Alma Mater Studiorum University of Bologna

Artificial Intelligence

Machine Learning and Data Mining (MLDM)

Course Notes

Author: Simone Reale

Academic Year 2023/2024

Contents

Introduction and main concepts

This course is organised into two big topics which we will go deeper on later. These topics are:

- **Data Mining**, in which we will focus on the *data* side, studying the enabling technologies which have been developed for other purposes, but can positively influence the success of data mining processes;
- **Machine Learning**, in which we will focus on the techniques that support *data–driven decisions*, including learning models and algorithms which allow to extract actionable patterns from data.

First of all we have to say that *Data*, *Data Mining* and *Machine Learning* have many differences. While the last two techniques share different concepts, Data exist independently from them, but they need Machine Learning and Data Mining to infer interesting and actionable insights. Especially in the last years these techniques have acquired much more importance with the diffusion of Big Data.

Now we focus on some concepts that represent the basis of all the things that will come after:

Data: a collection of raw value elements;

Information: the result of collecting, interpreting and organising data (relationships between items, context and meaning)

Knowledge: putting together information in order to recognise patterns, according to the needs of the system;

Insight: all the things that we can infer from knowledge on the basis of what is our goal.

When an event in the *real world* changes the state of the enterprise, one of the events below happens:

- a *transaction* is executed, i.e. a business event that changes or modifies data stored in an information system (*database*);
- a *signal* is collected by the infrastructure and stored, which is the reading of a measure provided by a sensor.

The concept of **business** in this case is referred to a **business process**, i.e. a set of activities that will achieve an *organization goal*, once completed. Outside these two events data can also come from *external subjects*.

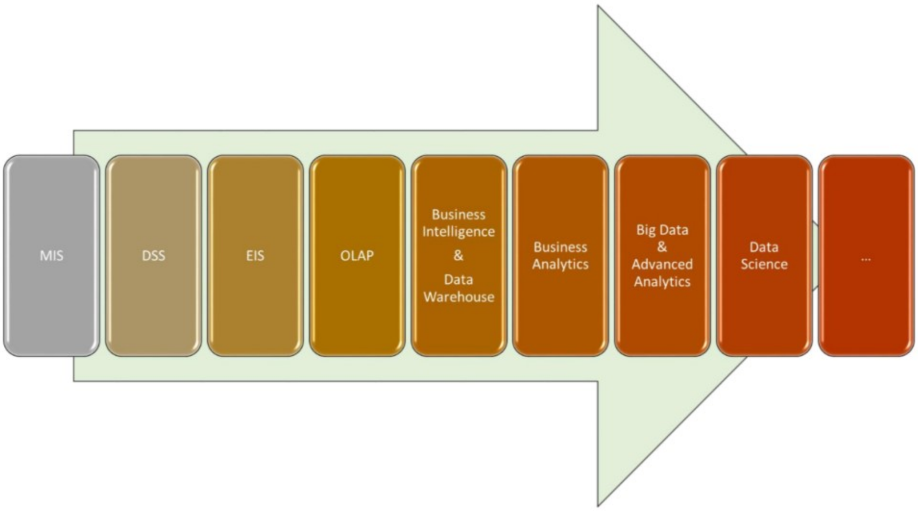

MIS
DSS
EIS
OLAP
Business Intelligence & Data Warehouse
Business Analytics
Big Data & Advanced Analytics
Data Science
...

Figure 1.1: Increasing insights

OLTP (On-Line Transaction Processing) It is a class of software programs capable of supporting transaction-oriented applications and data storage. It is designed to record the daily routine transactions necessary to run the business.

ERP (Enterprise Resource Planning) It is a software system that integrates and manages key business processes of all departments within a single software product into a single, unified platform. It operates in or near real time and provide a common database, which supports all the applications.

MIS (Management Information Systems) They are standardized reporting systems built on existing OLTP, which work by gathering, processing, managing and delivering important information to support decision-making and management activities within an organization. It is used in working envirnments to generate performance indicators of any type.

DSS (Decision Support Systems) They are interactive and computerized information systems that aid individuals and organizations in decision-making by providing data analysis, modeling, and decision-making tools to support a wide range of complex and unstructured decisions.

EIS (Executive Information Systems) They are computer-based information systems tailored for senior executives, offering a user-friendly interface that provides consolidated and summarized information from various data sources to support strategic decision-making and organizational performance monitoring.

OLAP (On-Line Analytical Processing) It is a computer-based approach that allows users to interact with and analyze data from multiple dimensions, providing a flexible and intuitive way to explore, query, and report on large datasets. It uses algorithms and data structures specifically designed to ease operations like selections, projections, column exchanges, etc.

BI (Business Intelligence) It is a set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information used to enable more effective strategic, tactical, and operational insights and decision-making (Forrester Research).

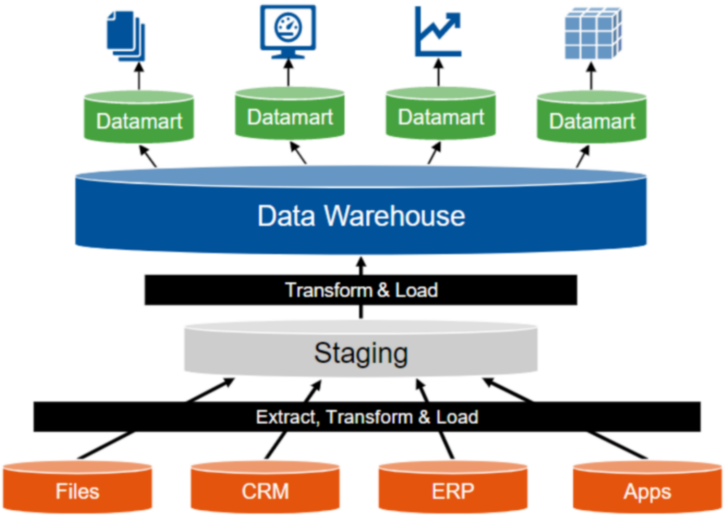


Figure 1.2: BI architecture

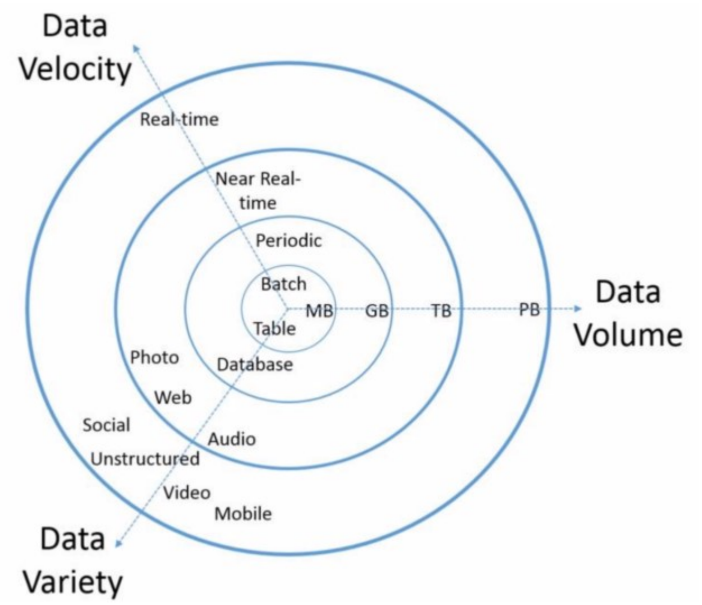
Analytics Structured decisions driven by data and there are different types:

- **descriptive** - understand data
- **diagnostic** - understand causes
- **predictive** - calculate the most probable value of a variable in a future time, given the history of a set (sequence) of variables
- **prescriptive** - suggest actions to be taken to obtain the desired effect.

Cloud Computing Cloud computing refers to the delivery of computing services, including storage, processing power, and applications, over the internet.

It is typically categorized into three main service models:

- **Software as a Service (SaaS):** Delivers software applications over the internet on a subscription basis. Users can access and use software applications running on providers' cloud infrastructure.
- **Platform as a Service (PaaS):** Offers a platform that allows users to develop, run, and manage applications without dealing with the complexities of infrastructure. Then applications are deployed on the provider's cloud infrastructure.
- **Infrastructure as a Service (IaaS):** Consumer can use computing resources within provider's infrastructure upon which they can deploy and run arbitrary software, including OS and applications.



Big Data Big Data are a collection of data sets so large and/or complex and/or fast changing that they are difficult to process using traditional DBMSs or traditional data processing applications.

Their taxonomy is divided into data that are:

- **Structured:** relational tables, spreadsheet (or data which could easily fit in them);
- **Unstructured:** does not have an associated or pre-defined data model (video, audio, pictures, etc.);
- **Semi-structured:** there is some structure, perhaps data refer to different structures (XML, JSON, etc.).

Data Mining

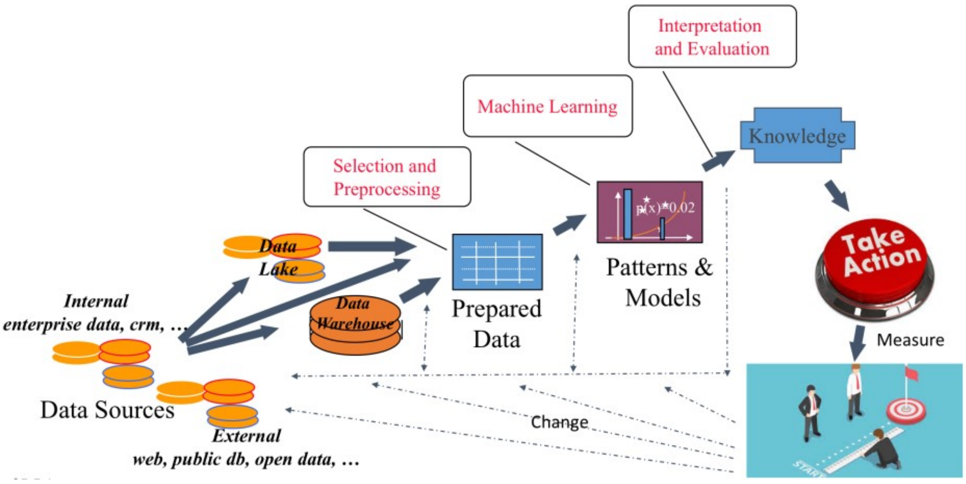


Figure 2.1: The Data Mining Process

Data Mining is the process of discovering patterns, trends, correlations, or meaningful information from large datasets. It involves the application of various techniques from *statistics*, *machine learning*, and *database systems* to extract valuable knowledge from raw data.

Business Intelligence and Data Warehouses

Business Intelligence (BI)

Business Intelligence (BI) represents a key concept in the field of Data Mining and can be described as the process of:

- transforming raw data into useful information to support effective and aware business strategies
- capturing the business data and getting the right information to the right *people*, at the right *time*, through the right *channel*.

There are different definition that has been provided during the years, but there are two of them in particular that we can highlight:

> *Business intelligence (BI) is an umbrella term that includes the applications, infrastructure and tools, and best practices that enable access to and analysis of information to improve and optimize decisions and performance.* - ***Gartner***

> *Business Intelligence is a set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information used to enable more effective strategic, tactical, and operational insights and decision-making.* - ***Forrester Research***

Data Warehouse (DWH) One of the main tools to support BI is the ***Data Warehouse (DWH)***, which is a type of *Decision Support System (DSS)* and can be seen informally as an optimized repository that stores information for the decision-making process. With the huge and increasing number of information that companies have to manage in order to find relevant business strategies DWHs answer to the necessity of more sophisticated solutions than classical operational databases.
The main advantages are the following:

- they provide the ability to manage sets of historical data;
- they provide the ability to run multidimensional analysis accurately and rapidly;
- they are based on a simple model that can be easily learned by its users;
- they are the basis for indicator-calculating systems.

More formally, we can say that a ***Data Warehouse (DWH)*** is a specialized database that stores large volumes of historical data and facilitates the analysis and reporting of that data to support *decision-making processes.* It provides several key features:

- **Subject-Oriented:** A data warehouse is designed to focus on specific subjects or domains relevant to the enterprise's operations, such as customers, products, sales, etc. By organizing data around these core concepts, it enables analysts and decision-makers to gain insights into various aspects of the business.
- **Integration and Consistency:** One of the primary functions of a DWH is to integrate data from multiple disparate sources, including transactional databases, spreadsheets, and external systems. This integration ensures that data from different sources is harmonized and provides a unified view across the organization. Consistency in data representation and structure is maintained to ensure accuracy and reliability in analysis.
- **Evolution Over Time and Non-Volatility:** A crucial aspect of a data warehouse is its ability to capture and track changes in data over time. Historical data are preserved, allowing for the analysis of trends and patterns spanning various time periods. Unlike transactional databases where data may be constantly updated or overwritten, data in a data warehouse are non-volatile. Once committed, the data remains static, read-only, and preserved for future reporting and analysis.

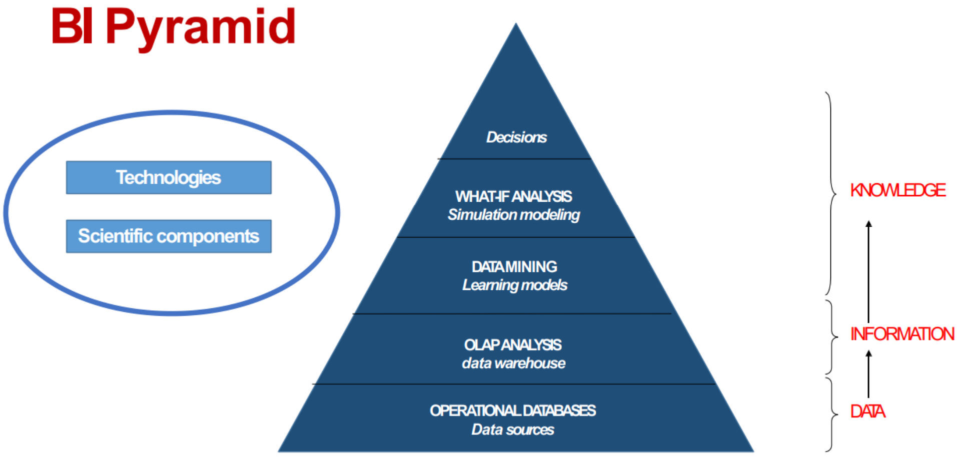
In summary, a ***DWH*** serves as a centralized repository for historical data, providing a comprehensive and consistent view of the organization's information assets. By offering subject-oriented, integrated, and non-volatile data, it empowers decision-makers with valuable insights for strategic planning, performance analysis, and informed decision-making.

Data Mart (DM) A **Data Mart (DM)** serves as a specialized subset or aggregation of the data stored within a primary DWH. Unlike the comprehensive nature of a DWH, a **DM** contains a focused set of information tailored to *meet the needs of a specific business area*, corporate department, or category of users.

One of the key roles of **DMs** is to act as building blocks during the incremental development of DWHs. Rather than attempting to construct an entire DWH in one go, organizations often adopt an iterative approach, creating smaller, more *targeted DMs* that address immediate business needs. As the organization's analytical requirements evolve, additional DMs can be added or expanded upon, gradually contributing to the development of a comprehensive DWH architecture.

Moreover, **DMs** help to address the users' queries more efficiently. By tailoring the data content and structure to align with the analytical needs of particular departments or user categories, DMs facilitate more efficient and focused analysis. This granularity enables users to access and analyze relevant data without being overwhelmed by the vast volume of information typically found in a primary DWH.

Furthermore, **DMs** often offer improved performance compared to primary DWHs. Due to their smaller size and targeted scope, DMs can deliver faster query response times and better overall system performance. By focusing on specific subsets of data, data marts reduce the complexity of queries and minimize the processing overhead associated with accessing and retrieving information.



Online Analytical Processing (OLAP)

First of all, data are organised in **tables**, which are defined as collections of related, homogeneous data arranged into a row-column format, which includes:

- **Rows:** the various components stored in the table about a *specific individual*;
- **Columns:** the type and the meaning of a *specific component* of the individuals represented in the table;
- **Key:** a column or a set of columns whose values allow to distinguish *univocally* the rows of the table.

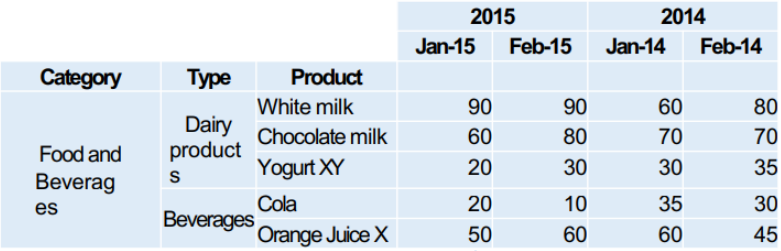
On-Line Analytical Processing (OLAP) allows users to interactively navigate the data warehouse information exploiting the multidimensional model, providing a flexible and intuitive way to explore, query, and report on large datasets. Typically, the data are analyzed at different levels of aggregation, by applying subsequent ***OLAP operators***, each yielding one or more different queries.

In an **OLAP Session** the user can scout the multidimensional model choosing the next operator based on the outcome of the previous ones. In this way, the user creates a navigation path that corresponds to an analysis process for facts according to different points and at different detail levels.

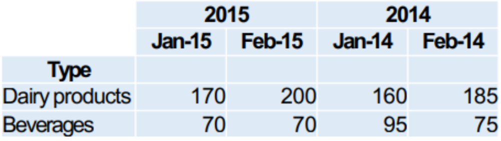
$$\textit{Product} \longrightarrow \textit{Sub-Category} \longrightarrow \textit{Category}$$

The **OLAP operators** are the following:

- **Roll-up:** causes an increase in data aggregation and removes a detail level from a hierarchy by collapsing the rows that have a feature in common.

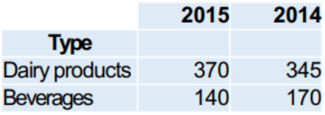


			2015		2014	
			Jan-15	Feb-15	Jan-14	Feb-14
Category	Type	Product				
Food and Beverages	Dairy products	White milk	90	90	60	80
		Chocolate milk	60	80	70	70
		Yogurt XY	20	30	30	35
	Beverages	Cola	20	10	35	30
		Orange Juice X	50	60	60	45

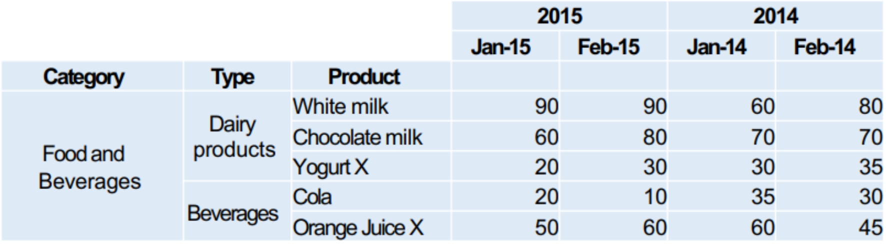


	2015		2014	
	Jan-15	Feb-15	Jan-14	Feb-14
Type				
Dairy products	170	200	160	185
Beverages	70	70	95	75

- **Drill-down:** is the complement to the roll-up operator; it reduces data aggregation and adds a new detail level to a hierarchy (e.g., from category to subcategory).

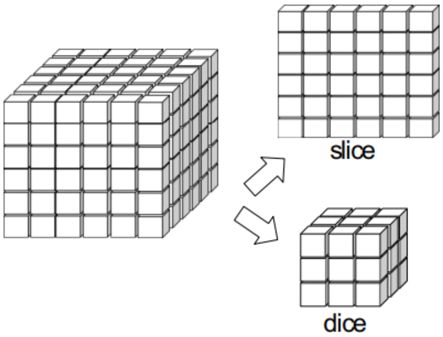


	2015	2014
Type		
Dairy products	370	345
Beverages	140	170

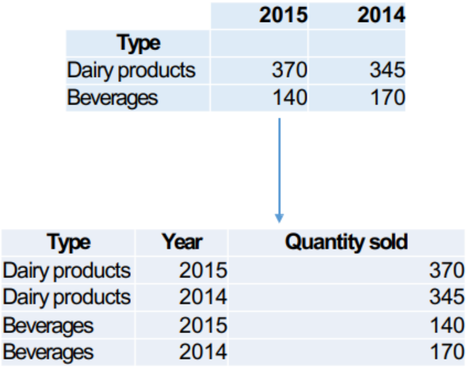


			2015		2014	
			Jan-15	Feb-15	Jan-14	Feb-14
Category	Type	Product				
Food and Beverages	Dairy products	White milk	90	90	60	80
		Chocolate milk	60	80	70	70
		Yogurt X	20	30	30	35
	Beverages	Cola	20	10	35	30
		Orange Juice X	50	60	60	45

- **Slice-and-dice:** the *slicing* operation reduces the number of cube dimensions after setting one of the dimensions to a specific value (e.g., category = 'Food and Beverages'); the *dicing* operation reduces the set of data being analysed by a selection criterion.



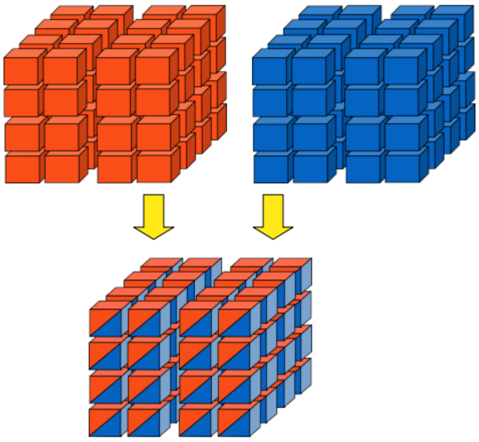
- **Pivot:** implies a change in layouts, aiming at analysing a group of data from a different viewpoint.



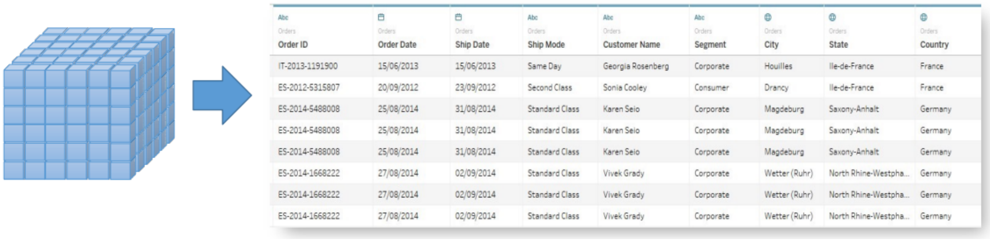
	2015	2014
Type		
Dairy products	370	345
Beverages	140	170

Type	Year	Quantity sold
Dairy products	2015	370
Dairy products	2014	345
Beverages	2015	140
Beverages	2014	170

- **Drill-across:** allows to create a link between concepts in interrelated cubes, in order to compare them.



- **Drill-through:** switches from multidimensional aggregate data to operational data insources or in the reconciled layer.



Orders Order ID	Orders Order Date	Orders Ship Date	Orders Ship Mode	Orders Customer Name	Orders Segment	Orders City	Orders State	Orders Country
IT-2013-1191900	15/06/2013	15/06/2013	Same Day	Georgia Rosenberg	Corporate	Houilles	Ile-de-France	France
ES-2012-5315807	20/09/2012	23/09/2012	Second Class	Sonia Cooley	Consumer	Drancy	Ile-de-France	France
ES-2014-5488008	25/08/2014	31/08/2014	Standard Class	Karen Seio	Corporate	Magdeburg	Saxony-Anhalt	Germany
ES-2014-5488008	25/08/2014	31/08/2014	Standard Class	Karen Seio	Corporate	Magdeburg	Saxony-Anhalt	Germany
ES-2014-5488008	25/08/2014	31/08/2014	Standard Class	Karen Seio	Corporate	Magdeburg	Saxony-Anhalt	Germany
ES-2014-1668222	27/08/2014	02/09/2014	Standard Class	Vivek Grady	Corporate	Wetter (Ruhr)	North Rhine-Westpha...	Germany
ES-2014-1668222	27/08/2014	02/09/2014	Standard Class	Vivek Grady	Corporate	Wetter (Ruhr)	North Rhine-Westpha...	Germany
ES-2014-1668222	27/08/2014	02/09/2014	Standard Class	Vivek Grady	Corporate	Wetter (Ruhr)	North Rhine-Westpha...	Germany

Extraction, Transformation and Loading (ETL)

The ***ETL (Extract, Transform, Load)*** process is a crucial component of data warehousing and business intelligence. It involves **extracting** data from various sources, **cleansing** them, **transforming** them into a consistent format, and **loading** them into a target destination, such as a data warehouse, where they can be analyzed and queried effectively. Let's analyse the single phases.

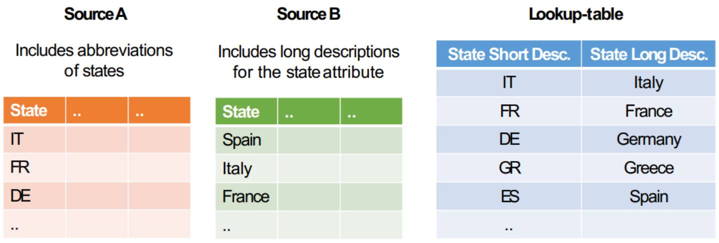
Extraction

In the **extraction phase**, data (*structured* or *unstructured*) is collected from multiple disparate sources, including databases, flat files, APIs, and external systems. The methods used for this process vary depending on the source systems and the nature of the data:

- **Static extraction:** retrieving all the data from the source. Used to populate the data warehouse for the first time.
- **Incremental extraction:** updating the data warehouse regularly only with the data that have been modified. In this case data comes with an associated *timestamp* and *triggers* (related to change transactions for relevant data).
- **Real-Time extraction:** continous stream of data.

Cleansing

This phase is about all those procedures to improve the **quality** of the retrieved data, by standardizing it and correcting **mistakes** like *duplicate or missing data*, *unexpeted use of some fields*, *impossible or wrong values* and **inconsistences** due to *different practices used* or *typing mistakes*. Each type of problem requires different techniques for its solution. We can distinguish three main techniques:



Source A

Includes abbreviations of states

State	..	..
IT		
FR		
DE		
..		

Source B

Includes long descriptions for the state attribute

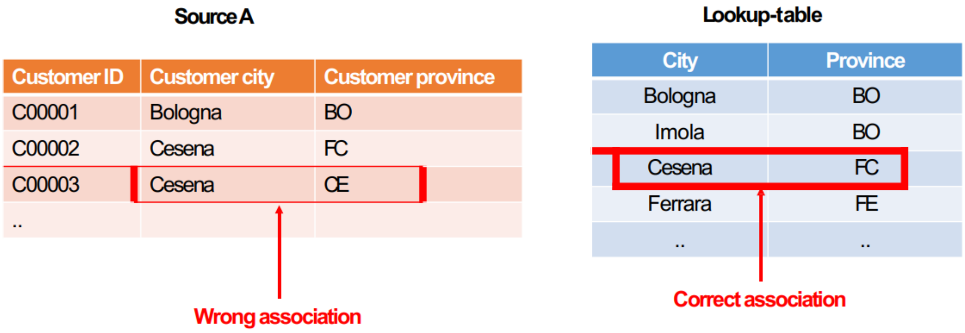
State	..	..
Spain		
Italy		
France		
..		

Lookup-table

State Short Desc.	State Long Desc.
IT	Italy
FR	France
DE	Germany
GR	Greece
ES	Spain
..	

Figure 2.2: Format discrepancies

- **Dictionary-based techniques:** they are used to check the correctness of the attribute values based on *lookup tables* and *dictionaries* to search for abbreviations and synonyms. We can apply these techniques if the domain is known and limited. These techniques are suitable for typing mistakes and format discrepancies [Figure 2.2, 2.3].



Source A

Customer ID	Customer city	Customer province
C00001	Bologna	BO
C00002	Cesena	FC
C00003	Cesena	CE
..		

Lookup-table

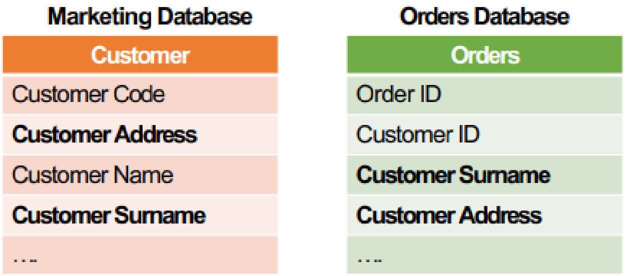
City	Province
Bologna	BO
Imola	BO
Cesena	FC
Ferrara	FE
..	..

Wrong association

Correct association

Figure 2.3: Inconsistences between correlated attributes

- **Approximate merging:** we use this technique when we need to merge data coming from different sources and we don't have a common key to identify matching tuples:
  - *Approximate join* - comparing records from different datasets using similarity measures or matching algorithms to identify potentially related records. (?)



Marketing Database

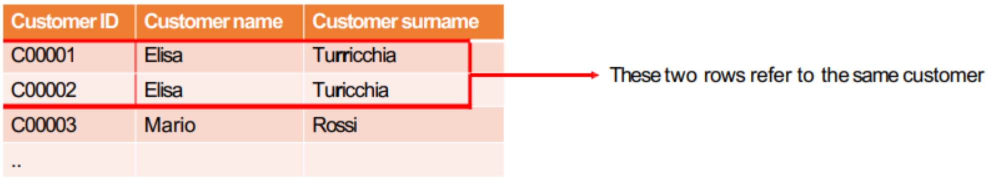
Customer
Customer Code
Customer Address
Customer Name
Customer Surname
….

Orders Database

Orders
Order ID
Customer ID
Customer Surname
Customer Address
….

Figure 2.4: Approximate join on *Customer Address* and *Customer Surname*

  - *Similary functions* - usage of **affinity functions** (Levenshtein distance, Jaccard similarity, etc.) to compute the similarity between two words and if the result is higher/lower than a treshold, then the two words are the same and the rows can be merged.



Customer ID	Customer name	Customer surname
C00001	Elisa	Turricchia
C00002	Elisa	Turicchia
C00003	Mario	Rossi
..		

These two rows refer to the same customer

- **Ad-hoc algorithms:** custom algorithms based on specific business rules.

Transformation

In this phase, data from sources is properly transformed to adjust its format to the reconciled schema. It includes:

- **Conversion:** changes on data types and format, like:
  - *date conversion*: from date to number (e.g. 12/11/2018 ⟶ 20181112)
  - *string conversion*: lowercase to uppercase (e.g. unibo ⟶ UNIBO)
  - *naming convention transformation*: short description to long description (e.g. IT ⟶ Italy)
- **Enrichment:** combination of one or more attribute to create new information, like derived data (e.g. Profit = Receipts - Expenses).
- **Separation/Concatenation:** attributes concatenation (e.g. customer surname || customer name)
- **Denormalization/Normalization:** organization of data in tables, where each piece appears only once for normalization and introduction of small redundancy to improve query performance. Typically, in the DWH the data is *denormalized*.

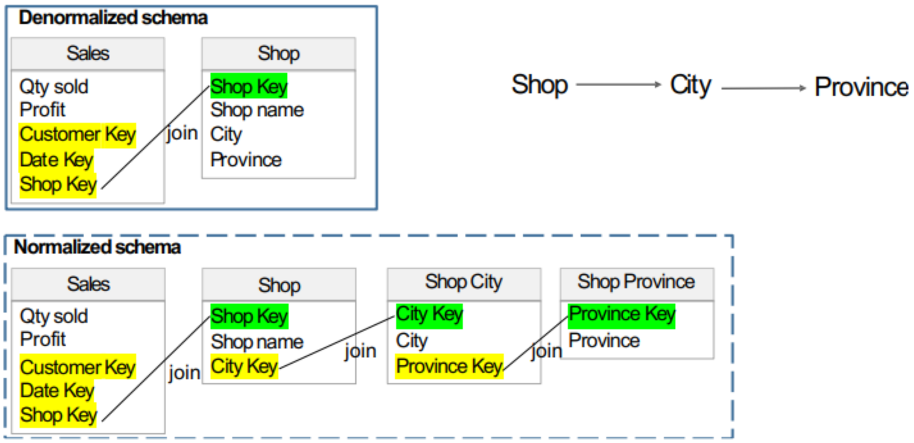


Figure 2.5: Denormalization process

Loading

This is the phase of loading data into a DWH, which can be done in batches or in real-time, depending on the volume of data. During the loading process, data integrity and consistency are maintained to ensure that the data remains accurate and usable for analysis. There are two ways of approaching this phase:

- **Refresh:** the DWH is completely rewritten (i.e. older data is replaced). It's used in combination with static extraction.
- **Update:** only those changes applied to source data are added to the DWH. Pre-existing data is not deleted or modified. It's used in combination with incremental extraction to regularly update the DWH.

Data Warehouse Architectures

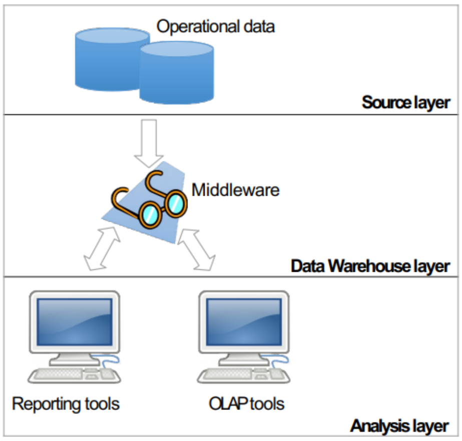
The requirements that a Data Warehouse has to satisfy are the following:

- **Separation:** analytical and transactional processing should be kept apart as much as possible.
- **Scalability:** hardware and software architectures should be easy to upgrade as the data volume, which has to be managed and processed, and the number of users' requirements, which have to be met, progressively increase.
- **Extensibility:** the architecture should be able to host new applications and technologies without redesigning the whole system.
- **Security:** monitoring accesses is essential because of the strategic data stored in data warehouses.
- **Administrability:** DWH management should not be overly difficult.

Single-Layer Architecture

A ***Single-Layer architecture*** for a data warehouse provides a straightforward and integrated approach to data storage, processing, and presentation. The source layer is the only physically available layer and its goal is to minimize the amount of data stored, removing data redundancies. DWH is implemented as a multidimensional view of operational data created by specific *middleware*.

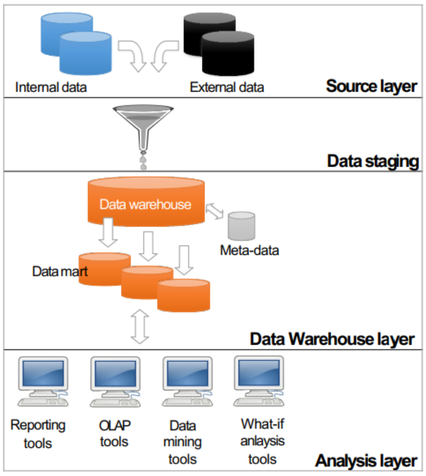
Even if this architecture minimizes the space occupation, there is no separation between analytical and transactional processing, which is not ideal for large DWHs and for supporting complex analytical workloads efficiently.



Two-Layer Architecture

A ***Two-Layers architecture*** provides a separation between physically available sources and data warehouses and includes:

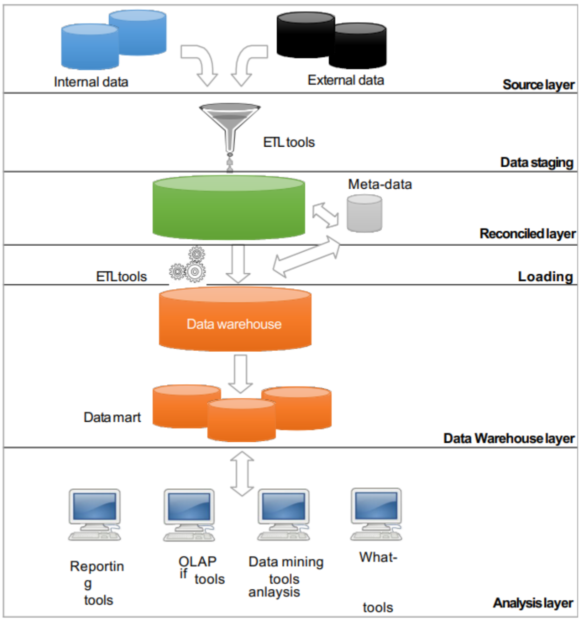
- **Source layer:** : it includes a set of heterogeneous data sources (both *internal* and *external sources*)
- **Data staging:** the data stored to sources should be extracted, cleansed to remove inconsistencies and fill gaps, and integrated to merge heterogeneous sources into one common schema. It includes *Extraction, Transformation and Loading (ETL)* procedures.
- **Data Warehouse layer:** information is stored to one logically centralized single repository that can be directly accessed or it can be used as source for creating *data marts*. *Meta-data repositories* then store information on sources, data staging, data mart schemata, etc.
- **Analysis layer:** it is accessed by end-user to create reports, dashboard, simulate hypothetical business scenarios, etc.



Three-Layer Architecture

In a ***Three-Layers architecture*** the layers are the same as described in the previous case, with one difference:

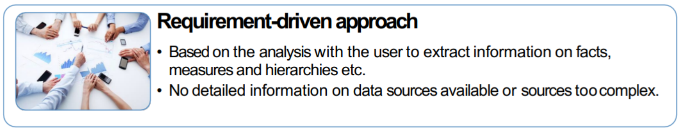
- **Reconciled layer:** this layer materializes operational data obtained after integrating and cleansing source data. The result data are integrated, consistent, correct, current and detailed.



Then, the reconciled data layer creates a common reference data model for a whole enterprise and it separates the problems of source data extraction and integration from those of data warehouse population. In addiction, the reconciled data leads to more redundancy of operational source data.

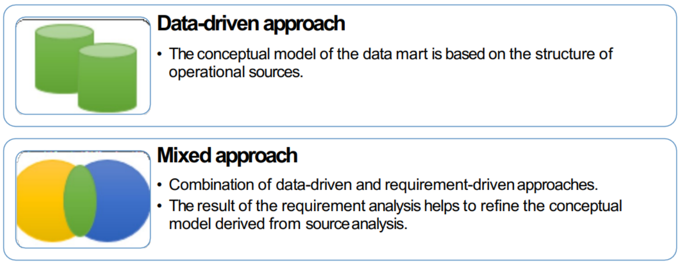
Conceptual Modeling: The Dimensional Fact Model (DFM)

Conceptual modeling is about creating a high-level representation of a data mining problem that serves as a blueprint to design and implement the solutions, through a framework that guides the entire process. Basically, we have three approaches:



Requirement-driven approach

- Based on the analysis with the user to extract information on facts, measures and hierarchies etc.
- No detailed information on data sources available or sources too complex.



Data-driven approach

- The conceptual model of the data mart is based on the structure of operational sources.

Mixed approach

- Combination of data-driven and requirement-driven approaches.
- The result of the requirement analysis helps to refine the conceptual model derived from source analysis.

The ***Dimensional Fact Model (DFM)*** is a conceptual model created specifically to function as *data mart design support* in order to emphasize simplicity, flexibility and ease to use for end-users. It is graphic and based on the multidimensional model. Its aim is to:

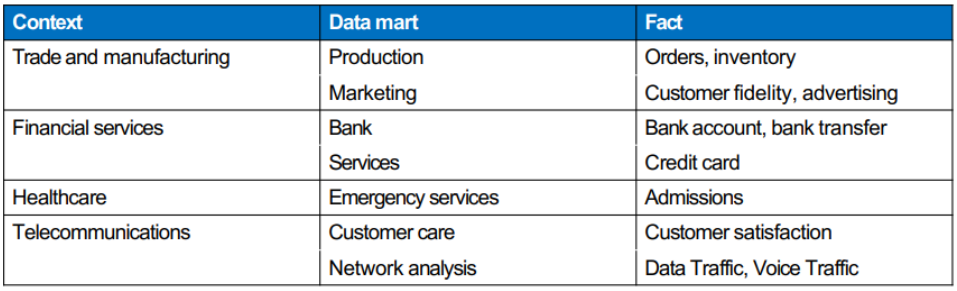
- provide support to conceptual design;
- create an environment in which user queries may be formulated intuitively;
- favor communication between designers and end users to formalize requirement specifications;
- build a stable platform for logical design;
- provide clear and effective design documentation.

The entire model is deigned around some key concepts:

Fact	It is a concept relevant to decision-making processes. It typically models a set of events taking place within a company.	Sales, purchases
Measure	It is a numerical property of a fact and describes a quantitative fact aspect that is relevant to analysis.	Quantity, discount
Dimension	It is a fact property with a finite domain and describes an analysis coordination.	Date, product, store
Dimensional attribute	Dimensions and other possible attributes, always with discrete values, that describe them.	Category of product
Hierarchy	It is a directed tree whose nodes are dimensional attributes and whose arcs model many-to-one associations between dimensional attribute pairs. It includes a dimension, positioned at the tree's root and all of the dimensional attributes that describe it.	Date → Month → Year
Primary event	It is a particular occurrence of a fact, identified by on n-ple made up of a value for each dimension. A value for each measure is associated with each primary event.	

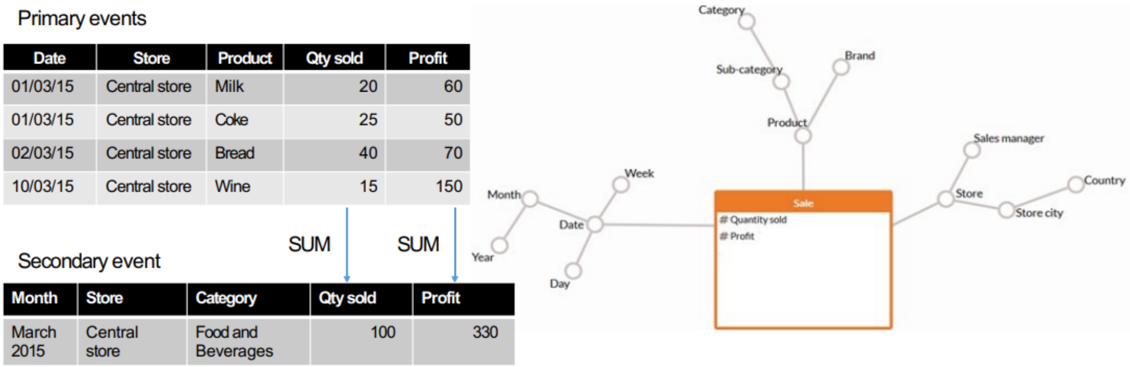
Secondary event

Given a set of dimensional attributes, each n-ple of their values identifies a secondary event that aggregates all of the corresponding primary events. Each secondary event is associated with a value for each measure that sums up all the values of the same measure in the corresponding primary events.



Context	Data mart	Fact
Trade and manufacturing	Production	Orders, inventory
	Marketing	Customer fidelity, advertising
Financial services	Bank	Bank account, bank transfer
	Services	Credit card
Healthcare	Emergency services	Admissions
Telecommunications	Customer care	Customer satisfaction
	Network analysis	Data Traffic, Voice Traffic

Figure 2.6: Examples of data marts and related facts



Primary events

Date	Store	Product	Qty sold	Profit
01/03/15	Central store	Milk	20	60
01/03/15	Central store	Coke	25	50
02/03/15	Central store	Bread	40	70
10/03/15	Central store	Wine	15	150

Secondary event

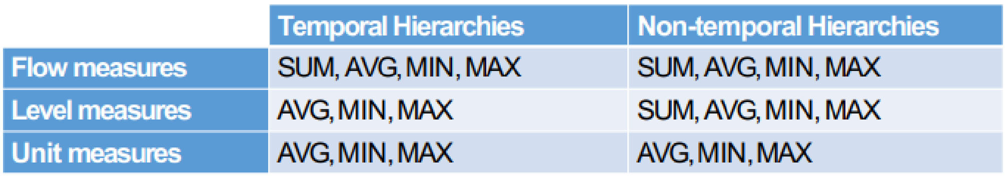
Month	Store	Category	Qty sold	Profit
March 2015	Central store	Food and Beverages	100	330

Figure 2.7: Examples of primary and secondary events

Aggregation requires the definition of a suitable *operator* to compose the measure values that mark primary events into values to be assigned to secondary events. Measures are classified into:

- **Flow measures:** refer to a timeframe, at the end of which they are evaluated cumulatively (e.g. quantity sold).
- **Level Measures:** are evaluated at particular times (e.g. number of products ininventory).
- **Unit measures:** are evaluated at particular times but are expressed inrelative terms (e.g. unit price).

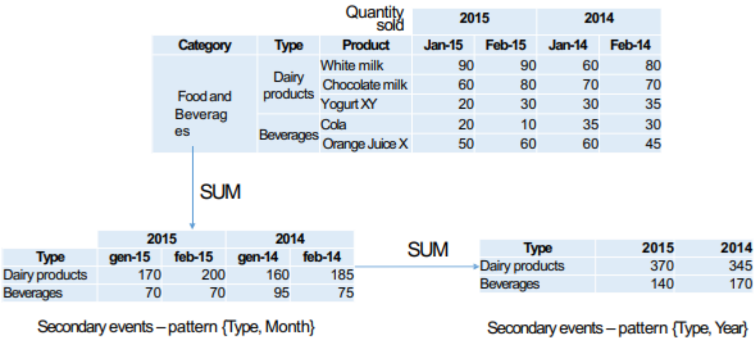
A measure is called **additive** along a dimension when you can use the *SUM operator* to aggregate its values along the dimension hierarchy. If this is not the case, it is called **non-additive**. A non-additive measure is **non-aggregable** when you can use no aggregation operator for it.



	Temporal Hierarchies	Non-temporal Hierarchies
Flow measures	SUM, AVG, MIN, MAX	SUM, AVG, MIN, MAX
Level measures	AVG, MIN, MAX	SUM, AVG, MIN, MAX
Unit measures	AVG, MIN, MAX	AVG, MIN, MAX

Given these measures, we can apply different types of *aggregation operators*, as shown in the figure above:

- **Distributive:** calculating aggregates from partial aggregates (e.g. SUM, MIN, MAX).



		Quantity sold	2015		2014	
Category	Type	Product	Jan-15	Feb-15	Jan-14	Feb-14
Food and Beverag es	Dairy products	White milk	90	90	60	80
		Chocolate milk	60	80	70	70
		Yogurt XY	20	30	30	35
	Beverages	Cola	20	10	35	30
		Orange Juice X	50	60	60	45

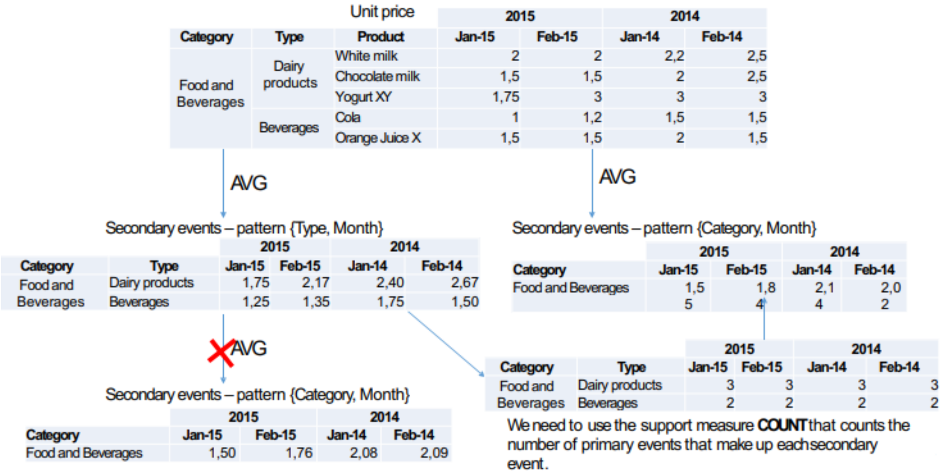
	2015		2014	
Type	gen-15	feb-15	gen-14	feb-14
Dairy products	170	200	160	185
Beverages	70	70	95	75

Secondary events – pattern {Type, Month}

Type	2015	2014
Dairy products	370	345
Beverages	140	170

Secondary events – pattern {Type, Year}

- **Algebraic:** requiring the usage of additional information in the form of a finite number of support measures to correctly calculate aggregates from partial aggregates (e.g. AVG).



		Unit price	2015		2014	
Category	Type	Product	Jan-15	Feb-15	Jan-14	Feb-14
Food and Beverages	Dairy products	White milk	2	2	2,2	2,5
		Chocolate milk	1,5	1,5	2	2,5
		Yogurt XY	1,75	3	3	3
	Beverages	Cola	1	1,2	1,5	1,5
		Orange Juice X	1,5	1,5	2	1,5

Secondary events – pattern {Type, Month}

		2015		2014	
Category	Type	Jan-15	Feb-15	Jan-14	Feb-14
Food and Beverages	Dairy products	1,75	2,17	2,40	2,67
	Beverages	1,25	1,35	1,75	1,50

Secondary events – pattern {Category, Month}

	2015		2014	
Category	Jan-15	Feb-15	Jan-14	Feb-14
Food and Beverages	1,55	1,84	2,14	2,02

Secondary events – pattern {Category, Month}

	2015		2014	
Category	Jan-15	Feb-15	Jan-14	Feb-14
Food and Beverages	1,50	1,76	2,08	2,09

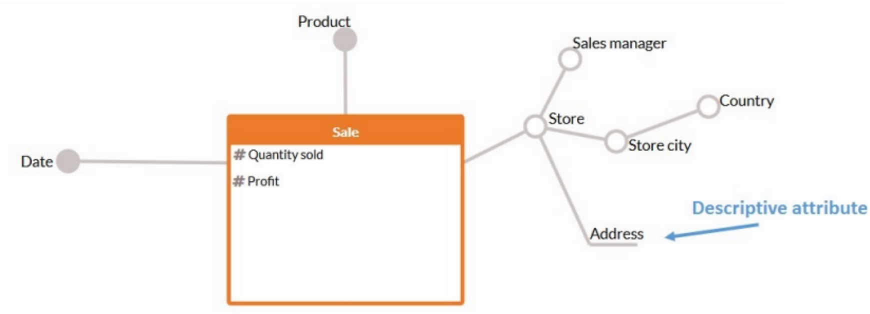
		2015		2014	
Category	Type	Jan-15	Feb-15	Jan-14	Feb-14
Food and Beverages	Dairy products	3	3	3	3
	Beverages	2	2	2	2

We need to use the support measure **COUNT** that counts the number of primary events that make up each secondary event.

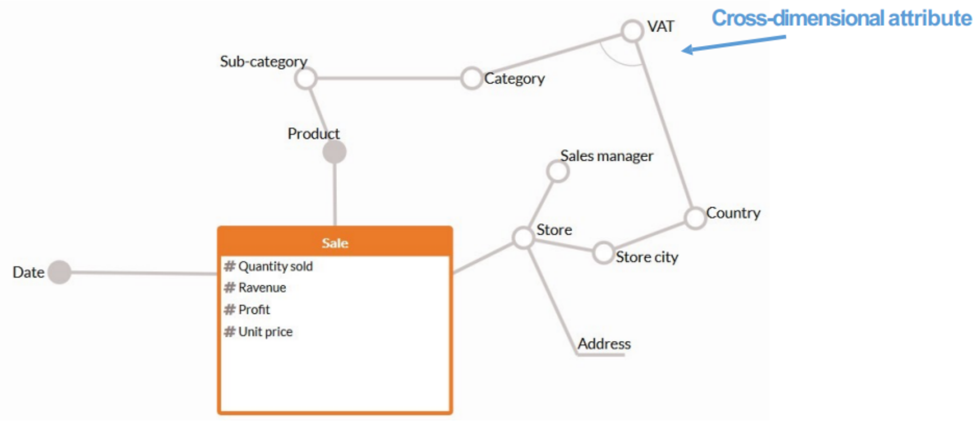
- **Holistic:** calculating aggregates from partial aggregates only via an infinite number of support measures (e.g. RANK).

In addiction to the base attributes, we also have some advanced ones:

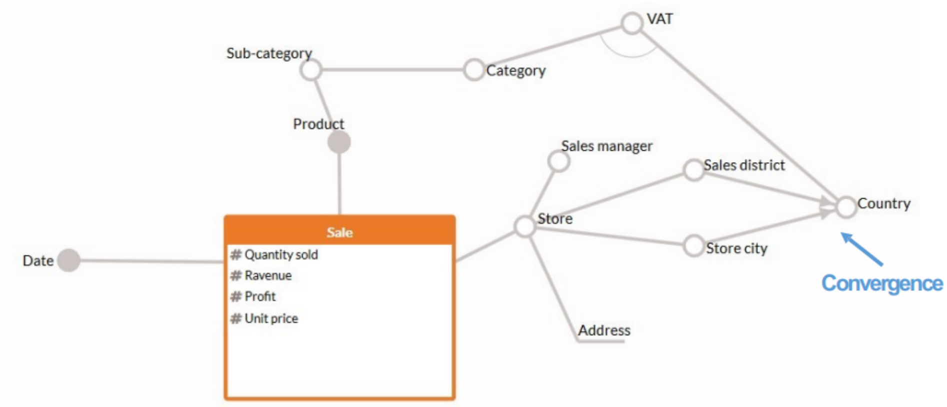
- **Descriptive attributes:** used to give additional information to specific dimensional attributes, but they are not used as aggregation criteria.



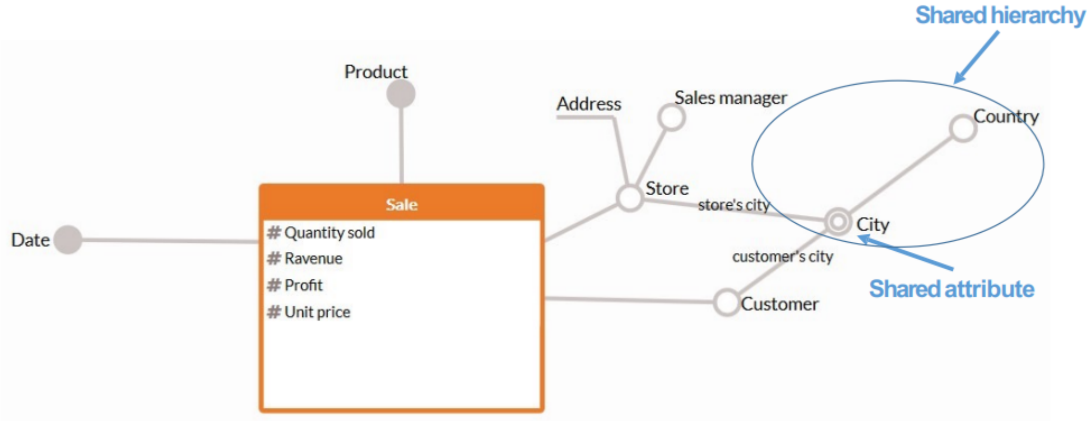
- **Cross-dimensional attributes:** a dimensional or descriptive attributes whose value is defined by the combination of two or more dimensional attributes, possibly belonging to different hierarchies.



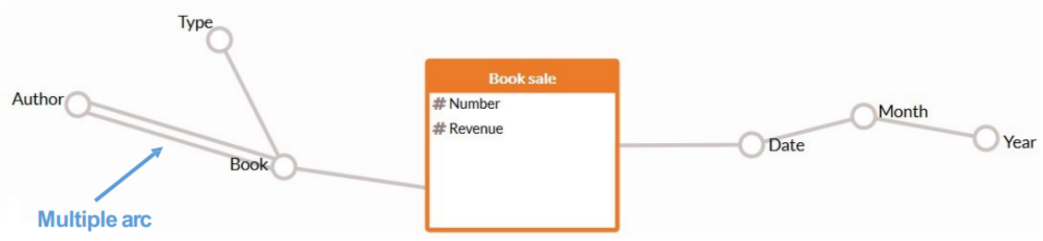
- **Convergence:** Two or more arcs belonging to the same hierarchy and ending at the same dimensional attribute.



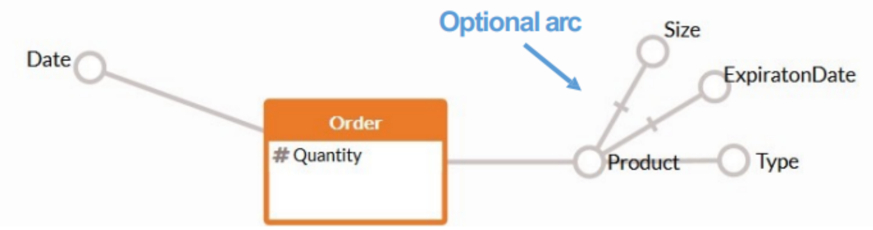
- **Shared hierarchies:** a double circle represents and emphasizes the first attribute to be shared (e.g., City). All descendants of the shared attribute are shared, too. For each incoming arc a role must be added (e.g. customer's city).



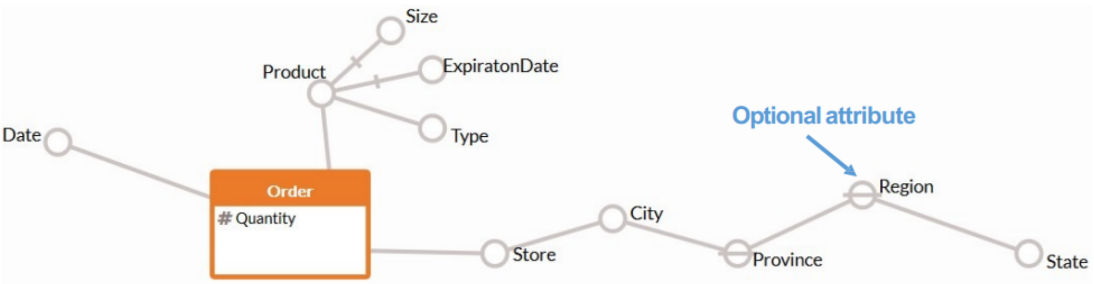
- **Multiple arcs:** the meaning of a multiple arc that goes from an attribute called a (e.g., book) to an attribute called b (e.g., author) is that a many-to-many association exists between a and b.



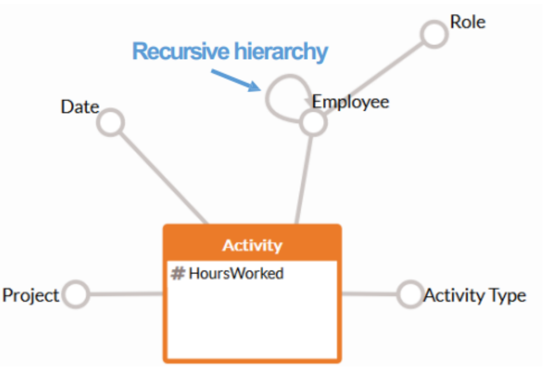
- **Optional arcs:** used to model scenarios for which an association represented in a fact schema is not defined for a subset of events.
- **Coverage:** when two or more optional arcs exit from the same attribute a (e.g., Product), you can specify their coverage:
  - the coverage is ***total*** if the value of at least one of the children is linked to each value of a. If, instead, values of a exist for which all of the children are undefined, the coverage is ***partial***.
  - the coverage is ***disjoint*** if you have a value for at most one of the children corresponding to each value of a. If, instead, values of a exist linking to values of two or more children, the coverage is ***overlapping***.

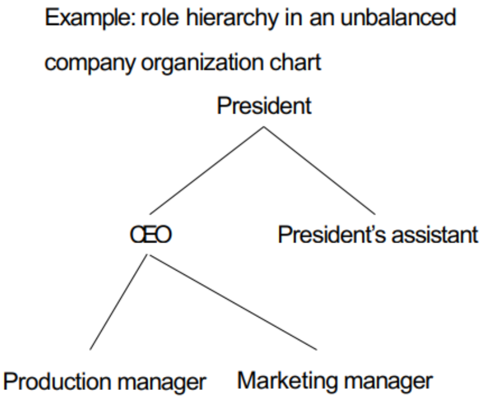


- **Incomplete hierarchies:** one in which one or more levels of aggregation prove missing in some instances (because they are not known or defined).



- **Recursive hierarchies:** represents a parent-child relationship among the levels.





Logical Design The ***logical design phase*** defines the data structures (e.g., set of tables and relationships between tables) that represent data marts according to the preselected logical model and optimizes performance by fine tuning these structures.

The logical design phase includes a set of steps that, starting from the conceptual schema, make it possible to define the logical schema of a data art. Three main steps to implement a logical schema in a relational DBMS:

- Translating fact schemata into logical schemata: mainly star or snowflake schemata.
- Materializing views: set of secondary view that aggregate primary view data to improve query performance.
- Fragmenting fact tables vertically and horizontally.

Star schema A star schema is characterized by fact tables and dimension tables. A fact table contains all the measures and descriptive attributes linked to a act. A dimensional table is created for each dimension and it includes all the hierarchy attributes [Figure 2.8].

Snowflake schema A star schema variant with dimension tables partially normalized [Figure 2.9].

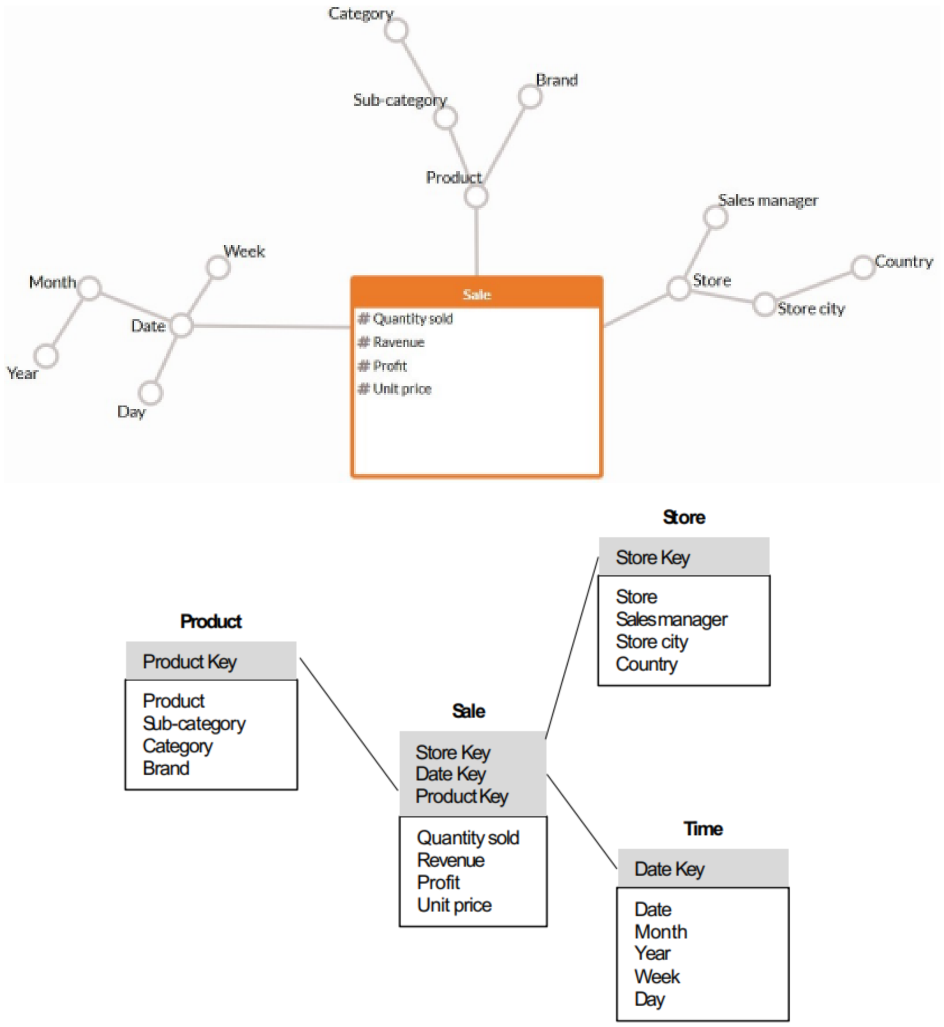
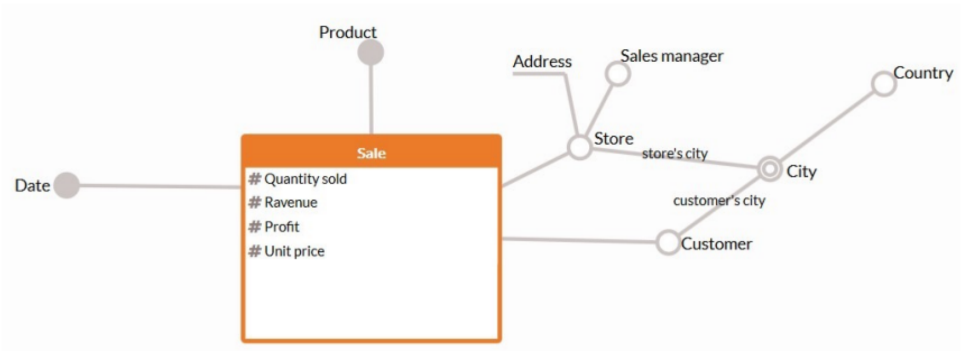


Figure 2.8: Star Schema



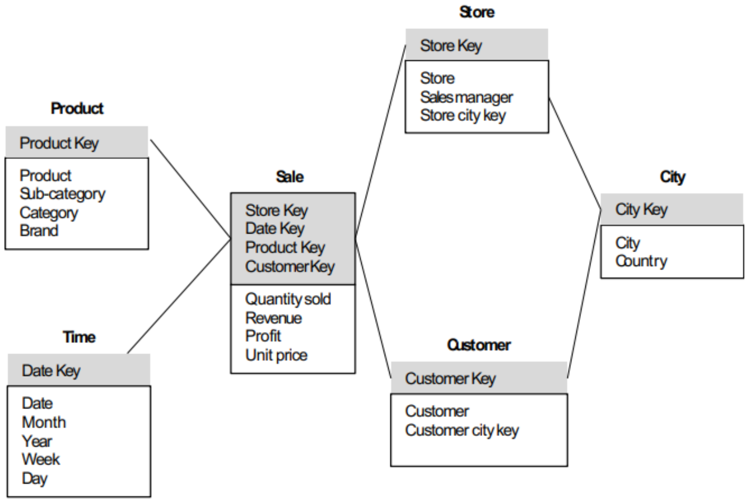


Figure 2.9: Snowflake Schema

Data Lake

The downside of data warehousing systems is that they can store only structured data and require a predefined schema. Hence, there are a lot of data that are not used for any business process and they need to be stored somewhere.

> *”The storage environments of EMEA organizations consist of 54% dark data, 32% redundant, obsolete and trivial data and 14% business-critical data. By 2020, this can add up to $891 billion in storage and management costs that can otherwise be avoided.”* [sources: Wikipedia and Datamation].

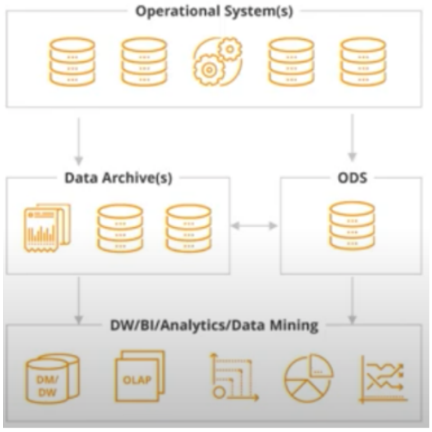
The solution is to use a ***Data Lake***, which is a *centralized repository* that allows you to store a vast amount of data in its **natural/raw format**, usually *object blobs* or *files*. Some of the keypoints of this approach are the following:

- **Flexibility:** data lakes support various types of data, including *structured data* from relational databases, *semi-structured data* like JSON or XML, and *unstructured data* such as text documents, images, videos, and log files.

File format	*Properties*
ORC	Columnar, schema in footer
Parquet	Columnar, schema in footer
AVRO	Row-major, schema and data separate
CSV	Human-readable, fixed schema
JSON	Human-readable, fixed schema

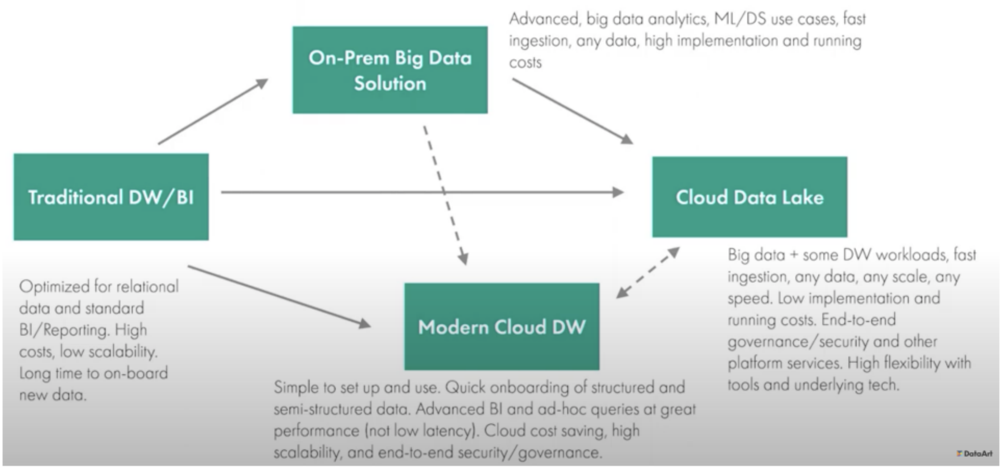
- **Schema-on-read:** unlike the **schema-on-write** approach used for traditional databases, where data needed to be processed in a particular predefined schema before being stored, in data lakes each data is stored in an unprocessed form and the schema is applied during the query phase. This allows users to interpret and analyze data in different ways without modifying the original data.

- **Cost-Effectiveness:** data lakes usually rely on low cost scalable storage solutions, such as *cloud object storage*, or *distributed file systems*.
- **Scalability:** data lakes are designed to scale horizontally, s.t. large volumes of data can be efficiently stored and processed. With such approach, they accomodate users in managing growing data volumes without significant architectural changes.



In the figure above, we can see how the traditional data analytics structure is, even if there are some negative aspects to be taken into account, like:

- siloed departmental data;
- long implentation cycles;
- limits of scalability and high costs, since the computation and the storage have to be scaled together;
- lack of support for unstructured data, ML/DS workloads and ad-hoc analytical queries.



Advanced, big data analytics, ML/DS use cases, fast ingestion, any data, high implementation and running costs

Optimized for relational data and standard BI/Reporting. High costs, low scalability. Long time to on-board new data.

Big data + some DW workloads, fast ingestion, any data, any scale, any speed. Low implementation and running costs. End-to-end governance/security and other platform services. High flexibility with tools and underlying tech.

Simple to set up and use. Quick onboarding of structured and semi-structured data. Advanced BI and ad-hoc queries at great performance (not low latency). Cloud cost saving, high scalability, and end-to-end security/governance.

Figure 2.10: Data architectures evolution and comparison [sources: DataArt]

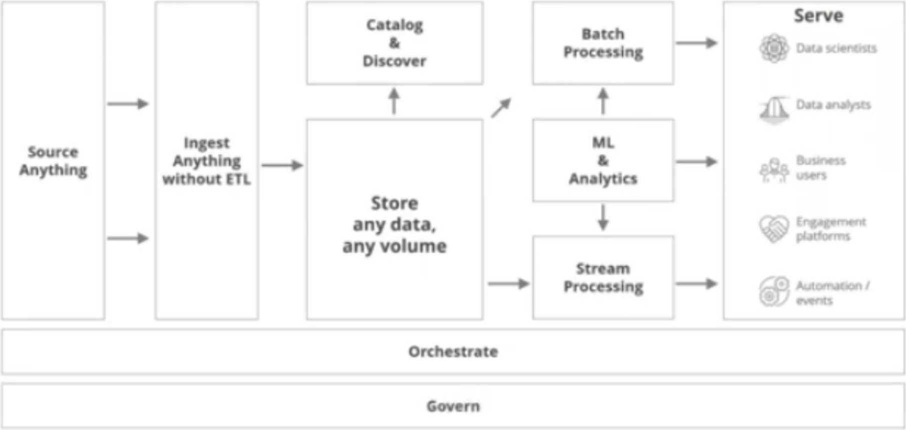
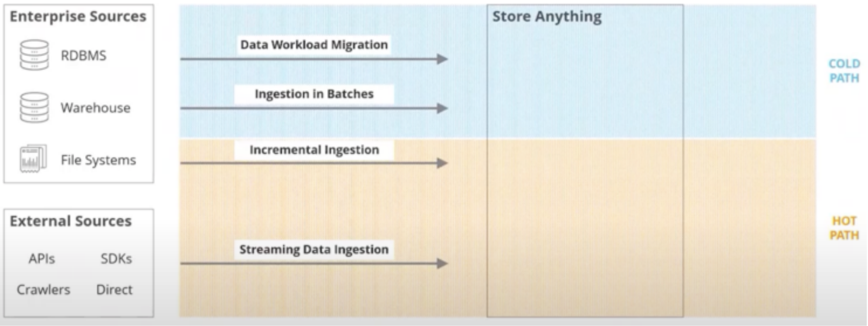
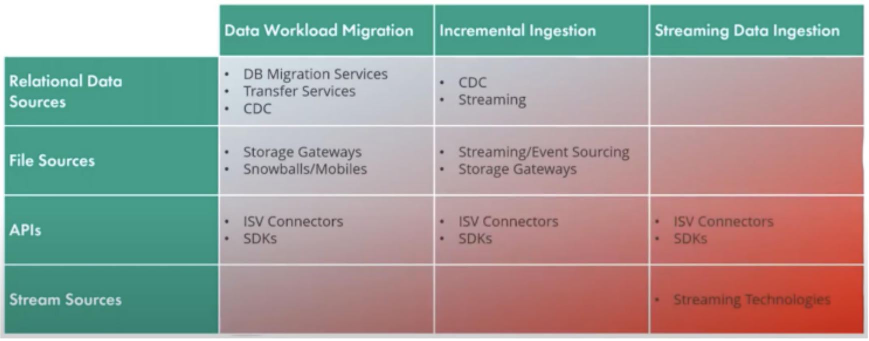


Figure 2.11: Data Lake components

Ingestion In the figure 2.11 we see how a data lake works, where the **data ingestion** is not so direct as it seems, since the ETL process is not performed and we have different ways of acquiring all the information depending on the sources, as shown below.





	Data Workload Migration	Incremental Ingestion	Streaming Data Ingestion
Relational Data Sources	• DB Migration Services • Transfer Services • CDC	• CDC • Streaming	
File Sources	• Storage Gateways • Snowballs/Mobiles	• Streaming/Event Sourcing • Storage Gateways	
APIs	• ISV Connectors • SDKs	• ISV Connectors • SDKs	• ISV Connectors • SDKs
Stream Sources			• Streaming Technologies

Figure 2.12: *Streaming data ingestion* for data logs or application events, while *CDC (Change Data Capture)* for continous ingestion, capturing streaming changes or database migration to cloud

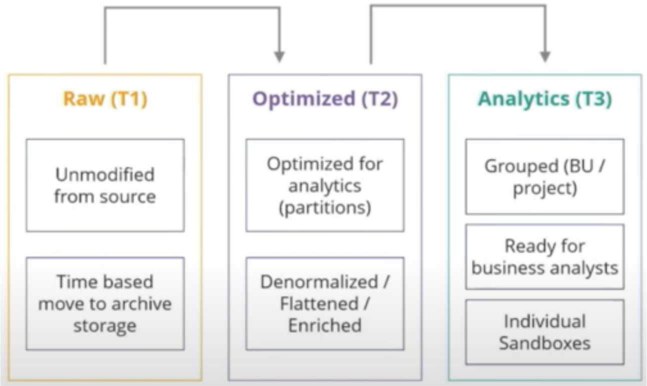
There are also some ***no-code tools*** which are based on high level blocks where you tune the parameters in order for the process to work. They can be useful for basic operations for their *simplicity*, but they have a very low *flexibility* with limited opportunities.

Storage Once we ingest anything, in the **storage phase** data can belong to different types:

Raw: immutable store that cannot/should not be changed after it has been written; self-descriptive with metadata; useful for disaster recovery.

Optimized: as raw data grow, querying directly them become slower to gain speed data can be transformed in optimized formats.

Analytics: BI-ready and machine-learning ready data and tables.


Raw (T1)
Unmodified from source
Time based move to archive storage
Optimized (T2)
Optimized for analytics (partitions)
Denormalized / Flattened / Enriched
Analytics (T3)
Grouped (BU / project)
Ready for business analysts
Individual Sandboxes

One important step now is to **catalog** data, in order to avoid the so called **data swamp**, which is a situation in which a data lake becomes disorganized, chaotic and unmanageable such that *no useful analysis is possible.* A lack of schema or descriptive metadata makes hard to consume or query the data, while a lack of semantic consistency makes challenging to perform analysis/mining.

In this case, ***metadata*** assume a crucial role by helping engineers, data scientists, and analysts to discover, understand, and use the data effectively and ensuring that the data lake remains a valuable resource for the organization. Examples are:

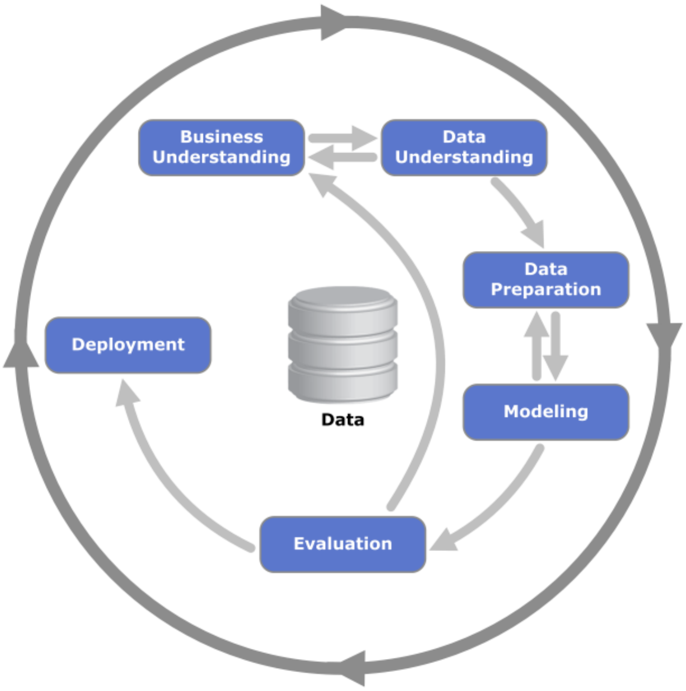
- **Data source:** information about where the data originated, such as the source system, data provider, or data feed.
- **Data schema:** describes the structure of the data, including the names and the data types of columns or attributes.
- **Data format:** specifies the file format or encoding used for the data files, which can be important for data processing and analytics.
- **Data quality metrics:** information about data quality such as the percentage of missing values or data accuracy statistics.
- **Data lifecycle:** information about how long the data should be retained in the data lake, data retention policies, and archiving rules.
- **Data ownership:** information about the team or department responsible for managing and maintaining the data.
- **Data lineage:** tracks the data's lineage, including its transformation processes, dependencies, and any data transformations applied to it.

- **Access control:** metadata can specify who has access to the data, what permissions they have, and any security measures in place.
- **Data classification:** indicates the sensitivity or classification level of the data, which can be crucial for compliance with data regulations.
- **Data usage information:** records how frequently the data is accessed, who accesses it, and what types of analytics or applications use the data.

The CRISP-DM methodology

Data Mining is a process and as all the other processes has steps and complex choices to be analyzed, managed and most of all standardized. One of the most used approaches in modern DM problems is the ***CRISP-DM methodology*** (*Cross-Industry Standard Process for Data Mining*), which provides a model to guide the Data Mining process in six major phases:

- **Business Understanding**
- **Data Understanding**
- **Data Preparation**
- **Modeling**
- **Evaluation**
- **Deployment**



Business Understanding In this initial phase, the focus is on understanding the **business objectives**, **requirements**, **costs** and **goals** of the data mining project. This involves identifying the business problem or opportunity, defining the project scope, and establishing **success criteria**.

Data Understanding In this phase, the focus shifts to understanding the data that will be used for the analysis. This includes **data collection**, **exploration**, and initial assessment to gain insights into the data **quality**, **structure**, and content. It involves identifying relevant data sources, assessing data availability, and understanding the relationships between different data elements.

Data Preparation In the data preparation phase, the raw data is processed and transformed into a format suitable for analysis. This involves tasks such as **cleaning**, **filtering**, **integrating**, and **transforming** the data to address issues such as missing values, outliers, and inconsistencies. Data preparation also includes feature engineering, where new variables or features may be created to improve the predictive power of the data. Usually this is a very expensive and time consuming phase.

Modeling The modeling phase focuses on selecting appropriate data mining techniques and building *predictive* or *descriptive models* to address the business objectives. This involves **selecting algorithms**, **partitioning the data** into training and testing sets, and **building** and **evaluating** models using techniques such as classification, regression, clustering, or association rule mining.

Evaluation In the evaluation phase, the performance of the models is assessed and evaluated against the project objectives and success criteria established in the business understanding phase, on a *qualitative* and *quantitative* basis. This involves testing the models on unseen data, analyzing their *performance metrics* (such as accuracy, precision, recall, or AUC), and **selecting the best-performing model** for deployment.

Deployment The final phase of the CRISP-DM process is deployment, where the selected model is integrated into the business operations or decision-making processes. This involves deploying the model into *production environments*, monitoring its performance over time, and incorporating feedback to ensure its effectiveness and reliability in real-world scenarios.

Machine Learning