ÉCOLE SUPÉRIEURE EN SCIENCES ET TECHNOLOGIES DE L'INFORMATIQUE ET DU NUMÉRIQUE

FUNDAMENTALS OF DATA SCIENCE AND DATA MINING

CHAPTER 2:

DATA INTEGRATION

Dr. Chemseddine Berbague

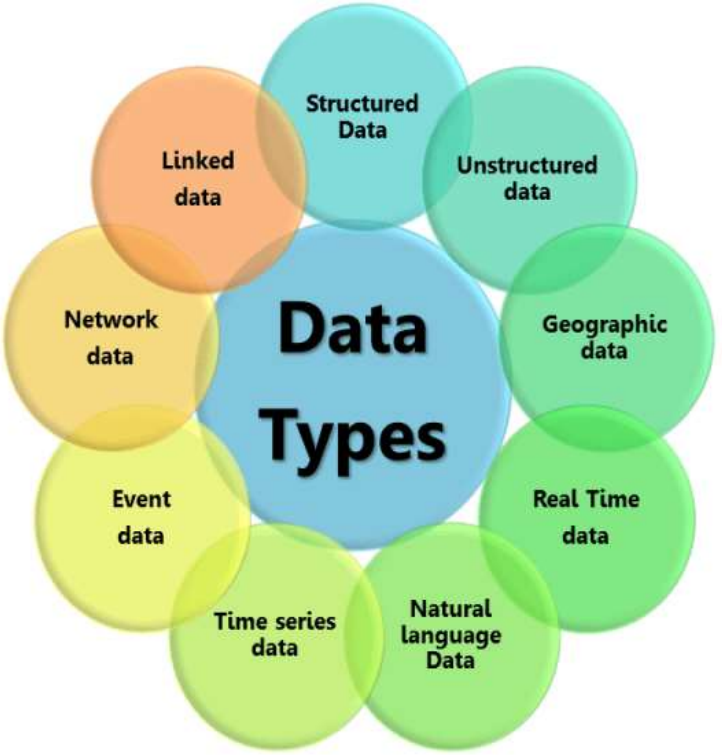
2024-2025

CONTENT

- Data Structures and Types
- Data Collection Methods
- Data Storage Systems
- Introduction to Big Data
- Techniques of Data Integration:
  - Duplicates Detection and Resolution
  - Scheme Matching
  - Semantic Integration
  - Value Alignment
  - ETL Example.
- Data Integration Challenges
- Ethics and Privacy in Data Integration

INTRODUCTION

- **Data** growth is related to the development of **(a)** new computations machines (such as computers, self-driving cars, sensors, …etc.), and **(b)** new storage and connectivity technologies led to generating more information everywhere while exerting daily activities (i.e., automatic, professional or entertaining activities). Such data can take different forms in terms of storage strategy (i.e., centralized or distributed data), and the type of the structure.

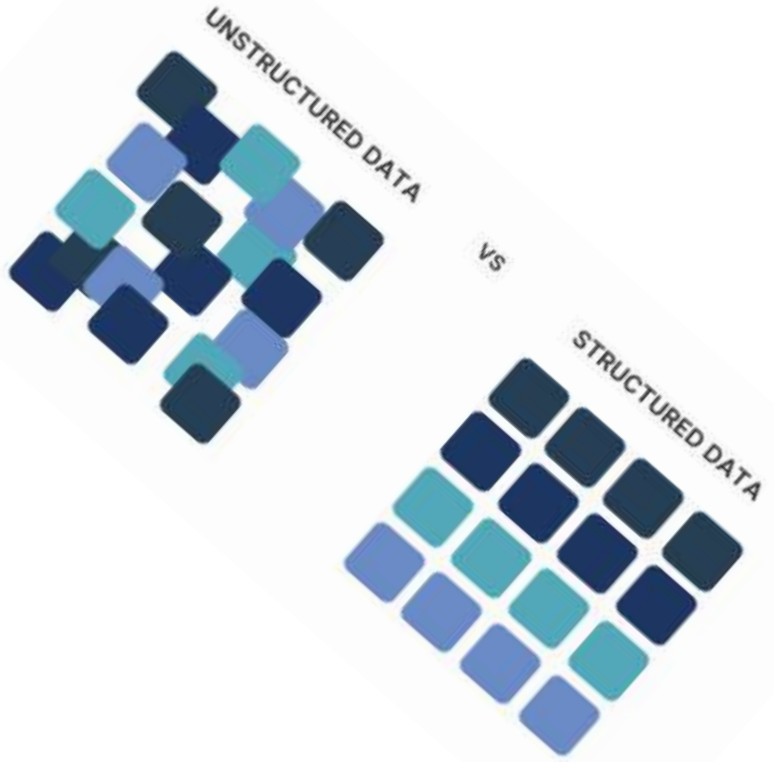


DATA STRUCTURES

- Mainly, IoT devices such as sensors, social websites, and medical reports are sources of data. In this contexts, the data are of different forms: unstructured, "quasi" structure, semi- structure, and structured such as the form of tables (rows and columns) or graphs with a lot of missing values, or noisy images.
- Such data requires specific approaches, tools, and technologies to handle it such as massive parallel processing MPP, and distributed computation architectures.

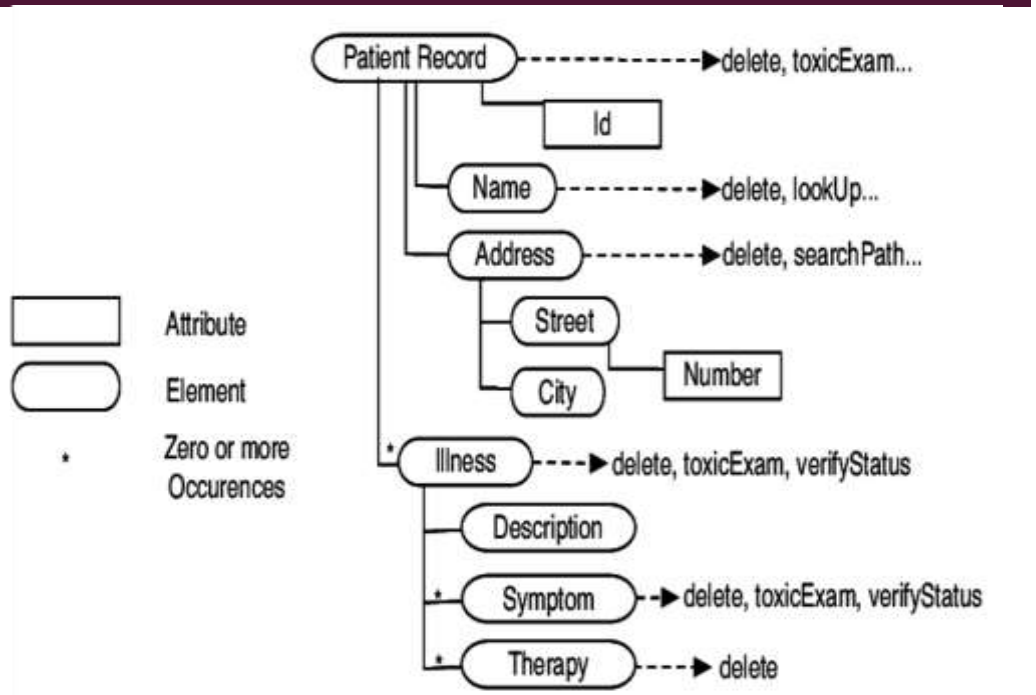
DATA STRUCTURES

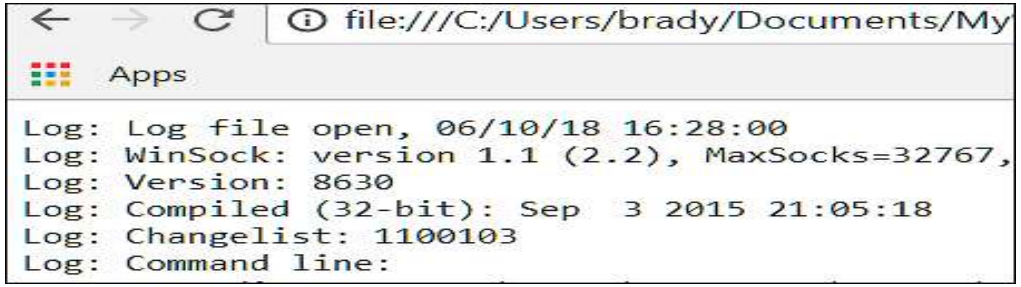
- **Structured data:** relational databases are a typical example of structure data, where different values are stored in form of rows, columns, within linked tables. Each data value has a type and characteristics of some specific rules. Another example of structured data is the calculation sheets (e.g., csv, or excel files …etc.), or **NoSQL** tables holding structured tables …etc.
- **Unstructured data:** raw data from text files (pdf, word …etc) or multi-media files (images, videos …etc).



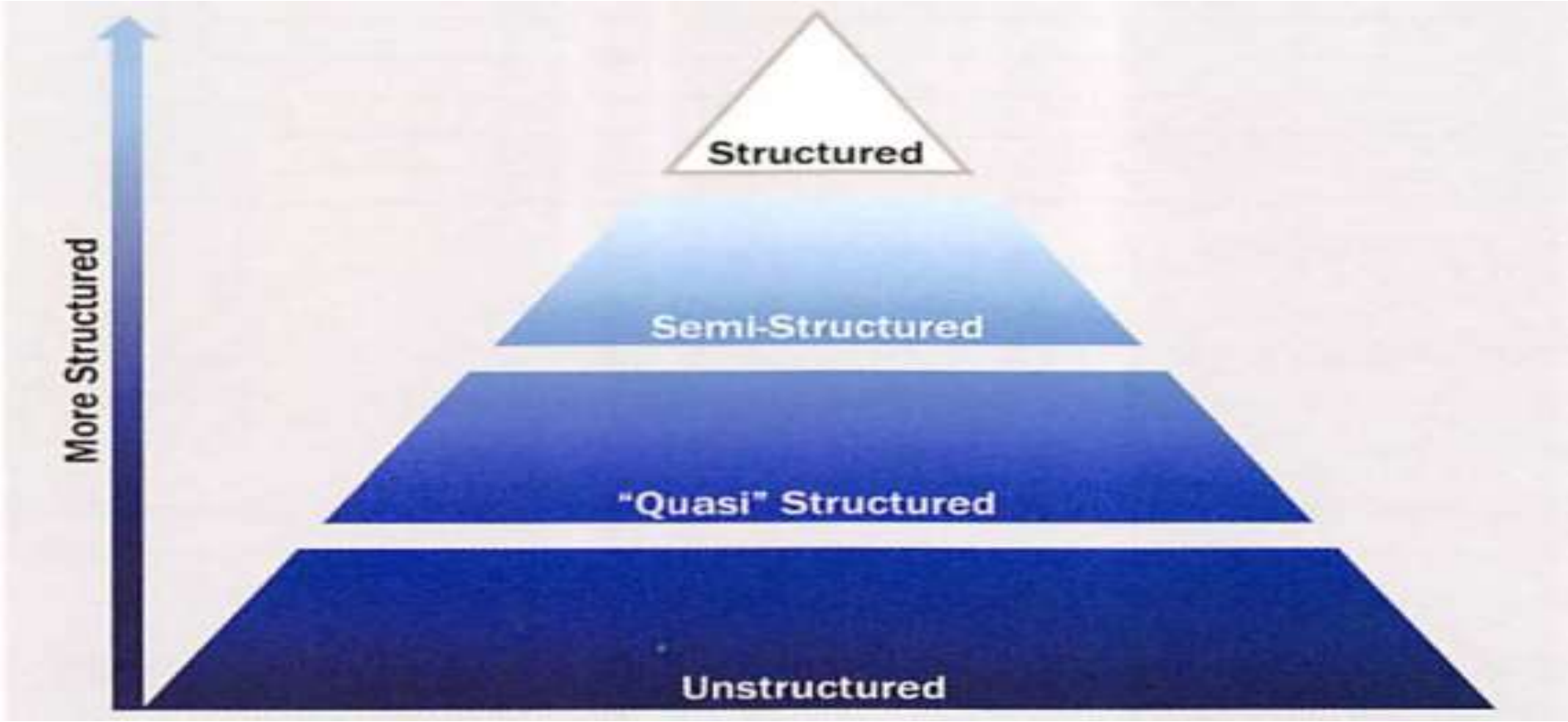
DATA STRUCTURES

- **Semi-structured data:** In particular, textual data in form of a **tree** or **graph** such as in JSON files, XML files are examples of semi-structured data.
- **Quasi-structured data:** textual data that can be formatted using tools, clicks stream on the web are a typical example of such type.

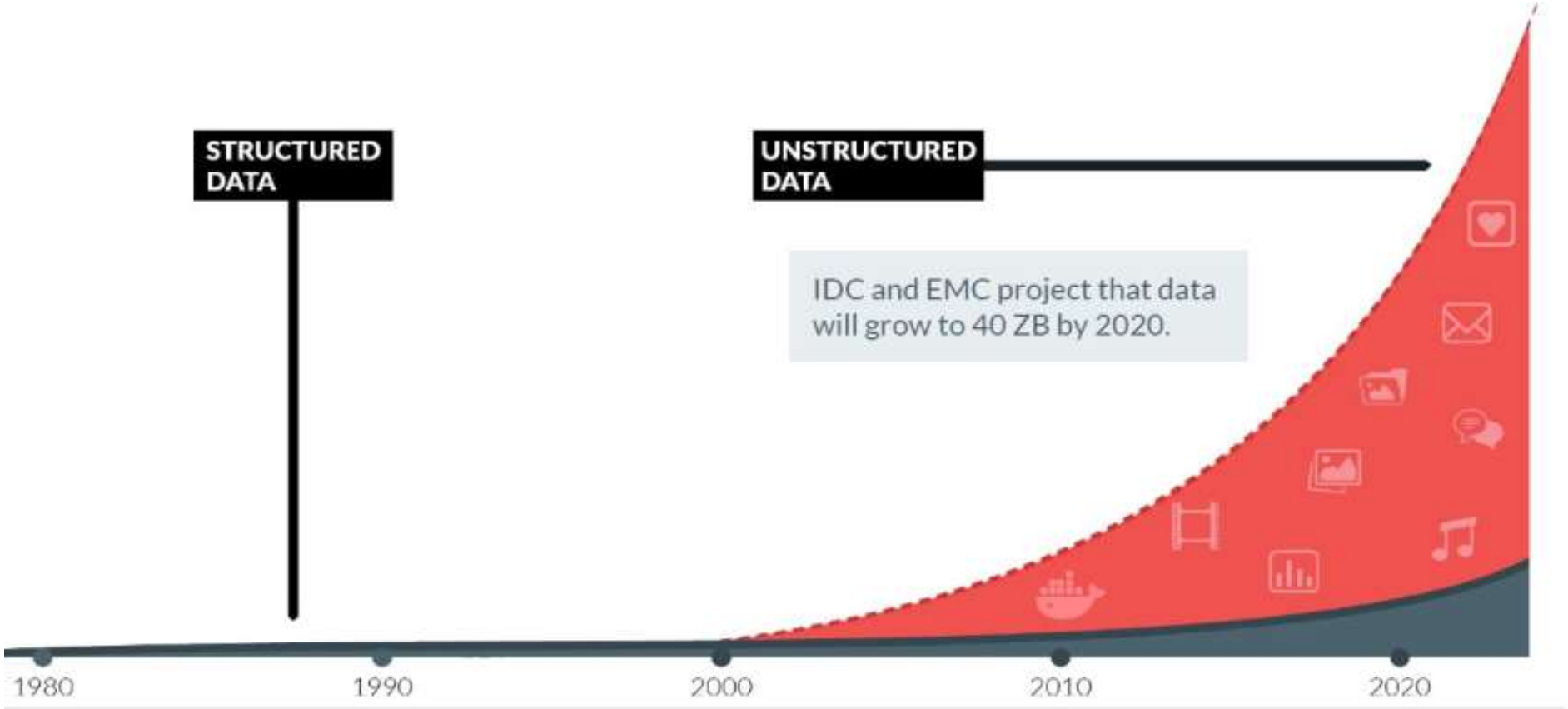




DATA STRUCTURE: EXISTING PERCENTAGES



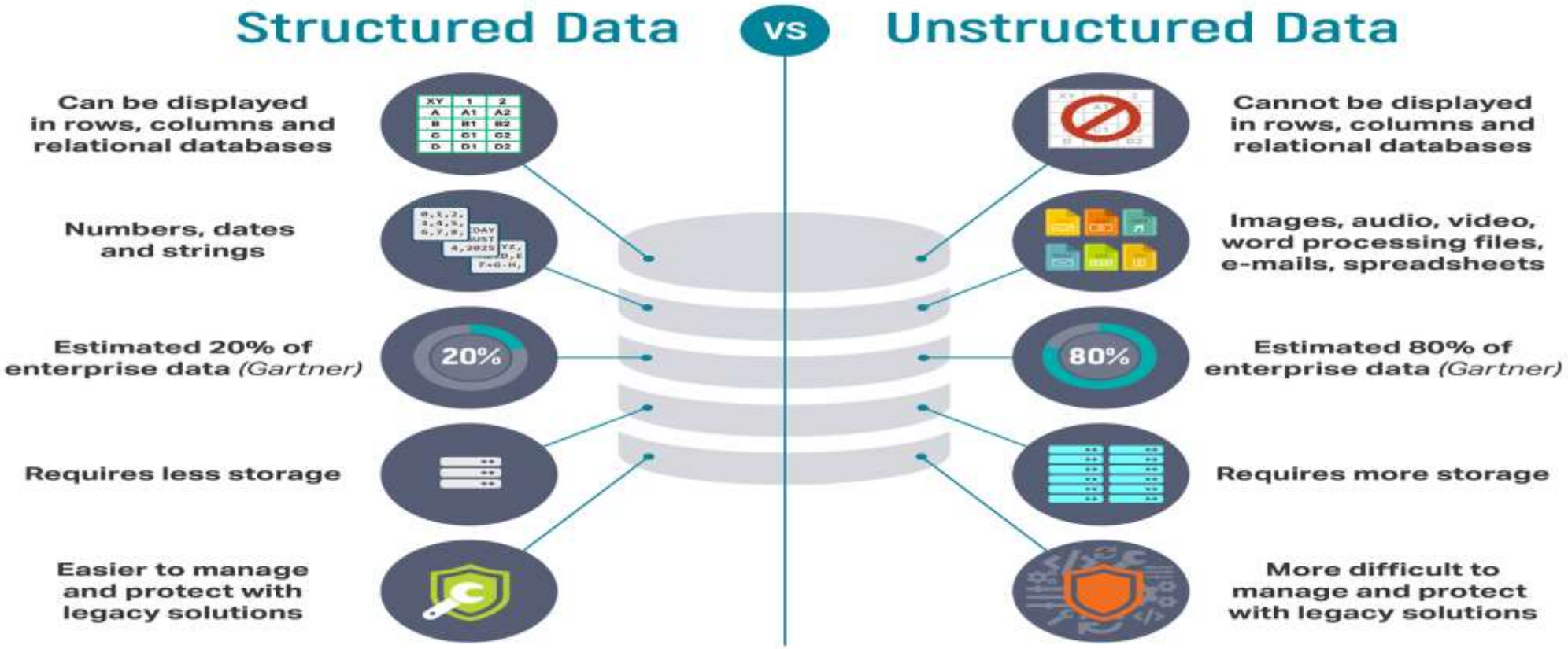
DATA STRUCTURE: EXISTING PERCENTAGES



DATA STRUCTURES WITH EXAMPLES

Structured	Semi-structured	Unstructured
List of people with their phone numbers	Wikipedia pages with links	Text of Encyclopedia Britannica
Temperature in all rooms of a building at every minute for the last 20 years	Collection of scientific papers in JSON format with authors, data of publication, and abstract	File share with corporate documents
Data for age and gender of all people entering the building	Internet pages	Raw video feed from surveillance camera

DATA STRUCTURES WITH COMPARISON EXAMPLE



TYPES OF DATA

- **Record data:**
  - Transaction data
  - Matrix data
  - Sparse matrix data
- **Graph based data**
  - Data with Relationships among Objects
  - Data with Objects That Are Graphs
- **Ordered data**
  - Sequential data (i.e., each data row has an order using timestamps or positions.)
  - Time Series Data (i.e., each data row is a time series.)
  - Spatial Data (i.e. geographical coordinates are orders).

CHARACTERISTICS OF DATA

- **Dimensionality** represents the number of features data has. It can be a low number (i.e., in terms of decades.), as it can be widely large (i.e., thousands.). In case where the number of features is large in a way where data modeling is hard, or impossible due to data complexity, or computational resources we describe the situation by data dimensionality curse.
- **Sparsity** represents the portion of missing values within a dataset. Sparsity may strongly limit the ability to model data, as data values may be so distributed over the set of features (i.e., the model can't capture the relationships between the features and the target data.)

- **Resolution** affects the ability of a model to detect data pattern.
  - (a) if the resolution is high (i.e., number of data points, and or outliers), a model may be unable to recognize data pattern as it can be affected by noise.
  - (b) if the resolution is low, the data pattern may not be present in data.
  - As an example, we can take temperature change of a given city over a year.
    - (a) **case 1 high resolution:** we consider temperature change of every hour, during one year.
    - (b**) case 2 low resolution**: we consider average temperature change of every month.

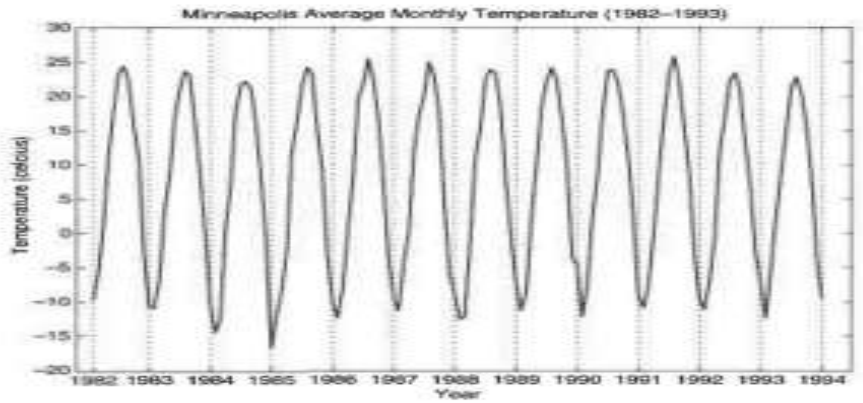
Time	Customer	Items Purchased
t1	C1	A, B
t2	C3	A, C
t2	C1	C, D
t3	C2	A, D
t4	C2	E
t5	C1	A, E

Customer	Time and Items Purchased
C1	(t1: A,B) (t2:C,D) (t5:A,E)
C2	(t3: A, D) (t4: E)
C3	(t2: A, C)

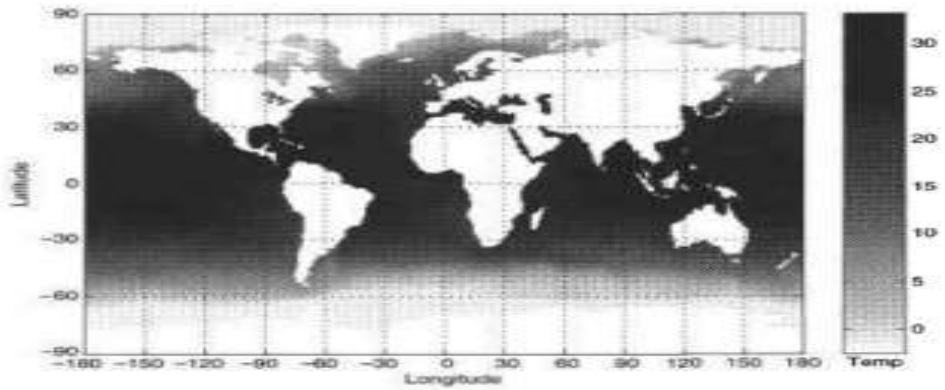
(a) Sequential transaction data.

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```

(b) Genomic sequence data.

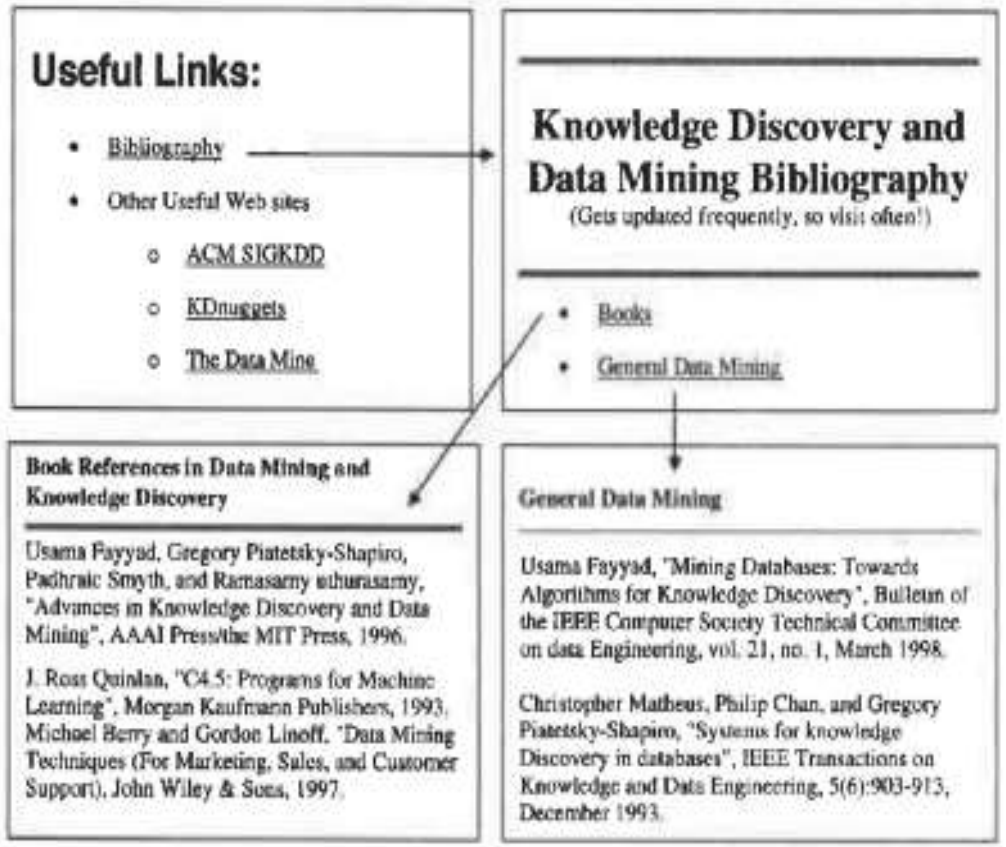


(c) Temperature time series.

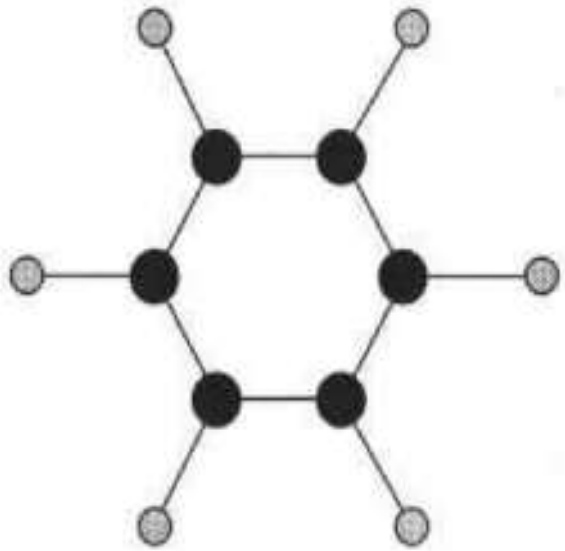


(d) Spatial temperature data.

Sequential dataset



(a) Linked Web pages.



(b) Benzene molecule.

graph dataset

Tid	Refund	Marital Status	Taxable Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

(a) Record data.

TID	ITEMS
1	Bread, Soda, Milk
2	Beer, Bread
3	Beer, Soda, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Soda, Diaper, Milk

(b) Transaction data.

Projection of x Load	Projection of y Load	Distance	Load	Thickness
10.23	5.27	15.22	27	1.2
12.65	6.25	16.22	22	1.1
13.54	7.23	17.34	23	1.2
14.27	8.43	18.45	25	0.9

(c) Data matrix.

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

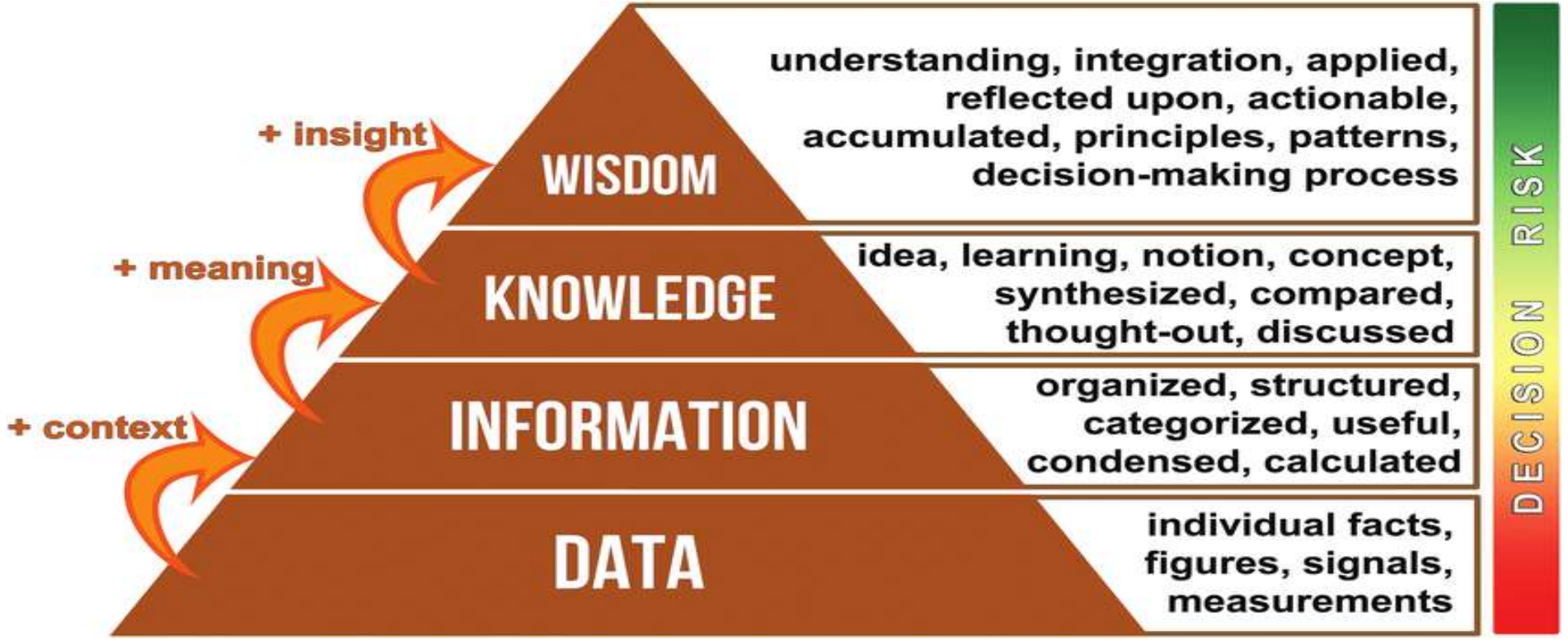
(d) Document-term matrix.

Record dataset

TYPES OF DATA

Attribute Type		Description	Examples	Operations
Categorical (Qualitative)	Nominal	The values of a nominal attribute are just different names; i.e., nominal values provide only enough information to distinguish one object from another. $(=, \neq)$	zip codes, employee ID numbers, eye color, gender	mode, entropy, contingency correlation, χ^2 test
	Ordinal	The values of an ordinal attribute provide enough information to order objects. $(<, >)$	hardness of minerals, $\{good, better, best\}$, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Numeric (Quantitative)	Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. $(+, -)$	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, t and F tests
	Ratio	For ratio variables, both differences and ratios are meaningful. $(*, /)$	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

COMPARISON OF DATA MATURITY



Data types according to usability level

CHALLENGES RELATED TO DATA

- **Scalability** Advances in data generation and collection has led to data sets with huge sizes. Many data mining algorithms employ special search strategies to handle exponential search problems.
  - Scalability can be improved using:
    - sampling
    - or developing parallel and distributed algorithms.

CHALLENGES RELATED TO DATA

- **High Dimensionality** It is now common to encounter data sets wide number of attributes. As an example:
  - in bioinformatics, progress in microarray technology has produced gene expression data involving thousands of features.
  - data set that contains measurements of temperature at various locations taken repeatedly for an extended period generates a huge number of dimensions.

CHALLENGES RELATED TO DATA

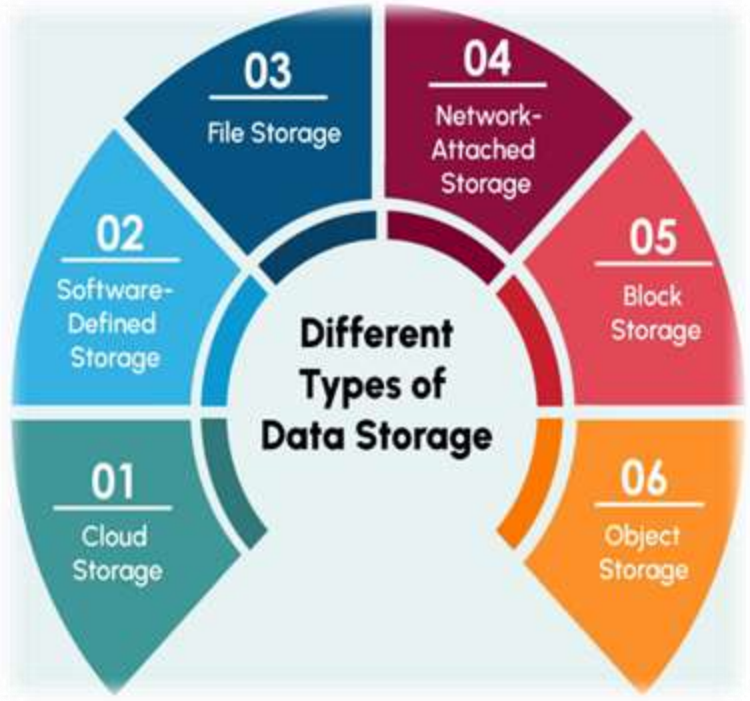
- **Data ownership and Distribution data sources can be distributed/ owned by over/**by more than one location/organization, which requires the development of distributed data mining techniques. This task is related to some challenges:
  - (1) how to reduce the amount of communication needed to perform the distributed computation,
  - (2) how to effectively consolidate the data mining results obtained from multiple sources, and
  - (3) how to address data security issues.

CHALLENGES RELATED TO DATA

- **Non-traditional Analysis** Traditional statistical approach is based on a hypothesize-and-test paradigm, which requires an experiment designed to gather the data, and analyze it. Unfortunately, this process is extremely intensive, where current data analysis tasks often require the generation and evaluation of thousands of hypotheses.
  - Furthermore, the data sets analyzed in data mining are typically not the result of a carefully designed experiment and often represent opportunistic samples of the data, rather than random samples.

DATA COLLECTION AND STORAGE STRATEGIES

- Effective data collection and storage are fundamental to any IT management strategy. Achieving success in data maturity requires a well-planned approach to handling these key elements. This text covers best practices and technologies for efficient data collection and storage.
  - **Data Collection:**
    - **Define Objectives:** Clearly define data collection goals, specifying the type of data needed and how it aligns with organizational goals.
    - **Identify Data Sources:** Determine relevant data sources, whether internal (e.g., sales records, customer interactions) or external (e.g., market data, consumer trends).
    - **Use Collection Tools:** Employ appropriate tools like online forms, IoT sensors, or CRM systems to capture data accurately and efficiently.



DATA COLLECTION TECHNIQUES

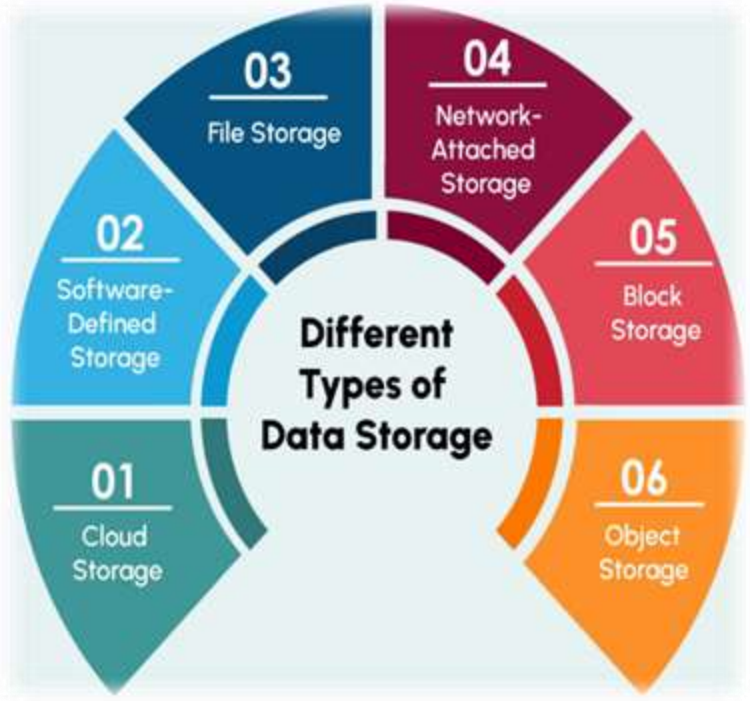
- 1. **Primary Data Collection Methods**
  - **Surveys**: Questionnaires for gathering large-scale quantitative data.
  - **Interviews**: One-on-one conversations for in-depth qualitative insights
  - **Polls**: Short surveys to quickly gauge public opinion.
  - **Observations**: Recording behaviors in natural settings.
  - **Experiments**: Manipulating variables to observe effects.
  - **Focus Groups**: Group discussions for exploring opinions.
  - **Document Review**: Using existing records or documents for research.

DATA COLLECTION TECHNIQUES

- **2. Secondary Data Collection methods:** is information previously gathered and available from both internal and external sources.
  - Health and safety records,
  - Financial statements,
  - Sales reports, and CRM software.
  - External sources include:
    - Government reports, press releases,
    - Business journals, and online resources.

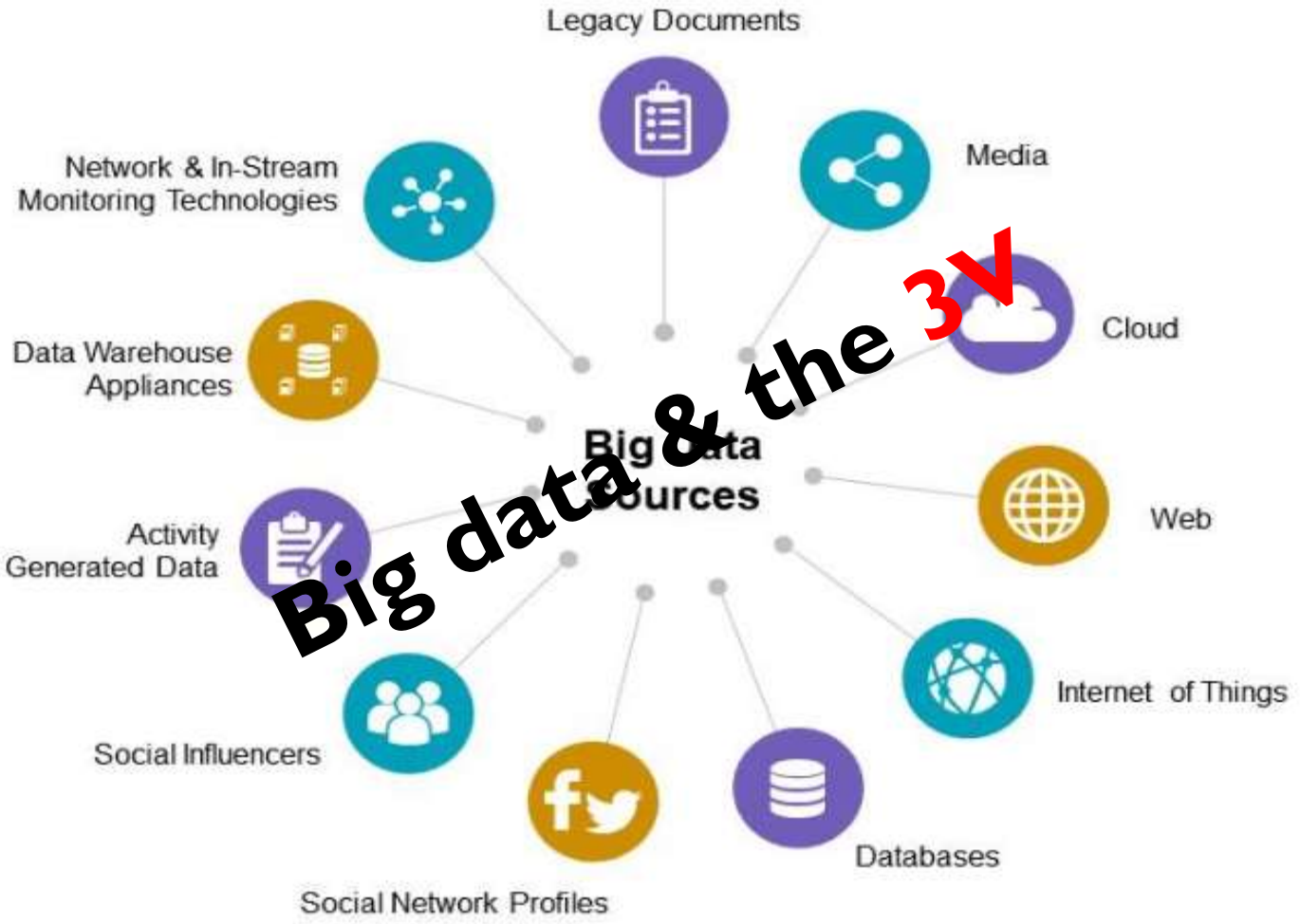
DATA STORAGE

- **Data Storage:**
  - **Storage Options:** Choose between local or cloud storage, weighing factors like cost, security, and scalability. Cloud offers flexibility, while local storage provides more control.
  - **Select Databases:** Opt for relational (SQL) or non-relational (NoSQL) databases based on your data structure, volume, and speed requirements.
  - **Ensure Data Security:** Protect data with encryption, authentication, and regular backups to safeguard against unauthorized access and loss.
  - **Adopt Data Management Practices:** Use methods like categorization and indexing to streamline data retrieval and analysis.



BIG DATA

- Huge Volume of data.
- Complexity of data and Variety of types and structures.
- High Velocity of new data creation and growth.



BIG DATA

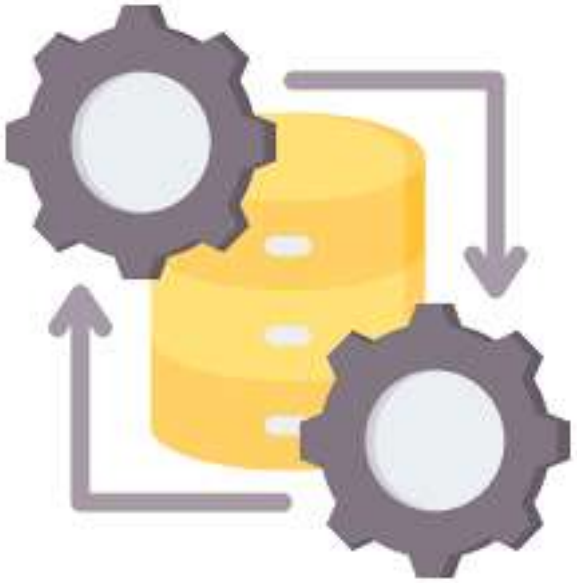
- **Distributed storage architectures:**
  - HADOOP distributed file system (HDFS)
  - Amazon distributed file system (S3)
  - RedHat distributed System (CEPH)

- **No-SQL databases:**
  - Documents (json/xml files as in MongoDB, ElasticSearch)
  - Key/value
  - Graphs

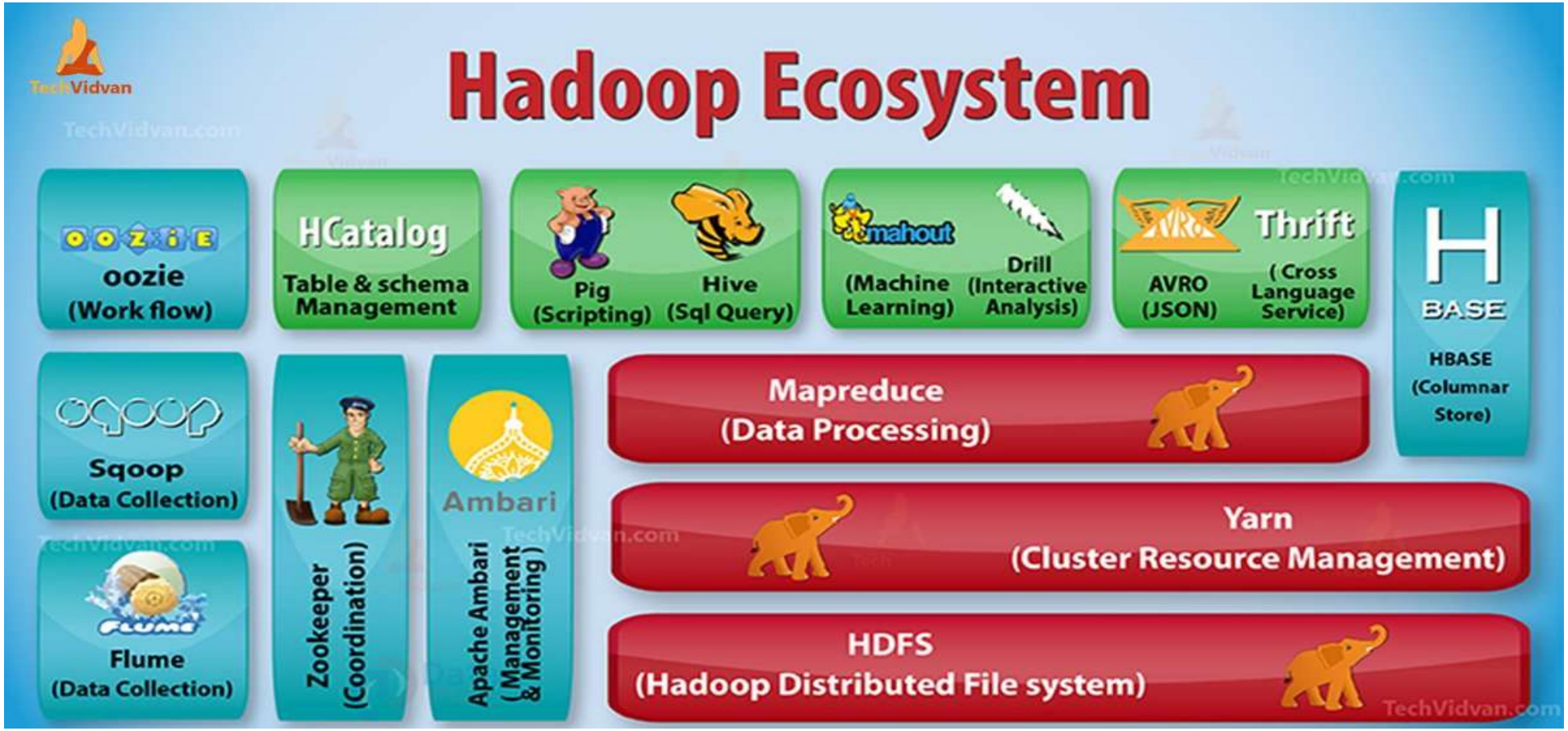
- **Distributed computation algorithms:**
  - Approach **Map-Reduce**
    - Some implementations:
      - HIVE
      - PIG
      - Cascading
      - …etc.
  - Approach Spark
    - Some implementations:
    - Approach based on **GraphX** algorithm
      - Storm (Twitter), Samza (LinkedIn), Spark Streaming

BIG DATA : DATA PROCESSING

- Types of Processing: Batch, Micro-Batch, and Streaming
  - **Batch Processing:** This involves one-time data migrations or periodic transformations on defined datasets. ETL developers use it for bulk processing to support analytics, such as processing a restaurant's nightly orders for financial and HR reporting.
  - **Micro-Batch Processing:** This method processes smaller datasets more frequently, allowing for timely feedback and automated responses without continuous streaming. For instance, a truck transporting potatoes may send GPS data to the data lake every five minutes, alerting the restaurant if the truck breaks down shortly after.
  - **Stream Processing:** This is an always-on flow of data from source to destination, suitable for real-time applications. Examples include customer interactions and sensor data. For a restaurant accepting online orders, a recommendation engine might use event stream processing to suggest items instantly, as a delay would be ineffective.



BIG DATA: TOOLS

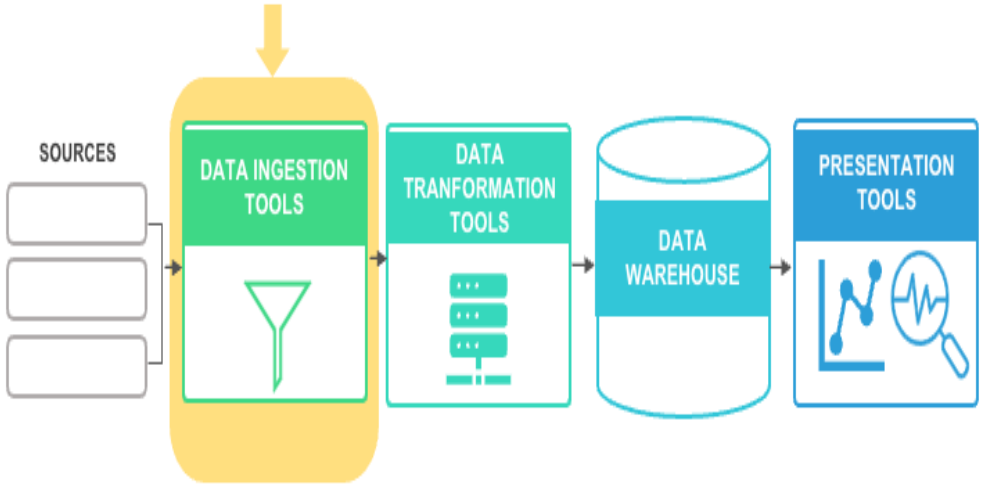


BIG DATA ANALYSIS: OBJECTIVES

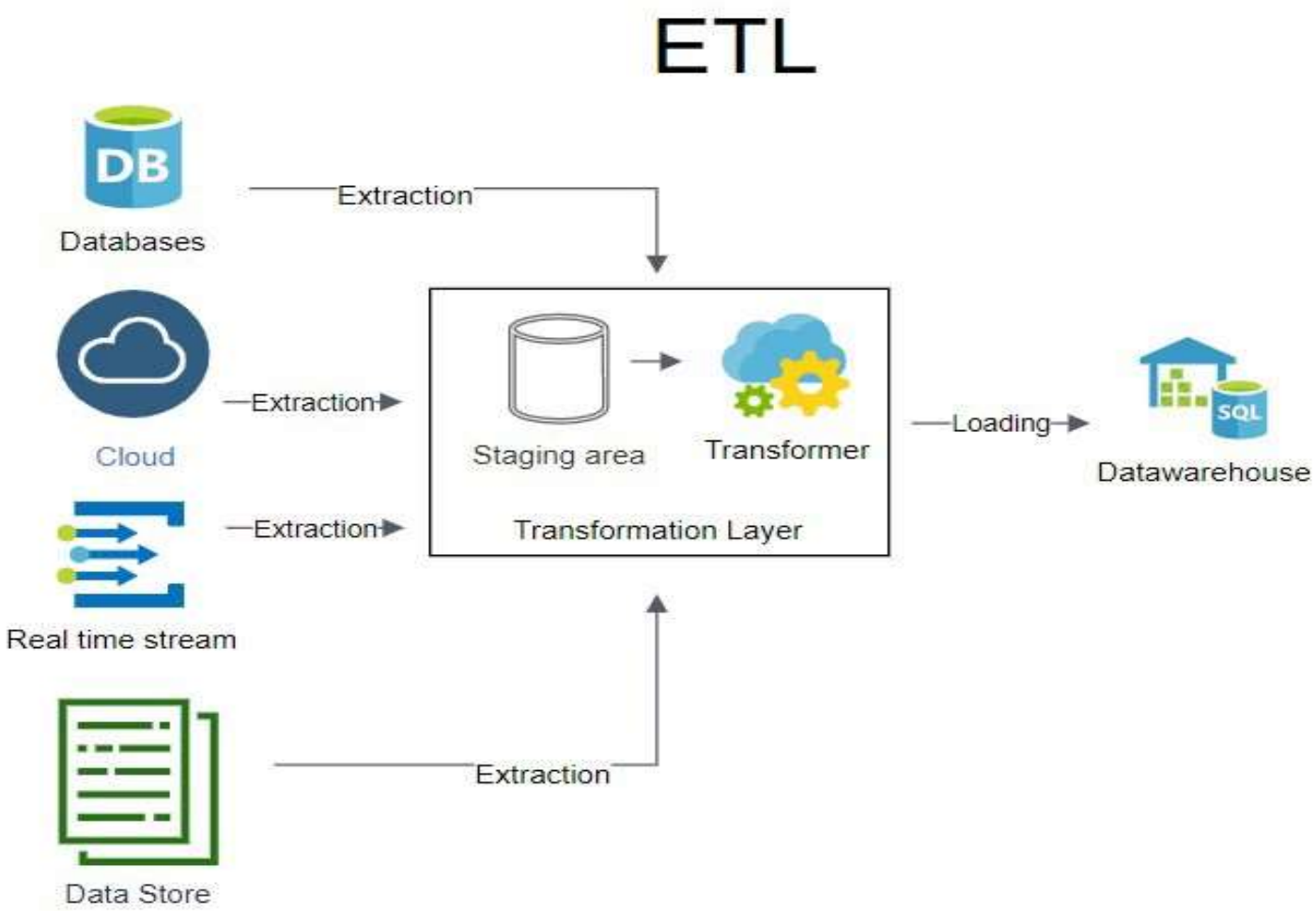
- **Managing large volumes of data**: Data science provides advanced tools and techniques to efficiently handle and analyze vast amounts of data generated from various sources, ensuring that organizations can process and utilize this information effectively.
- **Extracting valuable insights**: By applying statistical analysis and machine learning algorithms, data science enables organizations to uncover hidden patterns and trends within their data, transforming raw information into actionable insights that drive strategic initiatives.
- **Making informed decisions**: With the ability to analyze complex datasets, data science empowers decision-makers to base their choices on empirical evidence rather than intuition, leading to more accurate predictions and improved outcomes for the organization.

DATA INTEGRATION

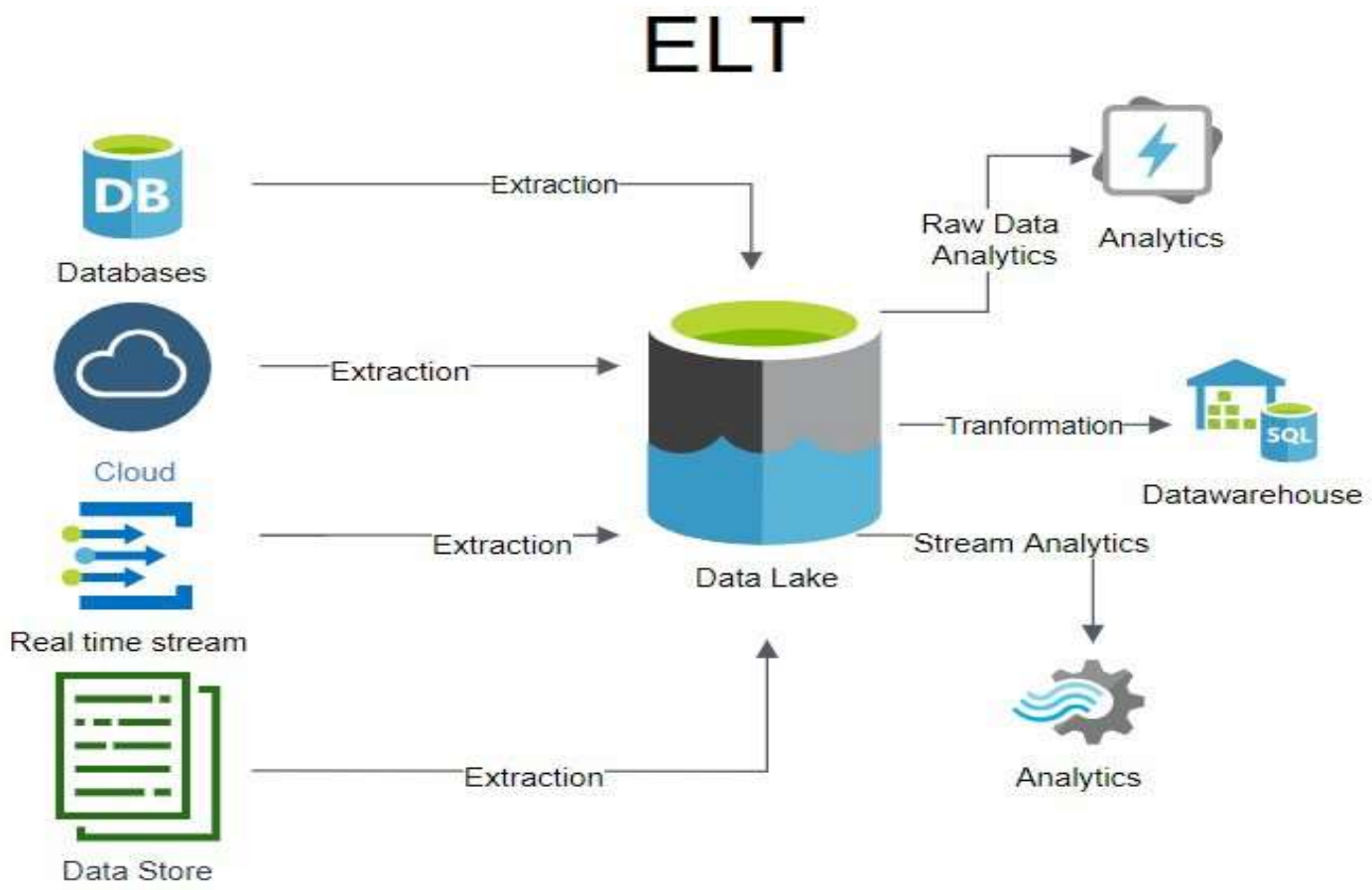
- Data integration techniques are methods used to combine data from multiple sources, in multiple formats into a single, unified view, clean set of data can easily be used to triggering workflows.
- . Common data integration techniques include:
  - **Extract, Transform, Load (ETL):** Tools like Talend and Informatica gather data, transform it, and load it into a data warehouse.
  - **Extract, Load, Transform (ELT):** Solutions like Amazon Redshift and Google BigQuery load data first, then transform it, optimizing for large-scale datasets.
  - **Change Data Capture (CDC):** Apache Nifi monitors and captures data changes in real-time for immediate processing.
  - **Enterprise Application Integration (EAI):** Mulesoft integrates data across various enterprise systems, allowing them to communicate seamlessly.
  - **Data Virtualization:** Denodo and Dremio provide real-time access to data without physically moving it, improving efficiency.
  - **Master Data Management (MDM):** Tools like Informatica MDM ensure consistent, reliable, and unified data across the organization.



DATA INTEGRATION



DATA INTEGRATION

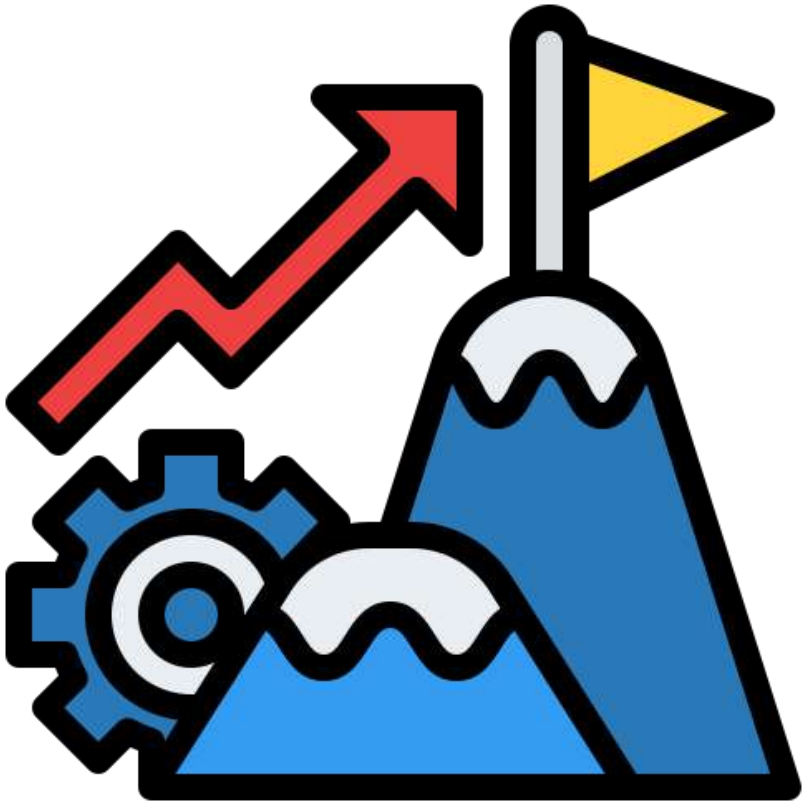


DATA INTEGRATION

- **ETLT** is the best of both **ETL** and **ELT.**
- Data may grow up rapidly during the **ETLT**, in this case big data solutions may be required (i.e., we say Big ETLT.).
- The objective of **ETLT** is to collect, aggregate, and clean data obtained from its source and load it into more exploitable data store manager.
- **ETLT** adds another transformation operation to integrate other data sources and transformed data.
- **A**pplication **P**rogramming **I**nterfaces (APIs) are nowadays popular solutions to access data online (Facebook, Tweeter …etc.).

DATA INTEGRATION: CHALLENGES

- The challenges of data integration include:
  - Data being in many different places and many different formats
  - Inconsistent definitions of data
  - Continuously ensuring data security, quality, privacy and compliance
  - Technical complexity of data engineering requires specialized staff and tools
  - Costs of data integration can quickly escalate

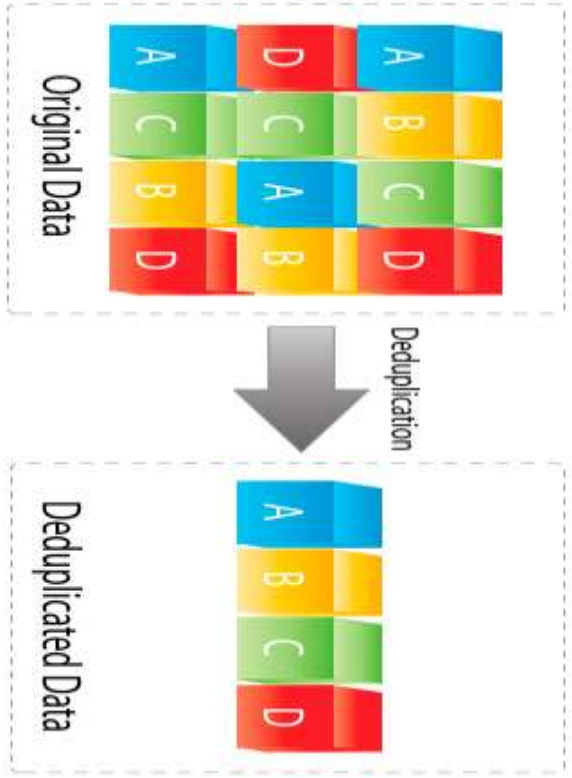


DATA INTEGRATION TECHNIQUES

- **Data integration** in the context of data science refers to the process of combining data from multiple sources into a unified dataset. This involves collecting, cleaning, transforming, and standardizing data to ensure consistency and accuracy.
- **Key objectives of data integration include:**
  - **Data unification:** Creating a single, comprehensive view of data from various sources.
  - **Data consistency:** Ensuring that data is consistent in terms of format, units, and terminology.
  - **Data quality:** Improving data quality by identifying and correcting errors, inconsistencies, or missing values.
  - **Data accessibility:** Making data easily accessible for analysis and decision-making.

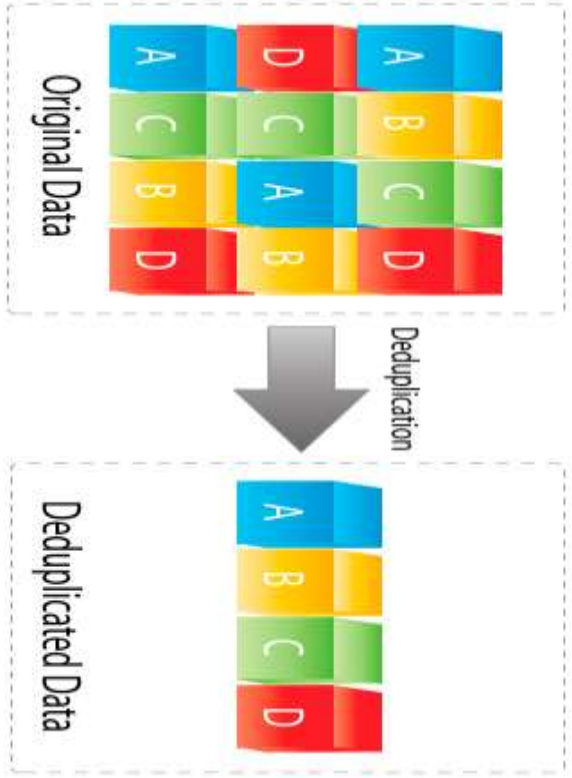
REDUNDANCY DETECTION AND RESOLVING

- **1. Identify Duplicate Records**
  - **Exact Matching:** Use techniques like hashing or string comparison to find records that are identical in all attributes.
  - **Partial Matching:** For records that are similar but not identical, consider fuzzy matching algorithms (e.g., Levenshtein distance, Jaro-Winkler distance) to identify potential duplicates.
- **2. Analyze Attribute Correlations**
  - **Correlation Analysis:** Calculate correlation coefficients between attributes to identify pairs of attributes that are highly correlated. High correlation often indicates redundancy.
  - **Principal Component Analysis (PCA):** Use PCA to reduce the dimensionality of the dataset and identify redundant features.



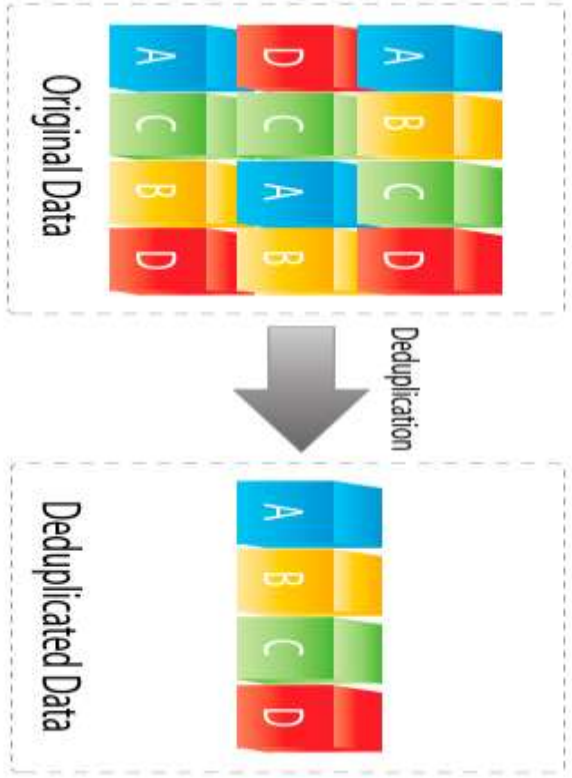
REDUNDANCY DETECTION AND RESOLVING

- **3. Check for Data Quality Issues**
  - **Data Profiling:** Analyze data quality aspects like completeness, accuracy, consistency, and uniformity to identify potential redundancies caused by errors or inconsistencies.
- **4. Review Data Sources and Integration Processes**
  - **Source Analysis:** Examine the sources where the data originated to understand if there are inherent redundancies in the source data.
  - **Integration Review:** Assess the data integration processes to identify if there are steps that might introduce redundancy.



REDUNDANCY DETECTION AND RESOLVING

- **Data Deduplication:** Remove duplicate records entirely, ensuring that you keep only one instance of each unique record.
- **Data Consolidation:** Combine redundant data into a single, consistent representation. This might involve merging attributes or creating new attributes.
- **Data Standardization**: Apply consistent formatting, units, and terminology to data elements to reduce redundancy caused by inconsistencies.
- **Data Normalization**: Organize data into tables that are related by common fields, reducing redundancy and improving data integrity.



REDUNDANCY DETECTION AND RESOLVING : EXAMPLE

- **Attributes redundancy:**
  - An attribute is redundant when it can be conducted from another attribute or a combination of sub attributes.
  - Using the X2 correlation test for nominal values:

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}},$$

  - Correlation coefficient or covariance for numerical values:

$$r_{A,B} = \frac{\sum_{i=1}^{m}(a_i - \overline{A})(b_i - \overline{B})}{m\sigma_A\sigma_B} = \frac{\sum_{i=1}^{m}(a_i b_i) - m\overline{A}\,\overline{B}}{m\sigma_A\sigma_B},$$

$$Cov(A, B) = E((A - \overline{A})(B - \overline{B})) = \frac{\sum_{i=1}^{m}(a_i - \overline{A})(b_i - \overline{B})}{m}.$$

REDUNDANCY DETECTION AND RESOLVING : EXAMPLE

- **Instance redundancy:**
  - It may appear because of merging data from different sources, errors in indexing the instances or using deformalized data tables. It may lead to overspecialization of model training.
  - Instance redundancy detection is mostly amounted using distance-based techniques such as:
    - The edit distance
    - Jaro distance:

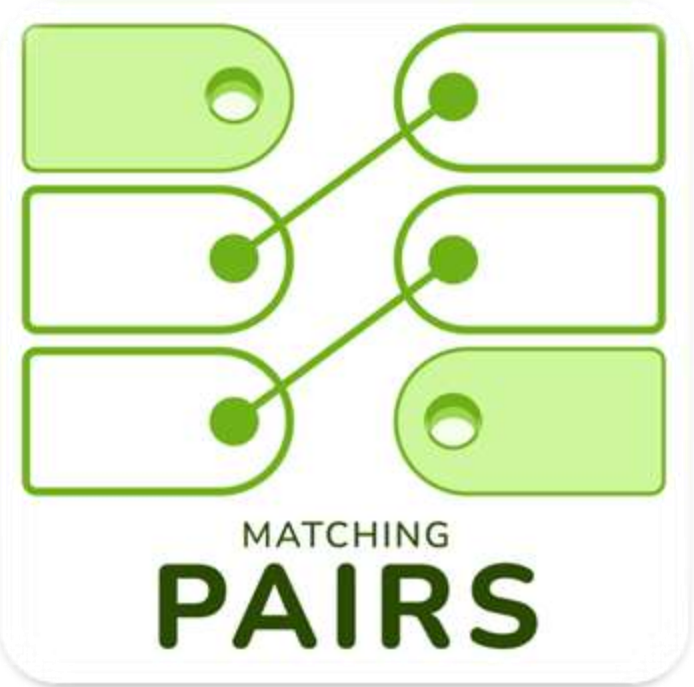
$$Jaro(\sigma_1, \sigma_2) = \frac{1}{3}\left(\frac{c}{|\sigma_1|} + \frac{c}{|\sigma_2|} + \frac{c - t/2}{c}\right)$$

  - Or probabilistic techniques/ machine learning techniques /clustering techniques.

➢ Main difference between the techniques is in the ability of detecting the redundancy when changing the order, containing spaces and so on …

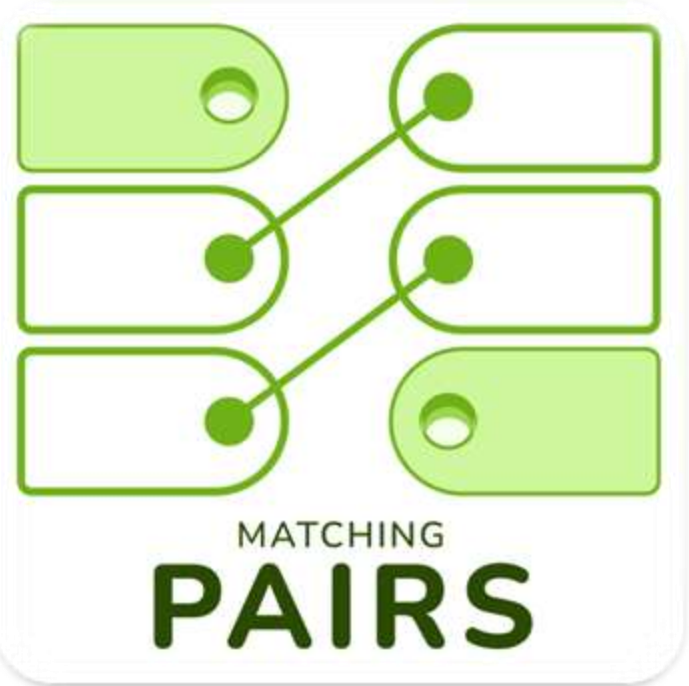
SCHEME MATCHING

- **1. Manual Matching:**
  - **Expert Knowledge:** Rely on domain experts to identify semantic equivalences between attributes in different schemas.
  - **Thesaurus or Ontology:** Use thesauri or ontologies to map terms from different schemas to common concepts.
- **2. Automated Matching:**
  - **String Similarity:** Compare strings based on character sequences, edit distance, or soundex algorithms.
  - **Statistical Methods:** Use statistical techniques like Jaccard similarity or cosine similarity to measure the similarity between sets of terms or attributes.
  - **Machine Learning:** Employ machine learning algorithms (e.g., supervised learning, unsupervised learning) to learn patterns in data and automatically identify semantic equivalences.
  - **Semantic Web Technologies:** Leverage technologies like RDF, OWL, and SPARQL to represent and reason about the semantic relationships between data elements.



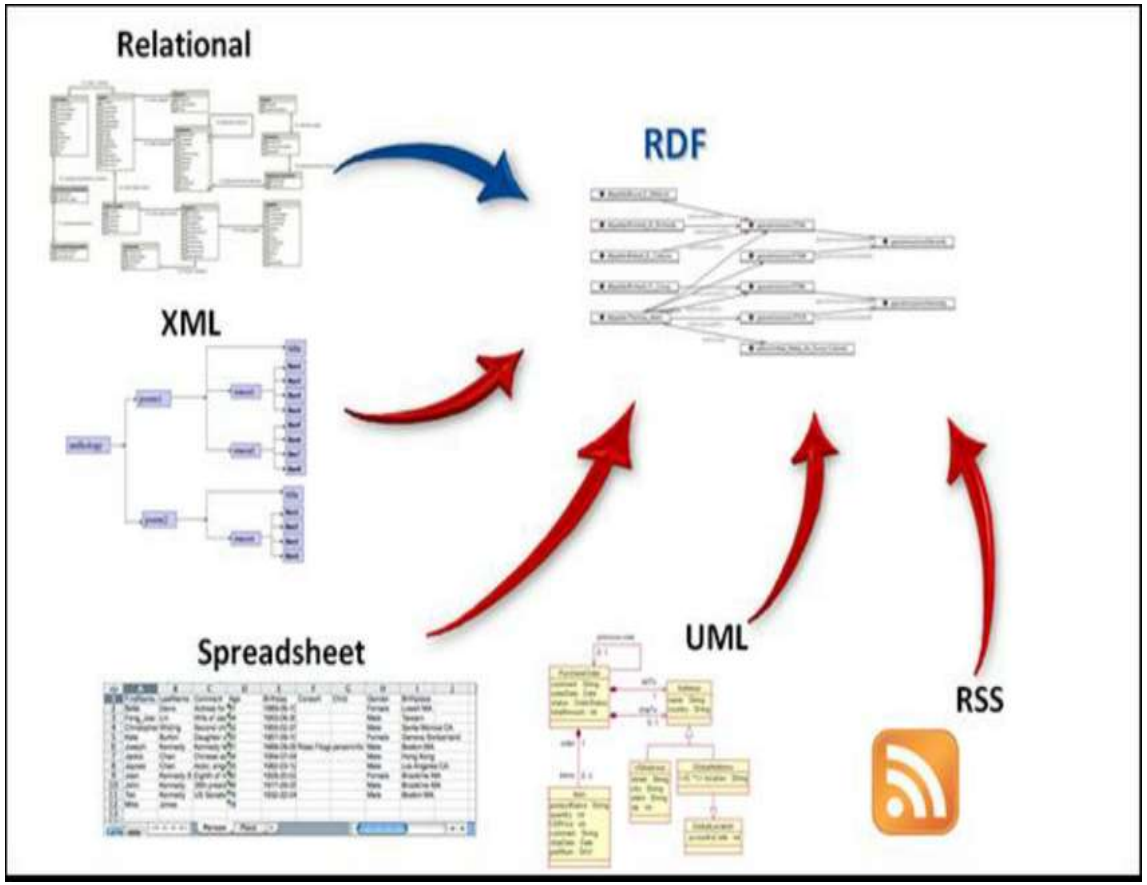
SCHEME MATCHING

- 3. Hybrid Approaches:
  - **Combination of Methods**: Combine manual and automated techniques to leverage the strengths of both.
  - **Iterative Refinement:** Start with automated matching and refine the results using manual review and feedback.
- 4. Considerations for Scheme Matching:
  - **Contextual Information**: Consider the context in which terms are used to improve matching accuracy.
  - **Data Quality:** Ensure that the data is clean and consistent to avoid errors in matching.
  - **Matching Threshold**: Set appropriate thresholds for similarity measures to balance precision and recall.
  - **Evaluation Metrics**: Use metrics like precision, recall, and F1-score to evaluate the quality of the matching results.



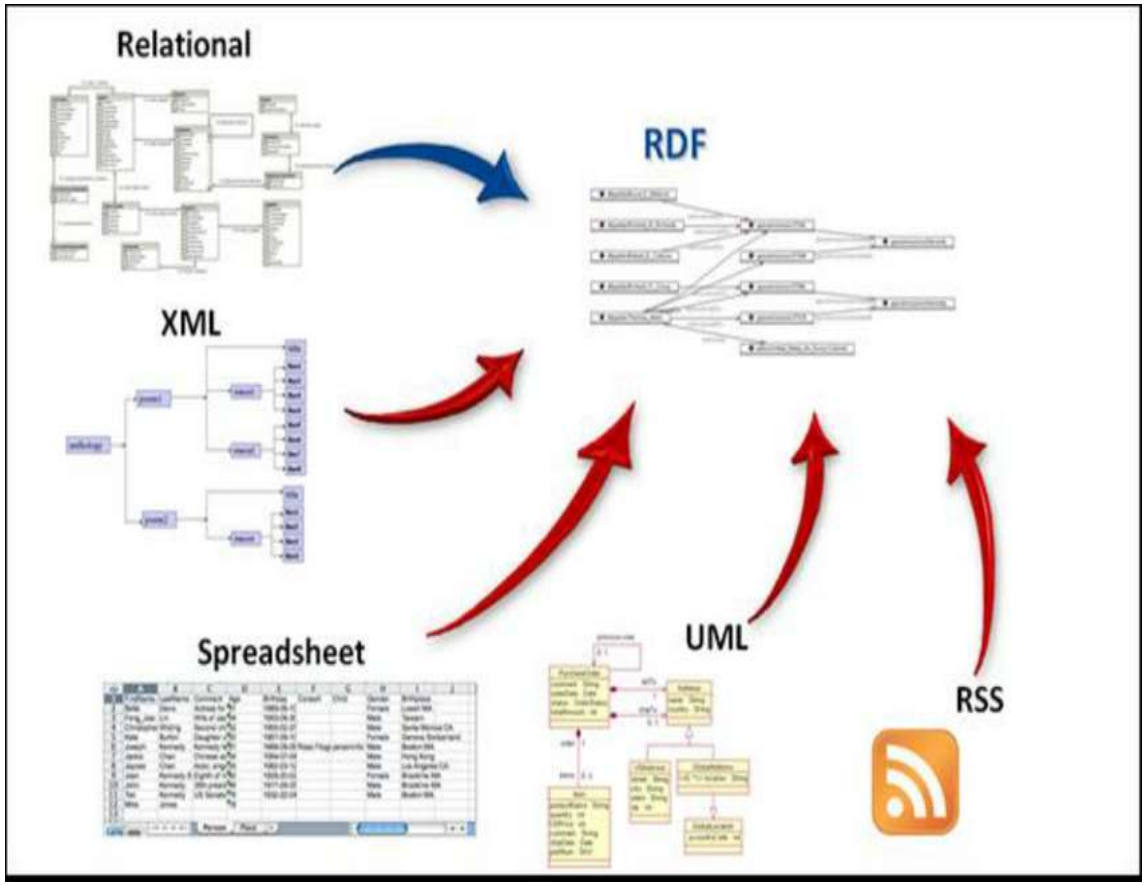
SEMANTIC INTEGRATION

- **1. Ontology-Based Approaches:**
  - **Ontology Development:** Create ontologies that define the concepts, properties, and relationships relevant to the domain of interest.
  - **Ontology Mapping:** Align concepts and relationships from different ontologies to establish correspondences.
  - **Reasoning:** Use reasoning techniques to infer new relationships and draw conclusions based on the ontology.
- **2. Semantic Similarity Measures:**
  - **Concept-Based Similarity:** Measure the similarity between concepts based on their definitions, properties, and relationships.
  - **Instance-Based Similarity:** Compare instances of concepts to identify similarities and patterns.



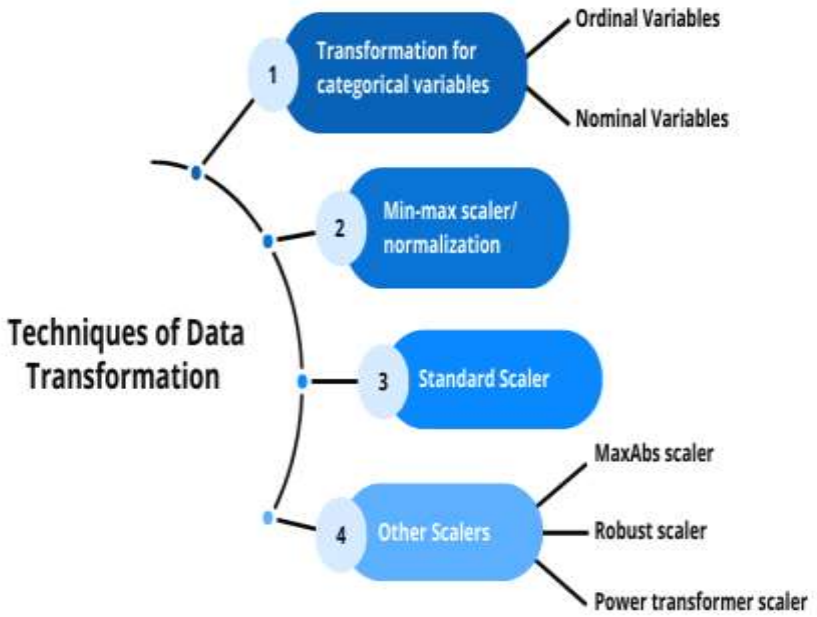
SEMANTIC INTEGRATION

- **Semantic integration** goes beyond simple scheme matching by focusing on the underlying meaning or semantics of data elements. It aims to establish correspondences between concepts and relationships across different schemas, ensuring that data from diverse sources can be meaningfully combined and analyzed.
- **3. Natural Language Processing (NLP):**
  - **Text Mining:** Extract semantic information from textual descriptions of data elements.
  - **Named Entity Recognition:** Identify named entities (e.g., people, organizations, places) and their types.
  - **Sentiment Analysis:** Determine the sentiment associated with text data.



VALUES ALIGNMENT

- **Value alignment** in the context of data integration refers to the process of ensuring that data values from different sources are consistent and comparable. This involves:
  - **Data standardization**: Applying consistent formats, units, and terminology to data elements.
  - **Data normalization**: Transforming data into a standard format or structure.
  - **Data cleaning**: Identifying and correcting errors, inconsistencies, or missing values.
  - **Data conversion**: Converting data from one format or encoding to another.
  - **Data enrichment**: Adding missing or complementary information to data.



EXAMPLE OF DATA INTEGRATION USING AN ETL SYSTEM

- **Extraction:**
  - **Point-of-Sale (POS) Systems:** Extract transaction data, including customer IDs, product information, and purchase amounts.
  - **Loyalty Programs:** Extract customer loyalty program data, such as membership numbers, points balances, and redemption history.
  - **CRM Systems:** Extract customer contact information, demographics, and preferences.
  - **Online Store:** Extract online purchase data, including customer behavior, browsing history, and abandoned cart information.

- **Transformation:**
  - **Data Cleaning:** Cleanse data to remove inconsistencies, errors, or duplicates.
  - **Data Standardization:** Standardize customer IDs, product codes, and other data elements to ensure consistency across sources.
  - **Data Enrichment:** Enhance customer data by adding demographic information or preferences from external sources.
  - **Data Aggregation:** Combine data from different sources to create a comprehensive customer profile.

- **Loading:**
  - **Data Warehouse:** Load the transformed data into a data warehouse for analysis and reporting.
  - **Data Mart:** Create targeted data marts for specific use cases, such as customer segmentation or marketing campaign analysis.

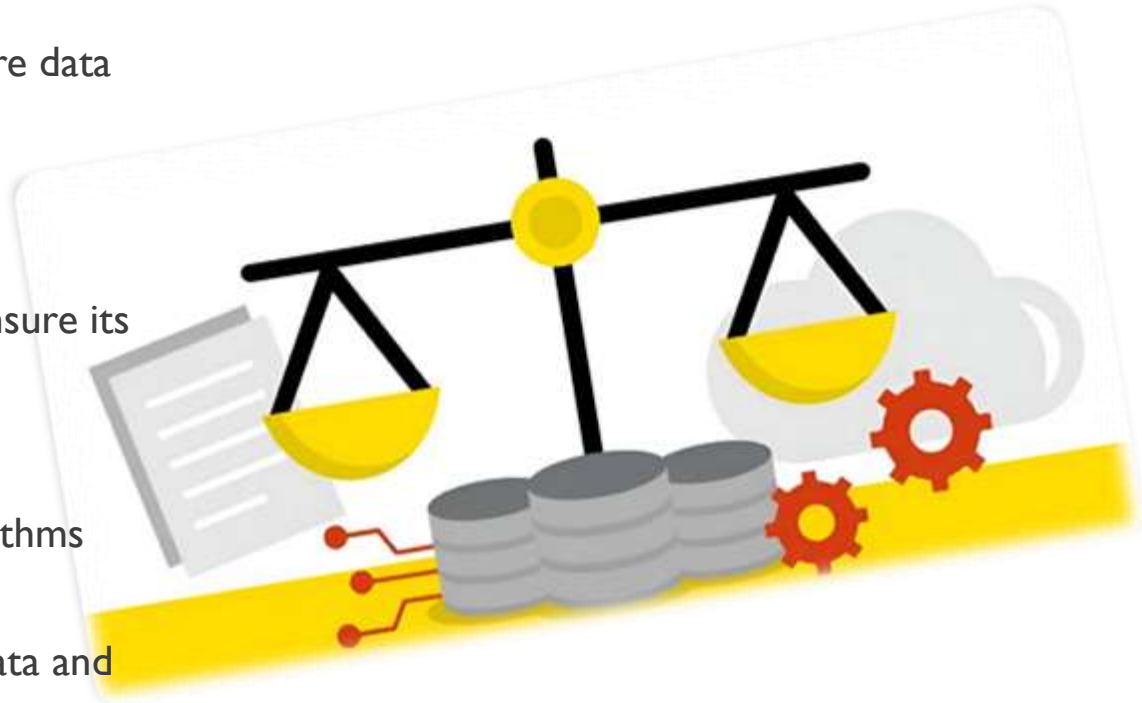
- **Benefits:**
  - **Personalized Marketing:** Tailor marketing campaigns based on customer preferences and purchase history.
  - **Customer Segmentation:** Identify customer segments with similar characteristics to target marketing efforts effectively.
  - **Inventory Management:** Optimize inventory levels based on demand patterns and sales trends.

ETHICAL AND PRIVACY CONCERNS

- **1. Data Privacy:**
  - **Consent:** Obtain explicit consent from individuals before collecting and using their personal data.
  - **Data Minimization:** Collect only the necessary data to achieve the intended purpose.
  - **Data Security:** Implement robust security measures to protect data from unauthorized access, disclosure, alteration, or destruction.
  - **Data Retention:** Establish clear policies for data retention and deletion to prevent unnecessary storage of personal information.

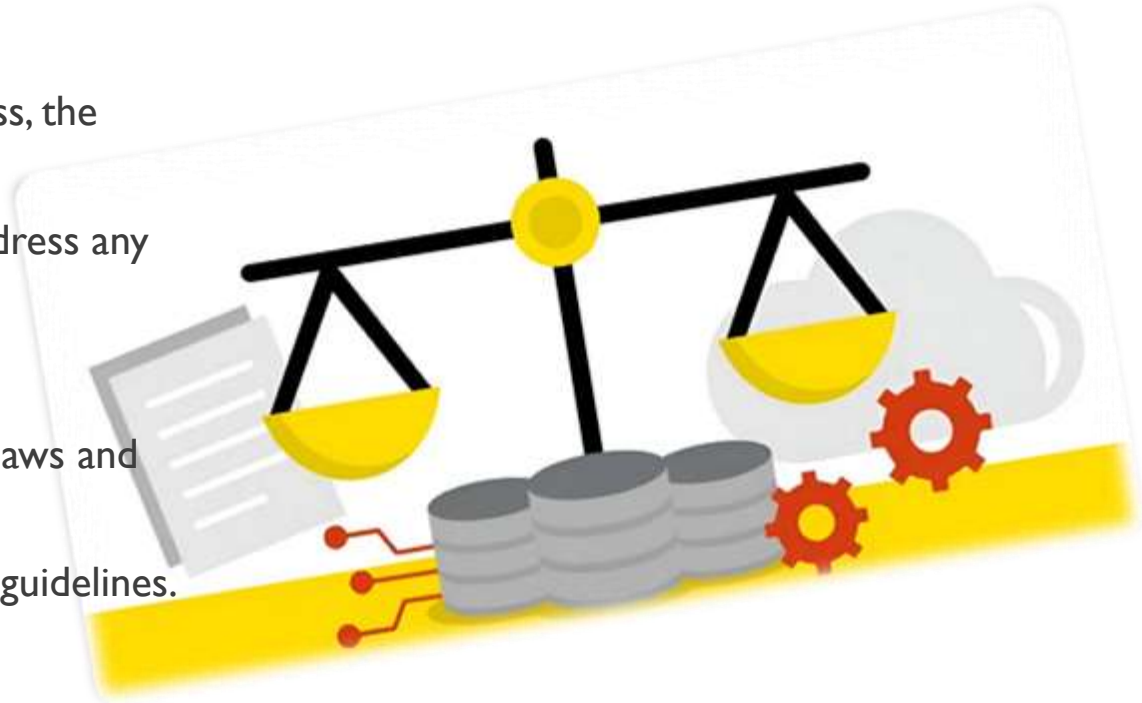
ETHICAL AND PRIVACY CONCERNS

2. **Data Quality and Accuracy:**
   - **Data Governance:** Establish data governance practices to ensure data quality and accuracy throughout the integration process.
   - **Data Validation:** Validate data to identify and correct errors or inconsistencies.
   - **Data Lineage:** Track the origin and transformation of data to ensure its integrity.
3. **Bias and Fairness:**
   - **Bias Detection:** Identify and address biases in the data or algorithms used for integration.
   - **Fairness Assessment:** Evaluate the fairness of the integrated data and ensure it does not perpetuate discrimination or inequality.



ETHICAL AND PRIVACY CONCERNS

4. **Transparency and Accountability:**
   - **Transparency:** Be transparent about the data integration process, the sources of data, and how it is used.
   - **Accountability:** Establish mechanisms for accountability and address any privacy or ethical concerns that arise.
5. **Regulatory Compliance:**
   - **Data Protection Laws:** Comply with relevant data protection laws and regulations (e.g., GDPR, CCPA).
   - **Industry Standards:** Adhere to industry-specific standards and guidelines.



ETHICAL AND PRIVACY CONCERNS

- **Data generalization** is performed by replacing some value by another value **more general or less precise** for privacy purposes. It's known also by **blurring**. Classical examples of this transformation consists of binning methods such as assigning a specific value to an interval (e.g., age 25 [18-30[.).
  - **Automated generalization:** an algorithm that distorts values till we get K similar individuals (e.g., k individuals belonging to the same interval).
    - ➢ Check the definition of K-Anonymity
  - **Declarative generalization:** in this case, data ranges are fixed in upfront, so the data scientist decides which generalization level is enough to preserve privacy. In ages examples, we can replace a full date by year and month or a decade.
    - ➢ Check also, data masking

```
Input:
    Dataset D, quasi-identifiers QI, anonymity level k

Output:
    Anonymized dataset D'

1. Generalize quasi-identifiers QI to broader categories.

2. Group records in D by generalized QI values.

3. For each group C:
     - If |C| < k, generalize further or suppress sensitive data.

4. Return anonymized dataset D' with all groups having at least k records.

End.
```

REFERENCES

- I strongly recommend this series of articles:
  - *Data Preprocessing in Data Mining : Salvador García • Julián Luengo Francisco Herrera*