Data storage: simple solutions

Introduction to Databases

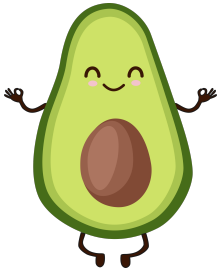
Hiba ALQASIR

2021-2022

télécom
saint-étienne
école d'ingénieurs / nouvelles technologies

Data

- Any numerical value, it can be:
  a string ("avocat"), an integer (2207), a date (22/11/2003)
- Always associated with the context allowing to know
  what information it represents.



Data vs. information

There is a subtle difference between **data** (raw value) and **information** (value and interpretative context).

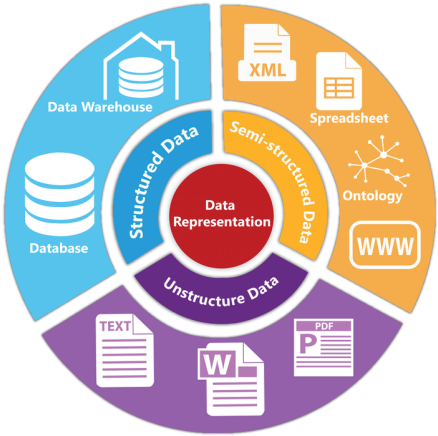
Where does data come from?

- From an application domain.
- Describes objects, facts or concepts

entities

Data representation

- The solution that helps to distinguish precisely and without ambiguity the elementary information.
- Database is a structure that stores organized data.



Source: Siriyasatien, Padet, et al.

Database

- A set of organized related data.
- Potentially large, but not necessarily.
- Stored in a persistent manner.

Does a file stored on your PC's disk conform to this definition?

Paris is the capital and most populous city of France, with an estimated population of 2175601 residents as of 2018.

State	Capital	Population
France	Paris	2175601

Database structure

- A database file necessarily has a structure that makes it possible to distinguish the data from one another and to represent their links.
- One of the simplest and most widespread structures, the CSV file.

CSV Files

In a CSV file:

- The elementary data are represented by fields delimited by commas.
- The fields are associated with each other by being placed in the same line.
- The lines are independent of each other.
- You can place as many lines as you want in a file,
  or change their order without changing the information they represent.

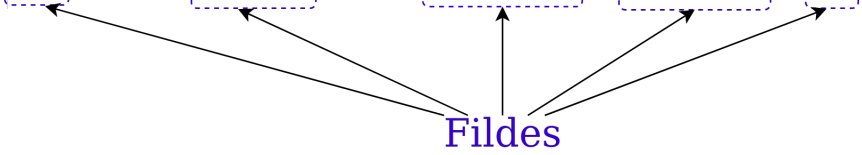
CSV Files

```
station_id, name          , latitude  , longitude , capacity
101       , Châteaucreux  , 45.442972 , 4.398855  , 32
102       , Chaléassiere  , 45.453601 , 4.383062  , 16
103       , Carnot        , 45.447093 , 4.385487  , 24
104       , Jacquard      , 45.441888 , 4.382492  , 12
105       , Jaures        , 45.440725 , 4.387295  , 32
106       , Dorian        , 45.43836  , 4.388435  , 32
107       , Clapier       , 45.436903 , 4.378789  , 16
```

CSV File

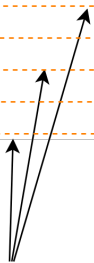
CSV Files

```
station_id, name          , latitude   , longitude , capacity
101       , Châteaucreux  , 45.442972  , 4.398855  , 32
102       , Chaléassiere  , 45.453601  , 4.383062  , 16
103       , Carnot        , 45.447093  , 4.385487  , 24
104       , Jacquard      , 45.441888  , 4.382492  , 12
105       , Jaures        , 45.440725  , 4.387295  , 32
106       , Dorian        , 45.43836   , 4.388435  , 32
107       , Clapier       , 45.436903  , 4.378789  , 16
```



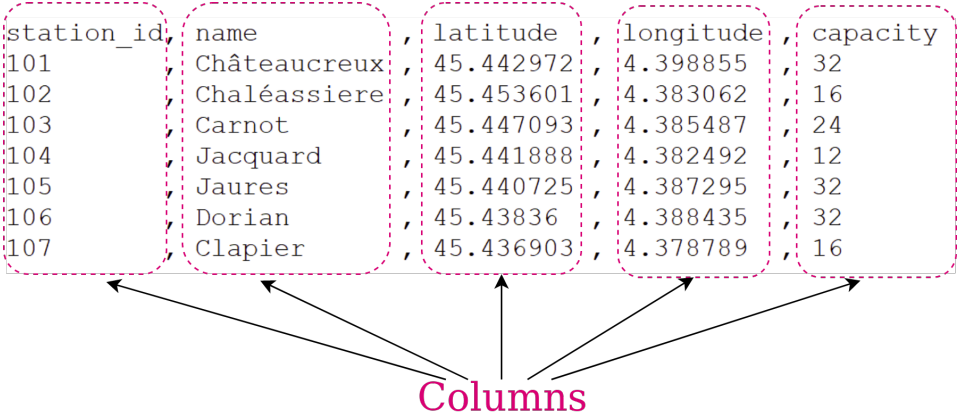
CSV Files

```
station_id, name          , latitude  , longitude , capacity
101       , Châteaucreux  , 45.442972 , 4.398855  , 32
102       , Chaléassiere  , 45.453601 , 4.383062  , 16
103       , Carnot        , 45.447093 , 4.385487  , 24
104       , Jacquard      , 45.441888 , 4.382492  , 12
105       , Jaures        , 45.440725 , 4.387295  , 32
106       , Dorian        , 45.43836  , 4.388435  , 32
107       , Clapier       , 45.436903 , 4.378789  , 16
```



Lines

CSV Files


station_id,name,latitude,longitude,capacity
101,Châteaucreux,45.442972,4.398855,32
102,Chaléassiere,45.453601,4.383062,16
103,Carnot,45.447093,4.385487,24
104,Jacquard,45.441888,4.382492,12
105,Jaures,45.440725,4.387295,32
106,Dorian,45.43836,4.388435,32
107,Clapier,45.436903,4.378789,16
Columns

CSV Files

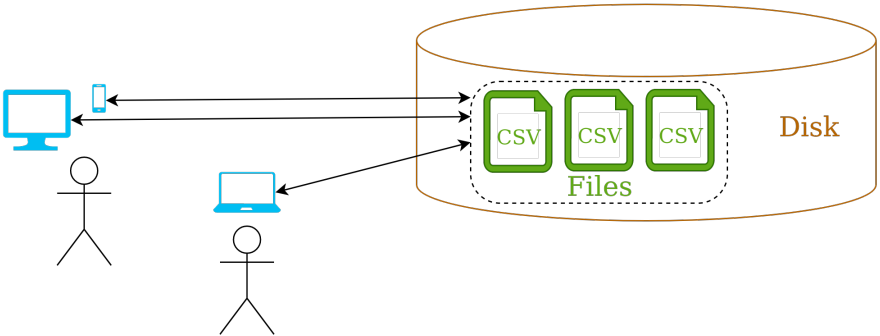
France , Paris , 543940 km^2

CSV Files

France , Paris , 543940 km^2
Germany , Berlin , 357386 km^2
Spain , Madrid , 505990 km^2

Can we be satisfied with this solution?

Can we build applications based directly on structured files,
for example CSV files?



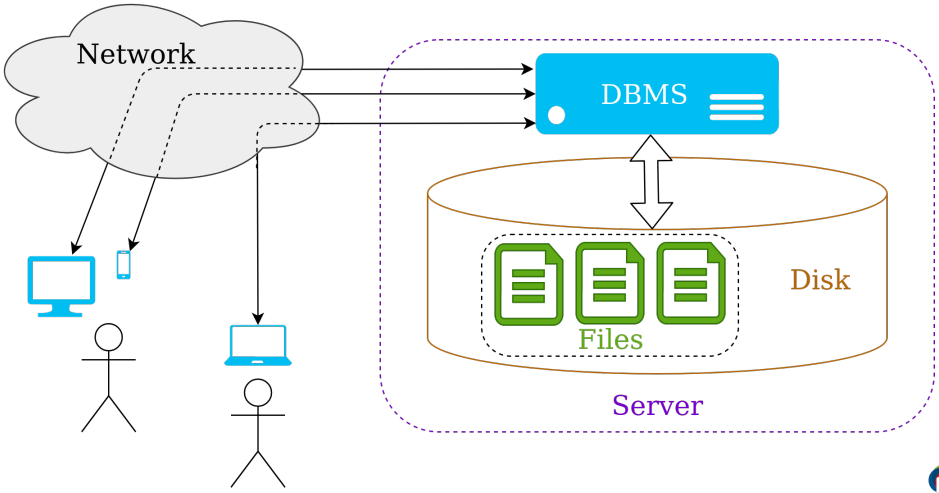
Challenges

- Heavy access to data, from many, changing apps.
- Data privacy and security.
- Concurrency control (access by multiple users).
- Performance: throughput, latency, etc.
- Reliability in the face of hardware crashes, bugs, bad user input, etc.

Database Management System (DBMS)

- DBMS handles the management of data stored in a database.
- Two major functions:
  1. Access to database files
     - guaranteeing their integrity
     - controlling concurrent operations
     - optimizing searches and updates
  2. Interactions with applications and users
     - thanks to query and manipulation languages

Database Management System (DBMS)



Operations on databases

- Design
  - Define the structure and data types.
- Construction
  - Create the data structures of the databases.
  - Populate DB with data.
- Manipulation of Data
  - Insert, delete, update.
  - Query.
  - Create reports.

Database History

- The concept of a database was around years before computers.
- Computers provide the means to improve data management.

Database History

- 1960s: Hierarchical and Network database systems
- 1970s: Relational database systems, SQL
- 1980s: Object-oriented database systems
- 1990s: MySQL, XML

Timeline of RDBMS

Database Today

Rank			DBMS	Database Model	Score		
Nov 2021	Oct 2021	Nov 2020			Nov 2021	Oct 2021	Nov 2020
1.	1.	1.	Oracle	Relational, Multi-model	1272.73	+2.38	-72.27
2.	2.	2.	MySQL	Relational, Multi-model	1211.52	-8.25	-30.12
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	954.29	-16.32	-83.35
4.	4.	4.	PostgreSQL	Relational, Multi-model	597.27	+10.30	+42.22
5.	5.	5.	MongoDB	Document, Multi-model	487.35	-6.21	+33.52
6.	6.	↑ 7.	Redis	Key-value, Multi-model	171.50	+0.15	+16.08
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model	167.52	+1.56	+5.90
8.	8.	8.	Elasticsearch	Search engine, Multi-model	159.09	+0.84	+7.54
9.	9.	9.	SQLite	Relational	129.80	+0.43	+6.48
10.	10.	10.	Cassandra	Wide column	120.88	+1.61	+2.13

Source: db-engines.com

Database types

- Relational: data is organized in tables.
- Non-relational: non-tabular form.

Relational Database SQL

- Structured Query Language (SQL).
- Tables, columns, and rows.

Popular SQL databases

- Oracle
- MySQL
- Microsoft SQL Server
- PostgreSQL
- Microsoft Access
- MariaDB

Advantages of a relational database

- The data is structured into categories without difficulty.
- The data is coherent with respect to input, significance and navigability.
- Relationships can be easily defined between data points.

Non-relational Database NoSQL

- Not Only SQL (NoSQL).
- Less structured/confined in format.
- More flexibility and adaptability.

Popular NoSQL databases

- MongoDB
- Redis
- Elasticsearch
- Cassandra
- Splunk
- Amazon DynamoDB

Advantages of a non-relational database

- Data is not restricted to a particular structure.
- More flexibility.
- Dynamic analysis.

Where databases are used?

Databases are used all over the place

- Banks:
  track customer accounts, balances and deposits.
- Retail stores:
  store available quantities, prices, customer and sales information.

What is data in a database?

- Any kind of data stored in the computer's memory.
- To be used by a website, application or other business clients.