2. Data Warehouse conceptual design

A data repository containing the database data, tipically obtained through operational databases, that will be used for analysis etc.

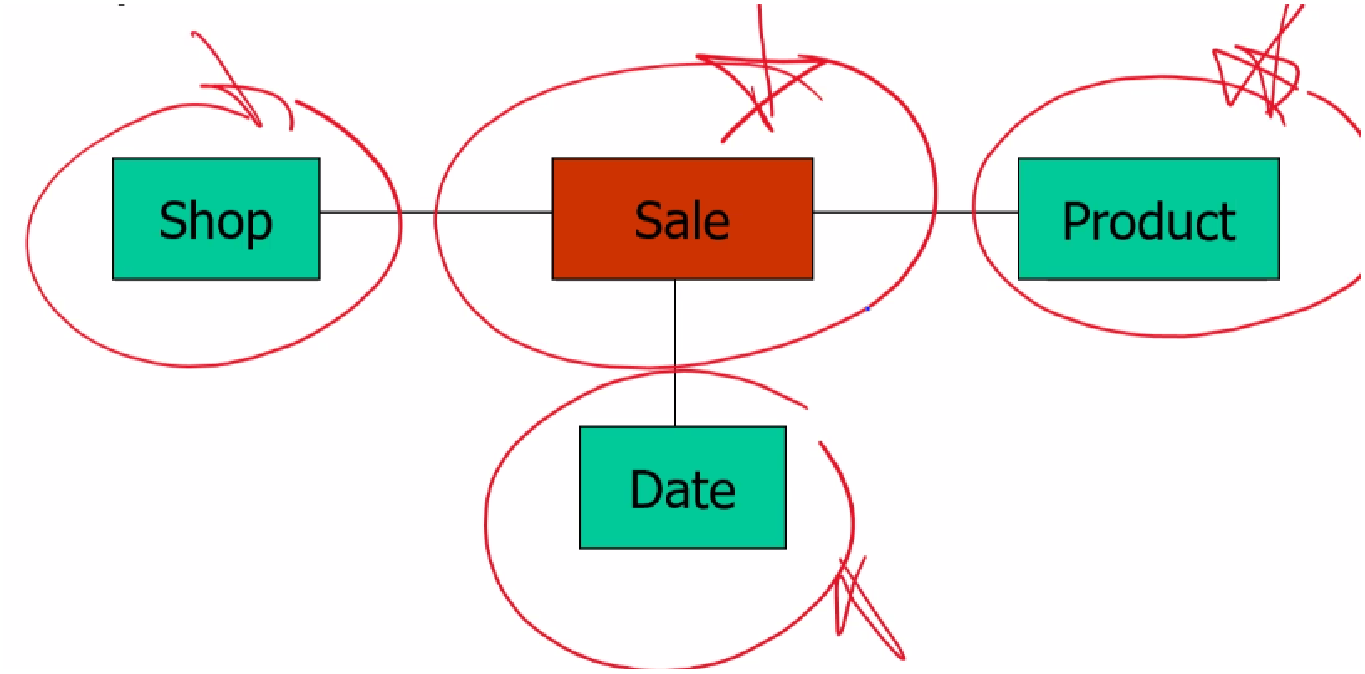
We separate those data to improve **performance:**

- complex queries reduce performances on a operational transaction management
- different access methods at the physical level
  and also to enforce the **ACID** properties on the operational database, since those are not needed on a data warehouse, where data are *only* updated periodically.
  Obviously we are introducing data redundancy.

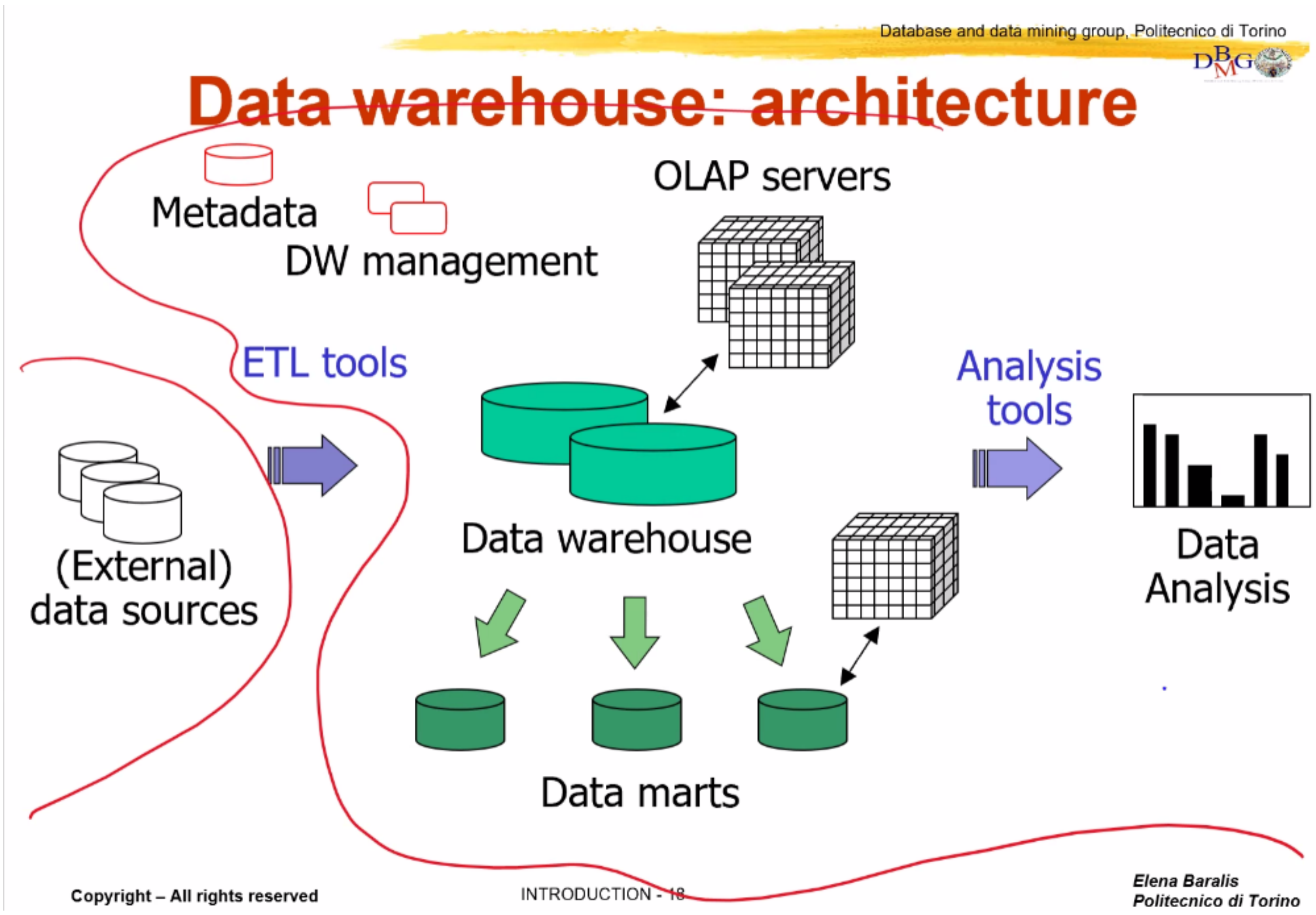
Representation of data

Data is conceptually represented as an hypercube, a multidimensional cube: this way we can display data aggregates on a 3d fashion.

Logically, is represented as a based-Entity-Relationship model.



Green boxes represent the three axis on a 3d chart. These dimensions are called slowly-changing dimensions as they rarely change.
*Sale is called a **fact table** and is quite huge in size as it keeps growing.*


Database and data mining group, Politecnico di Torino
Data warehouse: architecture
Metadata
DW management
OLAP servers
ETL tools
Analysis tools
(External) data sources
Data warehouse
Data Analysis
Data marts
Copyright – All rights reserved
INTRODUCTION - 18
Elena Baralis Politecnico di Torino

External data sources provide data warehouse the actual data

Data mart

A small subset of a data warehouse that focuses on a single subject.
It's less time-consuming and expensive to design a Data mart, as it's way smaller.
Also it's more extensible

Servers for data warehouse

- **ROLAP:** A realation OLAP server. Basically an RDBMS. Provides a more compact representation(only non-empty cells).
- **MOLAP:** A multidimensional OLAP server. Similar to the logical hypercube mentioned above. Sparse data require compression.
- **HOLAP:** Hybrid OLAP server.

ETL - Extraction, Transformation and Loading

Performed the first time the datawarehouse is loaded and during periodical datawarehouse refresh.

Metadata

Data about data for:

- **data transformation and loading:** data sources and needed transformation.
- **data management:** structure of the data in the datawarehouse
- **query management:** sql query code,pre-computed execution plan etc.

Three Level Architecture

The architecture shown above is unfeasible. The ETL process would be too complex.
Thus, an additional layer is introduced, called **Staging Area**, between the external sources and the actual data warehouse.
In this area, ETL operations are performed.
Obviously, this additional area introduces both redundancy and overhead.

Conceptual Design

We'll use the Dimensional Fact Model. A graphical model that supports conceptual design.

Fact

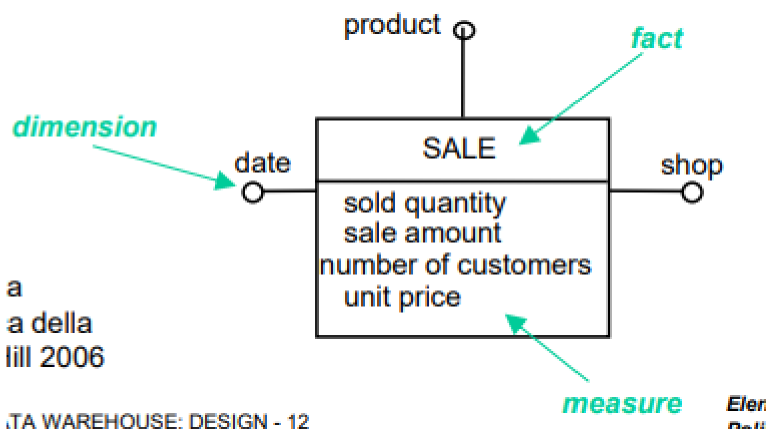
Models a **set of relevant events**(sales,shippings etc) and **evolves over time**. Basically, it represents a **fact of interest.**
It's the actual value of an hypercube

Dimension

Describes the coordinates of a fact(e.g. each sale is represented by date,shop and sold product).

Measure

Describes a numerical property of a fact(sold quantity for a sale)

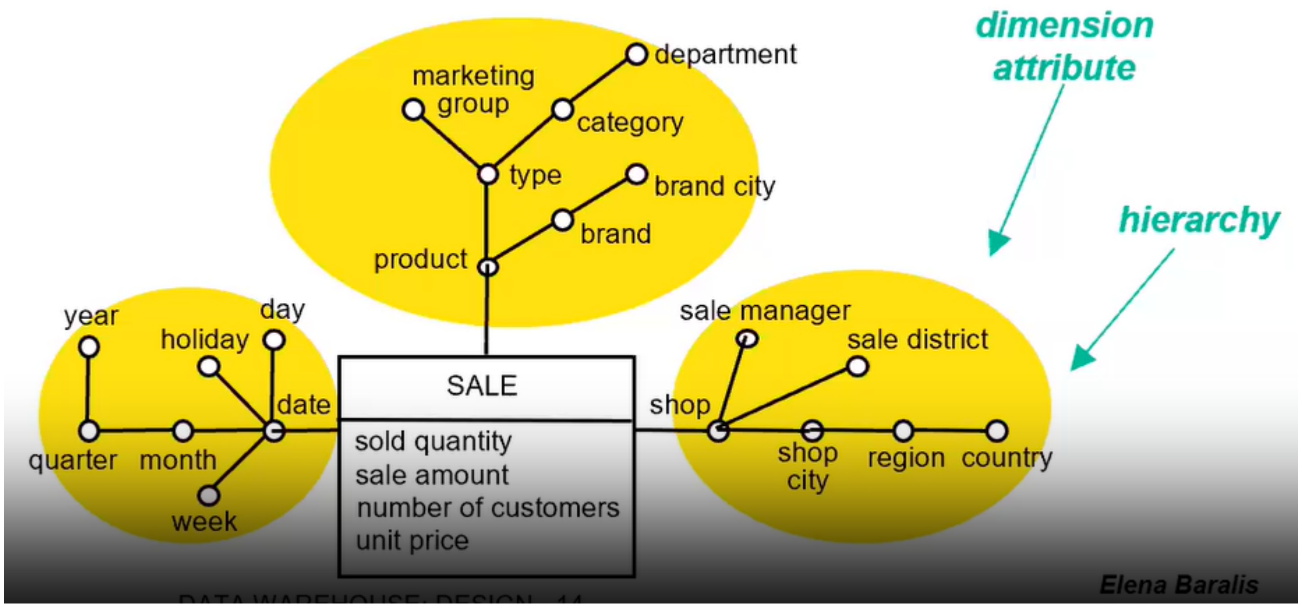


Example of the relation between fact,dimension and measure.

Hierarchy of DFM

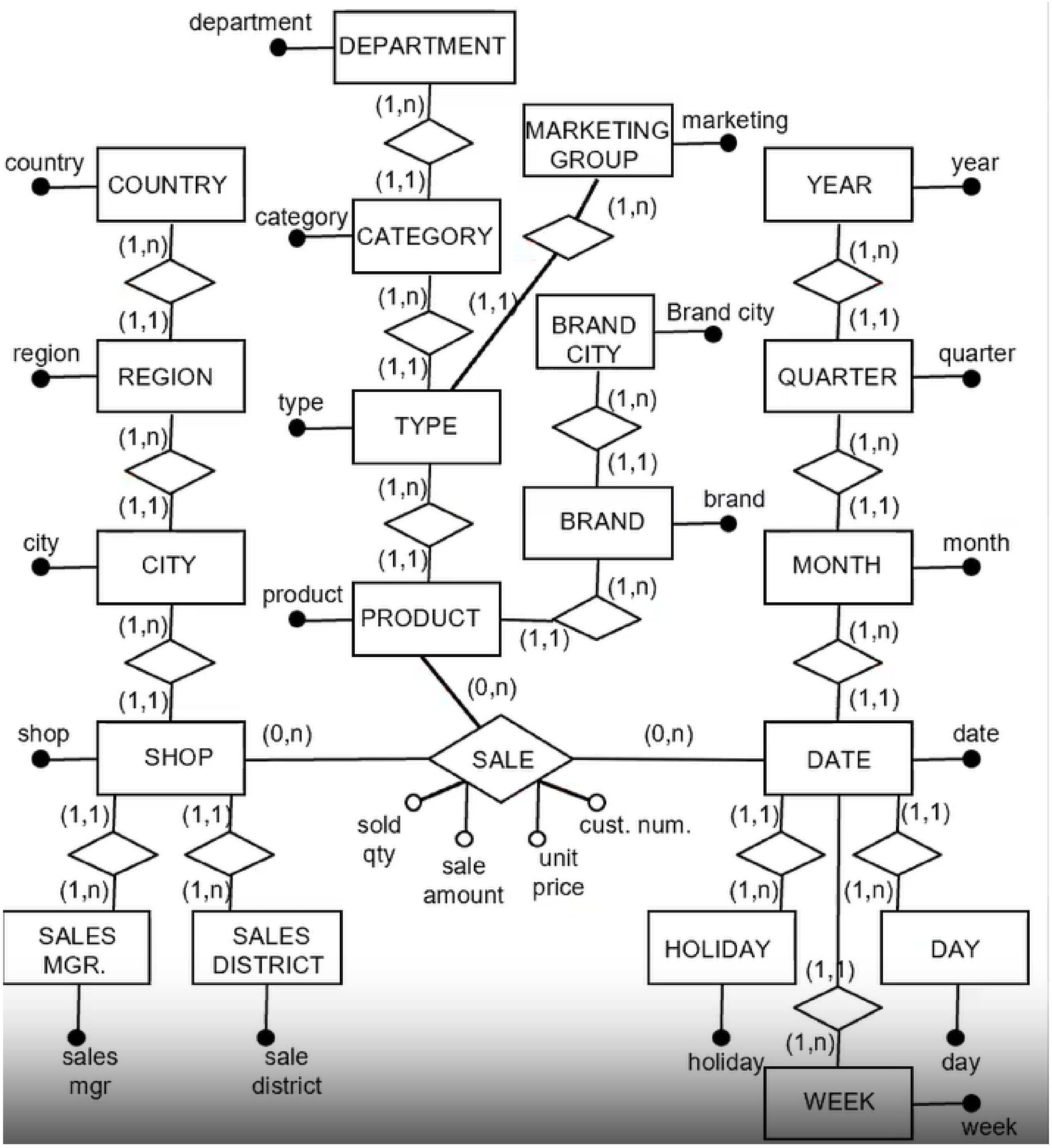
Each dimension can have a set of associated attributes.
This hierarchy represents:

- a generalization relationship among a subset of attributes in a dimension
- a function dependency(1:n)



Example of hierarchy.
Shop and ShopCity has a 1 to many relantionship(1:n). So does ShopCity with Region.

The DFM shown above is the same as the following ER

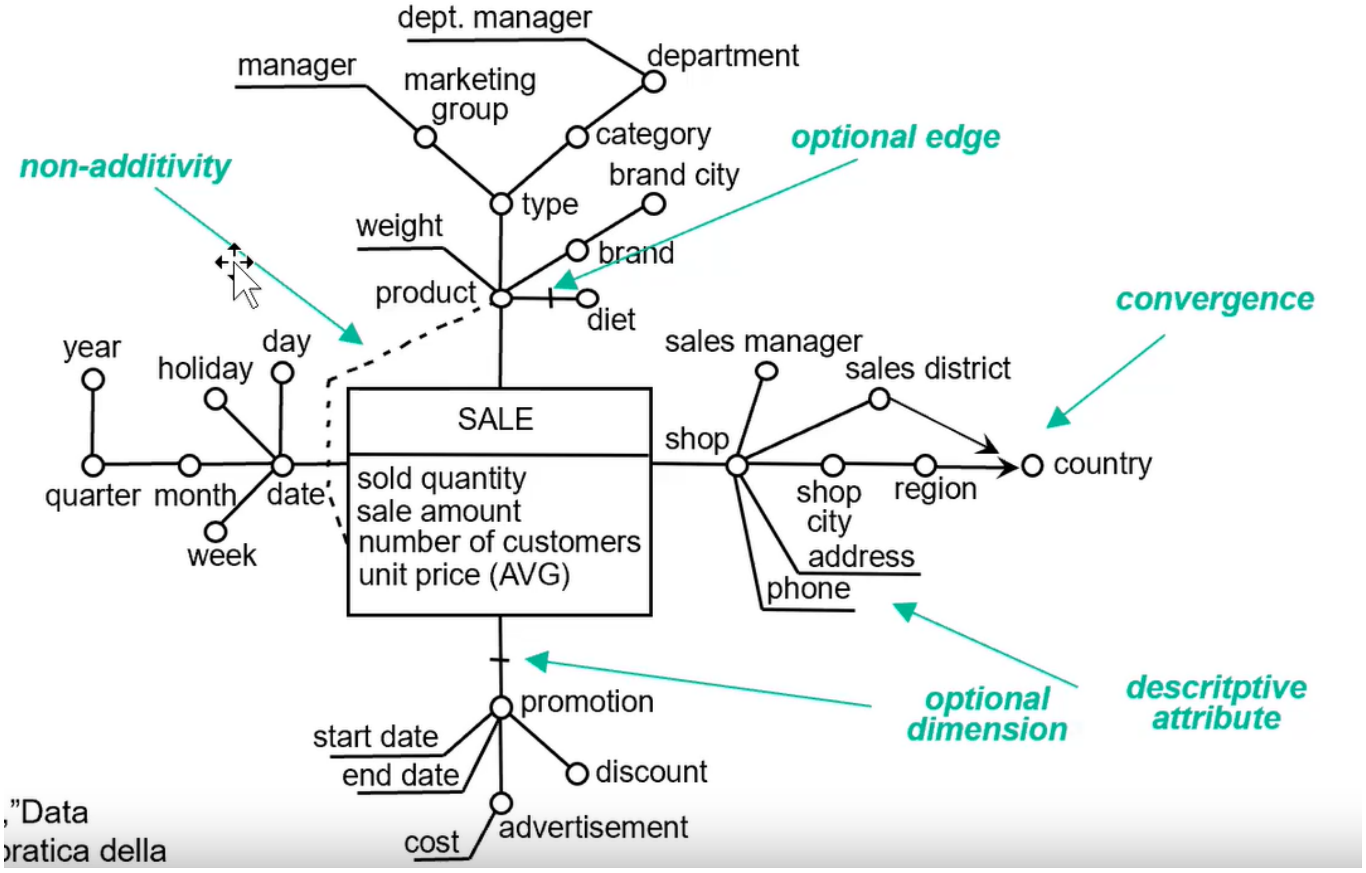


Aggregation

Detail reduction is obtained by climbing the hierarchy.
For example, referring to the DFM above, we can aggregate all **sales** in a given **date** for a given **product**(the very first two nodes in the date and product dimension) in a certain **region**(this is not the first node in the shop dimension). With this aggregation we can compute the numeric measures of the Sale(sold quantity,sale amount etc).
We can also climb multiple dimensions at the same time or not to climb all of them.

Advanced DFM

We can enrich the DFM so we can represent better our requirements.



Descriptive attriute

Attributes not used to perform aggregation operation(it wouldnt make any sense/granularity would be too low).
They are used to describe the dimension.
E.g. : It wouldnt make any sense aggregating by phone-number.

Optional Edge

Optional attribute that is not always available

Optional Dimension

An optional dimension that is not always available

Convergence

Different paths on a hierarchy that converge on the same attribute.

Non-additivity

A measure that cannot be aggregated along the indicated hierarchy by the means of the sum operator.
For example: you can't aggregate the number of customers by the product dimension only.

Measure characteristics and classification

They can be:

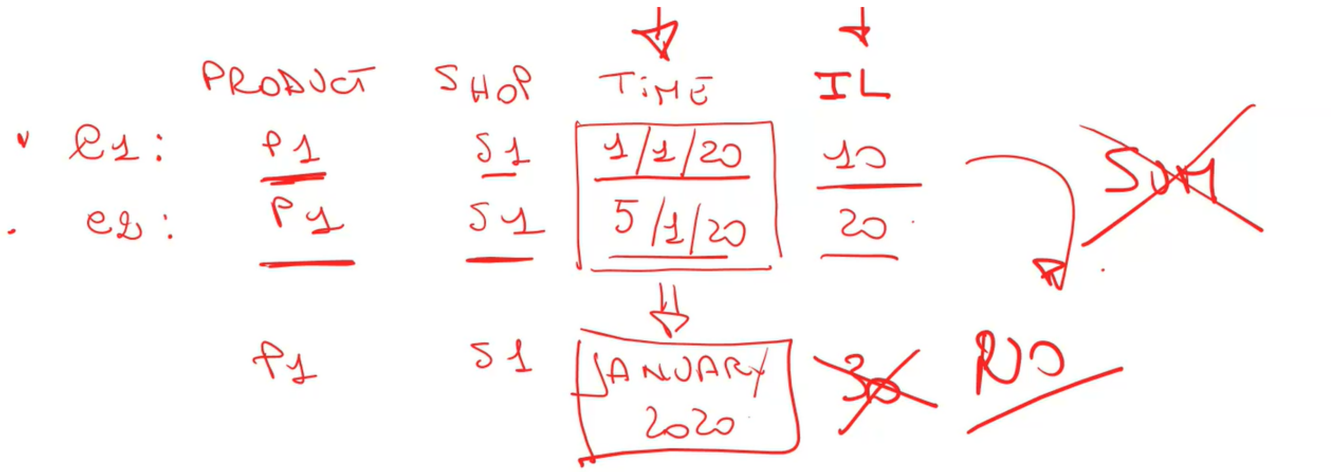
- **additive**
- **non-additive**
- **not aggregable**

Stream Measure

Evaluated cumulatively at the end of the time period and can be aggregated by means of all standard operators. E.g.: sold quantity,sale amount

Level Measure

Evaluated at a given time but it's not additive along the time dimension.
For example: inventory level, account balance.



We can't sum 2 inventory level since it's basically a snapshot of a given time window. We can however compute the max/min of that measure.

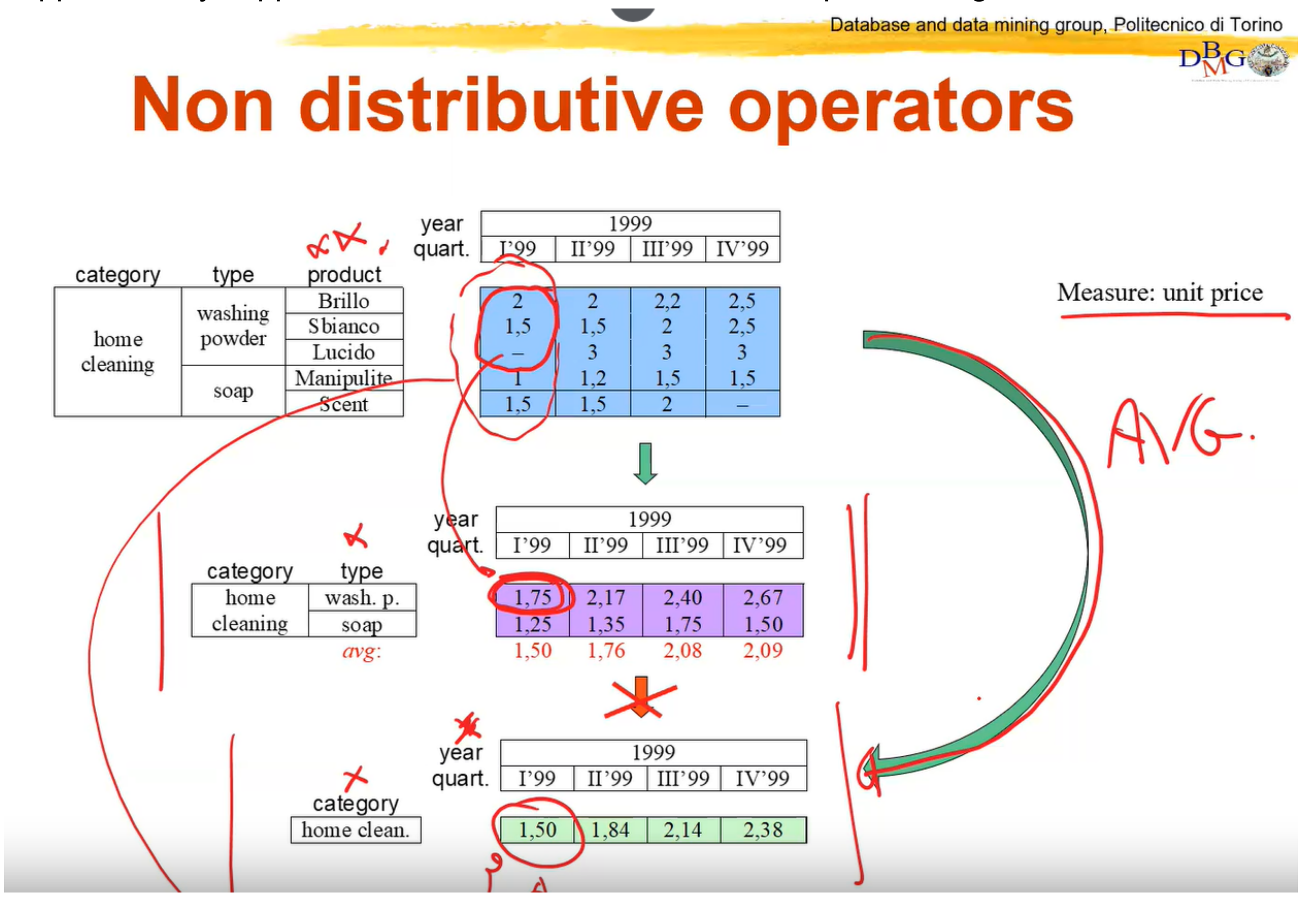
Unit Measure

Evaluated at a given time and expressed in relative terms.
It's not additive along any dimension.
For example, the unit price of a product cannot be aggregated by the means of the sum operator, it wouldnt make any sense.

Aggregate Operators

They can be:

- **distributive:** can always compute higher level aggregations from more detailed data. E.g. : sum,min,max.
- **algebraic:** can compute higher level aggregations from more detailed data onlywhen supplementary support measures are available. For examples average.



Database and data mining group, Politecnico di Torino

Non distributive operators

Measure: unit price

category	type	product	I'99	II'99	III'99	IV'99
			1999			
home cleaning	washing powder	Brillo	2	2	2,2	2,5
		Sbianco	1,5	1,5	2	2,5
		Lucido	–	3	3	3
	soap	Manipulite	1	1,2	1,5	1,5
		Scent	1,5	1,5	2	–

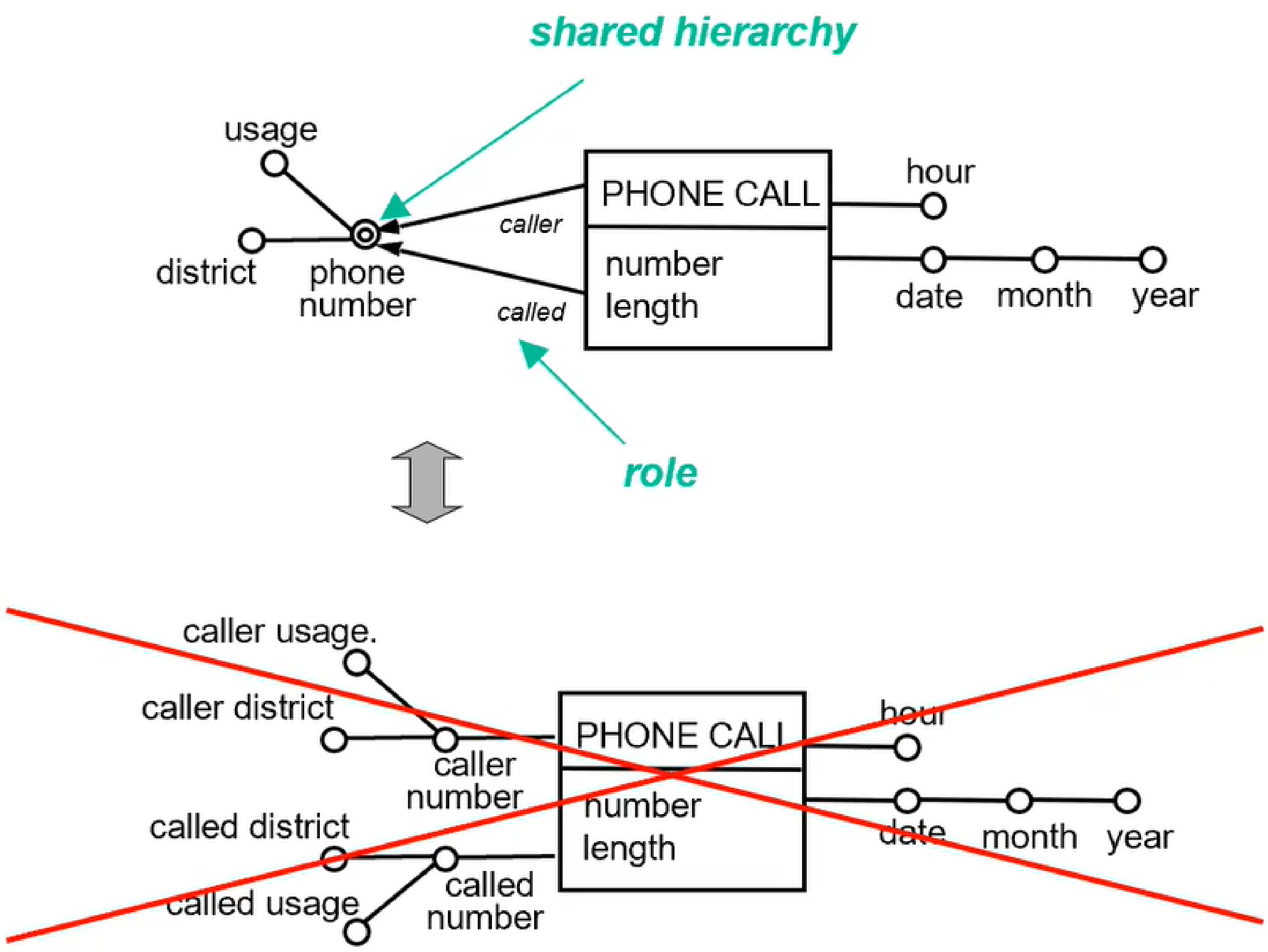
category	type	I'99	II'99	III'99	IV'99
		1999			
home cleaning	wash. p.	1,75	2,17	2,40	2,67
	soap	1,25	1,35	1,75	1,50
	avg:	1,50	1,76	2,08	2,09

category	I'99	II'99	III'99	IV'99
	1999			
home clean.	1,50	1,84	2,14	2,38

From step 2 to step 3, you'd need the number of products for each average so you can properly compute the weigthed mean

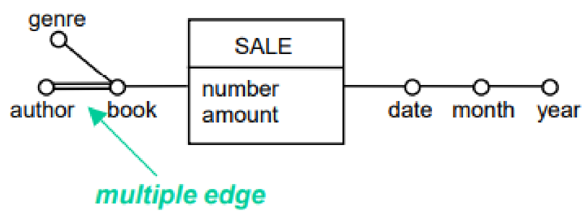
- **olistic:** can't compute higher level aggregations from more detailed data. For examples, mode and median

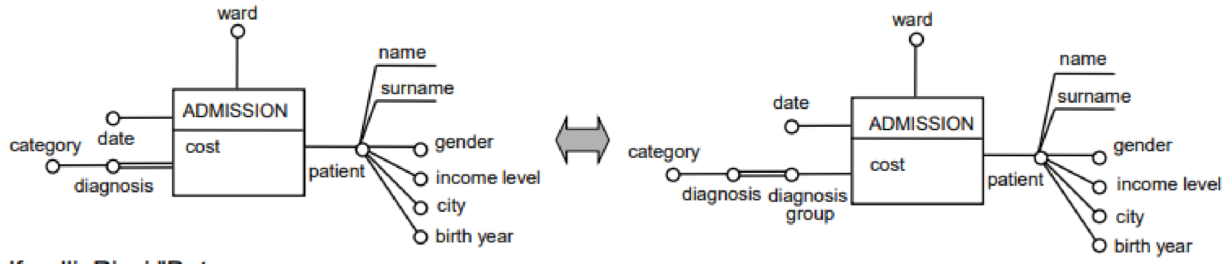
Advanced DFM 2



Shared Hierarchy

Instead of duplicating a dimension, we can **share it**.
The two different dimensions are differentiated by its **role**.





Multiple Edge

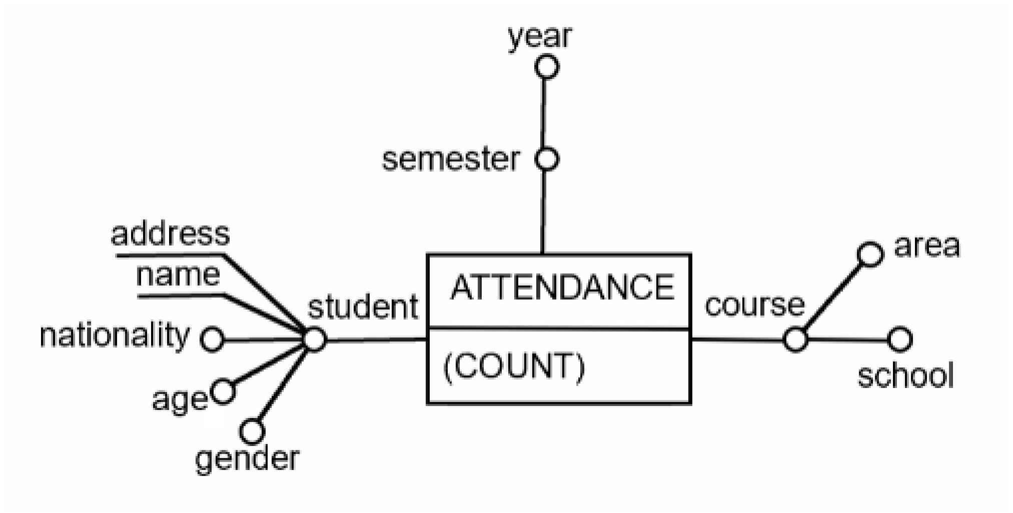
It models a N:N relationship between two attributes.

Configuration Attribute

Multivalued categorical attribute that can assume several values at the same time.
Can be represented as boolean columns.

Factless fact schema

Sometimes we may want to record only the *fact* that the event has happened, without having any measures. It is thus factless since it's empty(no measures).



factless fact schema used to represent the attendance of students to a certain course

Representing time

Let's say we want to represent the sales manager of a shop.
This sales manager may vary in time.

- **Type 1**: a snapshot of the current value is used. Used when we dont really care of the impact the sales manager has had. Also we dont introduce any extra data
- **Type 2:** after each state change in a dimension, a new instance of the dimension is created.
- **Type 3:** new additional informations are added such as *validity_start* and *validity_end* to indicate the validity of the dimension instance. Also a **master** attribute may be used to point the new instance to the root distance.