El futuro digital es de todos

MinTIC

‹Mision TIC 2022›

Non Relational Databases

NoSQL - MongoDB

By Carlos Agudelo

Order.

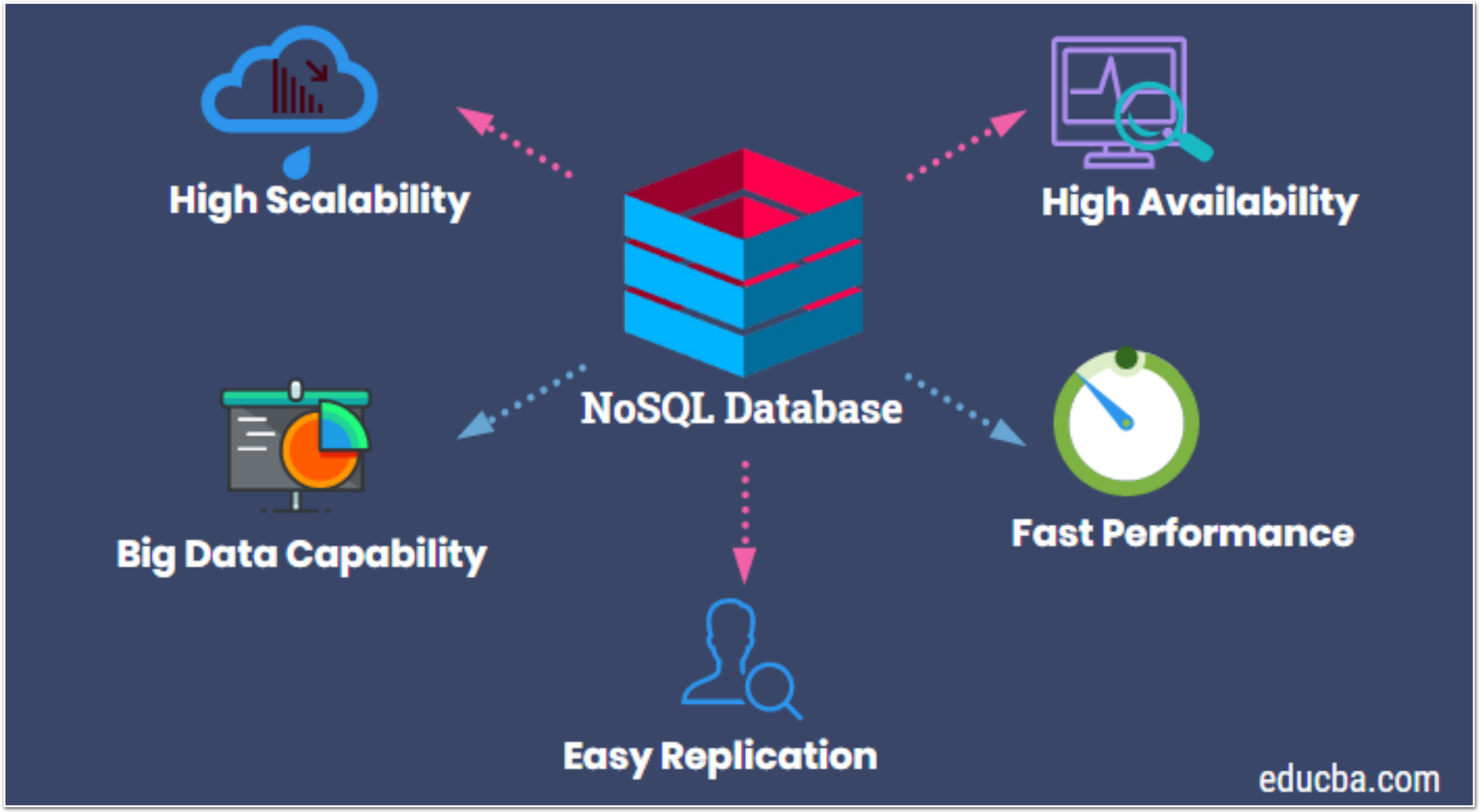
- What’s NoSQL?
- Introduction to NoSQL - Concept’s
- Kinds of noSQL Databases
- Differences between SQL and NoSQL - Which one should I’ll use?
- MongoDB introduction.
- MongoDB code examples.
- Real time code / NodeSchool

NoSQL

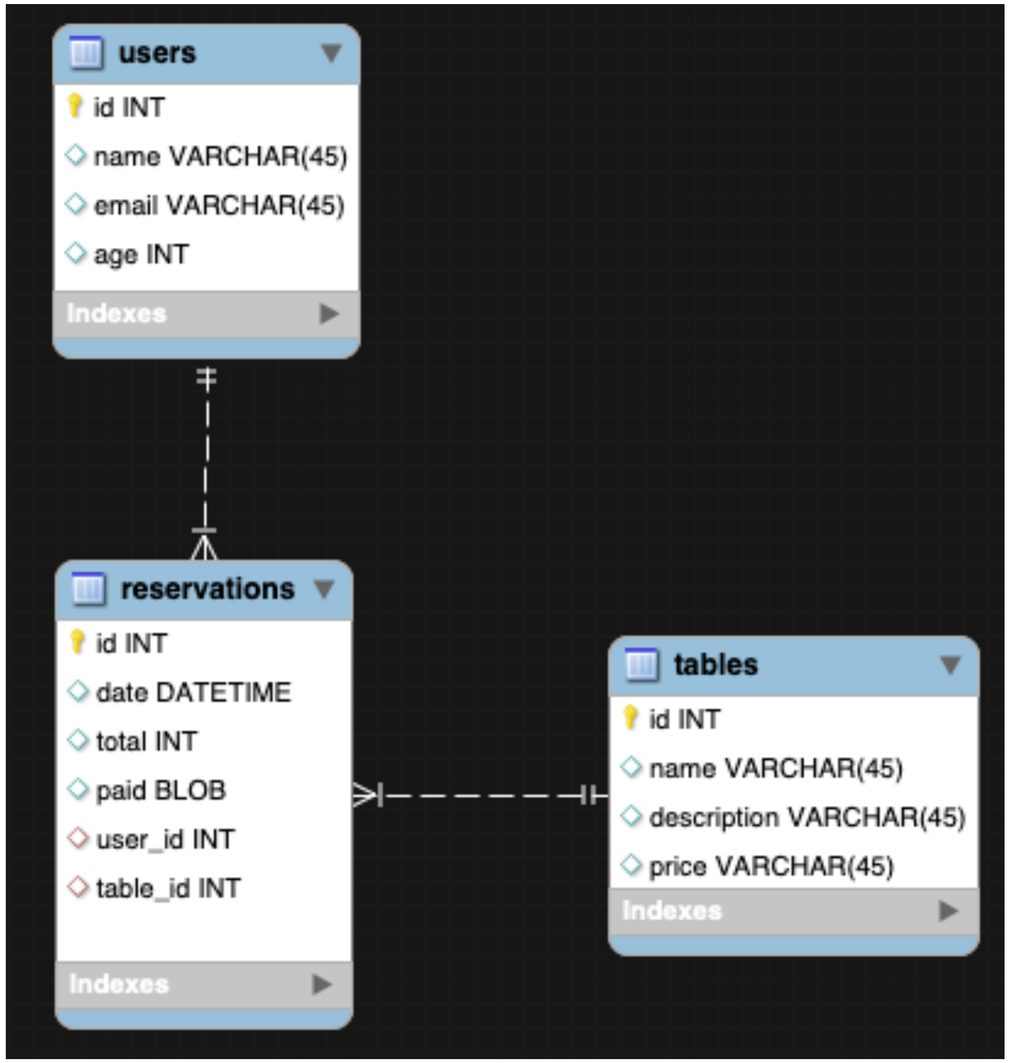
System recovery of information that don’t have a defined structure and use SQL to make queries on the data.

SQL - Structrure Query Language - https://www.w3schools.com/sql/

Other features.



Uses case.



Relational model


```
{
  "id": "bde4c99a-0c04-11ec-9a03-0242ac130003",
  "date": "03/09/2021",
  "total": 150.000,
  "paid": false,
  "user": {
    "id": "caa64168-0c04-11ec-9a03-0242ac130003",
    "name": "Juan Bedoya",
    "email": "prueba@example.com",
    "age": 26
  },
  "table": {
    "id": "cf5a956a-0c04-11ec-9a03-0242ac130003",
    "name": "Mesa # 1",
    "description": "mesa con vista al mar",
    "price": 150.000
  }
}
```


No Relational example

Advantage

- Flexibility
- Scalability
- Highly performance
- High functionallity
- Easy learn.

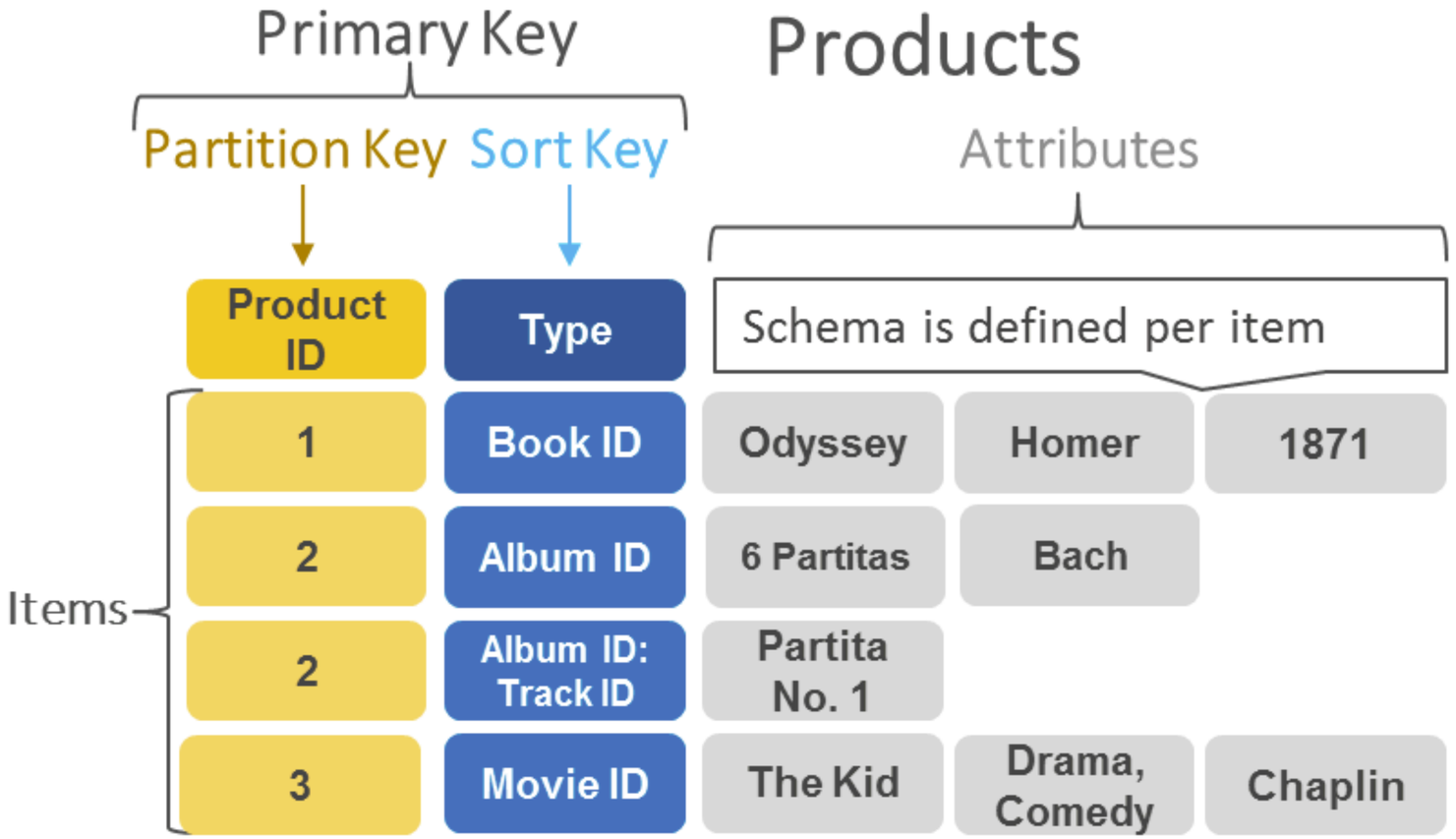
Disadvantage

- Don’t allow ACID Model. (Atomicidad, consistencia, aislamiento y durabilidad)
- Don’t have a standard language like a SQL.
- Some queries can will be so complicated and fastest on the server.
- It is rellatively new on industry.

Kinds of NoSQL Database

Key - value

Store data based on key-value pairs where each key is a uniq identification and apply a respective value.



Advantage.

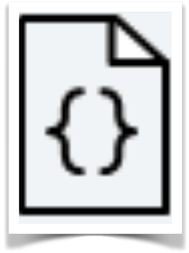
* Highly divisible
* Horizontal scalability
* Super high performace

Disadvantage.

* Scheme is too simple
* Trouble to real life cases model

Use Cases.

* Store sessions
* Shopping car



Documents

Save documents like a a collections (JSON) type.

```
[
    {
        "year" : 2013,
        "title" : "Turn It Down, Or Else!",
        "info" : {
            "directors" : [ "Alice Smith", "Bob Jones"],
            "release_date" : "2013-01-18T00:00:00Z",
            "rating" : 6.2,
            "genres" : ["Comedy", "Drama"],
            "image_url" : "http://ia.media-imdb.com/images/N/O9ERWAU7FS797AJ7LU8HN09AMUP908RLlo5JF90EWR7LJKQ7@@._V1_SX400_.jpg",
            "plot" : "A rock band plays their music at high volumes, annoying the neighbors.",
            "actors" : ["David Matthewman", "Jonathan G. Neff"]
        }
    },
    {
        "year": 2015,
        "title": "The Big New Movie",
        "info": {
            "plot": "Nothing happens at all.",
            "rating": 0
        }
    }
```

Advantage.

* flexible, semi-structured and hierarchical.
* Agile and functionality

Disadvantage.

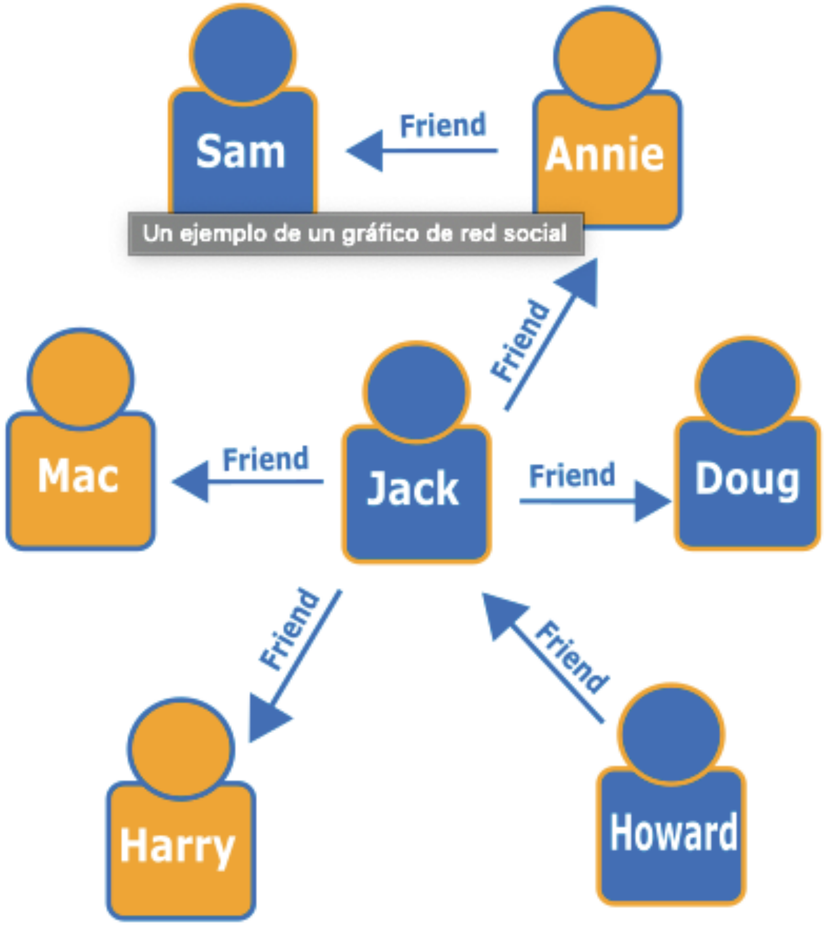
* Maybe can occupate a lot of memory capacity.
* Don’t guarantee ACID properties

Use Cases.

* Content administration
* Catalogs

Graph

It was design to record relations and surf through them.



Advantage.

* Explicit relation between data.
* Scheemaless

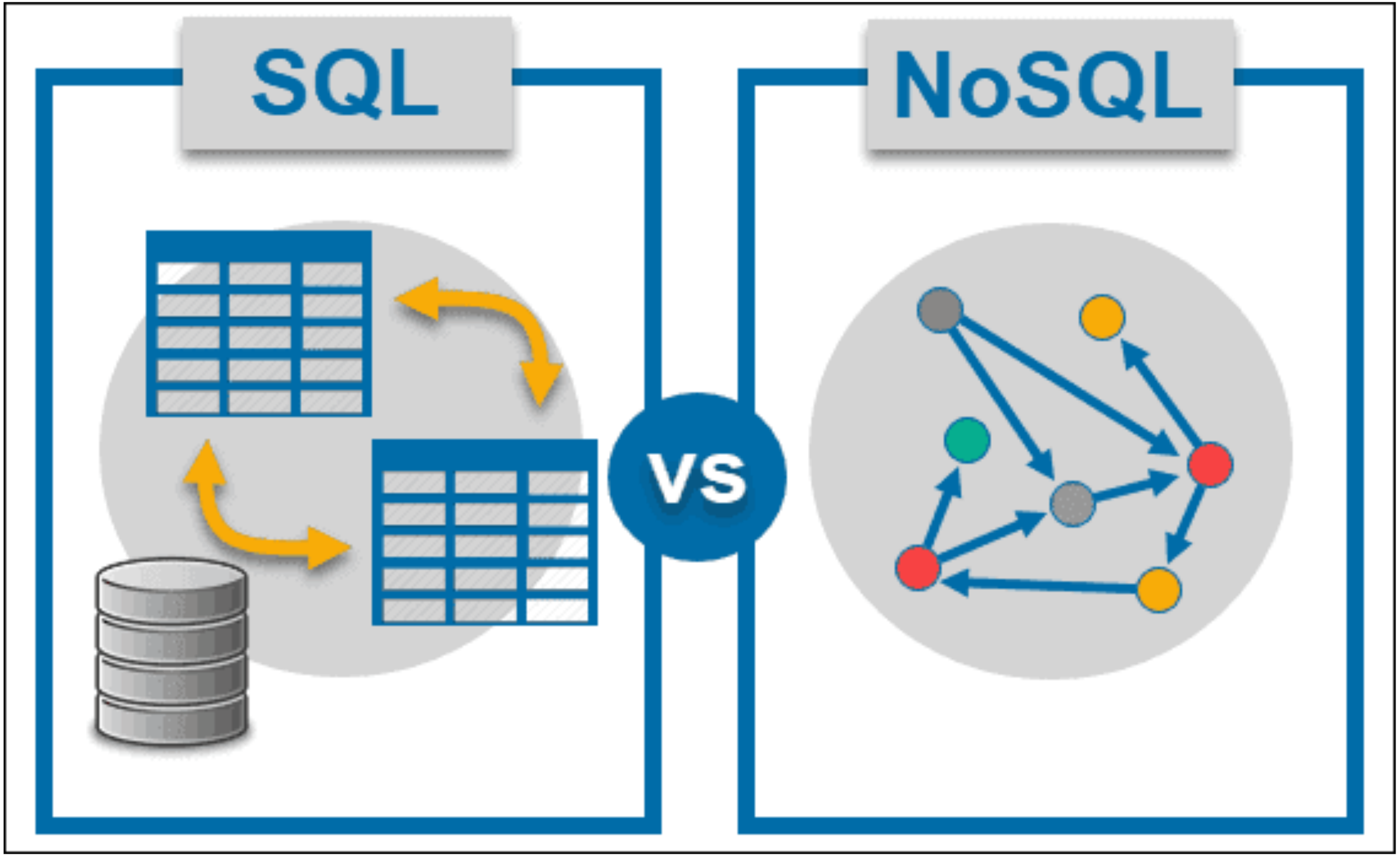
Disadvantage.

* No easy to scale.

Use Cases.

* Social network.
* Fraud detection
* Recommendation engines

“The tool that you use depend’s of the problem that you want solve.”


SQL
NoSQL
VS

- In relational databases the information is organized in a structured way in tables; in non-relational no.
- NoSQL does not use SQL as the host language for its queries.
- NoSQL is mostly used to store unstructured or semi-structured data.
- A relational database does not comply with ACID properties as effectively as a relational database
- The scalability is greater in a non-relational database, and they are also prepared to support a greater volume of data.
- NoSQL also offer greater flexibility and horizontal scalability.
- Unlike relational databases, non-relational databases do not yet have a standardized language (SQL).
- Community support is best for non-relational databases.

Conclusions.

- NoSQL is better if you app have a higher data volumes and it is not structure.
- Depend’s your implementation you have multiple options to choose the correct NoSQL engine.
- Relational database will be an a good option to develop software.
- Relational databases are the mostly use on development. But this can change on the future.
- If you system need scale on the future, NoSQL are the perfect option.

mongoDB

NoSQL database based on Documents

Features.

- **Easy to model data.**
  Have drivers for multiple languages and frameworks to work.

- **Flexible.**
  Easy to work and quick scalability

- **High availability**
  Allow to have multiple distributed clusters.

- **Powerful syntax.**
  Allow to make Wonderfull queries with code less

- **Open source.**
  Don’t have pay for use it and have a great community supporting it.

Concept’s

- **Collections.**
  It’s like a table an a relational databases. This group all documents for example collection user group all documents that contain users information.

- **Documents.**
  It’s a fundamental unity on this engine, in this all information will be recovery. Like a JSON or a BSON file.

- **Drivers.**
  They are all library or plugins that provide a easy communication with all languages and frameworks.

- **Methods.**
  They are all functions that provide make a query on database information.

Queries comparison.

- **Select all.**
  SQL - select * from ${table_name}
  NoSQL - db.${collection_name}.find()

- **Find by.**
  SQL - select * from ${table_name} where ${table_fild} = ${item_searh}
  NoSQL - db.${collection_name}.find(“${field}”: “${item_searh}”)

- **Add register.**
  SQL - insert into ${table_name} values { … }
  NoSQL - db.${collection_name}.insertOne({…})

- **Update.**
  SQL - update ${table_name} set { … } where {…}
  NoSQL - db.${collection_name}.update({…})

- **Delete.**
  SQL - delete ${table_name} where { … }
  NoSQL - db.${collection_name}.deleteOne({…})

Documentation.

- https://aws.amazon.com/nosql/?nc1=h_ls
- https://docs.mongodb.com/
- https://www.mongodb.com/nosql-explained
- https://www.mongodb.com/
- https://www.json.org/json-en.html

El futuro digital es de todos

MinTIC

‹Mision TIC 2022›