# Practical AI

Emanuele Fabbiani

# Problem Statement

# Discussion

Let’s try to build a couple of problem statements.

# Problem 1

You are a manager in a challenger bank (say Aidexa or CF+).

You want to build a new credit rating model to evaluate loan applications from small and medium enterprises.

# Problem 2

You are a manager in a utility (say A2A or ENEL).

You want to build a new forecasting model to predict the power load demand in Italy.

# Problem Statement Checklist

1. **Who will use the system?**
2. Why will they use it?
3. What is the goal the company wants to achieve?
4. How can we measure such progress towards the goal?
5. What data do we need?
6. When do we need the data to be available?
7. What outcome should we produce?
8. When should the outcome become available?
9. What constraints should we comply with (e.g. regulation, business processes, ...)?
10. (Bonus) What is the budget and the resources we can use to build and run the system?

# Problem Statement Checklist

1. Who will use your system?
2. **Why will they use it?**
3. What is the goal the company wants to achieve?
4. How can we measure such progress towards the goal?
5. What data do we need?
6. When do we need the data to be available?
7. What outcome should we produce?
8. When should the outcome become available?
9. What constraints should we comply with (e.g. regulation, business processes, ...)?
10. (Bonus) What is the budget and the resources we can use to build and run the system?

# Problem Statement Checklist

1. Who will use your system?
2. Why will they use it?
3. **What is the goal the company wants to achieve?**
4. How can we measure such progress towards the goal?
5. What data do we need?
6. When do we need the data to be available?
7. What outcome should we produce?
8. When should the outcome become available?
9. What constraints should we comply with (e.g. regulation, business processes, ...)?
10. (Bonus) What is the budget and the resources we can use to build and run the system?

# Problem Statement Checklist

1. Who will use your system?
2. Why will they use it?
3. What is the goal the company wants to achieve?
4. **How can we measure such progress towards the goal?**
5. What data do we need?
6. When do we need the data to be available?
7. What outcome should we produce?
8. When should the outcome become available?
9. What constraints should we comply with (e.g. regulation, business processes, ...)?
10. (Bonus) What is the budget and the resources we can use to build and run the system?

# Problem Statement Checklist

1. Who will use your system?
2. Why will they use it?
3. What is the goal the company wants to achieve?
4. How can we measure such progress towards the goal?
5. **What data do we need?**
6. When do we need the data to be available?
7. What outcome should we produce?
8. When should the outcome become available?
9. What constraints should we comply with (e.g. regulation, business processes, ...)?
10. (Bonus) What is the budget and the resources we can use to build and run the system?

# Problem Statement Checklist

1. Who will use your system?
2. Why will they use it?
3. What is the goal the company wants to achieve?
4. How can we measure such progress towards the goal?
5. What data do we need?
6. **When do we need the data to be available?**
7. What outcome should we produce?
8. When should the outcome become available?
9. What constraints should we comply with (e.g. regulation, business processes, ...)?
10. (Bonus) What is the budget and the resources we can use to build and run the system?

# Problem Statement Checklist

1. Who will use your system?
2. Why will they use it?
3. What is the goal the company wants to achieve?
4. How can we measure such progress towards the goal?
5. What data do we need?
6. When do we need the data to be available?
7. **What outcome should we produce?**
8. When should the outcome become available?
9. What constraints should we comply with (e.g. regulation, business processes, ...)?
10. (Bonus) What is the budget and the resources we can use to build and run the system?

# Problem Statement Checklist

1. Who will use your system?
2. Why will they use it?
3. What is the goal the company wants to achieve?
4. How can we measure such progress towards the goal?
5. What data do we need?
6. When do we need the data to be available?
7. What outcome should we produce?
8. **When should the outcome become available?**
9. What constraints should we comply with (e.g. regulation, business processes, ...)?
10. (Bonus) What is the budget and the resources we can use to build and run the system?

# Problem Statement Checklist

1. Who will use your system?
2. Why will they use it?
3. What is the goal the company wants to achieve?
4. How can we measure such progress towards the goal?
5. What data do we need?
6. When do we need the data to be available?
7. What outcome should we produce?
8. When should the outcome become available?
9. **What constraints should we comply with (e.g. regulation, business processes, ...)?**
10. (Bonus) What is the budget and the resources we can use to build and run the system?

# Problem Statement Checklist

1. Who will use your system?
2. Why will they use it?
3. What is the goal the company wants to achieve?
4. How can we measure such progress towards the goal?
5. What data do we need?
6. When do we need the data to be available?
7. What outcome should we produce?
8. When should the outcome become available?
9. What constraints should we comply with (e.g. regulation, business processes, ...)?
10. **(Bonus) What is the budget and the resources we can use to build and run the system?**


Data

# What is data?

Information, especially facts or numbers, collected to be examined and considered and used to help decision-making, or information in an electronic form that can be stored and used by a computer. [1]

[1] https://dictionary.cambridge.org/it/dizionario/inglese/data

# HISTORY OF DATA

19,000 BC

The Ishango bone holds the first evidence of data collection and storage.

1600s


John Graunt introduces the concept of data analysis in 1663.

1800s

Herman Hollerith designs a machine that helped complete the US census in 1890.

1900s

Fritz Pfleumer invents the magnetic tape which later inspired the invention of floppy disks and hard disk drives.

1990s

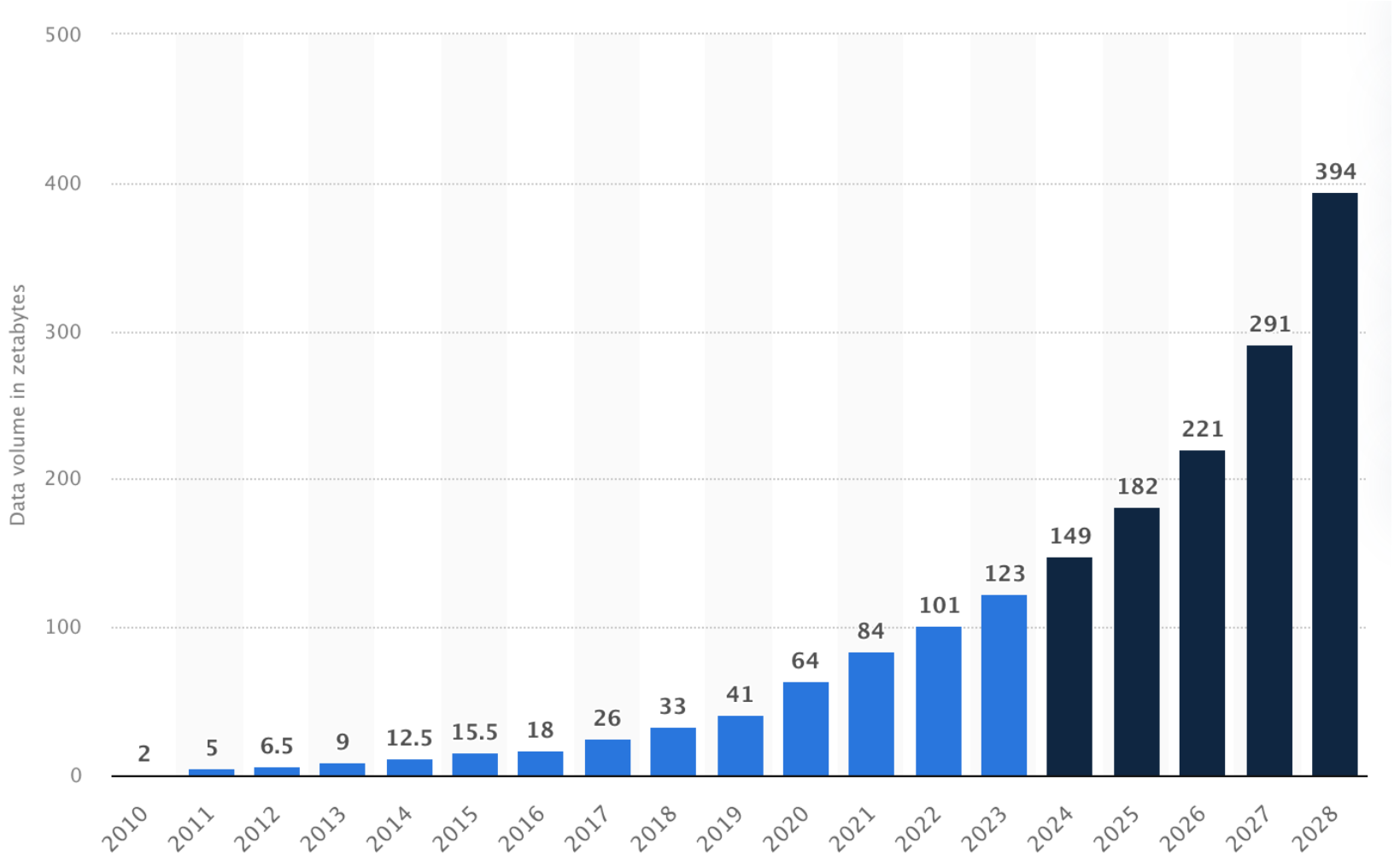
Sir Tim Berners Lee invents the World Wide Web.


The Economist
Crunch time in France
Ten years on: banking after the crisis
South Korea's unfinished revolution
Biology, but without the cells
MAY 6TH–12TH 2017
The world's most valuable resource
Google
amazon
Google
TESLA
TESLA
Data and the new rules of competition

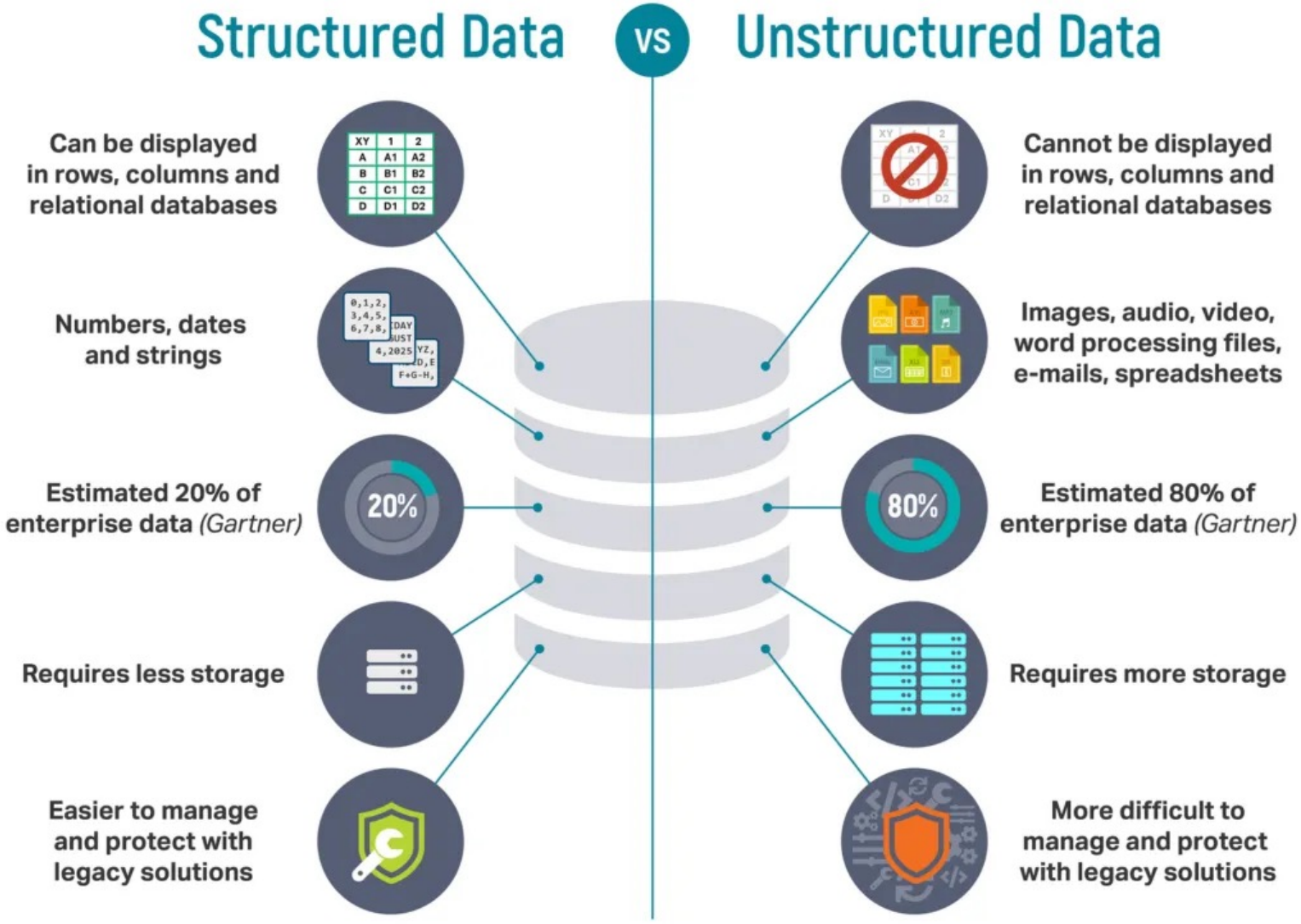
# How Did Data Become Valuable?

- In the early 2000s, many businesses transitioned to digital platforms, gaining the ability to track user behaviour.
- This data became an asset, enabling companies to make informed business decisions and boost profits.
- Data-driven business models transformed industries:
  - Companies like Google and Meta **rely entirely** on data for their operations.
  - Businesses such as Amazon, Netflix, and Uber became larger and much **more profitable** due to data utilization.
  - Traditional industries like banking leveraged data to **decrease expenses**.
- The availability of hardware and strong economic incentives to harness data spurred rapid advancements in software and analytical methods.

# How Much Data Do We Create?




https://www.statista.com/statistics/871513/worldwide-data-created/

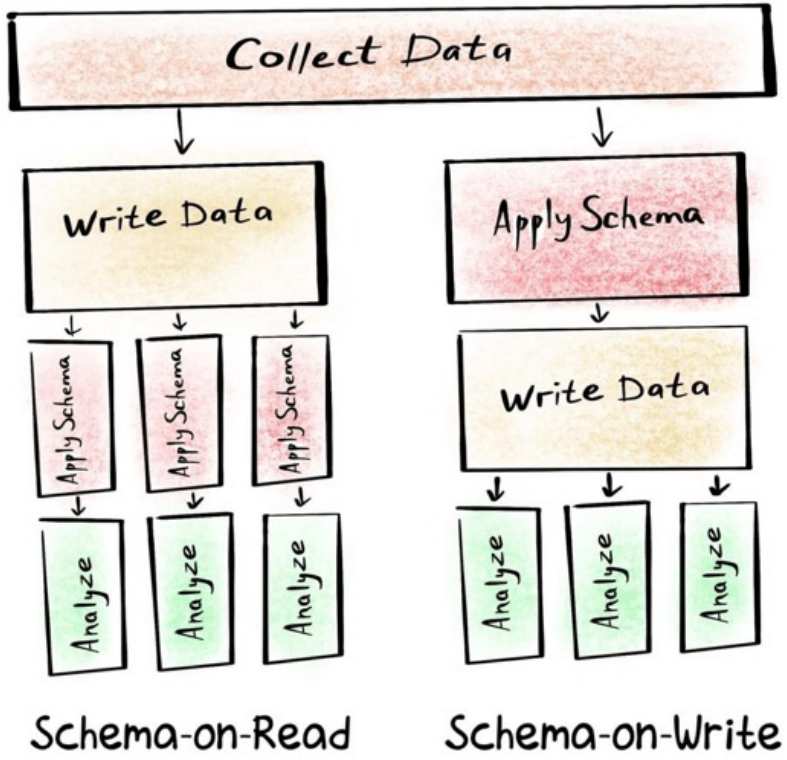



https://www.mongodb.com/resources/basics/unstructured-data/structured-vs-unstructured

# Data Schemas

Structured data follow a **schema-on-write** approach, requiring data to conform to a predefined schema before it can be written.

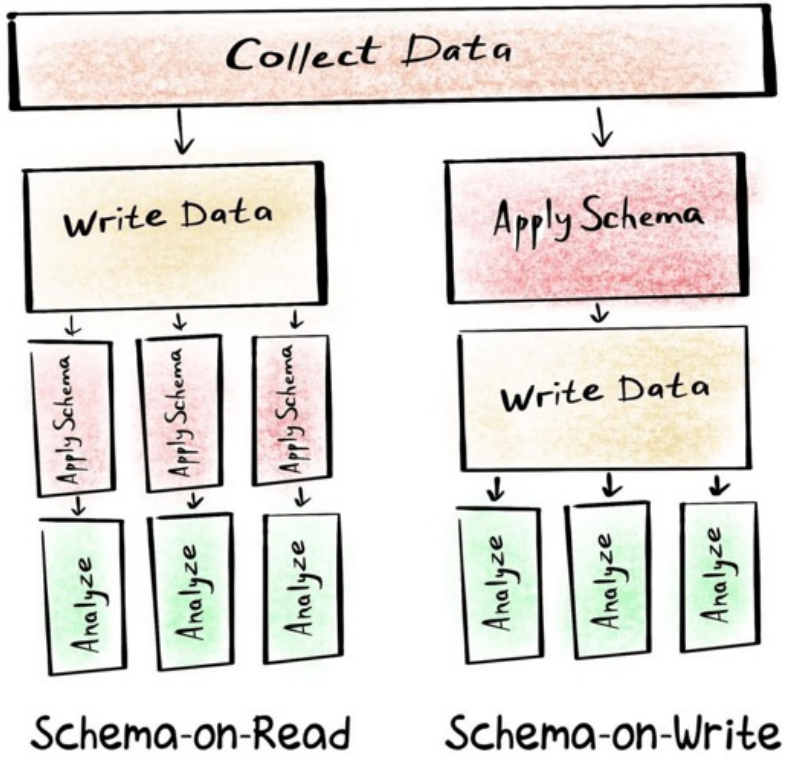
Unstructured data follow a **schema-on-read** approach, allowing data to be written in its raw form, with a suitable schema applied during reading.



# Pros and Cons

Structured data are:

- **Harder to collect**, as they must fit into pre-defined schemas
- **Easier to analyse**, for the same reason
- **Easier to share** with non-technical stakeholders, as everyone can use Excel
- Historically **easier to manage**, as unstructured data only became popular in the 2010s




Databases

# What is a Database?

A database is an **organised** collection of data or a type of data store based on the use of a **database management system** (DBMS), the software that interacts with end users, applications, and the database itself to capture and analyse the data.

https://en.wikipedia.org/wiki/Database

# So, Excel...

Spreadsheets are great tools, but their **improper and indiscriminate** use causes all sorts of pain. Data gets lost, updates cannot be tracked, and accountability is impossible.

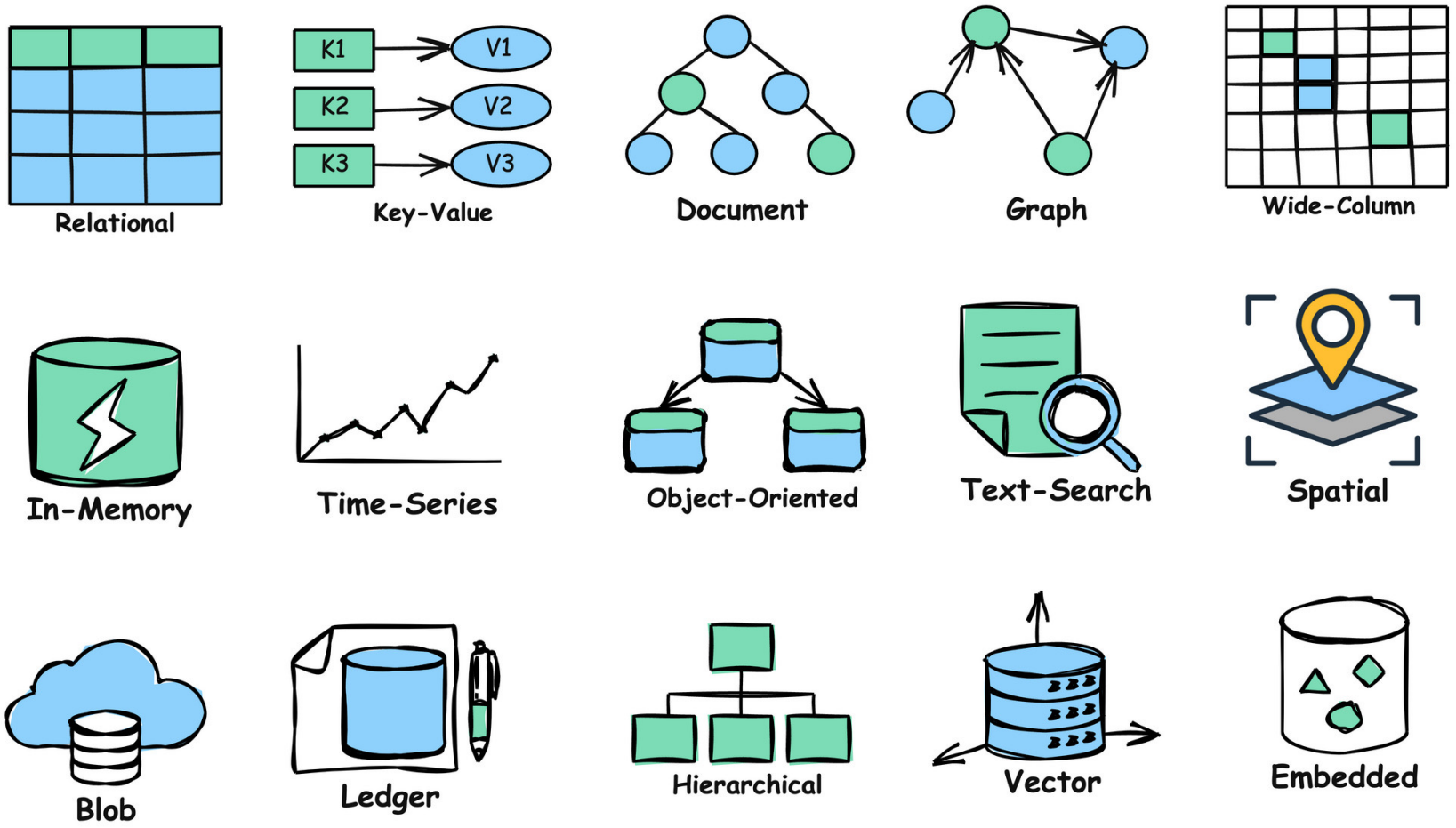
Use Excel for:

- **Individual** analysis
- Ephemeral data

Never use Excel for:

- Business-critical data
- **Sharing**
- **Persistent** storage

# Database Types



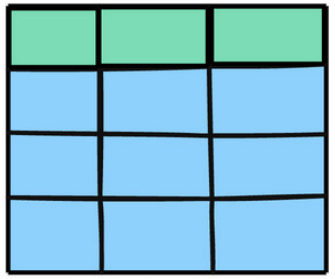

https://blog.algomaster.io/p/15-types-of-databases

# Database Types

We will focus solely on **relational databases** and **blob storage**.

Blob storage is not a true database but a method for storing unstructured data.

While you may never need to select a database yourself, it's important to understand that **many options exist**, each with its advantages and disadvantages.



Relational

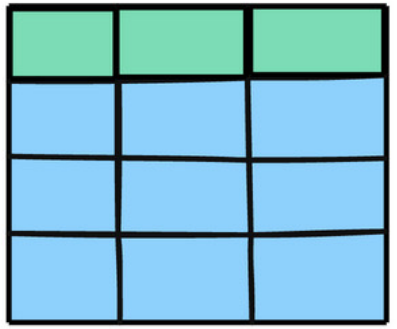
Blob

# Relational Databases (1)

A relational database is a type of database that organizes data into **rows and columns**, which collectively form a **table** where the records are related to each other.

Data is typically structured across multiple tables, which can be **joined** together.

Analysts use **SQL queries** to combine different data points and summarize business performance, allowing organizations to gain insights, optimize workflows, and identify new opportunities.



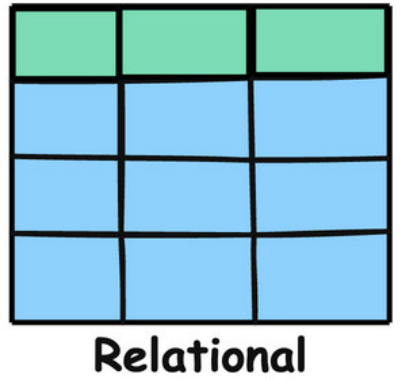

https://www.ibm.com/think/topics/relational-databases

# Relational Databases (2)

The relational database management system (RDBMS) is the **database software** that allows users to create, update, insert, or delete data in the system and provides:

- Data structure
- Multi-user access
- Privilege control
- Network access

Examples of popular RDBMS systems include MySQL and PostgreSQL.

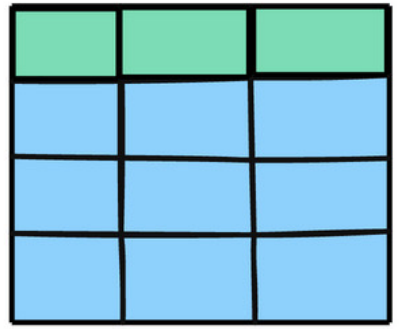



https://www.ibm.com/think/topics/relational-databases

# SQL (1)

Structured Query Language (SQL) is the standard programming language for interacting with relational database management systems.

```
SELECT COMPANY_NAME, SUM(TRANSACTION_AMOUNT)
FROM TRANSACTION_TABLE A
LEFT JOIN CUSTOMER_TABLE B
ON A.CUSTOMER_ID = B.CUSTOMER_ID
WHERE YEAR(DATE) = 2022
GROUP BY COMPANY_NAME
ORDER BY SUM(TRANSACTION_AMOUNT) DESC
LIMIT 10
```




https://www.ibm.com/think/topics/relational-databases

# SQL (2)

Different from Python, SQL is a **declarative** language.

You tell the RDBMS what to do, **it decides** what is the best sequence of steps (algorithm) to accomplish your task.

SQL is **much more limited** than Python: it only serves to query data.

```
SELECT COMPANY_NAME, SUM(TRANSACTION_AMOUNT)
FROM TRANSACTION_TABLE A
LEFT JOIN CUSTOMER_TABLE B
ON A.CUSTOMER_ID = B.CUSTOMER_ID
WHERE YEAR(DATE) = 2022
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10
```

# Blob Storage (1)

Blob storage is a type of storage for **unstructured data**.

A "blob", which is short for Binary Large Object, is a mass of data in binary form that **does not necessarily conform** to any file format.

Blob storage keeps these masses of data in non-hierarchical storage areas called **data lakes**.


https://www.cloudflare.com/learning/cloud/what-is-blob-storage/

# Blob Storage (2)

Blob storage is a cloud-native technology designed to support **unstructured data.**

However, due to its cost-effectiveness and scalability compared to traditional databases, it is increasingly being used to **store structured data for analytical purposes.**

This is made possible by tools that enable **queries on collections of CSV or Parquet** files.

# Resources

IBM, *What is a relational database?:* a quick introduction to relational databases.

W3School, *SQL Tutorial:* a complete tutorial to learn SQL from zero (not required for this class).

# Live Coding

- Setting up docker and VS Code
- Cloning the repository
- Inspecting a relational database
- Learning about primary and foreign keys
- Playing with SQL
- Inspecting a blob storage
- Playing with Google Cloud Storage

# Discussion

Let’s try to discuss a couple of real-world datasets.

# Case 1

You are a manager in a challenger bank (say Aidexa or CF+).

You want to build a new credit rating model to evaluate loan applications from small and medium enterprises.

You have numeric and categorical data (revenues, category, employees of each company) as well as documents (income statements, ID cards, ...)

# Case 2

You are a manager in a utility (say A2A or ENEL).

You want to build a new forecasting model to predict the power load demand in Italy.

You have time series data of power demand, generation, and price, as well as meteorological data.


The End