

OpenCourseWare

# Database

## 3.1 NoSQL databases Introduction

---

**Lourdes Moreno López**

**Paloma Martínez Fernández**

**José Luis Martínez Fernández**

**Rodrigo Alarcón García**



# CONTENTS

---

- Introduction
- NoSQL DB features

# Learning objectives

---

- The student must be able to:
  - Identify the situations where using NoSQL DB is a solution.
  - Know the features of NoSQL DB

# INTRODUCTION

## web 2.0

---

- Web 2.0 is the second generation of the world wide web, where it moved static HTML pages **to a more interactive and dynamic web experience.**
- Web 2.0 is focused on the ability for **people to collaborate and share information** online via social media, blogging, and Web-based communities.



<https://www.pexels.com/es-es/foto/mano-iphone-telefono-inteligente-internet-5082581/>

# INTRODUCTION

## web 2.0 in 60 seconds



# INTRODUCTION

## Change of situation

---

- Web 2.0 provokes a change of situation because new forms of business based on the large volume of data arise. **The data becomes the most important.**
- What are the new requirements for this new situation?
  - A large and indeterminate number of users distributed worldwide who can upload data, they are making hundreds of thousands of daily requests, requests must be answered, => therefore, the systems that support these applications **must always be online and available.**



<https://www.pexels.com/es-es/foto/ordenador-portatil-en-blanco-y-negro-sobre-un-escritorio-de-madera-marron-356056/>

# INTRODUCTION

## Change of situation

---

- The intelligent use of information and its exponential growth leads us to a new trend that the ICT industry called **big data**.
- Big Data (7 V's), data is characterized by a 7-parameters:
  - **Volume:** a large amount of data
  - **Velocity:** Need to be quickly analyzed data
  - **Variety:** Structured and unstructured data
  - **Variability:** Data whose meaning is constantly changing
  - **Veracity:** truthfulness
  - **Visualization:** Present the data in a compressible way
  - **Value:** Extracting information for decision-making

# INTRODUCTION

## Change of situation

---

- New computing models are needed to handle this large volume of distributed data and unstructured data.
- Companies face issues such as:
  - There is no available **storage** for more data
  - There are no **tools** to process such a volume of data
- Relational databases do not support this new context  
=> In order to respond the new requirements arise **NoSQL databases**



# INTRODUCTION

## Background

---

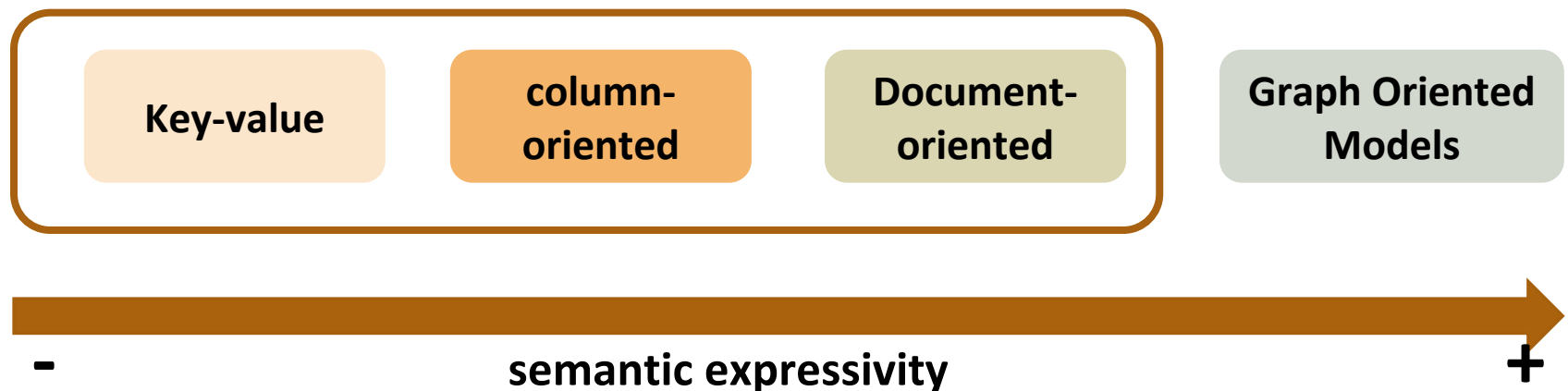
- In 1998 the term NoSQL appears, which means not only SQL databases.
- The term as is understood today began to become popular in 2009
- The first implementation of NoSQL DB:
  - BigTable, Google (2003)
- Implementations well recognized:
  - DynamoDB (Amazon 2007)
  - Cassandra (Facebook, 2008)



# NoSQL DB TAXONOMY

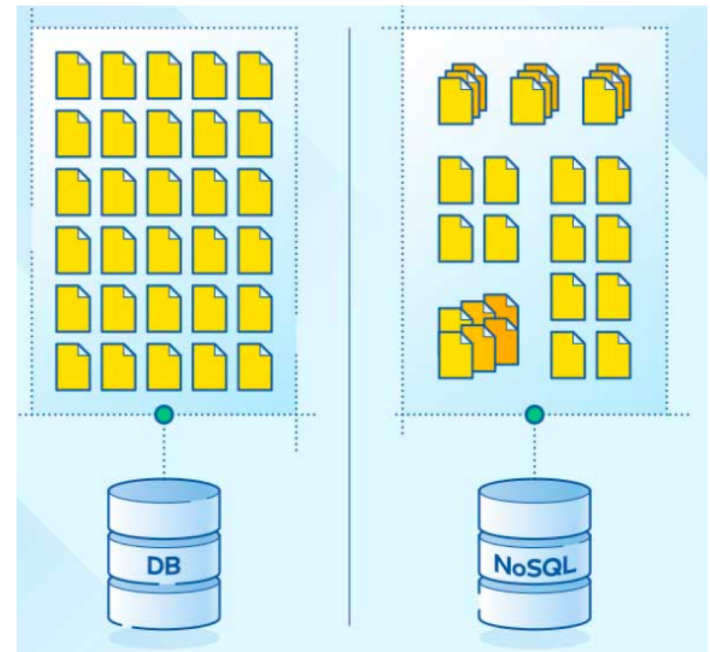
- **Aggregation data models** are used to treat structured and unstructured data
- **Graph data models** are useful in real-world domains with multiple and complex interrelationships (social network)

## Aggregation Oriented Models



# NoSQL DB FEATURES

- Structured, semi-structured and unstructured data. Heterogeneous data
- No single data model
- Flexible data scheme
- No standard data access language
- Horizontal scalability
- Distributed DB
- ACID properties are not guaranteed



# NoSQL DB FEATURES

## Flexible data scheme (schemaless)

---

- You don't have to define an explicit scheme
- The scheme is implicitly defined based on the data inserted
- In some cases, the scheme is non-existent from the point of view of the DBMS
- The scheme is implicit in the programs that access the database.
- Advantage:
  - Changes in the DB scheme are facilitated
  - Working with heterogeneous data is allowed.
  - Query operations are optimized avoiding join operations.

# NoSQL DB FEATURES

## No standard data access language

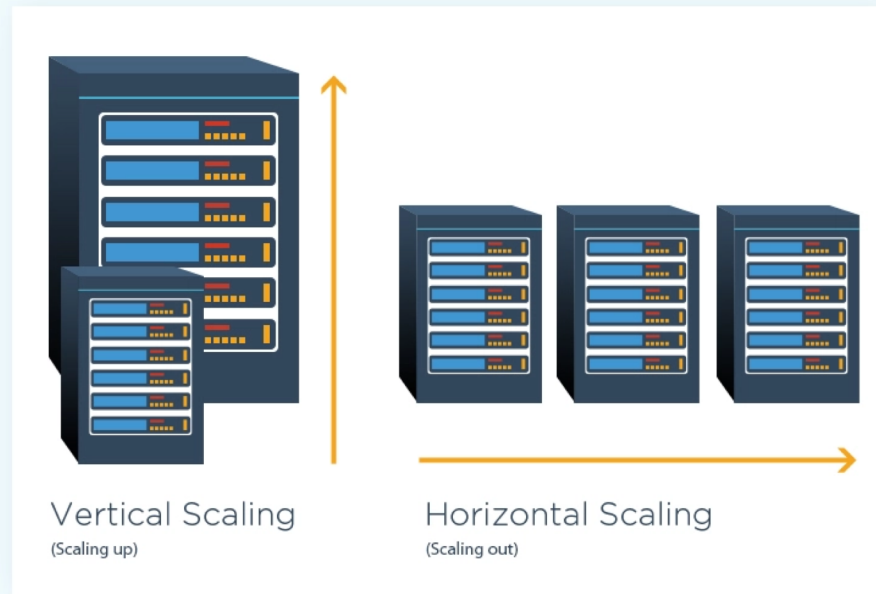
---

- No SQL language
- There is no standard access language
- There is your own data manipulation language or access via API
- BD access drivers are provided for different programming languages
- Mechanisms are provided that allow integration with distributed computing systems (MapReduce Framework)

# NoSQL DB FEATURES

## Horizontal scalability

- NoSQL DBs are based on “horizontal scalability” as opposed to relational DBs that are based on “vertical scalability”



# NoSQL DB FEATURES

## Horizontal scalability

---

- The NoSQL DBs based on aggregation models, the processing and storage of the data is done in a distributed way through a network of computers in general of low cost.
- This horizontal scalability makes it possible to grow according to the needs of the distributed system, in such a way that it is possible to add computers, new nodes to our computer network without therefore having to stop the system, the system is still available and online.

# NoSQL DB FEATURES

## Distributed DB

---

- Horizontal fragmentation (sharding)
- Massive data replication
  - Dispersion techniques, hash (Key-value DB)
  - Depending on the value of attributes (Document-oriented DB, column-oriented DB)
  - Vertical fragmentation: Depending on different columns (column-oriented DB)
- Advantages:
  - Increase level of parallelism
  - Improve query efficiency



# NoSQL DB FEATURES

## Transaction and Data Consistency Models

---

- Transaction – Relational Model
  - Transaction : a sequence of operations to be executed atomically.
  - Oracle is a sequence of SQL statements that are treated as a single unit.
  - Model ACID
    - Atomicity
    - Consistency
    - Isolation
    - Definitive

# NoSQL DB FEATURES

## Transaction and Data Consistency Models

---

### Relational model: **ACID**

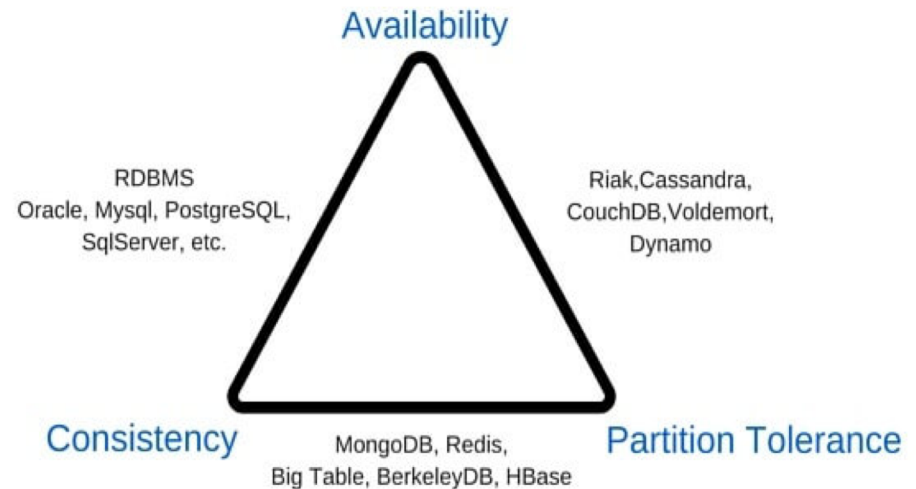
- **Atomicity:** the transaction is executed completely or nothing
- **Consistency:** the transaction must preserve the integrity of the DB (integrity constrains).
- **Isolation:** A transaction cannot see its execution interfered because other transactions that are accessing concurrently the same part of the DB
- **Definitive:** the changes produced by the execution of a transaction must be definitive for the DB, they cannot be lost
- With ACID model is very complex for non-SQL databases => therefore No-SQL databases follow an alternative transaction model called BASE

# NoSQL DB FEATURES

## Transaction and Data Consistency Models

---

- **Consistency (C):** System users must be able to retrieve the same values for the same data at the same time.
- **Availability (A):** The user requests to an available node must be answered.
- **Partition Tolerance (P):** The system must provide service to users despite unfavorable system failure situations that cause the system to be partitioned into different components.

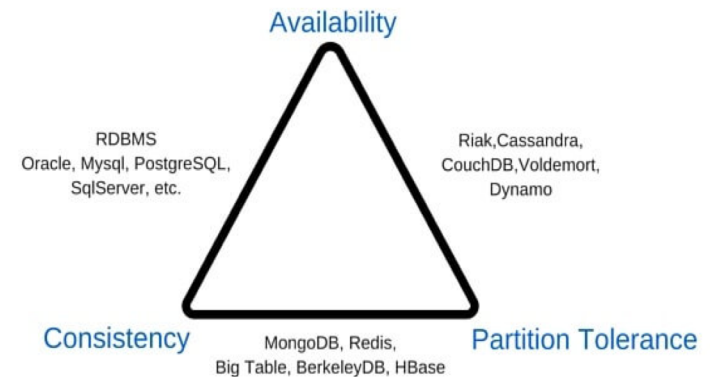


# NoSQL DB FEATURES

## Transaction and Data Consistency Models

---

- **CAP theorem** (Brewer, 2000): aspects that a distributed system must guarantee (only two !!)
  - **AP:** System that is always available, although it may temporarily show inconsistent data (in the presence of partitions): RIAK; DYNAMODB; CASSANDRA
  - **CP:** The system always has a consistent view of the data, although it is not always available (in the presence of partitions): MONGODB, HBASE; REDIS



# NoSQL DB FEATURES

## Transaction and Data Consistency Models

---

- NoSQL DB Model: BASE
  - Limited availability (Basic Availability)
  - Flexible state (Soft-state)
  - Final consistency over time (Eventual consistency)
    - The important aspect is the **change of meaning or focus of the consistency**, it is applied to replicas of the same data, it means that the content of the replicas in an instant of time may be different, but it is expected that over time, the replicas they converge on the same data in such a way that it will be consistent in the end

# NoSQL DB FEATURES

## NoSQL Disadvantages

---

- **Low portability:** There is a lack of standards because there is no "unified" query language as happened with SQL in relational databases => this lack of standards complicates the portability of applications, that is, it does not make it easier to change the manufacturer.
- **Complex data scheme.:** The flexible data scheme allows dealing with a variety of data types, mixing or working with heterogeneous data. => This can complicate the DB scheme, managing that scheme can be complex.

# NoSQL DB FEATURES

## NoSQL Disadvantages

---

- **Complex administration DB tasks:** There are no administration functions in the DBMS
- **Lack of maturity:** They are relatively new technologies, and there is a lack of maturity of user support by manufacturers, at least if we compare it with the manufacturers of relational databases
- **Learning curve:** There is a lack of specialists in these technologies, and they are not trivial tasks: the installation and maintenance of the DB can be complex tasks with a considerable learning curve

# NoSQL DB FEATURES

## RELATIONAL DBs VS. NoSQL DBs

---

| <b>RELATIONAL DBs</b>                      | <b>NoSQL DBs</b>                                  |
|--|---|
| Single data model                          | Different types of data models                    |
| Fixed scheme                               | Flexible scheme                                   |
| Structured data                            | Structured, semi-structured and unstructured data |
| undistributed, distributed (simple models) | Highly distributed                                |
| ACID                                       | BASE  |



# NoSQL DB FEATURES

## Conclusions

---

- A NoSQL DB is required if it meets at least one:
  - Application environments that require more flexible data schemas in relation to the schemas offered by relational DB
  - Highly distributed systems that need to manage large volume of data and this data should always be available.
- Questions to ask:
  - Data volume
  - Estimated Concurrency
  - No. of operations / unit time
  - Most frequent types of operations
  - Desired scalability
  - The desired degree of Integrity and consistency
  - Nature of the data

# REFERENCES

---

- Eric Redmond. Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement. 2012
- Harrison et al. Next Generation Databases: NoSQL, NewSQL, and Big Data. 2015. Apress.