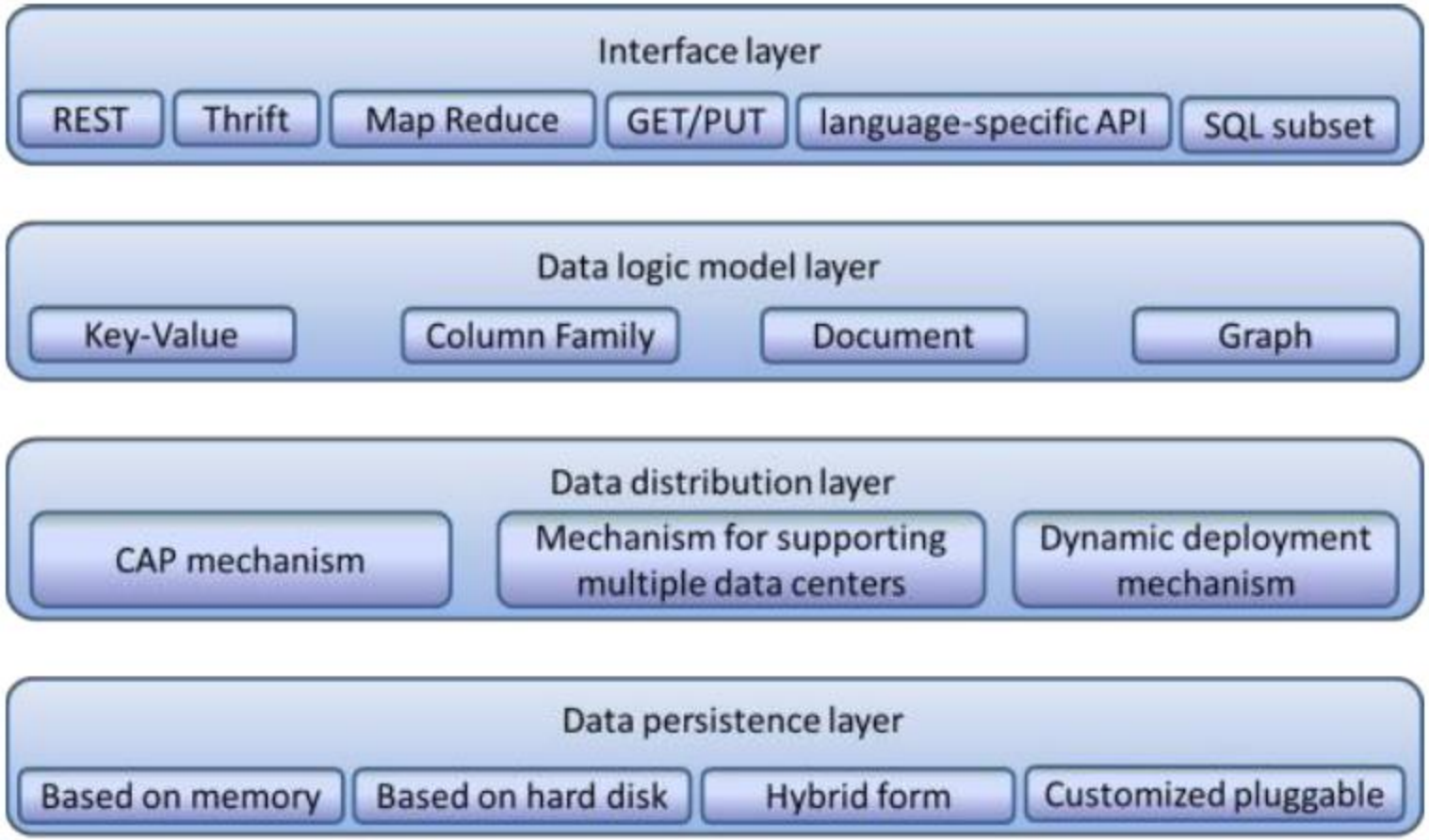
Don’t forget about the DBA

- It does not matter if the data is deployed on a NoSQL platform instead of an RDBMS.
- Still need to address:
  - Backups & recovery
  - Capacity planning
  - Performance monitoring
  - Data integration
  - Tuning & optimization
- What happens when things don’t work as expected and nodes are out of sync or you have a data corruption occurring at 2am?
- Who you gonna call?
  - DBA and SysAdmin need to be on board

NoSQL vs. SQL Summery

	SQL Database	NoSQL Database
Types	One type (SQL database) with minor variations	Many different types including key-value stores, document databases, wide-column stores, and graph databases
Development History	Developed in 1970s to deal with first wave of data storage applications.	Developed in 2000s to deal with limitations of SQL databases, particularly concerning scale, replication and unstructured data storage.
Examples	MySQL, Postgres, Oracle Database	MongoDB, Cassandra, HBase, Neo4j ,Riak, Voldemort, CouchDB ,DynamoDB
Schemas	Structure and data types are fixed in advance.	Typically dynamic. Records can add new information on the fly, and unlike SQL table
Scaling	Vertically	Horizontally
Data Manipulation	Specific language using Select, Insert, and Update statements, e.g. SELECT fields FROM table WHERE...	Through object-oriented APIs

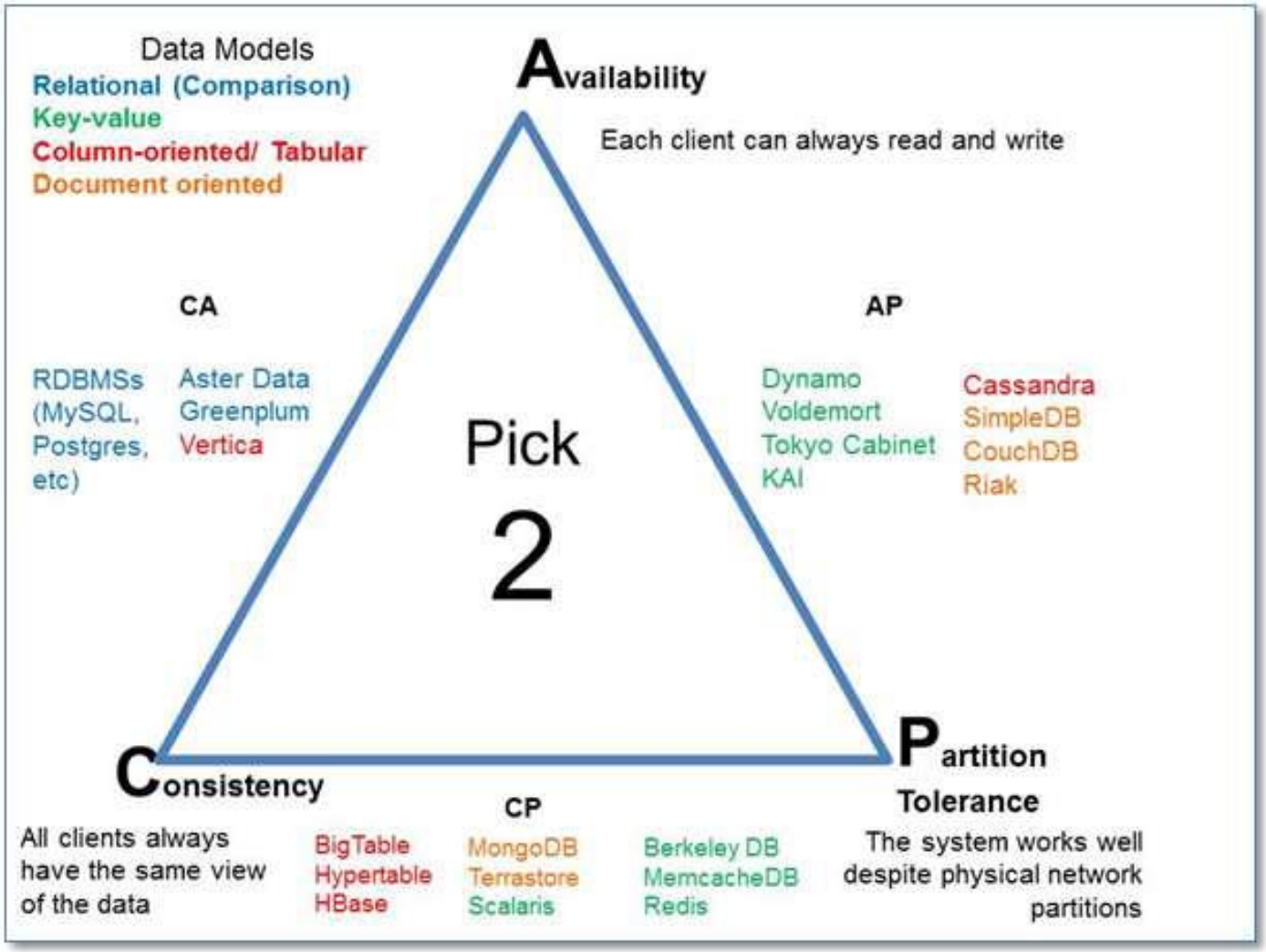
NOSQL FRAMEWORK



NoSQL vs. SQL Summery Features

Feature	NoSQL Databases	Relational Databases
Performance	High	Low
Reliability	Poor	Good
Availability	Good	Good
Consistency	Poor	Good
Data Storage	Optimized for huge data	Medium sized to large
Scalability	High	High (but more expensive)

Visual Guide to NoSQL Systems



Redis System Properties (1)

- Description
  - In-memory data structure store
- database model
  - Key-Value Store
- Initial release
  - 2009
- Current release
  - 5.0.1, November 2018
- License
  - Open Source
- Implementation language
  - C
- Server operating systems
  - BSD, Linux, OS X and Windows

Redis System Properties (2)

- Data scheme
  - Schema-free
- Typing
  - Strings, Hash, Lists, Sets and …
- XML support
  - No
- Secondary indexes
  - Yes
- SQL
  - Not Support
- Supported programming languages
  - C, C++, C#, Java, PHP, Perl, R, Python, Scala and …
- Triggers
  - No

Redis System Properties (3)

- Partitioning method
  - Sharding
- Replication methods
  - Replication
- MapReduce
  - Not Support
- Consistency concepts
  - Eventual Consistency
- SQL
  - No
- Foreign keys
  - No
- Concurrency
  - Yes

Redis System Properties (4)

- Durability
  - Yes
- In-memory capabilities
  - Yes
- User concepts
  - Simple password-based access control
- Cloud-based only
  - No
- Website
  - redis.io
- Developer
  - Salvatore Sanfilippo

HBase System Properties (1)

- Description
  - Wide-column store based on Apache Hadoop and on concepts of BigTable
- database model
  - Column families Store
- Initial release
  - 2008
- Current release
  - 1.4.3, April 2018
- License
  - Open Source
- Implementation language
  - Java
- Server operating systems
  - Linux and Unix

HBase System Properties (2)

- Data scheme
  - Schema-free
- Typing
  - No
- XML support
  - No
- Secondary indexes
  - No
- SQL
  - Not Support
- Supported programming languages
  - C, C++, C#, Java, Perl, PHP, Python, Scala and ...
- Triggers
  - Yes

HBase System Properties (3)

- Partitioning method
  - Sharding
- Replication methods
  - Selectable replication factor
- MapReduce
  - Yes
- Consistency concepts
  - Immediate Consistency
- Foreign keys
  - No
- Concurrency
  - Yes
- Durability
  - Yes

HBase System Properties (4)

- In-memory capabilities
  - No
- User concepts
  - Access Control Lists (ACL)
    - Implementation based on Hadoop and ZooKeeper
- Cloud-based only
  - No
- Website
  - hbase.apache.org
- Developer
  - Apache Software Foundation

Cassandra System Properties (1)

- Description
  - Wide-column store based on ideas of BigTable and DynamoDB
- database model
  - Column families Store
- Initial release
  - 2008
- Current release
  - 3.11.3, August 2018
- License
  - Open Source
- Implementation language
  - Java
- Server operating systems
  - BSD, Linux and Windows

Cassandra System Properties (2)

- Data scheme
  - Schema-free
- Typing
  - Yes
- XML support
  - No
- Secondary indexes
  - Restricted (Only Equality Queries)
- SQL
  - SQL-like SELECT, DML and DDL statements (CQL)
- Supported programming languages
  - C, C++, C#, Java, Perl, PHP, Python, Scala and …
- Triggers
  - Yes

Cassandra System Properties (3)

- Partitioning method
  - Sharding
- Replication methods
  - Selectable replication factor
- MapReduce
  - Yes
- Consistency concepts
  - Eventual Consistency
- Foreign keys
  - No
- Concurrency
  - Yes
- Durability
  - Yes

Cassandra System Properties (4)

- In-memory capabilities
  - No
- User concepts
  - Access rights for users can be defined per object
- Cloud-based only
  - No
- Website
  - cassandra.apache.org
- Developer
  - Apache Software Foundation

MongoDB System Properties (1)

- Description
  - One of the most popular document stores
- Database model
  - Document Store
- Initial release
  - 2009
- Current release
  - 4.0.3, October 2018
- License
  - Open Source
- Implementation language
  - C++
- Server operating systems
  - Solaris, Linux and Windows

MongoDB System Properties (2)

- Data scheme
  - Schema-free
- Typing
  - Yes (String, Integer, Double, Decimal)
- Secondary indexes
  - Yes
- SQL
  - Read-only SQL queries
- Supported programming languages
  - C, C++, C#, Java, Perl, PHP, R, Python, Scala and ...
- Triggers
  - No

MongoDB System Properties (3)

- Partitioning method
  - Sharding
- Replication methods
  - Replication
- MapReduce
  - Yes
- Consistency concepts
  - Eventual Consistency
- Foreign keys
  - No
- Concurrency
  - Yes
- Durability
  - Yes

MongoDB System Properties (4)

- In-memory capabilities
  - Yes
- User concepts
  - Access rights for users and roles
- Cloud-based only
  - No
- Website
  - www.mongodb.com
- Developer
  - MongoDB, Inc

Neo4j System Properties (1)

- Description
  - Open source graph database
- database model
  - Graph DBMS
- Initial release
  - 2007
- Current release
  - 3.4.9, October 2018
- License
  - Open Source
- Implementation language
  - Java, Scala
- Server operating systems
  - Solaris, Linux and Windows

Neo4j System Properties (2)

- Data scheme
  - schema-free and schema-optional
- Typing
  - Yes
- Secondary indexes
  - Yes
- SQL
  - No
- Supported programming languages
  - .Net, Java, Perl, PHP, Python, Scala and …
- Triggers
  - Yes

Neo4j System Properties (3)

- Partitioning method
  - None
- Replication methods
  - Restricted
- MapReduce
  - No
- Consistency concepts
  - Eventual Consistency
- Foreign keys
  - Yes
- Concurrency
  - Yes
- Durability
  - Yes

Neo4j System Properties (4)

- User concepts
  - Users, roles and permissions
  - Pluggable authentication with supported standards (LDAP, Active Directory, Kerberos)
- Cloud-based only
  - No
- Website
  - neo4j.com
- Developer
  - Neo4j, Inc.

Most popular DBMS

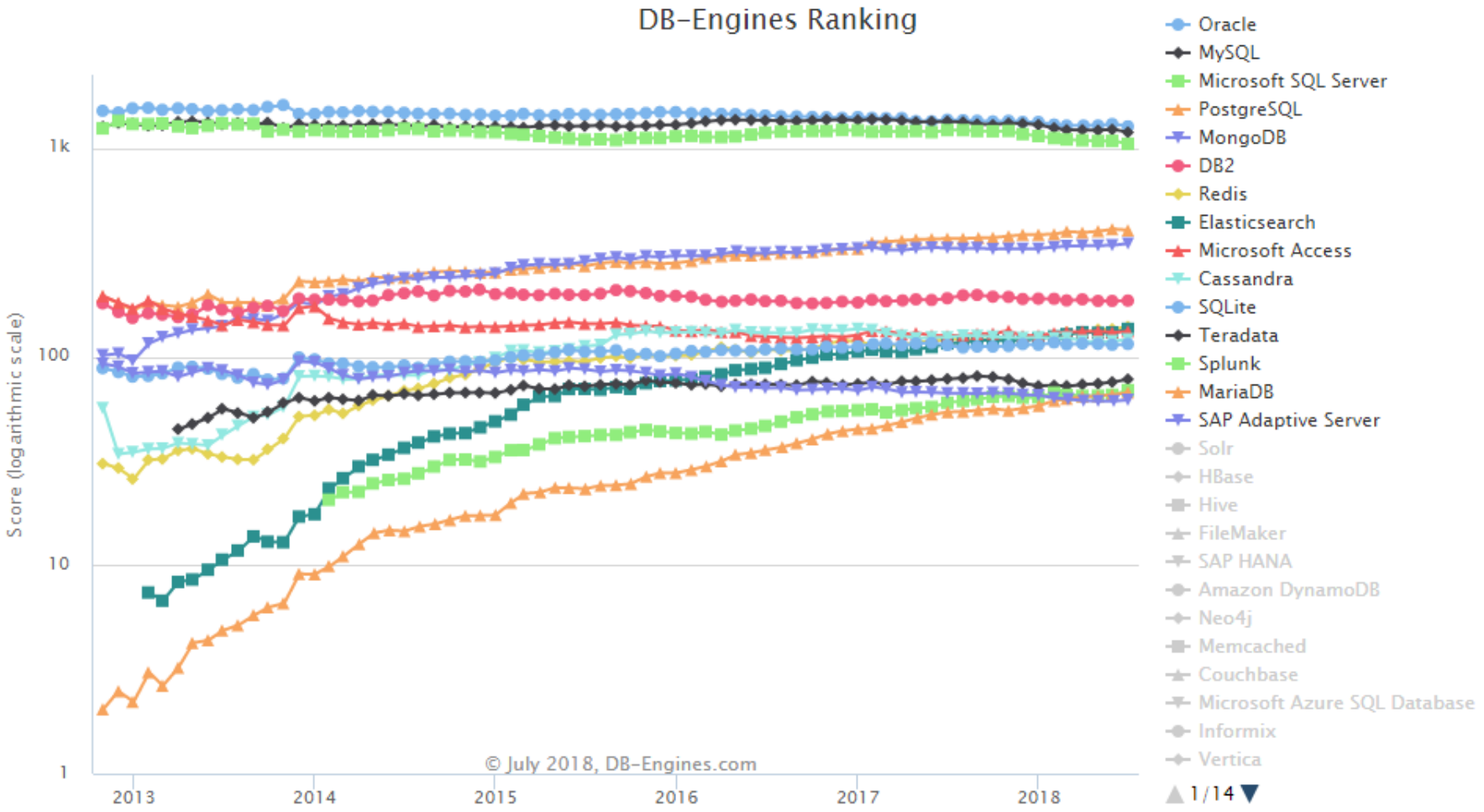
348 systems in ranking, November 2018

Rank			DBMS	Database Model	Score		
Nov 2018	Oct 2018	Nov 2017			Nov 2018	Oct 2018	Nov 2017
1.	1.	1.	Oracle	Relational DBMS	1301.11	-18.16	-58.94
2.	2.	2.	MySQL	Relational DBMS	1159.89	-18.22	-162.14
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1051.55	-6.78	-163.53
4.	4.	4.	PostgreSQL	Relational DBMS	440.24	+20.85	+60.33
5.	5.	5.	MongoDB	Document store	369.48	+6.30	+39.01
6.	6.	6.	DB2	Relational DBMS	179.87	+0.19	-14.19
7.	7.	↑ 9.	Redis	Key-value store	144.17	-1.12	+22.99
8.	8.	↑ 10.	Elasticsearch	Search engine	143.46	+1.13	+24.05
9.	9.	↓ 7.	Microsoft Access	Relational DBMS	138.44	+1.64	+5.12
10.	↑ 11.	↑ 11.	SQLite	Relational DBMS	122.71	+5.96	+9.95

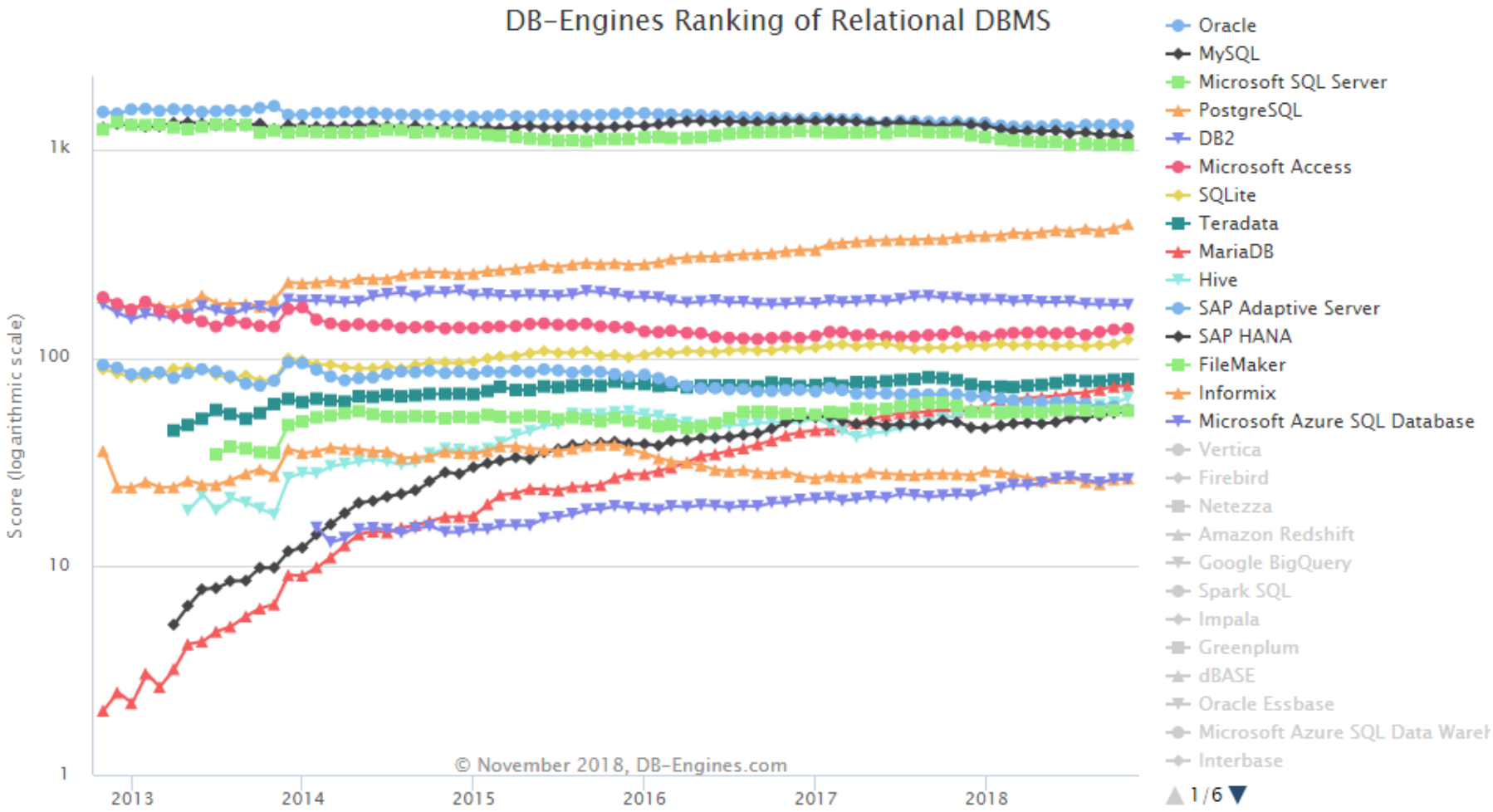
NoSQL DB Catagories Example

Name	Manufacturers	Language	Platform	License	No definitionSchema	Extensible
Column-oriented orderly storage						
HBase	Hadoop	Java	Java	Apache 2.0	Y	Y
Azure Tables	Microsoft	.Net	Azure	SaaS	Y	Y
Cassandra	Apache	Java	Java	Apache 2.0	Y	Y
Hypertable	Open source	C++	Linux/Mac	GPLv2	Y	Y
SimpleDB	Amazon	Erlang	EC2	SaaS	Y	Y
Document-oriented data storage system						
MongoDB	Open source	C++	Linux/Mac/Windows	Friendly AGPL	Y	Y
CouchDB	Open source	Erlang	Linux	Apahce 2.0	Y	N
Terrastore	Open source	Java	Java	Apache 2.0	Y	Y
Key/value storage						
Redis	Open source	C	Linux	BSD	Y	Y
LevelDB	Open source	C	Linux	BSD	Y	Y
Tokyo Cabinet/Tyrant	Open source	C	Linux/Windows	LGPL	Y	N
Berkeley DB	Open source	C	Linux/Mac/Windows	Sleepycat License	Y	Y
MemchachedDB	Open source	C	Linux/Mac/Windows	BSD	Y	Y
Graphic data management system						
Neo4J	Neo Technologies	Java	Java	AGPL/Commercial	Y	N
InfoGrid	NetMesh Inc	Java	Java	AGPL/Commercial	N	N
HyperGraphDB	Kobrix	Java	Java	LGPL	N	N

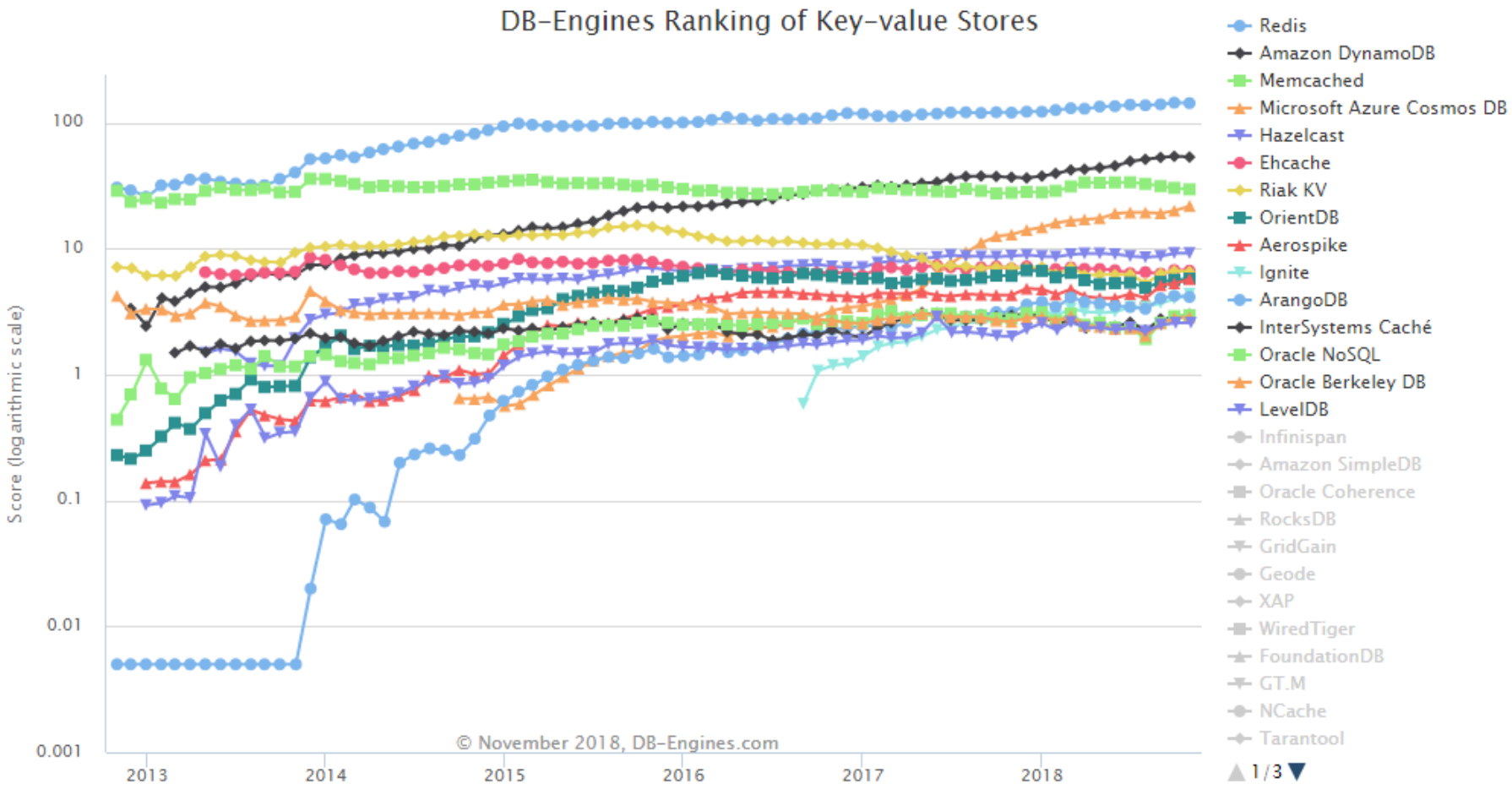
Trend Popularity



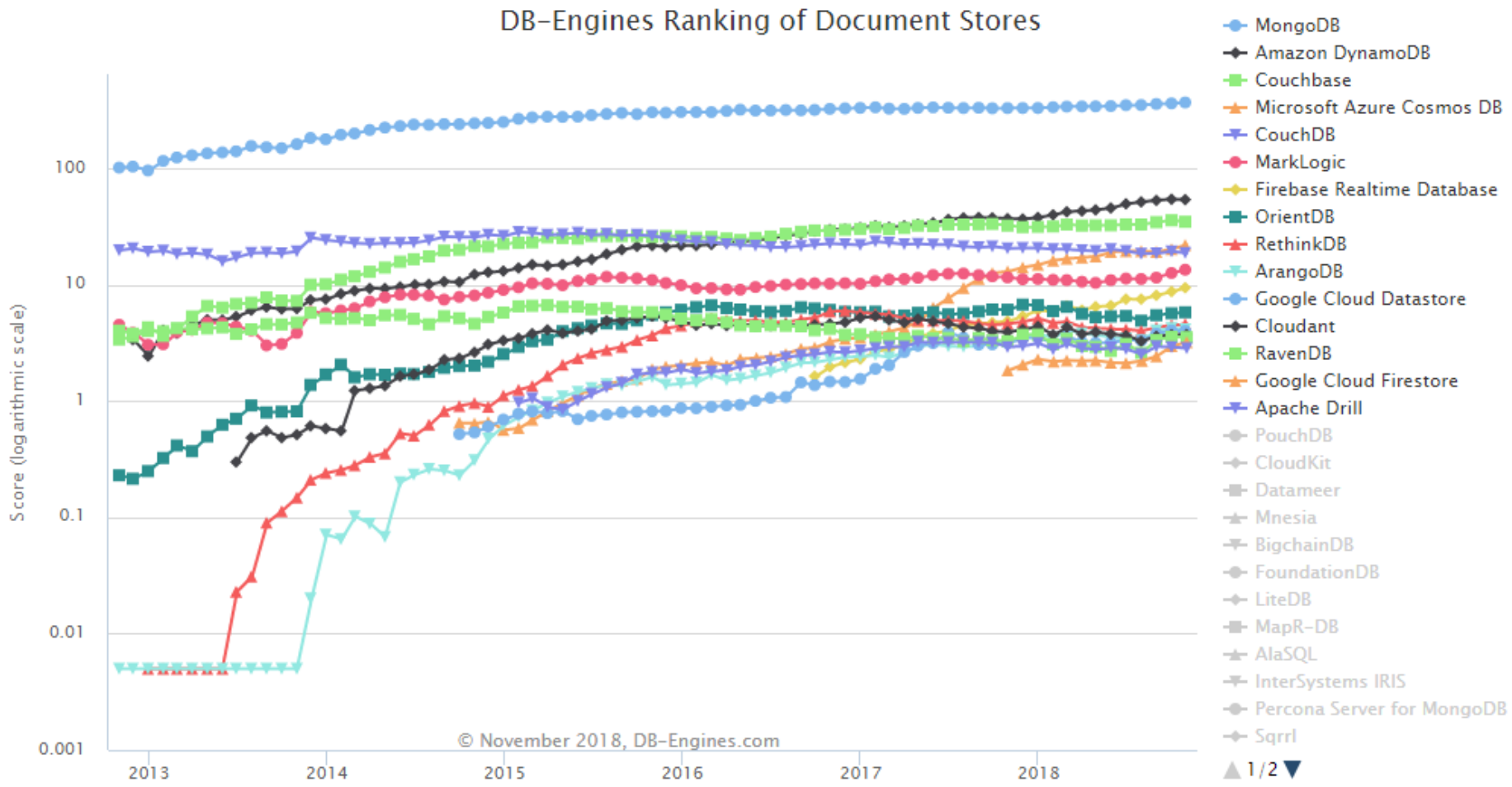
Ranking of Relational DBMS



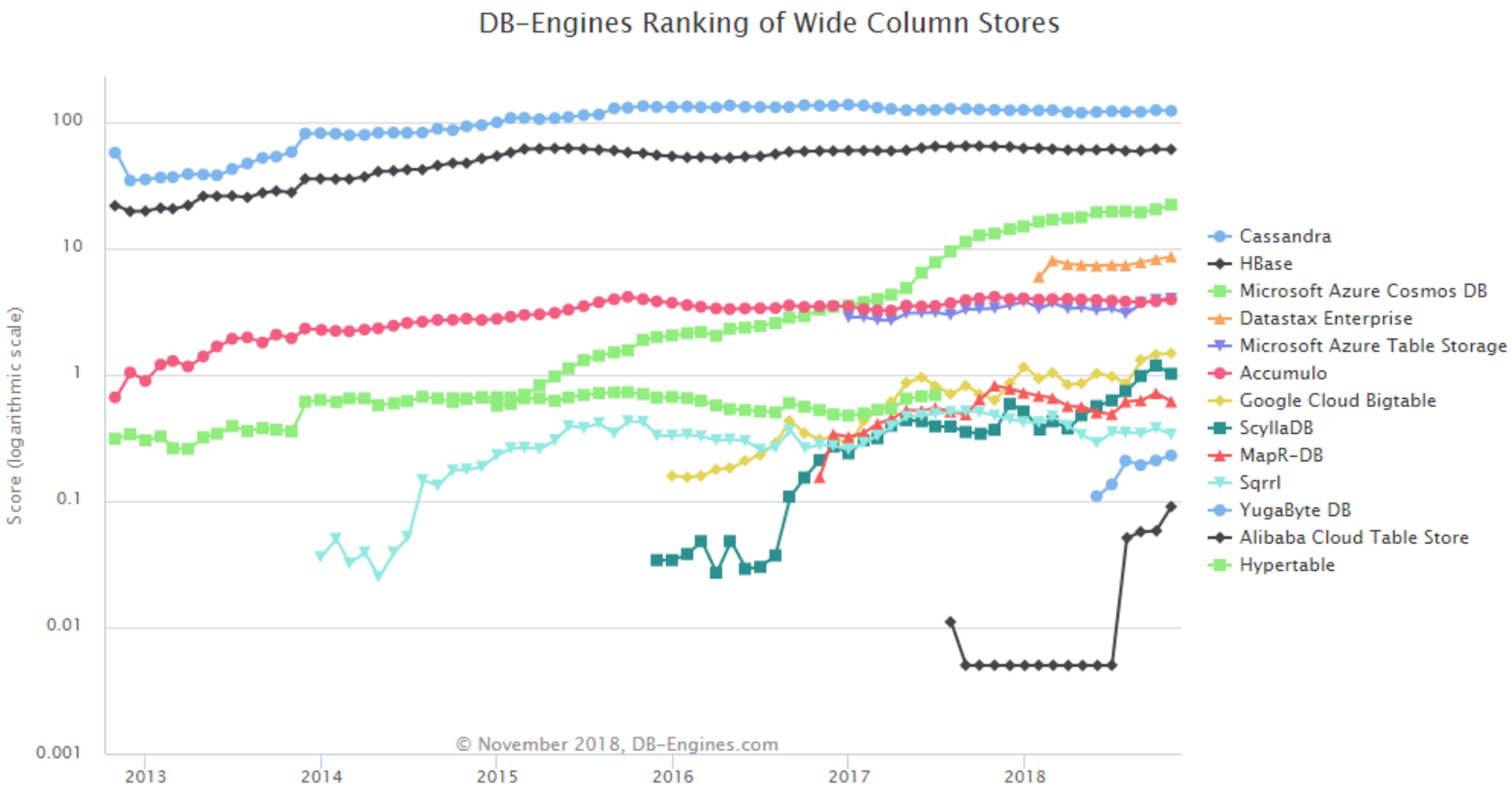
Ranking of Key-value Stores



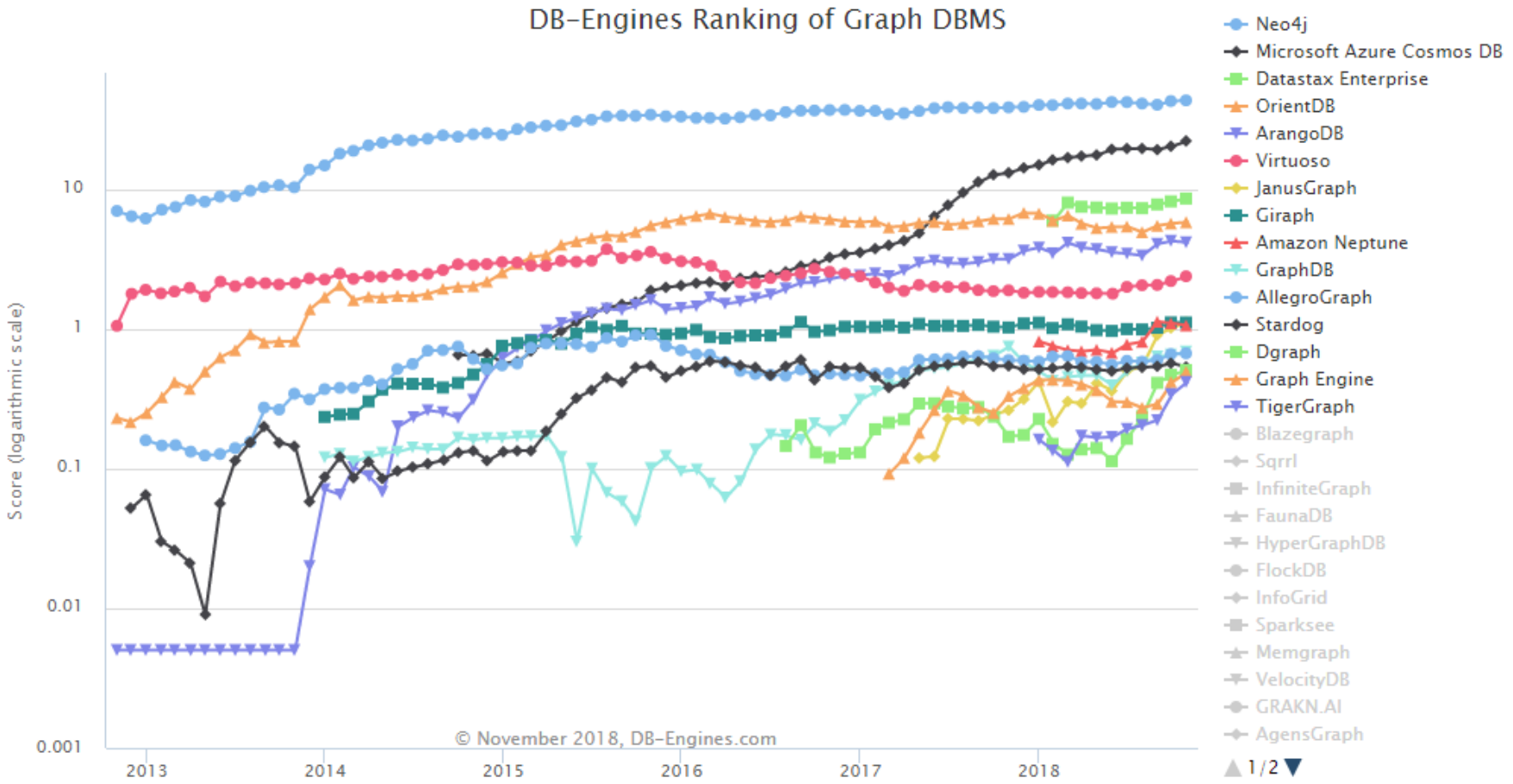
Ranking of Document Stores



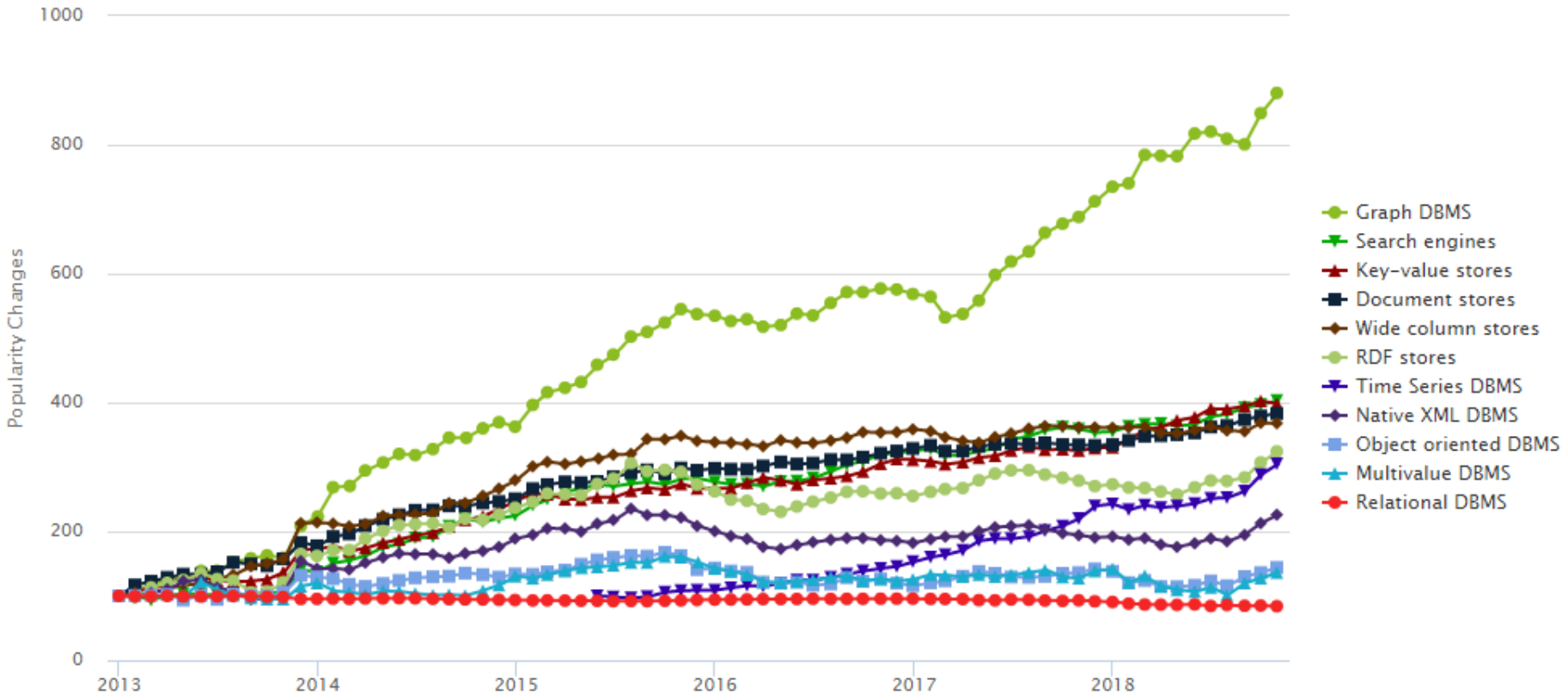
Ranking of Column Families Stores



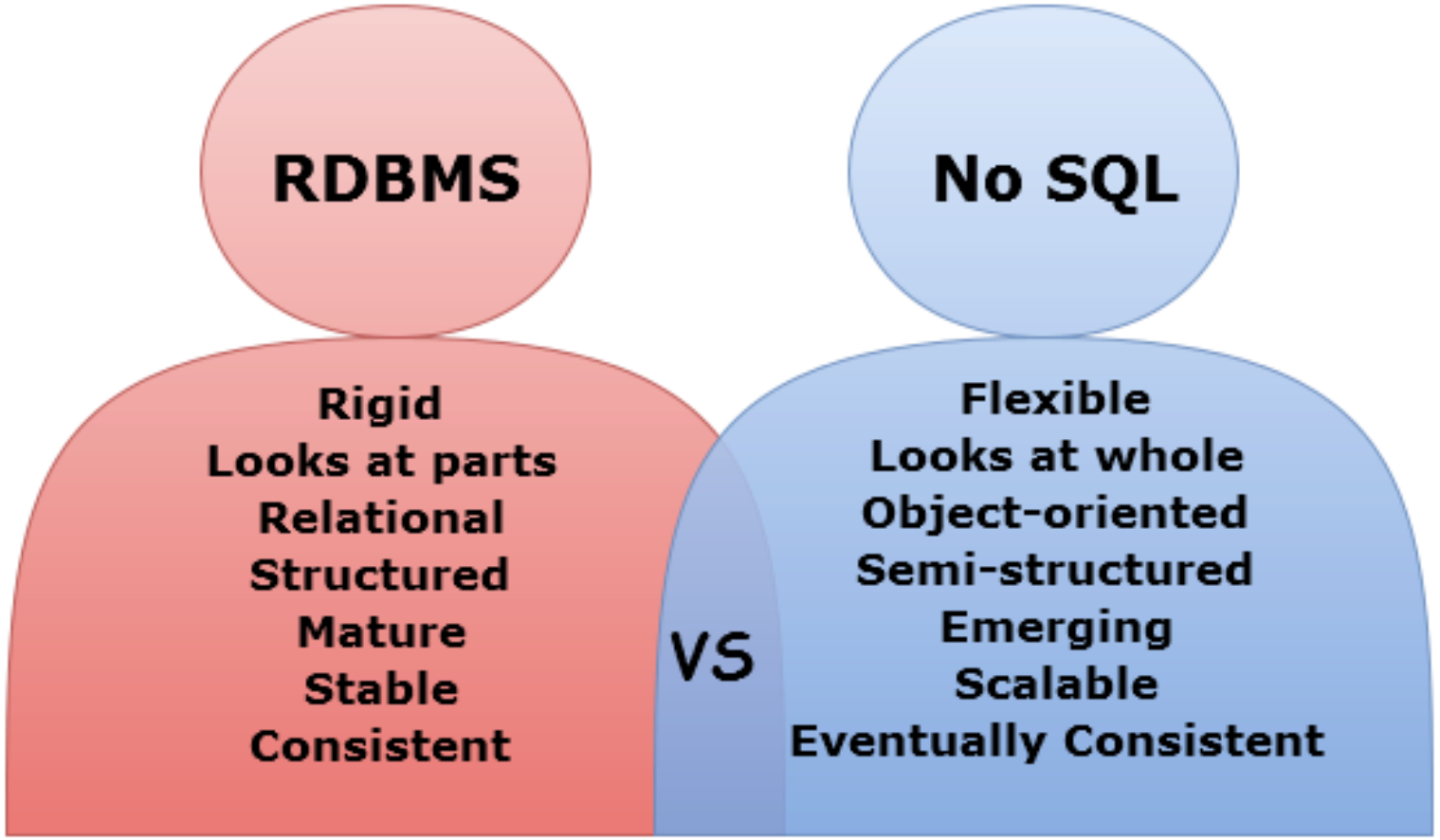
Ranking of Graph DBMS



Popularity changes per category, July 2018



Summary


RDBMS
Rigid
Looks at parts
Relational
Structured
Mature
Stable
Consistent
VS
No SQL
Flexible
Looks at whole
Object-oriented
Semi-structured
Emerging
Scalable
Eventually Consistent

Reference

- http://nosql-database.org/
- http://wikibon.org/wiki/v/21_NoSQL_Innovators_to_Look_for_in_2020#Introduction
- https://db-engines.com
- http://basho.com/posts/technical/why-vector-clocks-are-easy/
- …


TapadhLeat
Köszönöm
Murakoze
Grazzi
Grazie
Shukriya
Buznyg
WaadMahadsantahay
aDank
Nouari
ThintKo
Takk
Enkosi
FaafetaiLava
Ngiyabonga
TapadhLeibh
M-Sapo
Bedankt
Zikomo
BarakAllahuFiik
Mèsi
Chokrane
Kiitos
AsanteSana
Blagodaram
Waita
Rahmat
Thank
Dhanyavaad
Xièxie
Dziakuju
Maururu
GoRaibhMaithAgat
Trugarez
KyayTzuTinPaTe
Mochchakkeram
Matondo
Aabhar
Welalin
TangioTumas
Dhanyavad
TeşekkürEderim
Sobodi
AabhariAahe
Barkal
KurreSumanga
GratiasAgimus
Mercé
TerimaKasih
Gracias
Yekeniele
Dhanyabaad
TesekkurEdirem
Tanan
Sukriya
Mamnun
Taiku
KeYaLeboha
Tanemirt
Vinaka
KhobChaiDeu
Miesker
TänanVäga
Ačiū
KopKhunKrap
Barka
Dankon
DankJe
Tenki
Tau
Nanni
Dakujem
aNiKié
Merci
You
Dankie
Toda
Paldies
DjiereDieuf
Gratzias
Chnorakaloutioun
Trugéré
Motashakkeram
Mersi
Nizzik ajr
Diolch
ANiKié
Akpé
Misaotra
Tak
Spacibo
Danke
Aguyjé
Dhanyavadagalu
Sagolun
Grandmercé
Dhanyavadalu
Sulpáy
EskerrikAsko
Dankewol
Rahmet
GratiasAgo
Tanmirt
Gràcies
Blagodaria
Nandri
Arigatô
Hvala
Spas
MaloTeOfa
Akiba
NajisTuke
Merkzi
Akun
Wado
Asante
KopKhunKha
Faleminderit
Mahalo
Graciñas
KamSahHamnida
Gracie
CámOn
Efharisto
DankJuWel
Marahaba
ÇoxSagOlun
Multumesc
Saha
Diky
Meharbani
Obrigado
TakkFyri
Salamat
Nirringrazzjak
Bayarlalaa
Miigwetch