**Key Value Pair Based**

Data is stored in key/value pairs. It is designed in such a way to handle lots of data and heavy load.

Key-value pair storage databases store data as a hash table where each key is unique, and the value can be a JSON, BLOB(Binary Large Objects), string, etc.

For example, a key-value pair may contain a key like “Website” associated with a value like “Google”.

Table

Description automatically generated

It is one of the most basic NoSQL database example. This kind of NoSQL database is used as a collection, dictionaries, associative arrays, etc. Key value stores help the developer to store schema-less data. They work best for shopping cart contents.

Redis, Dynamo, Riak are some NoSQL examples of key-value store DataBases. They are all based on Amazon’s Dynamo paper.

### Column-based

Column-oriented databases work on columns and are based on BigTable paper by Google. Every column is treated separately. Values of single column databases are stored contiguously.

Table

Description automatically generated

Column based NoSQL database

They deliver high performance on aggregation queries like SUM, COUNT, AVG, MIN etc. as the data is readily available in a column.

Column-based NoSQL databases are widely used to manage data warehouses, [business intelligence](https://www.guru99.com/business-intelligence-definition-example.html), CRM(Customer relationship management), Library card catalogs,

HBase, Cassandra, Hypertable are NoSQL query examples of column based database.

### Document-Oriented:

Document-Oriented NoSQL DB stores and retrieves data as a key value pair but the value part is stored as a document. The document is stored in JSON or XML formats. The value is understood by the DB and can be queried.

A picture containing table

Description automatically generated

Relational Vs. Document

In this diagram on your left you can see we have rows and columns, and in the right, we have a document database which has a similar structure to JSON. Now for the relational database, you have to know what columns you have and so on. However, for a document database, you have data store like JSON object. You do not require to define which make it flexible.

The document type is mostly used for CMS systems, blogging platforms, real-time analytics & e-commerce applications. It should not use for complex transactions which require multiple operations or queries against varying aggregate structures.

Amazon SimpleDB, CouchDB, MongoDB, Riak, Lotus Notes, MongoDB, are popular Document originated DBMS systems.

### Graph-Based

A graph type database stores entities as well the relations amongst those entities. The entity is stored as a node with the relationship as edges. An edge gives a relationship between nodes. Every node and edge has a unique identifier.

Diagram

Description automatically generated

Compared to a relational database where tables are loosely connected, a Graph database is a multi-relational in nature. Traversing relationship is fast as they are already captured into the DB, and there is no need to calculate them.

Graph base database mostly used for social networks, logistics, spatial data.

Neo4J, Infinite Graph, OrientDB, FlockDB are some popular graph-based databases.

## Query Mechanism tools for NoSQL

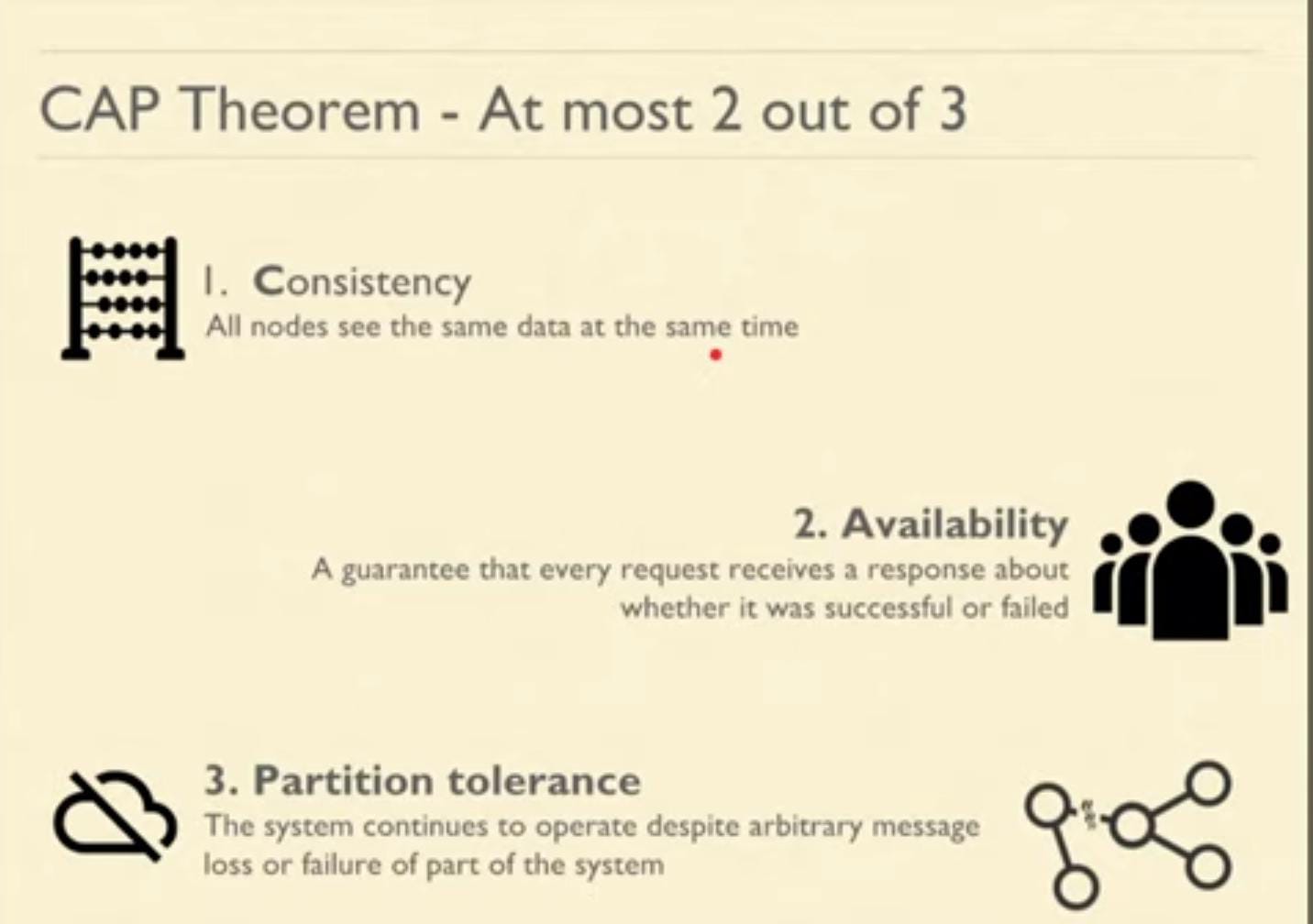
The most common data retrieval mechanism is the REST-based retrieval of a value based on its key/ID with GET resource

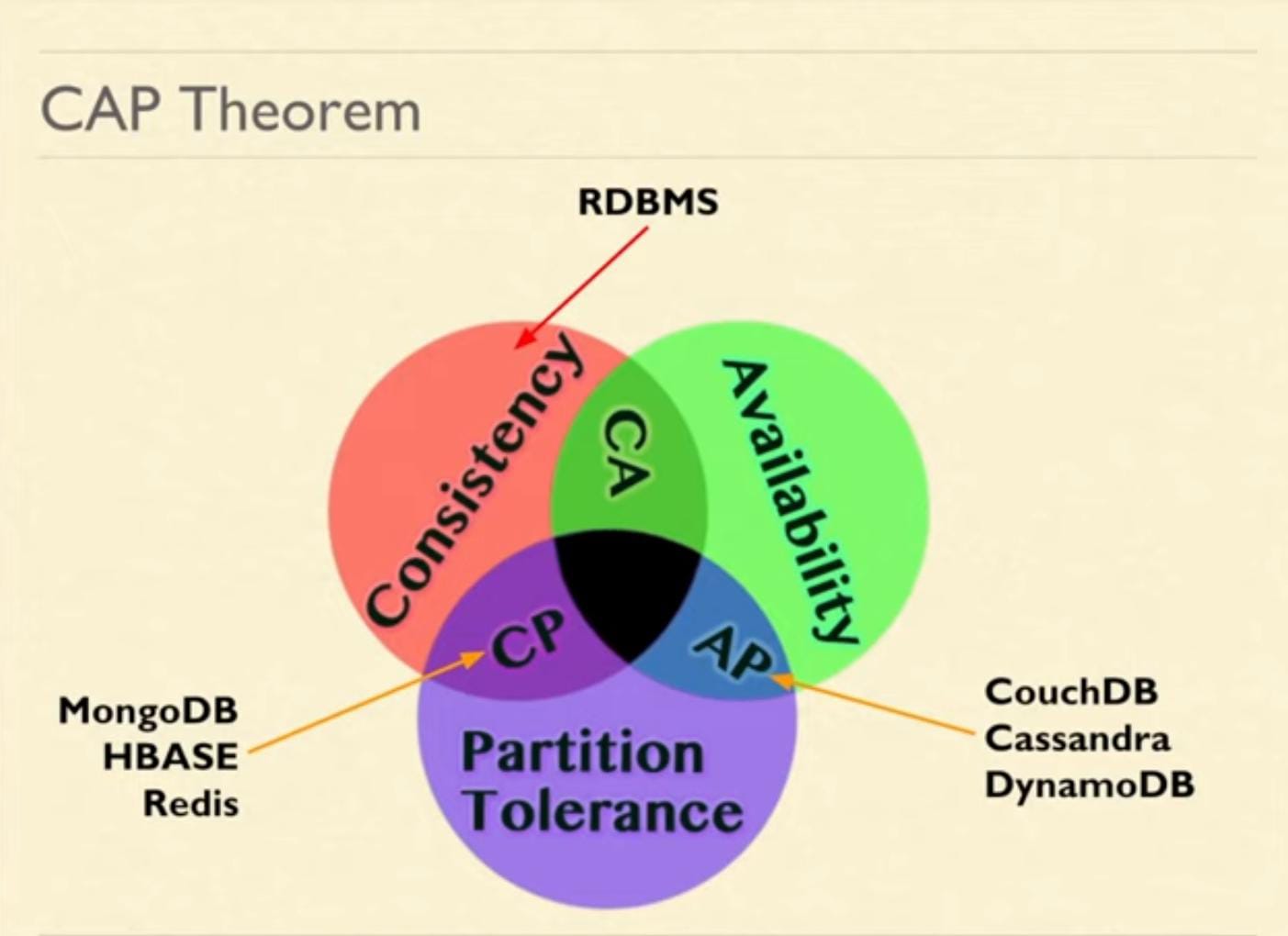
Document store Database offers more difficult queries as they understand the value in a key-value pair. For example, CouchDB allows defining views with MapReduce

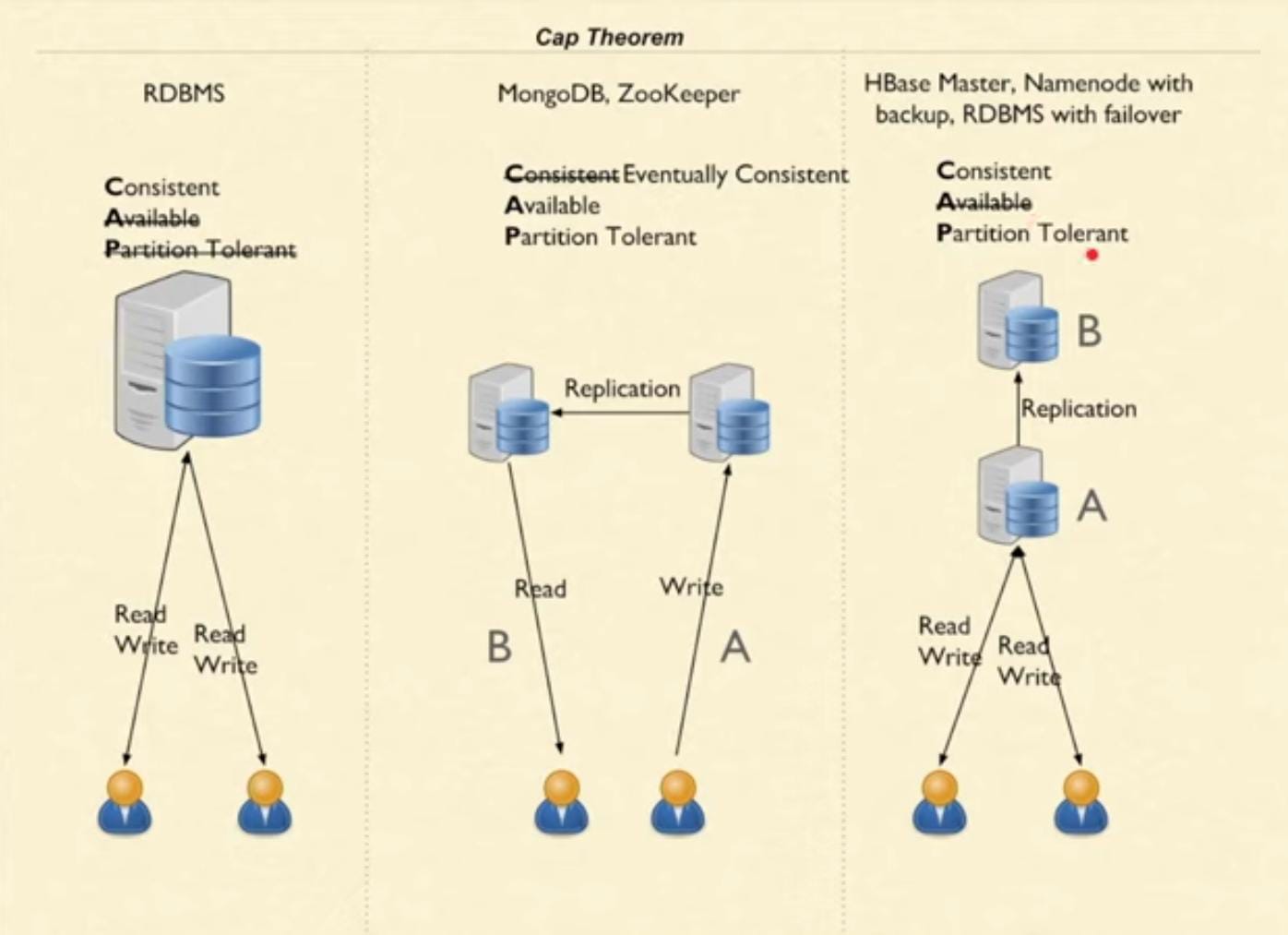
## **What is the CAP Theorem?**

CAP theorem is also called brewer’s theorem. It states that is impossible for a distributed data store to offer more than two out of three guarantees

1. Consistency
2. Availability
3. Partition Tolerance







**Consistency:**

The data should remain consistent even after the execution of an operation. This means once data is written, any future read request should contain that data. For example, after updating the order status, all the clients should be able to see the same data.

**Availability:**

The database should always be available and responsive. It should not have any downtime.

**Partition Tolerance:**

Partition Tolerance means that the system should continue to function even if the communication among the servers is not stable. For example, the servers can be partitioned into multiple groups which may not communicate with each other. Here, if part of the database is unavailable, other parts are always unaffected.

## Eventual Consistency

The term “eventual consistency” means to have copies of data on multiple machines to get high availability and scalability. Thus, changes made to any data item on one machine has to be propagated to other replicas.

Data replication may not be instantaneous as some copies will be updated immediately while others in due course of time. These copies may be mutually, but in due course of time, they become consistent. Hence, the name eventual consistency.

BASE: **B**asically **A**vailable, **S**oft state, **E**ventual consistency

* Basically, available means DB is available all the time as per CAP theorem
* Soft state means even without an input; the system state may change
* Eventual consistency means that the system will become consistent over time

Diagram

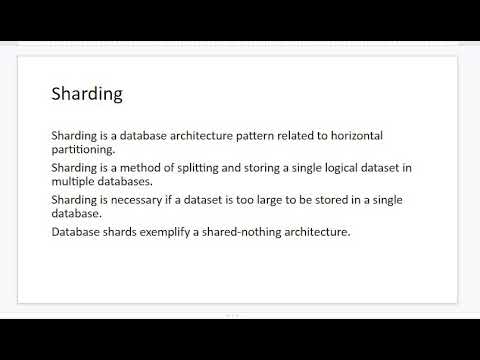
Description automatically generated

## Advantages of NoSQL

* Can be used as Primary or Analytic Data Source
* Big Data Capability
* No Single Point of Failure
* Easy Replication
* No Need for Separate Caching Layer
* It provides fast performance and horizontal scalability.
* Can handle structured, semi-structured, and unstructured data with equal effect
* Object-oriented programming which is easy to use and flexible
* NoSQL databases don’t need a dedicated high-performance server
* Support Key Developer Languages and Platforms
* Simple to implement than using RDBMS
* It can serve as the primary data source for online applications.
* Handles big data which manages data velocity, variety, volume, and complexity
* Excels at distributed database and multi-data center operations
* Eliminates the need for a specific caching layer to store data
* Offers a flexible schema design which can easily be altered without downtime or service disruption

## Disadvantages of NoSQL

* No standardization rules
* Limited query capabilities
* [RDBMS](https://www.guru99.com/difference-dbms-vs-rdbms.html) databases and tools are comparatively mature.
* It does not offer any traditional database capabilities, like consistency when multiple transactions are performed simultaneously.
* When the volume of data increases it is difficult to maintain unique values as keys become difficult
* Doesn’t work as well with relational data
* The learning curve is stiff for new developers
* Open source options so not so popular for enterprises.



## 

## Summary

* NoSQL is a non-relational DBMS, that does not require a fixed schema, avoids joins, and is easy to scale
* The concept of NoSQL databases became popular with Internet giants like Google, Facebook, Amazon, etc. who deal with huge volumes of data
* In the year 1998- Carlo Strozzi use the term NoSQL for his lightweight, open-source relational database
* NoSQL databases never follow the relational model it is either schema-free or has relaxed schemas
* Four types of NoSQL Database are 1). Key-value Pair Based 2). Column-oriented Graph 3). Graphs based 4). Document-oriented
* NOSQL can handle structured, semi-structured, and unstructured data with equal effect
* CAP theorem consists of three words Consistency, Availability, and Partition Tolerance
* BASE stands for **B**asically **A**vailable, **S**oft state, **E**ventual consistency
* The term “eventual consistency” means to have copies of data on multiple machines to get high availability and scalability
* [NOSQL](https://www.guru99.com/sql-vs-nosql.html) offer limited query capabilities