Methodological framework

Matteo Golfarelli

University of Bologna - Italy

Summary

- Methodological approaches
- Conceptual design
  - ✓ The dimensional fact model
- Logical design
  - ✓ The star schema
  - ✓ Translating a conceptual schema
- Behond data warehousing
  - ✓ Data mining
  - ✓ What-if analysis

Why?

- Building a DW is a very complex task, which requires an accurate planning aimed at devising satisfactory answers to organizational and architectural questions
- A large number of organizations lack the experience and skills required to meet the challenges involved in DW projects
- The reports of DW project failures state that a major cause lies in the absence of a global view of the design process: in other terms, in the absence of a design methodology
- Methodologies are created by closely studying similar experiences and minimizing the risks for failure by basing new approaches on a constructive analysis of the mistakes made previously

Top-down approach

- Analyze global business needs, plan how to develop a data warehouse, design it, and implement it as a whole
  - This procedure is promising: it is based on a global picture of the goal to achieve, and in principle it ensures consistent, well integrated data warehouses
  - High-cost estimates with long-term implementations discourage company managers from embarking on these kind of projects
  - Analyzing and integrating all relevant sources at the same time is a very difficult task, even because it is not very likely that they are all available and stable at the same time
  - It is extremely difficult to forecast the specific needs of every department involved in a project, which can result in the analysis process coming to a standstill
  - Since no working system is going to be delivered in the short term, users cannot check for this project to be useful, so they lose trust and interest in it

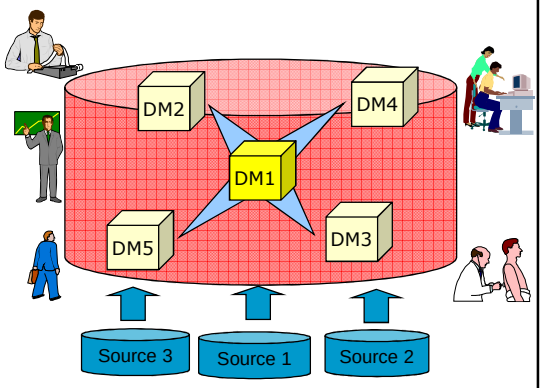
Bottom-up approach

- DWs are incrementally built and several data marts are iteratively created. Each data mart is based on a set of facts that are linked to a specific department and that can be interesting for a user group
  - Leads to concrete results in a short time
  - Does not require huge investments
  - Enables designers to investigate one area at a time
  - Gives managers a quick feedback about the actual benefits of the system being built
  - Keeps the interest for the project constantly high
  - May determine a partial vision of the business domain

5

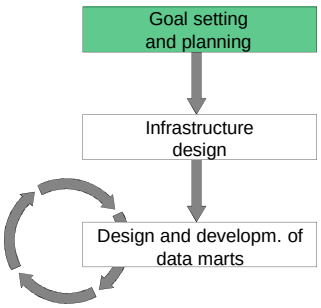
The first data mart to be prototyped...

- ✓ is the one playing the most strategic role for the enterprise
- ✓ should be a backbone for the whole DW
- ✓ should lean on available and consistent data sources



6

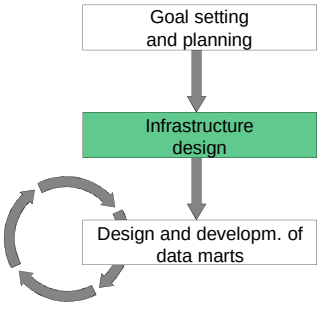
The life-cycle



- set system goals, borders, and size
- select an approach for design and implementation
- estimate costs and benefits
- analyze risks and expectations
- examine the skills of the working team

7

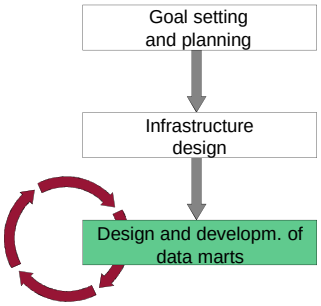
The life-cycle



- analyze and compare the possible architectural solutions
- assess the available technologies and tools
- create a preliminary plan of the whole system

8

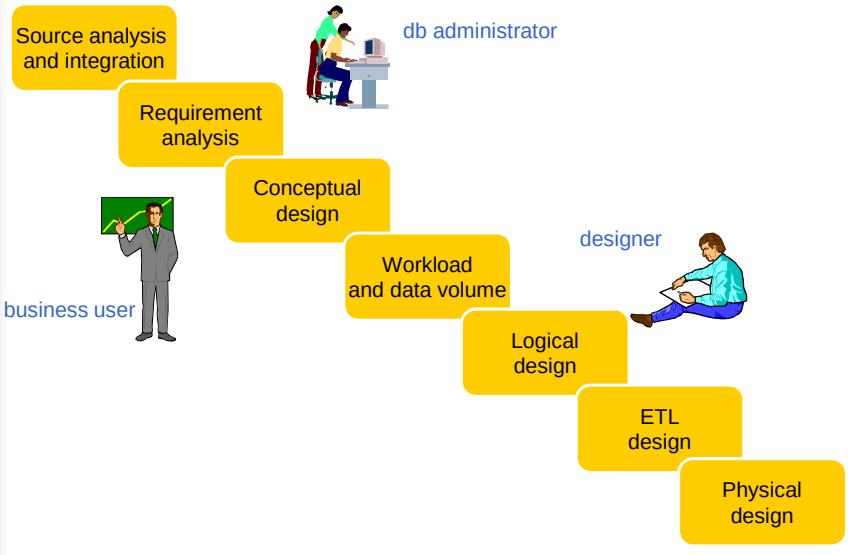
The life-cycle



Every iteration causes a new data mart and new applications to be created and progressively added to the DW system

9

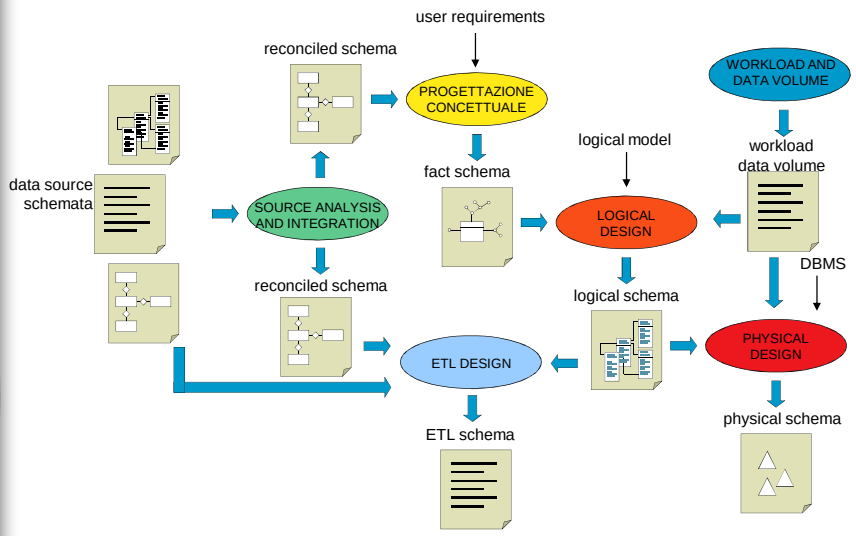
Data mart design phases



Methodological scenarios

- *Supply-driven approach*
  - ✓ they design data marts on the basis of a close operational data source analysis
  - ✓ user requirements show designers which groups of data, relevant for decision-making processes, should be selected and how to define data group structures on the basis of the multidimensional model
- *Demand-driven approach*
  - ✓ they begin with the definition of information requirements of data mart users
  - ✓ the problem of how to map those requirements into existing data sources is addressed at a later stage, when ETL procedures are implemented

Supply-driven approach



Supply-driven approach

- Pros
  - ✓ an initial conceptual schema for data marts can be algorithmically derived from the reconciled layer—that is, it strictly depends on data source structures
  - ✓ ETL design is extremely streamlined because every single information piece stored in a data mart is directly associated with one or more source attributes
  - ✓ the resulting data marts are quite stable in time, because they are rooted in source schemata—that change less frequently than the requirements expressed by end users
- Cons
  - ✓ user requirements play a minor role when it comes to specifying information contents to carry out an analysis
  - ✓ designers have a limited support when facts, dimensions, and measures need to be determined

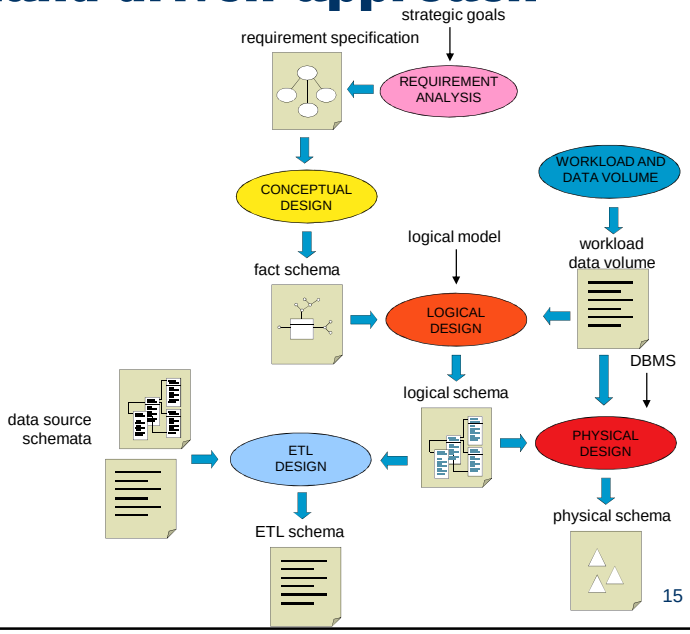
13

Supply-driven approach

- Applicability
  - ✓ It can be applied when:
    1. An in-depth knowledge of data sources to populate data marts is available or it can be achieved at a reasonable price and in the short term
    2. Data source schemata show a good level of normalization
    3. Data source schemata are not too complex
  - ✓ When your architecture includes a reconciled layer, you can amply meet those requirements: normalization and in-depth knowledge are obtained at the end of the reconciliation process. If your data sources are reduced to one single, small, well-designed database, you can achieve the same result after accurately completing the inspection phase.
- Our experience in design shows that the data-driven approach, if applicable, is better than other approaches, because it gives you the opportunity to reach your project goals within a very short time

14

Demand-driven approach



15

Demand-driven approach

- Pros
  - ✓ users' wishes play a leading role
- Cons
  - ✓ designers are required to have strong leadership and meeting facilitation qualities to properly grab and integrate the different points of view
  - ✓ designers make great efforts in the data-staging design phase
  - ✓ facts, measures, and hierarchies are drawn directly from the specifications provided by users, and only at a later stage can designers check for the information required to be actually available in source databases
  - ✓ this may undermine customers' confidence in designers and in the advantage gained by data marts on the whole

16

Demand-driven approach

- Applicability
  - ✓ This is your only alternative if you cannot conduct any preliminary, detailed source analysis (for example, when an ERP system is used to feed your data mart) or if sources are represented by legacy systems with such complexity that it is not recommended that you explore and normalize them, and as a result you do not think it appropriate to create the reconciled layer
- Demand-driven approaches are typically more time-expensive than data-driven approaches, because users often do not have a clear and shared understanding of business goals and processes

17

Conceptual design

Which formalism?

- While it is now universally recognized that a data mart is based on a multidimensional view of data, there is still no agreement on how to implement its conceptual design
- Use of the Entity-Relationship model is quite widespread throughout companies as a conceptual tool for standard documentation and design of relational databases, but *it cannot be used to model DWs*
- In some cases, designers base their data marts design on the logical level—that is, they directly define star schemata that are the standard ROLAP implementation of the multidimensional model. But a star schema is nothing but a relational schema; *it contains only the definition of a set of relations and integrity constraints!*

19

The Dimensional Fact Model

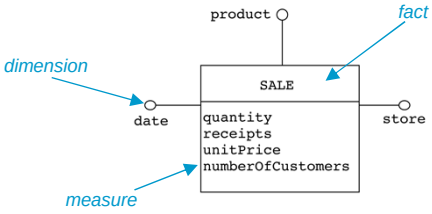
- The DFM is a graphical conceptual model for data mart design, devised to:
  1. lend effective support to conceptual design
  2. create an environment in which user queries may be formulated intuitively
  3. make communication possible between designers and end users with the goal of formalizing requirement specifications
  4. build a stable platform for logical design (*independently of the target logical model*)
  5. provide clear and expressive design documentation
- The conceptual representation generated by the DFM consists of a set of *fact schemata* that basically model facts, measures, dimensions, and hierarchies

20

DFM: basic concepts

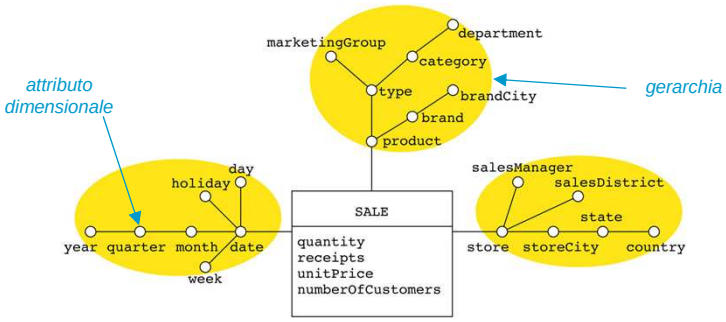
- A *fact* is a concept relevant to decision-making processes. It typically models a set of events taking place within a company (e.g., sales, shipments, purchases, ...). It is essential that a fact have dynamic properties or evolve in some way over time
- A *measure* is a numerical property of a fact and describes a quantitative fact aspect that is relevant to analysis (e.g., every sale is quantified by its receipts)
- A *dimension* is a fact property with a finite domain and describes an analysis coordinate of the fact. Typical dimensions for the sales fact are products, stores, and dates

A fact expresses a many-to-many relationship between its dimensions

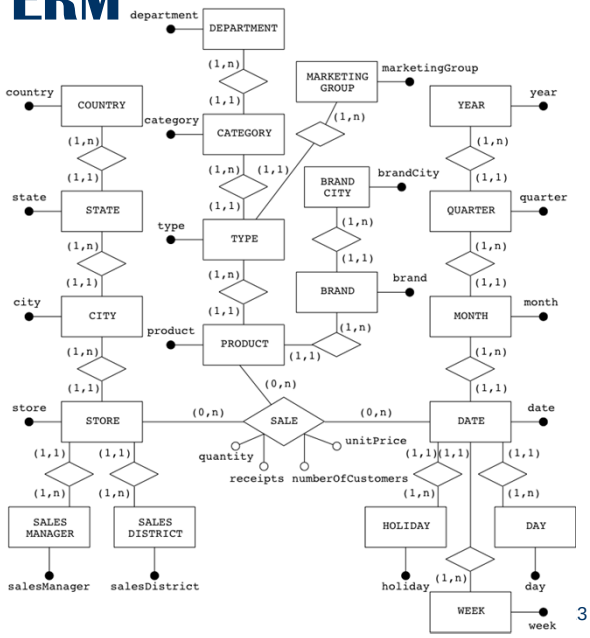


DFM: basic concepts

- The general term *dimensional attributes* stands for the dimensions and other possible attributes, always with discrete values, that describe them (e.g., a product is described by its type, by the category to which it belongs, by its brand, and by the department in which it is sold)
- A *hierarchy* 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



DFM vs. ERM



3

Naming conventions

- All of the attributes and measures within a fact schema must have different names
- You can differentiate similar names, if you qualify them with the name of the dimensional attribute that comes before them in hierarchies
  - ✓ For example, `storeCity` and `brandCity`
- Attributes names should not explicitly refer to the fact they balong to
  - ✓ Avoid `shipped product` and `shipment date`
- Attributes with the same meaning in different fact schemata should have the same name

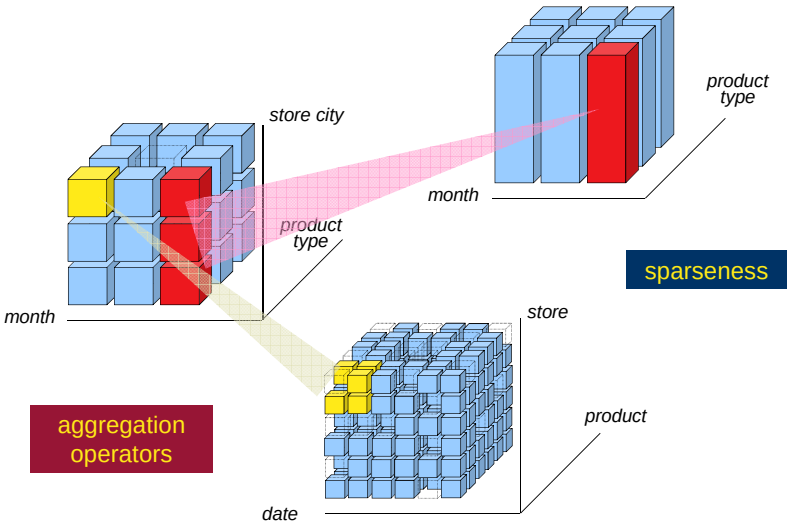
24

Events and aggregation

- A *primary event* is a particular occurrence of a fact, identified by one n-ple made up of a value for each dimension. A value for each measure is associated with each primary event
  - ✓ In reference to the sales example, a possible primary event records that 10 packages of Shiny detergent were sold for total sales of $25 on 10/10/2008 in the SmartMart store
- Given a set of dimensional attributes (group-by set), 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
  - ✓ This makes it possible to use hierarchies to define the way you can aggregate primary events and effectively select them for decision-making processes. While the dimension in which a hierarchy takes root defines its finest aggregation granularity, the other dimensional attributes correspond to a gradually increasing granularity

25

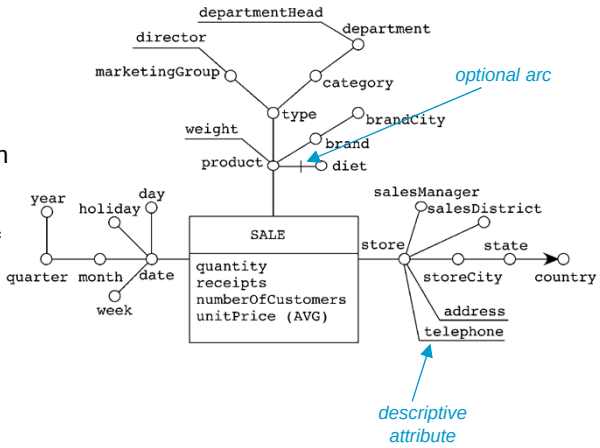
Events and aggregation



26

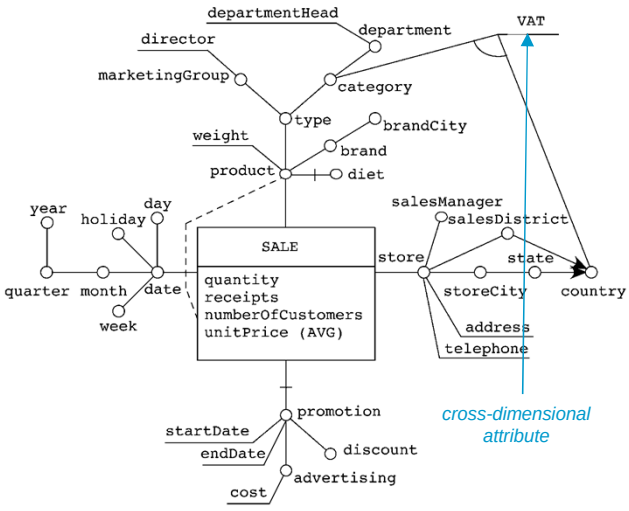
DFM: advanced concepts

- A *descriptive attribute* stores additional information about a dimensional attribute. It is not used for aggregation because it has a dense domain and/or it is a child of a one-to-one association
- Some arcs in a fact schema can be *optional*



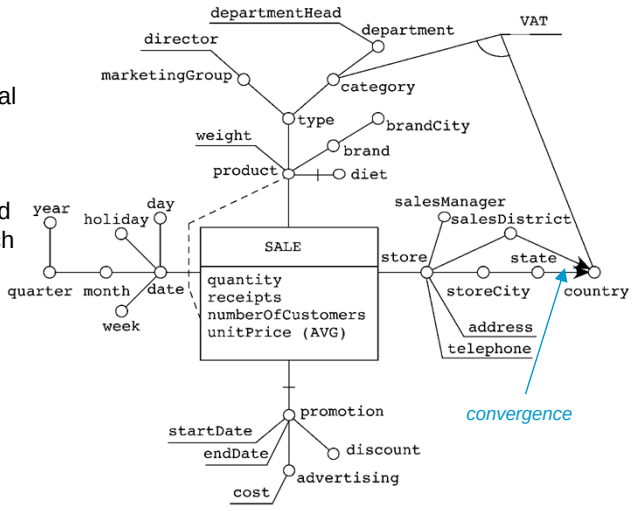
DFM: advanced concepts

- A *cross-dimensional attribute* is a dimensional or descriptive attribute whose value is defined by the combination of two or more dimensional attributes, possibly belonging to different hierarchies



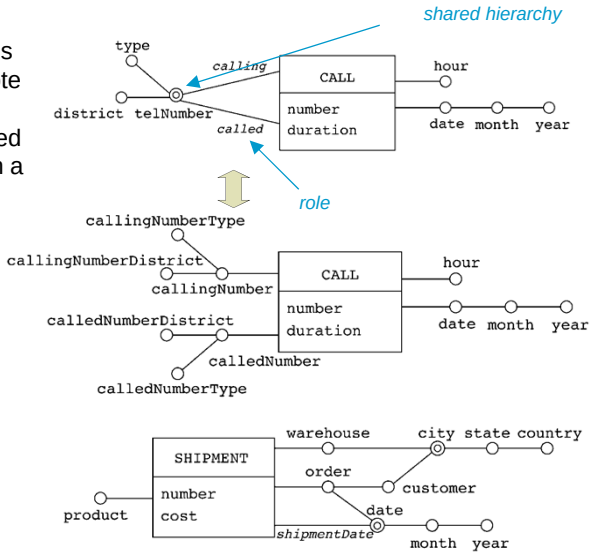
DFM: advanced concepts

- In a *convergence*, two dimensional attributes are connected by two or more distinct directed paths, and each of them still represents a functional dependency



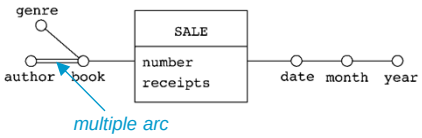
DFM: advanced concepts

- A *shared hierarchy* is a shorthand to denote that a part of a hierarchy is replicated two or more times in a fact schema



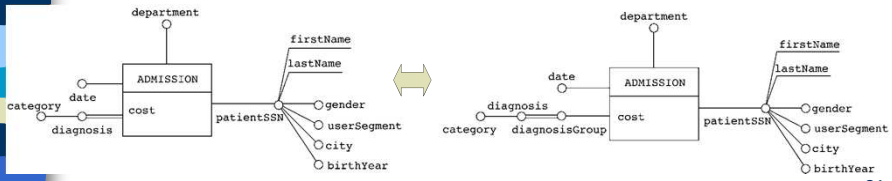
DFM: advanced concepts

- A *multiple arc* models a many-to-many association between two dimensional attributes



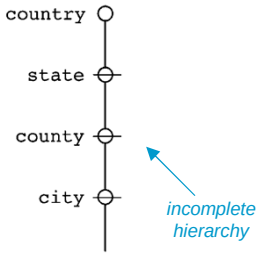
Facts & Crimes	Golfarelli, Rizz	3
Sounds Logical	Golfarelli	5
The Right Measure	Rizzi	10
Facts: How and Why	Golfarelli, Rizz	4
The Fourth Dimension	Golfarelli	8

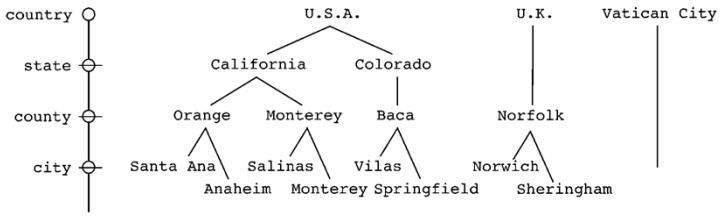
How much did Rizzi sell?



DFM: advanced concepts

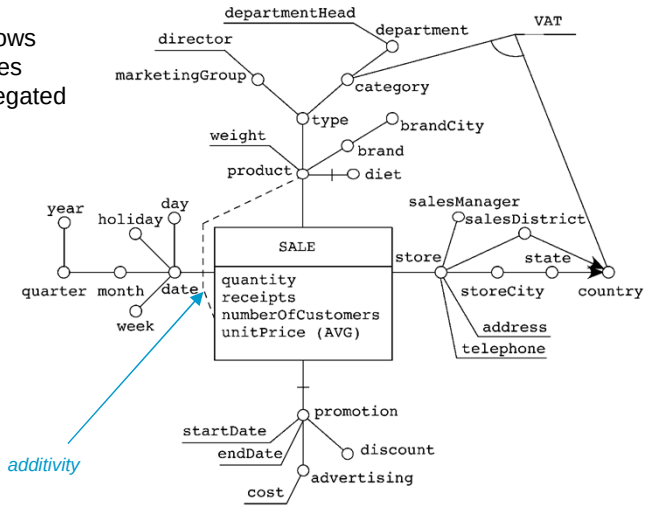
- An *incomplete hierarchy* is a hierarchy where, for some instances, one ore more aggregation levels are missing (because they are unknown or undefined)





DFM: advanced concepts

- *Additivity* shows how measures can be aggregated



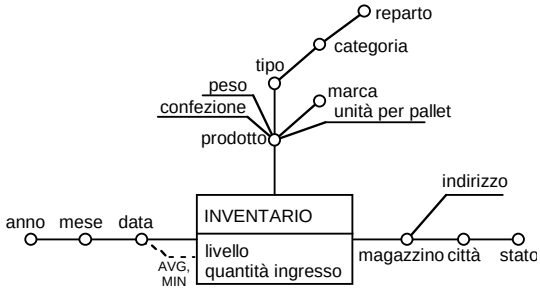
Additivity

- 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
- From this viewpoint, measures can be classified into three categories:
  - ✓ *Flow Measures* refer to a timeframe, at the end of which they are evaluated cumulatively (the number of products sold in a day, monthly receipts, yearly number of births)
  - ✓ *Level Measures* are evaluated at particular times (the number of products in inventory, the number of inhabitants in a city)
  - ✓ *Unit Measures* are evaluated at particular times but are expressed in relative terms (product unit price, discount percentage, currency exchange)

	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

Additivity

- 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



Additive measures

category	type	product
House cleaning	Cleaner	Shiny
		Bleachy
		Brighty
	Soap	Cleanhand
		Scent
Food	Dairy product	F Slurp Milk
		U Slurp Milk
		Slurp Yogurt
	Drink	DrinkMe
		Coky

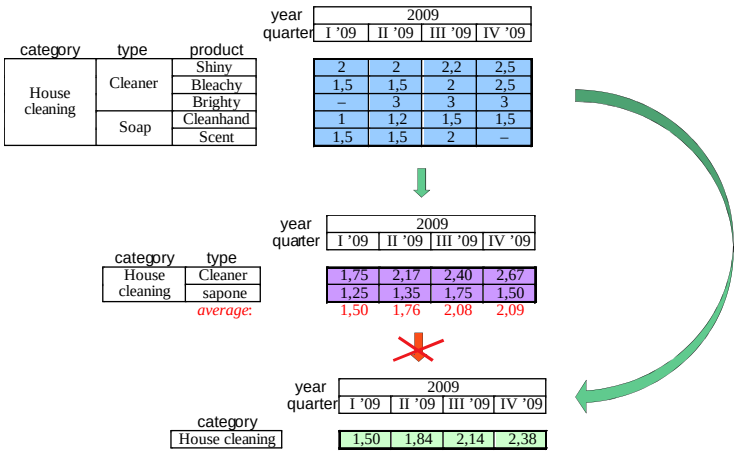
year	2009				2010			
quarter	I '09	II '09	III '09	IV '09	I '10	II '10	III '00	IV '10
	100	90	95	90	80	70	90	85
	20	30	20	10	25	30	35	20
	60	50	60	45	40	40	50	40
	15	20	25	30	15	15	20	10
	30	35	20	25	30	30	20	15
	90	90	85	75	60	80	85	60
	60	80	85	60	70	70	75	65
	20	30	40	35	30	35	35	20
	20	10	25	30	35	30	20	10
	50	60	45	40	50	60	45	40

year	2009				2010			
quarter	I'09	II'09	III'09	IV'09	I'10	II'10	III'10	IV'10
category								
House cleaning	225	225	220	200	190	185	215	170
Food	240	270	280	240	245	275	260	195

year	2009	2010
category		
House cleaning	870	760
Food	1030	975

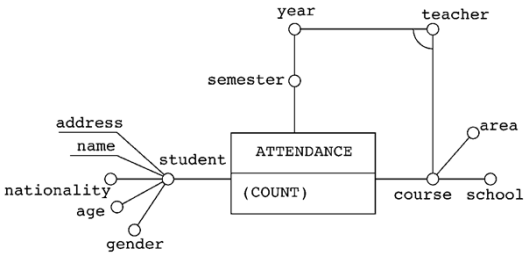
category	type	year 2009	year 2010
House cleaning	Cleaner	670	605
	Soap	200	155
Food	Dairy product	750	685
	Drink	280	290

Non-additive measures



Empty fact schema

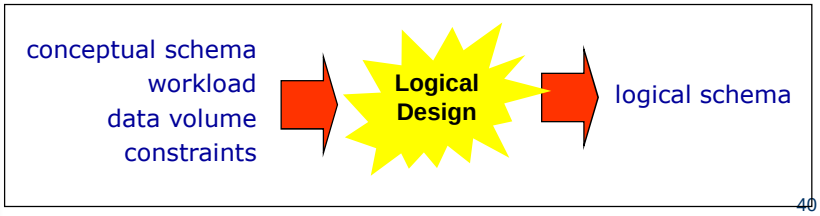
- A fact schema is said to be *empty* if it does not have any measures
  - ✓ primary events only record the *occurrence* of events in an application domain



Logical design

Logical design

- A phase aimed at determining a logical schema for the data mart starting from a conceptual schema
  - ✓ Choice of the *type* of logical schema
  - ✓ Translation of conceptual schemata
  - ✓ Optimization (view materialization, fragmentation)
- It is based on different principles from those used in operational databases
  - ✓ data redundancy
  - ✓ denormalization of relations



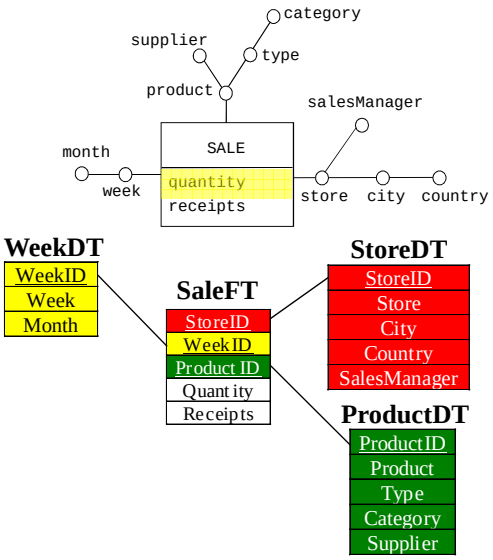
40

The star scheme

- Multidimensional modeling in relational systems is based on the so-called *star schema* and on its variants
- A star schema consists of the following:
  - ✓ A set of *dimension tables* (DT_1…, DT_n), each corresponding to a dimension. Every DT_i has a primary (typically surrogate) key (k_i) and a set of attributes at different aggregation levels
  - ✓ A *fact table* (*FT*) including measures. An *FT* primary key is the composition of the set of foreign keys (k_1 through k_n) referencing dimension tables
- Dimension tables are not in 3rd normal form because transitive functional dependencies exist due to the presence of all the attributes of a hierarchy in the same relation
  - ✓ There is some redundancy
  - ✓ The number of joins needed to retrieve information is reduced

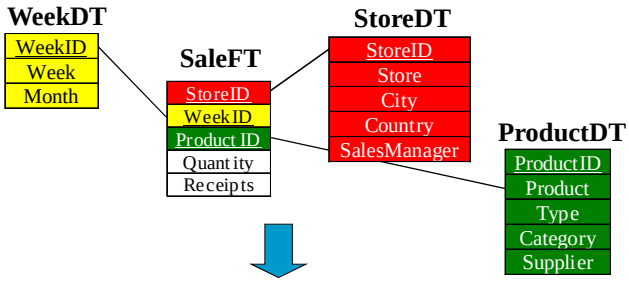
41

The star scheme



42

OLAP queries on a star schema

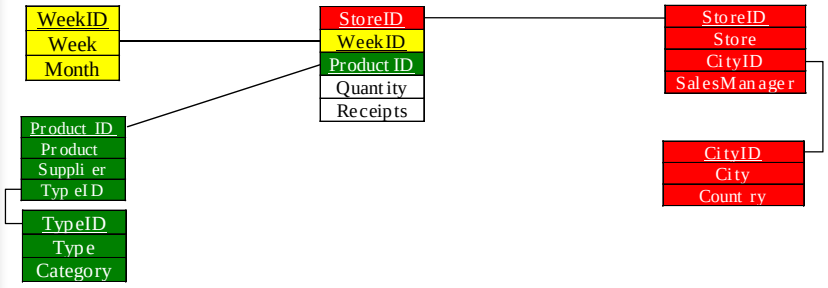


Total quantity sold for each product type, week, and city, only for food products

```
SELECT    City, Week, Type, SUM(Quantity)
FROM      WeekDT, StoreDT, ProductDT, SaleFT
WHERE     WeekDT.WeekID = SaleFT.WeekID AND
          StoreDT.StoreID = SaleFT.StoreID AND
          ProductDT.ProductID = SaleFT.ProductID AND
          ProductDT.Category = 'Food'
GROUP BY  City, Week, Type;
```


The snowflake scheme

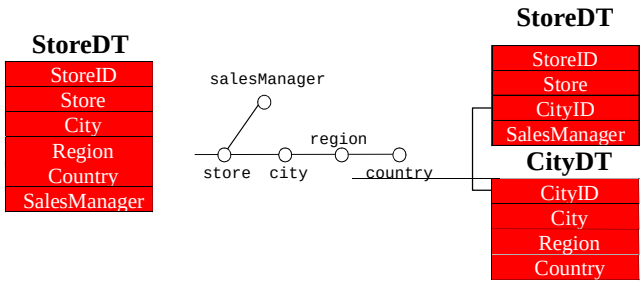
- The star schema can be optimized in terms of space if one or more dimensions are normalized



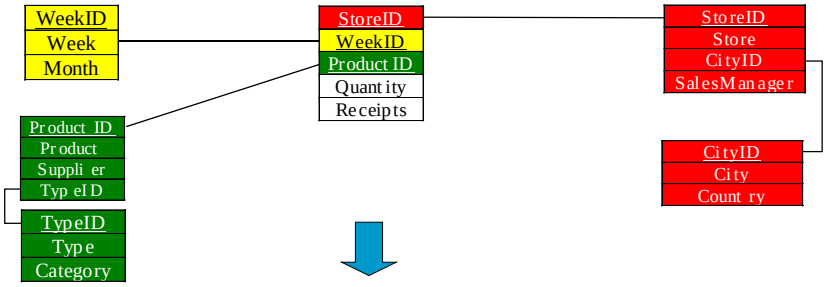
☺ DT size is decreased
☺ The number of joins to retrieve data may be higher

The snowflake scheme

- To break down a schema effectively, you need for all those attributes that—directly or transitively—depend on the snowflaking attribute (that is, on the natural key of the new relation) to be part of the new relation



OLAP queries on a snowflake schema

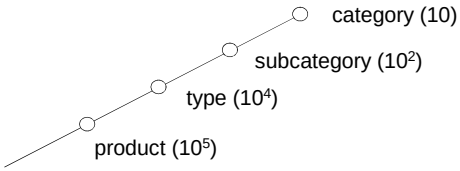


Total quantity sold for each product type, week, and city, only for food products

```
SELECT    City, Week, Type, SUM(Quantity)
FROM      WeekDT, StoreDT, ProductDT, CityDT, TypeDT, SaleFT
WHERE     WeekDT.WeekID = SaleFT.WeekID AND
          StoreDT.StoreID = SaleFT.StoreID AND
          ProductDT.ProductID = SaleFT.ProductID AND
          StoreDT.CityID = CityDT.CityID AND
          ProductDT.TypeID = TypeDT.TypeID AND
          ProductDT.Category = 'Food'
GROUP BY  City, Week, Type;
```

Star vs. snowflake

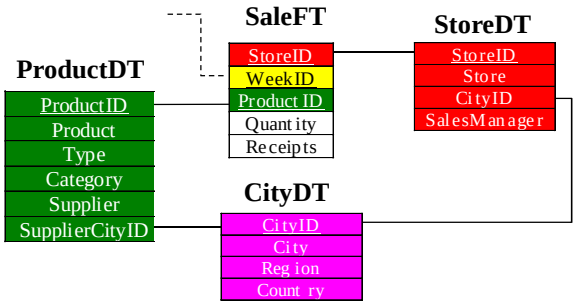
- Snowflaking may be useful:
  - ✓ When the ratio between the cardinalities of the primary and secondary DTs is high, because in this case it leads to a relevant space saving



47

Star vs. snowflake

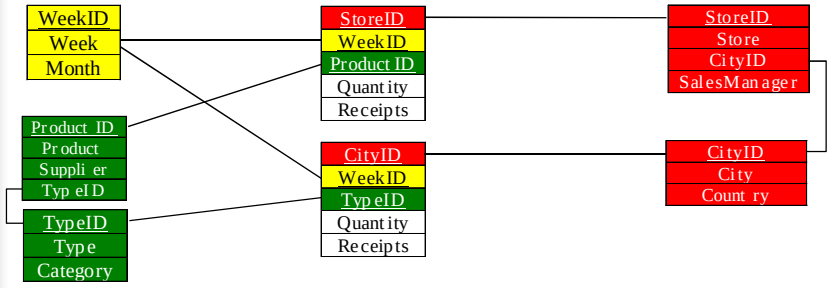
- Snowflaking may be useful:
  - ✓ When some section of a hierarchy is shared



48

Star vs. snowflake

- Snowflaking may be useful:
  - ✓ In presence of aggregate views



49

Translating conceptual schemata

- The basic rule for translating a fact schema into a star schema is:

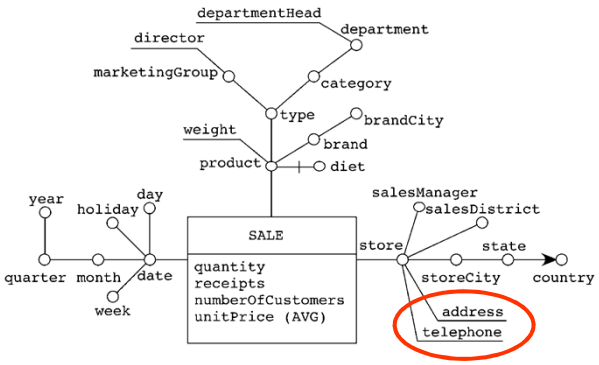
  Create a fact table including all measures; for each hierarchy, create a dimension table including all attributes

- Besides this obvious rule, specific solutions must be taken for different constructs...

50

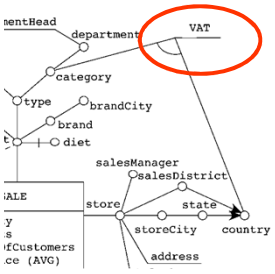
1) Descriptive attributes

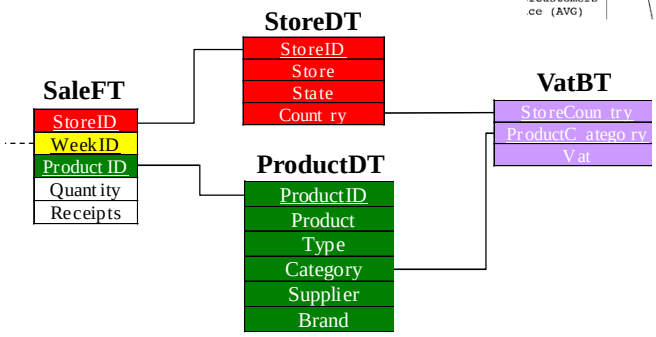
- A descriptive attribute stores information that cannot be used for aggregation
  - ✓ It must be included in the dimension table of the dimensional attribute it is connected to



2) Cross-dimensional attributes

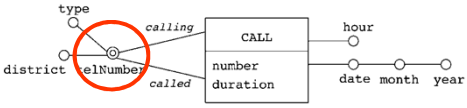
- A cross-dimensional attribute *b* defines a many-to-many association between two or more dimensional attributes a_1..., a_m
- Translating it to the star schema requires to create a new table including *b* and having a_1..., a_m as the key

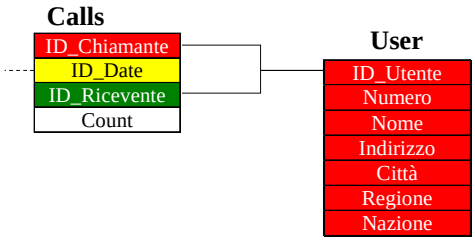




3) Shared hierarchies

- If a hierarchy appears twice or more in the same fact, it is not worth to create redundant copies of DTs
  - ✓ If the two hierarchies exactly contain the same attributes, it is sufficient to put two foreign keys in the same dimension table

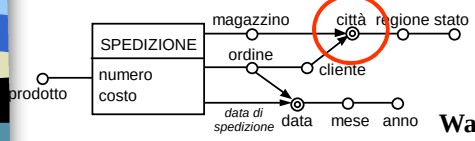


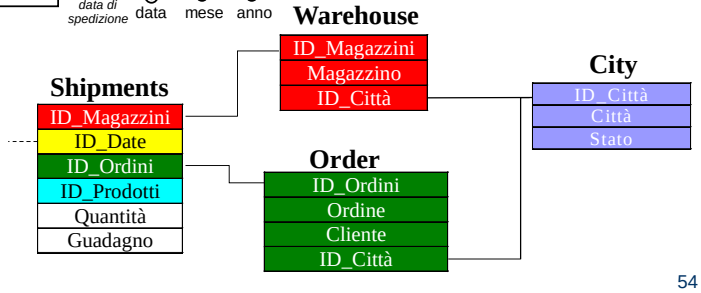


53

3) Shared hierarchies

- If the two hierarchies share only a part of the attributes wehave two choices:
  - I. Introduce further redundancy by duplicating hierarchies and the common attributes
  - II. Snowflake on the first common attribute

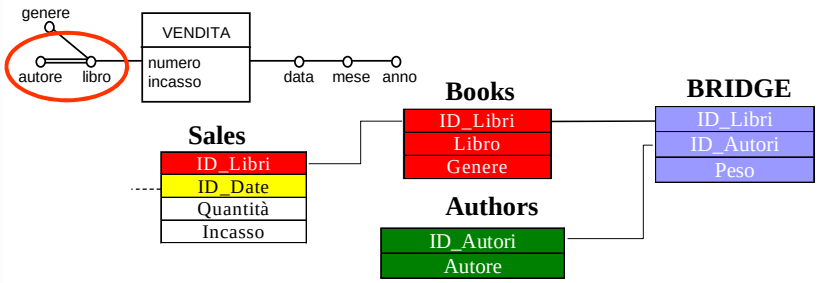




54

4) Multiple arcs

- Sometimes, a hierarchy may include many-to-many associations
- The most obvious solution is to create an additional table (*bridge table*) to model the multiple arc:
  - ✓ The key of the bridge table is the combination of the attributes connected by the multiple arc
  - ✓ A *weight* attribute allows to give different relevance to the tuples



4) Multiple arcs

- Up to 3 joins may become necessary to query the hierarchy
- Two kinds of queries are supported:
  - ✓ Weighted queries: the weight of the multiple arc is considered, so the real total is returned

Revenue for each author

```
SELECT  AUTORI.Autore, sum(VENDITE.Incasso * BRIDGE.Peso)
FROM    AUTORI, BRIDGE, LIBRI, VENDITE
WHERE   AUTORI.ID_Autori = BRIDGE.ID_Autori
AND     BRIDGE.ID_Libri = LIBRI.ID_Libri
AND     LIBRI.ID_Libri = VENDITE.ID_Libri
GROUP BY        AUTORI.Autore
```

4) Multiple arcs

- Up to 3 joins may become necessary to query the hierarchy
- Two kinds of queries are supported:
  - ✓ Impact queries: the weight is not considered, so higher values are returned

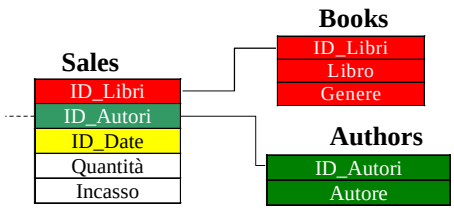
Copies sold for each author

```
SELECT  AUTORI.Autore, sum(VENDITE.Quantità)
FROM    AUTORI, BRIDGE, LIBRI, VENDITE
WHERE   AUTORI.ID_Autori = BRIDGE.ID_Autori
AND     BRIDGE.ID_Libri = LIBRI.ID_Libri
AND     LIBRI.ID_Libri = VENDITE.ID_Libri
GROUP BY        AUTORI.Autore
```

57

4) Multiple arcs

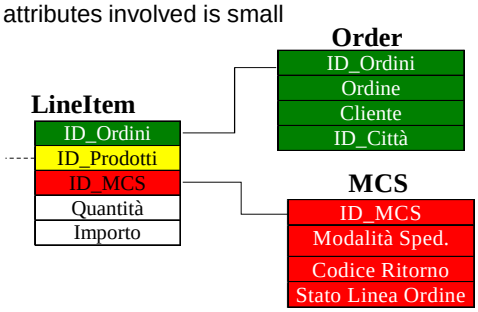
- If we want to use a “clean” star schema, we must make the fact granularity finer, so that the multiple arc is directly modeled in the FT
- This solution requires to add a new dimension to the FT



58

5) Degenerate dimensions

- This term refers to a hierarchy including only one attribute
  - ✓ If the attribute is not too long, its values can be directly included in the FT
- Alternatively, we may use one DT to model several degenerate dimensions (*junk dimension*)
  - ✓ Within a junk dimension there is no functional dependency between attributes, so all combination of values are valid
  - ✓ This solution is feasible only if the number of distinct values for the attributes involved is small



59

Beyond Data Warehouse

Data mining

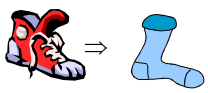
- Data mining is the process of automatically discovering useful information in large data repositories.
  - ✓ When huge datasets are involved a user could not be able to manually discover all the useful pattern
  - ✓ Data mining exploits techniques in the area of artificial intelligence and pattern recognition to help the user in discovering new patterns: the user is asked to describe what she is looking for
    - Market research
    - Marketing effectiveness analysis
    - Market segmentation
    - Market basket analysis
    - Planning
    - Investment modeling
    - Fraud detection
    - Risk analysis
    - Similarity search in event sequences
    - Outlier detection

Data mining: regole associative

- Allows hidden relationships (i.e. logical implication) to be discovered. Relationships point out groups of affine objects
- Applications:
  - ✓ *market-basket analysis*: useful for an effective positioning of products on shelfs
  - ✓ Analysis of sale when a product is not available

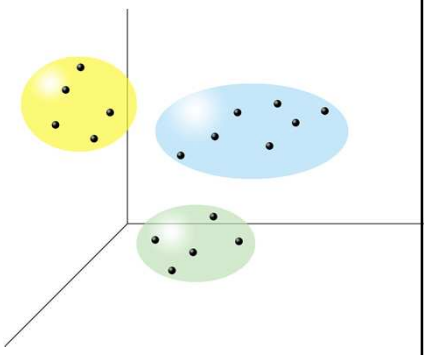
$\{shoes\} \Rightarrow \{socks\}$

support=70%
confidence=85%



Data mining: clustering

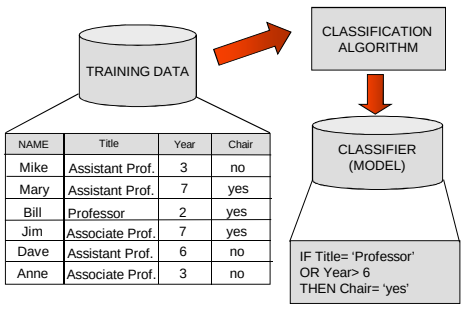
- Cluster analysis groups data objects based on information found in the data that describes the objects and their relationships. Typically objects are represented as points in a multi-dimensional space where each dimension corresponds to an object feature,
- Applications:
  - ✓ Customers segmentation
  - ✓ Emidemiological analyis



63

Data mining: classification

- Given a set of classes and a set of obects labeled according to the class they behave (training set), find a profile for each class using the features in the training set, so that other unlabeled objects (test set) can be properly assigned to a class
- Applications:
  - ✓ Market tendency
  - ✓ Profiling of customer risk classes for insurances and bank loan
  - ✓ Effectiveness of cures



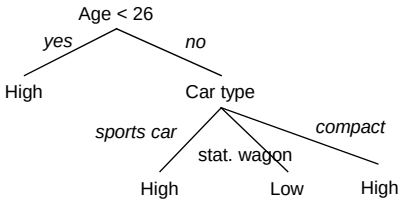
NAME	Title	Year	Chair
Mike	Assistant Prof.	3	no
Mary	Assistant Prof.	7	yes
Bill	Professor	2	yes
Jim	Associate Prof.	7	yes
Dave	Assistant Prof.	6	no
Anne	Associate Prof.	3	no

64

Data mining: decision tree

- A decision tree is a classifier that allows to order, according to their relevance, the causes determining a given event
- Applications:
  - ✓ Risk class detection for customer of insurances and banks

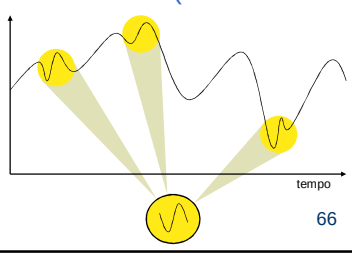
AGE	CAR TYPE	RISK
40	stat. wagon	low
65	sports car	high
20	compact	high
25	sports car	high
50	stat. wagon	low



65

Data mining: time series

- Look for recurrent or atypical patterns in sequential data
- Applications:
  - ✓ Search of schemata related to trends in stocks
  - ✓ Detect anomalies in a monitoring system
  - ✓ Study the correlation between different temporal series
  - ✓ Find out companies with a similar behavior
  - ✓ Analyze similar navigation paths in a web site (click stream analysis)



66

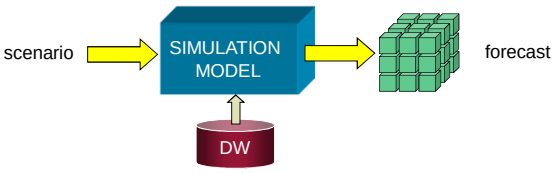
What-if analysis

- Understanding in advance the effects of a choice requires reliable previsional systems
- DWes support analyes of past data but are not able to forcast future trends

67

What-if analysis

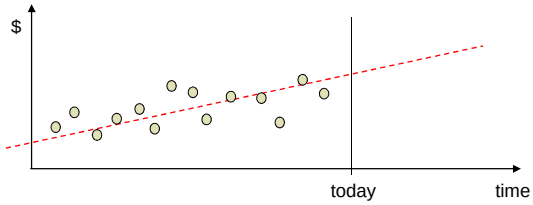
- What-if analysis is a data-intensive approach for studying the behavior of a complex system (the company or part of it) given a set of hypothesis (*scenario*)
- What-if analysis measures how a set of independent variable impacts on the values of a set of dependent ones based on a simulation model; the model is a simplification of the business model based on the historical data stored in the DW
  - ✓ For example in the marketing domain a what-if query could be: "how does profit change if I make a 3X2 promotion for a week on some of the products on sales?"



68

What-if analysis: forecasting

- Largely used in banks and insurance
- It is obtained by computing trends starting from time series stored in the information system
- Is based on statistical techniques: regression and interpolation

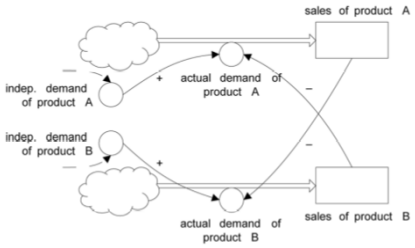


The price of a given product has a non monotonic behavior, but the long term trend can be approximated by a costant rise

69

What-if analysis: system dynamics

- Is an approach for to understanding the behaviour of complex systems over time.
- Cause-effects relationships are captured by dependencies between numerical variables; such dependencies can determine feedback loops
- From the mathematical point of view, such systems are often modeled through differential equations solved using numerical techniques



70