# Review of Data Fundamentals

Data includes facts, observations, numbers, symbols, images or a mix of everything.

Data structure

We have:

* Structure Data: which follows a predefined format and adheres to strict schema, it includes tables with rows and columns and ensures consistency and easy retrieval. Ex: SQL database, spreadsheet, online forms.
* Unstructured data: Lacks a specific format or organisation, does not conform to any predefined rules. Ex: text file that contain free form document, Media files, such video, images, audio and video, web pages, and social media contents.
* Semi structure data: it possess some organisational properties, does not follow tabular structure, and employs hierarchical structures or tags. Provides a balance between flexibility and structure. Ex: Json file, XML documents, email.

Data sources

Traditional databases, web scraping, API, flat files and XML data sets, IOT devices with sensors, social media platforms data streams and feeds.

**Common Files format**

We have delimited:

* Delimited text files: Rows of variables separated by character such as csv, tsv files.
* Spreadsheet: data exists in rows and columns for manipulation, creates csv files.
* Language files: includes XML and JSON, set rules and structures for encoding data.

Data Repositories

They actively store, manage and organise data, they offer a structured framework for retrieval and administration, categories include: Relational databases and non relational databases.

1. Relational Databases:

Includes structured data stored in related tables to minimise duplication, along with supporting system known as RDBMS such IBM DB2, Microsoft SQL Server Oracle and MySQL.

Relational Databases are primarily designed for OLTP systems, store high volumes of operational data. Ensures transactional integrity. OLAP includes various storage solutions, focuses on querying and analysing large data sets. For example Sourcing data form CRM generating sales.

1. Non relational databases:

Offer flexibility in handling diverse and unstructured data, examples include MongoDB, Cassandra and redis.

# Information and Data models

An Information Model is an abstract, formal representation of entities that includes their properties, relationships and the operations that can be performed on them.

An Information Model is at the conceptual level and defines relationships between objects. Data Models are defined at a more concrete level, are specific and include details. A data model is the blueprint of any database system.

There are several types of Information Models. The most familiar is the Hierarchical, typically used to show organization charts.

The root of the tree is the parent node followed by child nodes. A child node cannot have more than one parent; however, a parent can have many child nodes. The first hierarchical database management system was the Information Management System released by IBM in 1968 and was originally built as the database for the Apollo space program.

Relational Model

The Relational Model is the most used data model for databases because this model allows for data independence. Data is stored in a simple data structure, tables. This provides logical data independence, physical data independence, and physical storage independence.

An Entity-Relationship Data Model, or ER Data Model, is an alternative to a relational data model. Using a simplified library database as an example, this figure shows an Entity-Relationship Diagram, or ERD, that represents entities (called tables) and their relationships.

Entity Relationship Model

An Entity-Relationship Model proposes thinking of a database as a collection of entities. Rather than being used as a model on its own, the ER Model is used as a tool to design relational databases.

In the ER Model, entities are objects that exist independently of any other entities in the database. It is simple to convert an ER Diagram into a collection of tables. The building blocks of an ER Diagram are **entities and attributes**. Entities have attributes, which are the data elements that characterise the entity. Attributes tell us more about the entity. In an ER Diagram, an entity is drawn as a **rectangle**, and attributes are drawn as **ovals**. Entities can be a noun (person, place, or thing).