NoSQL

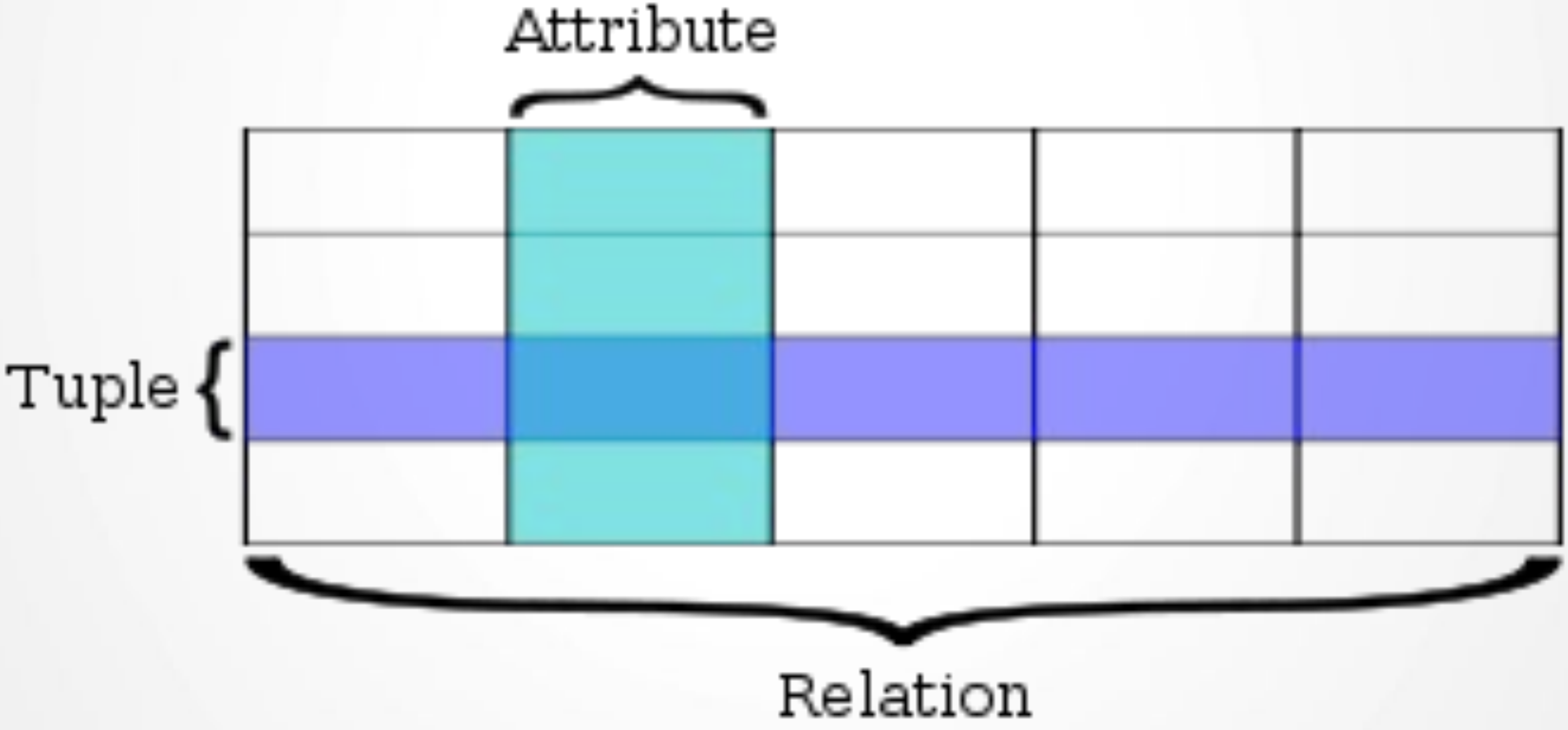
Contents

- Introduction
- SQL DB vs Non-SQL db
- What is NoSQL?
- Why NoSQL?
- CAP Theorem
- When to use?
- Motivations for this approach
- Types of NoSQL
- Conclusion

What is Database?

Relational Database

The relational database was invented in 1970 by E. F. Codd

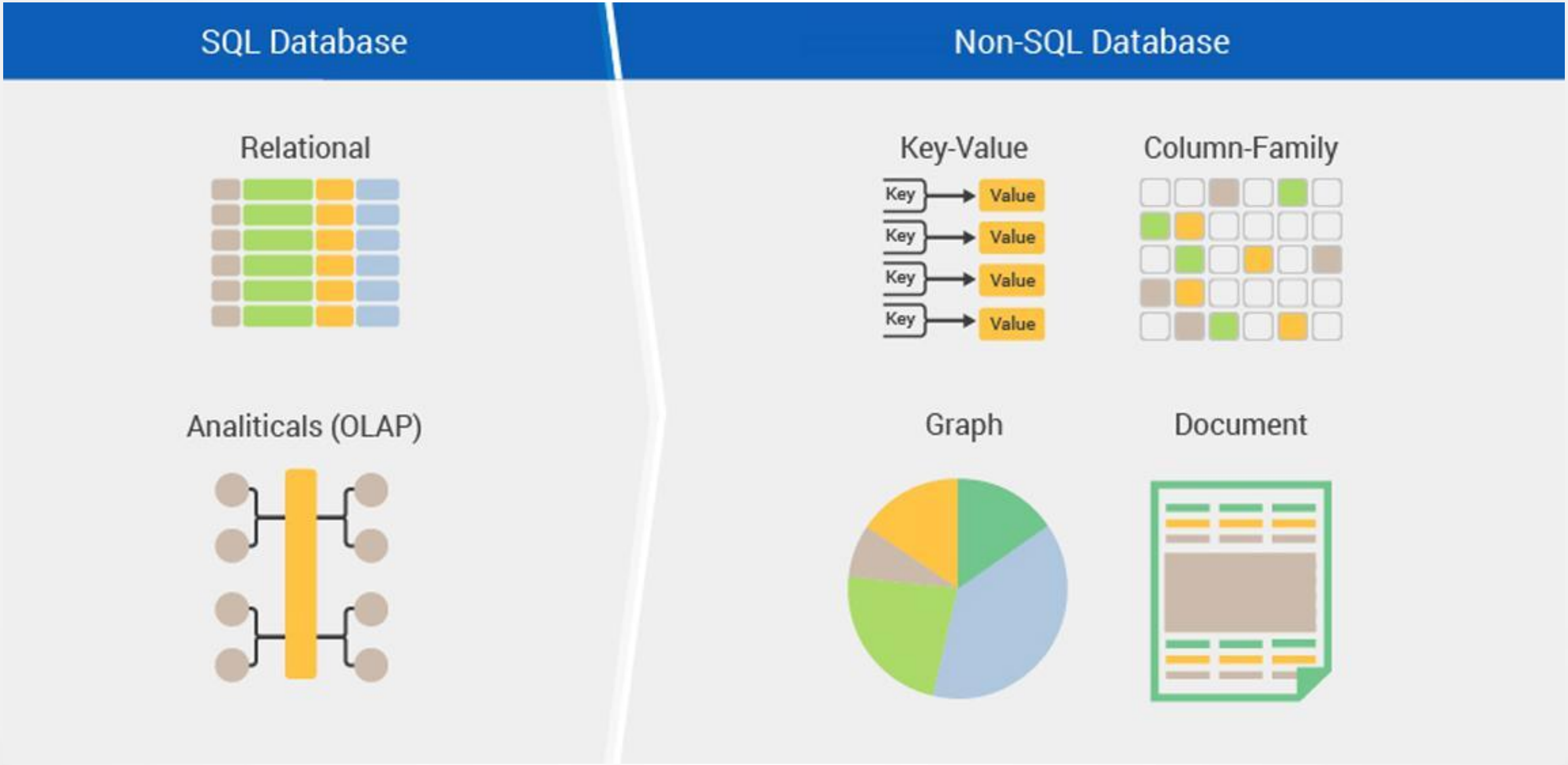


NoSQL

Non-SQL(non-realtional) databases are geared toward managing large set of varied and frequently updated data, often in distributed systems or in cloud.

They avoid the rigid schemas assosiated with RDBs.

SQL DB vs Non-SQL DB



Why NoSQL?

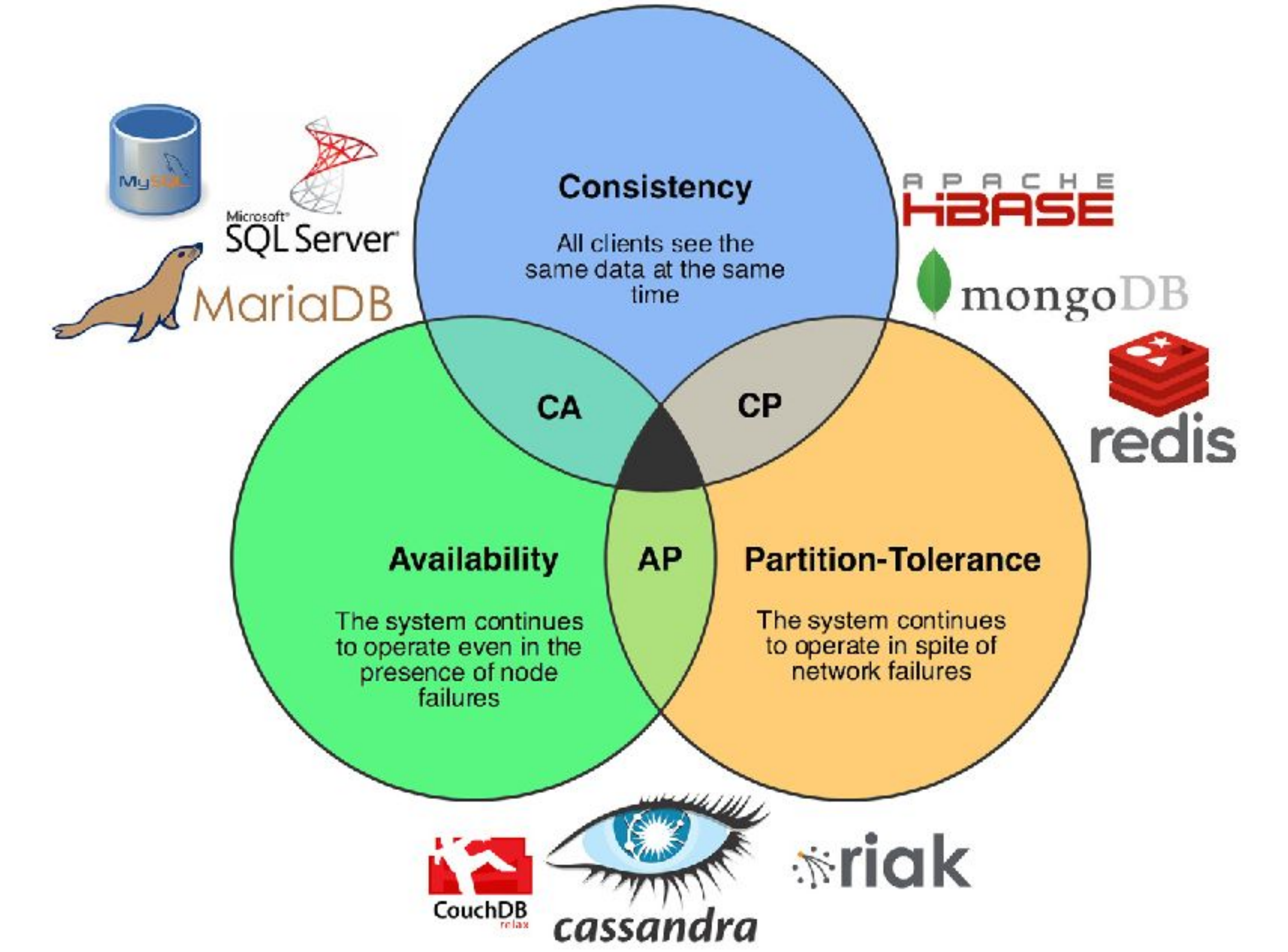
Relational databases were not designed to cope with the scale and agility challenges that face modern applications, nor were they built to take advantage of the storage and processing power available today.

It is flexible.

By flexibility we mean...

- Documents can be created without having to first define their structure
- Each document can have its own unique structure
- The syntax can vary from database to database, and
- Fields can be added on the go.

CAP Theorem



Consistency
All clients see the same data at the same time

Availability
The system continues to operate even in the presence of node failures

Partition-Tolerance
The system continues to operate in spite of network failures

CAP theorem with databases that “choose” CA, CP and AP

NoSQL

The NoSQL database movement has emerged particularly in response to three of these data challenges:

Data Volume

Data Velocity

Data Variety

When to use NoSQL?

NoSQL encompasses a wide variety of different database technologies that were developed in response to the demands presented in building modern applications.

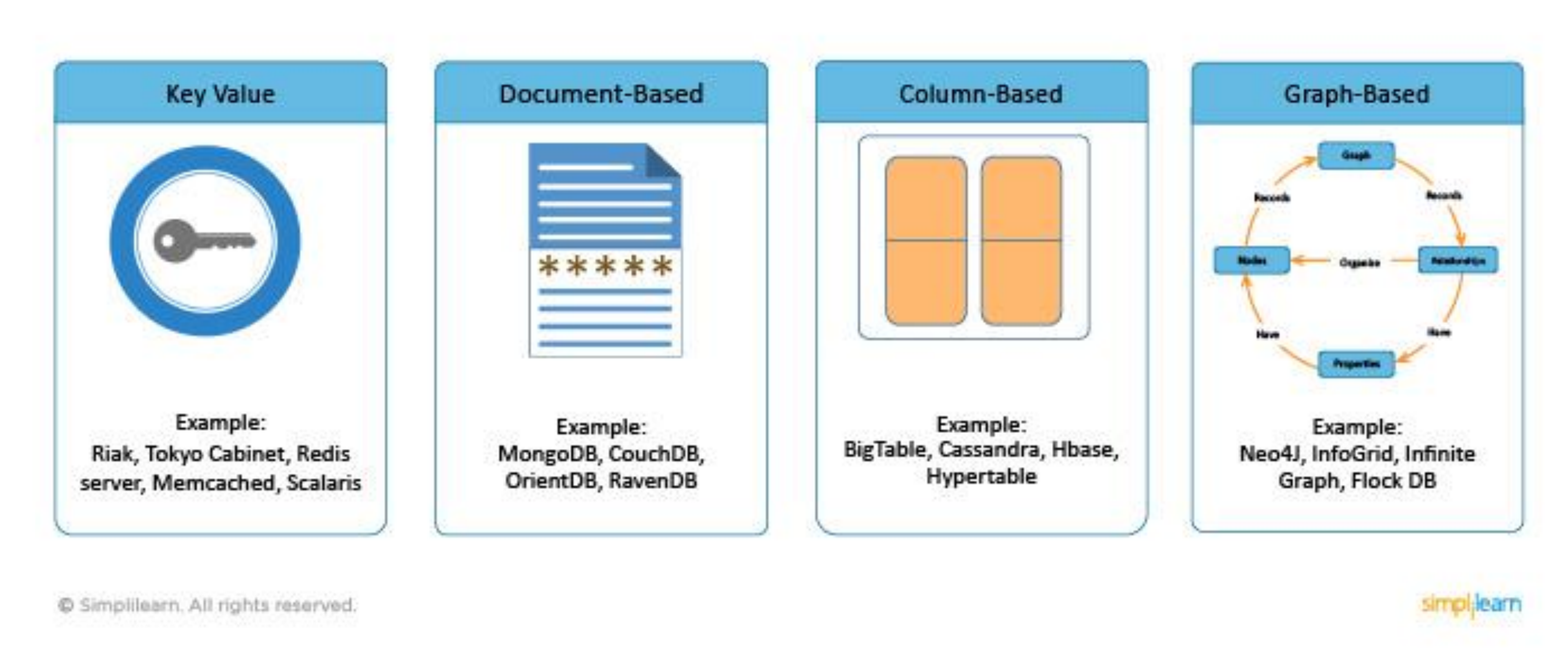
When to use??

- Developers are working with applications that create massive volumes of new, rapidly changing data types
- Small teams work in agile sprints, iterating quickly and pushing code frequently
- Applications that once served a finite audience are now delivered as services that must be always-on, accessible globally by millions.

Motivations for this approach include:

- Simplicity of design
- Simpler horizontal scaling to clusters of machines
- Finer control over availability
- The data structures used by are different, making some operations faster in NoSQL

Types of NoSQL Databases



Key Value

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

Document-Based

Example:
MongoDB, CouchDB, OrientDB, RavenDB

Column-Based

Example:
BigTable, Cassandra, Hbase, Hypertable

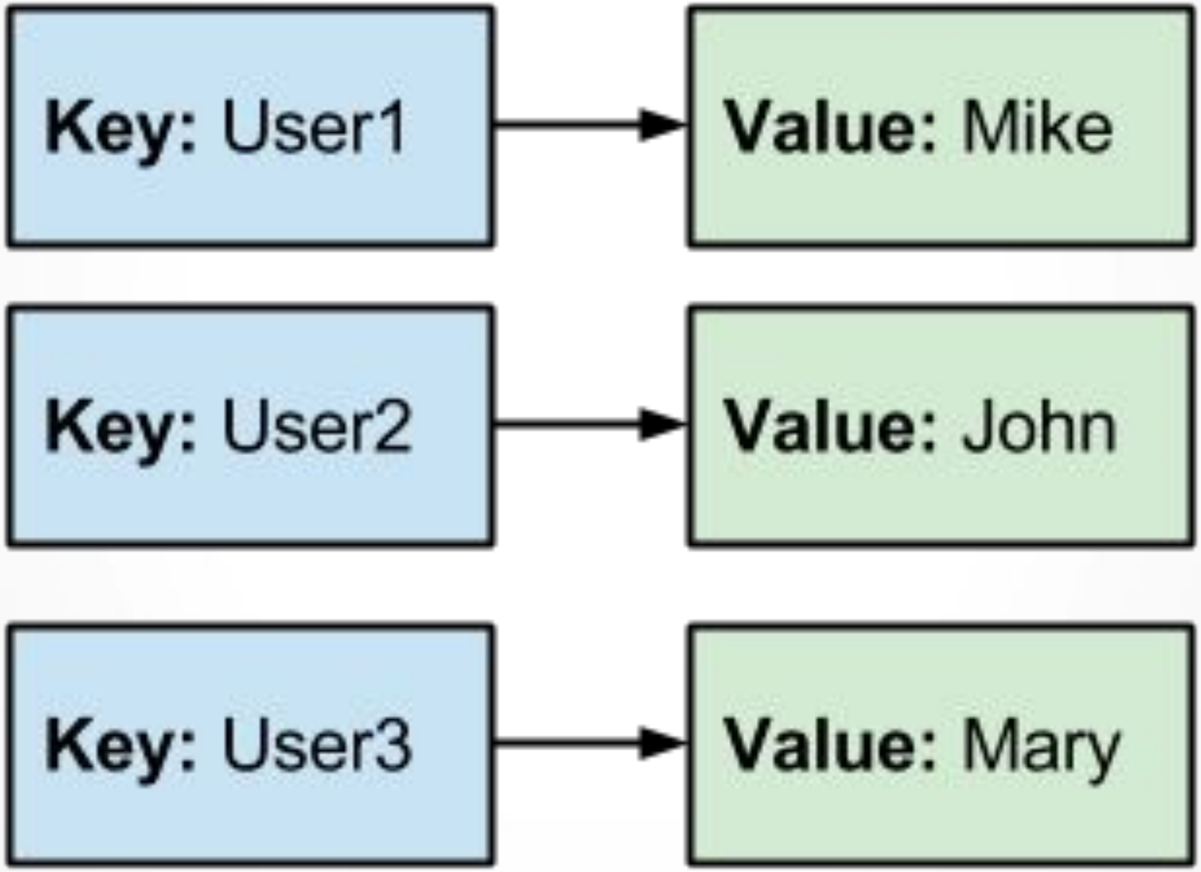
Graph-Based

Example:
Neo4J, InfoGrid, Infinite Graph, Flock DB

Key-Value Store

- Key-value store, data is represented as a collection of key–value pairs.
- They are the simplest NoSQL databases.
- Every single item in the database is stored as an attribute name (or 'key'), together with its value.
- The basic data structure is a dictionary or map.
- We can store value, such as an integer, string, a JSON structure, or an array, along with a key used to reference that value

Key-Value Store



Key: User1 → **Value:** Mike

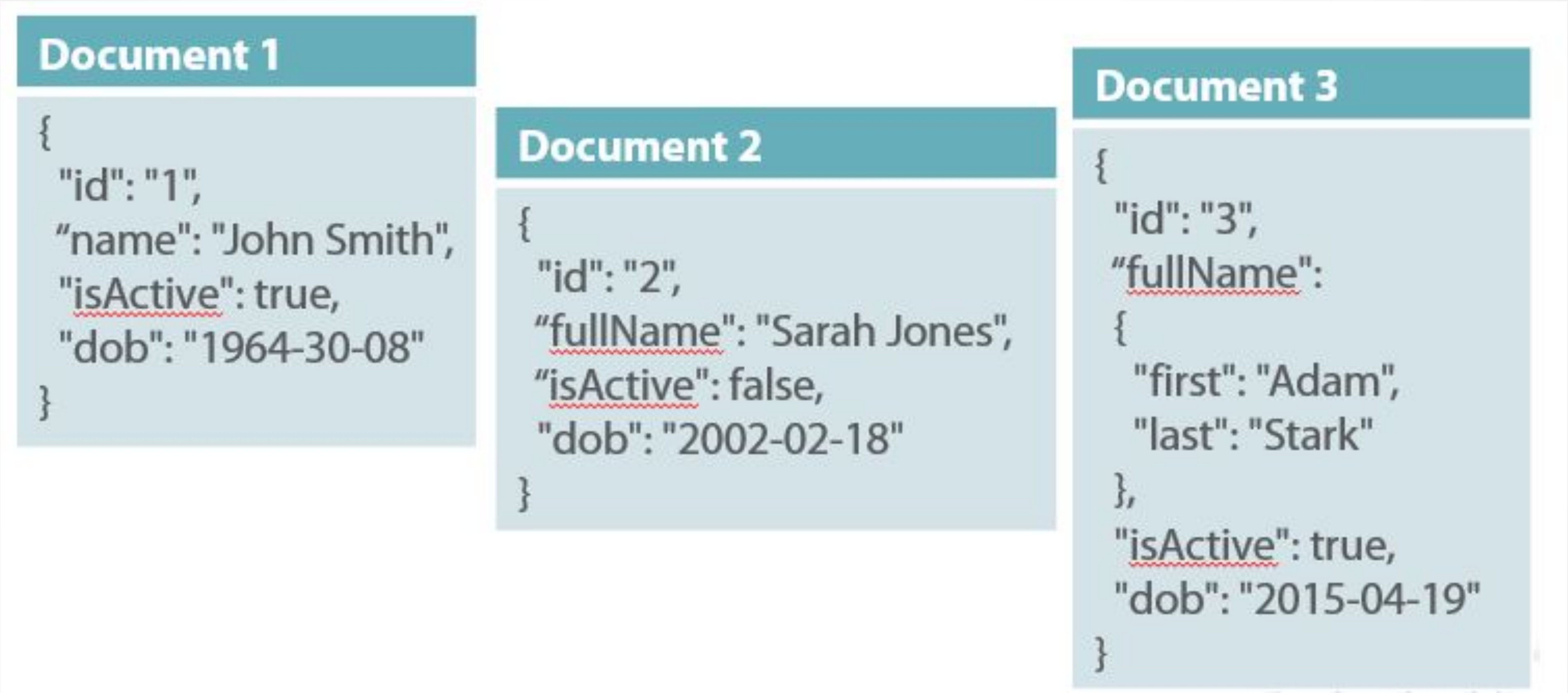
Key: User2 → **Value:** John

Key: User3 → **Value:** Mary

Document Store

- Designed to store semi-structured data as documents.
- Document-oriented databases are one of the main categories of NoSQL databases
- Document-oriented databases are inherently a subclass of the key-value store
- A document database typically stores data in JSON or XML formats

Document Store



Document 1

```
{
 "id": "1",
 “name": "John Smith",
 "isActive": true,
 "dob": "1964-30-08"
}
```

Document 2

```
{
 "id": "2",
 “fullName": "Sarah Jones",
 “isActive": false,
 "dob": "2002-02-18"
}
```

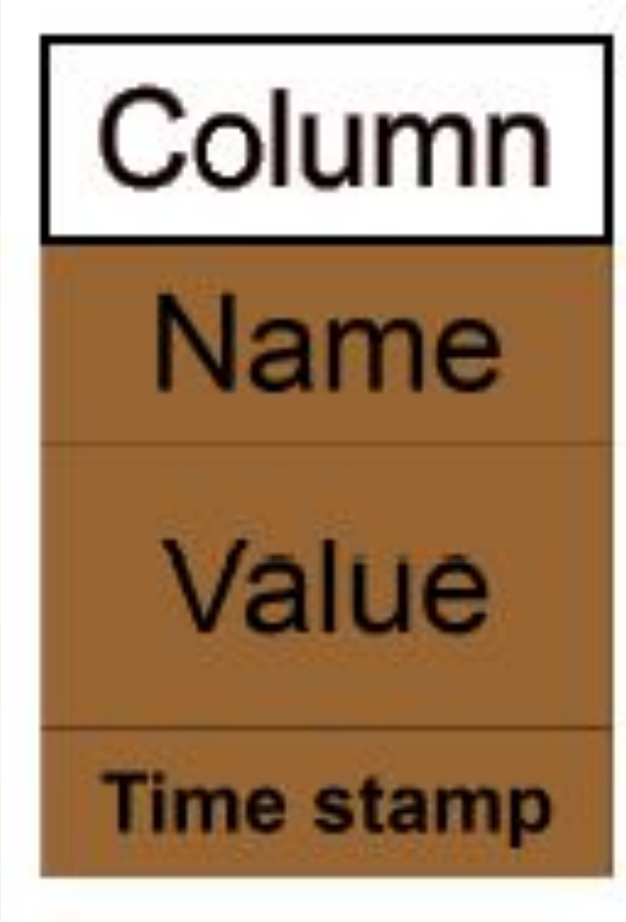
Document 3

```
{
 "id": "3",
 “fullName":
 {
  "first": "Adam",
  "last": "Stark"
 },
 "isActive": true,
 "dob": "2015-04-19"
}
```

Wide-Column Store

- A wide column store is a type of NoSQL database
- In column-oriented NoSQL database, data is stored in cells grouped in columns of data rather than as rows of data
- A column consists of a name, a value, and a timestamp
- A column is used as a store for the value and has a timestamp that is used to differentiate the valid content from stale ones
- In comparison, most relational DBMS store data in rows, the benefit of storing data in columns, is fast search/ access and data aggregation.

Wide-Column Store

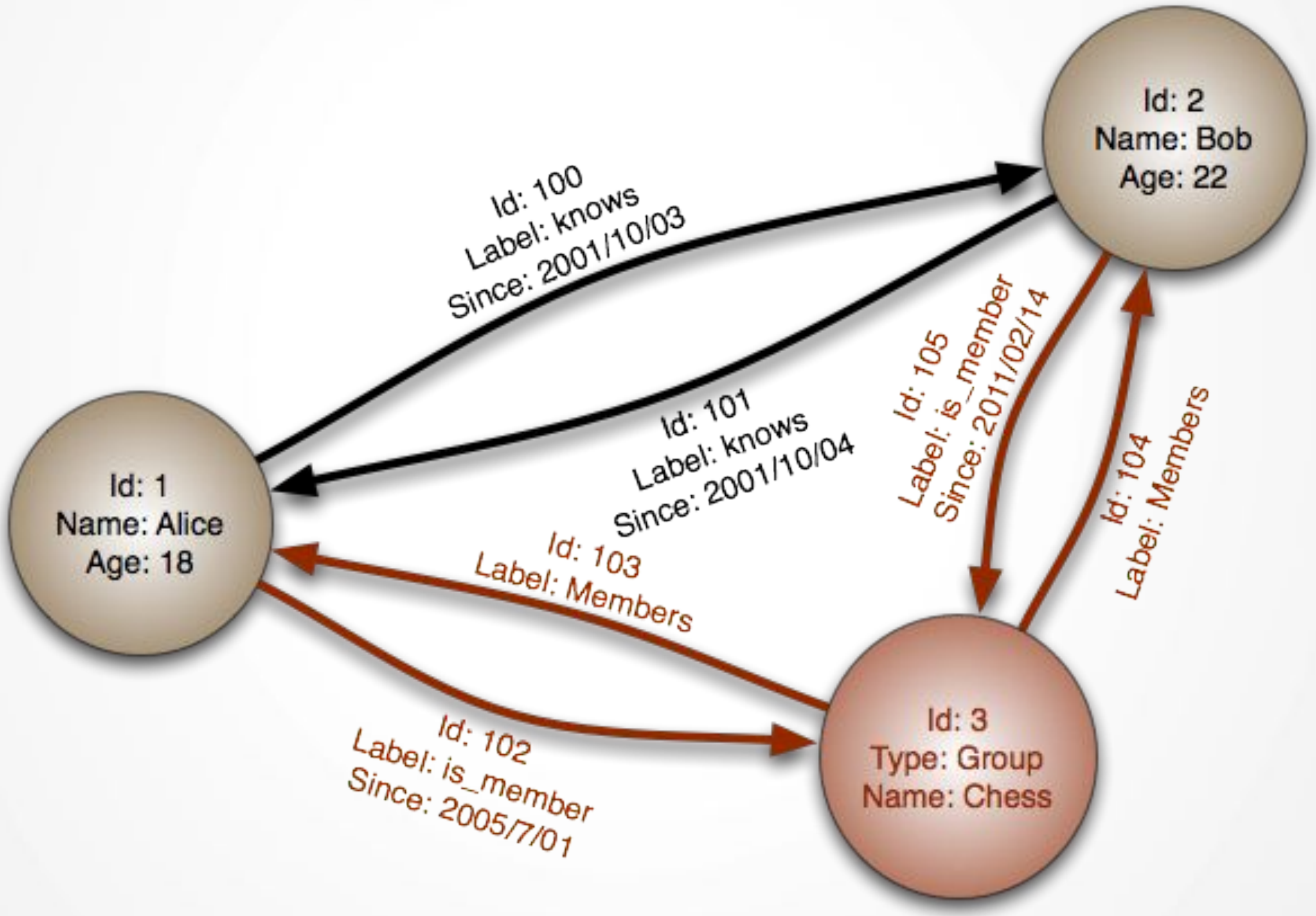


- **Unique name**: Used to reference the column
- **Value:** The content of the column. It can have different types, like AsciiType, LongType, TimeUUIDType, UTF8Type among others.
- **Timestamp:** The system timestamp used to determine the valid content.

Graph Database

- These databases that uses edges and nodes to represent and store data
- These nodes are organised by some relationships with one another, which is represented by edges between the nodes
- The relationships allow data in the store to be linked together directly, and in many cases retrieved with one operation
- Graph stores are used to store information about networks of data, such as social connections. Graph stores include Neo4J and Giraph.

Graph Database



RDBMS vs NoSQL

● Rigid	● Flexible
● Relational	● Object-Oriented
● Structured	● Semi-structured
● Stable	● Scalable
● Consistent	● Eventually Consistent

Conclusion

Relational databases can no longer handle today’s data volume, velocity and variety.

NoSql solves the problem of scalability and availability against that of atomicity or consistency.

Supriya.P
Programmer Trainee
InApp