

# Big Data y Visualización

## Modelos de datos

## Apache Spark

Universidad Autónoma de Manizales

**Docente:** Alejandro Mora Rubio



# Contenido

¿Qué es un modelo de datos?

Modelo de datos relacionales

Modelo de datos semiestructurado

Otros modelos

Data streams

Stream vs Batch processing

Apache Spark

# ¿Qué es un modelo de datos?

Un modelo describe características principales de algún objeto o fenómeno, en el caso de los datos, se describen 3 aspectos:

- Estructura
- Operaciones: selección, proyección, unión entre diferentes conjuntos (*union*) e intersección o relaciones con otros conjuntos de datos (*join*).
- Restricciones: de tipo, de valores o de estructura.

# Modelo de datos relacional

Datos estructurados, su representación más común son las tablas.

- Se definen alrededor de un esquema relacional que indica el tipo de dato y las restricciones para cada columna.
- Se pueden crear tablas “hijas” que se asocian a una tabla principal a través de identificadores únicos.
- Usualmente no admiten duplicados.

CTYNAME	CENSUS2010POP	ESTIMATESBASE2010	POPESTIMATE2010
Alabama	4779736	4780127	4785161
Autauga County	54571	54571	54660
Baldwin County	182265	182265	183193
Barbour County	27457	27457	27341
Bibb County	22915	22919	22861

# Modelo de datos semiestructurado

Estos datos comparten elementos estructurales fundamentales, pero sus valores y características cambian en cada caso.

## HTML

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Example</title>
5     <link rel="stylesheet" href="st
6   </head>
7   <body>
8     <h1>
9       <a href="/">Header</a>
10    </h1>
11    <nav>
12      <a href="one/">One</a>
13      <a href="two/">Two</a>
14      <a href="three/">Three</a>
15    </nav>
```

Imagen de [Wikipedia](#)



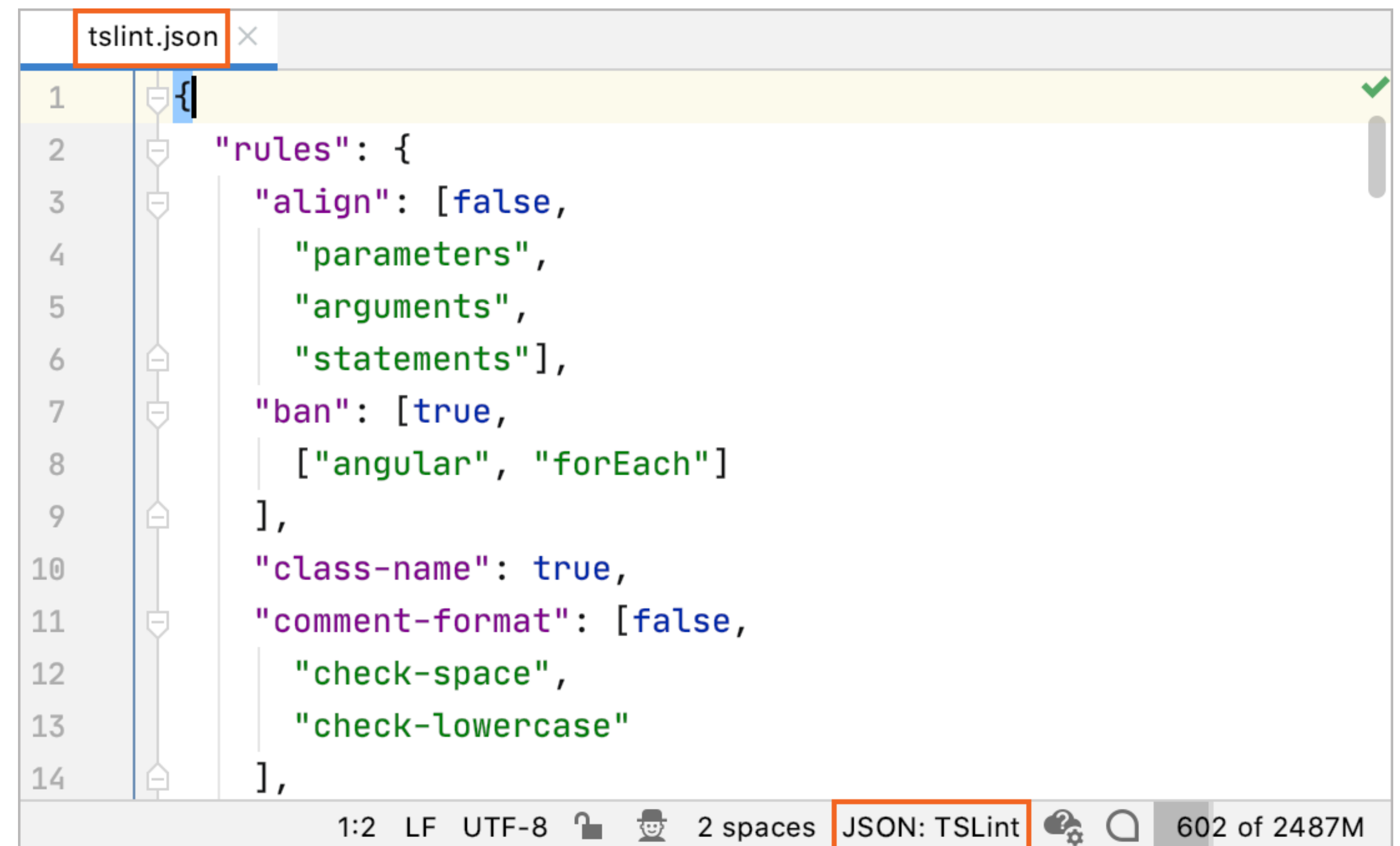
# Modelo de datos semiestructurado

## XML

```
<studentsList>
  <student id="1">
    <firstName>Greg</firstName>
    <lastName>Dean</lastName>
    <certificate>True</certificate>
    <scores>
      <module1>70</module1>
      <module12>80</module12>
      <module3>90</module3>
    </scores>
  </student>
  <student ind="2">
    <firstName>Wirt</firstName>
    <lastName>Wood</lastName>
    <certificate>True</certificate>
    <scores>
      <module1>80</module1>
      <module12>80.2</module12>
      <module3>80</module3>
    </scores>
  </student>
</studentsList>
```

Imagen de [Hubspot](#)

## JSON



```
tslint.json
1 {
2   "rules": {
3     "align": [false,
4       "parameters",
5       "arguments",
6       "statements"],
7     "ban": [true,
8       ["angular", "forEach"]
9     ],
10    "class-name": true,
11    "comment-format": [false,
12      "check-space",
13      "check-lowercase"
14    ],
```

Imagen de [JetBrains](#)

# Otros modelos de datos

## Array

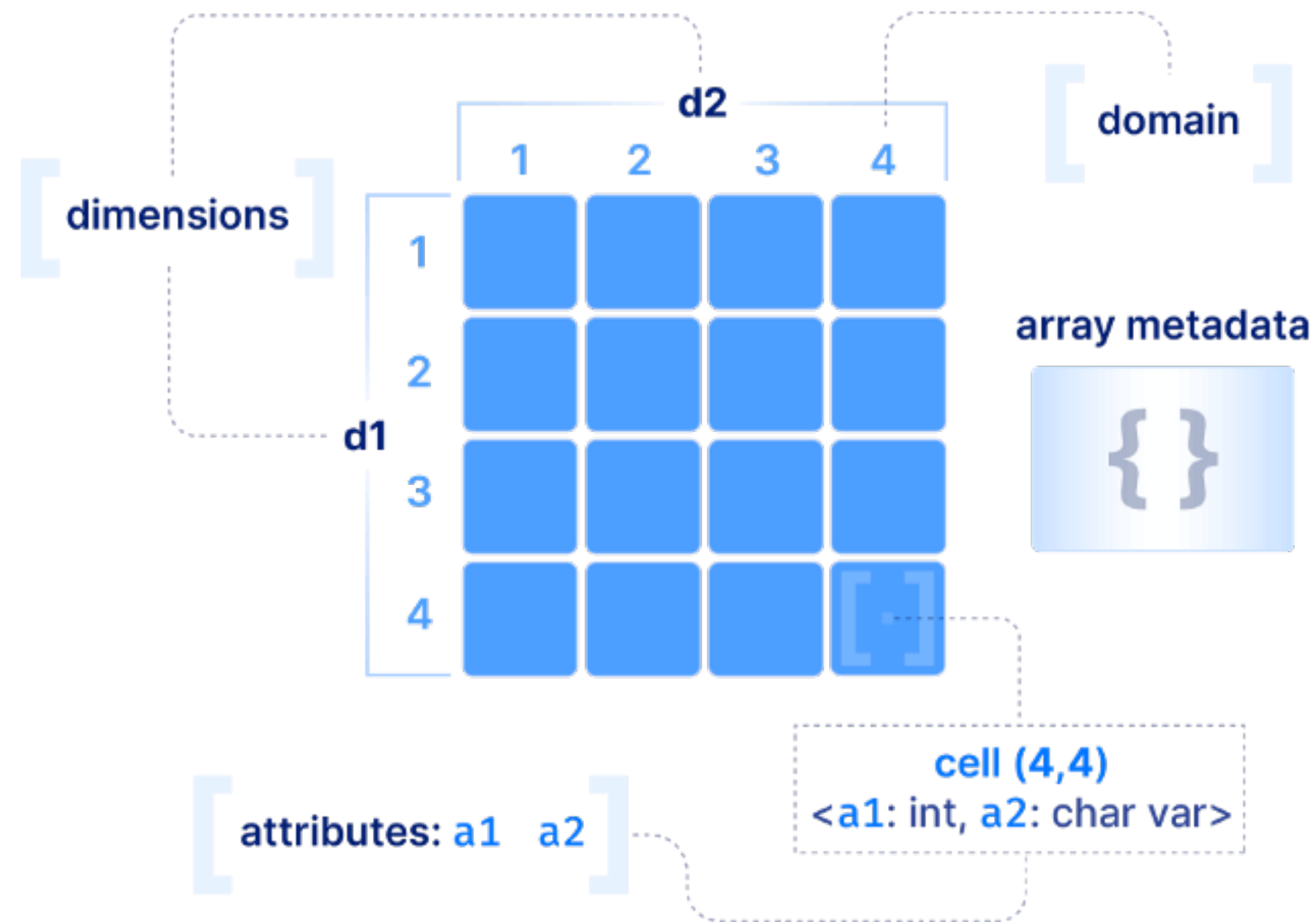


Imagen de [TileDB](#)

## Grafos

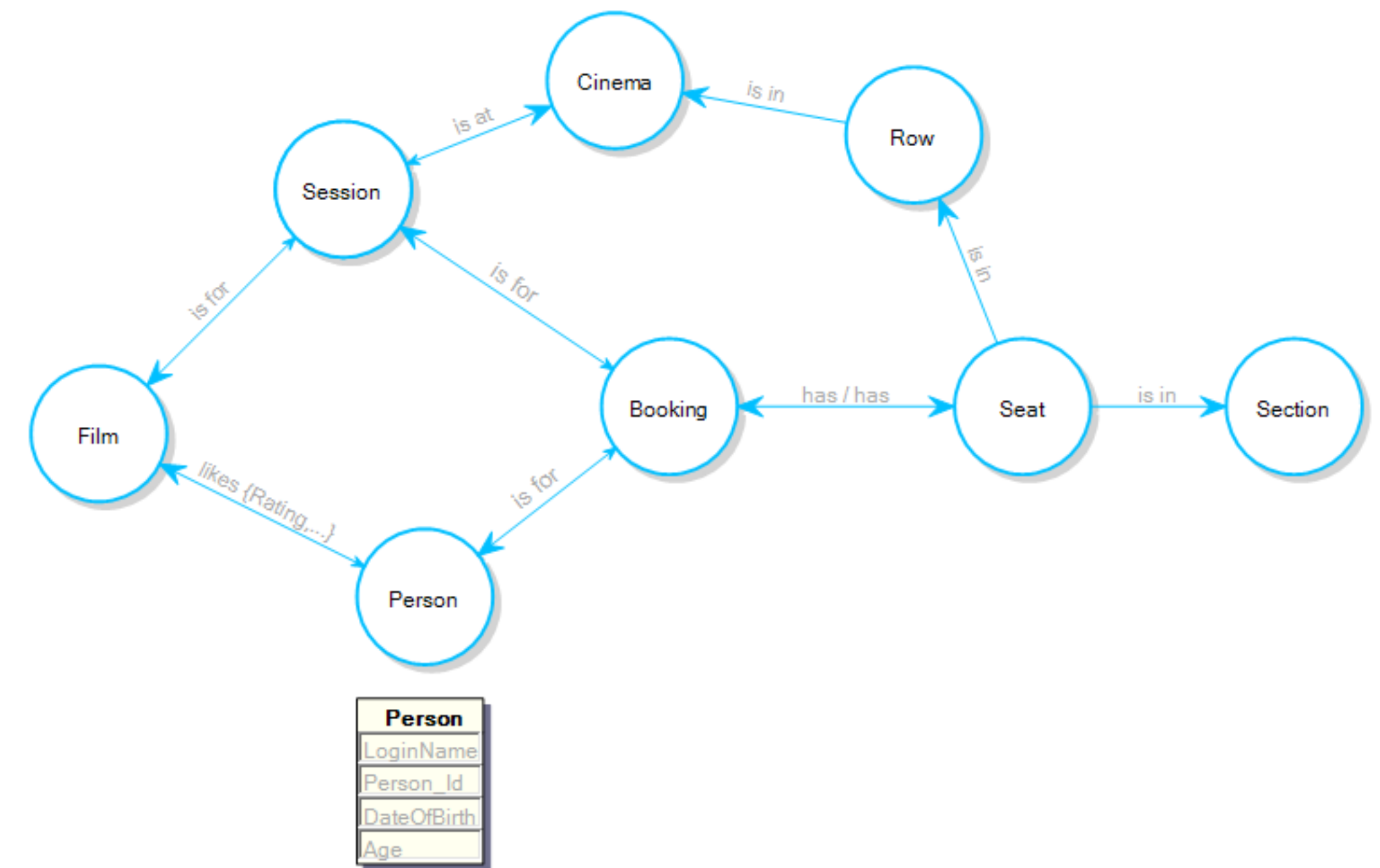


Imagen de [Towards Data Science](#)

# Data streams

Una secuencia posiblemente infinita/sin límite de datos o registros, los cuales pueden estar relacionados o no correlacionados. Usualmente cada registro tiene una etiqueta de tiempo e incluso una geográfica

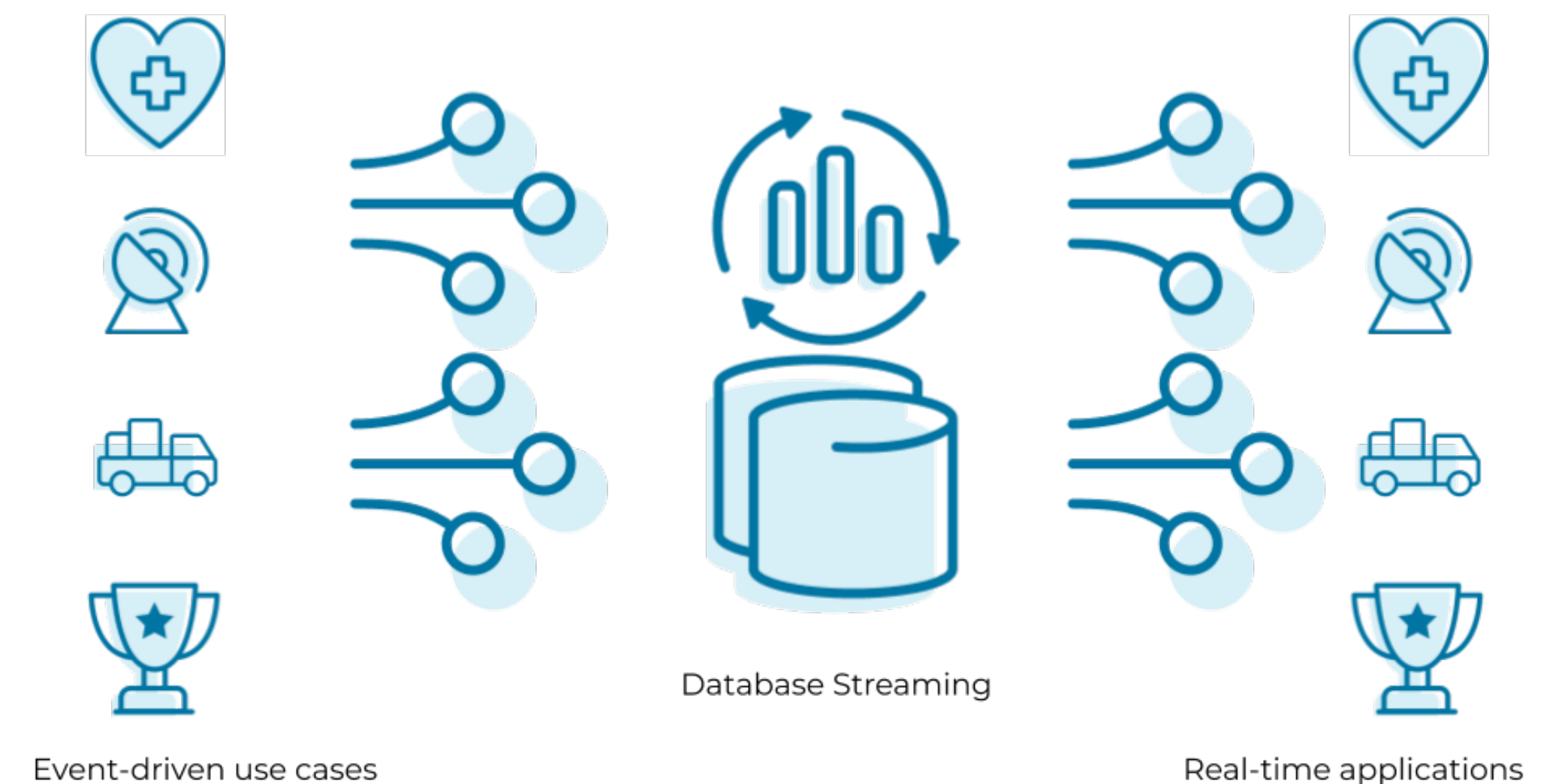


Imagen de [Confluent](#)



# Stream processing vs Batch processing

El procesamiento de Streams permite la toma dinámica de decisiones tan pronto como se genera y procesan los datos.

Implica cambios en la arquitectura para procesar datos en tiempo real y adaptarse a cambios en la frecuencia de adquisición de los datos.

No hay una límite en la cantidad de datos a procesar, pero sí en el tiempo de ejecución.

De manera general, las operaciones de procesamiento son más simples y de rápida ejecución.

# Apache Spark

Framework open source  
para procesamiento  
distribuido de datos en  
memoria y análisis iterativo  
sobre volúmenes masivos  
de datos.



Imagen de [Wikipedia](#)

# Apache Spark

Spark provee procesamiento en memoria (mayor velocidad), crea un framework de procesamiento de Big Data desde ingeniería de datos hasta analítica e inteligencia artificial, y ofrece flexibilidad al desarrollador a través de APIs en Python, Scala y Java.



Imagen de [Wikipedia](#)

# Apache Spark

Un trabajo típico de MapReduce crea iteraciones que requieren múltiples lecturas/escrituras al HDFS, las cuales consumen tiempo y resultan costosas; Spark resuelve este problema al mantener una gran parte de la información en memoria, evitando operaciones I/O y reduciendo en gran medida el tiempo de ejecución.

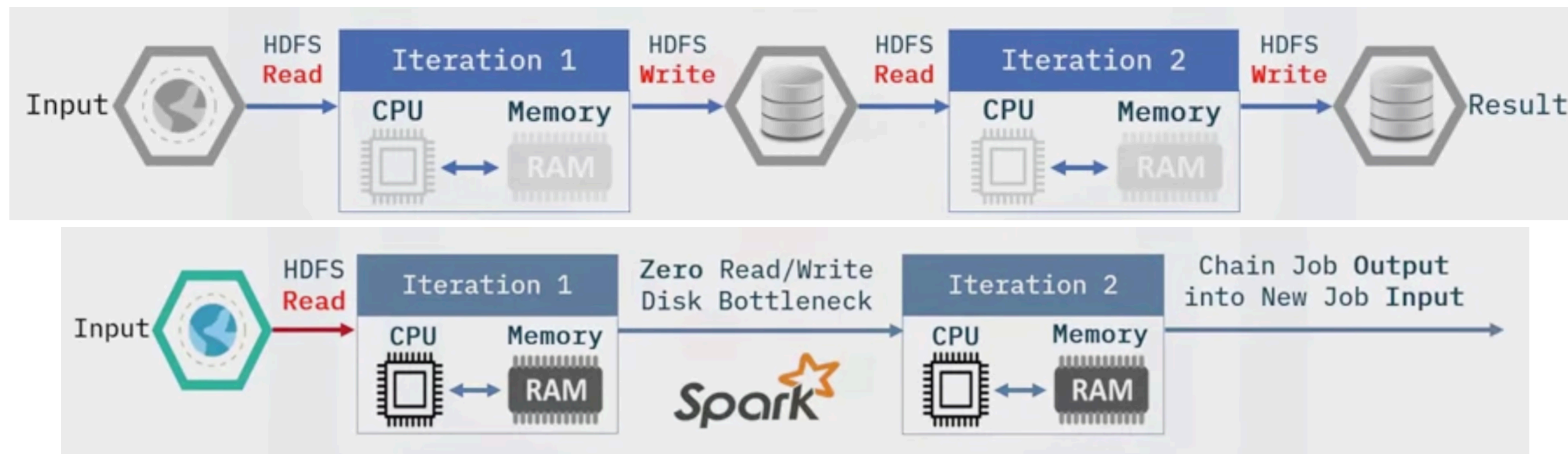


Imagen de Coursera



# Apache Spark

En Spark, los datos se reparten en una estructura llamada *Resilient Distributed Datasets (RDDs)* que son una colección **inmutable** de elementos, tolerante a fallos, que se distribuyen en los diferentes nodos y aceptan operaciones en paralelo.

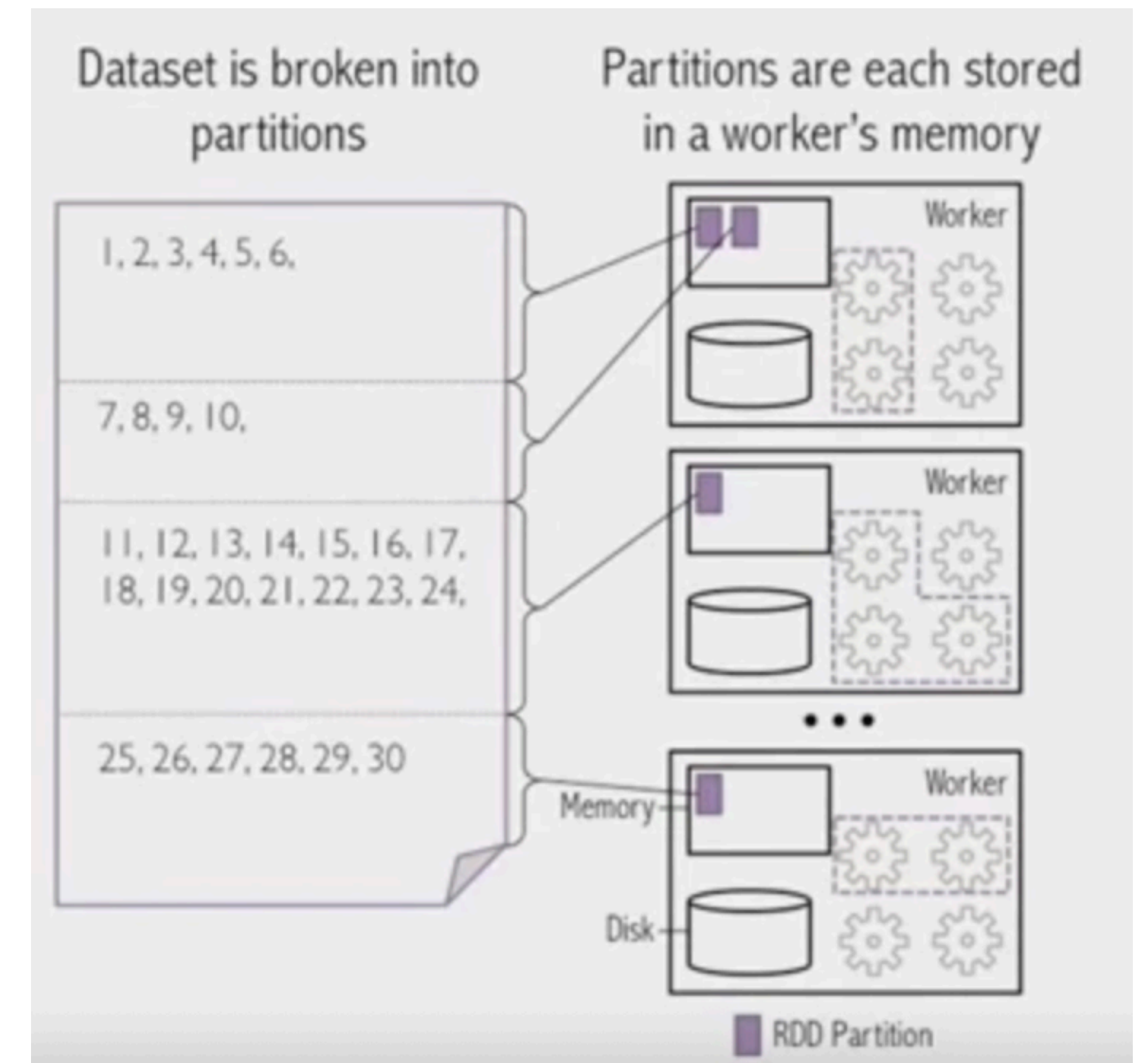


Imagen de Coursera



# Lecturas complementarias

Why Arrays as a Universal Data Model

What is a graph database?

Apache Spark documentation