UNIVERSITÉ FRANÇOIS - RABELAIS TOURS

Master SIAD

Data Warehousing and OLAP

Conceptual Design

Verónika Peralta

November 2008

Plan

- **Motivation**
  - The importance of conceptual models
  - Proposals for DW design
- **Requirement-based design**
  - CMDM model
  - Methodological Aspects
- **Data-based design**
  - DF model
  - Methodological Aspects
- **Translation to logical models**
- **Conclusions**

Data Models

- **Definition:**
  - They are languages used for specifying data
- **A data model allows to specify:**
  - Structures:
    - The “objects” of the problem
    - Ex. COURSES (id_course, name, hours).
  - Constraints:
    - Rules to be satisfied by data
    - Ex. forall c in COURSES . c.hours < 80
  - Operations:
    - Allows data manipulation
    - Ex. insert into COURSES (303,“DB”,90)

Classification of Data Models

- **Conceptual models:**
  - Oriented to represent real world entities and their relationships
  - Independent of implementation issues
  - Used at Analysis phase
  - Ex.: entities, relationships
- **Logical models:**
  - Oriented to express abstract data manipulations in a implementable way
  - Organize data in a way it can be managed efficiently by a DBMS
  - Independent of physical storage
  - Used at Design and Implementation phases
  - Ex.: tables, indexes
- **Physical models:**
  - Oriented to express physical implementation
  - Used at Implementation phase
  - Ex.: b-trees, hashes

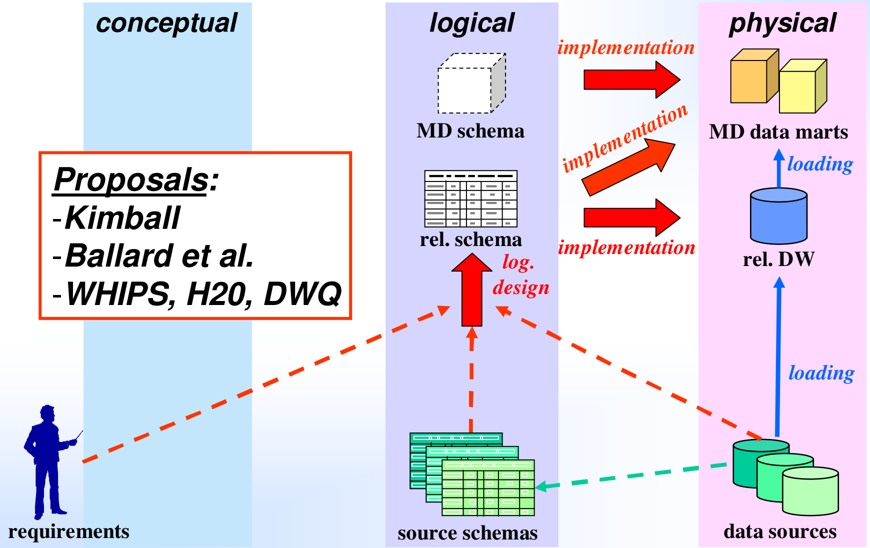
Importance of Conceptual Models

- **An accurate conceptual design is the necessary foundation for building an IS which is both well-documented and fully responding to requirements**
  - Ex. the E/R model is widespread in the enterprises as a conceptual formalism to provide standard documentation for relational IS
- **What happens if we directly design an IS using a logical model ?**

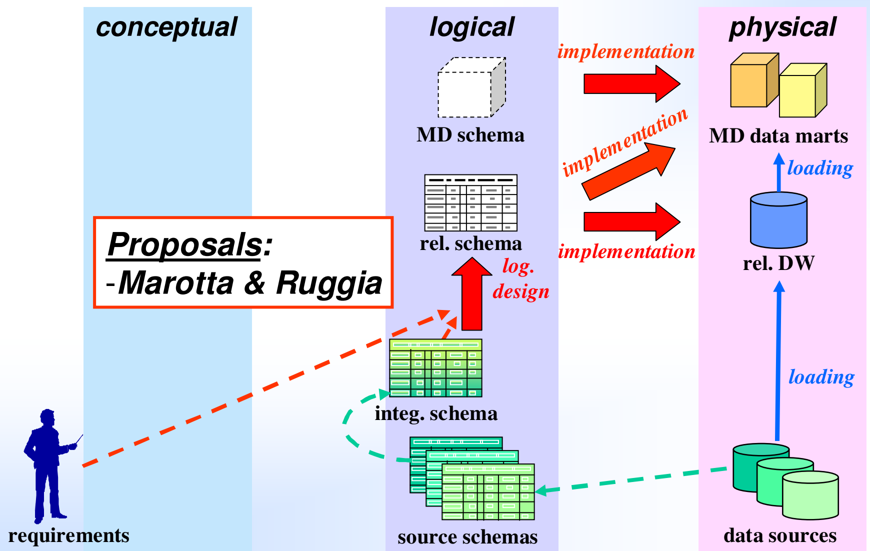
Data Models

	Focus	Record models	Multidimensional models
Conceptual	Objects and relationships	E/R	- No standards - Ex. CMDM / DF
Logical	Operations	Relational	- MD proprietary models - Star schema and variants
Physical	Storage	Proprietary of each DBMSs	Proprietary of each DBMSs

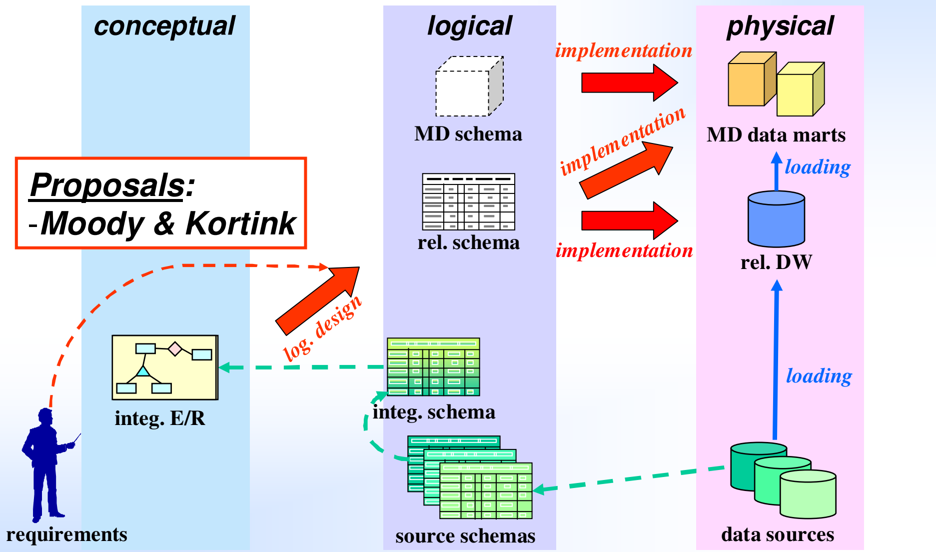
Proposals for DW Design



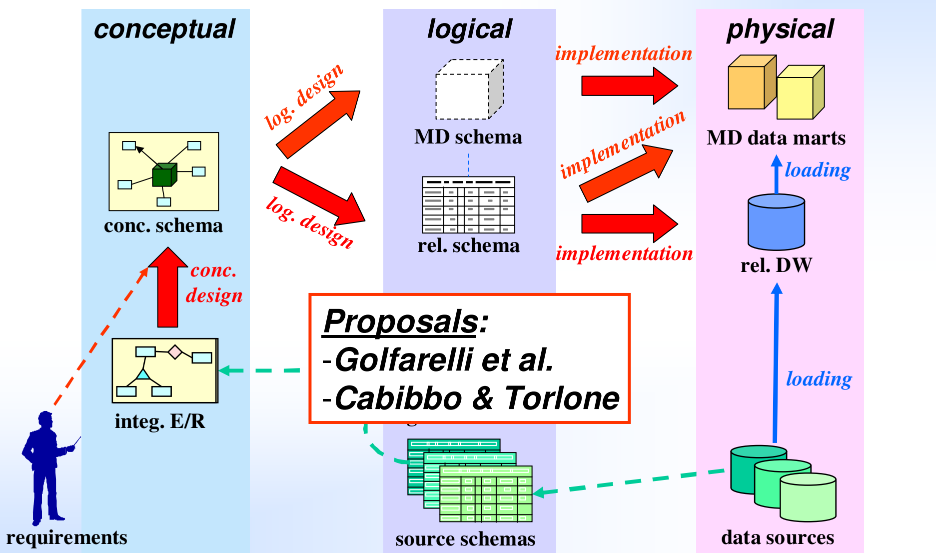
Proposals for DW Design



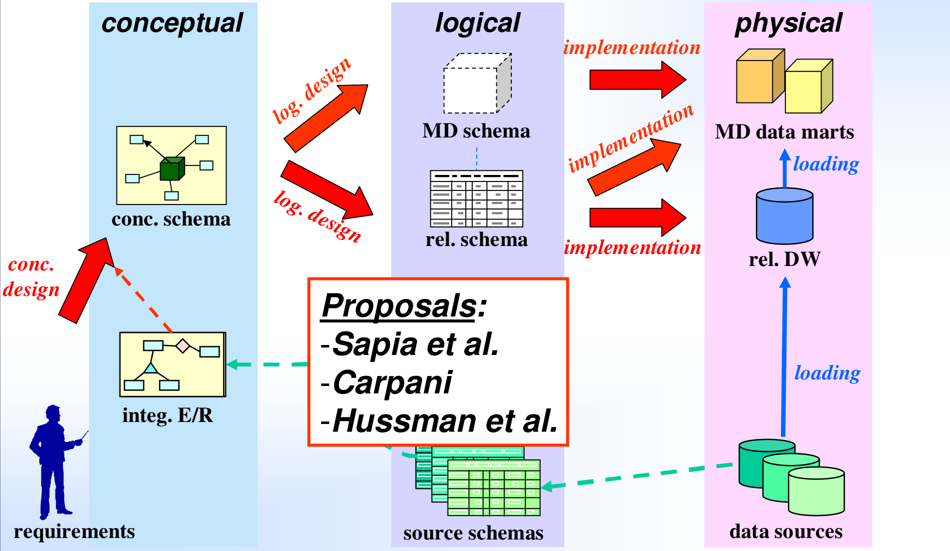
Proposals for DW Design



Proposals for DW Design

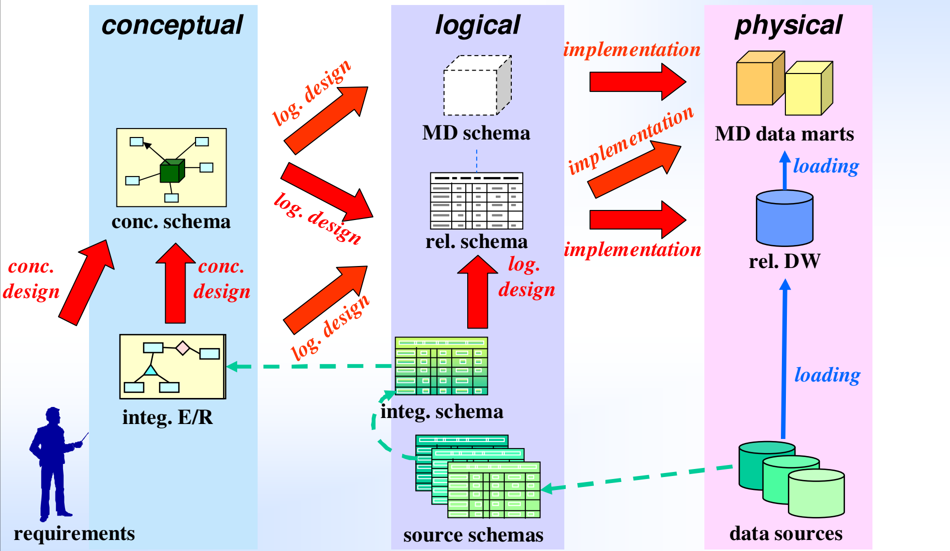


Proposals for DW Design



***Proposals*:**
- ***Sapia et al.***
- ***Carpani***
- ***Hussman et al.***

Proposals for DW Design



DW Conceptual Models

- **Main goals:**
  - Direct, expressive and precise representation of MD schemas
  - Based on accepted MD concepts
  - Independent of implementation
- **Must represent:**
  - MD structures: dimensions, measures, cubes… schemas
  - MD operations: slice, dice, rotation, drill-down, drill-up, roll-up, drill-across, drill-through…

Conceptual Design Approaches

- **Requirement-based design**
  - Requirements are the information universe
  - Data sources are related later
  - Applicable for complex sources
    - They can be analyzed knowing requirements
  - Works: Carpani et al., Sapia et al., Husmann et al., Franconi et al.
- **Data-based design**
  - Data sources are the information universe
  - DW results from transforming data sources
  - Applicable when requirements are not clear enough
  - Works: Golfarelli et al., Cabibbo et al.

Plan

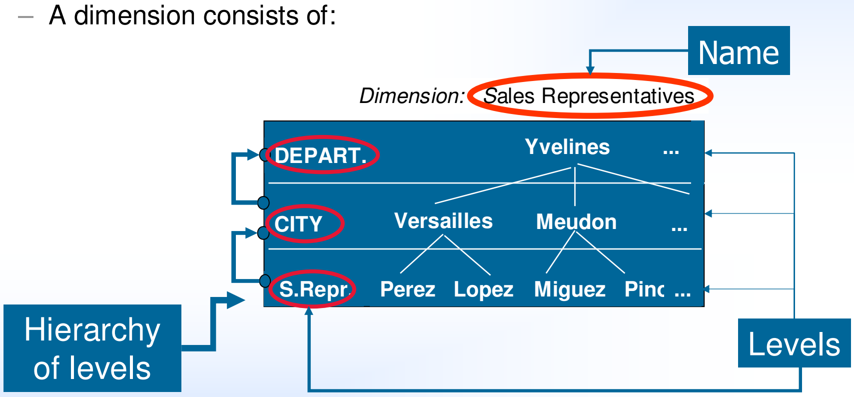
- Motivation
  - The importance of conceptual models
- Requirement-based design
  - CMDM model
  - Methodological Aspects
- Data-based design
  - DF model
  - Methodological Aspects
- Translation to logical models
- Conclusions

CMDM Model

- Basic structures:
  - Levels
  - Dimensions
    - Hierarchies of levels
    - Include measures (Generic dimensionality)
  - Dimensional relations
  - Cubes
    - Specific crossings of dimensional relations

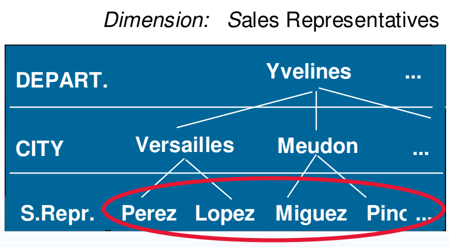
CMDM Model

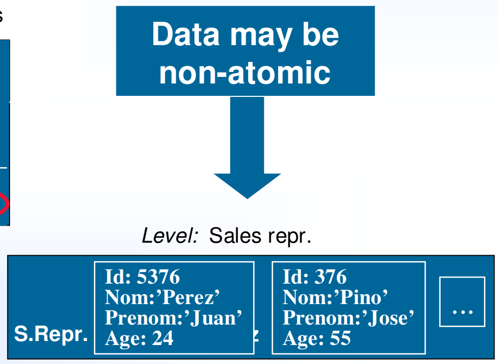
- **Dimensions:**
  - A dimension consists of:



CMDM Model

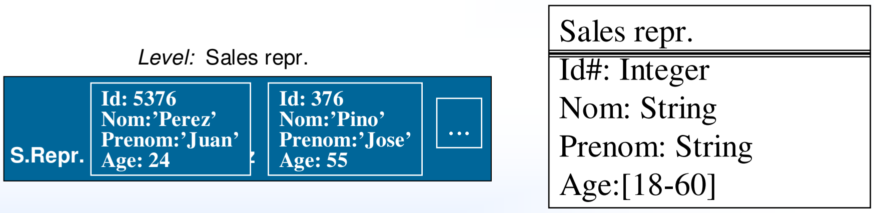
- **Levels:**
  - A level represents a set of data





CMDM Model

- **Levels:**
  - Specification



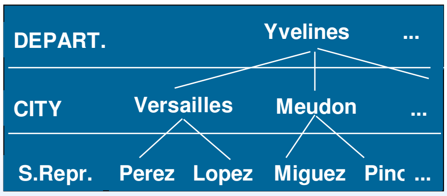
CMDM Model

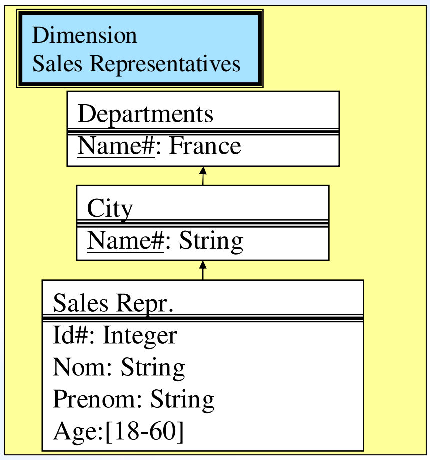
- **Hierarchies:**
  - Levels are organized in hierarchies
  - Each hierarchy may contain one or more levels
  - At each hierarchy
    - There is a <1-n> relation among objects of one level and objects of a parent level

CMDM Model

- **Dimensions:**
  - Specification:

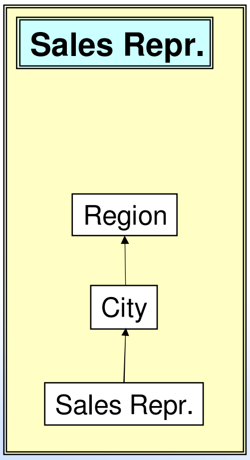
Dimension: Sales Representatives

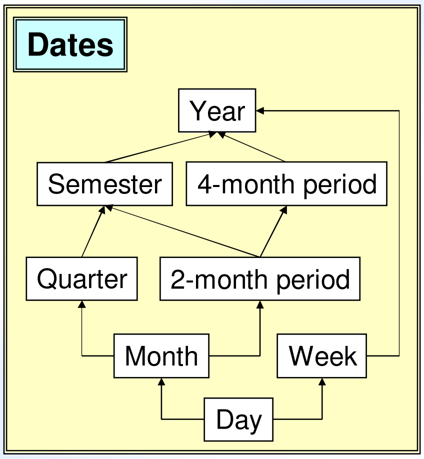




CMDM Model

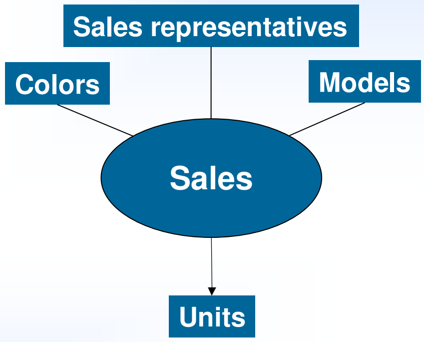
- **Hierarchies may be arbitrarily complex**





CMDM Model

- **Dimensional relations:**
  - Represent dimension crossings
  - View as a relation
    - An object belong to a dimensional relation iff there is a crossing
    - This warranties that participant dimensions are really crossable
  - Measures participate as dimensions

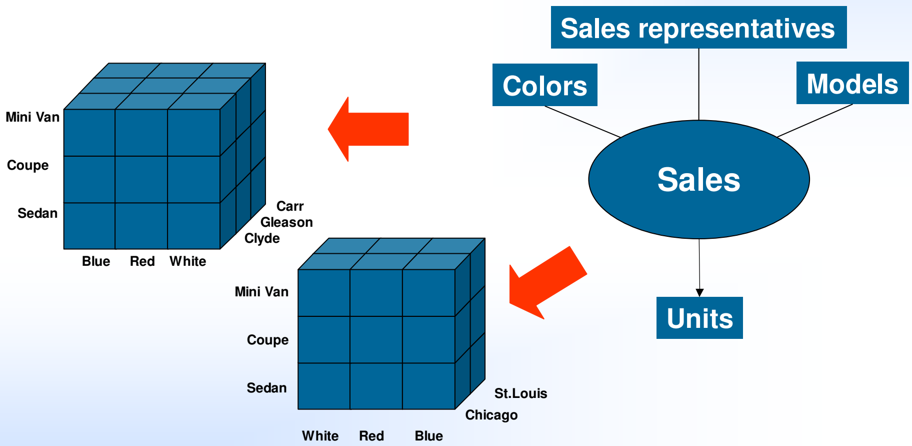


CMDM Model

- **CMDM schema:**
  - Consists of a set of dimensional relations
  - Dimensional relations may share dimensions
    - Allows drill-across
- **Cubes:**
  - Given a dimensional relation, a cube represents a crossing of concrete levels of (some) hierarchies of the relation.

CMDM Model

- **Cubes:**



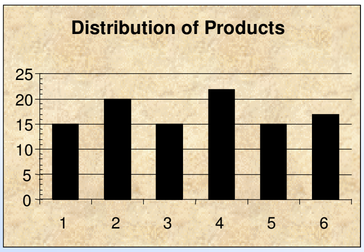
Methodological Aspects

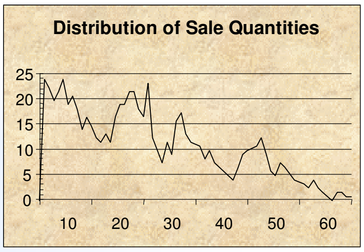
- **Conceptual design is a subjective task**
  - There are no automatic techniques
  - But there are guidelines based on best practices
- **Some key design questions:**
  - Which objects should be dimensions or measures?
  - How to build well-defined hierarchies?
  - How to build well-defined dimensional relations?
  - Which roll-ups are correct and convenient for measures?

Definition of Dimensions and Measures

- **Which objects should be dimensions or measures?**
  - There are no strong criteria but some tips:
    - Business objects and/or analysis criteria vs. Indicators
    - Interest in summarizing by some criteria
    - Identifiers vs. Non Identifiers
    - Categorical vs. Numerical attributes
    - Analysis of value distribution

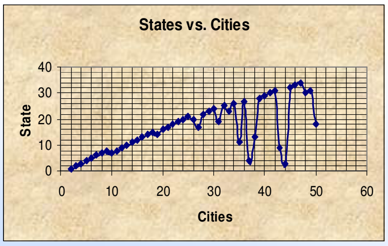
requirements (first two tips) — *data analysis* (last three tips)

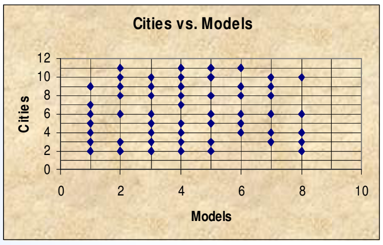




Hierarchies

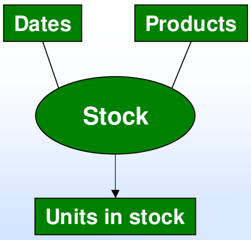
- **Criteria for defining hierarchies:**
  - Navigational paths (drill-down/up)
  - Natural grouping of data (ex. reports)
  - Levels associated to the calculation of indicators
- **How to build well-defined hierarchies?**
  - Each level instance should have a unique father in a hierarchy.
  - Tip: value-dependency analysis
    - Two levels of the same hierarchy should not be independent.

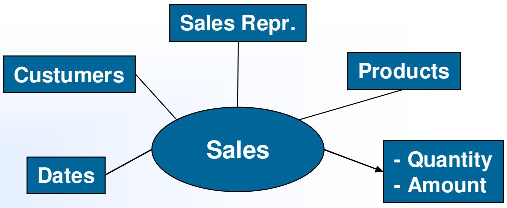




Dimensional Relations

- **Criteria for defining dimensional relations:**
  - Each relation should represent a set of facts
  - A relation may group a set of related measures
- **How to build well-defined dimensional relations?**
  - Dimensions involved in a dimensional relation must be crossable.

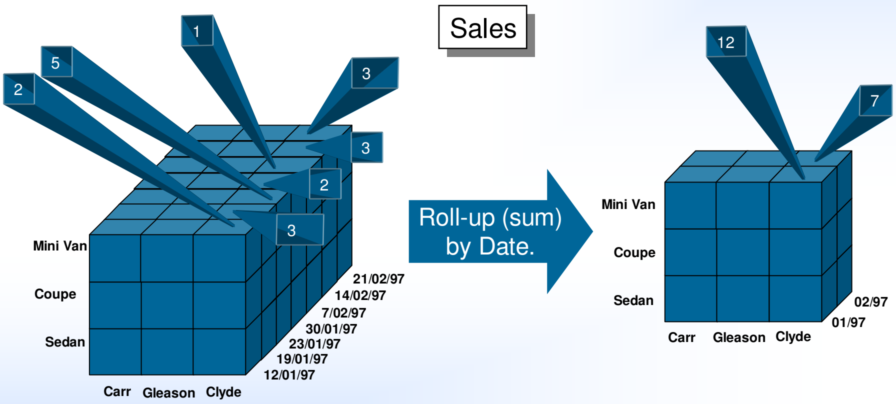




Funciones Roll-Up

- **Given a dimensional relation and a measure:**
  - All roll-ups are correct and convenient?
- **NO !**
  - It is not always possible to sum measures
- **For Lenz & Shoshani**
  (Summarizability in OLAP and Statistical Data Bases)
  - "Summarizability an important property. An incorrect summary may drive to erroneous analysis, conclusions and decisions."
- **Main advances in additivity topics came from Statistical DB**

Additivity

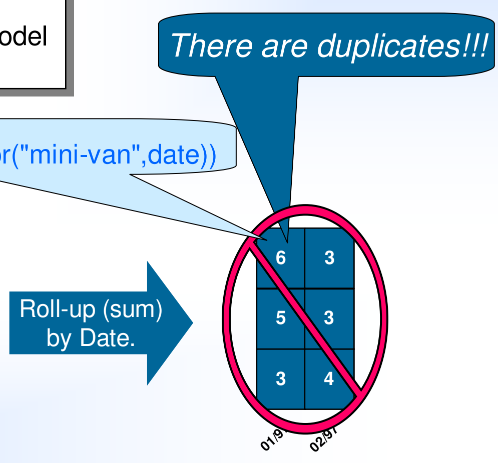


Example of Additivity Problem

<u>Measure</u>: Number of different sale representatives that sold a given model in a given date.

$\Sigma_{month=1 \wedge year=97}$ (NbrSalesRepr("mini-van",date))

	12/01/97	19/01/97	23/01/97	30/01/97	7/02/97	14/02/97	21/02/97
Mini Van	2	0	2	2	1	1	1
Coupe	0	2	3	0	1	1	1
Sedan	1	1	0	1	2	1	1



Types of Measures

- **Flow or additive:**
  - Preserve semantics when applying a sum as RollUp, for all dimensions
  - Refer to an event or period and are registered at its end.
  - Examples:
    - Sales amount, number of birth in a month, number of expedients processed in a week, quantity of sales in a day
- **Stock or half-additive:**
  - Preserve semantics when applying a sum as RollUp, for all dimensions except time
  - Are registered at a specific point in time and refer to that instant.
  - Examples :
    - Stock (of all kinds), account balances, citizen census

Types of Measures

- **Value-per-unit or non-additive:**
  - Don't preserve semantics when applying a sum as RollUp, for some (or all) dimensions
  - Refer to a specific instant but their units difference them from stock measures
  - Examples:
    - Prices per item, exchange taxes, ages, marks
    - Measurements: temperature, flow
    - Count with duplicates

Criteria for a Good Roll-up

- **3 main criteria** (Lenz et al.)**:**
  - Object grouping must determine disjoint sets when drilling-up a hierarchy
  - Object grouping must be complete
  - Roll-up operation should be compatible with the type of measure

Time dimension

	Flow	Stock	Vpu
Min	ok	ok	ok
Max	ok	ok	ok
Sum	ok	not ok	not ok
Avg	ok	ok	ok
Count	ok	ok	ok

Other dimensions

	Flow	Stock	Vpu
Min	ok	ok	ok
Max	ok	ok	ok
Sum	ok	ok	not ok
Avg	ok	ok	ok
Count	ok	ok	ok

Conceptual Design Steps

- **Main tasks are:**
  - Define a schema skeleton:
    - First group of dimensions and measures
    - First version of dimensional relations
  - Map requirements to data sources
  - Refine the schema:
    - Complete dimension hierarchies
    - Adjust dimensional relations
    - Specify calculated measures

iterate

Define a Schema Skeleton

- **Define:**
  - First group of dimensions
  - First group of measures
  - First group of dimensional relations
- **Arise from:**
  - General view of business and requirements
- **Documentation:**
  - Description of business objects
  - Formalization of requirements
  - Prototypes of graphics (for solving requirements)
  - First version of the conceptual schema

Description of Business Objects

Lists:
- Attributes
- Description
- Measure?
- Dimension?
- Optional?

Reference:
Hussman et al.

Attribute	Description	M	D	O
effectiveDay	data import date	no	yes	no
month	time aggregation	no	yes	no
quarter	time aggregation	no	yes	no
year	time aggregation	no	yes	no
accountID	account key	no	yes	no
balance	balance at effective day	yes	no	no
balanceClass	balance classification	no	yes	no
turnover	turnover at effective day	yes	no	no
turnoverClass	turnover classification	no	yes	no
creditlimit	creditlimit of the account	yes	no	no
interest	interest rate	yes	no	no
custID	customer key	no	yes	no
custName	costumer name	no	yes	no
custAge	age of a private customer	no	yes	yes
customerType	classification of customers	no	yes	no
profession	profession of a private customer	no	yes	yes
branch	branch of a business customer	no	yes	yes
productID	product description	no	yes	no
productType	classification of products	no	yes	no
orgID	attending organizational unit	no	yes	no
orgName	name of a organizational unit	no	yes	no
orgGroup	grouping of organizational units	no	yes	no
orgType	classification of organizational units	no	yes	no
businessector	classification of orgGroup and orgType	no	yes	no

Formalization of Requirements

<u>Lists</u>:
– Dimensions and measures involved in each requirement

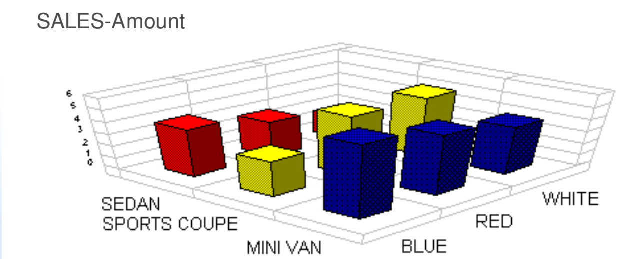
<u>Reference</u>:
Ballard et al.

Table 1. Dimensions, Measures, and Related Questions

Dimensions and Measures	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
Dimensions									
Sales		X		X	X			X	X
Manufacturing	X		X						
Product	X	X	X		X	X	X	X	X
Measures									
Average quantity on hand	X								
Total cost		X	X				X		
Total revenue		X	X				X		
Quantity sold							X		
Percentage of models eligible for discount				X					
Percentage of models eligible for discount that are actually discounted				X					
Percentage of a model sold through a retail outlet					X				
Percentage of a model sold through a corporate sales office order desk					X				
Percentage of a model sold through a sales person					X				

Prototypes of Graphics

- **Goal:**
  – Validate with users what can be obtained as requirement solutions
- **Example:**
  – Amount of sales by make and model



First Version of the Conceptual Schema

- **We must define:**
  - Dimension hierarchies
  - Measures
  - Dimensional relations
- **Formalized with conceptual models**

Map Requirements to Data Sources

- **Mappings between:**
  - Conceptual schema
  - Data sources
- **Goal:**
  - Locate the conceptual objects at sources
  - Verify that data effectively exists
- **Documentation:**
  - Mapping tables
  - Mapping functions

Mapping Tables

- **List pairs**
  - <conceptual item (level), source attribute (table)>
- **We can indicate calculations**

Dimension	Item	Table	Attribute
Products	Product-id	Products	PrCode
Products	Product-description	Products	PrName
Products	Type-id	ProductTypes	PTCode
Products	Type-description	ProductTypes	Concat(PTCode, PTDesc)

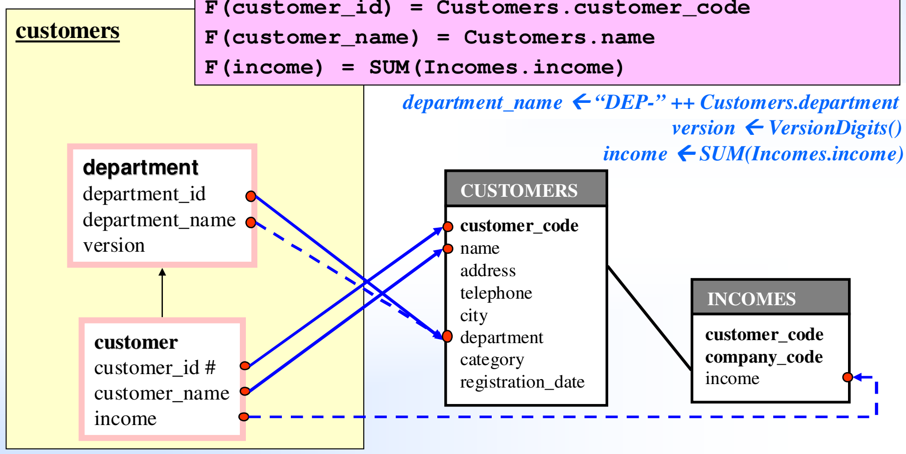
Mapping Functions

- **Map conceptual schema objects to data sources**
  - Associate each conceptual item with an expression:
    - An attribute of a source table
    - A calculation involving several source attributes (single tuple)
    - An aggregation involving several source attributes (several tuples)
    - An external value like a constant, a timestamp or version digit
- **A mapping function for each:**
  - Dimension
  - Dimensional relation

Mapping Functions

```
F(department_id) = Customers.department
F(department_name) = "DEP-" + Customers.department
F(version) = VersionDigits()
F(customer_id) = Customers.customer_code
F(customer_name) = Customers.name
F(income) = SUM(Incomes.income)
```

department_name ← "DEP-" ++ Customers.department
version ← VersionDigits()
income ← SUM(Incomes.income)



Refine the Schema

- **Apply design tips:**
  - Refine dimension hierarchies
  - Refine dimensional relations
  - Specify calculated measures
  - Study roll-up functions
  - Specify integrity constraints
- **Documentation:**
  - Conceptual schema
  - Roll-up analysis

Roll-up Analysis

- **Roll-up tables**

	dates	students	departm.	modules
performance	+	+	+	+
marks	⇒	AVG		
#modules	⇒	NA	NA	+

References:
+ sum
⇒ last period
AVG average
NA no additive count

Plan

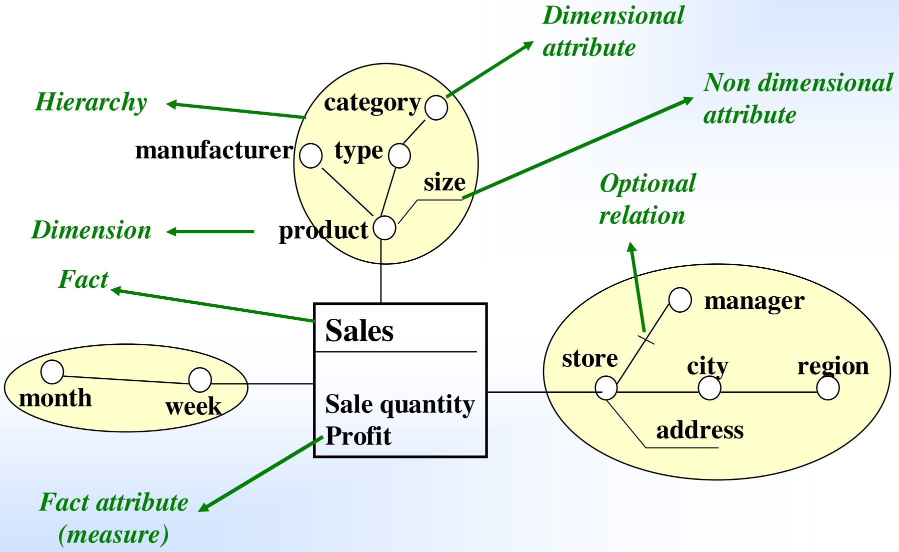
- **Motivation**
  - The importance of conceptual models
- **Requirement-based design**
  - CMDM model
  - Methodological Aspects
- **Data-based design**
  - DF model
  - Methodological Aspects
- **Translation to logical models**
- **Conclusions**

DF Model

- **Basic structures:**
  - Facts
    - Dynamic events associated to the phenomena we want to study
    - May contain measures
      - Without measures count events
  - Dimensions
    - Defines the granularity required for representing facts
  - Hierarchies
    - Grouping criteria
    - A dimension is the lowest level of some hierarchy

- **Conceptual schema = { fact schemas }**

DF Model – Graphic Representation

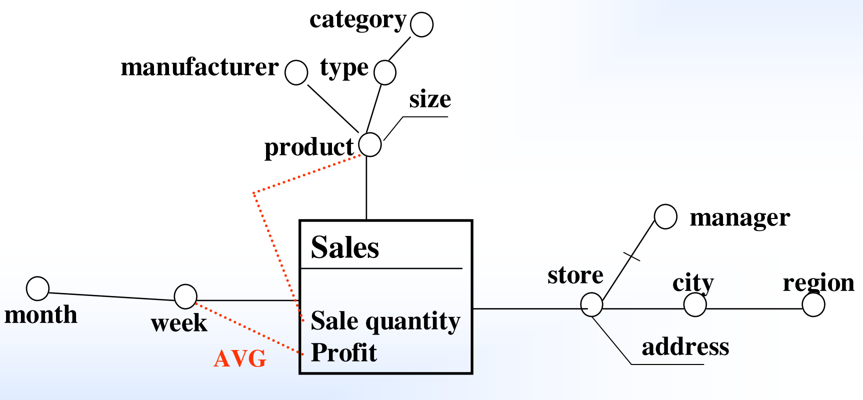


DF Model

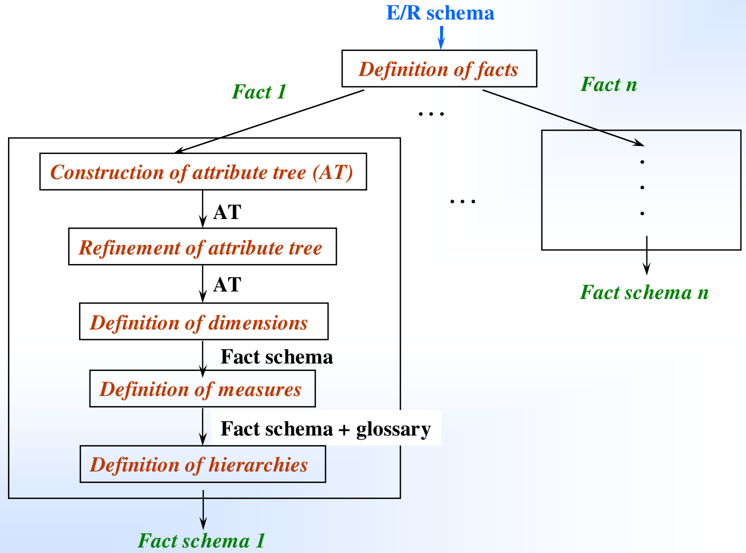
- **Constraints**
  - A fact represents a N:N relation among dimensions
  - Arcs among hierarchy nodes represent N:1 relations
  - A dimension defines the lowest granularity of a hierarchy

- **Additivity:**
  - By default, all measures are additive
  - DF provides graphical elements for representing half-additive and non-additive measures

DF Model – Graphic Representation

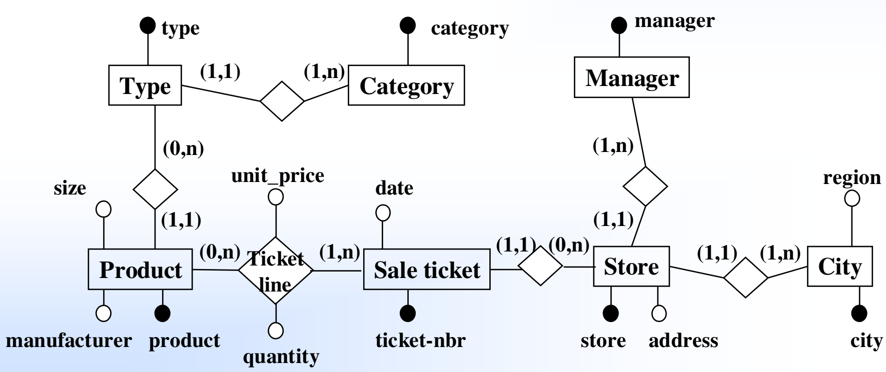


Conceptual Design Methodology



Input E/R schema

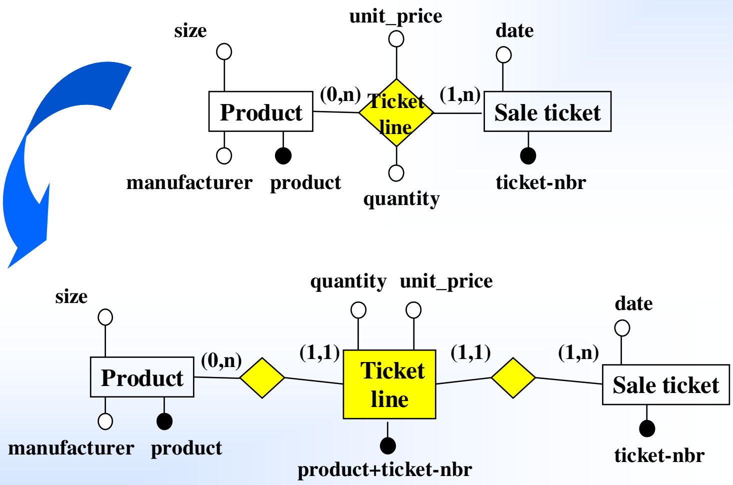
- Simple attributes
- May contain generalization hierarchies
  - Considered as 1:1 relations



Definition of facts

- **A fact**
  - is a concept of interest for the decision support process
  - corresponds to dynamic events associated to the phenomena we want to study
- **In the input E/R schema, facts may be:**
  - An entity F
  - A N:N relation among entities (transform into an entity)
- **Each identified fact becomes the root of an attribute tree**

Definition of facts

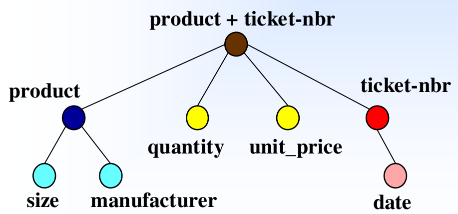


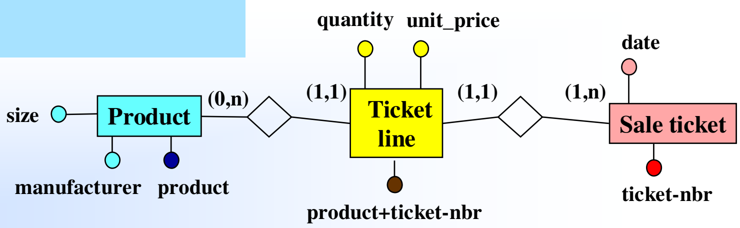
Construction of an attribute tree

- **Attribute tree (AT):**
  - Given an input E/R and an entity F, an AT is a tree that:
    - each node corresponds to an attribut of the E/R ,
    - the root is F,
    - each node functionally determines its descendents
- **Automatic construction**
  - Given an entity *F*, we call *translate(F, id(F))*.

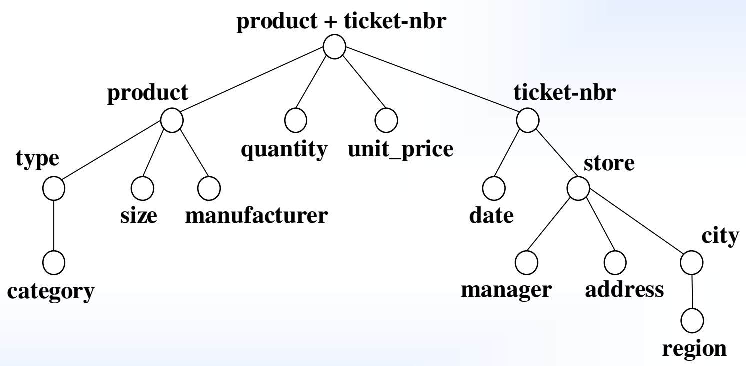
Construction of an attribute tree

translate (E,v)
{ for each **attribute a ∈ E / a ≠ id(E)**
do **addchild(v,a);**
for each **entity G connected to E by a N:1 relation R** *do*
{ for each **attribute b ∈ R** *do*
addchild(v,b);
addchild(v, id(G));
translate(G, id(G));
}
}



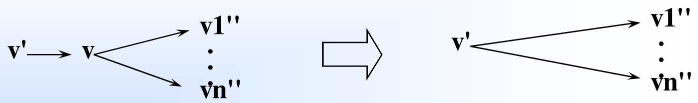


Construction of an attribute tree

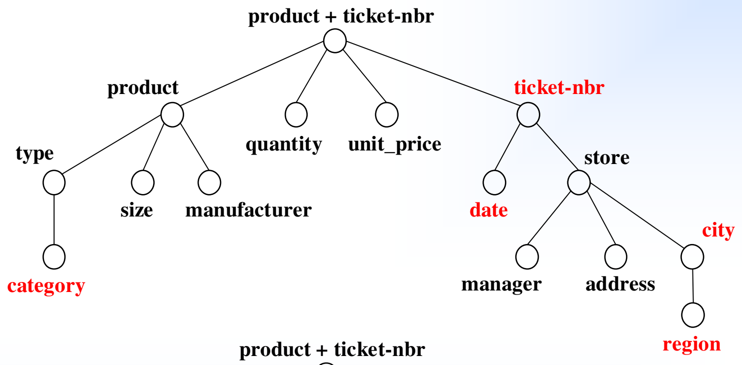


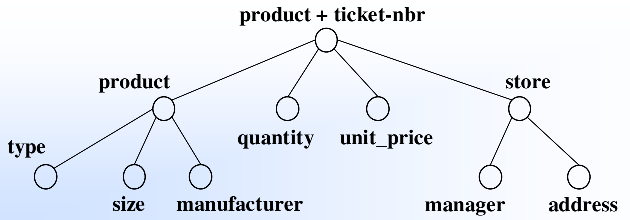
Refinement of the attribute tree

- **Not all attribute nodes may be of interest**
  - for a user
  - for an application domain
- **Two refinement operations:**
  - Pruning
    - Eliminating a sub-tree
  - Grafting
    - Eliminating an intermediate node and linking its children



Refinement of the attribute tree

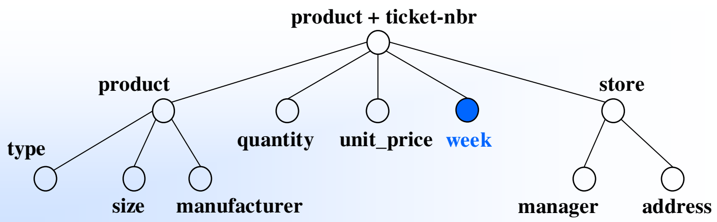




Definition of dimensions

- **Determines the *granularity* of fact instances**
- **Adding of *time* dimension (if not present)**

- **Dimensions in the example:**
  - Time (week)
  - Product
  - Store



Definition of measures

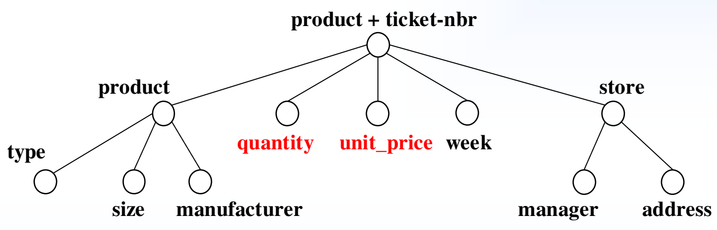
- **Definition of a glossary:**
  - Associates an expression to each measure
    - Describes its calculation from E/R attributes

- **Examples:**
  - salesQuantity = sum(ticket-line.quantity)
  - totalProfit = sum(ticket-line.quantity * ticket-line.unit-price)
  - transactionQuantity = count(ticket-line)

This operations are interpreted as applied to all instances of ticket-line corresponding to a same week, store and product

Definition of measures

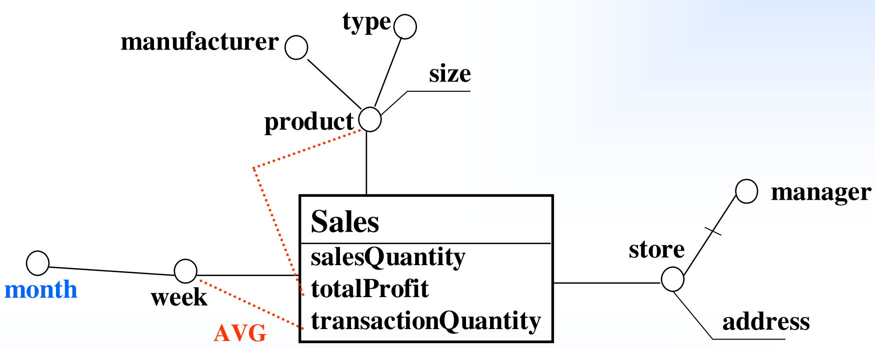
- **We prune unnecessary numerical attributes**



Definition of hierarchies

- **Concerns:**
  - Further refinement of the AT
    - For pruning and grafting unnecessary details
    - For adding new levels representing aggregations
  - Identification of non dimensional attributes
    - Those not used for aggregating
  - Specification of measure additivity

Definition of hierarchies



salesQuantity = sum(ticket-line.quantity)
totalProfit = sum(ticket-line.quantity * ticket-line.unit-price)
transactionQuantity = count(ticket-line)

Plan

- Motivation
  - The importance of conceptual models
- Requirement-based design
  - CMDM model
  - Methodological Aspects
- Data-based design
  - DF model
  - Methodological Aspects
- **Translation to logical models**
- **Conclusions**

Translation to Logical Models

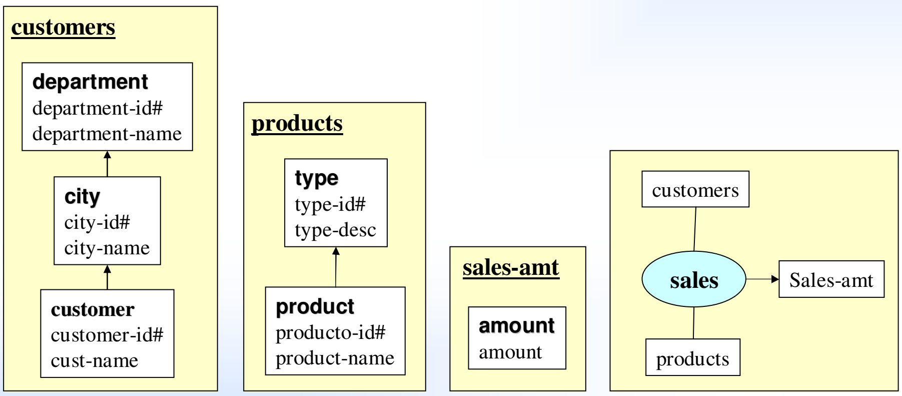
- **Consists of:**
  - Designing logical DB from conceptual schemas
- **Problems:**
  - Provide efficient and simple access
  - Adapt conceptual schemas to logical model
  - Take into account non functional requirements
    - Critical operations
    - Data volumes
- **Works:** Golfarelli et al., Cabibbo & Torlone, Peralta et al.
- **Translation options:**
  - Multidimensional, Relational, Hybrid

Translation to Relational Model

- **MD concepts are mapped to relational structures:**
  - Schemas more or less normalized (star, snowflake…)
- **Provide storage to:**
  - Dimensions
  - Dimensional relations
  - Aggregations

Translation to Relational Model

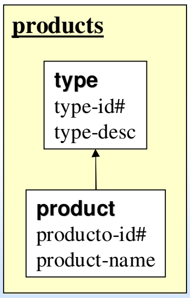
Example: Sales-amount by customer and product



Translation to Relational Model
Dimensions

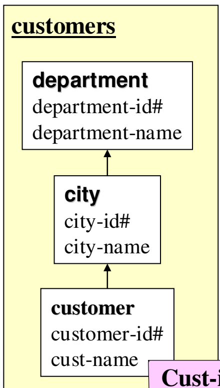
Product-id	Product-name	Type-id
P1	T-shirt	F1
P2	Jean	F1
P3	Dress	F2
P4	Suit	F2

Type-id	Type-desc
F1	Sport
F2	Gala



Product-id	Product-name	Type-id	Type-desc
P1	T-shirt	F1	Sport
P2	Jean	F1	Sport
P3	Dress	F2	Gala
P4	Suit	F2	Gala

Translation to Relational Model
Dimensions



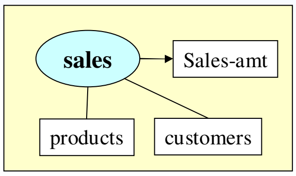
Cust-id	Cust-name	City-id	City-name	Dep-id	Dep-name
C1	Juan	1	Mercedes	K	Soriano
C2	Ana	1	Mercedes	K	Soriano
C3	Pablo	2	Colonia	L	Colonia
C4	José	2	Colonia	L	Colonia
C5	María	3	Carmelo	L	Colonia

Translation to Relational Model

Dimensional relations

Product-id	Product-name	Type-id	Type-desc
P1	T-shirt	F1	Sport
P2	Jean	F1	Sport
P3	Dress	F2	Gala
P4	Suit	F2	Gala

Cust-id	Product-id	Amount
C1	P1	2
C2	P1	5
C2	P2	1
C3	P3	10
C3	P4	3
C4	P1	4
C5	P3	8



Cust-id	Cust-name	City-id	City-name	Dep-id	Dep-name
C1	Juan	1	Mercedes	K	Soriano
C2	Ana	1	Mercedes	K	Soriano
C3	Pablo	2	Colonia	L	Colonia
C4	José	2	Colonia	L	Colonia
C5	María	3	Carmelo	L	Colonia

Translation to Relational Model

Aggregates

Product-id	Product-name	Type-id	Type-desc
P1	T-shirt	F1	Sport
P2	Jean	F1	Sport
P3	Dress	F2	Gala
P4	Suit	F2	Gala

Type-id	Type-desc
F1	Sport
F2	Gala

Cust-id	Product-id	Amount
C1	P1	2
C2	P1	5
C2	P2	1
C3	P3	10
C3	P4	3
C4	P1	4
C5	P3	8

City-id	Type-id	Amount
1	F1	8
2	F2	13
2	F1	4
3	F2	8

Cust-id	Cust-name	City-id	City-name	Dep-id	Dep-name
C1	Juan	1	Mercedes	K	Soriano
C2	Ana	1	Mercedes	K	Soriano
C3	Pablo	2	Colonia	L	Colonia
C4	José	2	Colonia	L	Colonia
C5	María	3	Carmelo	L	Colonia

City-id	City-name	Dep-id	Dep-name
1	Mercedes	K	Soriano
2	Colonia	L	Colonia
3	Carmelo	L	Colonia

Translation to Relational Model

- **Fact tables:**
  - One table for each dimensional relation
    - Contains dimension keys (lower levels) + measures
- **Dimension tables:**
  - Star: one table for each dimension
    - Contains all dimension attributes
  - Snowflake: one table for each dimension level
    - Contains all level attributes + father levels keys
- **Aggregates**
  - One table for each aggregate
    - Contains level keys (aggregation levels) + measures
  - Derived dimension tables may be necessaries
    - Contains dimension attributes (aggregation level)

Translation to MD Model

- **MD concepts are mapped to logical MD structures:**
  - Proprietary structures (different for each DBMS)
- **Translation consists on the definition of dimensions, measures and cubes in the DBMS interface**
- **Warning! Some tools merge concepts:**
  - Definition of dimensions, measures and cubes
  - Specification of loading mechanisms for these structures

  Sometimes a star/snowflake schema is required (auxiliar DB) to load MD structures from it

Plan

- Motivation
  - The importance of conceptual models
- Conceptual Models
  - Modeling needs
  - CMDM model
  - DF model
- Methodological Aspects
  - Requirement-based design
  - Data-based design
- Translation to logical models
- **Conclusions**

Conclusions

- **Conceptual design consists in specifying MD models for a given problem:**
  - It is a subjective task
  - There are modeling guidelines based on best practices
  - Most formalisms and properties comes from statistics DB
- **Approaches for conceptual design**
  - Requirement-based design
    - Applicable for complex sources
  - Data-based design
    - Applicable when requirements are not clear enough
- **Translation to logical models is quite direct**
- **Conceptual models classifications:** Carpani, Abello

Bibliography

- Abelló, A.; Samos, J.; Saltor, F.: "A Data Warehouse Multidimensional Data Models Classification". Technical Report LSI-2000, Universidad de Granada, España, 12/2000.
- Ballard, C. Herreman, D. Schau, D. Bell, R. Kim, E. Valncic, A.: "Data Modeling Techniques for Data Warehousing", SG24-2238-00. IBM Red Book. ISBN number 0738402451. 1998.
- Cabibbo, L.; Torlone, R.:"A Logical Approach to Multidimensional Databases", EDBT, 1998.
- Carpani, F.: "CMDM: A conceptual multidimensional model for Data Warehouse". Master Thesis. Advisor: Ruggia. InCo - Pedeciba, UdelaR, Uruguay, 2000.
- Franconi, E.; Sattler, U.:"A Data Warehouse Conceptual Data Model for Multidimensional Aggregation", DMDW'99, Germany, 1999.
- Golfarelli, M.; Maio, D.; Rizzi, S.:"Conceptual Design of Data Warehouses from E/R Schemes.", HICSS'98, IEEE, Hawaii,1998.

Bibliography

- Hüsemann, B.; Lechtenbörger, J.; Vossen, G.:"Conceptual Data Warehouse Design". DMDW'00, Sweden, 2000.
- Kenan Technologies:"An Introduction to Multidimensional Databases". White Paper, Kenan Technologies, 1996.
- Kimball, R.:"The Data Warehouse Toolkit". John Wiley & Son, Inc., 1996.
- Lenz, H.; Shoshani, A.: "Summarizability in OLAP and Statistical Databases ". Conf. on Statistical and Scientific Databases, 1997.
- Peralta, V.; Illarze, A.; Ruggia, R.: "On the Applicability of Rules to Automate Data Warehouse Logical Design"**.** DSE'2003, Klagenfurt, Austria, 2003.
- Sapia, C.; Blaschka, M.; Höfling, G.; Dinter, B.: "Extending the E/R Model for the Multidimensional Paradigm". DWDM'98, Singapure, 1998.
- Thomsen, E.: "OLAP Solutions. Building Multidimensional Information". John Wiley & Sons, Inc. 1997.