

# Getting Started with MongoDB

---

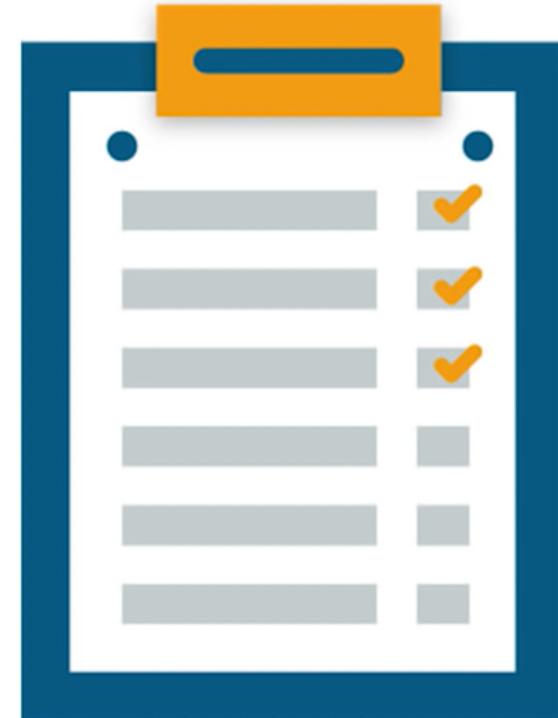


# Today's Training Objectives

---

After completing this module, you should be able to:

- Differentiate between the various categories of database
- Understand the goals and design of MongoDB
- List MongoDB tools
- Describe JSON and BSON
- Install MongoDB on Windows, Linux, Mac OS, etc.
- Setup MongoDB environment



# Database – The Need

---

1

Store, organize, and manage large amounts of data on a single platform

Facilitate a well-planned platform for data analysis

2

Promote a disciplined approach to data management

3

One stop solution to manage security, multiuser access control, backup & recovery, etc.

4

# Introduction to Database Categories

---

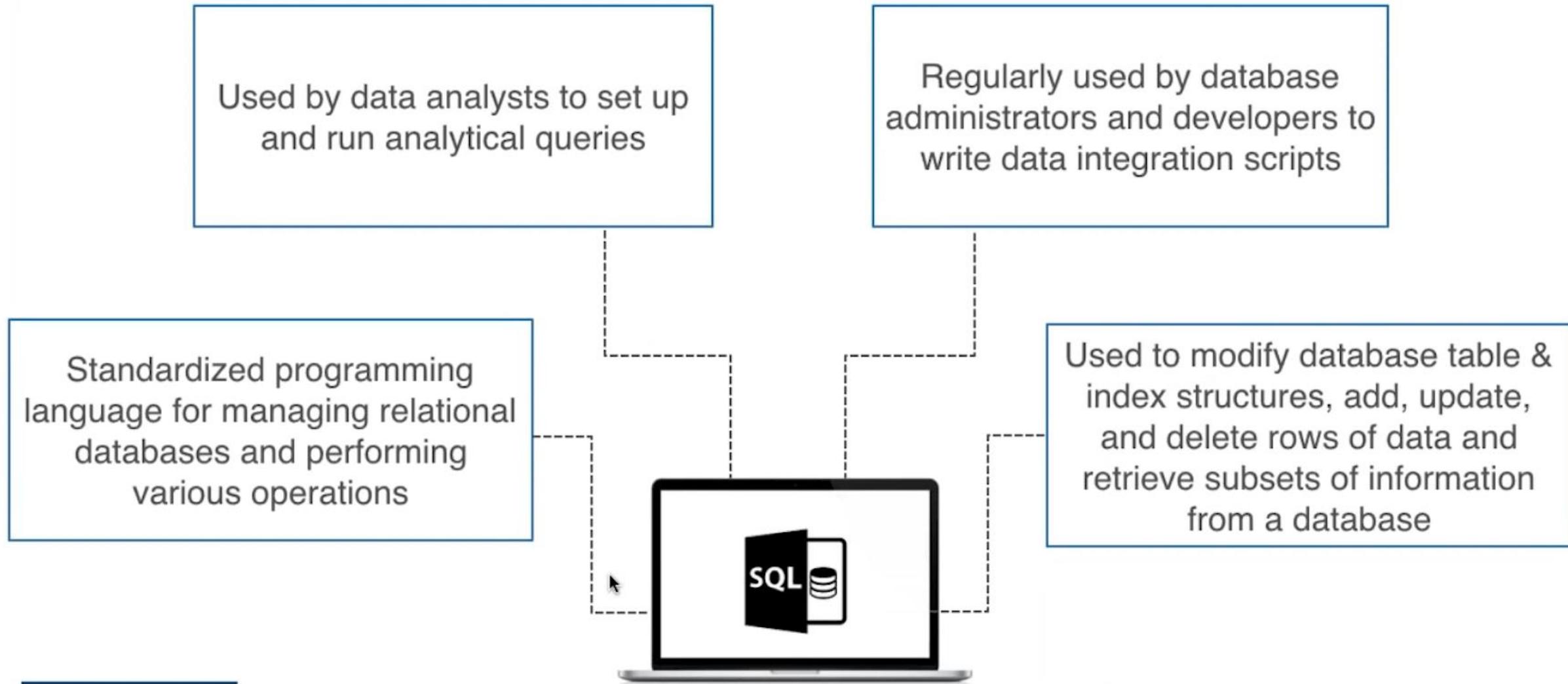
Categories	Also known as	Example 1	Example 2
OLTP	RDBMS/ Real Time	Oracle	MS SQL
OLAP	DSS/ DW	Netezza	Sap Hana
NewSQL	NoSQL/ Big Data	MongoDB	CouchDB

# Specifics of Database Categories

OLTP	OLAP	NewSQL
<ul style="list-style-type: none"><li>▪ Stores real-time data</li><li>▪ Heterogeneous in nature</li><li>▪ ATM/ Retail Transaction</li><li>▪ Short term storage</li></ul>	<ul style="list-style-type: none"><li>▪ Stores processed data</li><li>▪ Homogeneous in nature</li><li>▪ Access to multiple DBMS's</li><li>▪ Long term storage</li></ul>	<ul style="list-style-type: none"><li>▪ Stores all Application data</li><li>▪ Schema agnostic</li><li>▪ Handling Big Data requirements</li><li>▪ Long term storage</li></ul>

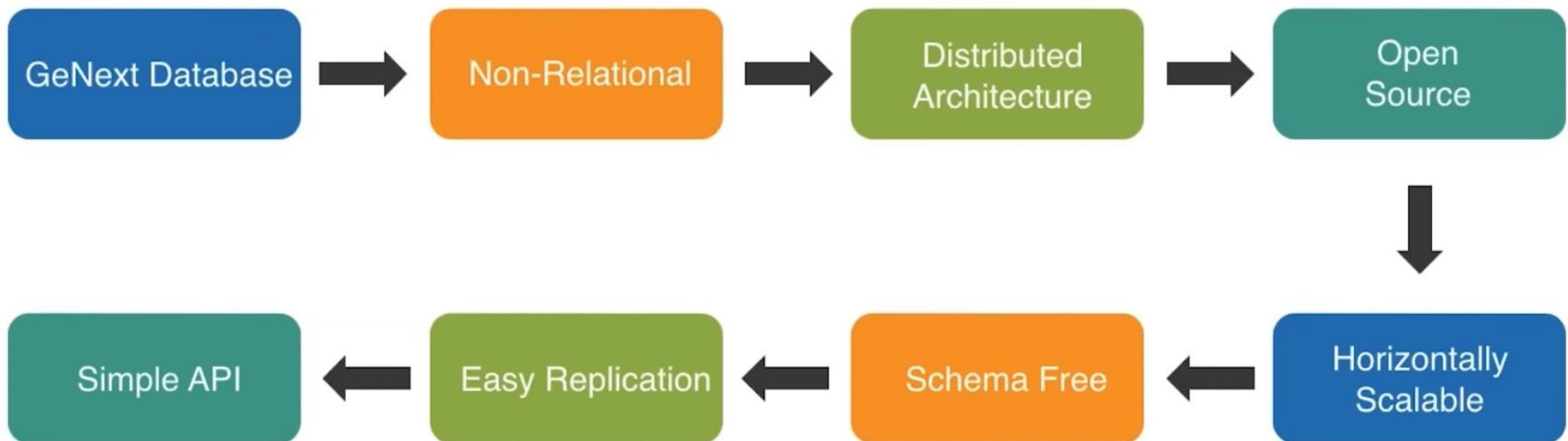
# SQL - Structured Query Language

---



# NoSQL – You don't need SQL

---



# SQL vs NOSQL

---

Entity	SQL Databases	NoSQL Databases
Type	One Type – SQL (with slight variations)	Multiple Types – Document, Key-Value, Tabular, Graph
Developed In	1970	2000
Examples	Oracle, MSSQL, DB2 etc.	MongoDB, Cassandra, HBase, Neo4J
Schemas	Fixed	Dynamic
Scaling	Vertical	Horizontal
Dev Model	Mix	Open Source
Properties Followed	ACID	BASE

# SQL:- ACID Property

---

## Atomicity

Entire transaction is either successful or fails to load completely, no partial execution

## Consistency

Integrity constraints are maintained within database to maintain consistency

## Isolation

Modification in a transaction will not be visible to any other transaction until the change is committed

## Durability

Committed data is persisted even after system failure as it is stored in non-volatile memory

# NoSQL Base Property

---

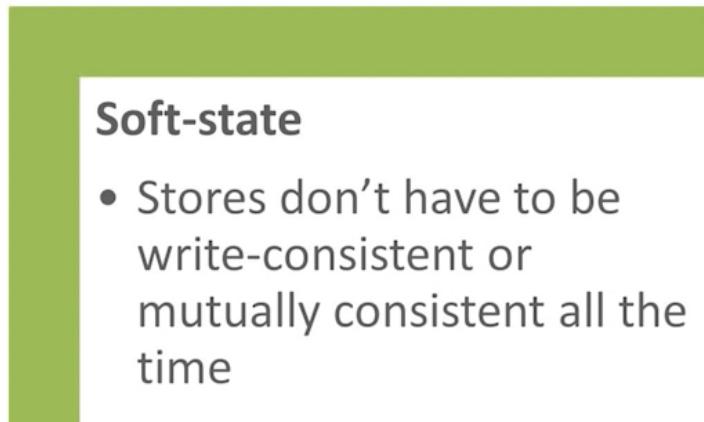
## Basic Availability

- The database appears to work most of the time



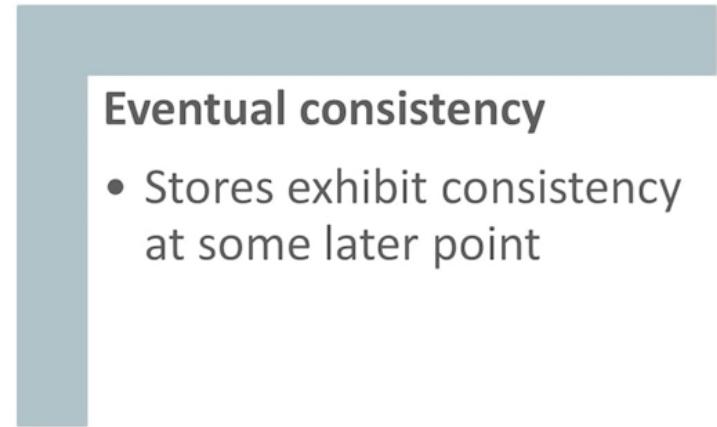
## Soft-state

- Stores don't have to be write-consistent or mutually consistent all the time



## Eventual consistency

- Stores exhibit consistency at some later point



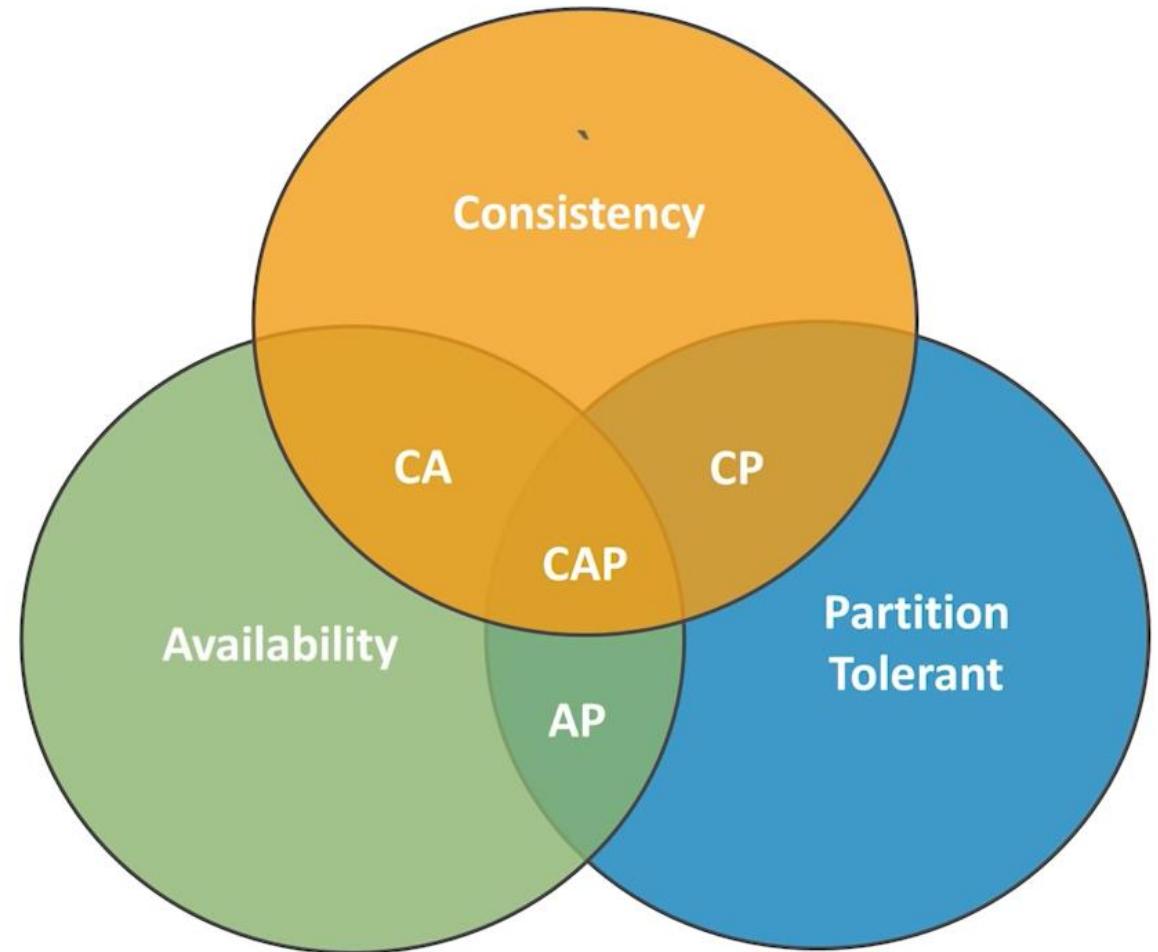
# The CAP Theory

Consistency	Availability	Partition Tolerance
<ul style="list-style-type: none"><li>This means that the data in the database remains consistent after the execution of an operation</li><li>For example, after an update operation, all clients see the same data</li></ul>	<ul style="list-style-type: none"><li>This means that the system will not have downtime (100% service uptime guaranteed)</li><li>Every node (if not failed) always executes query</li></ul>	<ul style="list-style-type: none"><li>This means that the system continues to function even when the communication between servers is unreliable</li><li>The servers may be partitioned into multiple groups that cannot communicate with one another</li></ul>

# CAP Combinations

---

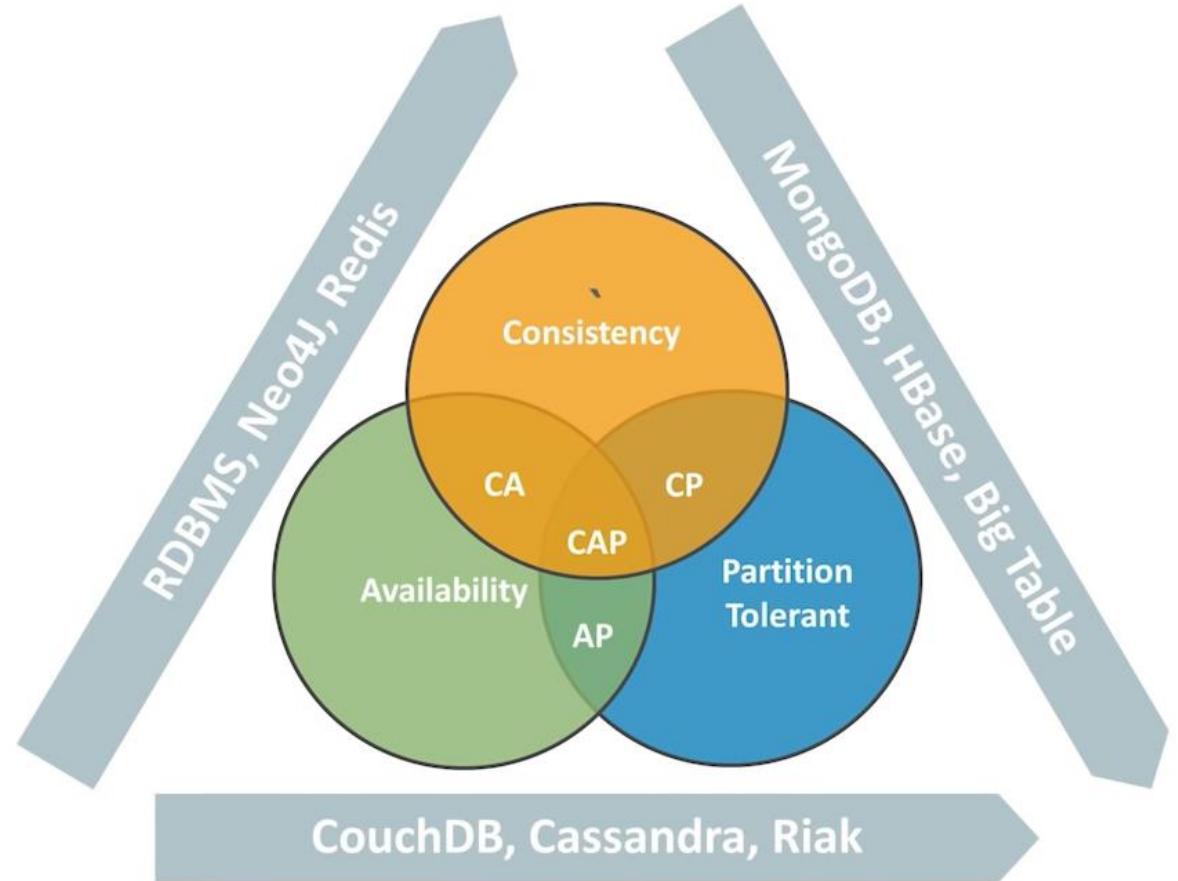
- Theoretically, it is not possible to fulfill all the 3 requirements
- CAP provides the basic requirement for a distributed system to follow 2 of the 3 requirements
- Hence, all the current NoSQL databases follow different combinations of C, A, P from the CAP theorem



# CAP – CA, CP, AP

---

- CA: Single site cluster, all nodes are always in contact
- CP: Some data may not be accessible, however, the rest is still consistent and accurate
- AP: System is available under partitioning, however, some of the data returned may be inaccurate



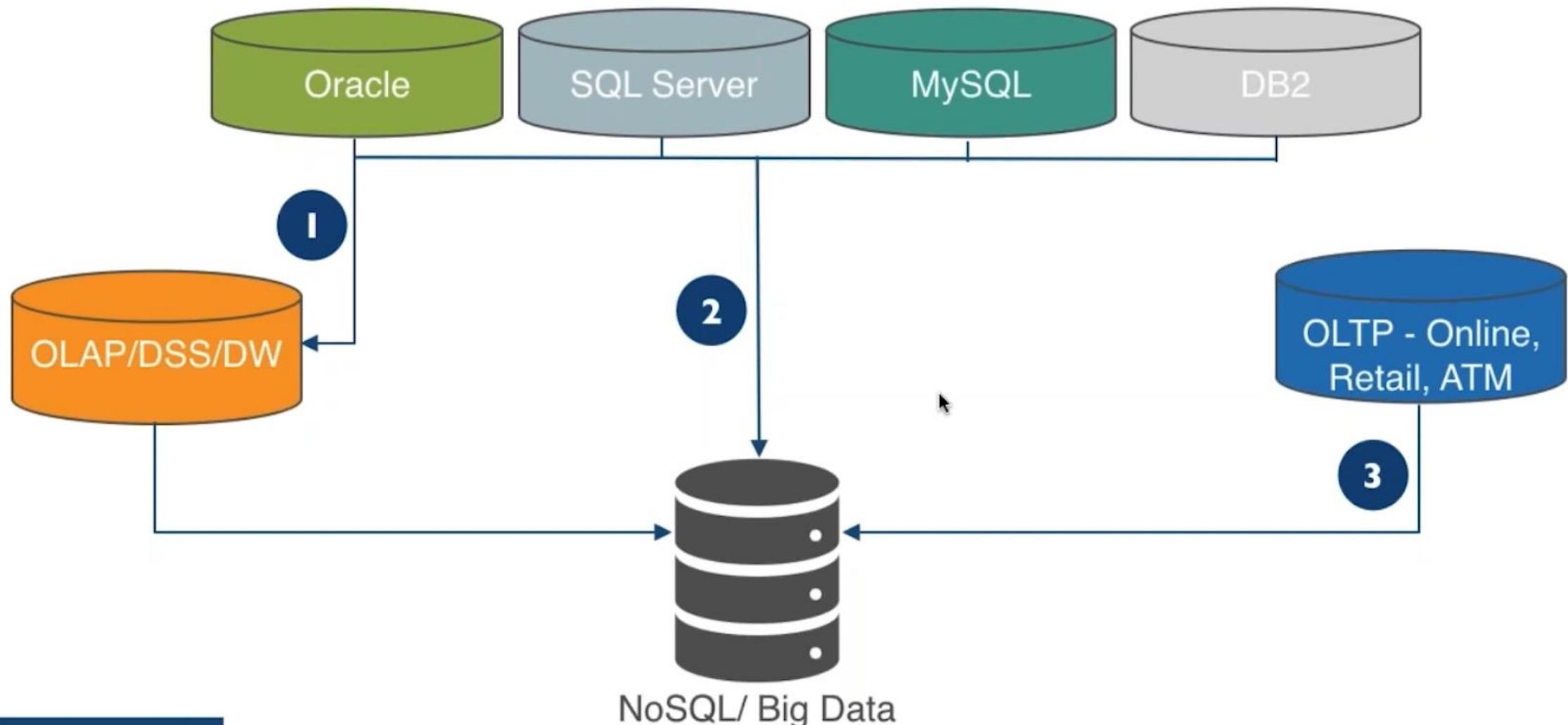
# Features of NoSQL Database

---

- Supports massive number of concurrent users – tens of thousands to millions
- Provides extremely responsive experience to a globally distributed base of users
- Always available with no downtime
- Quickly adapts to changing requirements with frequent updates and new features
- Handles semi and unstructured data

# NoSQL Database – Storage

---

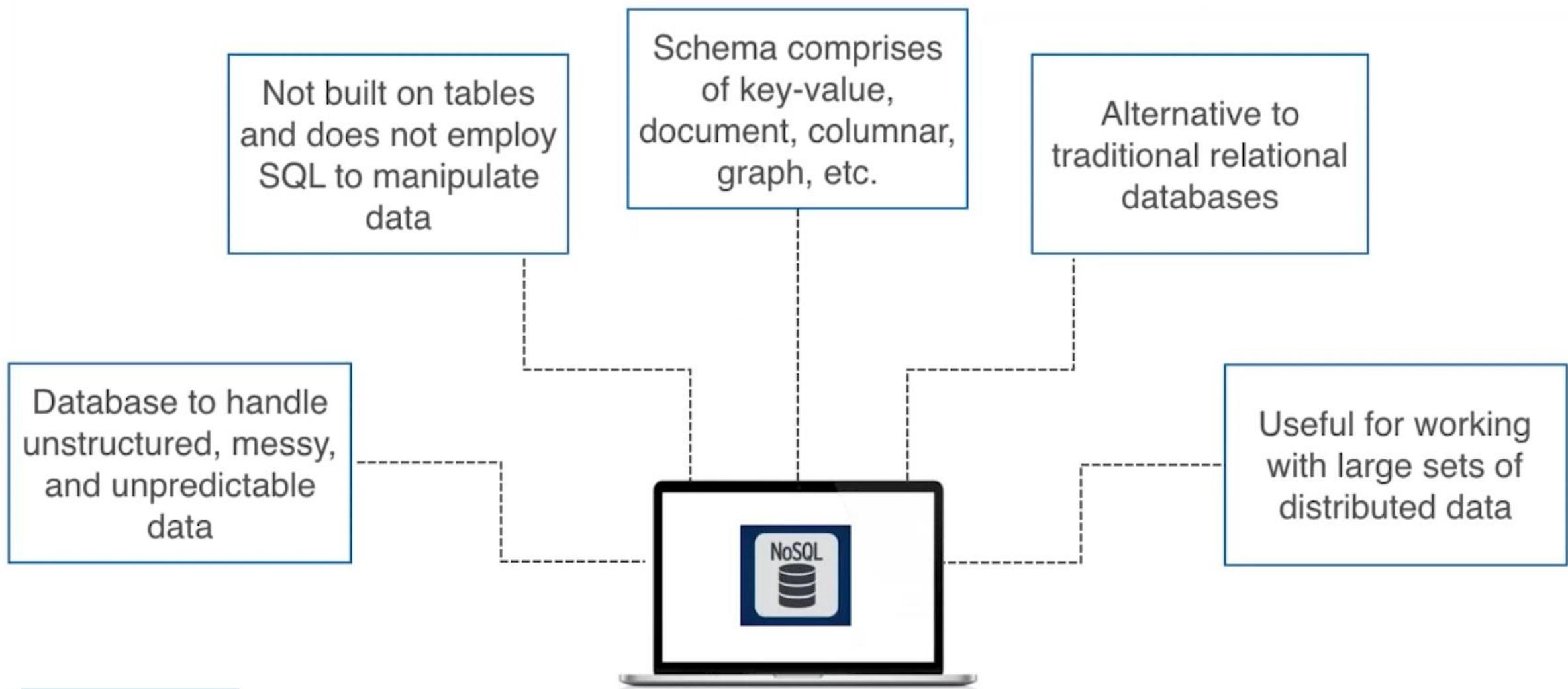


# Advantages of NoSQL - 1



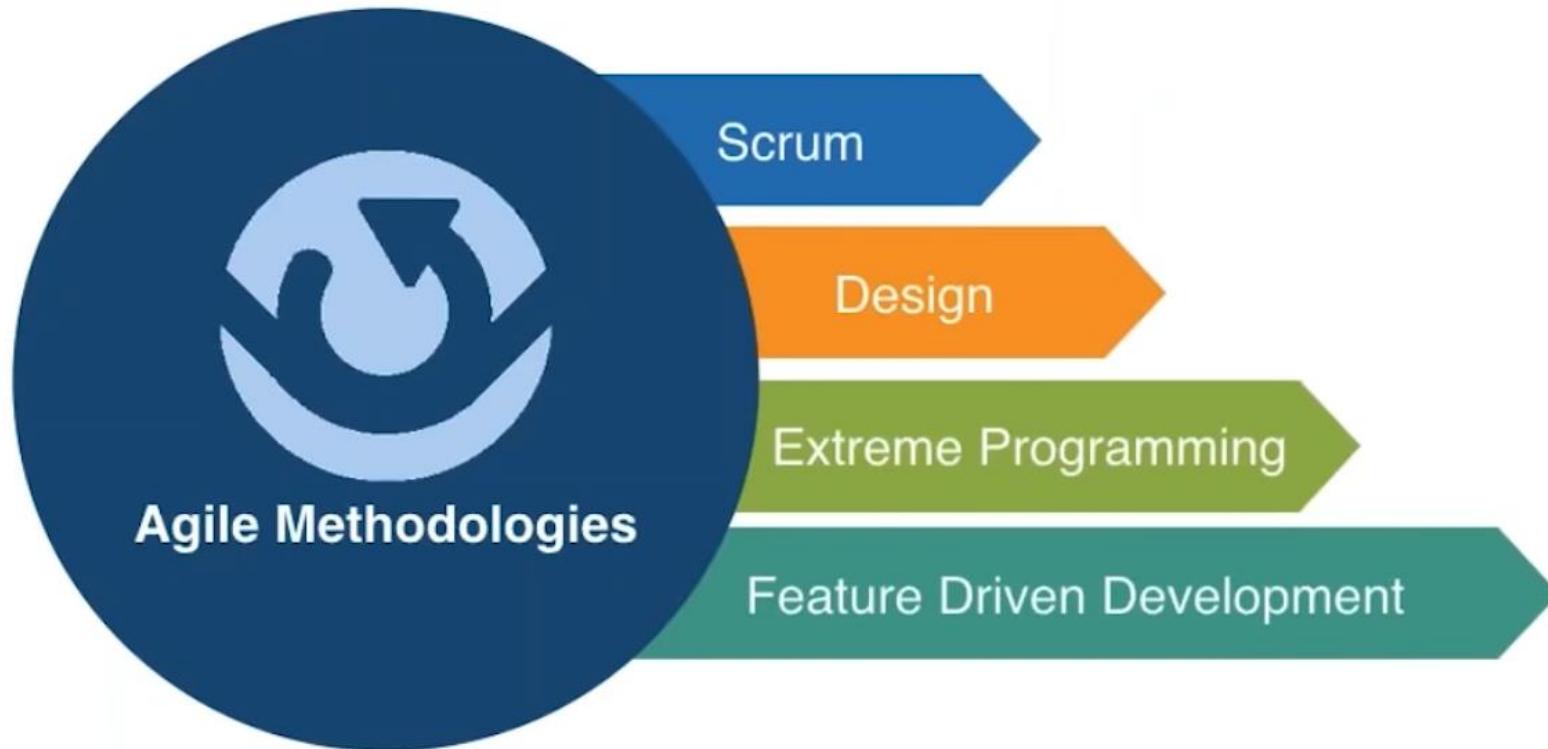
## Advantages of NoSQL - 2

---



## Advantages of NoSQL - 3

---



# Categories Of NoSQL Databases

## Document Base

- Pairs each key with a complex data structure known as a document
- Contains many different key-value pairs, or key-array pairs, or even nested documents

## Key Value Store

- Key-value stores are the simplest NoSQL databases.
- Every single item in the database is stored as an attribute name (or "key"), together with its value

## Graph Store

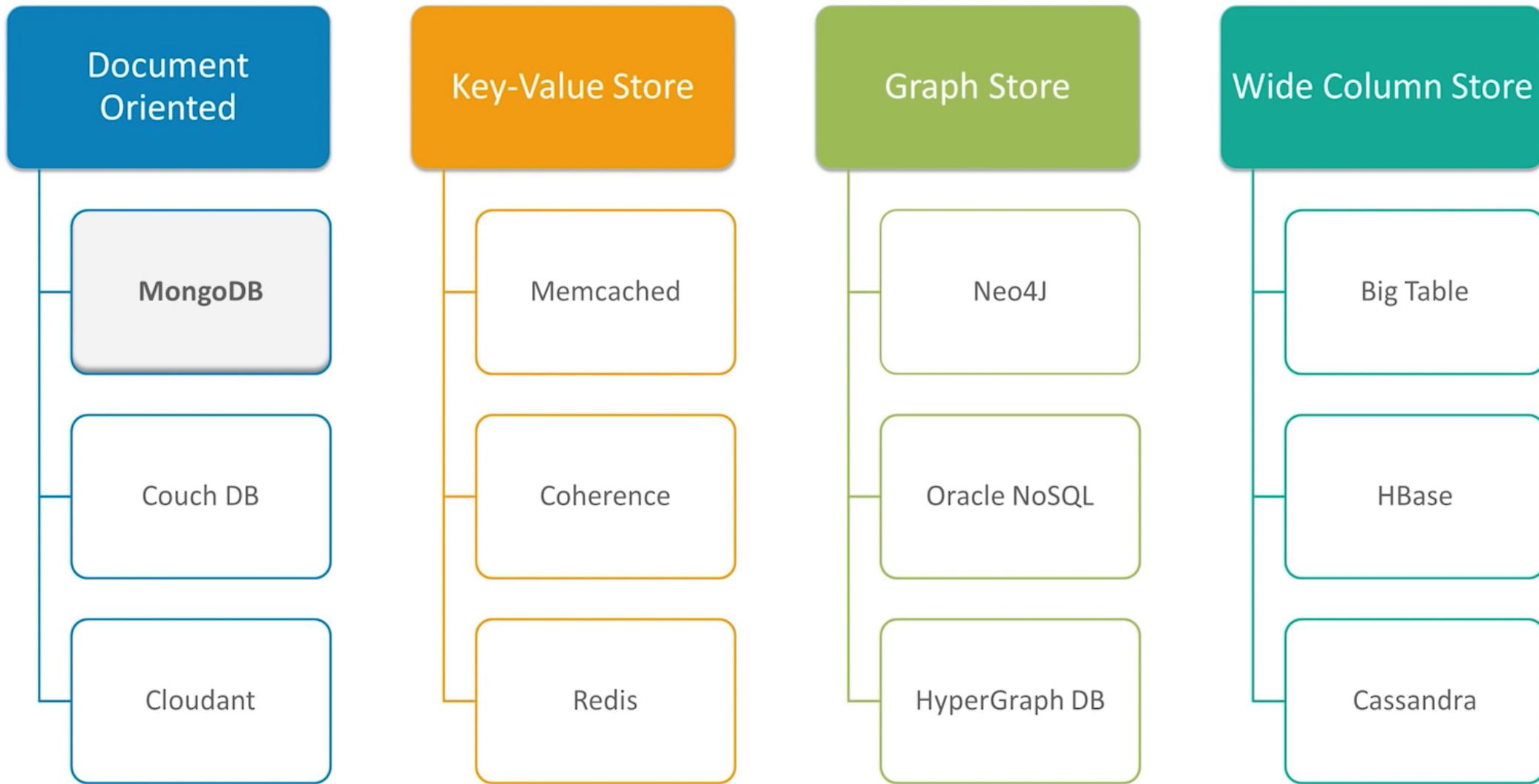
- Used to store information about networks, such as social connections.
- Graph stores include Neo4J and HyperGraphDB.

## Wide Column Store

- Cassandra and HBase are optimized for large dataset queries
- Stores columns of data together, instead of rows.

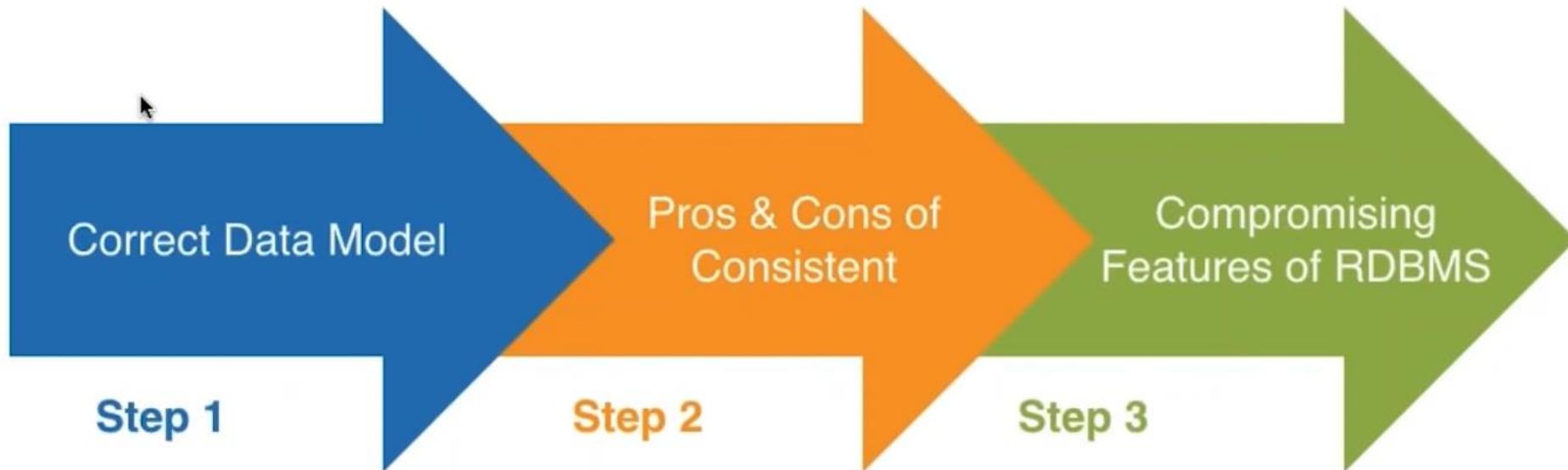
# Types Of NoSQL Databases

---



# NoSQL Database – Selection and Implementation

---



# **Introduction To MongoDB**

---

# Overview of MongoDB

---

- To avoid complex SQL queries with long processing time, you can shift to a NoSQL Database which is designed for ease of development and scaling i.e. MongoDB.
- MongoDB is an open source document database which is capable of handling big data.
- It works on the concept of collection and document.
- There is no concept of primary key and foreign key in the tables.
- It provides:



High Performance



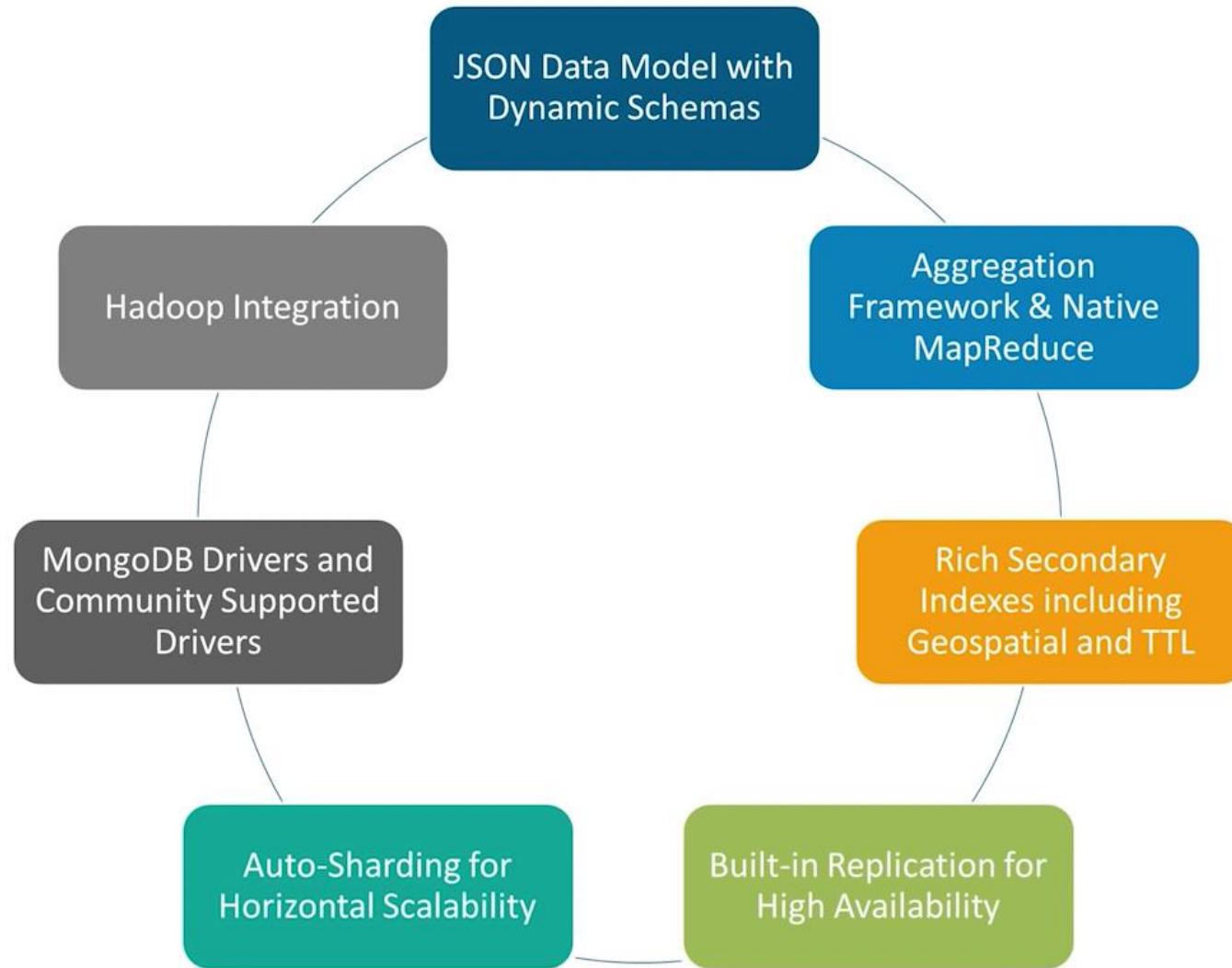
High Availability



Easy Scalability

# Features Of MongoDB

---



# Reasons to use MongoDB - 1

---

## I. Speed To Develop

- a) Change streams to create powerful data pipelines
- b) Shard-aware secondary reads ensure data consistency from any secondary
- c) Fully expressive array updates allow you to perform complex array manipulations
- d) Retryable writes reduce error handling

## 2. Speed To Scale

- a) Operational simplicity making you 10x to 20x more productive
- b) Data governance with JSON schema lets you precisely control document structures
- c) Most advanced security control with comprehensive access controls, end-to-end encryption, and auditing
- d) Offers greater network, memory, and storage efficiency than almost any other database, thus saving you time, money, and precious platform resources

# Reasons to use MongoDB - II

---

## 3. Speed To Insight

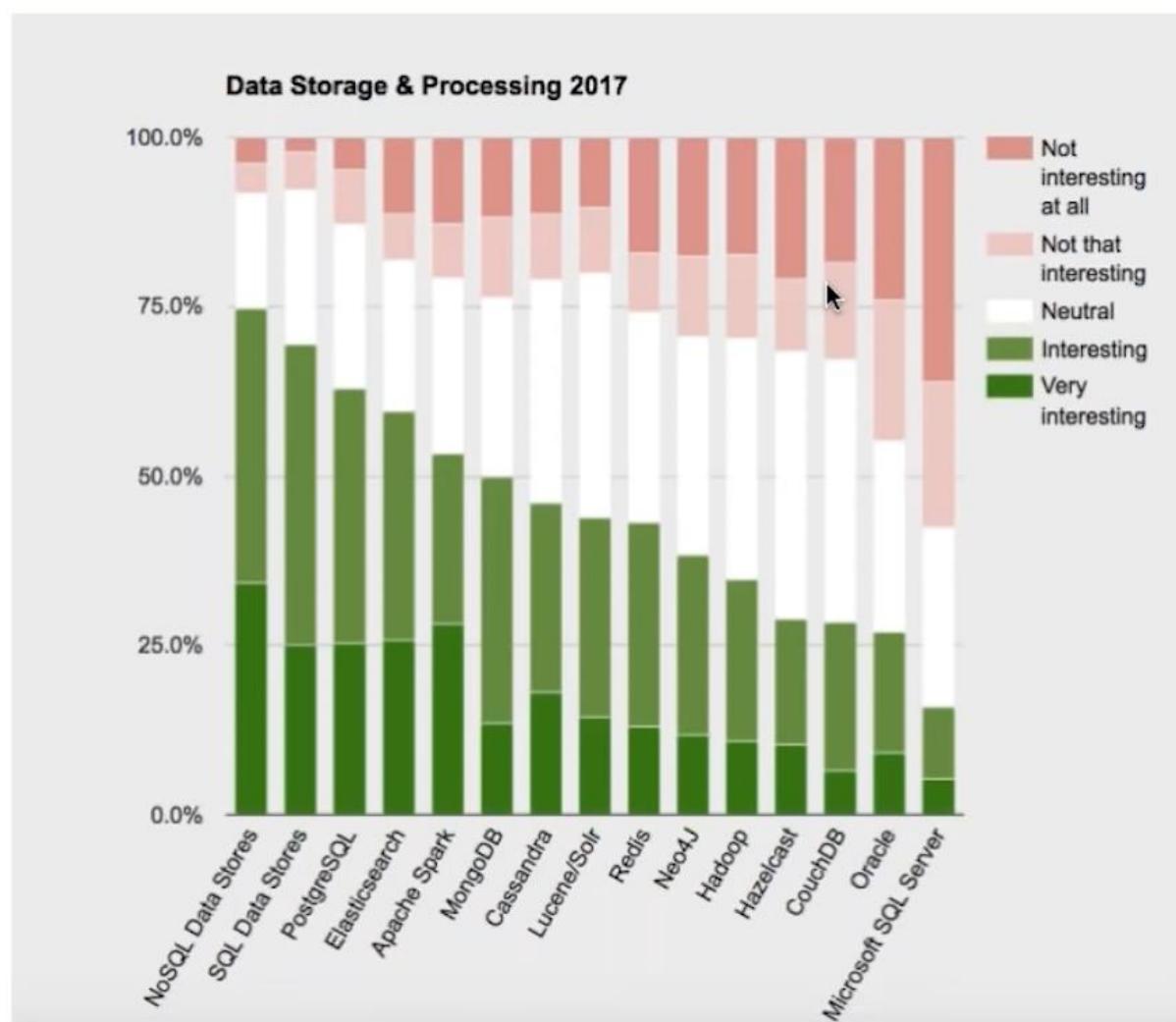
- a) Richer aggregation pipeline enables you to unlock new classes of analytics with less client-side code
- b) MongoDB Connector for BI provides integration with all leading SQL-based BI platforms such as Tableau, QlikView, SAP Business Objects, etc.
- c) R driver for MongoDB provides you with idiomatic, native language access to MongoDB to simplify and accelerate the speed of statistical analysis and data mining

## 4. Run Anywhere

- a) Globally distributed MongoDB service through MongoDB Atlas
- b) Best way to run MongoDB is, in the public cloud
- c) Automatic storage scaling reduces DevOps overhead by provisioning additional cluster storage capacity when it's needed

# Latest Market Trends in MongoDB

- NoSQL Market is expected to garner \$4.2 billion by 2020, registering a CAGR of 35.1% during the forecast period 2014-2020.
- MongoDB is the leading NoSQL database, with significant adoption among the Fortune 500 and Global 500.
- Software developers favor MongoDB's NoSQL database over most of the other databases, according to a recent survey by Stack Overflow of 64,000 developers.



# Few Customers of MongoDB

---



# Sectors using MongoDB - 1



Risk Analytics and Reporting

Reference Data & Market Data Management

Portfolio Management & Order Capture

Time Series Data

Surveillance Data Aggregation

Crime Data Management and Analytics

Citizen Engagement Platform

Program Data and Healthcare Record  
Management



## Sectors using MongoDB - II



**Content Management and Delivery**

**User Data and Digital Asset Management**

**Mobile and Social Apps**

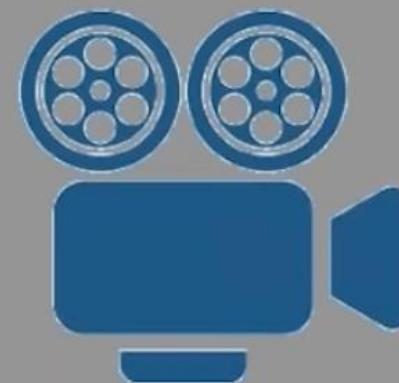
**Content Archiving**

**360-Degree Patient View**

**Population Management for At-Risk Demographics**

**Lab Data Management and Analytics**

**Mobile Apps for Doctors and Nurses**



# Sectors using MongoDB - III



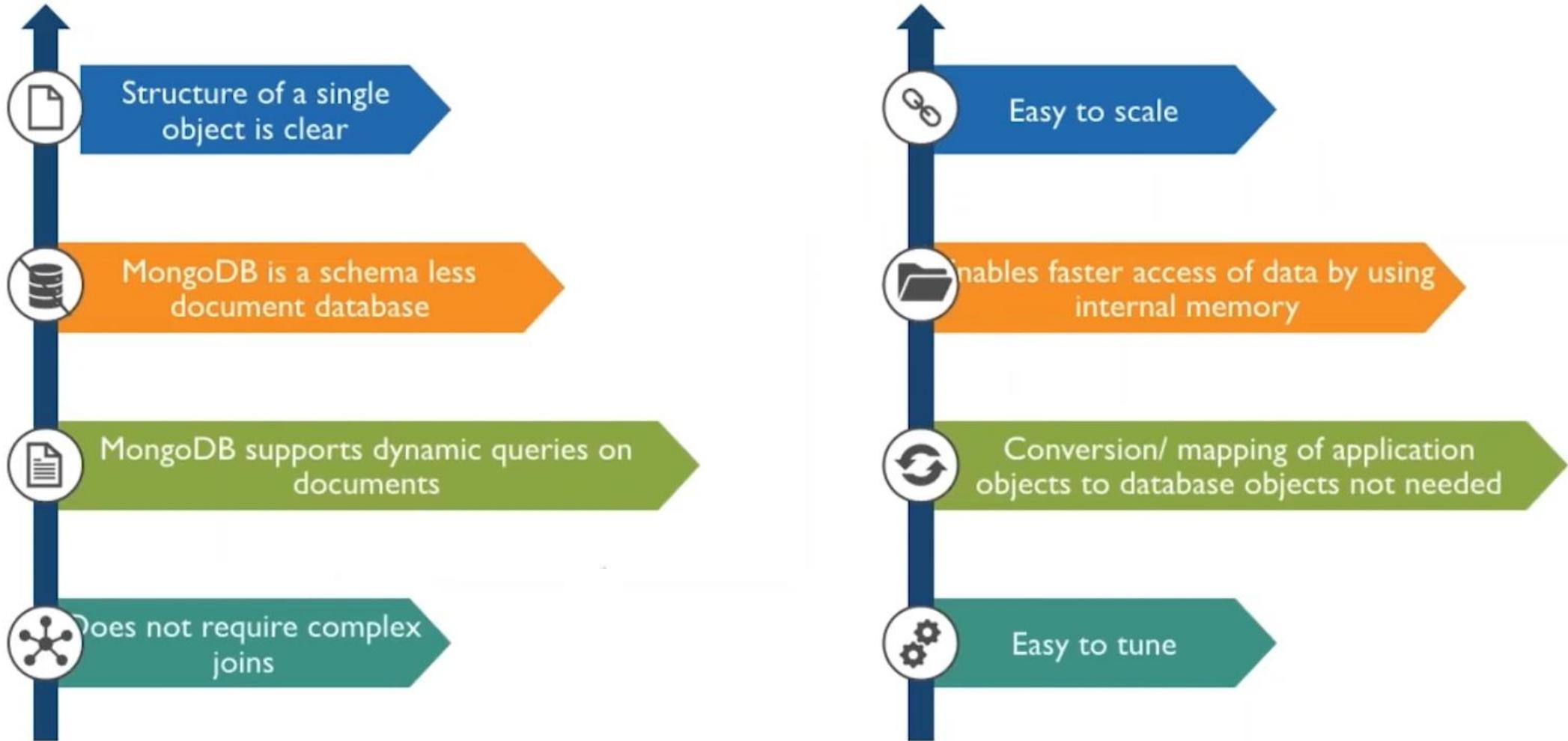
**Consumer Cloud and Product Catalog**  
**Customer Service Improvement**  
**Machine-to-Machine (M2M) Platform**  
**Real-Time Network Analysis and Optimization**

**Rich Product Catalogs**  
**Customer Data Management**  
**New Services and Digital Coupons**  
**Real-Time Price Optimization**



## Advantages of using MongoDB

---



# **MongoDB Tools And Terminologies**

---

# Mongo Database

---

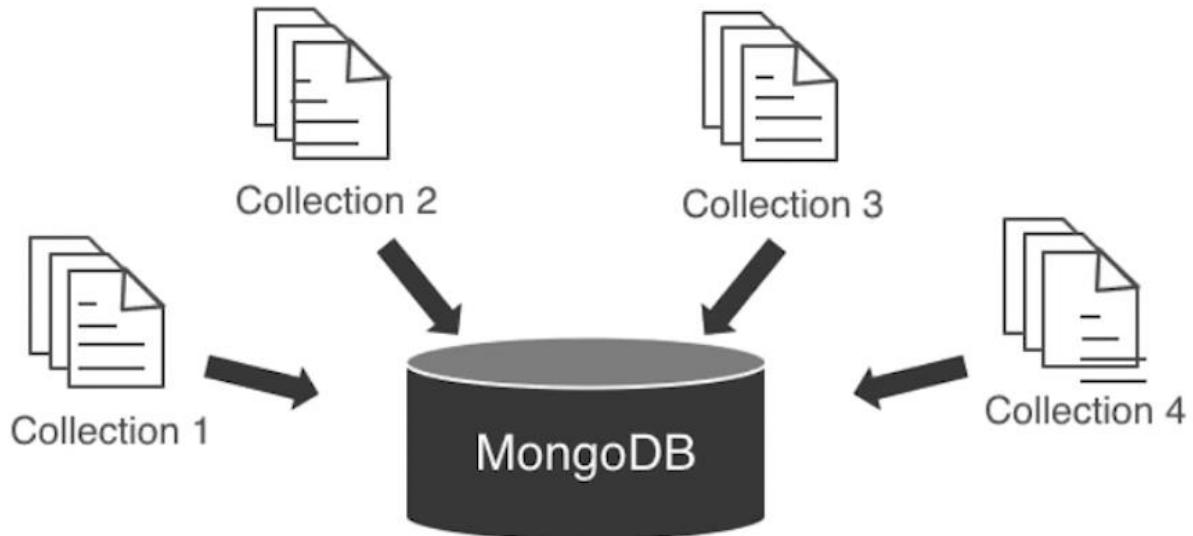
- Database is a physical container for collections
- Each database gets its own set of files on the file system
- A single MongoDB server has multiple databases
- Mongod is the primary resource for the MongoDB system
- It handles data requests, manages data format, and performs background management operations



# MongoDB Collection

---

- Collection is a group of similar or related purpose MongoDB documents within a single database
- MongoDB collection is equivalent to a RDBMS table
- Collections do not enforce a schema
- Documents within a collection can have different fields



## RDBMS Terminology and MongoDB

---

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
Column/ Attribute/ Variable	Field
Table Join	Embedded Documents
Database Server and Client	
Primary Key	Primary Key (Default key _id provided by mongodb itself)
Mysqld/Oracle	mongod
mysql/sqlplus	mongo

# Mongo DB tools

---

## Core Processes

- mongod
- mongos
- mongo

## Data Import and Export

- mongoimport
- mongoexport

## Diagnostic Tools

- mongostat
- mongotop
- mongosniff
- mongoperf

## Windows Services

- mongod.exe
- mongos.exe

## Binary Import and Export

- mongodump
- mongorestore
- bsondump
- mongooplog

## GridFS

- mongofiles

# **Introduction To JSON And BSON**

---

## JavaScript Object Notation (JSON)

---

JSON is a

- Lightweight, text-based, open standard format
- Syntax for storing and exchanging data
- Text, written with JavaScript object notation

Conventions used by JSON include C, C++, Java, Python, Perl, so on.



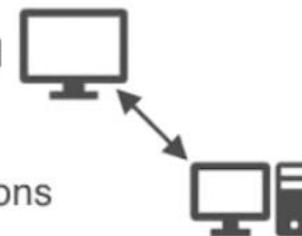
Web services and APIs use JSON format to provide public data



It is used while writing JavaScript based applications that include browser extensions and websites

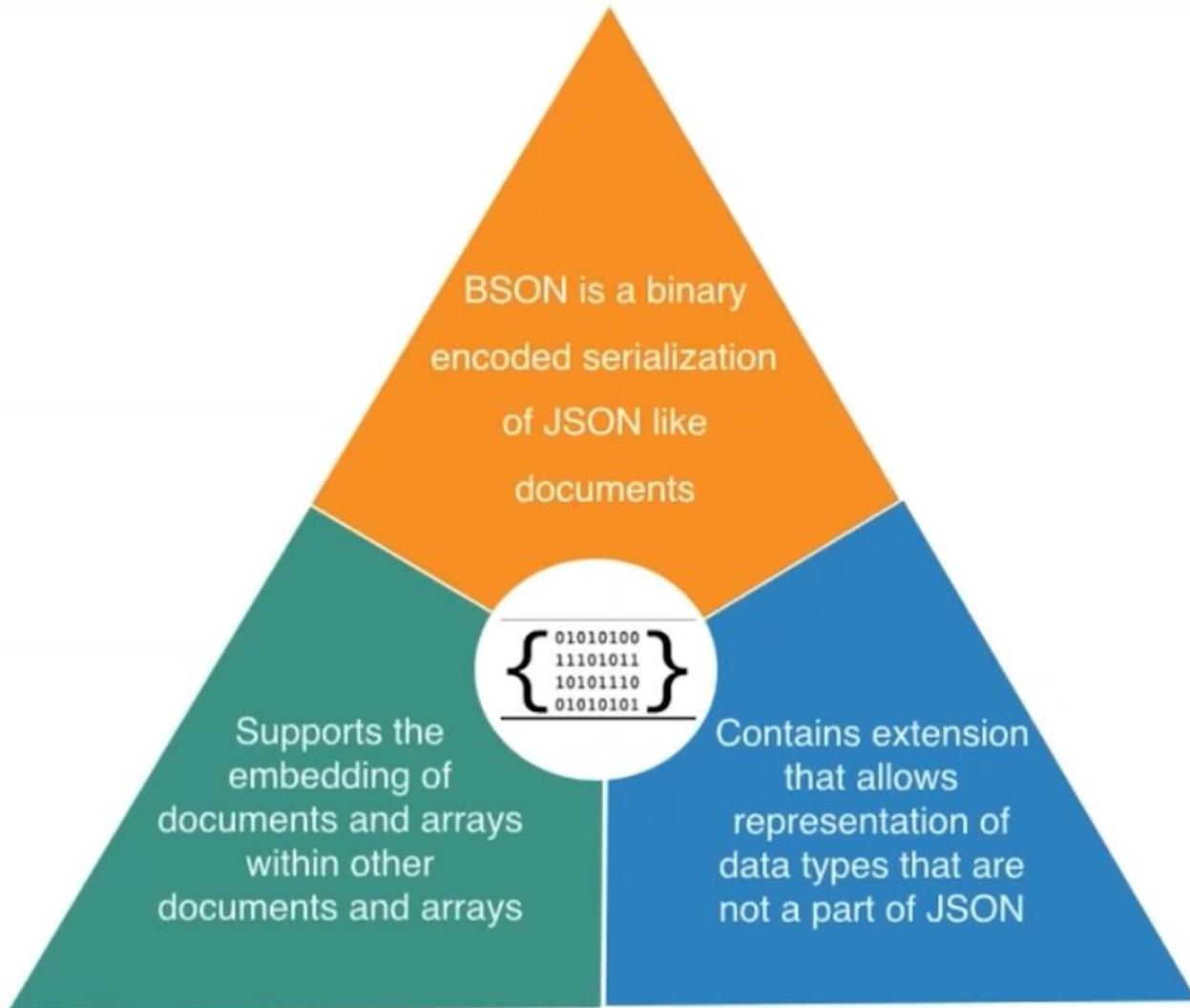


It is primarily used to transmit data between a server and web applications



# Binary JSON (BSON)

---



# Characteristics of BSON

---

**BSON** is designed to have the following characteristics:

**Lightweight**

Optimizing spatial overhead is important for any data representation format, specifically when used over the network

**Traversable**

For primary data representation, BSON can be used, which traverses data easily

**Efficient**

Encoding data to BSON and decoding from BSON can be performed quickly in most languages due to the use of 'C' data types

## JSON vs. BSON

JSON:

```
{“hello”: “world”}
```

## BSON:

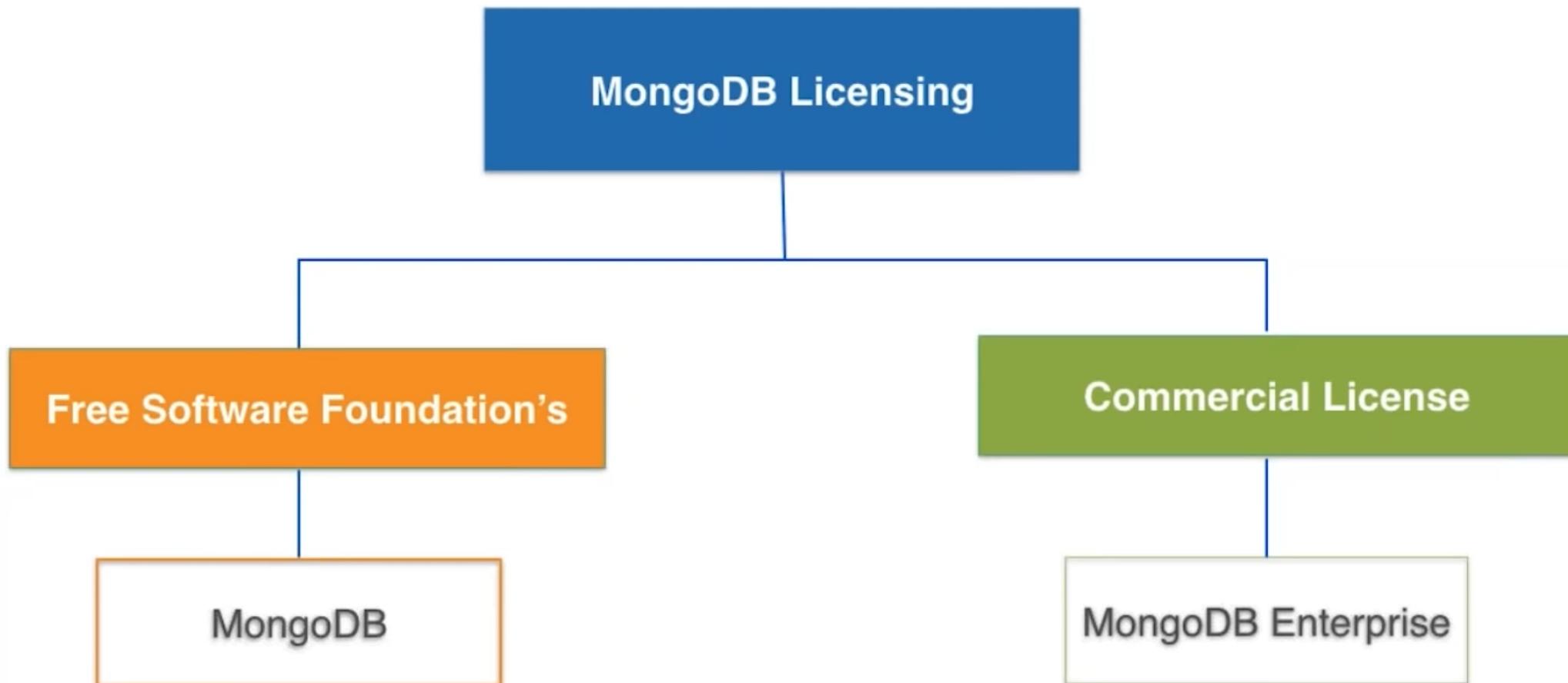
```
"\x16\x00\x00\x00\x02 hello\x00  
\x06\x00\x00\x00 world\x00\x00"
```

```
{"BSON": ["awesome", 5.05, 1986]}
```

1

# MongoDB Licensing

---



# MongoDB Enterprise

---

- MongoDB Enterprise is the commercial edition of MongoDB that provides enterprise-grade capabilities
- MongoDB Enterprise includes advanced **security features, management tools, software integration** and **certification**
- These value-added capabilities are not included in the open-source edition of MongoDB



# MongoDB Enterprise Includes

---

- Advanced Security
- Management Support
- Certified OS Support
- Enterprise Software Integration
- On-demand Training



**THANK YOU!**

