Data Modeling
NoSQL

What is NoSQL?

NoSQL ≠ ‘NO SQL’

NoSQL = ‘Not Only SQL’

Scaling issue with RDBMS ---> NoSQL

https://en.wikipedia.org/wiki/NoSQL;
https://www.dataversity.net/a-brief-history-of-non-relational-databases/#:~:text=The%20acronym%20NoSQL%20was%20first,to%20describe%20non%2Drelational%20databases.

NoSQL Databases

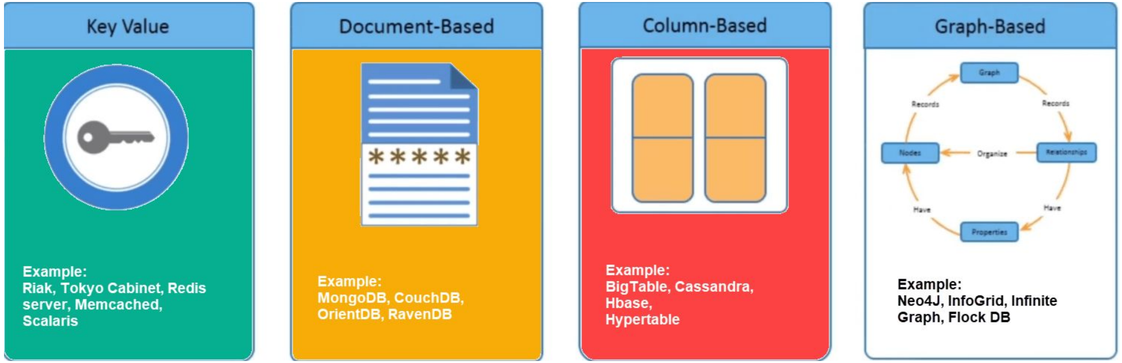
Quick Discussion

Difference between SQL and NoSQL

No SQL

Categories of NoSQL

- Key-Value
- Document
- Column
- Graph



Key Value

Example:
Riak, Tokyo Cabinet, Redis server, Memcached, Scalaris

Document-Based

Example:
MongoDB, CouchDB, OrientDB, RavenDB

Column-Based

Example:
BigTable, Cassandra, Hbase, Hypertable

Graph-Based

Example:
Neo4J, InfoGrid, Infinite Graph, Flock DB

Commonality ---> Non- Relational, Open source

https://www.guru99.com/nosql-tutorial.html#:~:text=NoSQL%20Databases%20are%20mainly%20categorized,its%20unique%20attributes%20and%20limitations.

NoSQL

DATASTAX®
Active Everywhere—Every Cloud

amazon
DynamoDB

mongoDB®

IBM Cloudant®

Characteristics of NoSQL

- High Scalability
- Developer friendly
- Cost effective
- Support flexible schema
- ‘Share data more easily than RDBMS

https://www.analyticssteps.com/blogs/nosql-database-characteristics-types-and-applications

Why NoSQL? ----> Benefits

- Scalability
- Performance
- High Availability
- Cloud integration
- Cost
- Flexible schema
- Different data structures

https://www.mongodb.com/nosql-explained/advantages

Categories of NoSQL DataBase

Key-Value

Key → **Value**

name → William

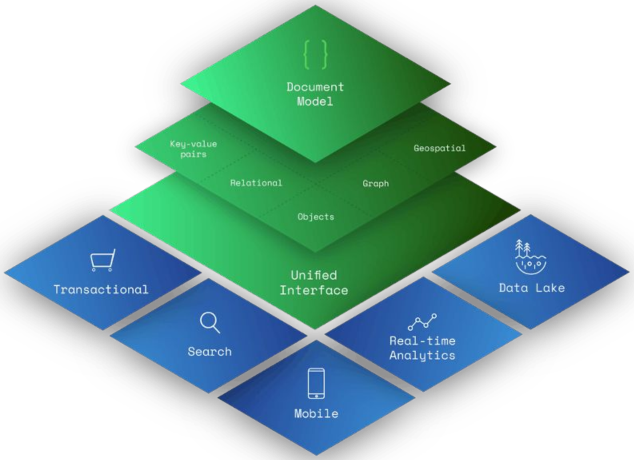
City → Chicago

- Least complex
- Basic CRUD
- Scale well
- Not suitable for complex queries
- Good for storing user profile
- Storing shopping data

https://en.wikipedia.org/wiki/Key%E2%80%93value_database

Categories of NoSQL DataBase

Document Based



- Allows Horizontal Scaling
- Sharding across multiple nodes

Suitable for

- Event logging
- Online blogging

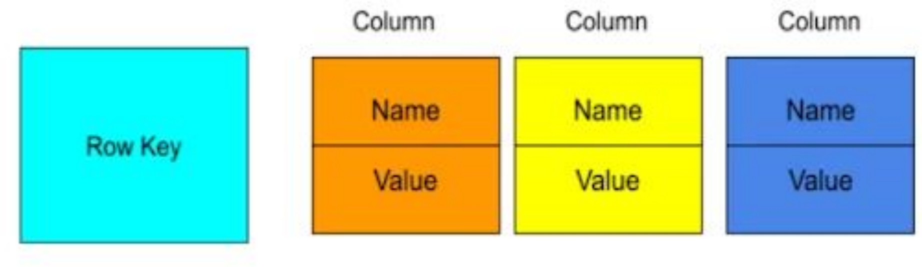
Not Suitable for

- ACID transaction
- Aggregated or oriented design

https://www.mongodb.com/document-databases

Categories of NoSQL DataBase

Column Based



Suitable for:

- Large amount of sparse data
- Deployment across nodes
- Logging and blogs

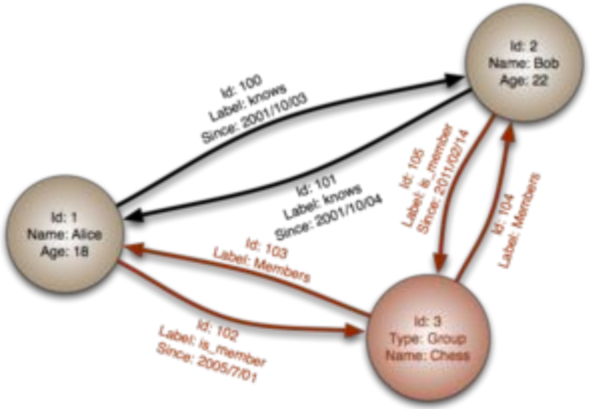
Not Suitable for:

- ACID transaction

https://www.blazeclan.com/blog/dive-deep-types-nosql-databases/

Categories of NoSQL DataBase

Graph Based



Suitable for;

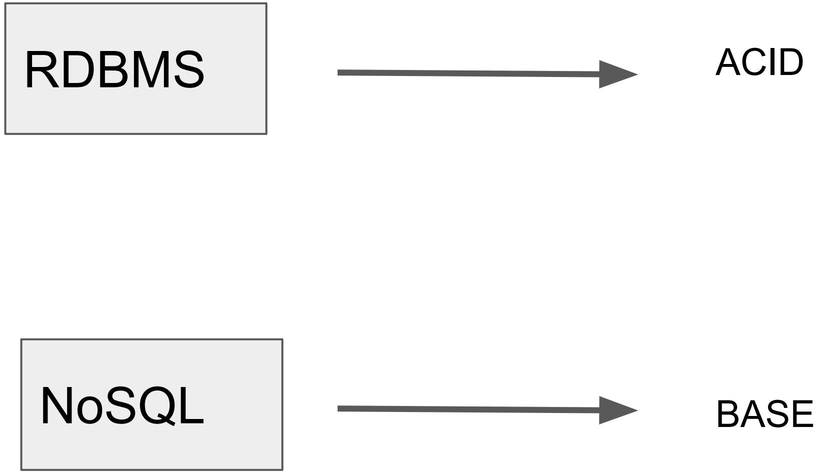
- Connected data
- routing

Not Suitable for;

- Scalability

https://en.wikipedia.org/wiki/Graph_database

Acid vs Base in Database



RDBMS → ACID

NoSQL → BASE

https://neo4j.com/blog/acid-vs-base-consistency-models-explained/

Acid Consistency Model

- For Relational Database
- Fully consistent

A- Atomic

C - Consistent

I - Isolated

D - Durable

Use cases:

- Financial institution
- Online transaction processing

https://medium.com/geekculture/acid-vs-base-in-databases-1bcad774da26

BASE Consistency Model

Basically Available

Soft State

Eventual Consistency

Use cases:

- Marketing
- Sentiment analysis
- Social media
- Online services

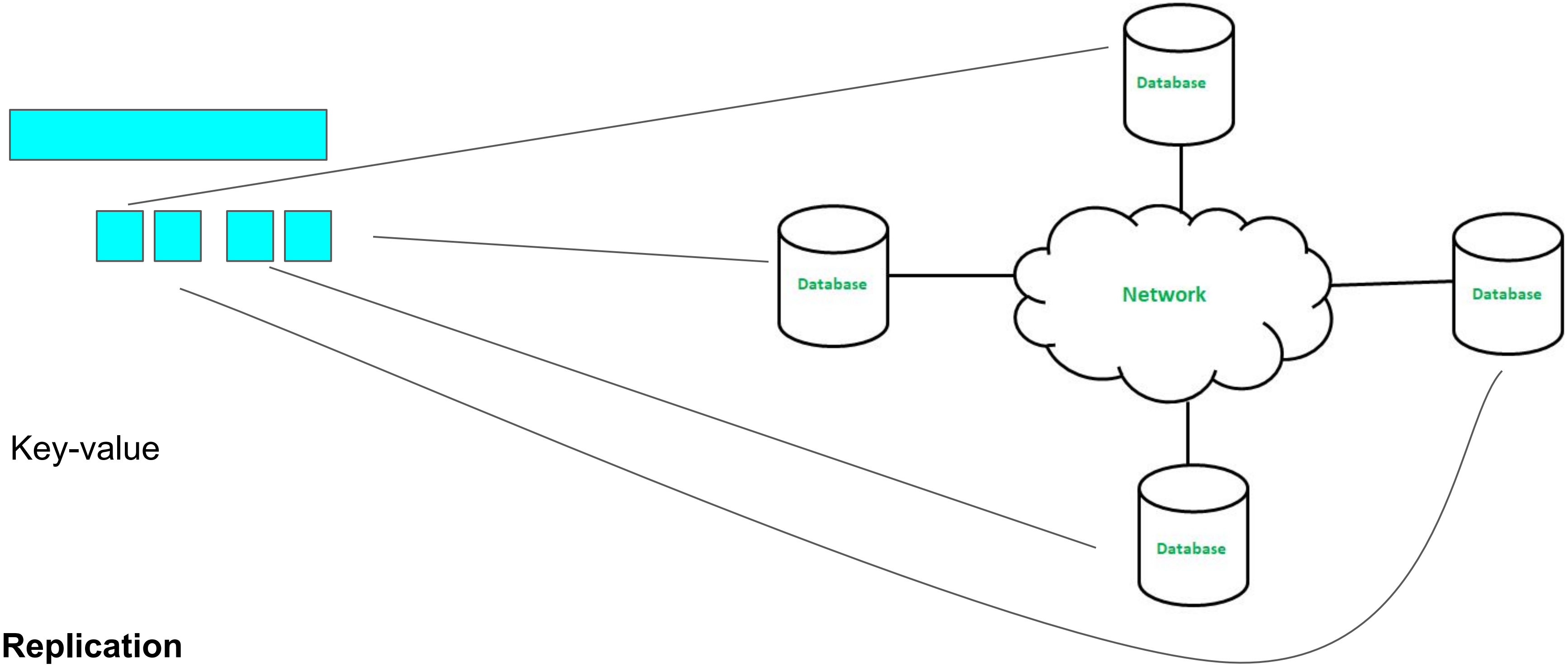
NETFLIX

Uber

Spotify

Quick Recap

Concept of Distributed Systems



Key-value

Replication

https://cs.uwaterloo.ca/~tozsu/courses/cs856/F02/lecture-1-ho.pdf

Advantages of Distributed Database

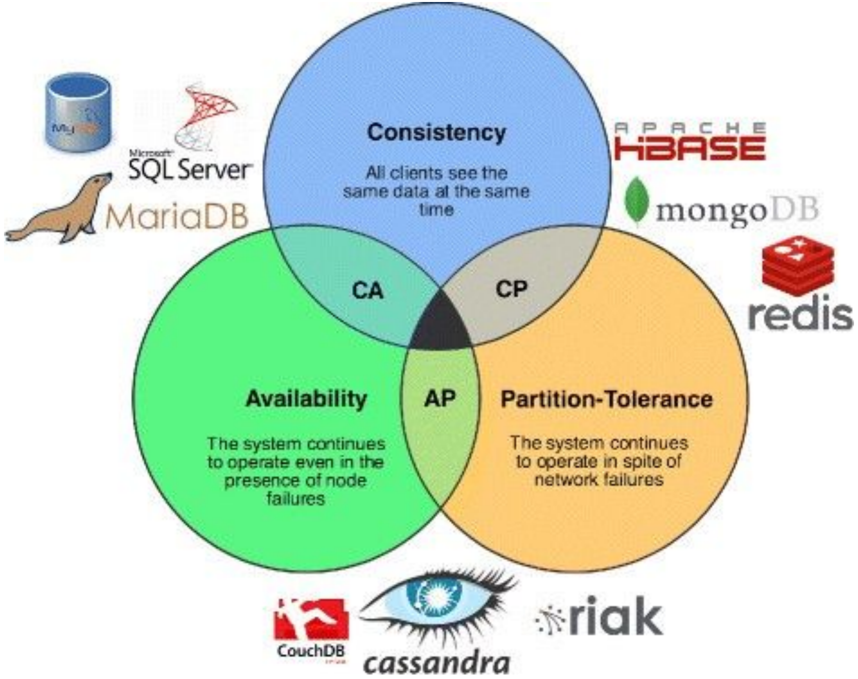
- Availability and reliability
- High performance
- Reduced query time
- Ease of scale
- Follow the base model

Challenges include:

- Concurrency control i.e consistency of data

Recap

CAP Theorem


Consistency
All clients see the same data at the same time
CA
CP
Availability
The system continues to operate even in the presence of node failures
AP
Partition-Tolerance
The system continues to operate in spite of network failures
MySQL
Microsoft SQL Server
MariaDB
APACHE HBASE
mongoDB
redis
CouchDB
cassandra
riak

Prof Eric Brewer

https://medium.com/system-design-blog/cap-theorem-1455ce5fc0a0
https://en.wikipedia.org/wiki/Eric_Brewer_(scientist)

Common challenges

Migration from RDBMS to NoSQL

When to use RDBMS or NoSQL

RDBMS

- Consistency
- Fixed schema
- Transactions
- Joins

NoSQL

- High Performance
- Availability
- Easy Scalability
- Unstructured Data

https://www.educba.com/rdbms-vs-nosql/

Working with RDBMS and NoSQL

- RDBMS is Data Driven ; NoSQL is Query driven
    - RDBMS ---> starts considers the relationships between entities
    - NoSQL considers the query that you want to perform
- RDBMS starts with Normalized data and builds queries
- NoSQL structures the data based on query

Migrating from RDBMS to NoSQL

CAP Theorem

Consistency

Availability

Partitioning