

% Database

% Silvestro Di Pietro

% 8/05/2023

Definition

Database



Data in Data structure

The simpler way to imagine a database is the simpler form of data structure: the `array`

1. We must imagine a database as a structured data.
2. This data can be structured also if his data are not (unstructured).
3. The data can be represented as `data structure` (tree, graph, hash tables ...).
4. A book is a database, with an index, is a searchable book

Database paradigm

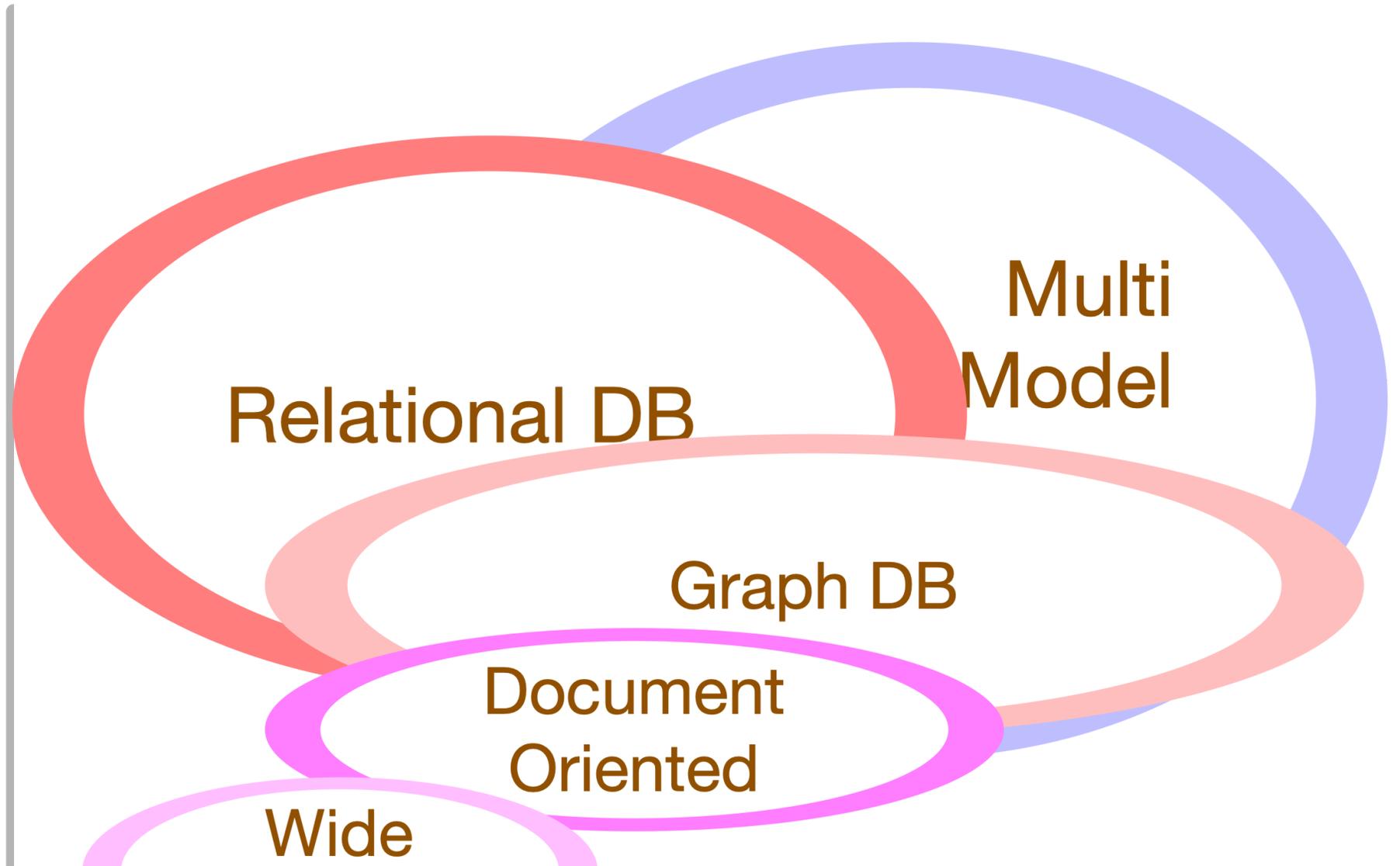
Databases type

There are seven different database paradigm, so 7 way to organize informations into a computer.

- key-value
- Wide Column
- Document oriented
- Relational
- Graph
- *Search*
- *Multi Model*

Different usage

**Units of
Information
/Relations**



Relational DB

Multi
Model

Graph DB

Document
Oriented

Wide

Database priorities

We can have different category of priorities:

- Data size (number of records)
- Write Many
- Read Many
- Number of computational element in data unit
- Speed
- Complex searches

The nature of data

The nature of your data will be the key for choosing the right database

Paradigma

key value

Structure

The key value database is mainly a python dictionary

Index

Keys

Values

000

leri

Yesterday

001

Oggi

Today

002

Usage

- In memory data (ram)
- Low number of computational element
- Speed
- low complex searches

Cache, Messaging,

Redis

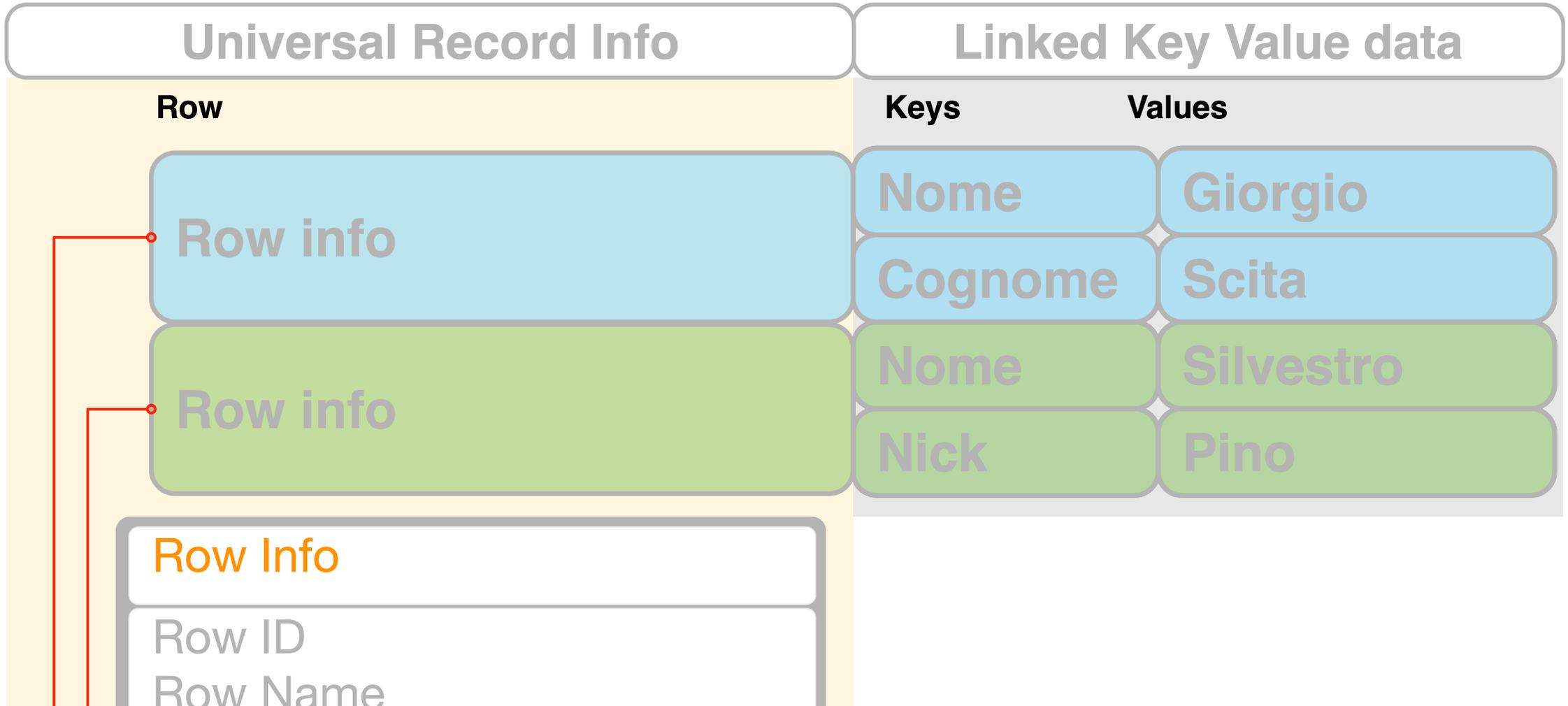
The open source, in-memory data store used by millions of developers as a database, cache, streaming engine, and message broker.

<https://redis.io/>



Structure

The wide Columns DB is an indexed collection of key-value records



Usage

- big Data size
- write many
- Low number of computational element
- low complex searches
- speed
- different column per record

handle large amounts of data and a high number of concurrent users. It is often used in applications that require high write throughput and low latency, such as online gaming, real-time analytics, and e-commerce.

RAP Tool

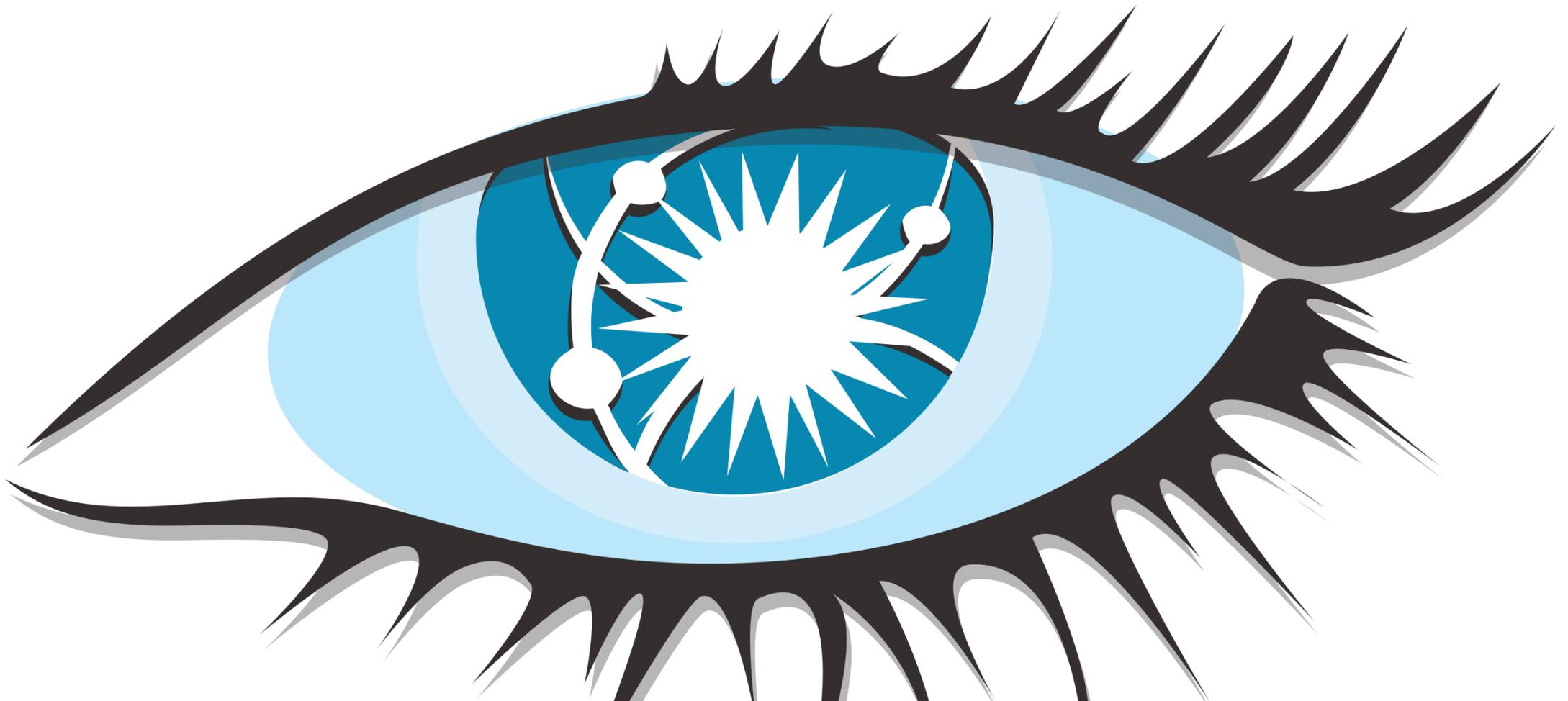
Rap Tool use the DDBlib that I wrote in 2006 in order to answer the logging procedure, the registration procedure.

The class library implement the: `GPD` approach

- GET
- PUT will create if item not found.
- DELETE

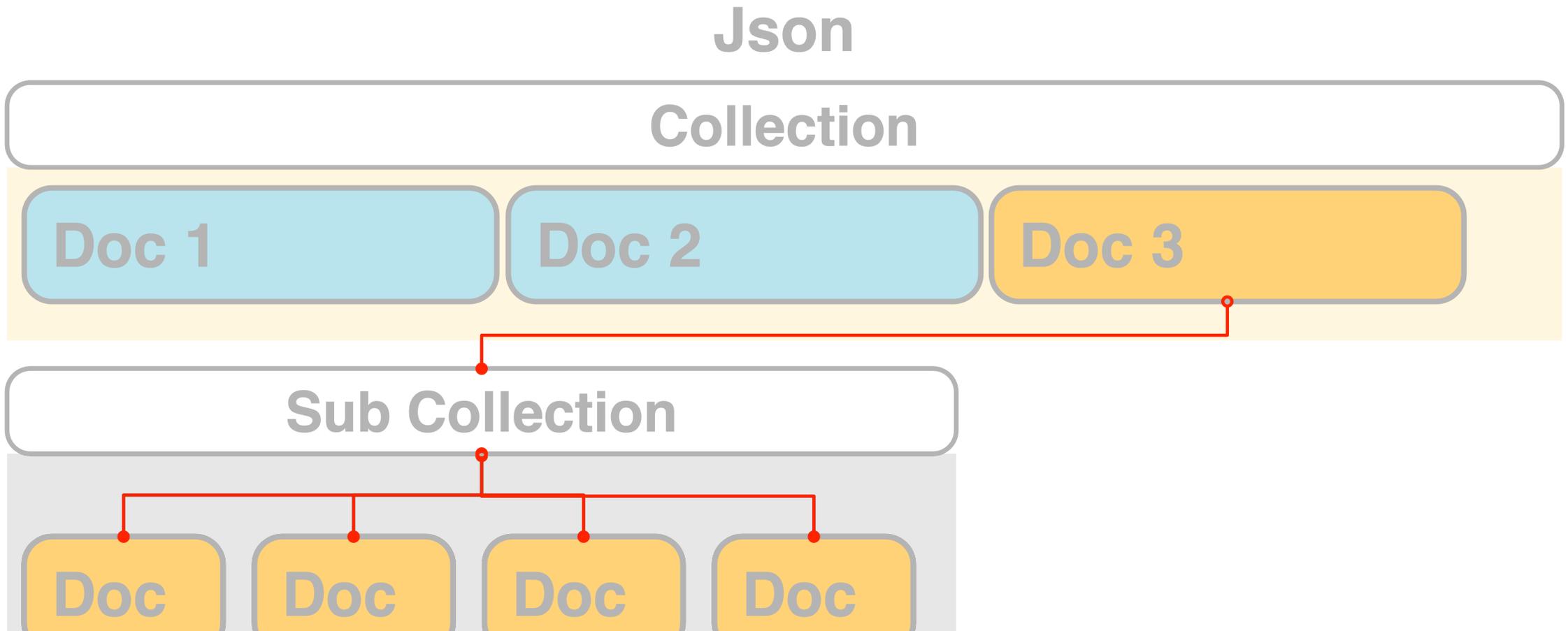
Cassandra

https://cassandra.apache.org/_/index.html



Structure

A key-value tree organised structure of json (or bson binary json) documents. Some still Use of XML for the document.



Usage

- big Data size
- write many
- Low number of computational element
- low complex searches
- speed
- different data per record

The particularity of those dbs is managing semi-structured data and developer friendly approach

CRUD

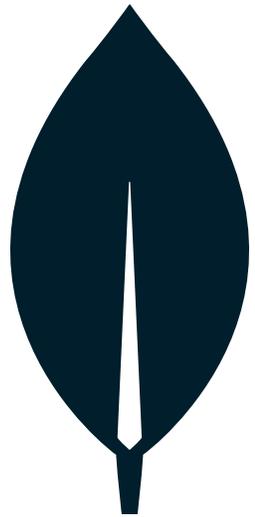
This is the one of feature of this 3 databases paradigm. The reason is their `noSql` approach, and the usage of `CRUD` managing of the record

- `C` reate
- `R` etrive
- `U` pdate
- `D` elete

Other main feature is the missing declaration of the database structure.

MongoDB

<https://www.mongodb.com/>

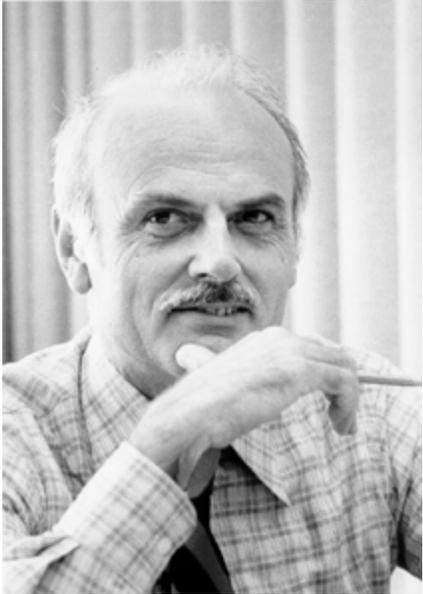


MongoDB®

{width=700}

Relational Model

Ted Codd



{width=150}

The relational model come from Edgar F "Ted". Codd

The purpose of the relational model is to provide a declarative method for specifying data and queries: users directly state what information the database contains and what information they want from it

Structure

Table User Struct

User ID
Field Name
Field Surname
Field Age

Table User

User ID	Name	Surname	Age
1	Silvestro	Di Pietro	62
2	Salvatore	Cerreto	33
3	Cosimo	Cutugno	29
4	Manuelo	Malizia	50
5	Igal	Janni	48

Membership Join (relation)

User ID
Group ID

Result Join Membership

Group ID	User Id
1	1
1	3
1	2
3	4

Table Groups Struct

Table Groups

Usage

- big Data size
- write many/read many
- large number of computational element
- complex searches

You need to structure the data, describe relation, assign type to fields, declare index and search keys: is powerful but not an easy task to project and maintain

SQL

Structured Query Language,

```
SELECT user.name as member, groups.name as group
FROM user
left JOIN groups
ON groups.group_id = user.user_id
```

Maria DB

<https://mariadb.com/>

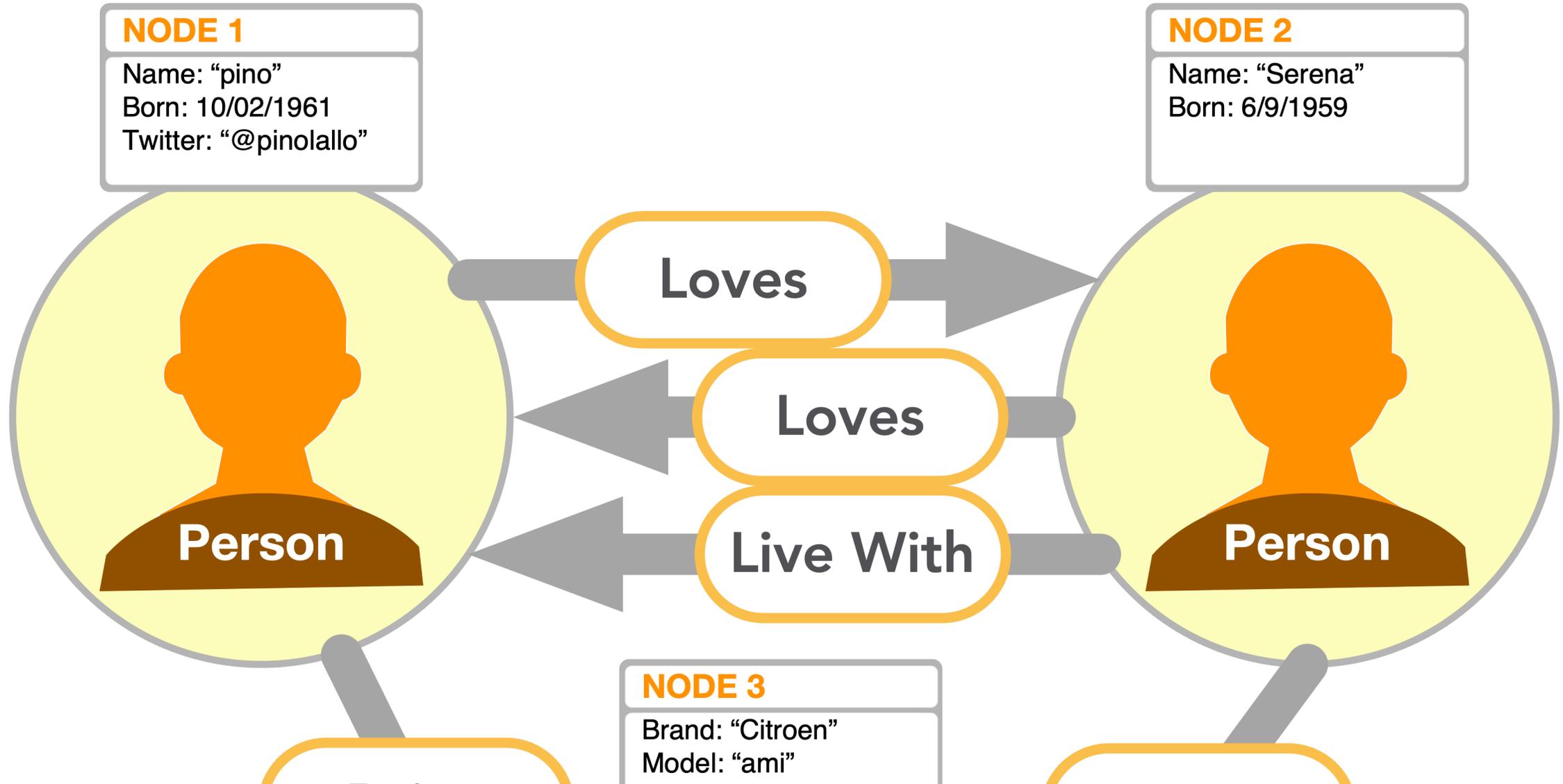


MariaDB

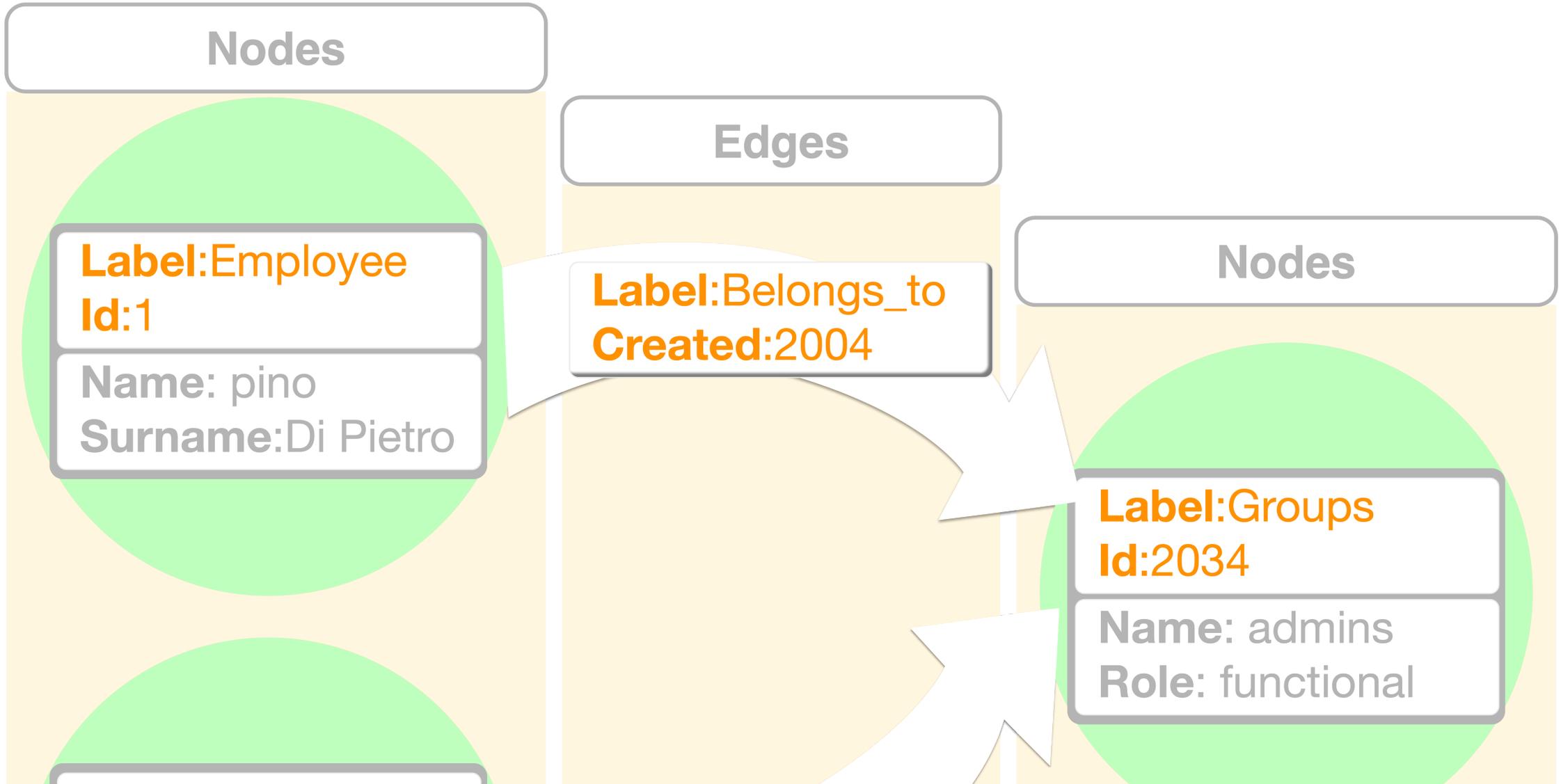
{width=750}

Graph

Structure a



Structure b



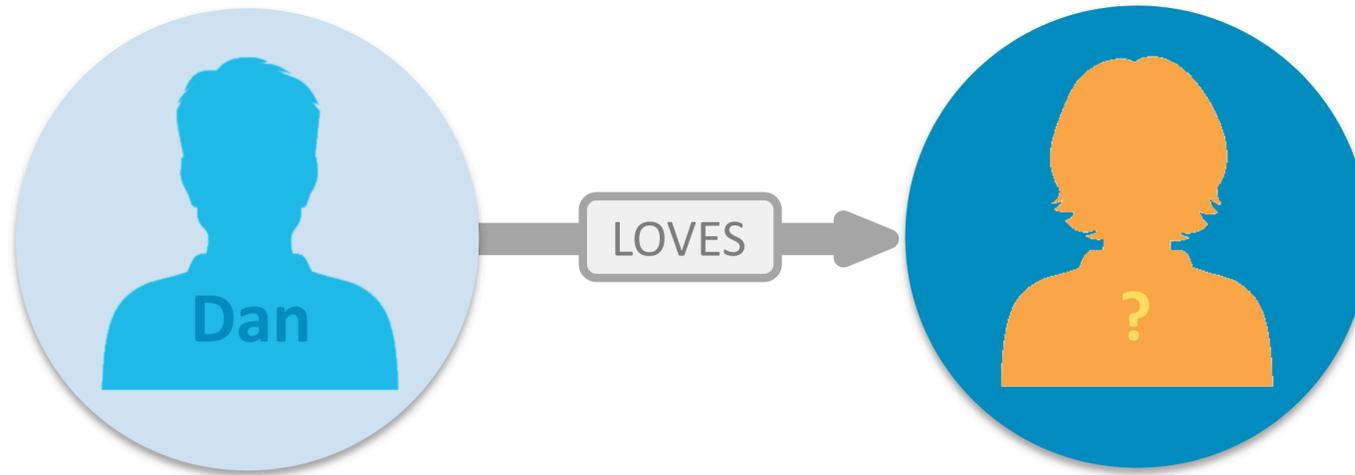
Usage

- big Data size
- write many/read many
- low number of computational element
- very complex searches
- speed

Is the database of choice when the relations among data are the most significant thing.

cypher

Cypher Query Language



```
MATCH (:Person { name:"Dan" } ) -[:LOVES]-> ( whom ) RETURN whom
```

LABEL

PROPERTY

VARIABLE

neo4j

<https://neo4j.com/>



{width=450}

Full-text search

Structure

Document a

Lorem ipsum dolor sit
amet incididunt

Document B

omnis iste natus error
sit voluptatem ipsum

Document C

sed do eiusmod
tempor incididunt ut

StopWord List

Lorem
ipsum
dolor
sit
amet
omnis
iste
natus
error
sit
voluptatem
sed
do
eiusmod
tempor
incididunt

Inverted Index

Id	Word	Document
1	Lorem	A
2	Ipsum	A,B
3	Dolor	A
4	Incididunt	A,C
5	Omnis	B
...

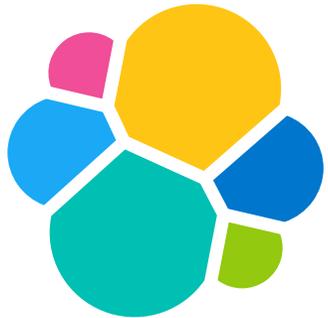
Usage

- big Data size
- read many
- very low number of computational element
- speed

Search engines, log browsers, data lake searches, filesystem browsers

elasticSearch

<https://https://www.elastic.co/>



elastic

{width=450}