Business Intelligence

Data analysis

OLTP

- Operating environment:
  - “Online” data management
  - The goal is to manage transactions that modify the data

On Line Transaction processing (OLTP)

OLAP

- Analysis environment:
  - “Static” data management
  - The goal are queries and statistical analyses

On Line Analytical Processing (OLAP)

Problems

- The promise of relational technology:
  - Flexible data access
    - One tool for the final user allowing all sorts of queries
- But the promise is not kept:
  - Emphasis on OLTP and complexity of applications
  - Little use of data for managing strategic decisions

OLTP: On Line Transaction Processing

- Traditional transaction management that implements the operating processes of a firm
  - Predefined and relatively simple operations
  - Every operation involves just a “small” amount of data
  - Data are detailed and up to date
  - “ACID” properties of transactions are essential

ACID properties

- Atomicity
- Consistency
- Isolation
- Durability

OLAP: On Line Analytical Processing

- Management of operations for decision support:
  - Complex and non-predefined operations
  - Every operation can involve lots of data
  - Aggregated, historical data, maybe not up to date
  - ACID properties irrelevant: read-only operations

OLTP vs OLAP

- It is very difficult to manage systems that need to do both OLAP and OLTP at the same time
- Only one of them is doable
- Several reasons:
  - Inhomogeneity of users and requirements
  - Technical reasons

OLTP and OLAP

	OLTP	OLAP
User	Employee	Manager
Function	Daily operations	Decision support
Design	Application-oriented	Data-oriented
Data	Current, up-to-date, detailed, relational, homogeneous	Historical, aggregated, multidimensional, heterogeneous
Usage	Repetitive	Non-repetitive
Access	Read-write, random	Read-only, sequential

OLTP and OLAP

	OLTP	OLAP
Work unit	Short transaction	Complex query
Size (# of records)	Dozens	Millions
# of users	Thousands or more	Hundreds or less
Max Size	a few GBs	1TB… and more
Metrics	Throughput	Time to answer

Technical reasons behind the OLTP/OLAP conflict

- Use of indexes
  - OLTP: few indexes, and only if needed
  - OLAP: many indexes, for every need
  - OLTP+OLAP
    - Either OLTP transactions slow down in order to update the many indices
    - Or OLAP queries cannot use the required indices

Technical reasons behind the OLTP/OLAP conflict

- Lock conflicts
  - OLTP: many quick transactions with exclusive locks
  - OLAP: few long transactions with shared locks
  - OLTP+OLAP
    - Either OLTP transactions are severely slowed down
    - Or OLAP queries cannot be executed

Technical reasons behind the OLTP/OLAP conflict

- Pre-computation of queries
  – OLTP: very rare, for consistency and load problems
  – OLAP: key aspect for improving response times
- Differences in the logical model
  – OLTP: high fragmentation and large number of tables
  – OLAP: few non-normalized tables
- Different join algorithms
  – OLTP: all join algorithms are possible
  – OLAP: only algorithms with indices make sense

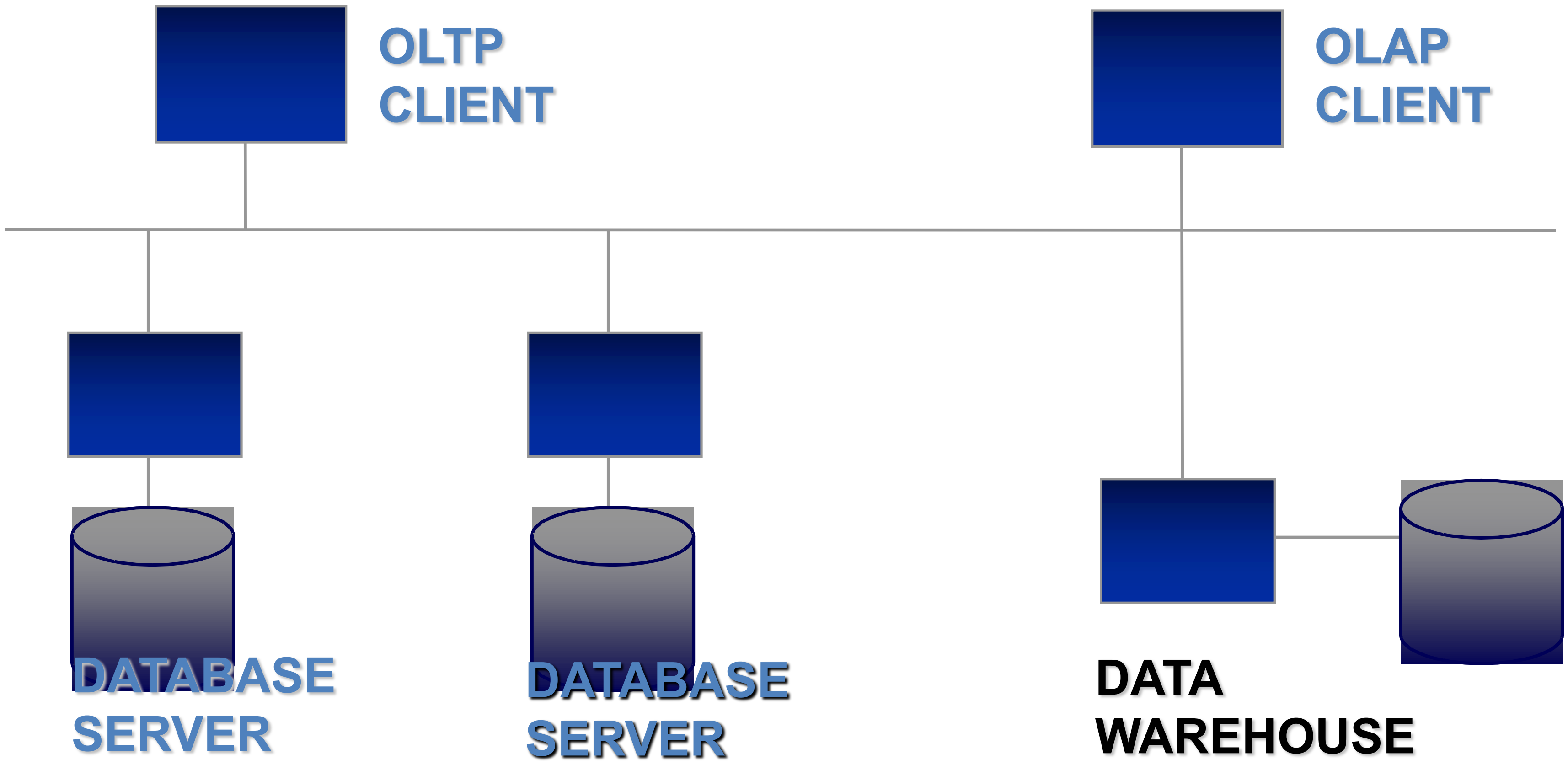
Observations on the OLTP/OLAP conflict

- The conflict is inherent
- It cannot be resolved by increased computing power
- The best solution is to separate OLAP from OLTP
  - This leads to the notion of

    Data Warehouse

- Asynchronousness of updates is essential

Interaction between OLTP and OLAP



Data Warehouse

Data warehouse: an environment for analysis

- DATA WAREHOUSE:
  - An organized description of all the data necessary for a strategic analysis of the behavior of a company
  - Techniques:
    - Multi-dimensional analysis
    - Data mining

Data warehouse

- A database
- Mainly used to support management decisions
  - Integrated: company-wide, not department-wide
  - Data oriented, not application-oriented
  - Historical, with a large timespan and (usually) explicit time points
  - Not volatile: data are loaded and accessed offline
  - Maintained separately from the operational databases

Data warehouse: integrated

- Interesting data come from all the sources of information
  - Each piece of data comes from one or more sources
- A data warehouse represents data univocally, by reconciling heterogeneity in the different representations
  - Names
  - Encoding
  - Multiple representation

Data warehouse: data oriented

- Operational databases are built to support the single operational processes and applications
  - Production
  - Sales
- Data warehouses are built around the main entities of the firm
  - Products
  - Customers

Data warehouse: historical data

- Operational databases keep information updated with their current value
- Timespan of interest: few months

- In a data warehouse one is interested in the historical evolution of information
- Timespan of interest: years

Data warehouse: non volatile

- In an operational database data are
  - Accessed, inserted, modified, deleted
- Few records at a time

- In a data warehouse we have
  - “daily” access and query operations
  - “nightly” data load and update operations
  - These involve millions of records

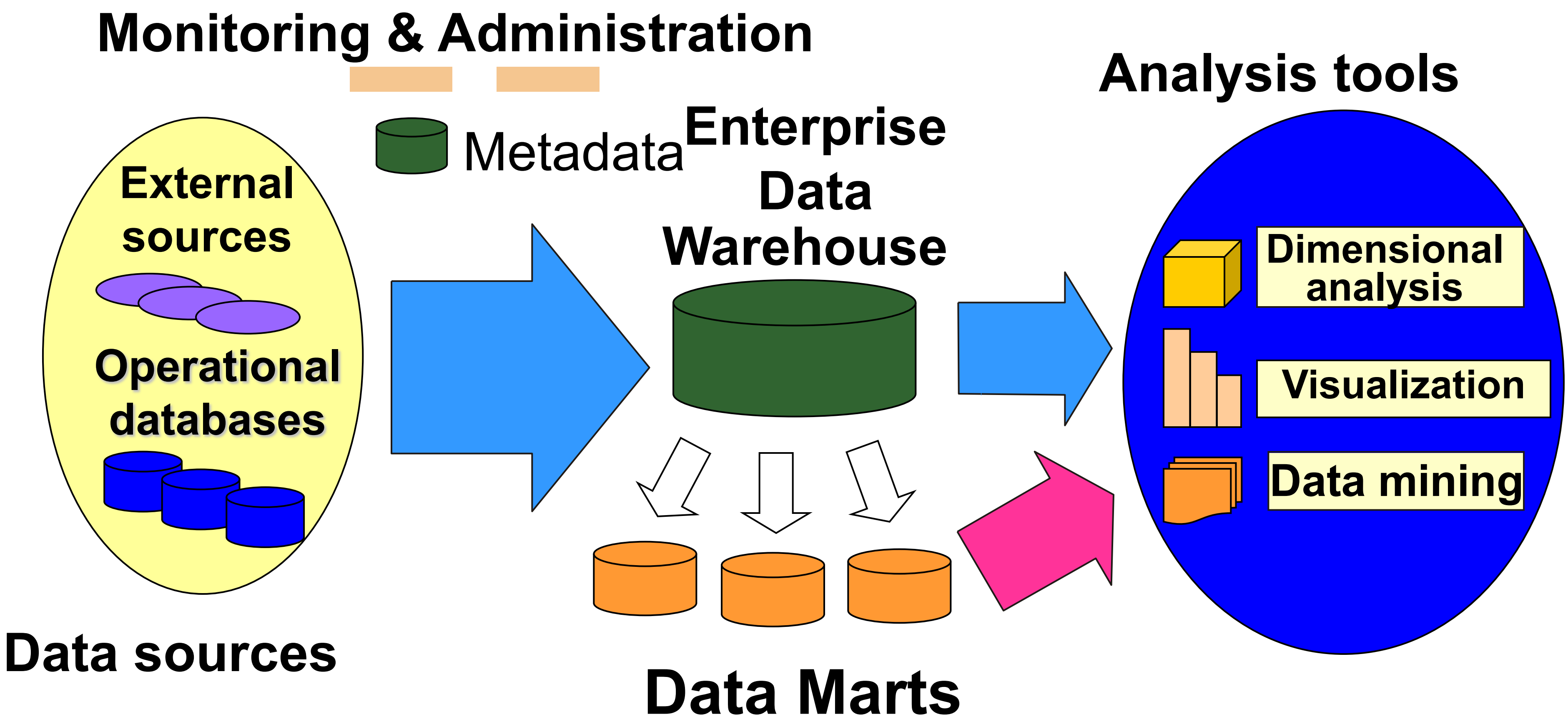
Data warehouse: a separated database

- Many reasons
  - Technical reasons, already discussed
  - There is no single operational database that contains all the data of interest
  - The databases must be integrated
  - The data of interest would anyhow be different
    - Historical data must be maintained
    - Aggregated data must be maintained
  - Data analysis requires a specific organization of the data as well as specific access methods
  - Lack of separation causes degraded performance

Critical success factors

- Replicating data without affecting the transactional system
- Loading data within the assigned timeframe
- Scalability of solution
- Presentation (users accept the system)
- Correctness of replicated data
- Use of standards
- Coherence between data model and reality

Architectures for data warehousing



DW and Data Mart

- A DW often integrates different Data Marts
- Users normally use a particular Data Mart
- Data are shared across different Data Marts
- Every Data Mart takes care of a particular aspect of the firm

Multidimensional model

Star schema model

- Normally, one uses the star schema model (aka multi-dimensional model)
- It's a conceptual model with restrictions
- Advantages:
  - Appropriate query interfaces can be used
  - Good performance
  - Immediate definition of the logical schema
- Experience confirms the effectiveness of this solution

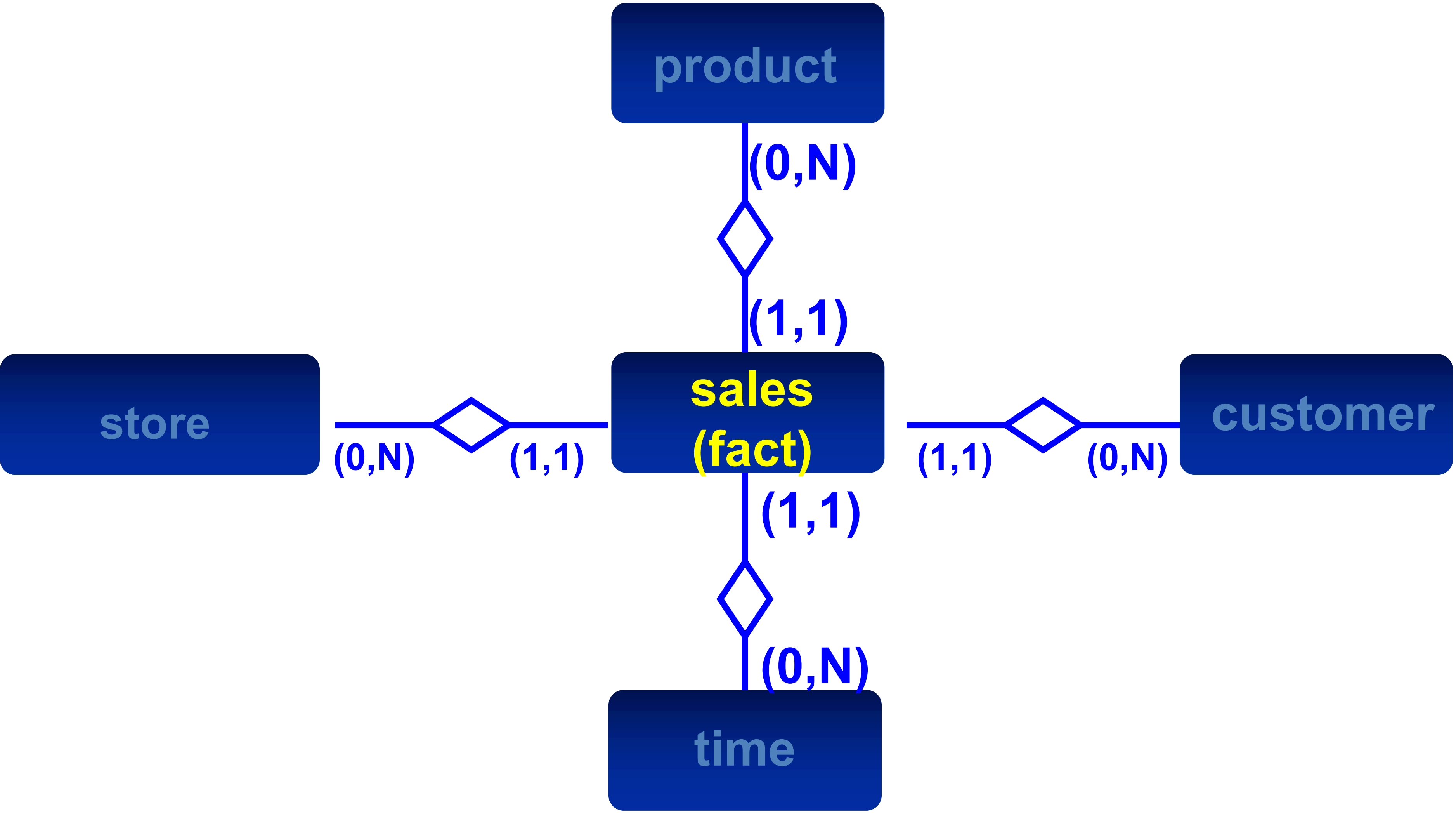
Multidimensional representation

- Relevant notions:
  - fact
    - a concept on which the analysis is centered
  - measure
    - an atomic property of a fact to be analyzed
  - dimension
    - description of a perspective for the analysis

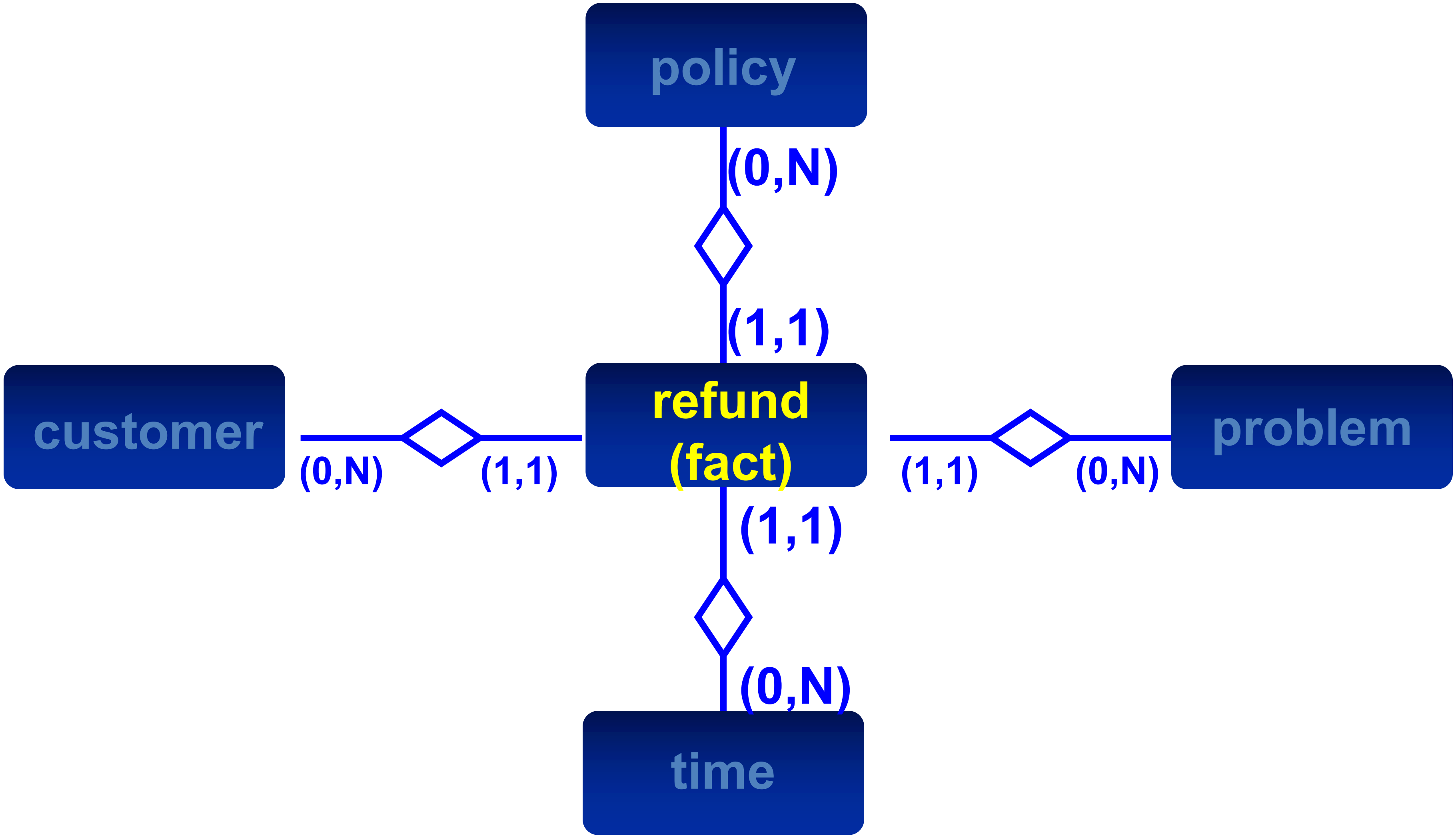
Examples of facts/measures/dimensions

- Chain stores
  - fact: sales
  - measures: sales quantity, revenue
  - dimensions: products, time, zone
- Telephone company
  - fact: call
  - measures: cost, duration
  - dimensions: caller, answerer, time

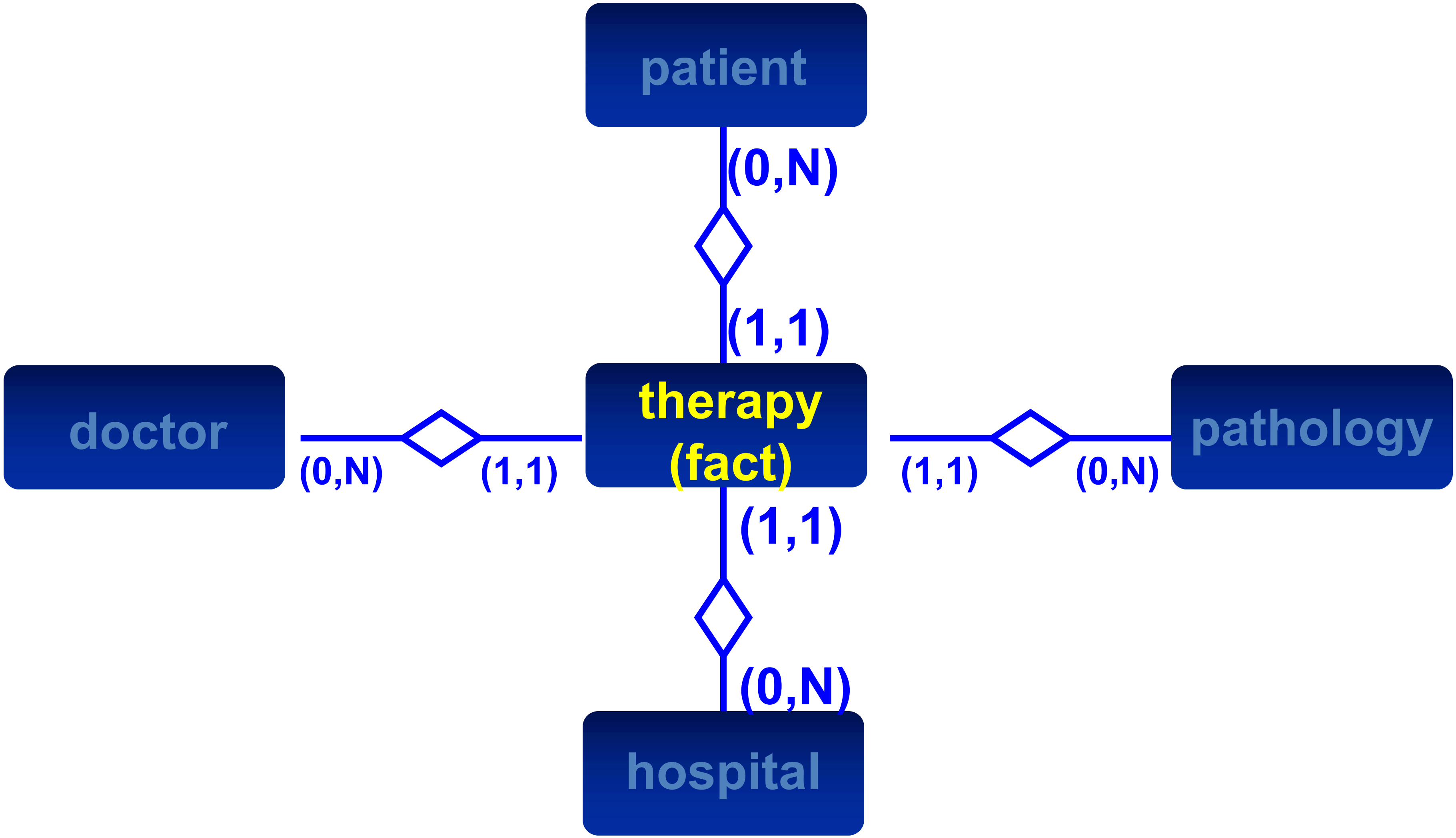
Example : sales management



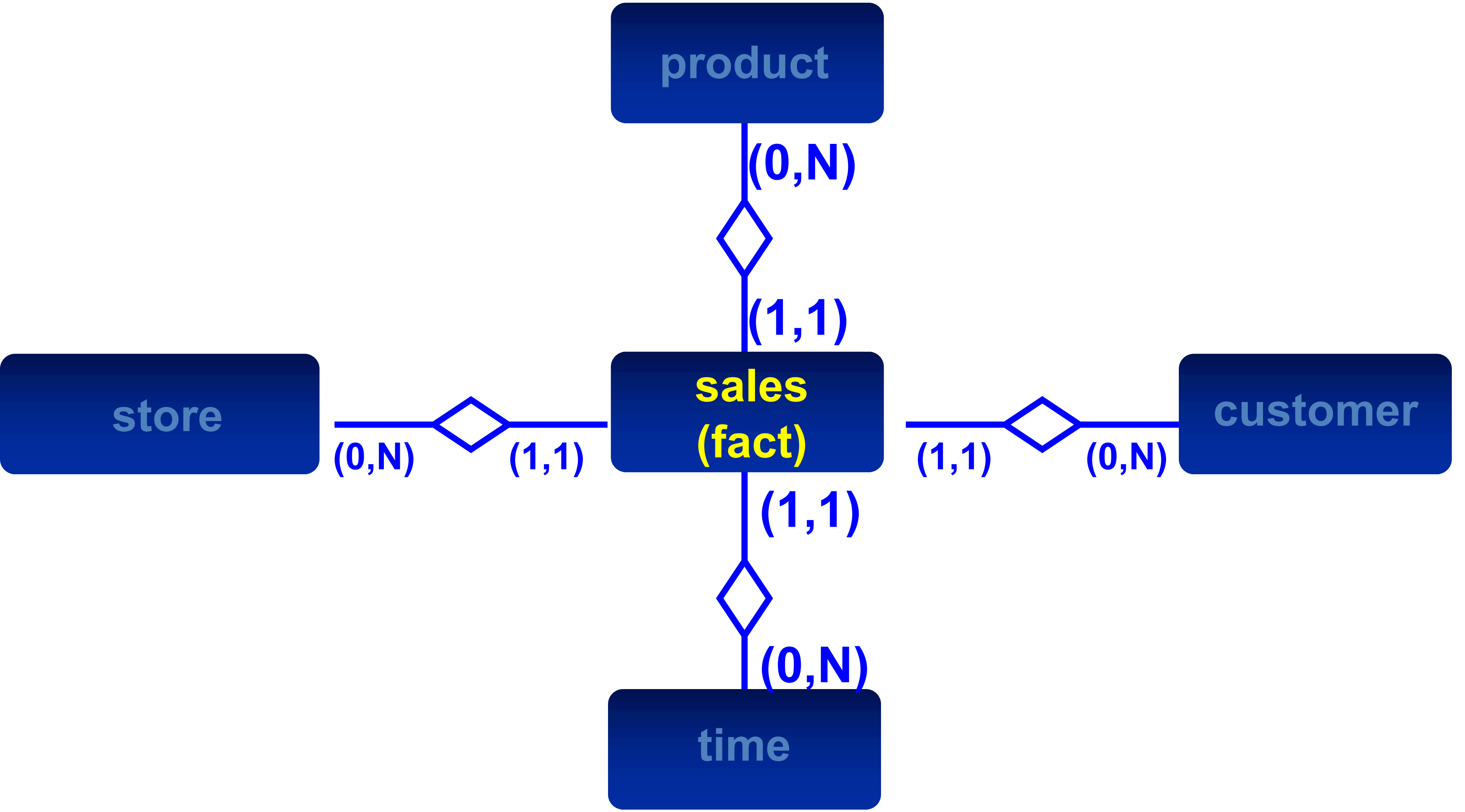
Example: reimbursement management



Example: therapy management



Back to: sales management



Needs of an enterprise

- In supermarkets, the person in charge needs to analyze the factors that affect sales:
  - Product types
  - Time instant of the sale
  - Characteristics of the store
  - Customers

Sales (Fact)

<u>TIME-CODE</u>
<u>STORE-CODE</u>
<u>PRODUCT-CODE</u>
<u>CUSTOMER-CODE</u>
QUANTITY
REVENUE

Product dimension

<u>PRODUCT-CODE</u>
NAME
COLOR
MODEL
CATEGORY-CODE
CATEGORY
…

Store dimension

<u>STORE-CODE</u>
NAME
ADDRESS
CITY-CODE
CITY
AREA-CODE
AREA
STATE-CODE
STATE
…

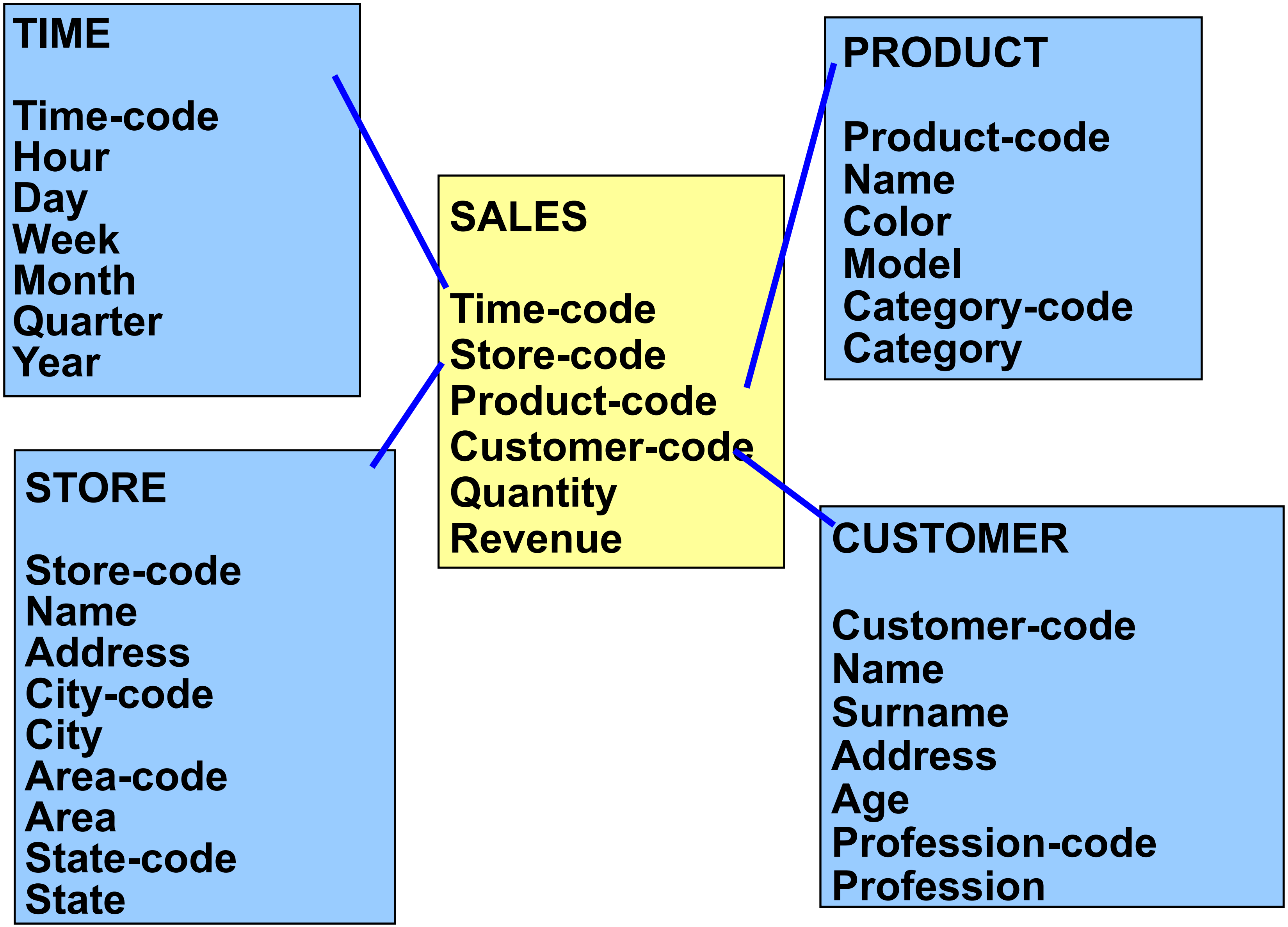
Time dimension

<u>TIME-CODE</u>
HOUR
DAY
WEEK
MONTH
QUARTER
YEAR
…

Customer dimension

<u>CUSTOMER-CODE</u>
NAME
SURNAME
ADDRESS
AGE
PROFESSION-CODE
PROFESSION
…

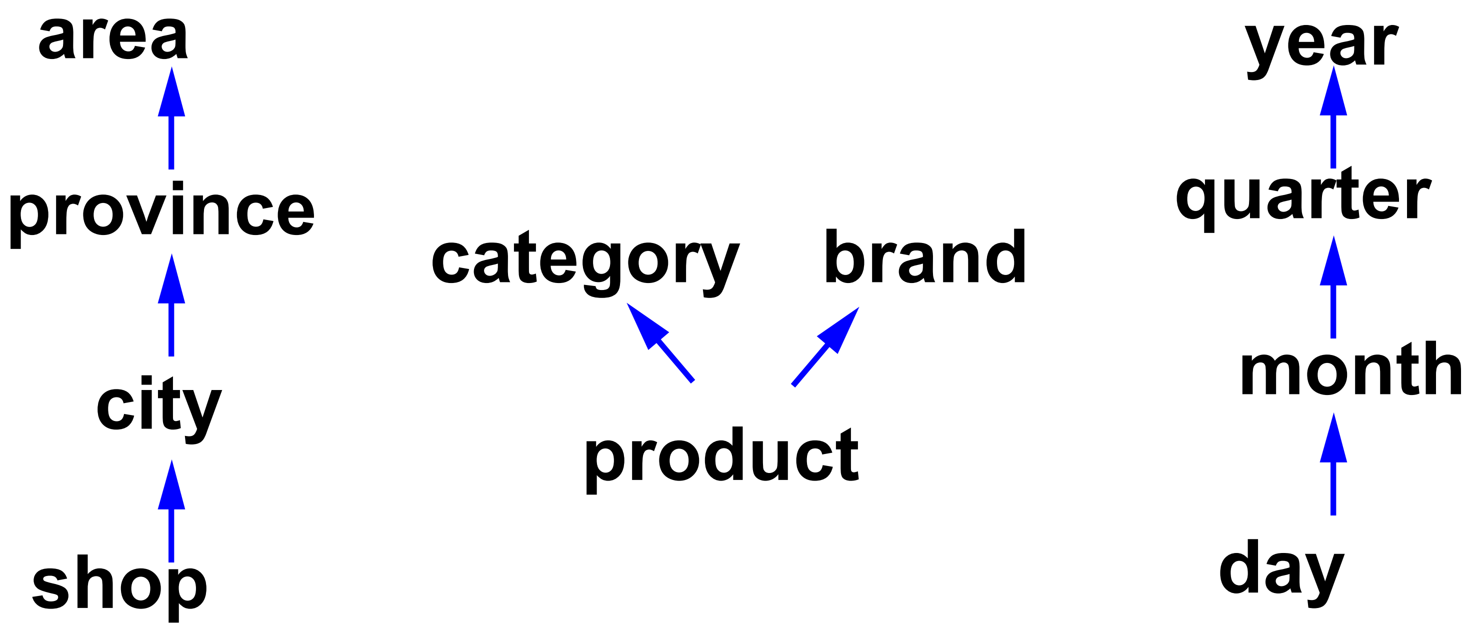
Star schema



TIME
Time-code
Hour
Day
Week
Month
Quarter
Year
SALES
Time-code
Store-code
Product-code
Customer-code
Quantity
Revenue
PRODUCT
Product-code
Name
Color
Model
Category-code
Category
STORE
Store-code
Name
Address
City-code
City
Area-code
Area
State-code
State
CUSTOMER
Customer-code
Name
Surname
Address
Age
Profession-code
Profession

Dimensions and hierarchical levels

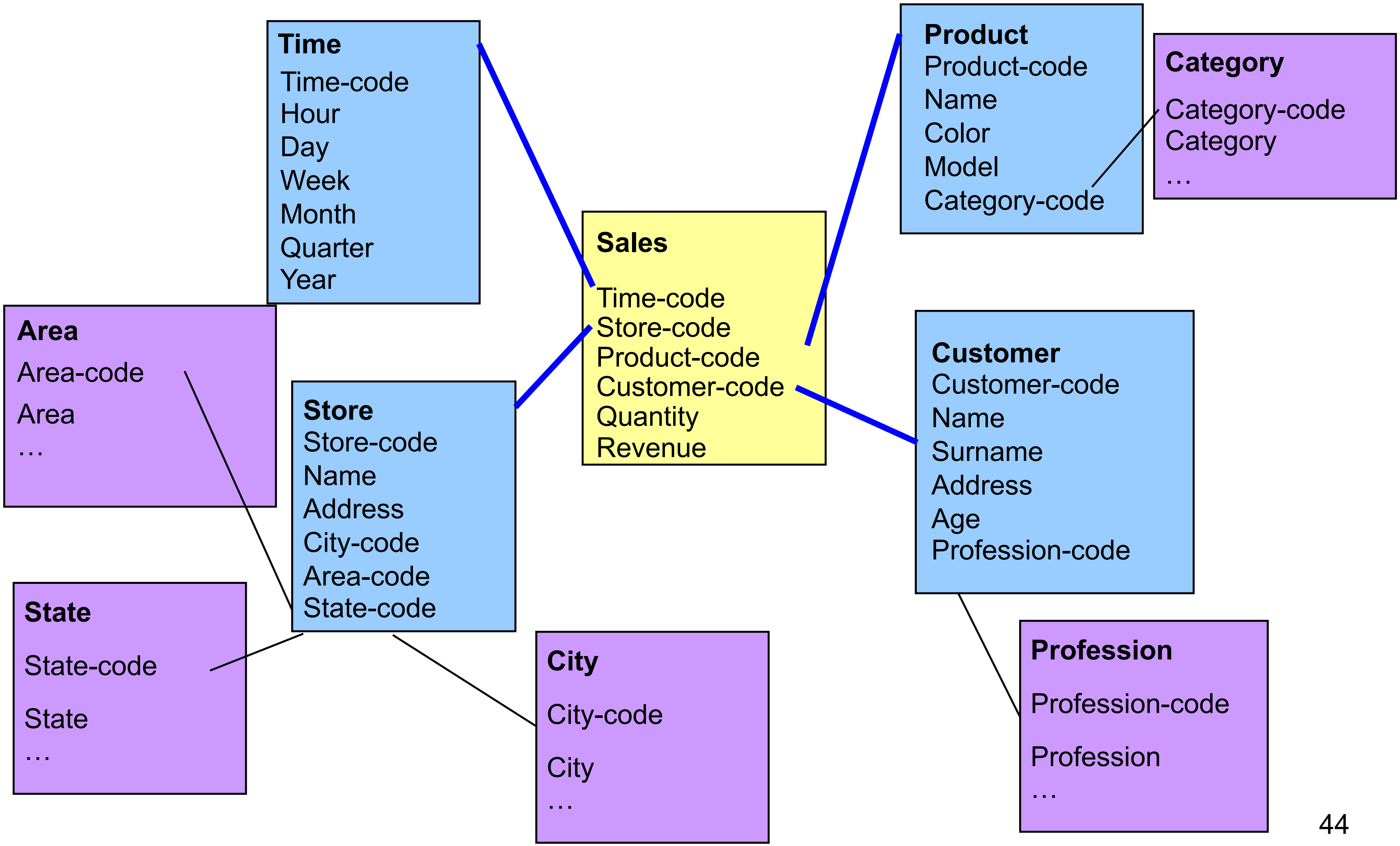
- Each dimension is organized in a hierarchy representing the possible aggregation levels for the data



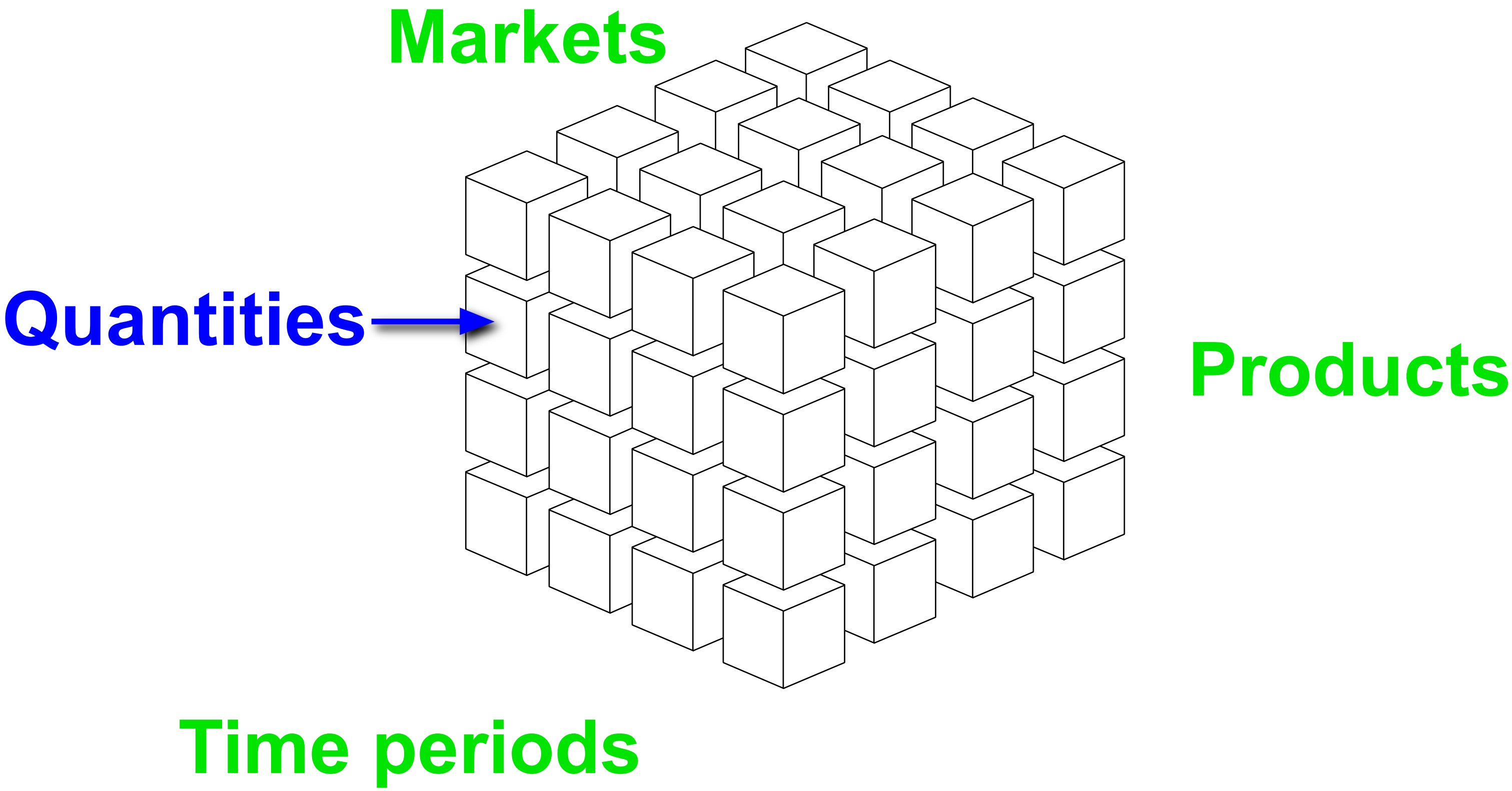
Snowflake model

- Extension of the star schema model
- Allows avoiding excessive redundancies in dimensions
- From the fact table one can reach all the dimension tables by moving along n:1 relationships

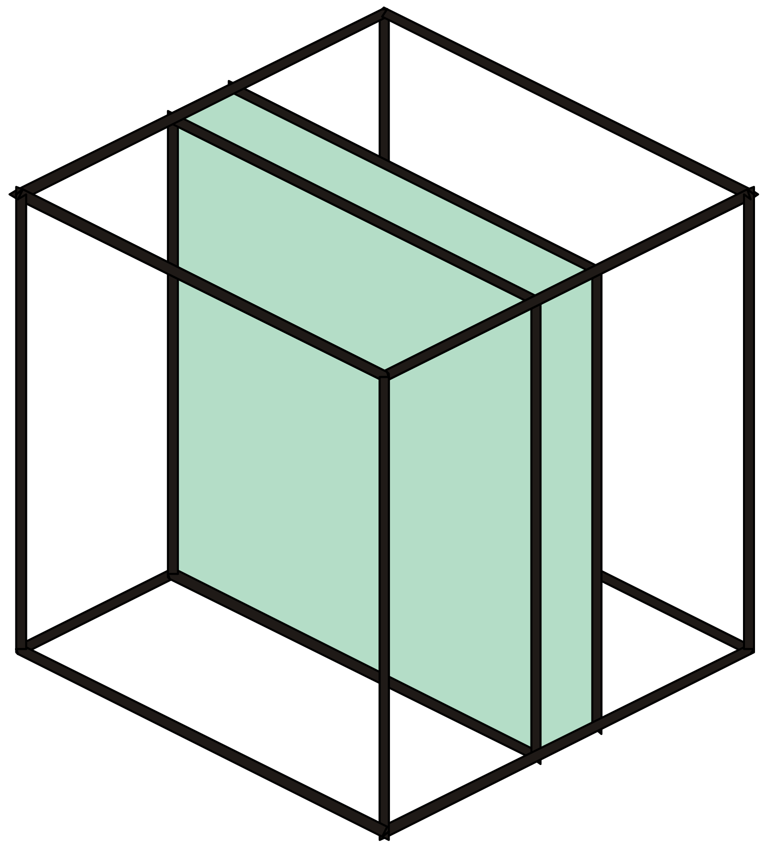
“Snowflake” model



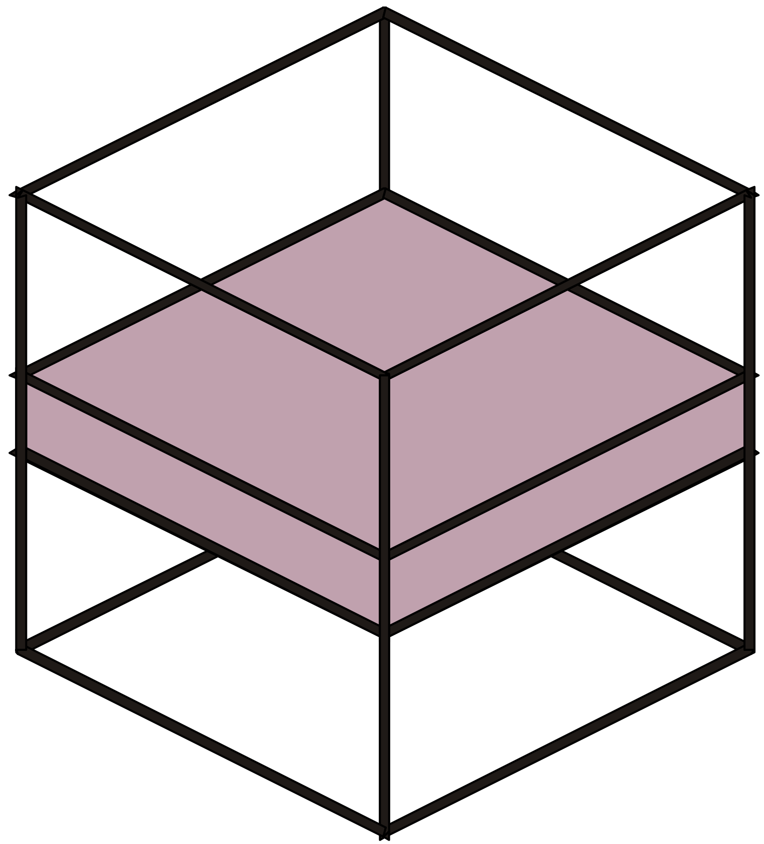
Multidimensional data representation



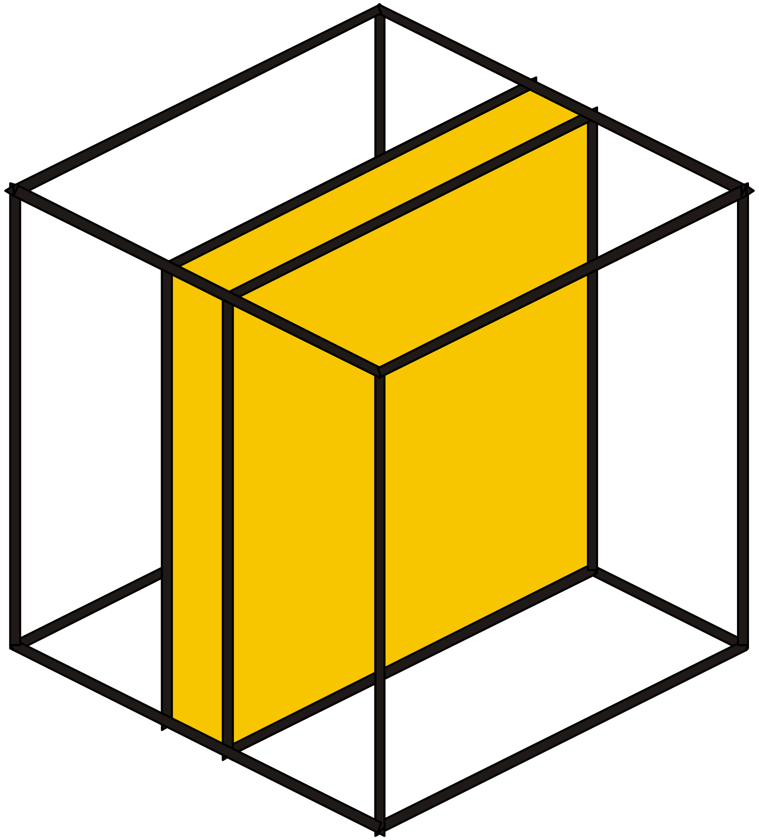
Area managers analyze product sales in all periods in their markets



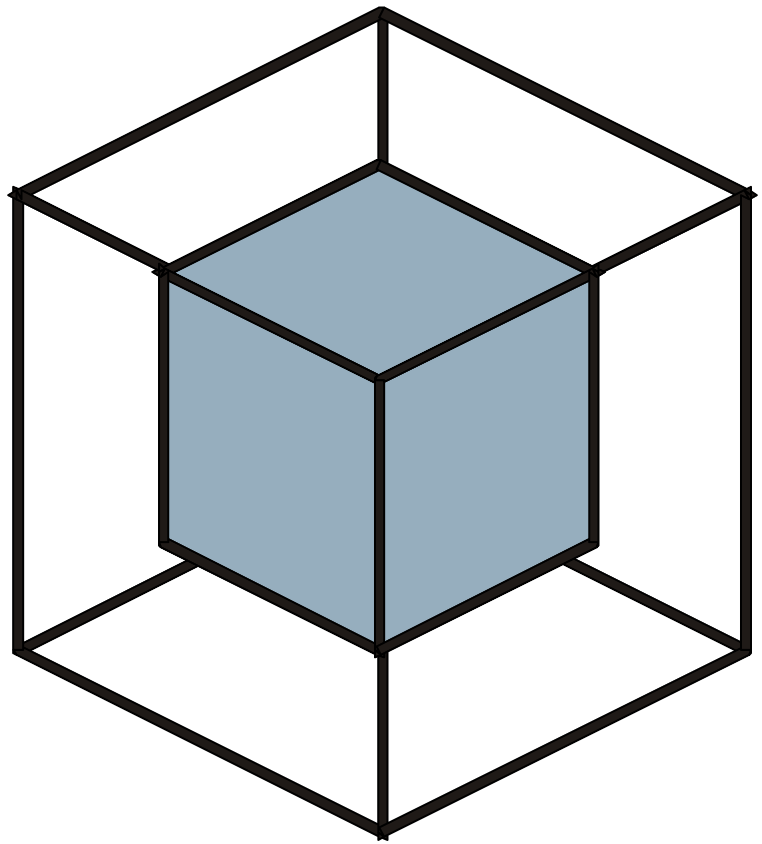
Product managers analyze product sales in all periods and in all markets for their products



Financial managers analyze product sales in all markets in the previous and current period



Strategic manager focus on one product category, an area, and an average time span



Data visualization

- Data are graphically rearranged in order to be easily understandable

- One uses:
  - Tables
  - Charts (bar charts, pie charts, bubble charts, 3D surfaces, …)

Example: query via browser

promotion	time	zone	product	observation
Buy 3 pay for 2 offer (40%) offer (20%) gift (...) …..	january february march	north east center	milk bread pasta	revenue quantity
	february/ april		pasta	sum(revenue) sum(quantity)

Same query in SQL

```
select T.Month, P.Name,
       sum(revenue), sum(quantity)

from Sales S, Time T, Product P

where S.TimeCode = T.TimeCode
 and S.ProductCode=P.ProductCode
 and T.Month between Feb and Apr
 and P.Name = "pasta"

group by T.Month,P.Name
order by T.Month,P.Name
```

Columns in the output

Tables involved

Filters (joins and selections)

Aggregation and order of results

Same query generalized

```
select  c1, c2,
        aggr(c3), aggr(c4)
```

Columns in the output

```
from facts, dim1, dim2
```

Tables involved

```
where join-pred(facts,dim1)
  and join-pred(facts,dim2)
  and select-pred(dim1)
  and select-pred(dim2)
```

Filters (joins and selections)

```
group by c1, c2
order by c1, c2
```

Aggregation and order of results

Result

month	name	Sum of revenues	Sum of quantities
february	pasta	130.000.000	45.000
march	pasta	140.000.000	50.000
april	pasta	135.000.000	51.000

Operations on multidimensional data

- Roll up — aggregating data
  - total sales from last year by category of product and region
- Drill down — disaggregating data
  - for a particular category of product and region, show detailed daily sales by shop
- Slice & dice — selecting and projecting
- Pivot — re-orienting the cube

Drill-down: adding a dimension

- Drill-down on the zone

month	name	zone	Sum of quantities
february	pasta	north	15.000
february	pasta	east	17.000
february	pasta	center	13.000
march	pasta	north	18.000
march	pasta	east	18.000
march	pasta	center	14.000
april	pasta	north	18.000
april	pasta	east	17.000
april	pasta	center	16.000

Roll-up: eliminating a dimension

- Roll-up on the month

product	zone	Sum of quantities
pasta	north	51.000
pasta	east	52.000
pasta	center	43.000

Aggregate queries

- Examples:
  - Total sales by category of product, by store and by day
  - Overall monthly sales by store
  - Average of monthly sales by category on all supermarkets

Aggregates in SQL: data cube

- One can express all possible aggregations of tuples in a table
- A new, polymorphic value is available: ALL

Data cube in SQL

```
select Model, Year,
       Color, sum(Quantity)
from Sales
where Model in {'Fiat','Ford'}
  and Color = 'Red'
  and Year between 1994 and 1995
group by Model, Year, Color
with  cube
```

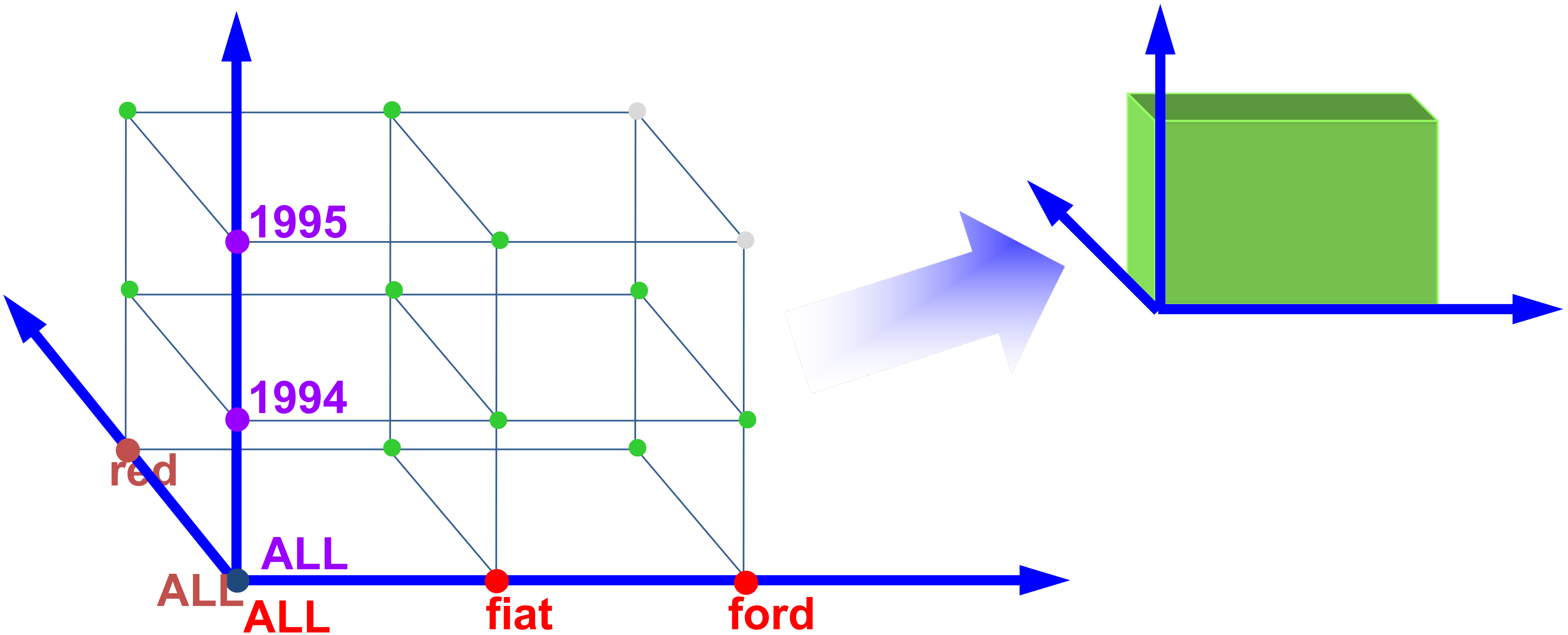
Relevant facts

Model	Year	Color	Quantity
fiat	1994	red	50
fiat	1995	red	85
ford	1994	red	80

Data in the cube

Model	Year	Color	Sum(Quantity)
fiat	1994	red	50
fiat	1995	red	85
fiat	1994	ALL	50
fiat	1995	ALL	85
fiat	ALL	red	135
fiat	ALL	ALL	135
ford	1994	red	80
ford	1994	ALL	80
ford	ALL	red	80
ford	ALL	ALL	80
ALL	1994	red	130
ALL	1995	red	85
ALL	ALL	red	215
ALL	1994	ALL	130
ALL	1995	ALL	85
ALL	ALL	ALL	215

Data cube visualization



Roll up

```
select Model, Year,
       Color, sum(Quantity)
from Sales
where Model in {'Fiat','Ford'}
  and Color = 'Red'
  and Year between 1994 and 1995
group by Model, Year, Color
with rollup
```

Data in the roll up

Model	Year	Color	Sum(Quantity)
fiat	1994	red	50
fiat	1995	red	85
ford	1994	red	80
fiat	1994	ALL	50
fiat	1995	ALL	85
ford	1994	ALL	80
fiat	ALL	ALL	135
ford	ALL	ALL	80
ALL	ALL	ALL	215

Size of a DW: an example

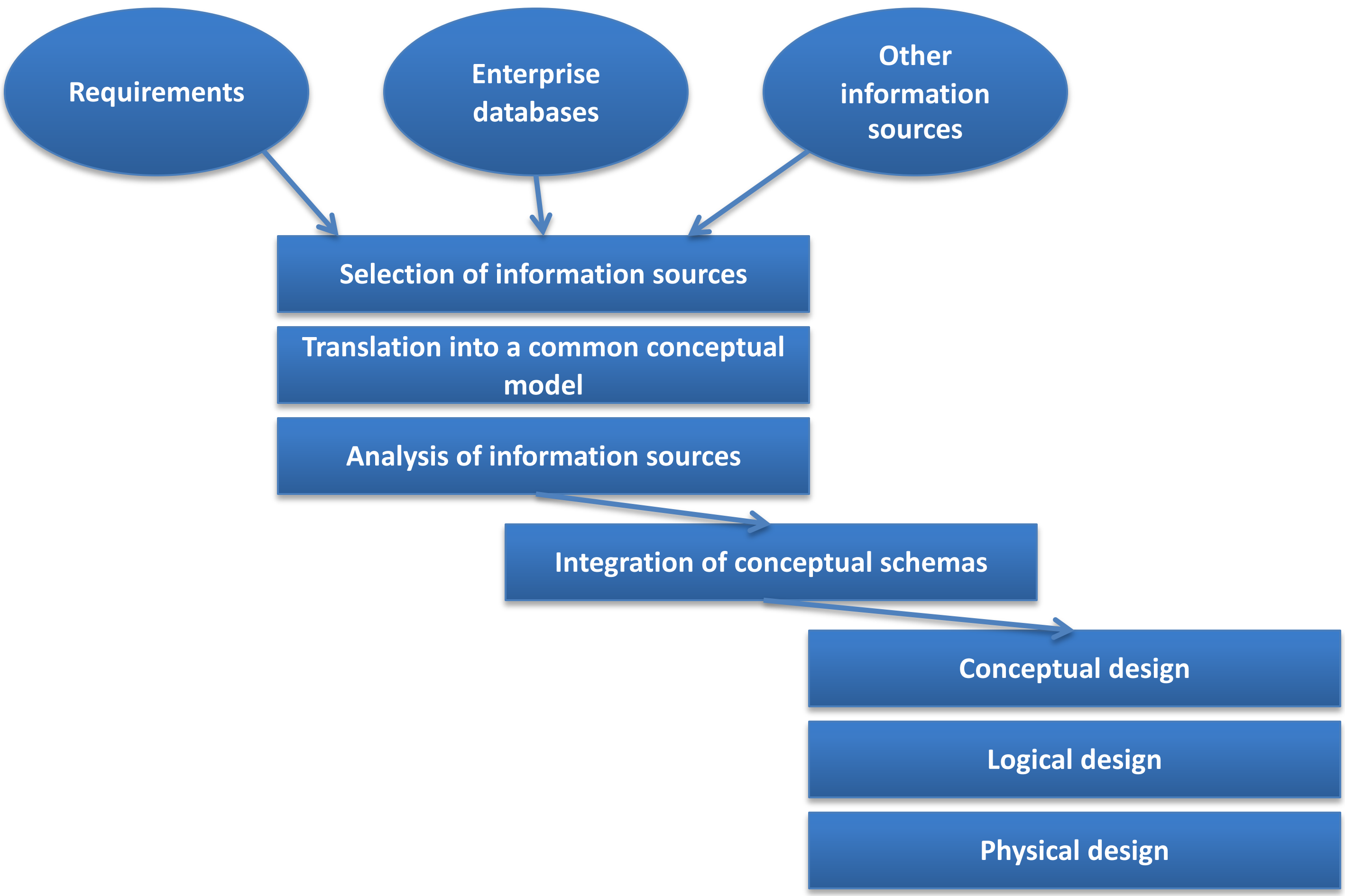
- Time: 730 days (= 2 years)
- Stores: 300
- Products: 10,000
- Average daily sales per product per store: 10
- Sales: 730x300x10,000x10 = 21,900 millions
- Size: 21,900 M x 8 attributes x 8 bytes = 1.4TB
  - Assuming sales data consist of 8 attributes
  - Each of which takes 8 bytes

Designing a data warehouse

Designing a Data Warehouse

- Designing a data warehouse is different from designing an operational database:
  - different characteristics of the data to be stored
  - constraints from the existing databases
  - different design criteria
- Emphasis on generalization and conceptual clarity
  - few entities
  - wide coverage
- Main activities
  - analysis — of existing information sources
  - integration
  - design — conceptual, logical and physical

Designing a data warehouse



Integrating information sources

- Integrating information sources consists of merging data from several sources into a single global database that represents the whole corporate information assets
- The main goal of integration is the identification of all the portions of the different information sources that refer to the same aspect of the domain of interest, so that their representation can be unified
- The approach is oriented towards the identification, analysis and **resolution of conflicts** (terminological, structural, pertaining to the encoding of information)

Examples of conflicts

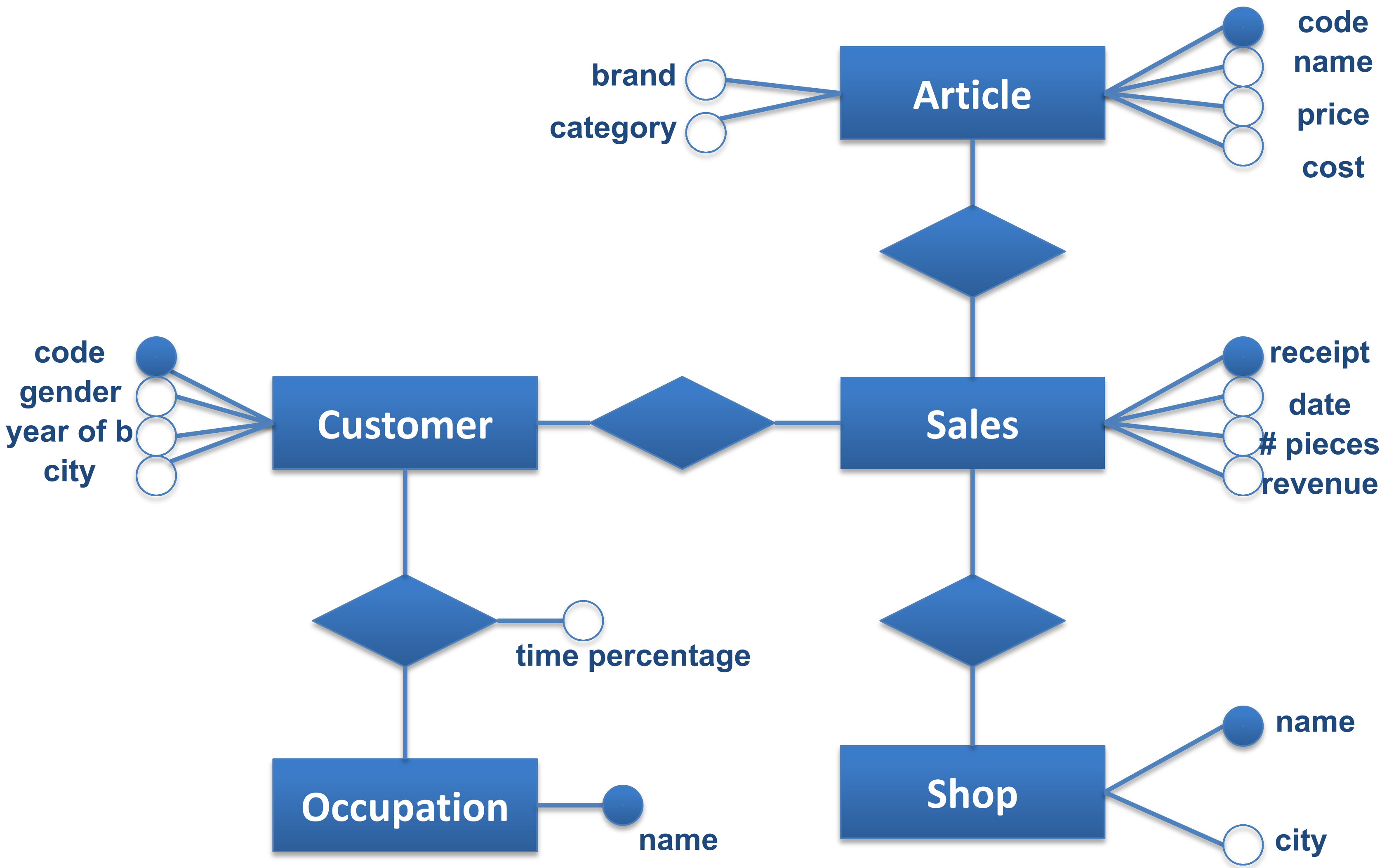
- Conflicts regarding the encoding of information
  - a “gender” attribute can be
    - represented by a character — M/F
    - represented by a digit — 0/1
    - implicitly represented in the SSN
    - not represented
  - the first, middle and last name of a person
    - “John”, “Patrick”, “Smith”
    - “John Patrick Smith”
    - “John P. Smith”
    - “Smith, John P.”
    - “Smith, J. P.”

Determining data marts

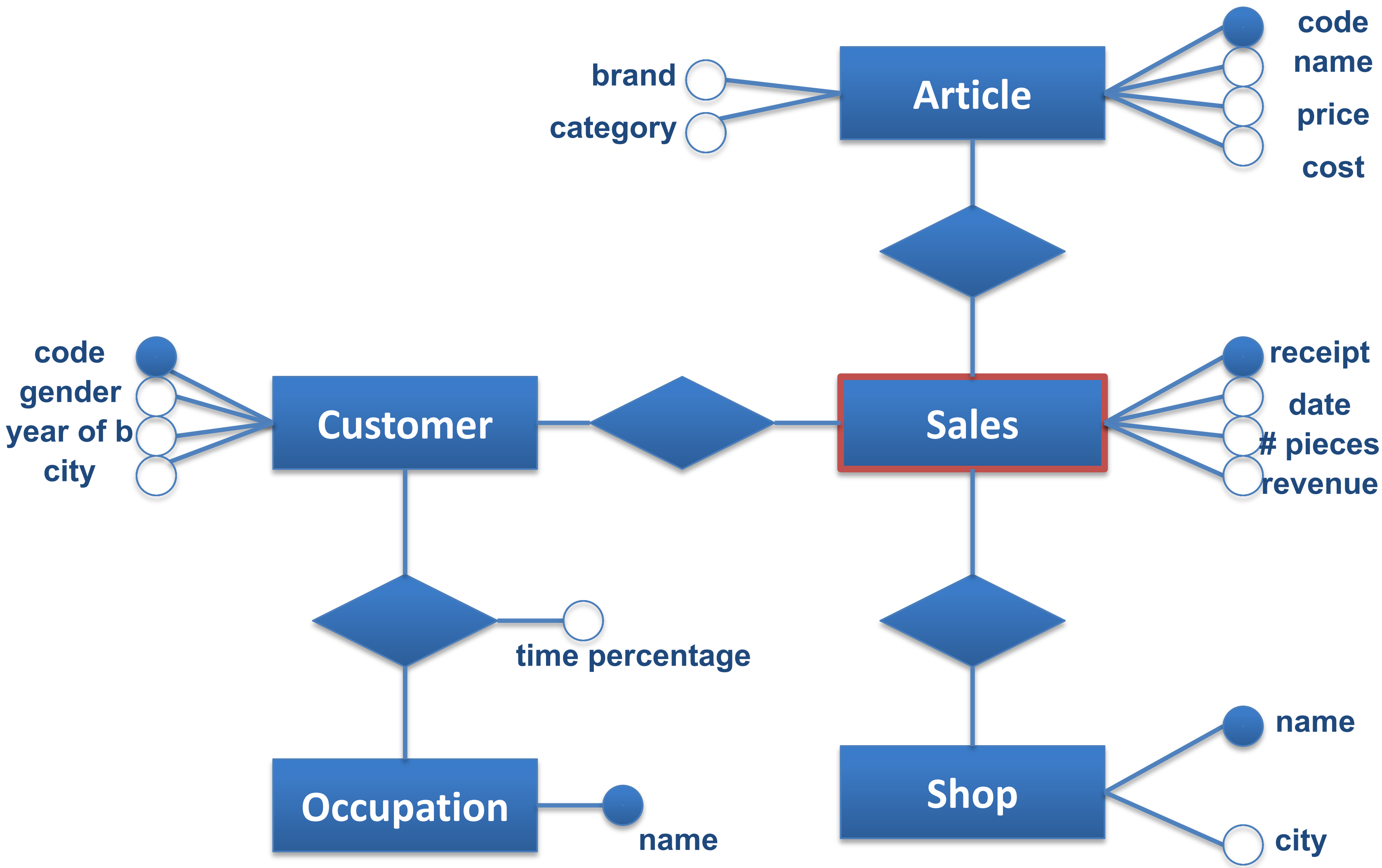
- Normally: Several data marts are present in the data warehouse

- Activities
  - identifying facts, measures and dimensions
  - restructuring the conceptual schema
    - representing facts via entities
    - determining new dimensions
    - refining the levels of each dimension
  - deriving a dimensional graph (star- or snowflake-shaped depending on the circumstances)
  - logical and physical design of the data mart and of the mechanism for populating it starting from the data warehouse

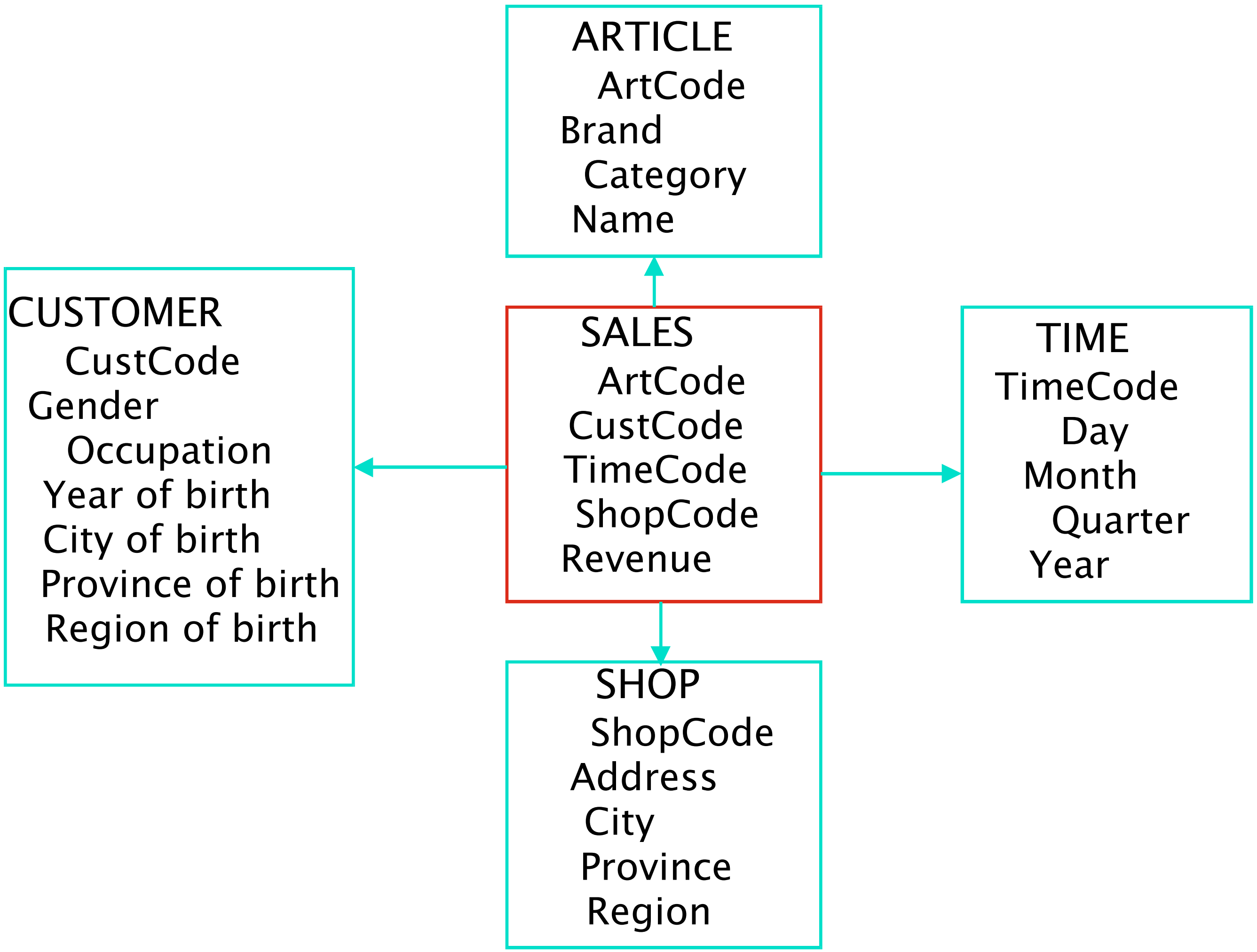
Identifying facts and dimensions



Identifying facts and dimensions



Logical design: star schema for Sale



Types of systems

- MOLAP (multidimensional-OLAP) as opposed to ROLAP (Relational-OLAP)
  - MOLAP: uses non-relational internal structures, better performance, data cubes carry precomputed and prefabricated data
  - ROLAP: uses relational internal structures, can handle large quantities of data, data cubes are created dynamically

Specific technologies

- Bitmap indices
  - Allow efficient evaluation of ORs and ANDs of simple comparisons
- Join indices
  - Precompute join between the dimension and the fact table
- View materialization
  - Precompute views that can be used to answer the most frequent queries