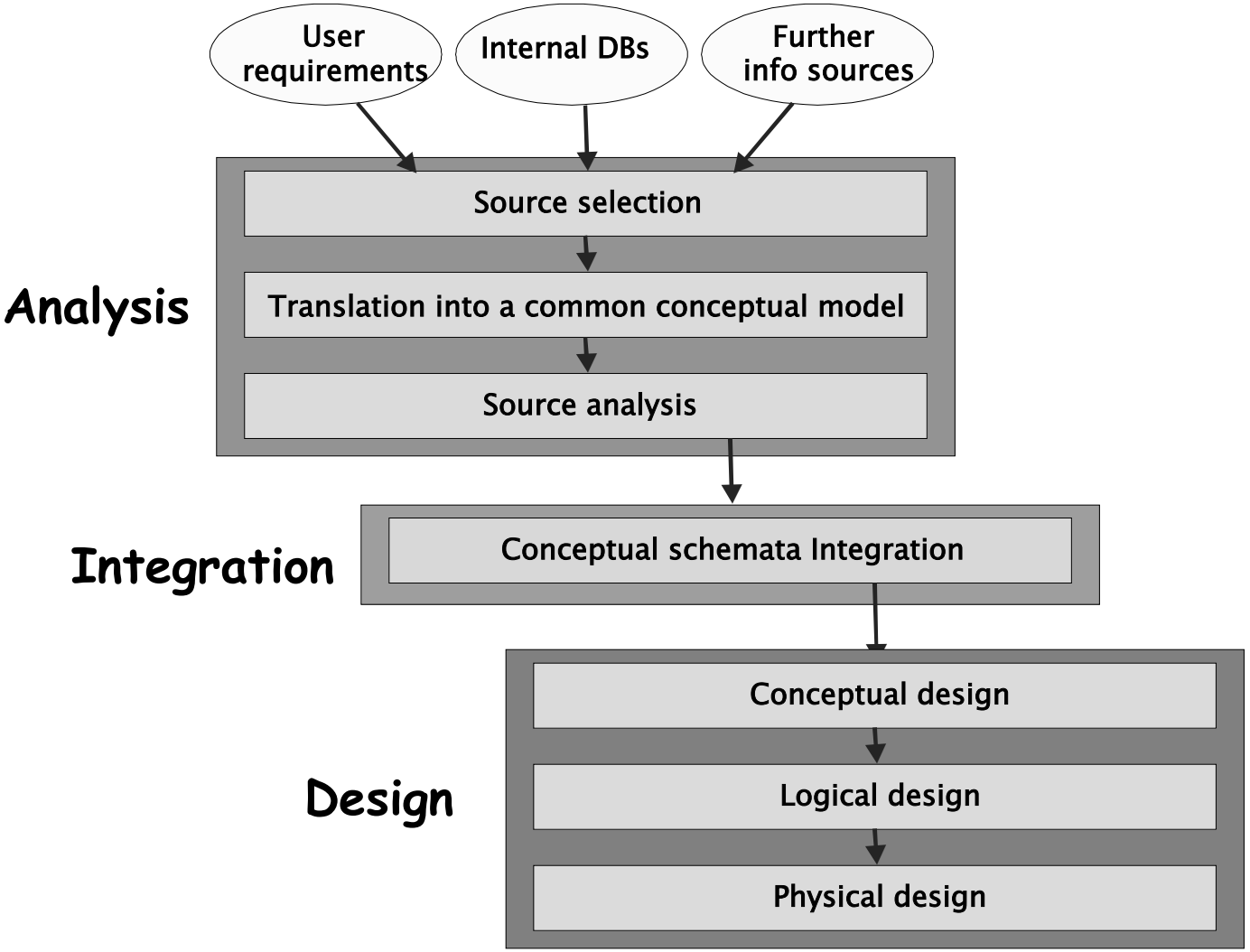
Data Warehouse Design

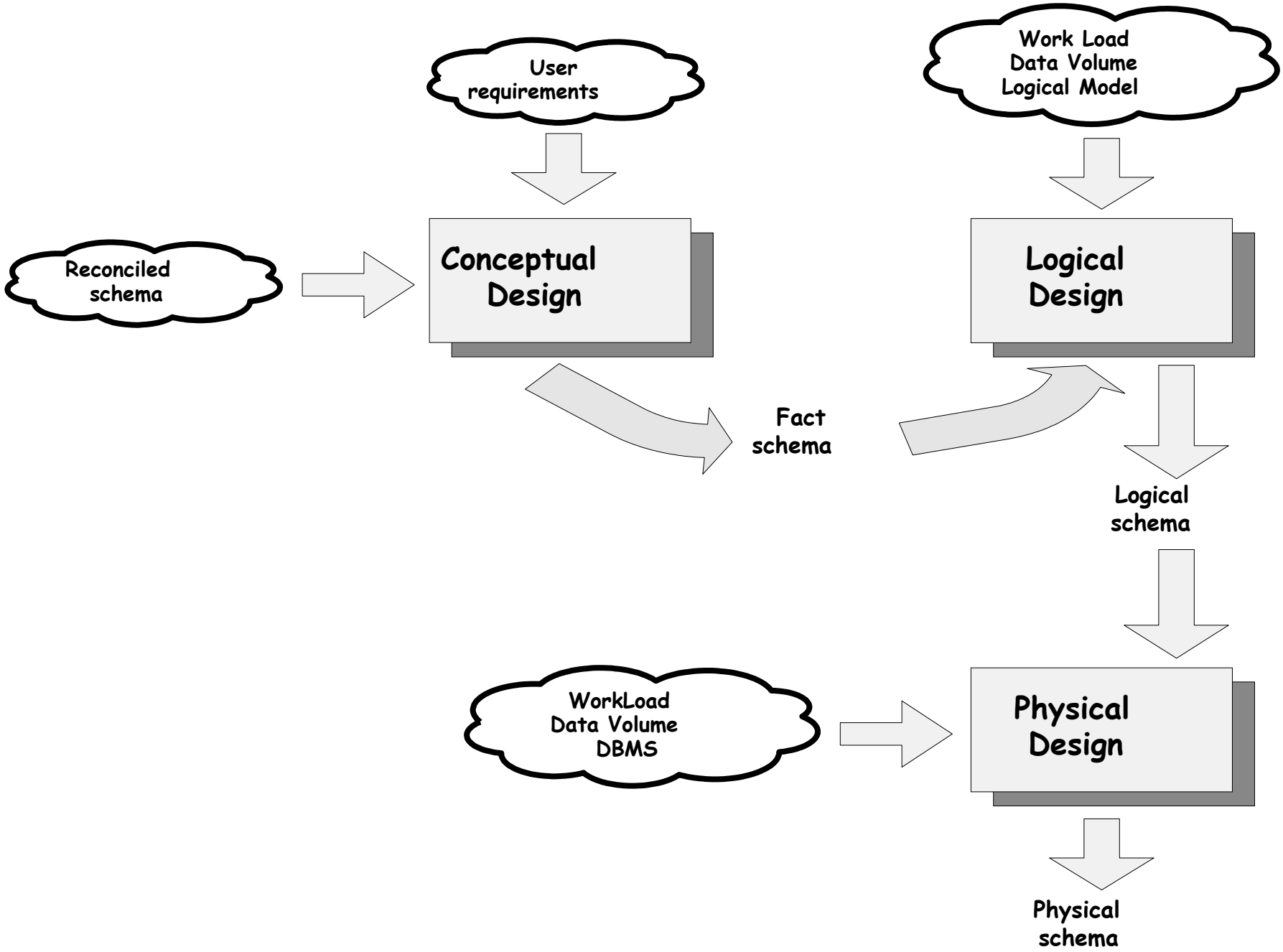
Letizia Tanca
Politecnico di Milano
(with the kind support of
Rosalba Rossato)

Data Warehouse Design



Data Warehouse Design

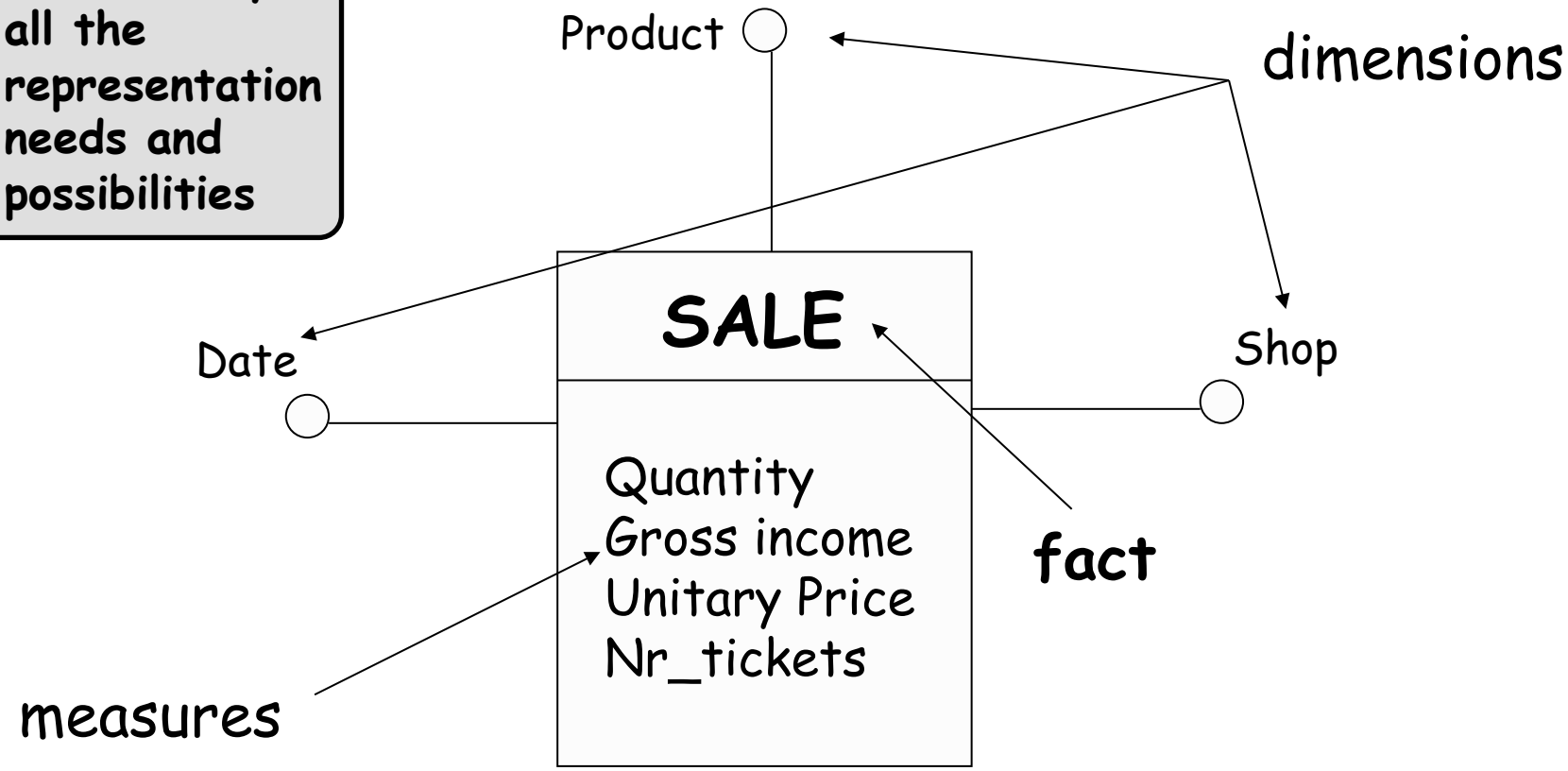
- Data Warehouses are based on the multidimensional model
- A common conceptual model for DW does not exist
- The Entity/Relationship model cannot be used in the DW conceptual design


User requirements
Work Load
Data Volume
Logical Model
Reconciled schema
Conceptual Design
Logical Design
Fact schema
Logical schema
WorkLoad
Data Volume
DBMS
Physical Design
Physical schema

Conceptual Model

Fact Schema

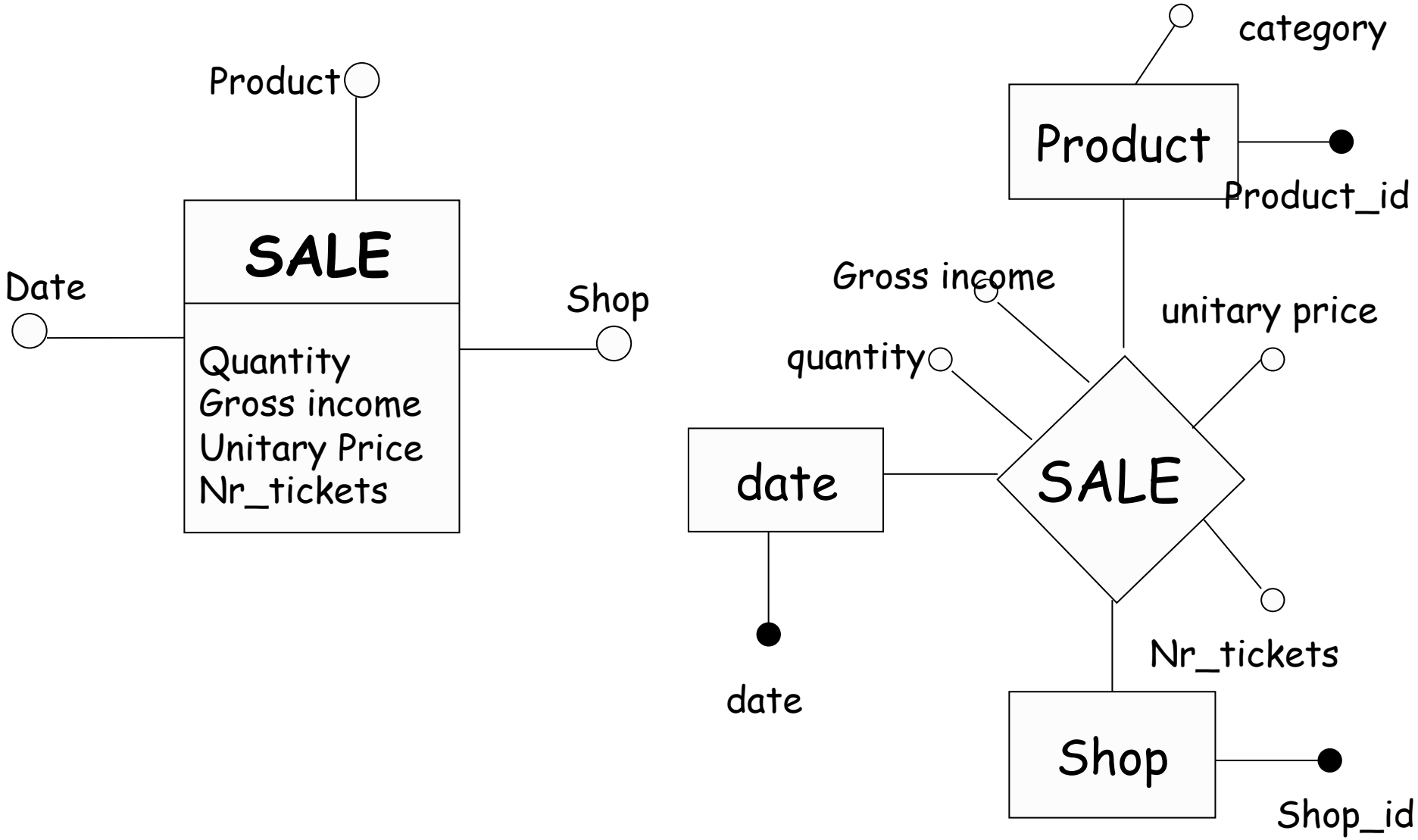
Let us analyze all the representation needs and possibilities



DFM and E/R

- A fact describes an N:M relationship among its dimensions
- There must be a functional dependency between the fact and its dimensions
- Naming convention: the dimensions of a same fact schema must have distinct names

DFM and E/R


Product
SALE
Date
Shop
Quantity
Gross income
Unitary Price
Nr_tickets
category
Product
Product_id
Gross income
unitary price
quantity
date
SALE
date
Nr_tickets
Shop
Shop_id

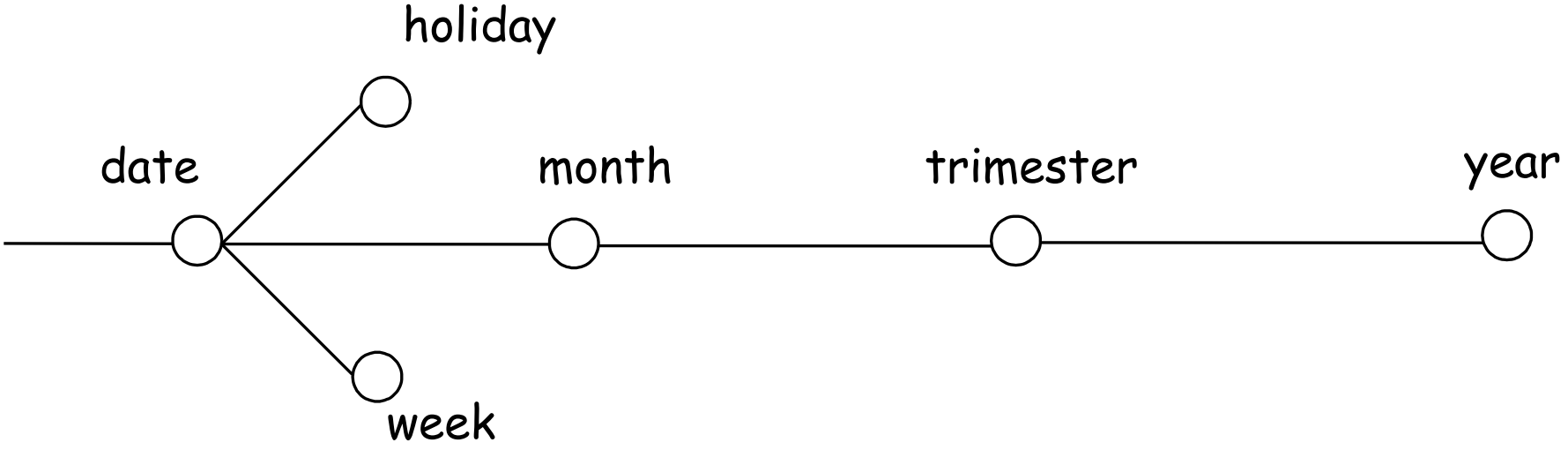
Dimensional attribute

- A dimensional attribute must assume discrete values, so that it can contribute to represent a dimension

- Dimensional attributes can be organized into hierarchies

Hierarchy

- A dimensional hierarchy is a directional tree whose
  - Nodes are dimensional attributes
  - Edges describe n:1 associations between pairs of dimensional attributes
  - Root is the considered dimension



Events and aggregations

- A primary event is an occurrence of a fact; it is represented by means of a tuple of values

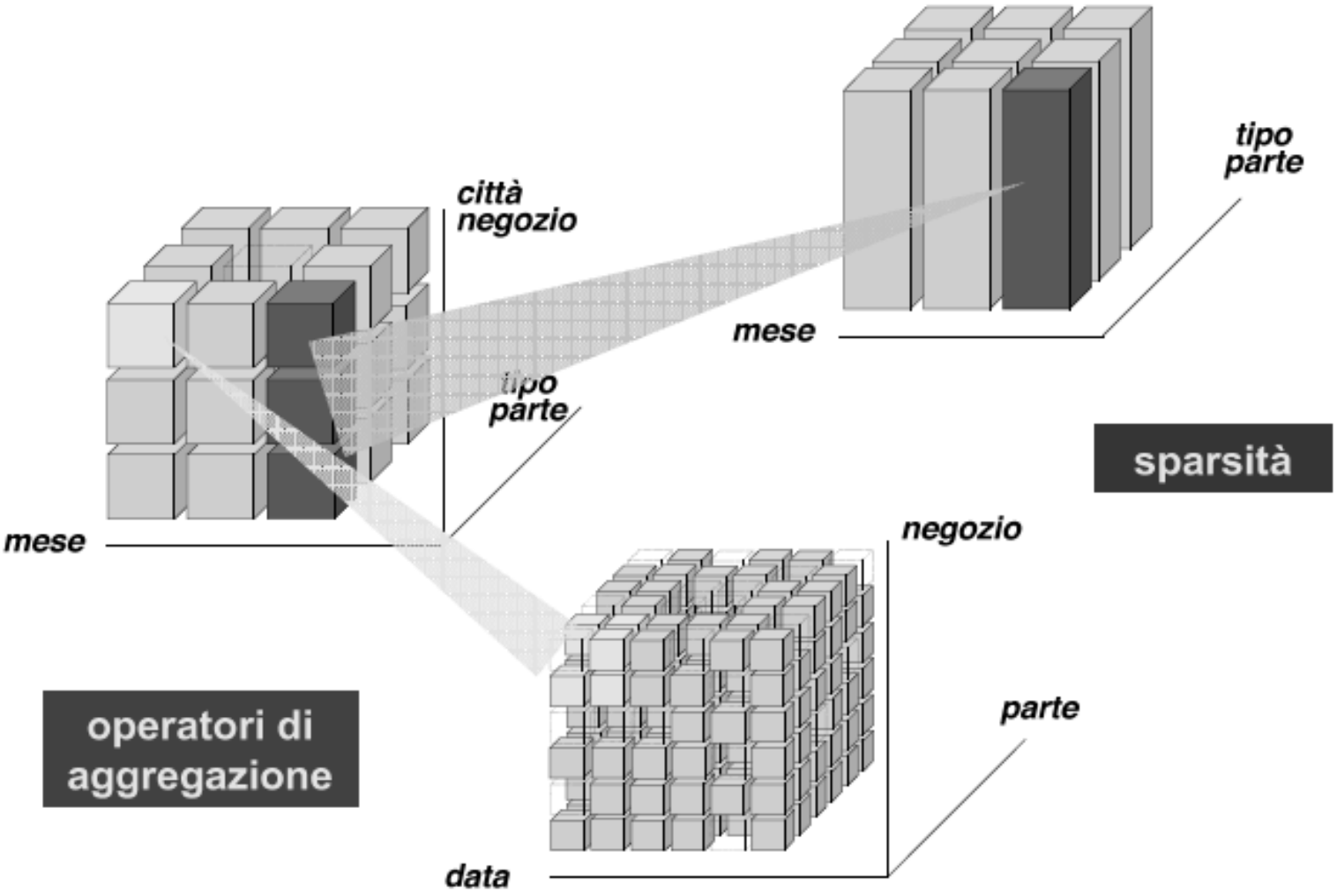
  - ✓ On 10/10/2001, ten 'Brillo' detergent packets were sold at the BigShop for a total amount of 25 euros

Events and aggregations (2)

- A hierarchy describes how it is possible to group and select primary events

- The root of a hierarchy represents the finest aggregation granularity

Events and aggregations



Events and aggregations (3)

- Given a set of dimensional attributes (pattern), each tuple of their values identifies a secondary event that aggregates (all) the corresponding primary events
- For each dimensional attribute, a value is associated with the secondary event; this value summarizes the values assumed by the corresponding measure in the primary events

  For example the sales can be grouped by Product and Month:

  ✓ in October 2001, 230 'Brillo' detergent packets were sold at the BigShop for a total amount of 575 euros

Secondary event

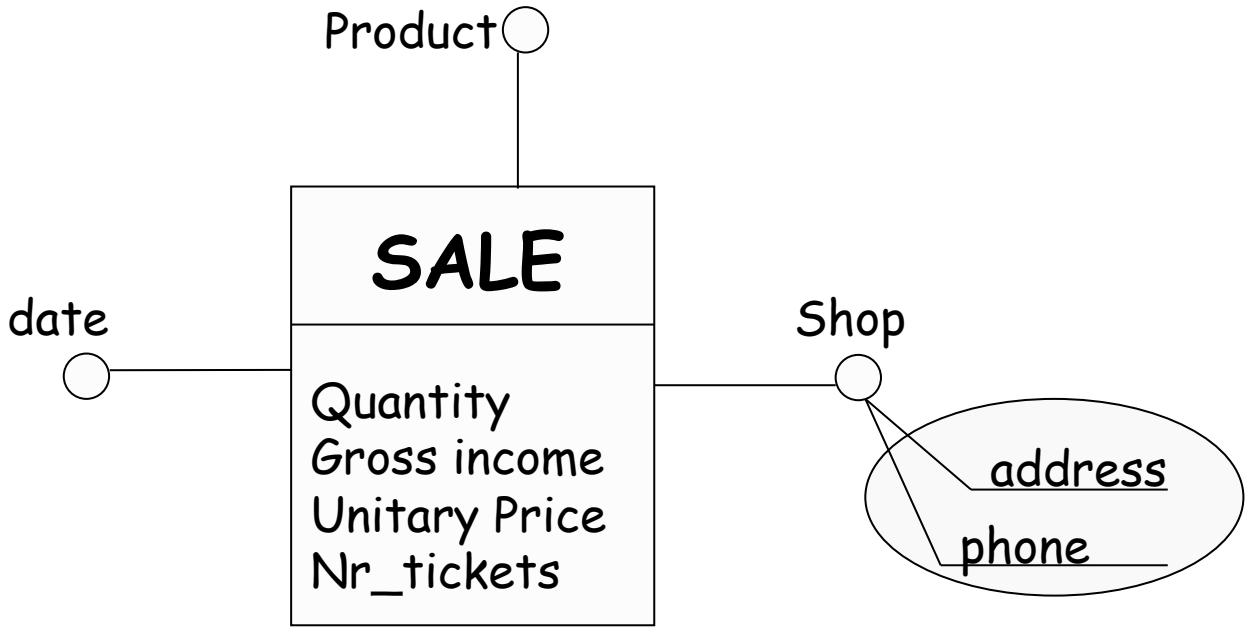
- The sales can be further grouped by Product, Month, and City
- If we consider city, product and month as dimensional attributes, the tuple

  (city: 'Rome' , product: 'Brillo' , month: 10/2001)

  identifies another secondary event
- It aggregates all the sales related to the product 'Brillo' in shops of 'Rome' during the month October 2001

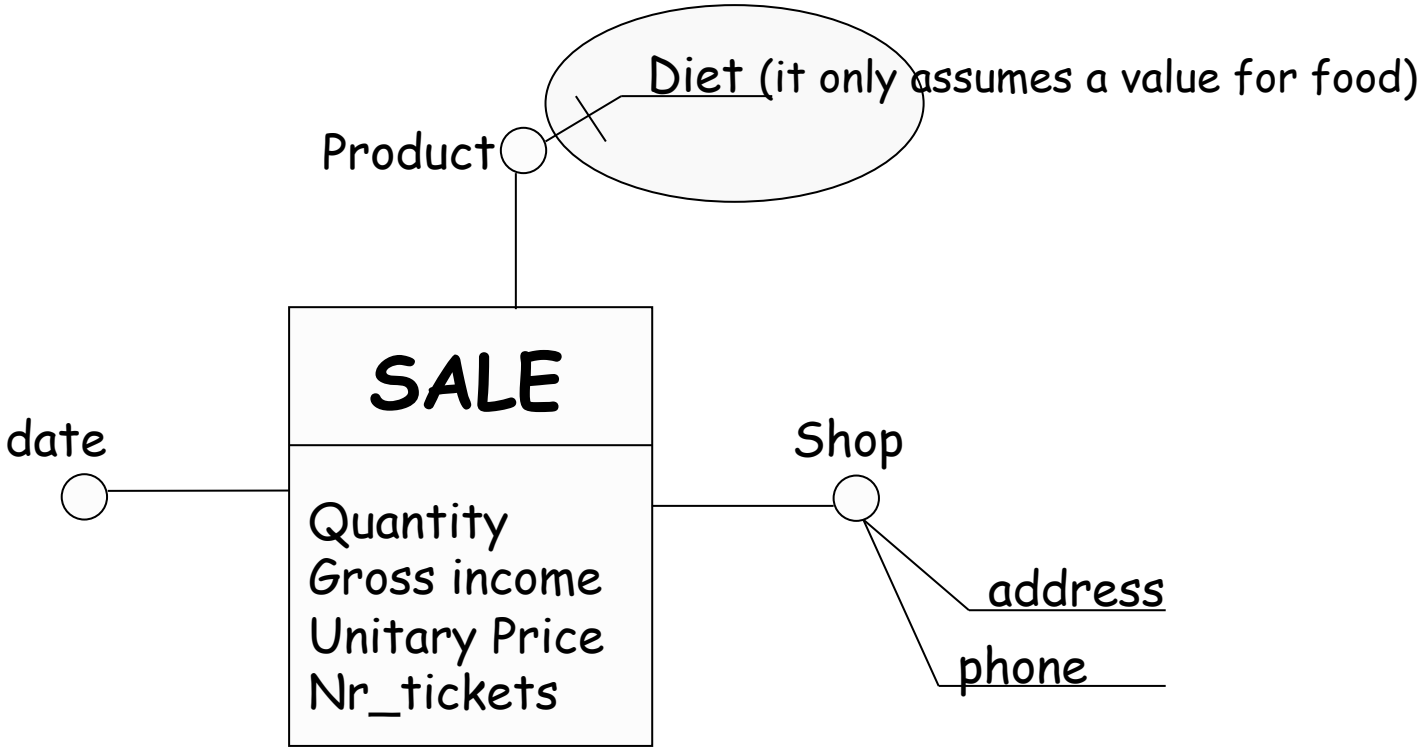
Descriptive attributes

- A descriptive attribute contains additional information about a dimensional attribute
- They are uniquely determined by the corresponding dimensional attribute

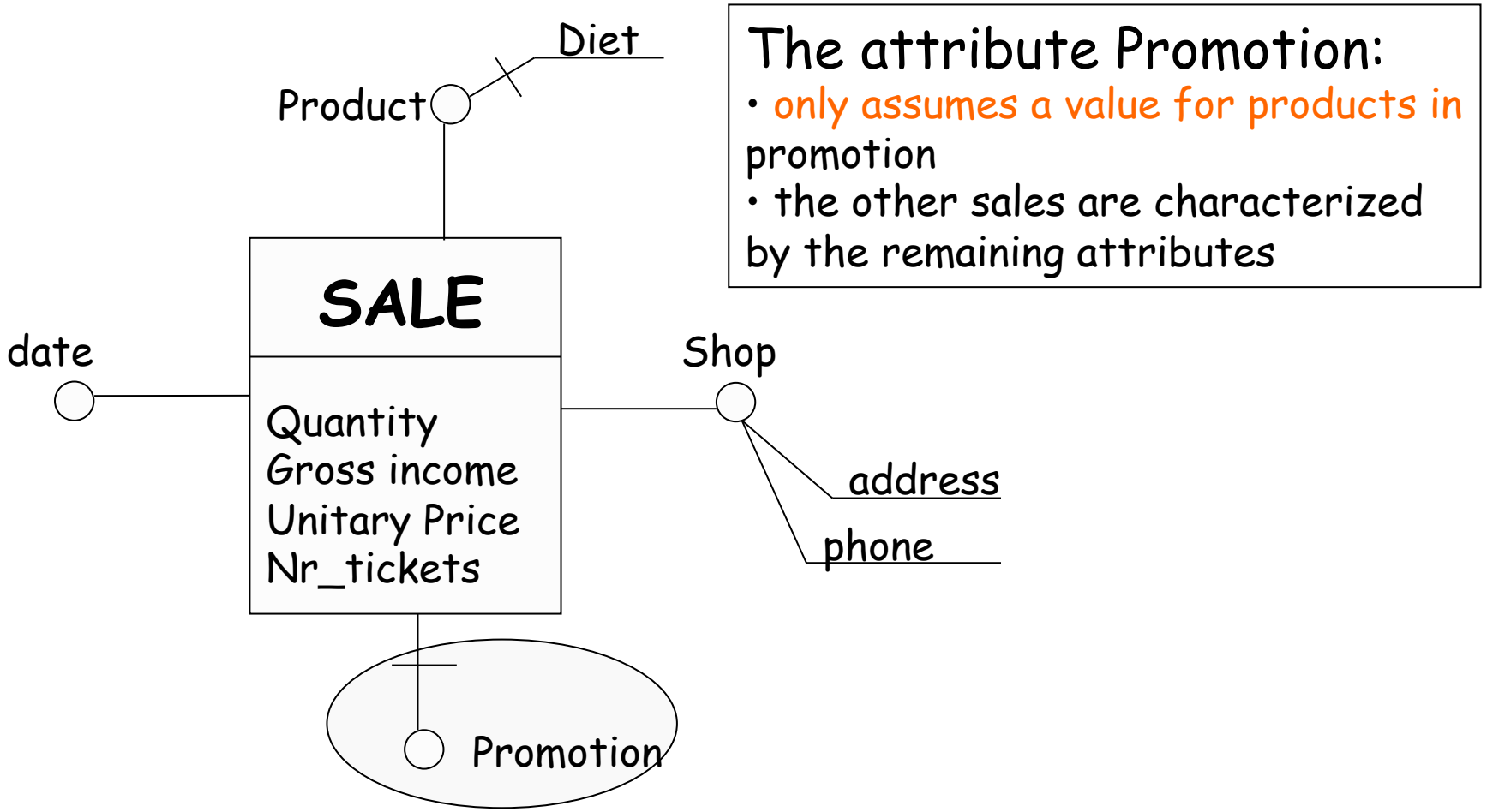


Optional edges

- Some edges of a fact schema could be optional



Optional dimensions



The attribute Promotion:

- only assumes a value for products in promotion
- the other sales are characterized by the remaining attributes

Cross-dimensional attributes

- A cross-dimensional attribute is a dimensional or a descriptive attribute whose value is obtained by combining values of some dimensional attributes
  - ✓For example, IVA (VAT) is computed based on the product category and the state

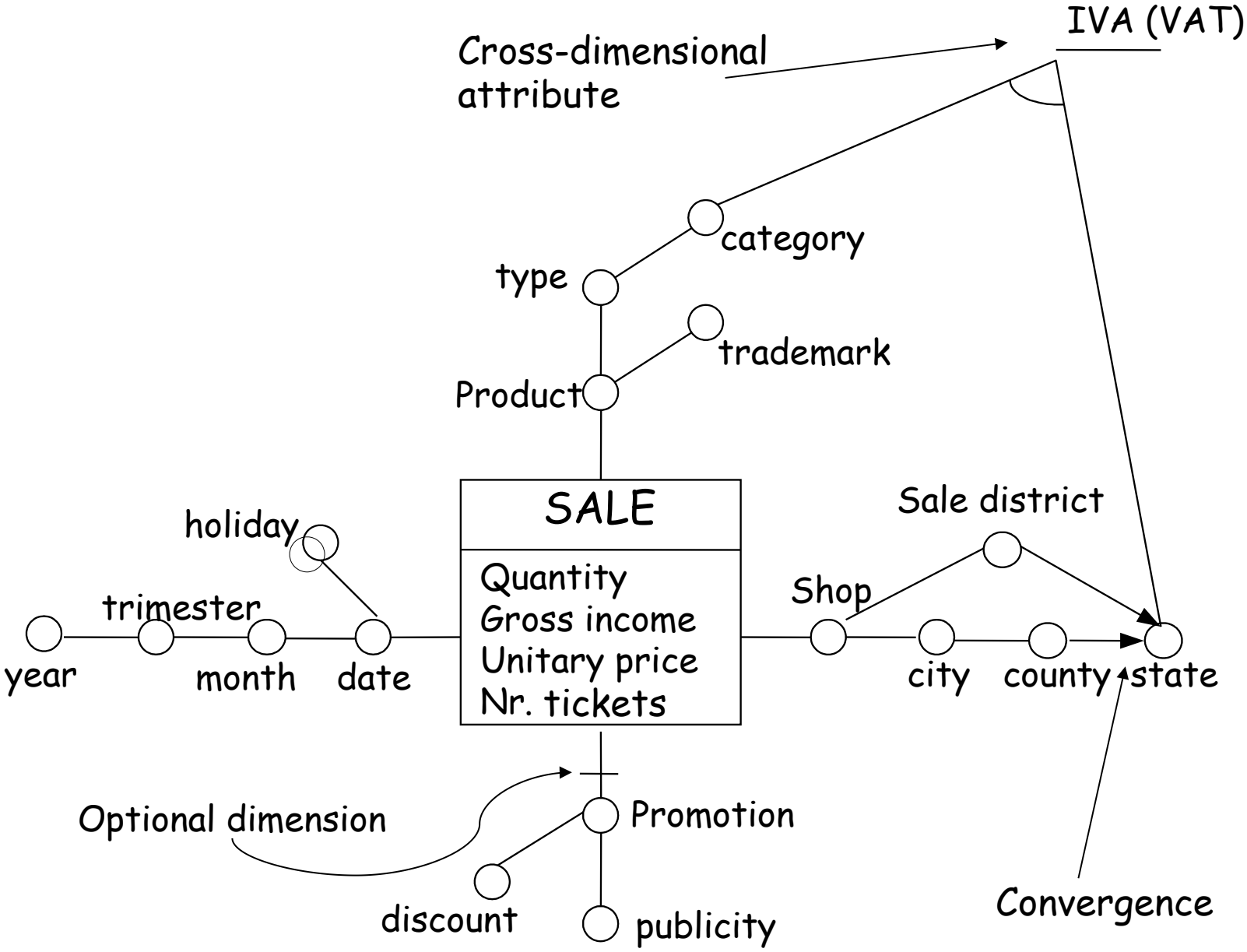
Convergence

- It is related to the structure of a hierarchy
  - ✓Two dimensional attributes can be connected by more than two distinct directed edges
  - ✓For example:

  Shop → city → county→ state

  or

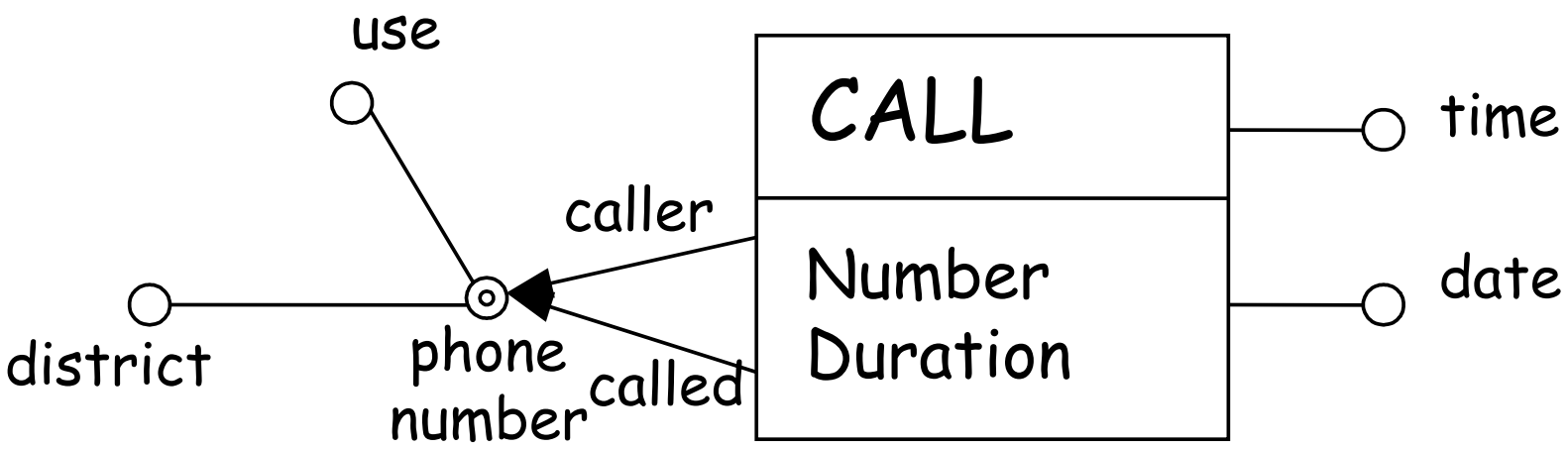
  Shop→ sale district → state


IVA (VAT)
Cross-dimensional attribute
category
type
trademark
Product
SALE
Quantity
Gross income
Unitary price
Nr. tickets
holiday
Sale district
trimester
Shop
year
month
date
city
county
state
Optional dimension
Promotion
discount
publicity
Convergence

Hierarchy Sharing

- In a fact schema, some portions of a hierarchy might be duplicated
- As a shorthand we allow hierarchy sharing
- If the sharing starts with a dimension attribute, it is necessary to indicate the *roles* on the incoming edges
- Necessary condition: the functional dependency must hold on both branches

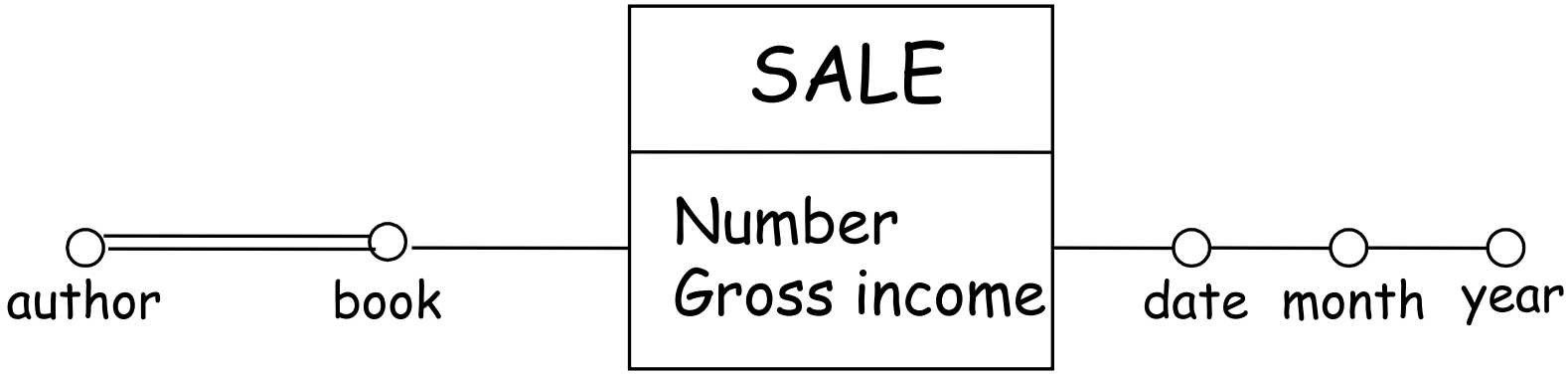
Hierarchy Sharing



It is in fact a shorthand to represent
the duplication of the whole hierarchy

Multiple edges

- Some attributes, or some dimensions, may be related by a many-to-many relationship



  - we denote them by multiple edges
  - they are dealt with in a special way

Measure Aggregation

- Aggregation requires to specify an operator to combine values related to primary events into a unique value related to a secondary event
- A measure is additive w.r.t. a given dimension iff the SUM operator is applicable to that measure along that dimension

The n.of tickets is non-additive (and non-aggregable) w.r.t. the product

- **By n. of tickets we mean the n. of "buyings" i.e. the ticket count**
- **The association between product and ticket is many-to-many**
- **E.g. by summing up the ticket count on the product type we count the same type twice if it is the type of products which are in the same ticket**

Ticket	Product	Type
S1	P1	T1
S1	P2	T1
S2	P1	T1
S2	P3	T2

how many tickets with p=p1 ? → 2

how many tickets with p=p2 ? → 1

how many tickets with p=p3 ? → 1

how many tickets with t=t1 ? → 2

BUT

sum(how many tickets with type(p) =t1) = 3 !!!

Additivity

It is possible to identify three different measure categories [Lenz' 97]:

- Flow measures: related to a time period
  - N. of sales per day, monthly gross income, n. of births in a year
- Level measures: evaluated in particular time instants
  - N. products in stock, n. inhabitants in a city
- Unitary measures: relative measures
  - Unitary price at a given instant, money change rate, interest rate

Additivity

- Flow measures: they are related to a time period; at the end of the period the measures are evaluated in a cumulative way
- Level measures: they are evaluated in particular time instants
- Unitary measures: they are evaluated in particular time instants *but they are relative measures*

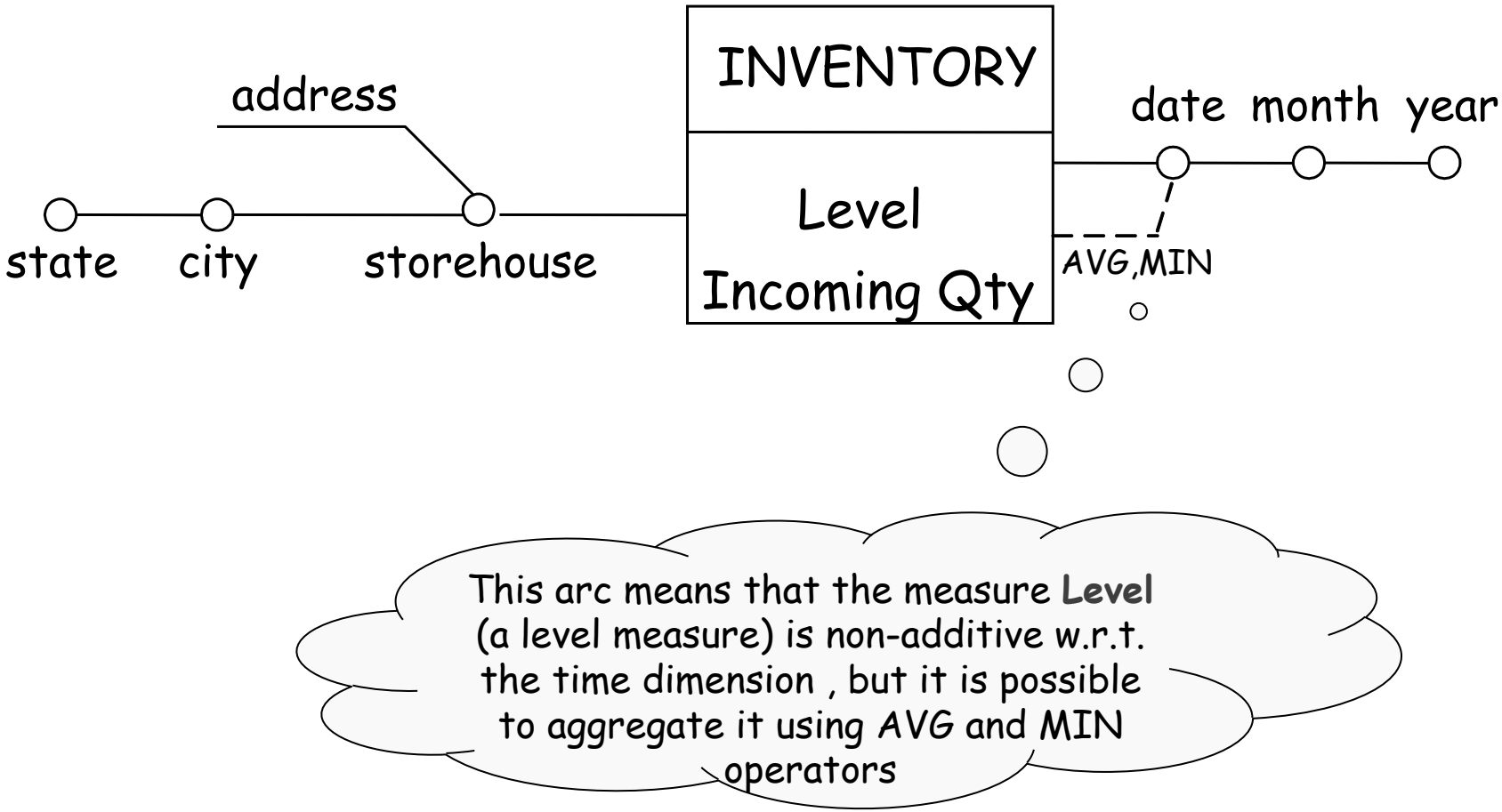
examples

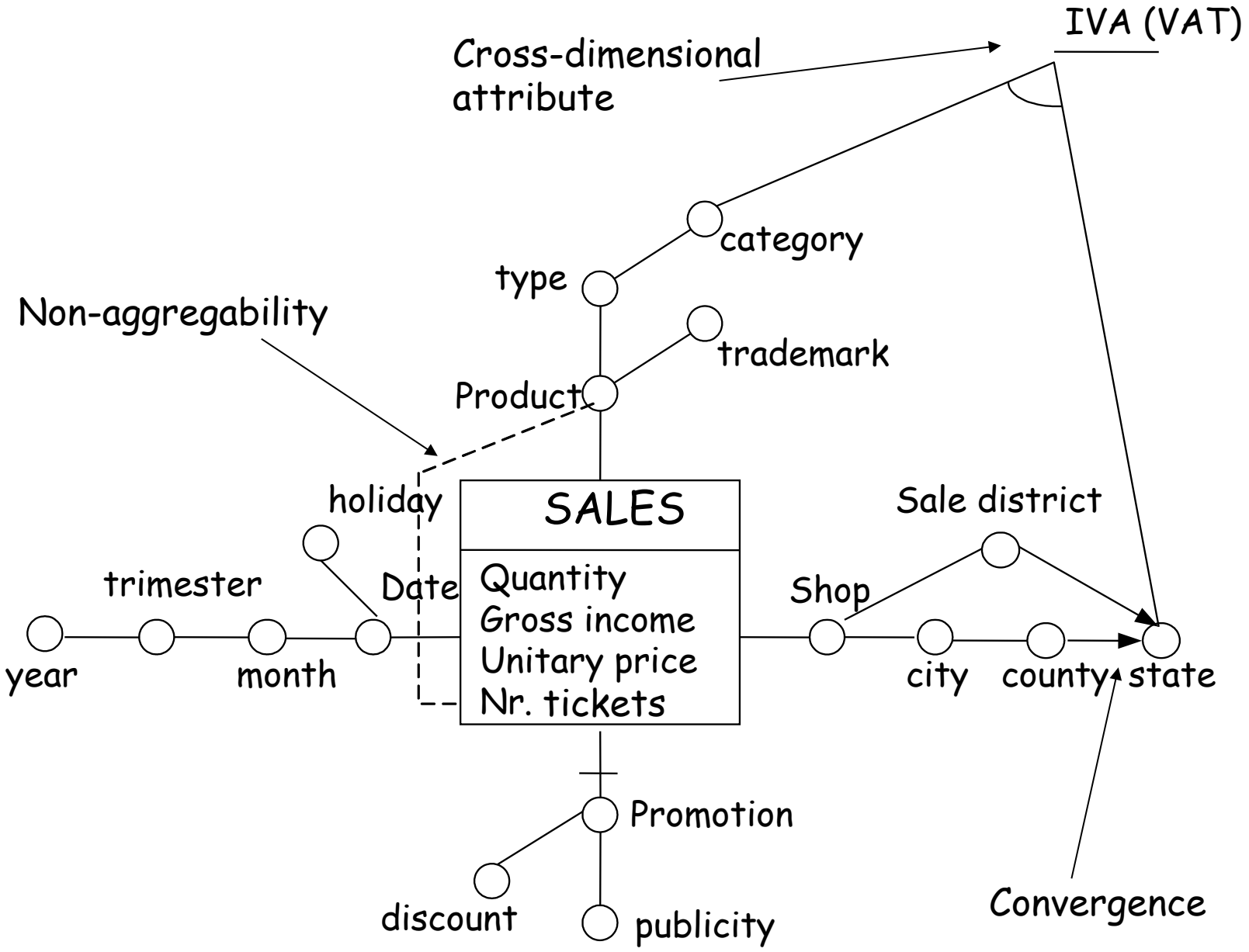
- Level measures: evaluated in particular time instants
  - n. products in stock can be aggregated by SUM over the category/type or the shop/city hierarchies, but NOT over the time hierarchy
  - n. inhabitants in a city can be aggregated by SUM over a region, again NOT over time
- Unitary measures: relative measures
  - unitary price at a given instant CANNOT be aggregated by sum over the category/type or the shop/city hierarchies, NOR over the time hierarchy
  - the same for money change rate, interest rate

Aggregation

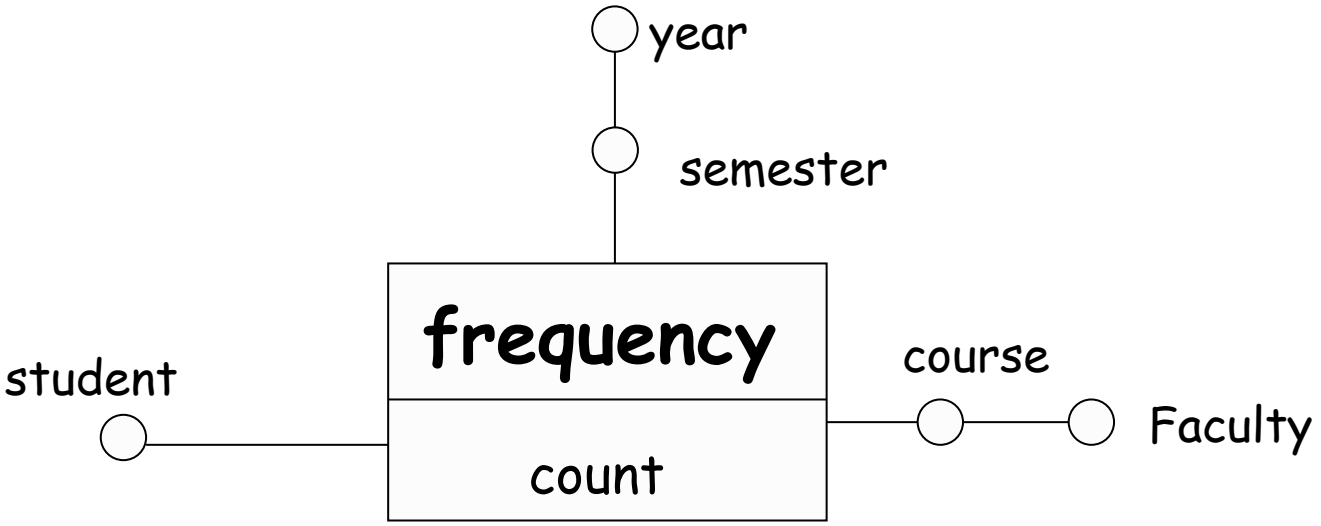
	Over temporal hierarchies	Over non-temporal hierarchies
Flow measures	SUM, AVG, MIN, MAX	SUM, AVG, MIN, MAX
Level measures	AVG, MIN, MAX	SUM, AVG, MIN, MAX
Unitary measures	AVG, MIN, MAX	AVG, MIN, MAX

Aggregability




IVA (VAT)
Cross-dimensional attribute
category
type
Non-aggregability
trademark
Product
holiday
SALES
Sale district
trimester
Date
Quantity
Gross income
Unitary price
Nr. tickets
Shop
year
month
city
county
state
Promotion
discount
publicity
Convergence

Empty fact schemata



A fact schema is empty if there are no measures. In fact, the only measure is the count

Conceptual design

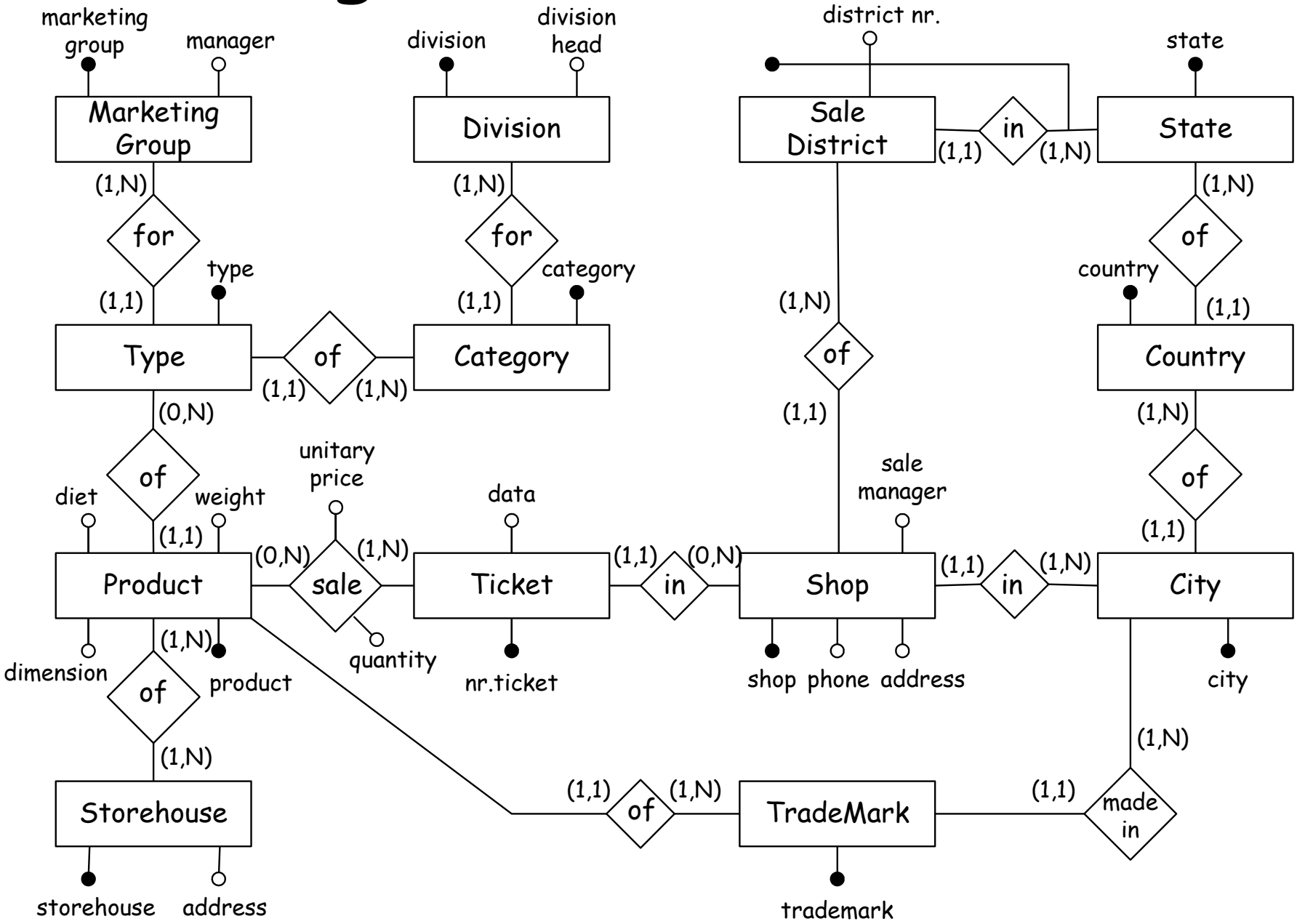
Conceptual design

- Conceptual design takes into account the documentation related to the reconciled database
  - E/R schema
  - Relational schema
  - XML schema

Top-down methodology

1. Fact definition
2. For each fact:
    1. Attribute tree definition
    2. Attribute tree editing
    3. Dimension definition
    4. Measure definition
    5. Fact schema creation

Starting from the E/R schema



Starting from the Relational Schema

Product(product,weight,dimension,trademark:TradeMark,type:Type)
Shop(shop,address,phone,salemanager,(ditrictnr,state):District,city:City)
Ticket(nrticket,date,shop:Shop)
Sale(product:Product,nrticket:Ticket,quantity,unitaryprice)
Storehouse(storehouse,address)
City(city,country:Country)
Country(country,state:State)
State(state)
District(district,state:State)
Prod_Storehouse(product:Product,storehouse:Storehouse)
TradeMark(trademark,madein:City)
Type(type,marketinggroup:MarketingGroup,category:Category)
MarketingGroup(marketinggroup,manager)
Category(category,division:Division)
Division(division,divisionhead)

Fact definition

- Facts correspond to events that dynamically happen in the organization
  - In an E/R schema, it can correspond to an entity F or to an association among *n* entities $E_1, E_2, \ldots, E_n$
  - In a relational schema, a fact corresponds to a relation (table) R

Fact definition

- Good fact candidates: entities or relationships representing **frequently updated archives**
- Static archives: **NO!**
- **Remark**: when a fact is identified, it becomes the root of a new fact schema

Attribute tree definition

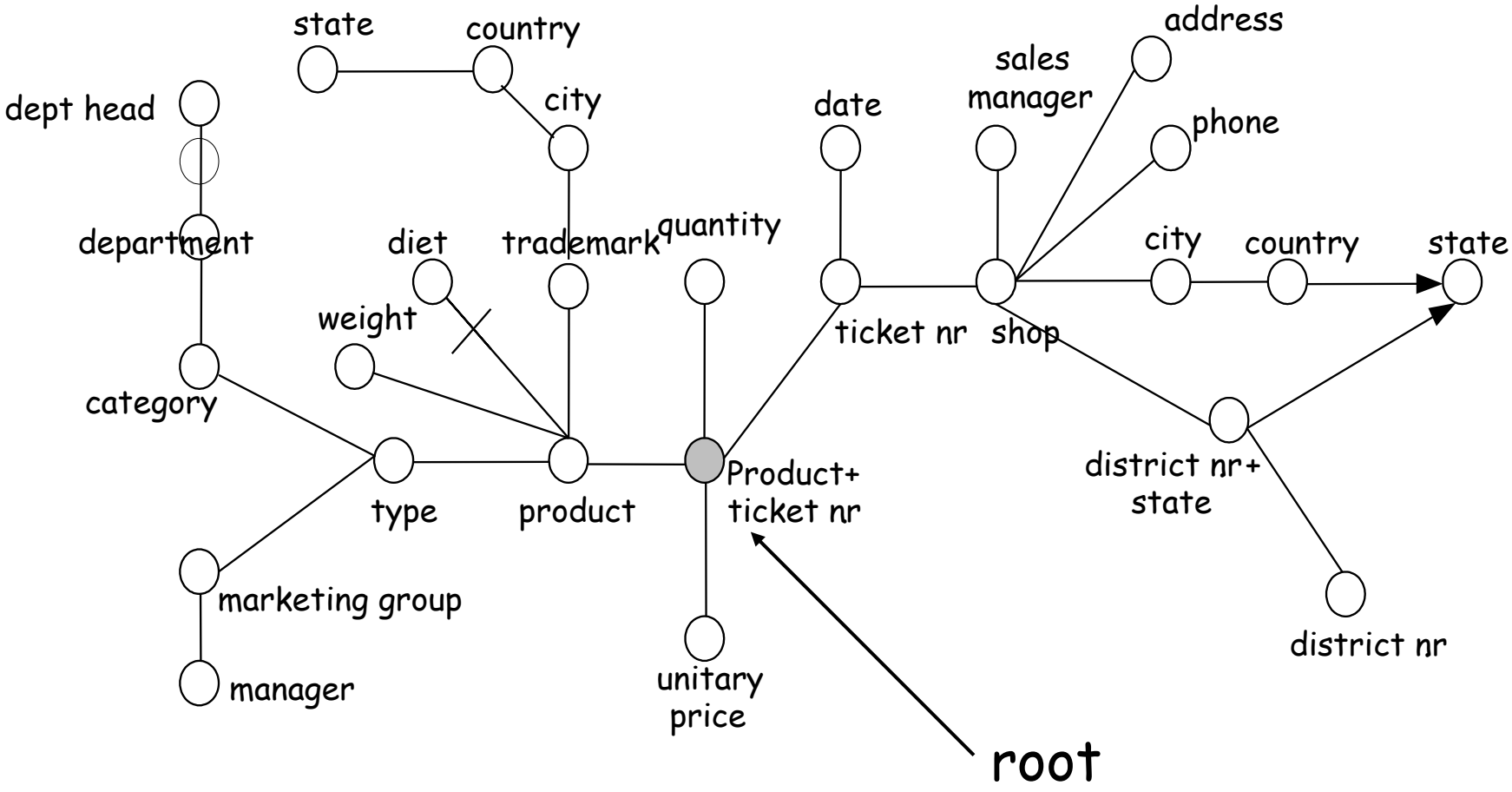
- The attribute tree is composed by:
  - Nodes, corresponding to attributes (simple or complex) of the source schema
  - Root, corresponding to the primary key of F
  - For each node, the corresponding attribute functionally determines its descendant attributes
- The attribute tree can be obtained by using a semi-automatic procedure

Attribute tree definition: procedure

```
Root=newNode(ident(F));
Translate(F, root);


Procedure Translate(E, v):
{for each attribute a of E, a≠Ident(E)
       addChild(v, newNode(a));
 for each entity G connected to E by an association R with
   max(E,R)=1
     { for each attribute b of G
               addChild(v, newNode(b));
               prox = newNode(Ident(G));
               addChild(v, prox);
               Translate(G, prox);
     }
}
```

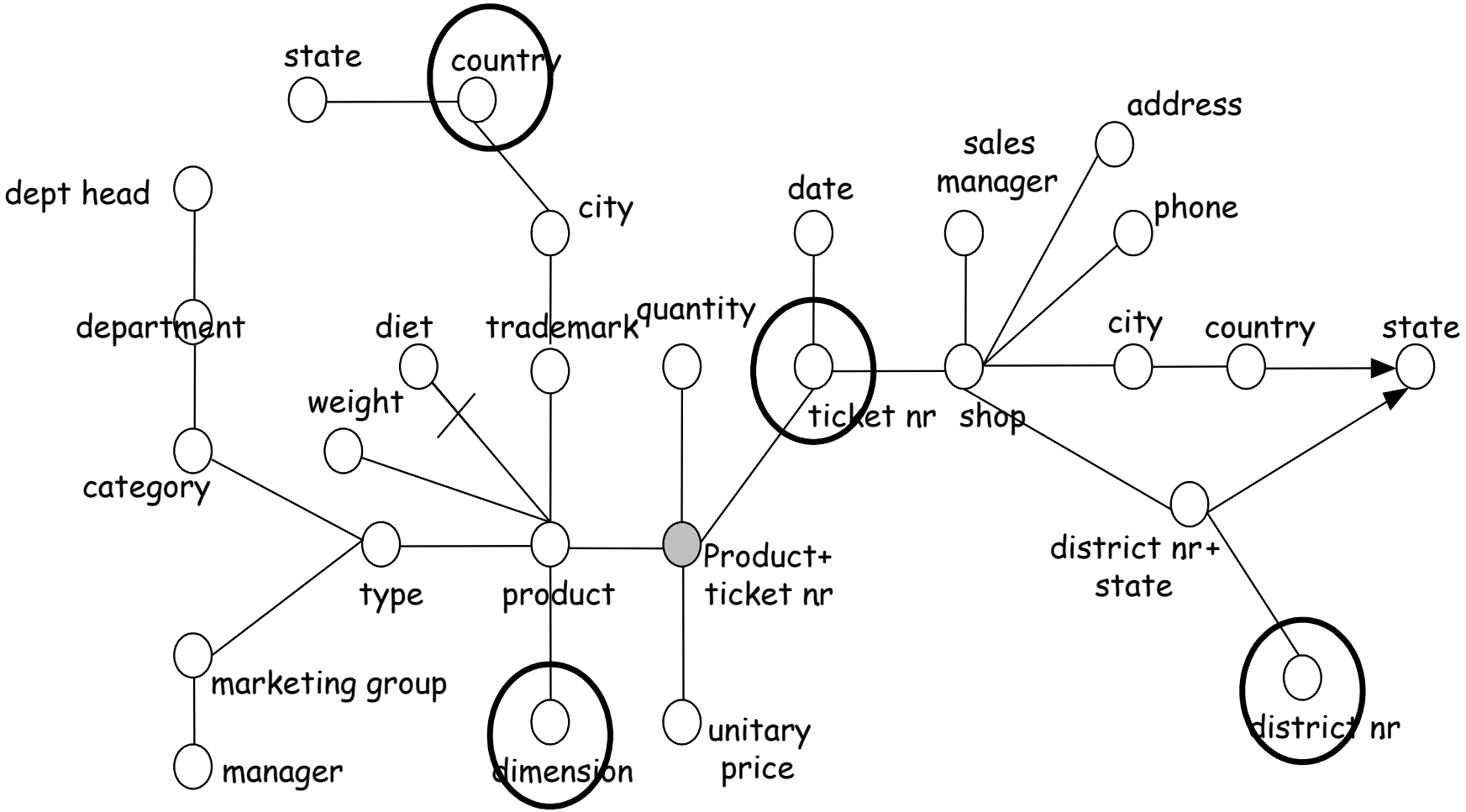
Attribute tree: example



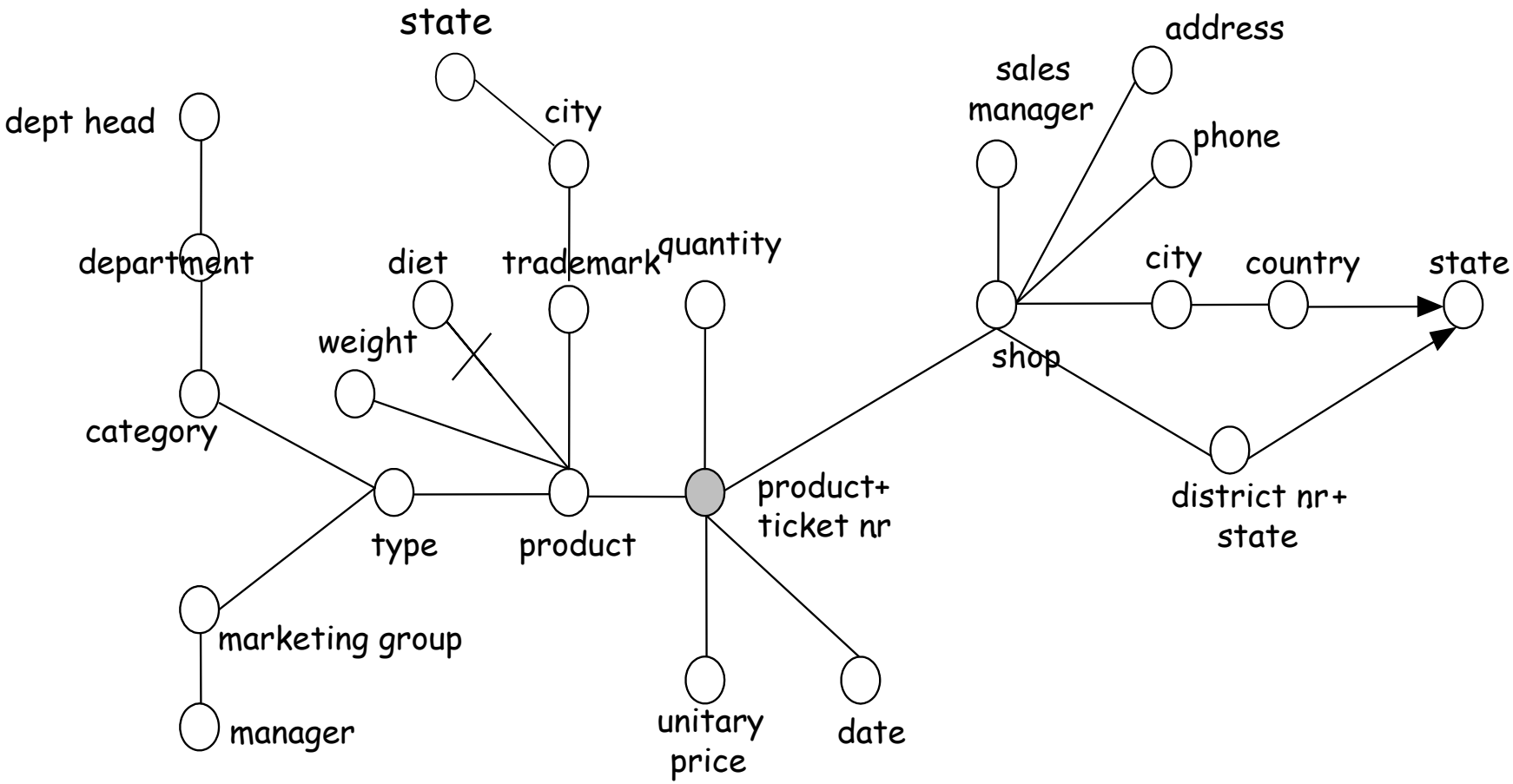
Attribute tree editing

- The editing phase allows to remove some attributes which are irrelevant for the data mart
  - Pruning of a node v: the subtree rooted in v is deleted
  - Grafting of a node v: the children of v are directly connected to the father of v

Attribute tree editing: example



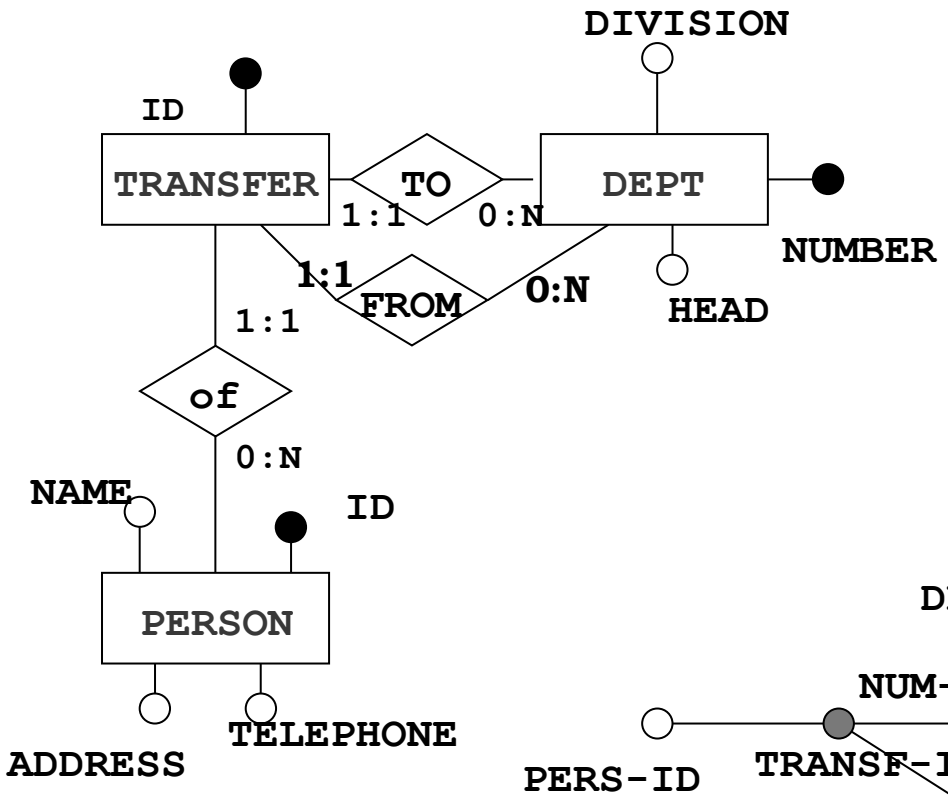
Attribute tree editing: example

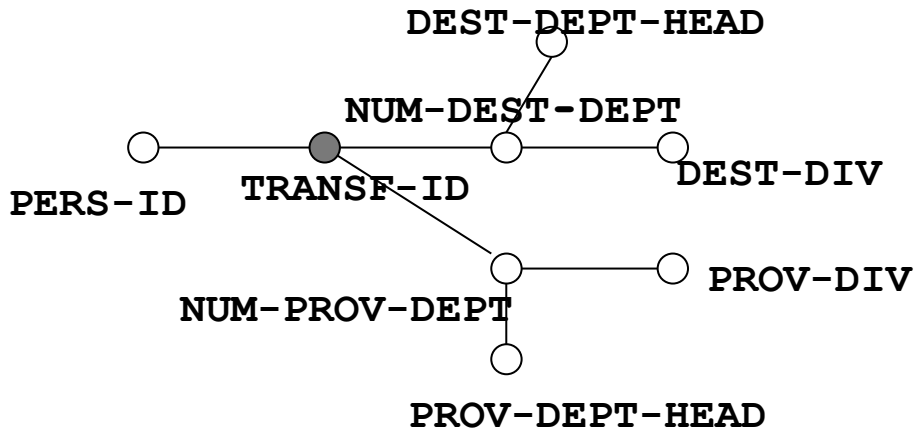


Special cases

- Cyclic relationships: (e.g. part-subpart, employee-manager) → they must be broken after a certain number of iterations
- Cycles in the schema → they must be broken, possibly choosing to keep the most convenient link (see next slide)
- ISA hierarchies: treated like 0-1 relationships (optional)
- Compound attribute: vertex with children

Cycles in the schema

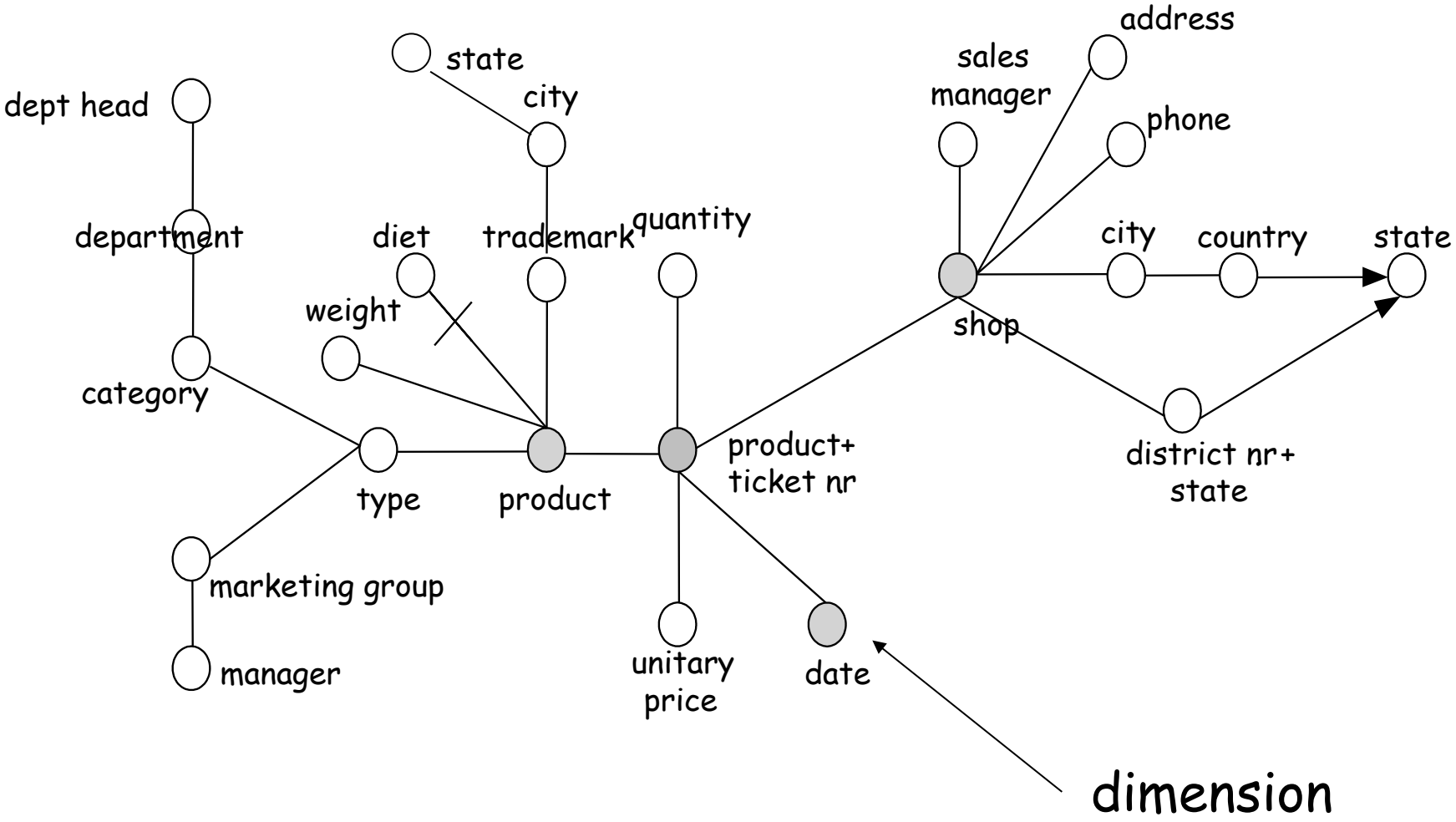




Dimension definition

- Dimensions can be chosen among the children of the root
- Time should always be a dimension
  - Historical source: time is an attribute
  - Snapshot source: not always time is directly represented. In this case it is necessary to add time.

Dimensions definition: example



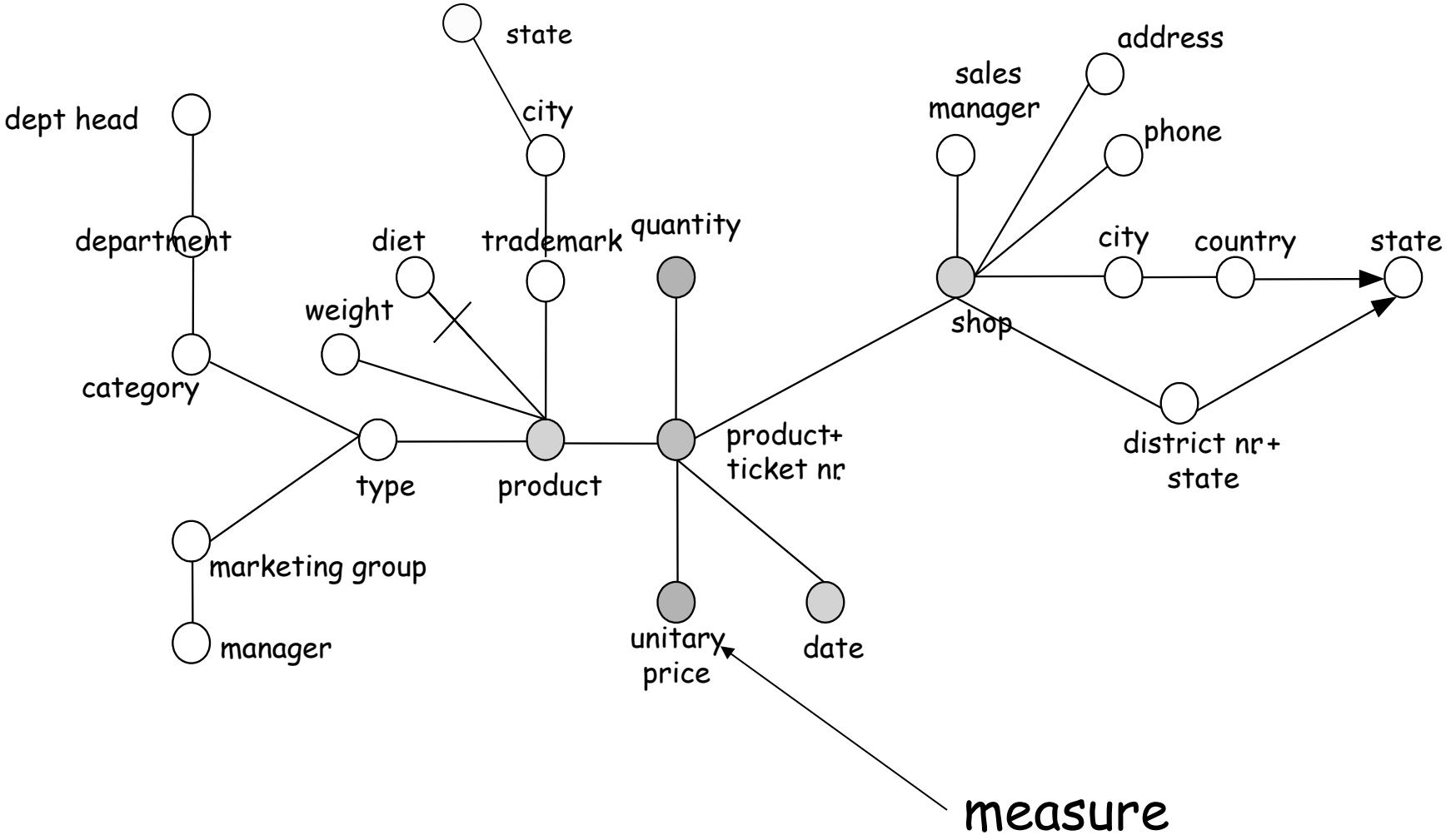
Measure definition

- If the fact identifier (set of attributes) is included in the set of dimensions, then numerical attributes that are children of the root (fact) are measures
- Further measures are defined by applying aggregate functions to numerical attributes of the tree
  - Generally: sum, average, min, max, count

Measure definition (2)

- It is possible that a fact has no measures (empty)
- If the granularity of a fact is different w.r.t. the granularity of the source schema, it can be useful to define suitable measures in order to aggregate the same attribute by using different operators

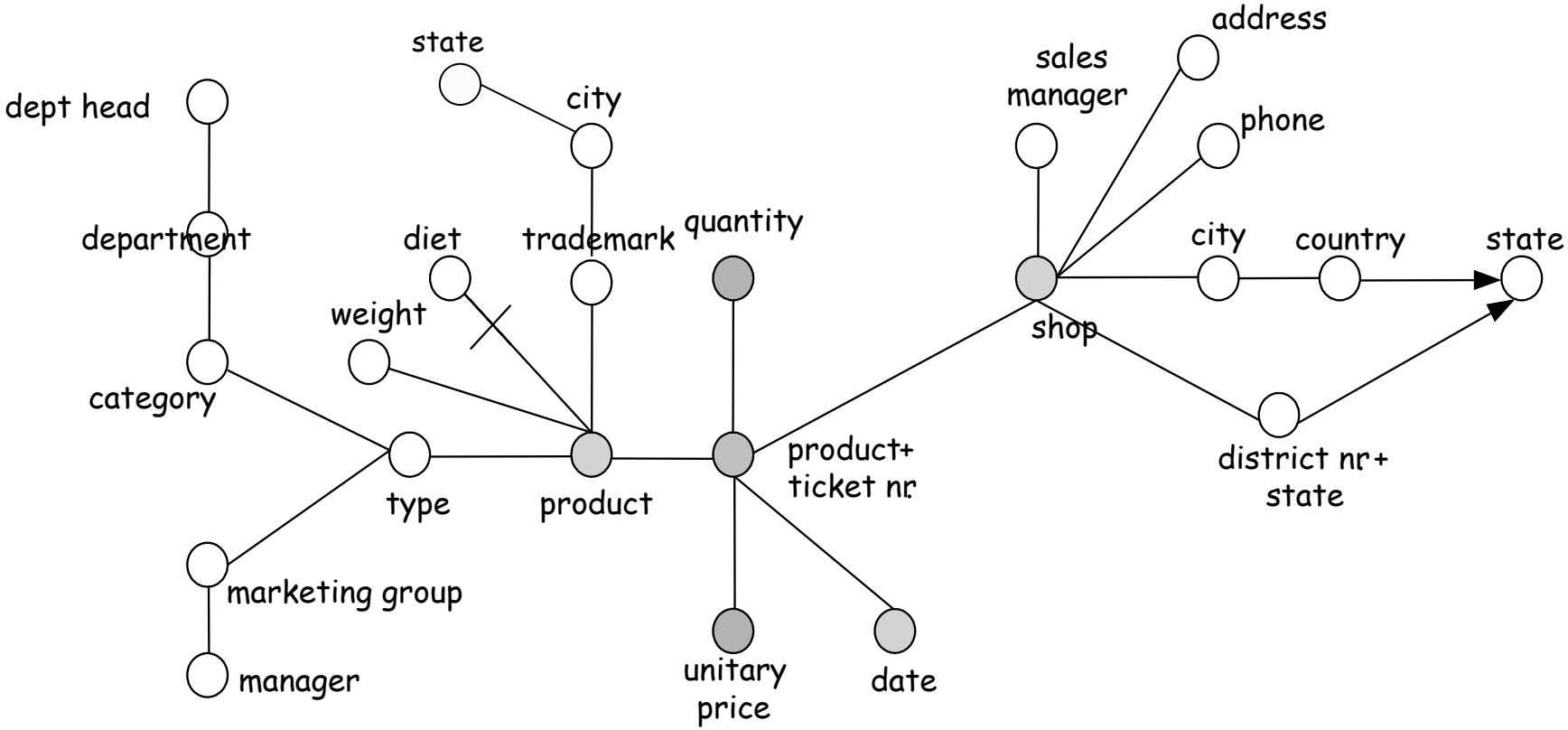
Measure definition: example



Glossary

- In the glossary, an expression is associated to each measure
  - The expression describes how it is possible to obtain the measure starting from the attributes of the source schema

Measure definition: example



Quantity = SUM(Sale.quantity)
Gross income=SUM(Sale.quantity*Sale.unitaryprice)
Unitary price=AVG(Sale.unitaryprice)
Nr-tickets=COUNT(*)

Fact schema creation

- The attribute tree is translated into a fact schema including dimensions and measures
  - Dimension hierarchies correspond to subtrees having as roots the different dimensions (with the least granularity)
  - The fact name corresponds to the name of the selected entity

Fact schema creation: example