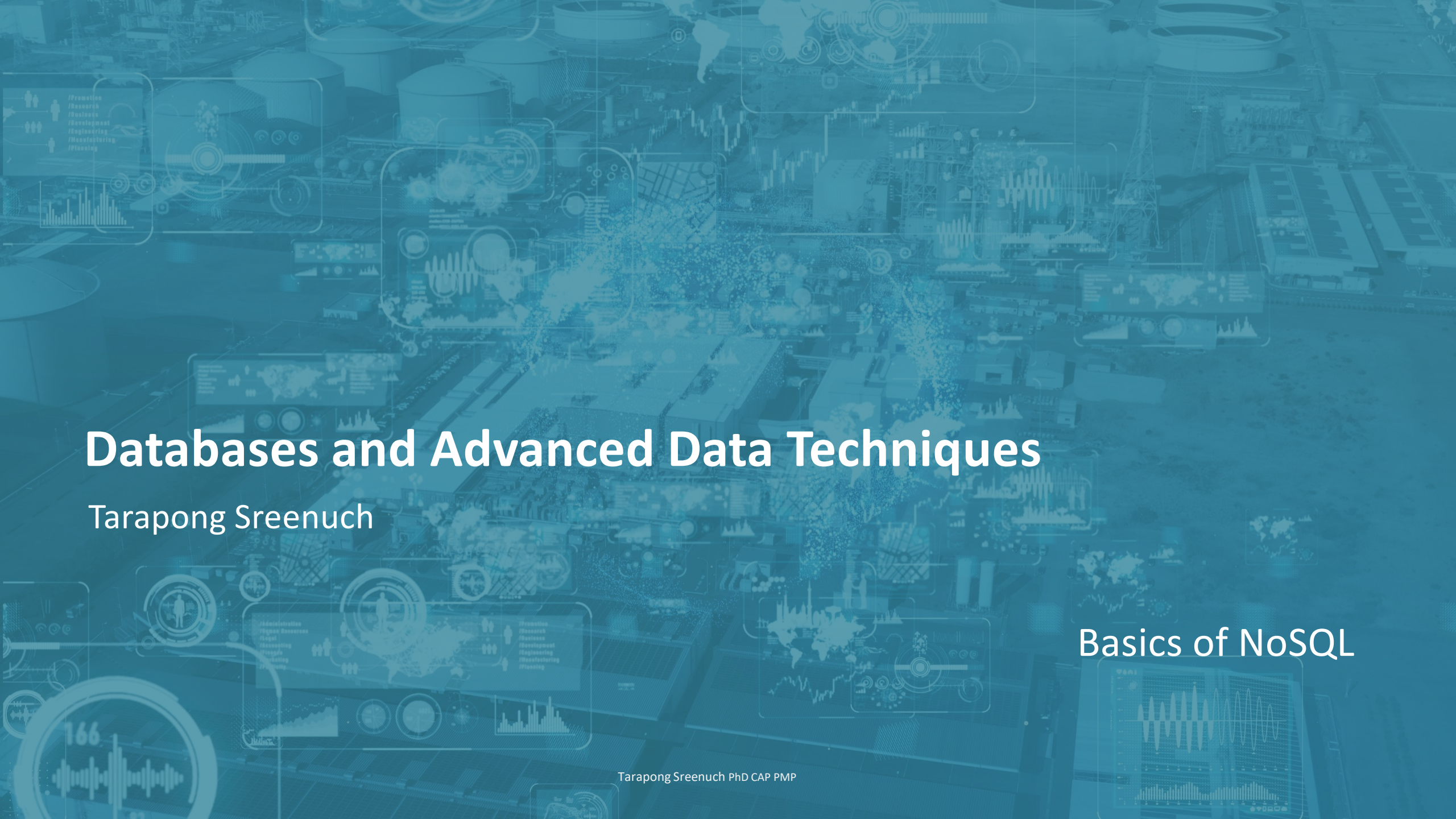


Databases and Advanced Data Techniques

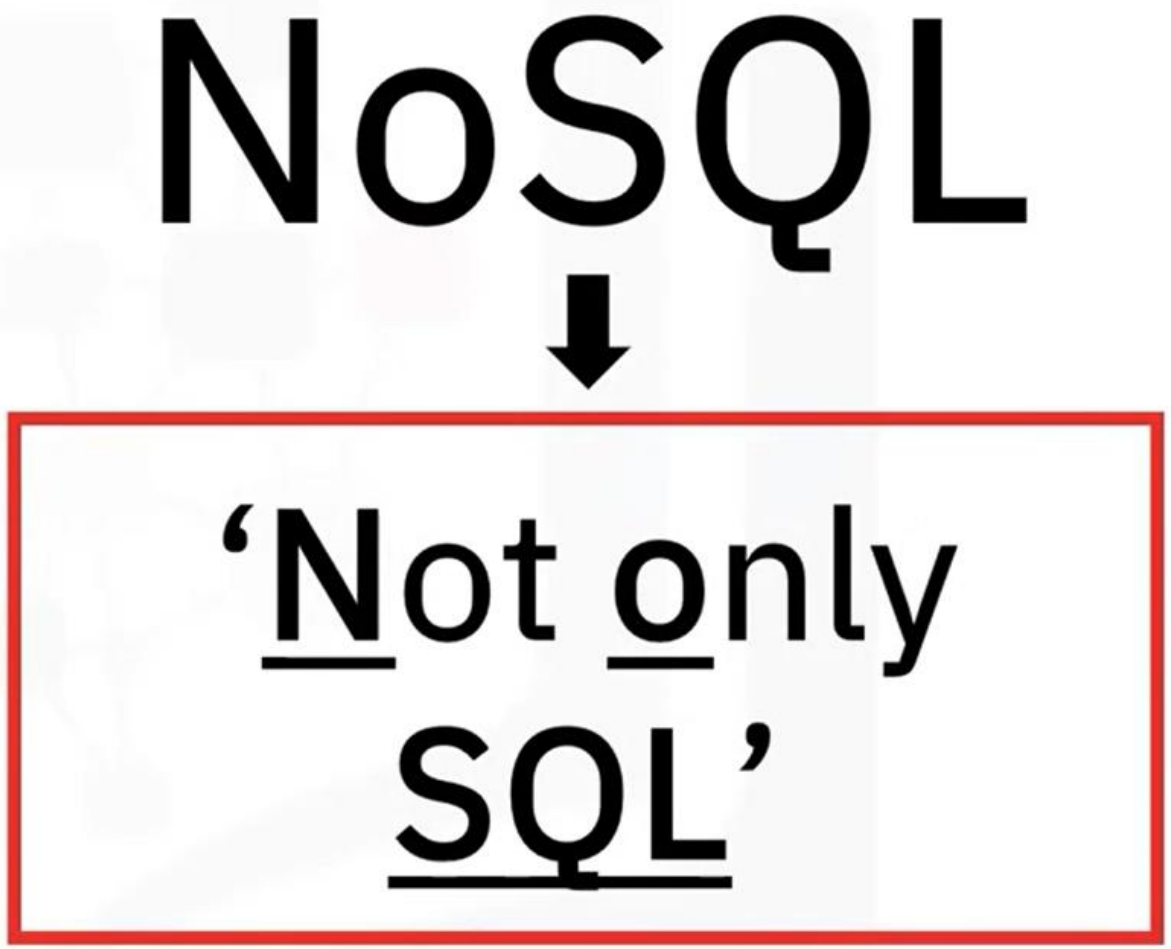
Tarapong Sreenuch

Basics of NoSQL

Overview of NoSQL

What is NoSQL?

- The NoSQL name was introduced during an open-source event on distributed databases
- It doesn't mean 'No SQL' – it means 'Not only SQL'



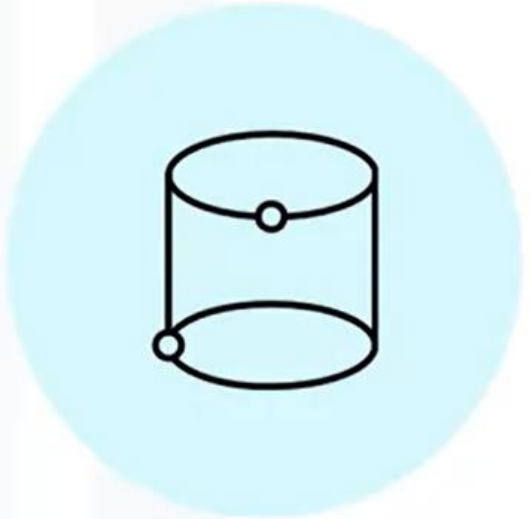
What is NoSQL?

- Refers to family of databases that vary widely in style and technology
- However, they share a common trait
  - Non-relational
  - Not standard row and column type RDBMS
- Could be referred to as ‘Non-relational databases’

What is NoSQL?

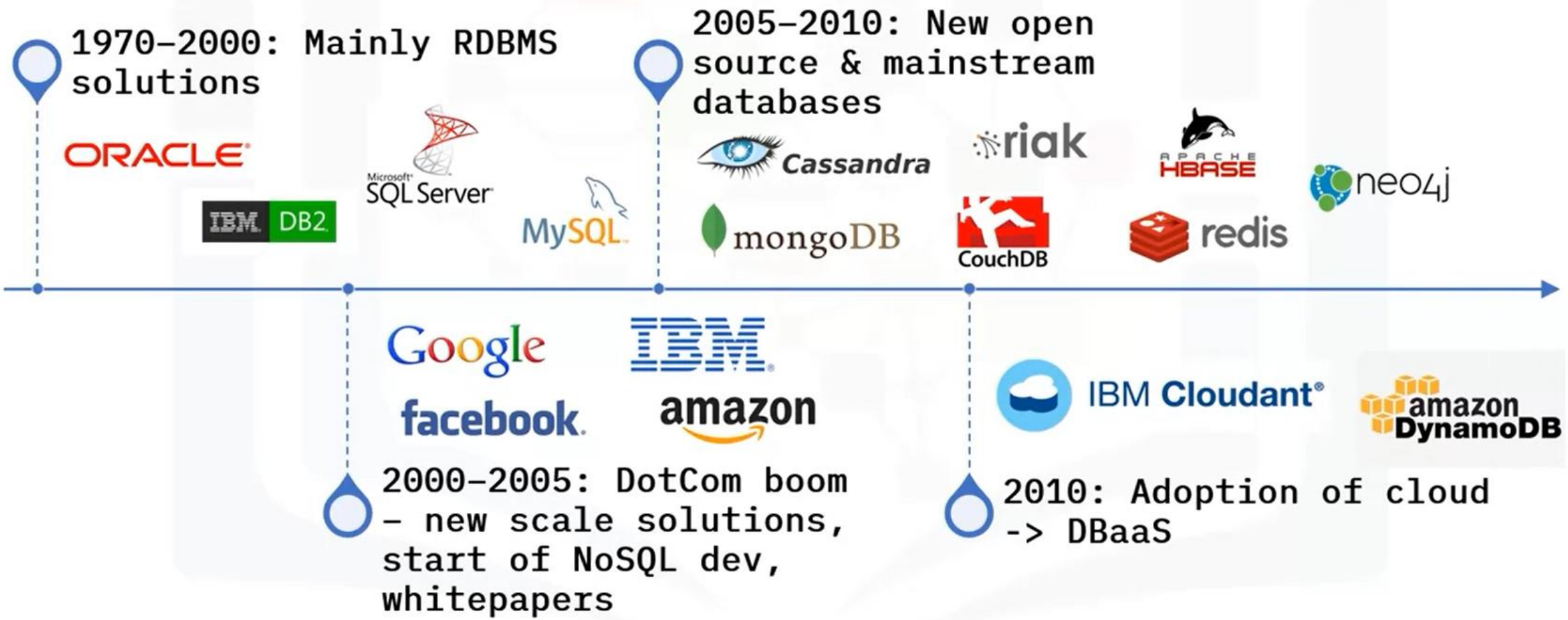
NoSQL databases:

- Provide new ways of storing and querying data
  - Address several issues for modern apps
- Are designed to handle a different breed of scale – 'Big Data'
- Typically address more specialized use cases
- Simpler to develop app functionality for than RDBMS



NoSQL Databases

History of NoSQL



Recap: Overview of NoSQL

- The name ‘NoSQL’ stands for Not only SQL
- NoSQL refers to a class of databases that are non-relational in architecture
- Implementations of NoSQL databases differ technically, but share common traits
- Since 2000 NoSQL databases have become more popular in the marketplace, due to scale demands of Big Data

Characteristics of NoSQL Databases

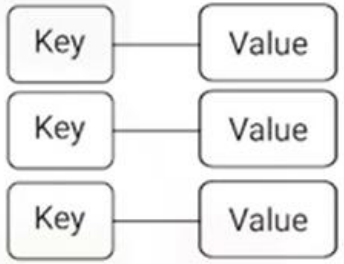
NoSQL Database Categories

- The most common trait amongst NoSQL databases is that they are non-relational in architecture
- What types of NoSQL databases are available?
- What is common to them?

NoSQL Database Categories

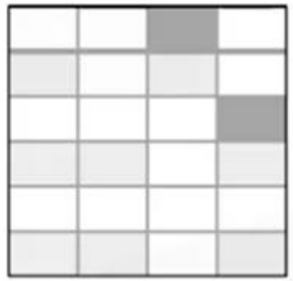
General consensus is...

...NoSQL databases fit into four categories



Key-Value

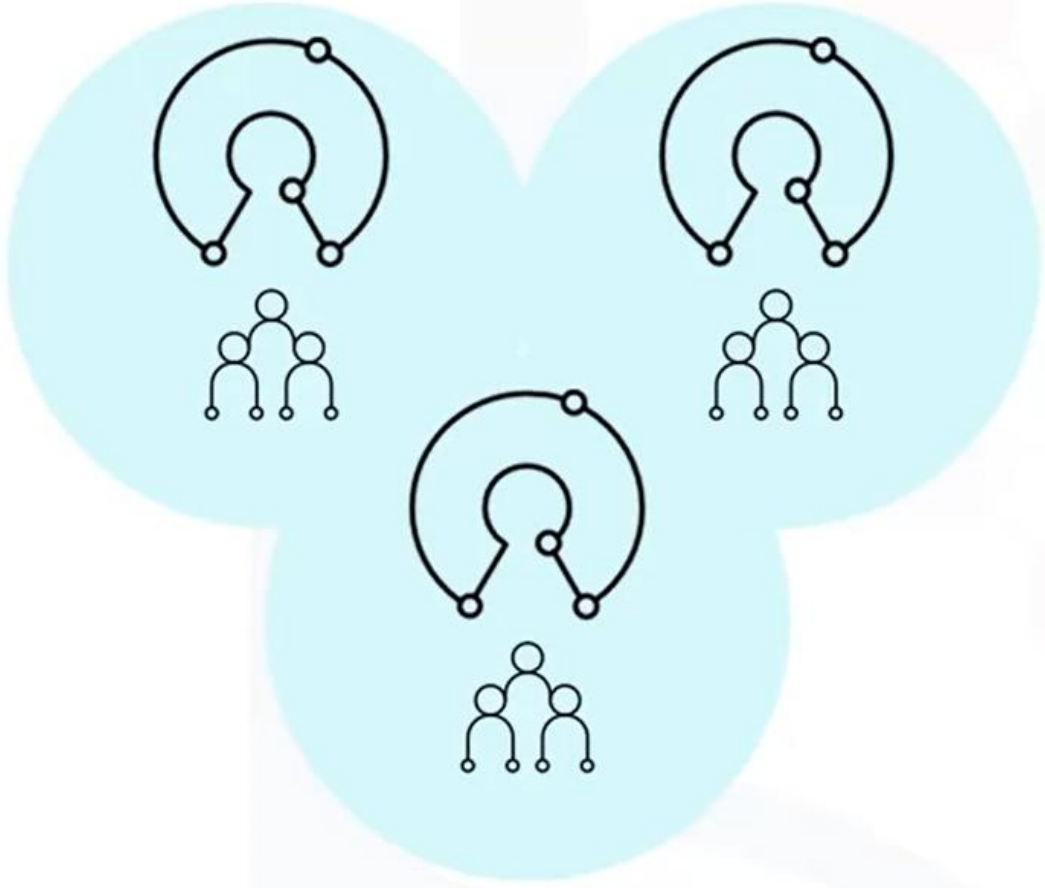
Document



Column

Graph

NoSQL Database Categories



Open Source Communities

But what ties NoSQL databases together?

- Majority have their roots in the open source community
- Many have been used and leveraged in an open source manner
- Open source community support is fundamental to their industry growth

NoSQL Database Categories

- Companies often develop a commercial version of the database alongside the open source version
- Examples include:

DATASTAX

IBM Cloudant®

mongoDB

NoSQL Database Categories

- They all differ technically speaking, but have a few commonalities
- Most NoSQL databases:
  - Are built to scale horizontally
  - Share data more easily than RDBMS
  - Use a global unique key to simplify data sharding
  - Are more use case specific than RDBMS
  - Are more developer friendly than RDBMS
  - Allow more agile development via flexible schemas

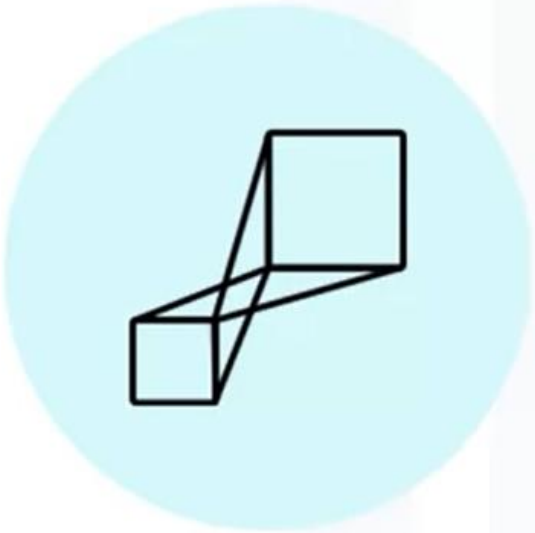
Benefits of NoSQL Databases

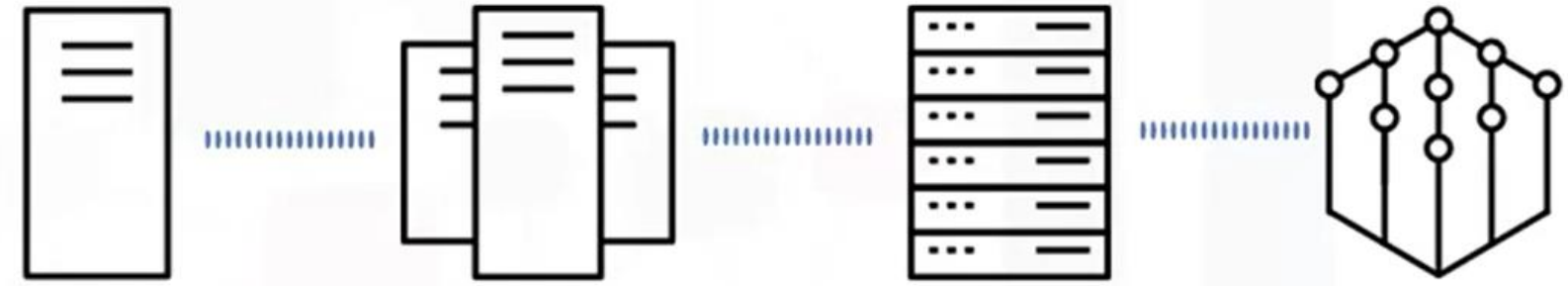
Why use a NoSQL database?

Why is their popularity growing so rapidly?

Benefits of NoSQL Databases

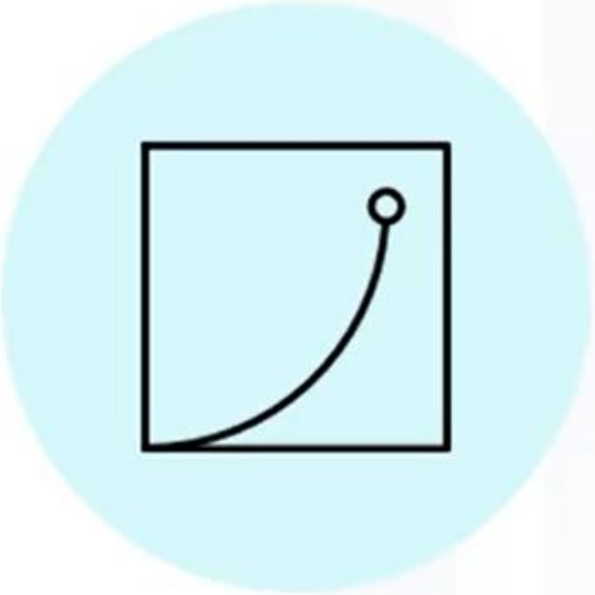
Scalability

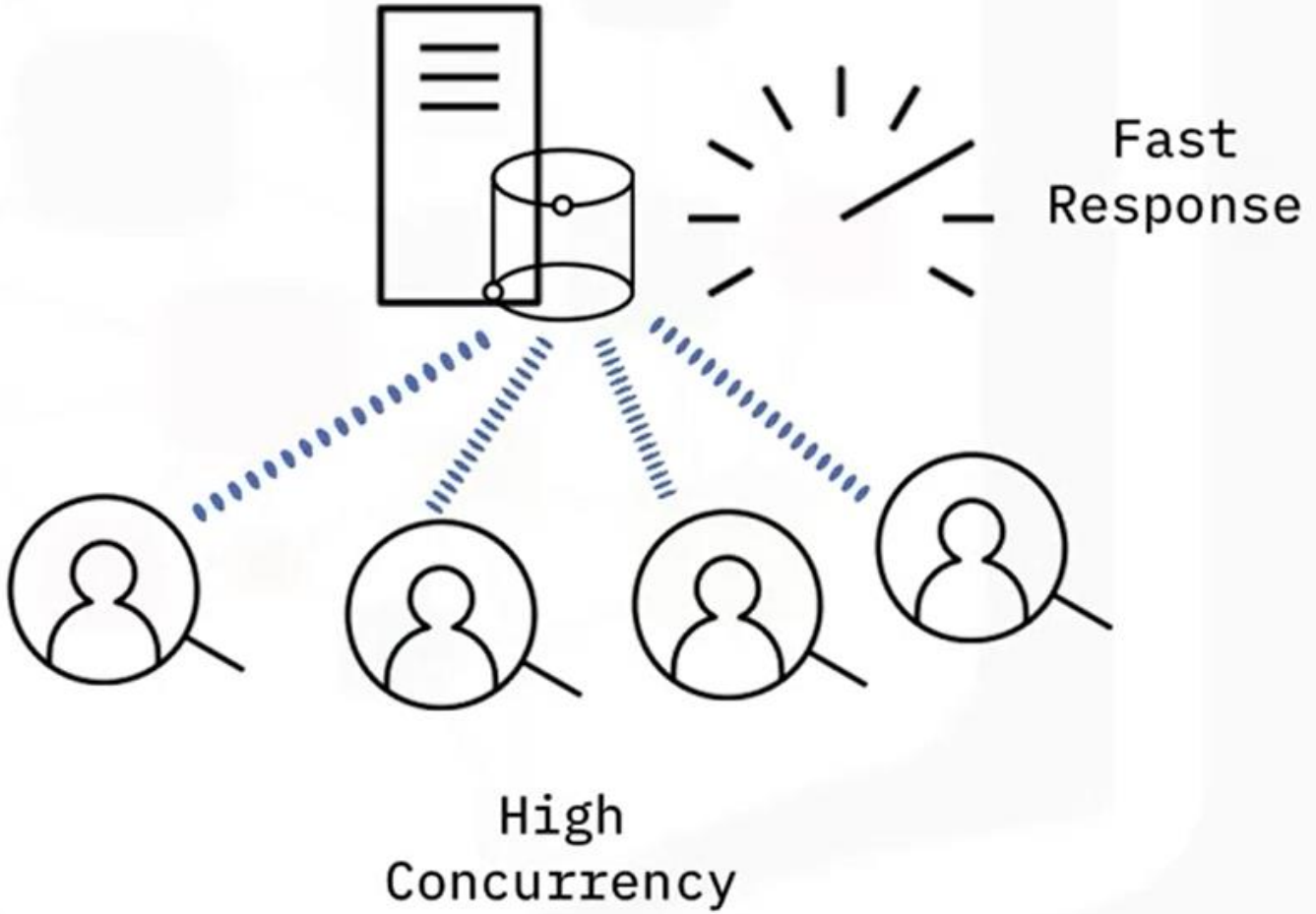




Benefits of NoSQL Databases

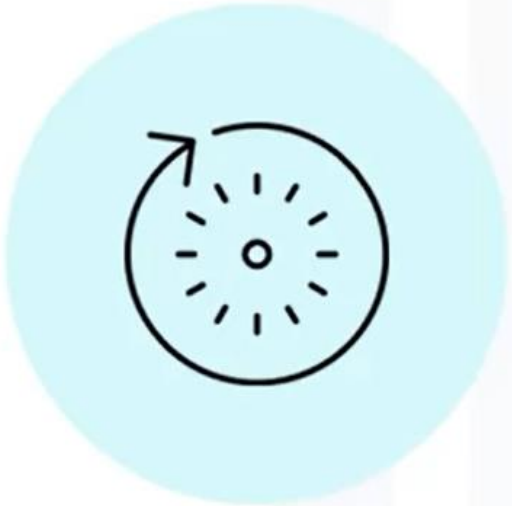
Performance

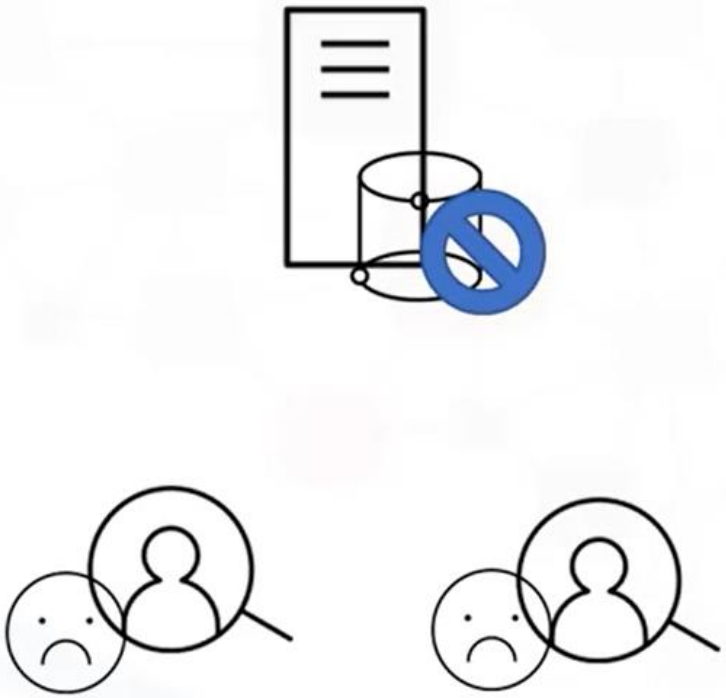


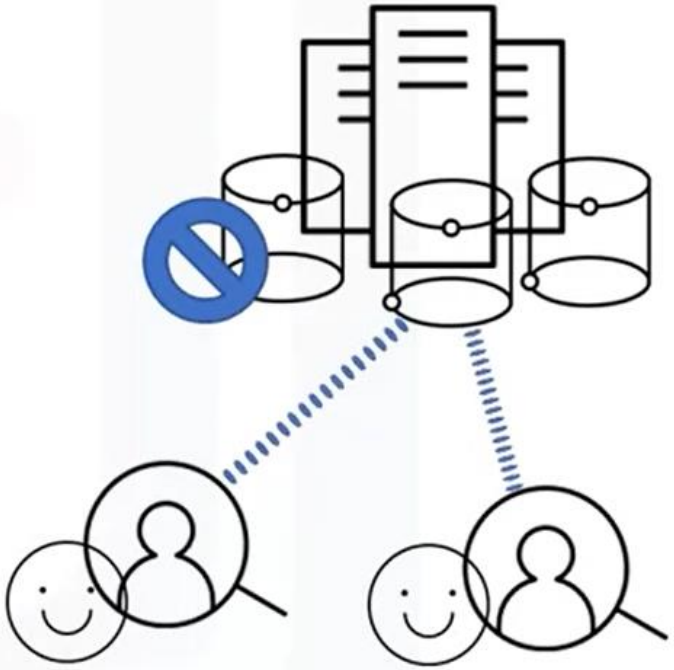


Benefits of NoSQL Databases

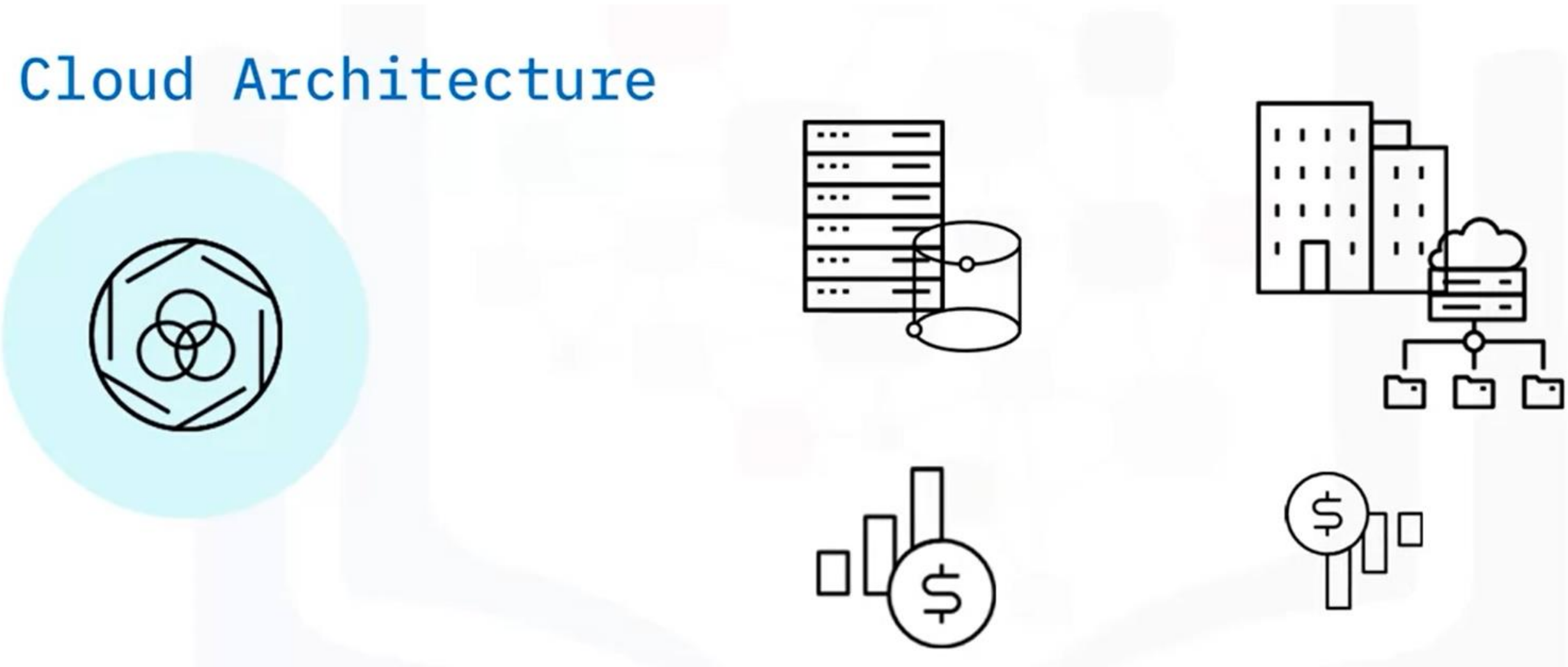
Availability





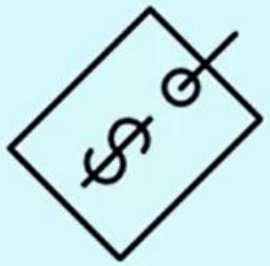


Benefits of NoSQL Databases



Benefits of NoSQL Databases

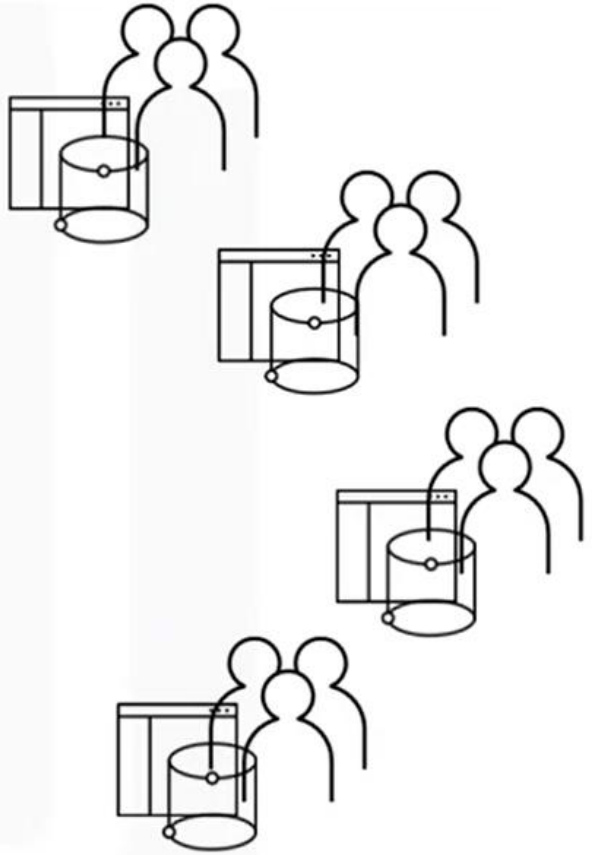
Cost



NoSQL

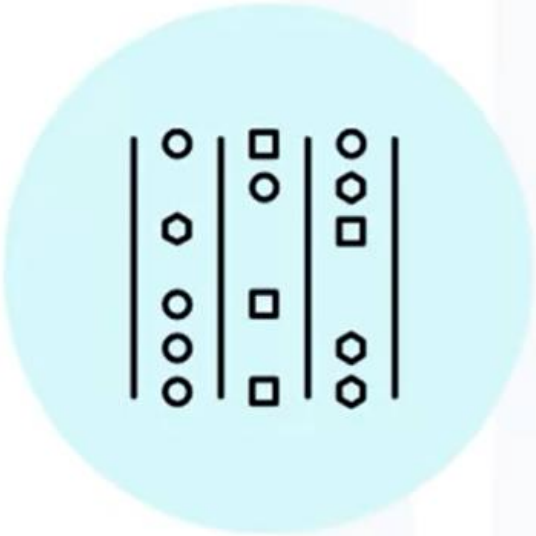
Benefits of NoSQL Databases

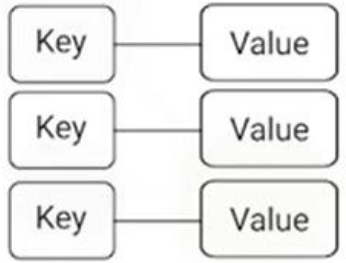
Flexible Schema



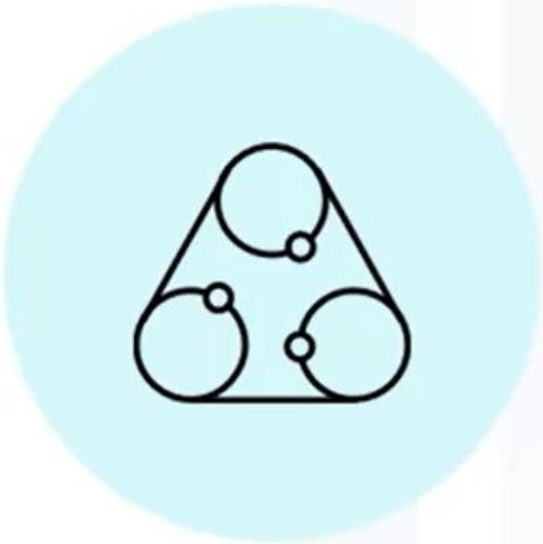
Benefits of NoSQL Databases

Varied Data Structures

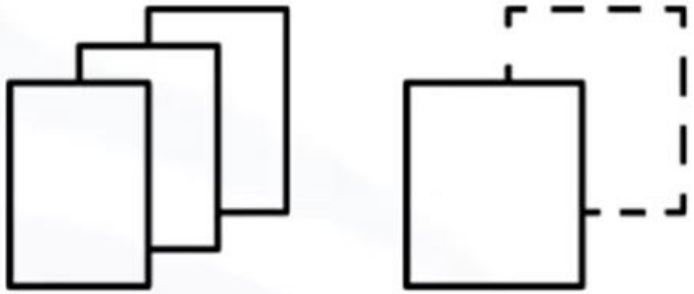




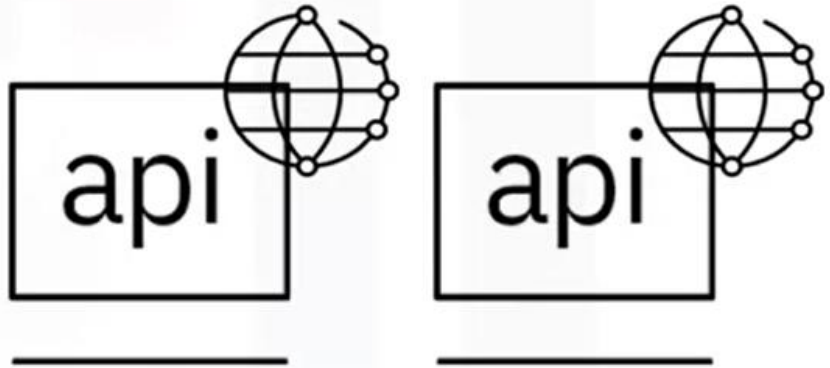
Specialized Capabilities



Indexing and Querying



Data Replication Robustness



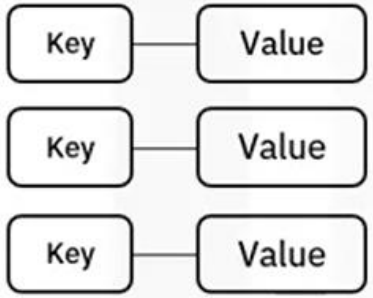
Modern HTTP APIs

Benefits of NoSQL Databases

- NoSQL databases are non-relational
- There are four categories of NoSQL database
- NoSQL databases have their roots in the open-source community
- NoSQL database implementations are technically different
- There are several benefits to adopting NoSQL databases

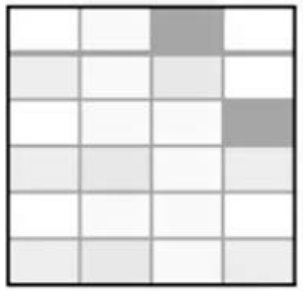
Key-Value NoSQL Databases

NoSQL Database Categories

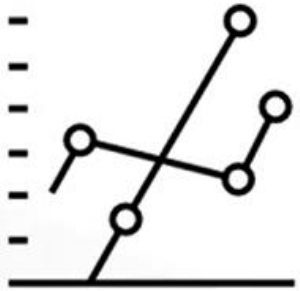


Key-Value

Document



Column

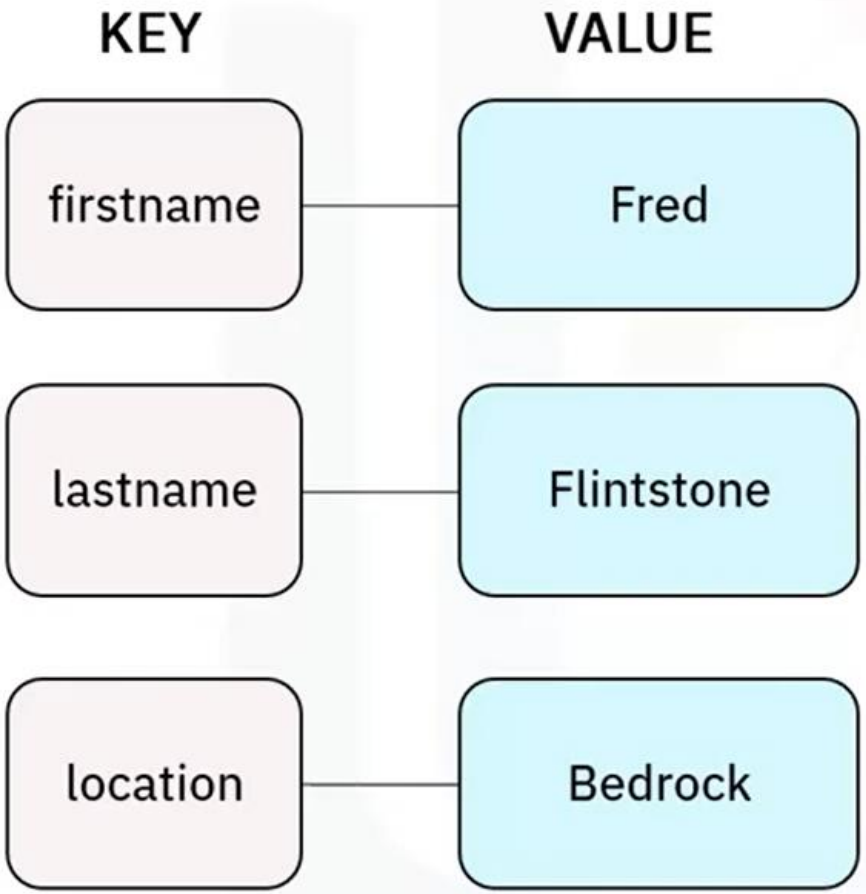


Graph

Each category has:

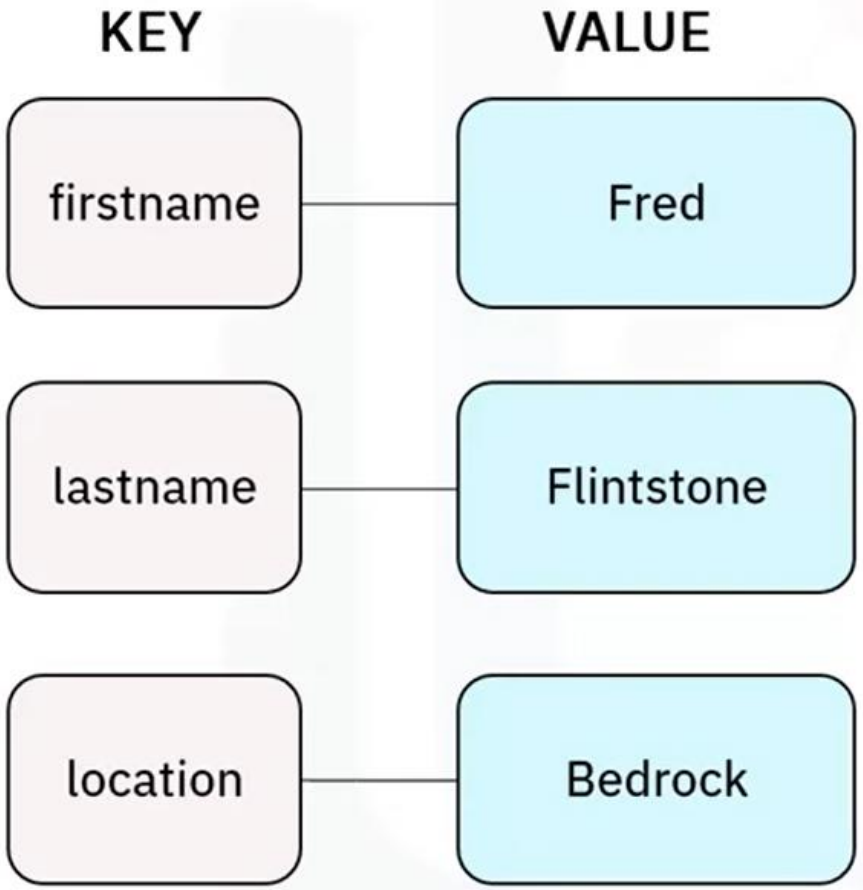
- Unique characteristics
- Architecture
- Use cases

Key-Value NoSQL Database Architecture



- Least complex
- Represented as hashmap
- Ideal for basic CRUD operations
- Scale well
- Shard easily

Key-Value NoSQL Database Architecture



- Not intended for complex queries
- Atomic for single key operations only
- Value blobs are opaque to database
  - Less flexible data indexing and querying

Key-Value NoSQL Database Use Cases

Suitable Use Cases

- For quick basic CRUD operations on non-interconnected data
  - E.g. Storing and retrieving session information for web applications
- Storing in-app user profiles and preferences
- Shopping cart data for online stores

Key-Value NoSQL Database Use Cases

Unsuitable Use Cases

- For data that is interconnected with many-to-many relationships
  - Social networks
  - Recommendation engines
- When high-level consistency is required for multi-operation transactions with multiple keys
  - Need a database that provides ACID transactions
- When apps runs queries based on value vs key
  - Consider using ‘Document’ category of NoSQL database

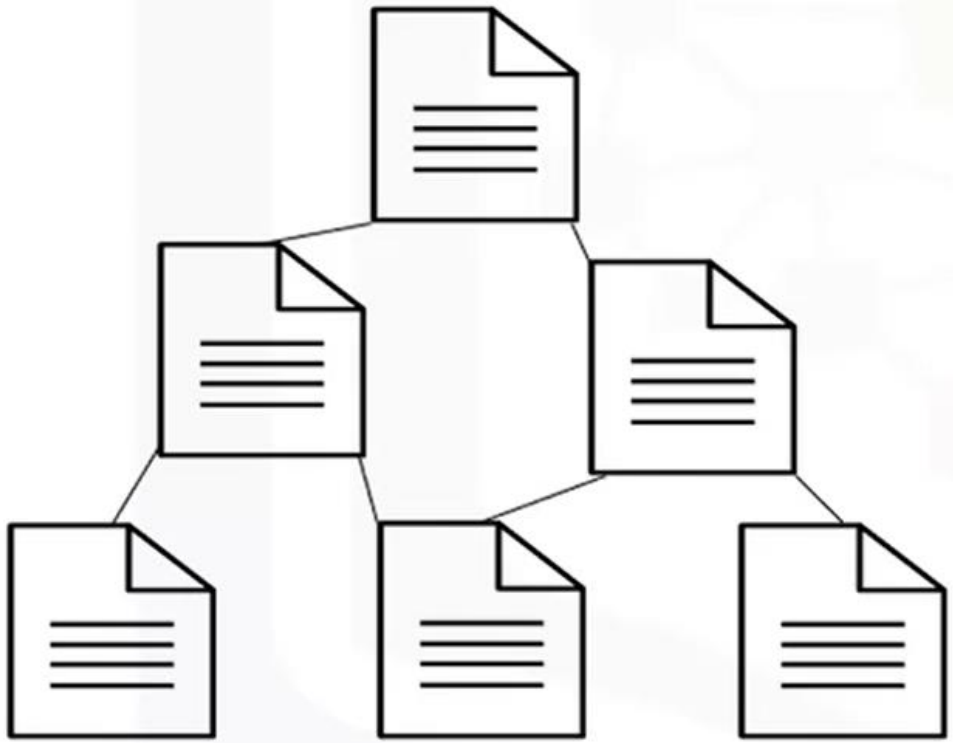
Key-Value NoSQL Database Examples

Recap: Key-Value NoSQL Databases

- The four main categories of NoSQL database are Key-Value, Document, Column, and Graph
- The Key-Value database architecture is the least complex; data is stored with a key and corresponding value blob and is represented by a hashmap
- The primary use cases for Key-Value databases are for quick CRUD operations; for example, storing and retrieving session information, storing in-app user profiles, and storing shopping cart data

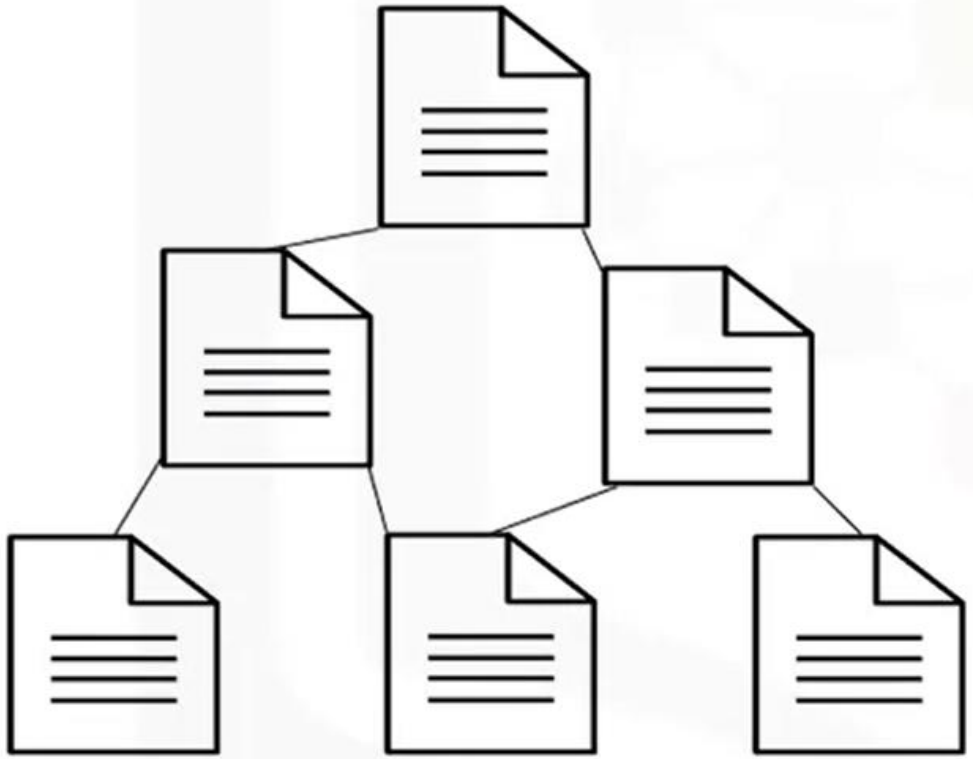
Document-Based NoSQL Databases

Document-Based NoSQL Database Architecture



- Values are visible and can be queried
- Each piece of data is considered a document
  - Typically JSON or XML
- Each document offers a flexible schema
  - No two documents need to contain the same information

Document-Based NoSQL Database Architecture



- Content of document databases can be indexed and queried
  - Key and value range lookups and search
  - Analytical queries with MapReduce
- Horizontally scalable
- Allow sharding across multiple nodes
- Typically only guarantee atomic operations on single documents

Document-Based NoSQL Database Use Cases

Suitable Use Cases

- Event logging for apps and processes – each event instance is represented by a new document
- Online blogs – each user, post, comment, like, or action is represented by a document
- Operational datasets and metadata for web and mobile apps – designed with Internet in mind (JSON, RESTful APIs, unstructured data)

Document-Based NoSQL Database Use Cases

Unsuitable Use Cases

- When you require ACID transactions
  - Document databases can't handle transactions that operate over multiple documents
  - Relational database would be a better choice
- If your data is in an aggregate-oriented design
  - If data naturally falls into a normalized tabular model
  - Relational database would be a better choice

IBM Cloudant®

mongoDB

CouchDB

terrastore

OrientDB®

Couchbase

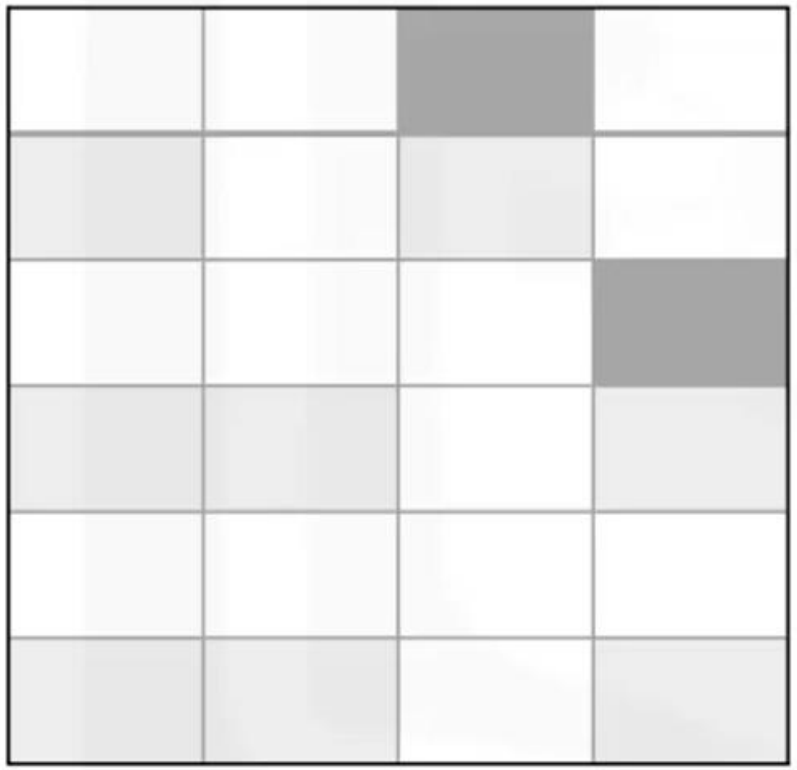
RAVENDB

Recap: Document-Based NoSQL Databases

- Document-based NoSQL databases use documents to make values visible and able to be queried
- Each piece of data is considered a document (JSON/XML)
- Each document offers a flexible schema
- The primary use cases for document-based NoSQL databases are event logging for apps/processes, online blogging, operational datasets or metadata for web and mobile apps

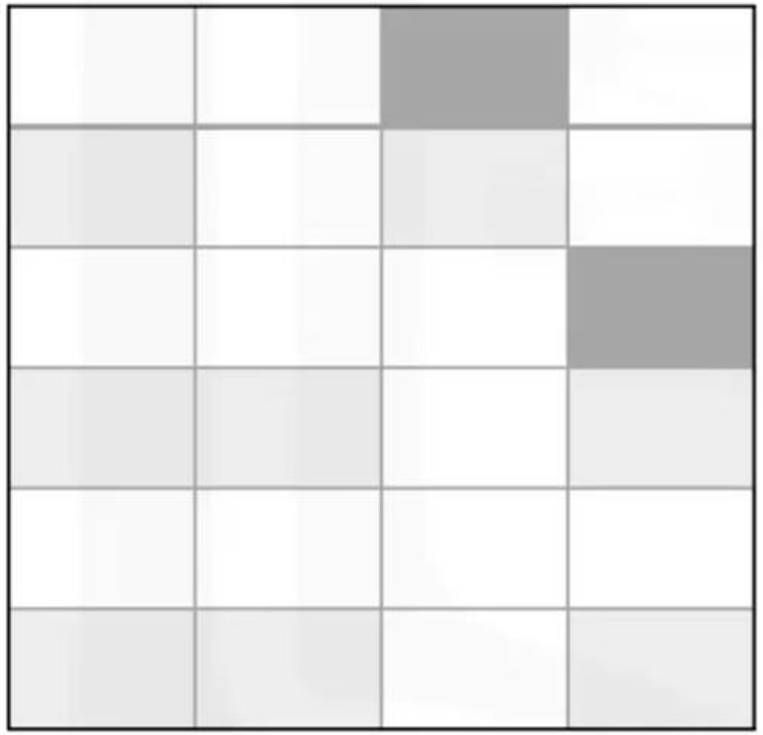
Column-Based NoSQL Databases

Column-Based NoSQL Database Architecture



- Spawned from Google's 'Bigtable'
- a.k.a. Bigtable clones or Columnar or Wide-Column databases
- Store data in columns or groups of columns

Column-Based NoSQL Database Architecture



- Column ‘families’ are several rows, with unique keys, belonging to one or more columns
  - Grouped in families as often accessed together
- Rows in a column family are not required to share the same columns
  - Can share all, a subset, or none
  - Columns can be added to any number of rows, or not

Column-Based NoSQL Database Use Cases

Suitable Use Cases

- Great for large amounts of sparse data
- Column databases can handle being deployed across clusters of nodes
- Column databases can be used for event logging and blogs
- Counters are a unique use case for column databases
- Columns can have a TTL parameter, making them useful for data with an expiration value

Column-Based NoSQL Database Use Cases

Unsuitable Use Cases

- For traditional ACID transactions
  - Reads and writes are only atomic at the row level
- In early development, query patterns may change and require numerous changes to column-based databases
  - Can be costly and can slow down the production timeline

Column-Based NoSQL Database Examples

Cassandra

HYPERTABLE INC

APACHE HBASE

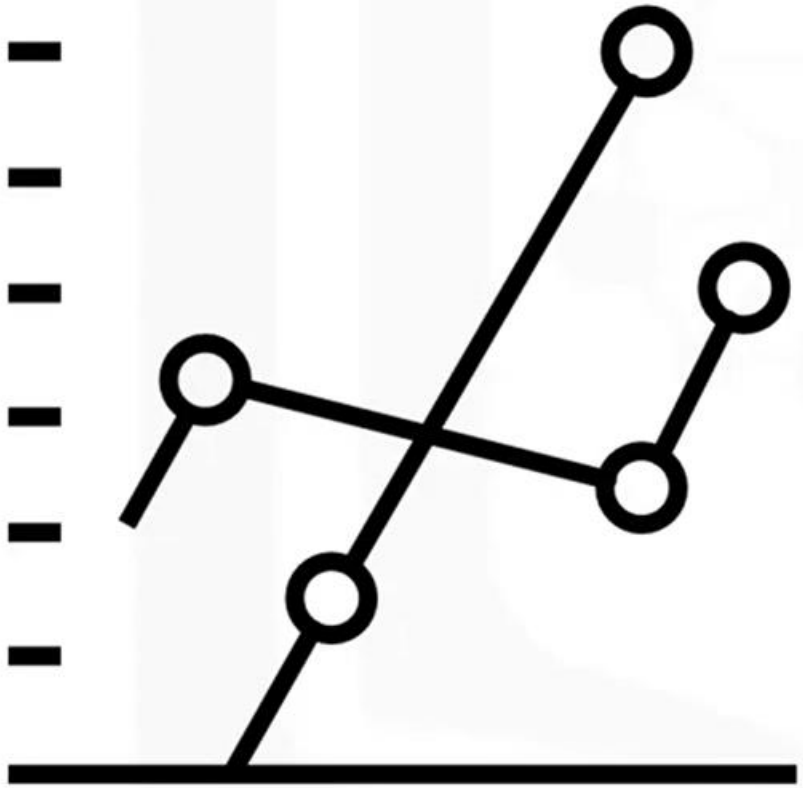
accumulo™

Recap: Column-Based NoSQL Databases

- Column-based databases were spawned from the architecture of Google's Bigtable storage system
- Column-based databases store data in columns or groups of columns
- Column 'families' are several rows, with unique keys, belonging to one or more columns
- The primary use cases for Column-based databases are event logging, blogs, counters, and data with expiration values

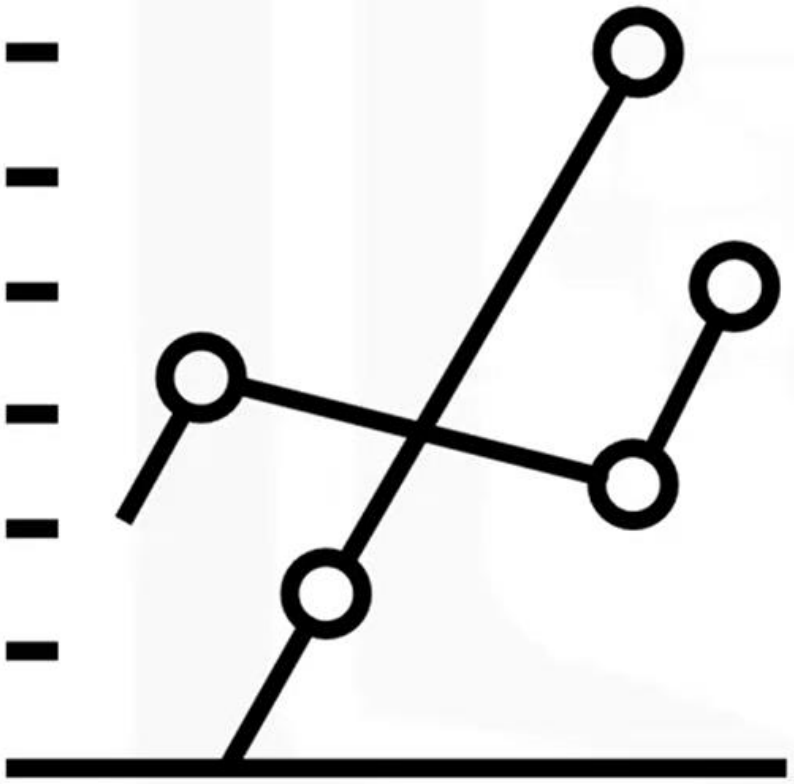
Graph NoSQL Databases

Graph NoSQL Database Architecture



- Graph databases store information in entities (or nodes), and relationships (or edges
- Graph databases are impressive when your data set resembles a graph-like data structure

Graph NoSQL Database Architecture



- Graph databases do not shard well
  - Traversing a graph with nodes split across multiple servers can become difficult and hurt performance
- Graph databases are ACID transaction compliant
  - Unlike other NoSQL databases discussed

Graph NoSQL Database Use Cases

Suitable Use Cases

- For highly connected and related data
- Social networking
- Routing, spatial, and map apps
- Recommendation engines

Graph NoSQL Database Use Cases

Unsuitable Use Cases

- When looking for advantages offered by other NoSQL database categories
- When an application needs to scale horizontally
  - You will quickly reach the limitations associated with these types of data stores
- When trying to update all or a subset of nodes with a given parameter
  - These types of operations can prove to be difficult and non-trivial


neo4j


OrientDB


ArangoDB


Amazon Neptune


amazon
web services™


APACHE
GIRAPH


JanusGraph

Recap: Graph NoSQL Databases

- Graph databases store information in entities and relationships
- Graph databases are impressive when your data set resembles a graph-like data structure
- Graph databases don’t shard well but are ACID transaction compliant
- The primary use cases for Graph databases are for highly connected and related data, for social networking sites, for routing, spatial and map applications, and for recommendation engines