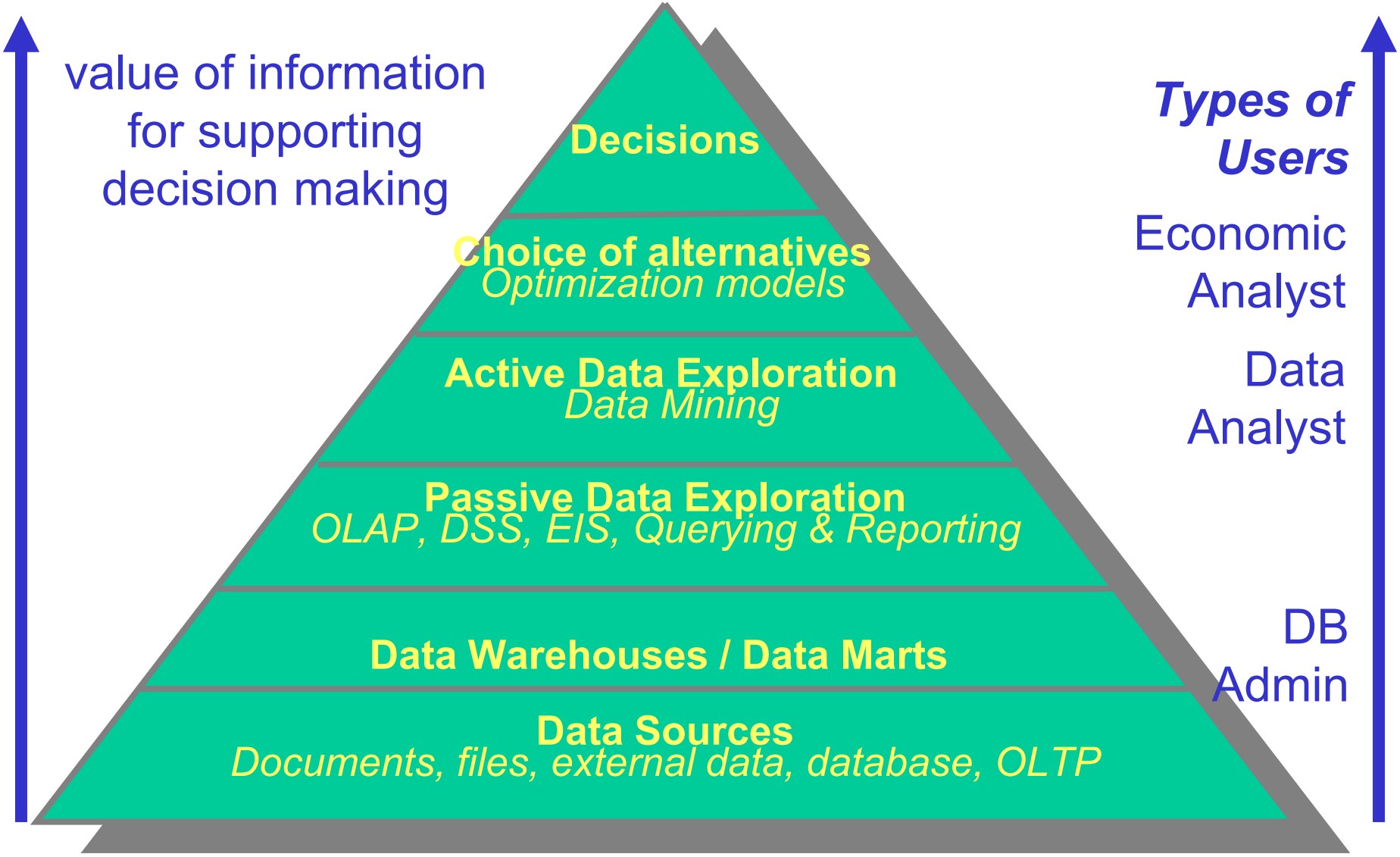
Business Intelligence

"Science is made up of data as a house of stones.
But a mass of data is not science
more than a pile of stones is a house "
Jules Henry Poincaré

Business Intelligence

- Business Intelligence: The set of processes, techniques and tools based on information technology to support the economic decision-making.
- *Goal*: have sufficient information and knowledge in a timely and accessible way so to have a positive impact on the strategies, tactics and business operations.
- Information / knowledge concern the specific company, or more general situations of the market.
- When such information only concern competition in the market, we also speak about competitive Intelligence.

Business Intelligence Technology



Business activities (Anthony's Pyramid, 1967)

- Data to support decision-making
- Distinct IT systems to support distinct company activities

(Ex .: Bank)

DSS and EIS

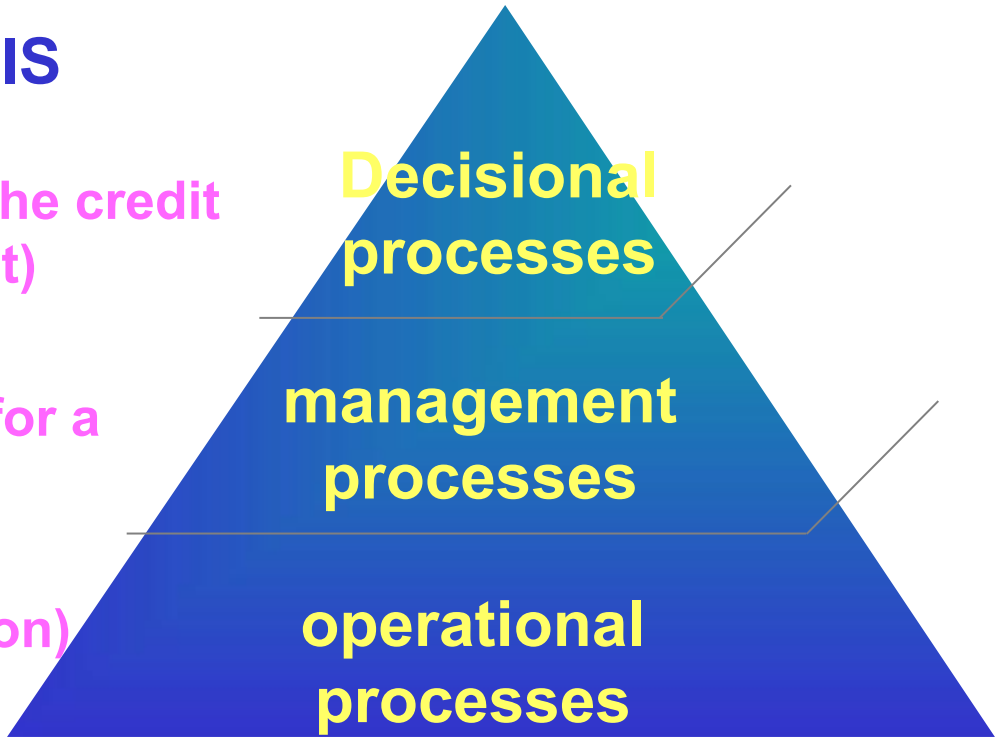
(Agreement with the credit card circuit)

MIS

(Agreement for a loan)

TPS

(Bank transaction)



DSS vs. EIS

Decision Support Systems, DSS and Executive Information Systems, EIS are softwares designed to support the management team for decision-making.

- DSS and EIS applications are different, but related
- DSSs are interactive systems that help decision makers to use data and mathematical models to solve **semi-structured** and **unstructured** decision problems.

The building blocks of a DSS are:

- *A database*
- *A set of mathematical models*
- *A module for the management of the dialogue between system and users*

DSS vs. EIS

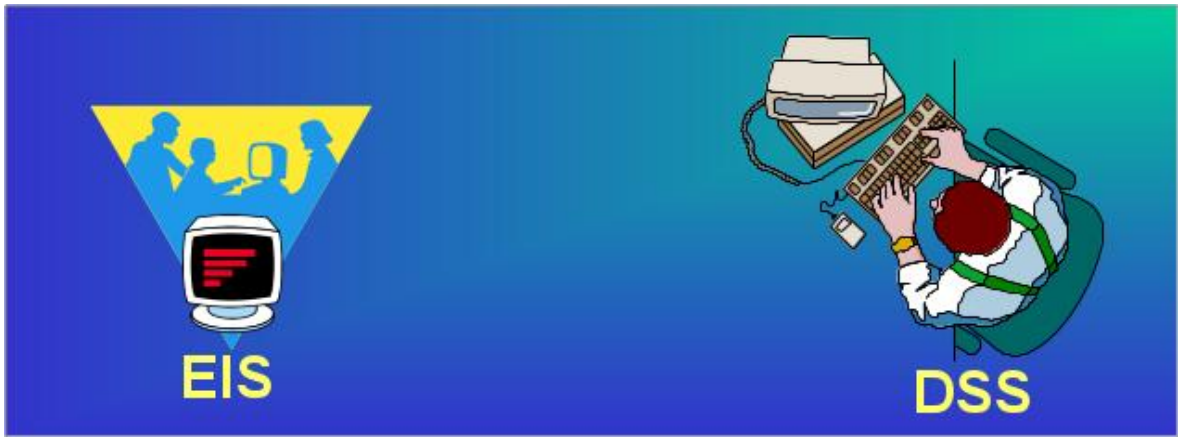
- The first DSSs were developed in the 70s.
- They used mathematical models *to solve* complex managerial problems.
- The DSSs were focused on specific problems
- The idea was that the manager of an organization had to use these systems autonomously
- Problem: the managers did not have the skills and the time to use these systems.

DSS vs. EIS

- In the early '80s, many organizations and vendors offered simpler systems, the EISs.
- The EISs were intended to provide a passive help, for timely and versatile access to the necessary information on the budget, on the history of production, personnel, as well as information on competitors.
- The first EISs had no data analysis capabilities, unlike the DSSs.

DSS vs. EIS

- "*An EIS is used by senior managers to find problems; the DSS is used by the staff people to study them and to offer alternatives*" (Rockart and Delong, 1988)
- The difference of users also evidently involves a difference in the type interface offered by the system.



Where are the data?

- Initially, both the EISs as the DSSs were deficient of a component adapted to collect, in an integrated and permanently way, the data of interest.
- They used **programs to "extract and processing"** capable of scanning a database of files to select the data of interest for the DSS and EIS.
- There was a proliferation of procedures to "extract", anyway
  - the data collected within an organization were typically oriented to operational processes, while ...
  - decision data required for tactical, managerial and strategic decisions have different characteristics.

Where are the data?

- In the early 90's the database technology was primarily aimed at the efficient and reliable management of "online data" (On Line Transaction Processing, OLTP).
- However, the diverse nature of the decision-making data requires different technologies, appropriate to the processing of analytical data (On Line Analytical Processing, OLAP).

OLTP

- Traditional transaction processing, that realize the operational processes of the company /organization
  - default operations, short and relatively simple
  - each operation involves "few" data
  - detailed data, updated
  - The "**acid**" properties of transactions are essential

Example of OLTP queries

```
UPDATE BankAccount
SET balance = balance - 500
WHERE account_number = '12345';

UPDATE BankAccount
SET balance = balance + 500
WHERE account_number = '67890';

INSERT INTO Transactions (account, type, amount, date, time)
VALUES ('12345', 'outgoing_transfer', 500, '2024-12-17',
   '10:32:15');
```

OLAP

- Operations for decision support
  - Complex and random operations
  - Each operation can involve a lot of data
  - Aggregate data, historical data, also not updated
  - The "acid" properties are not relevant, because the operations are read-only

Example of OLAP query

```
SELECT
    branch.region,
    branch.province,
    YEAR(loan.issue_date) AS year,
    QUARTER(loan.issue_date) AS quarter,
    customer_type.age_group,
    customer_type.profession,
    COUNT(*) AS loan_count,
    SUM(loan.amount) AS total_issued,
    AVG(loan.amount) AS average_amount,
    AVG(loan.interest_rate) AS average_rate
FROM loan
    JOIN branch ON loan.branch_id = branch.id
    JOIN customer ON loan.customer_id = customer.id
    JOIN customer_type ON customer.type_id = customer_type.id
WHERE loan.issue_date BETWEEN '2020-01-01' AND '2024-12-31'
GROUP BY branch.region, branch.province, year, quarter,
         customer_type.age_group, customer_type.profession
ORDER BY year, quarter, total_issued DESC;
```

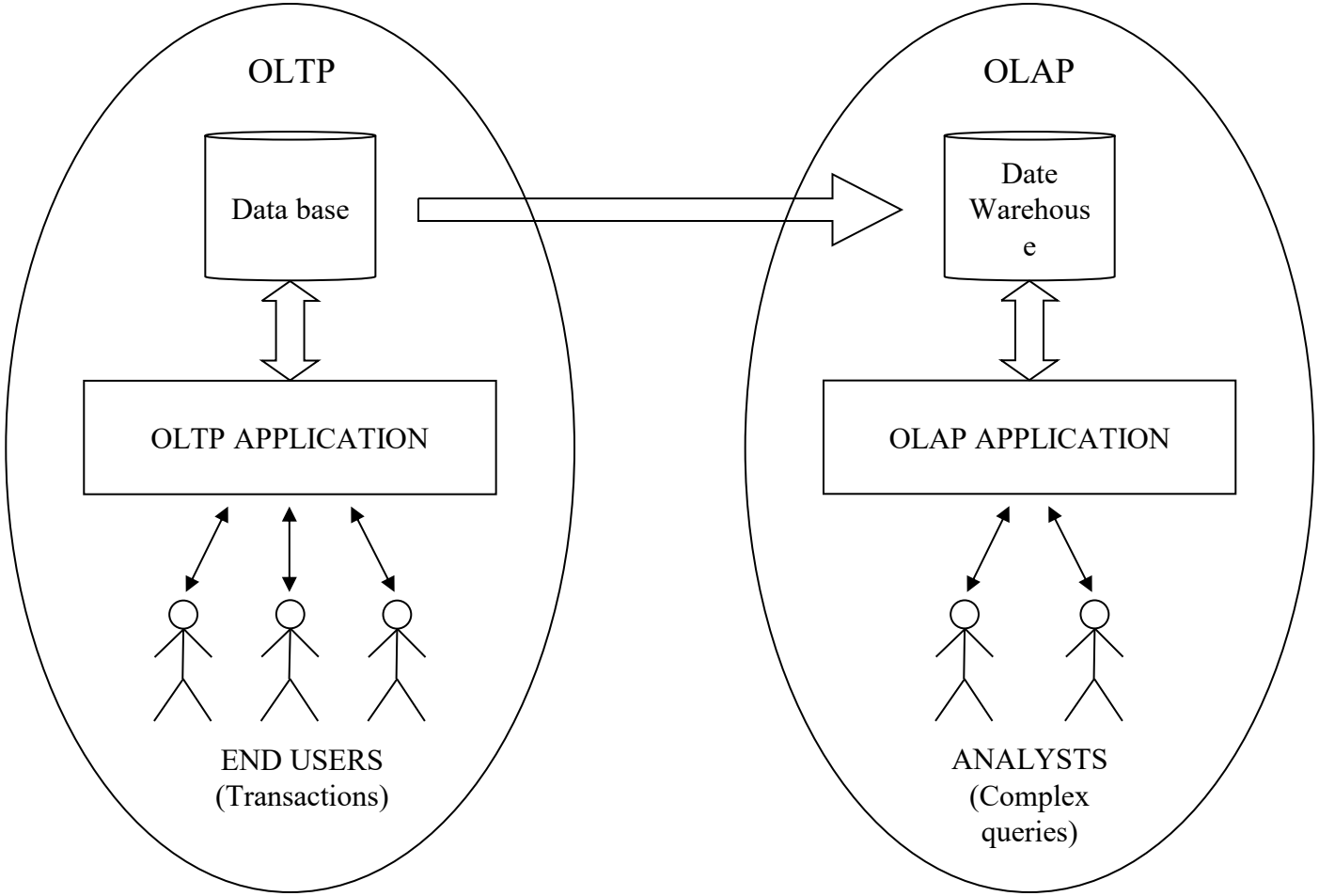
OLTP vs. OLAP

	OLTP	OLAP
User	employee	manager
Function	daily operations	decision support
Design	application-oriented	data-Centric
Data	current, updated, detailed, relational, homogeneous	historical, aggregates, multidimensional, heterogeneous
Use	repetitive	random
Access	read-write, indexed	read, sequential
Work units	short transaction	complex queries
No. of records	dozens	millions
No. of users	thousand	hundreds
Size	100GB – 1TB	100TB – 1PB
Metric	throughput	response time

OLTP and OLAP

- The requirements are then different
- The applications of the two types can harm each other
- It is necessary to keep separate the two environments, developing new databases, called data warehouses, because they hold large amounts of data, far more than needed for transaction processing, for a long period of time.

Separation of environments



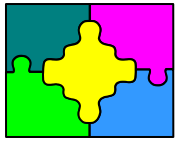
A Data Warehouse is ...

A data warehouse is a

- integrated,
- subject-oriented,
- time-variant, and
- nonvolatile

collection of data in support of management's decisions

Inmon, William Harvey (Bill)
Building the Data Warehouse
Wellesley, MA: QED Tech. Pub. Group, 1992

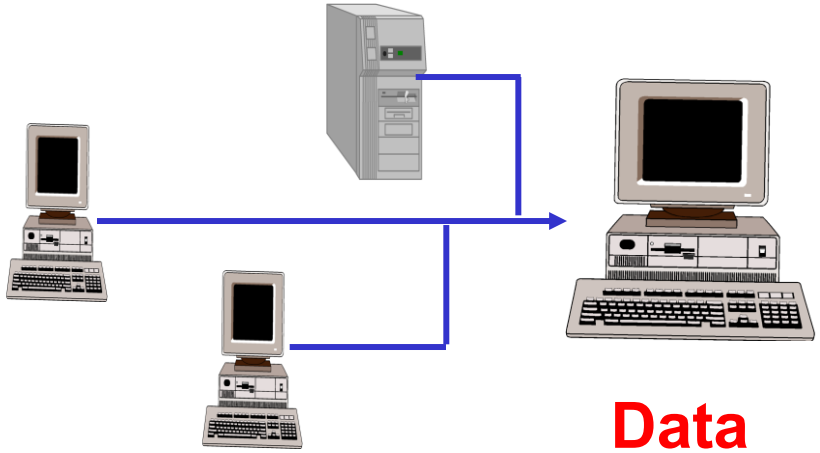


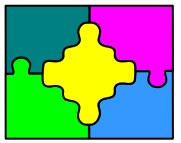
Integrated ...

- It is an integrated database: That is, consistent with respect to an overall conceptual data schema, which embraces the whole enterprise and not individual departments.

The data of interest in fact originate from several existing information sources and this requires a preparatory activity of reconciliation of heterogeneity:

Diversity in field names, structure, information coding





... Integrated

- In order to reconcile the data, it is necessary to perform:
  - Data Cleansing (*data cleaning*)
  - Data Validation
  - Integration

An example of data integration

Checking Account System
Jane Doe (name)
Female (gender)
Bounced check # 145 on 1/5/95
Account opened in 1994

← Operational data

Savings Account System
Jane Doe
F (gender)
Account opened in 1992

Investment Account System
Jane Doe
Owns 25 Shares Exxon
Account opened in 1995

Customer
Jane Doe
Female
Bounced check # 145
Married
Owns 25 Shares Exxon
Customer since 1992

↑ data warehouse

Subject-oriented

- Oriented to the subject: the data are grouped by area of interest (eg., customers, suppliers, products) rather than operating processes (eg. marketing and sales).
- The logic and physical design of the operational DB is oriented to computing requirements of the individual operating processes or applications.
- In contrast, data from a data warehouse are aimed at those who use them than to those who generate them
- *Data Normalization* is not relevant.

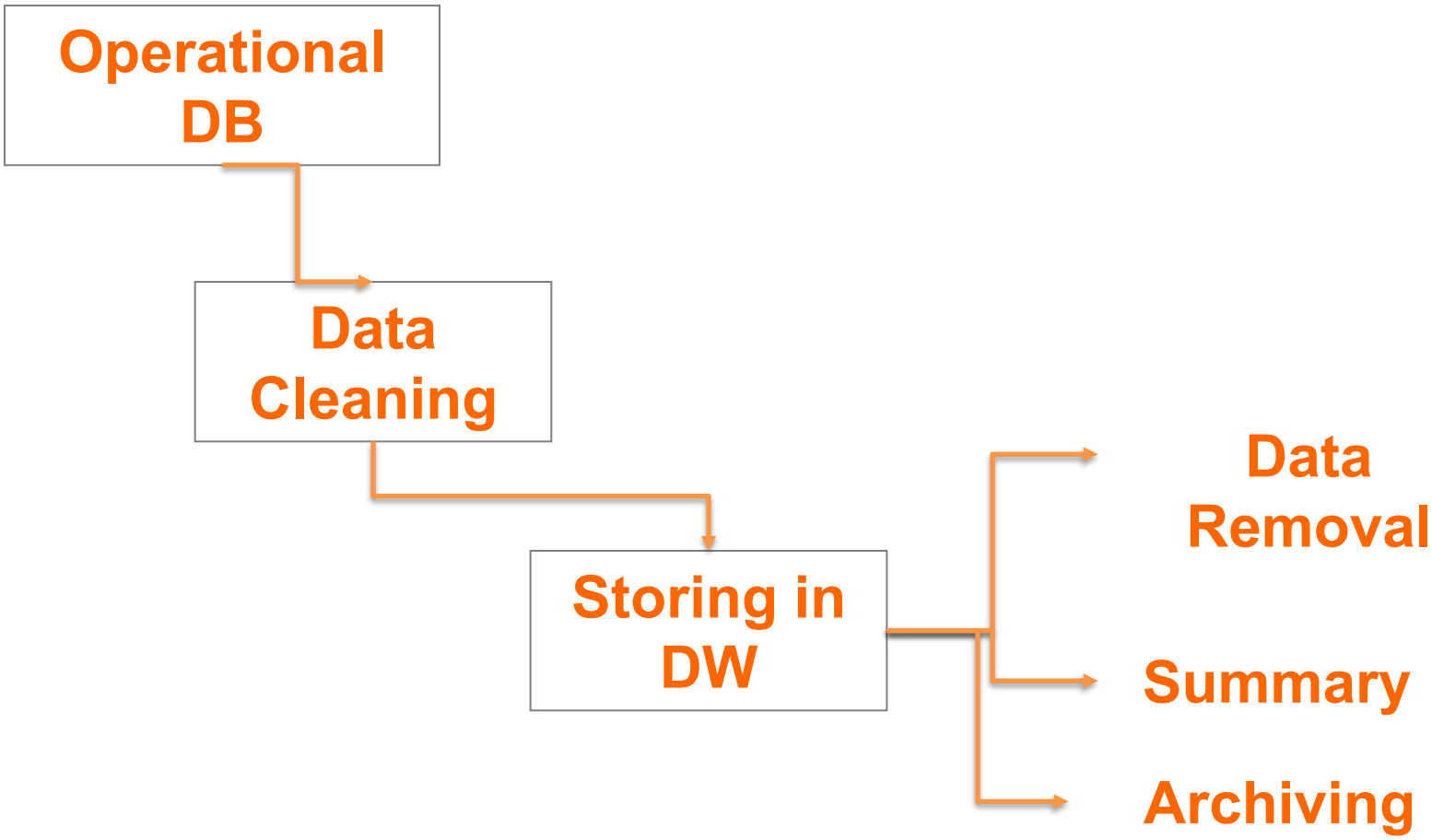
Time-variant

- Contains historical and temporal data. While in fact the DBs for operational data keep the current value of the information, in DW, by its nature, it is of interest the historical evolution of the data.
- One of the common objectives of the analysis is, in fact, to identify trends in the data, unexpected developments, regularity or irregularity.
- In order to represent the evolution of the data, we associate a *timestamp* to operational data at the level of individual fields or entire records.
- The keys of the entities stored in a data warehouse will contain a time element.

Non-volatile

- The data loaded into a data warehouse can be inspected but not modified by the user.
- The data are stable resources in order to ensure the consistency of the analysis and comparative analysis in time.
- In contrast, the operational data can be inserted, updated, and deleted from the operational DB.

The data flow



Which data are in the DW?

- A data warehouse can contain five types of data:
  - Current detailed data
  - Detailed data of the past
  - Slightly summarized data
  - Highly summarized data
  - Metadata
- Choosing the right level of *granularity* is a major problem in the data warehouse design

In addition, the DW ...

- It has an independent existence. The DW is always kept physically separated from the information sources.
- It is indeed difficult to integrate online because the data of interest are different and the maintenance of historical data, aggregation and analysis require special organizations and specific access methods.
- In addition, without the separation, we would witness to a general degradation of the performance of OLTP systems.

In addition, the DW ...

- It is an offline database. The data import mechanisms are normally asynchronous and periodic, so as not to penalize the performance of the data sources, especially if it is for OLTP systems with particularly critical performance. In this case, the DW does not contain fully updated data w.r.t. OLTP systems.
- A controlled misalignment of the data, however, is generally considered acceptable for many analytical applications.

The Data Mart ...

A data mart is a scaled version of a data warehouse, designed for a specific sector or department.

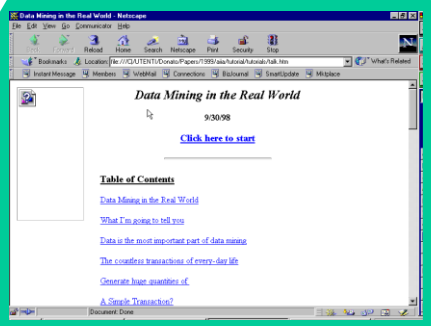
The data marts have several advantages, compared to the data warehouse:

- The cost is low (less than 100 thousand dollars compared to million of dollars)
- The implementation time is reduced (a few months)
- They are controlled locally rather than centrally, giving power to the group that uses it.
- They are smaller and thus with more rapid response times.

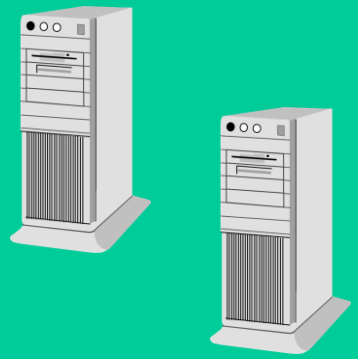
... the Data Mart

- For many organizations a data mart is an important first step towards a data warehouse.
- The problem is that the data mart of a company can greatly differ from sector to sector. So the development of a data warehouse at the global level may require a phase of integration.
- That if we develop the data warehouse after designing the data mart
- An alternative approach is to design a centralized data warehouse first and then, based on this, design a sectoral data mart, with better performance.

Architecture of a Data Warehouse



external sources



operational databases

metadata

Extraction Transformation and Loading

data warehouse

used by

data marts

used by

EIS

DSS

OLAP

data mining

queries

Multidimensional Model

- Data in a DW are presented to the end user by means of a high-level representation that is independent from the data storage criteria and favors the analysis.
- In this representation, the data is organized by *areas of interest* (dimensions of analysis).
- It is based on a conceptual model known as ***multidimensional*** model.
- There is no standardization of the multidimensional model, so it is not possible to give a formal definition.

Multidimensional Model

It is based on three concepts:

- the ***fact***: it is a concept of the information system on which it makes sense to carry out a process of analysis oriented to decision support.
- a ***measure***: it is an atomic property of a fact that we intend to analyze. Typically it is a numeric attribute.
- a ***dimension***: it is a particular perspective along which the analysis of a fact can be conducted. The possible values for a dimension are generally called *members*.

Multidimensional Model

Example: In a commercial company that has a chain of supermarkets a ***fact*** can be the concept of *sale* and possible ***measures*** for this fact may be the *quantity* sold for a product in a certain period of time and the relative *gross receipts*. Possible ***dimensions*** of analysis for a sale may be the *product* sold, the *period of time* in which the sale was made and *place* where it was held.

Multidimensional Model

Example: For an insurance company, the ***fact*** can be the *accident* subject of the insurance reimbursement and possible ***measures*** may be the *number* of claims in a certain period of time and the relative *cost*. Possible ***dimensions*** of analysis for an accident may be the *customer* requesting the reimbursment, the *typology of the accident* the *insurance policy* and the *period of time* in which the accident has occurred.

Multidimensional Model

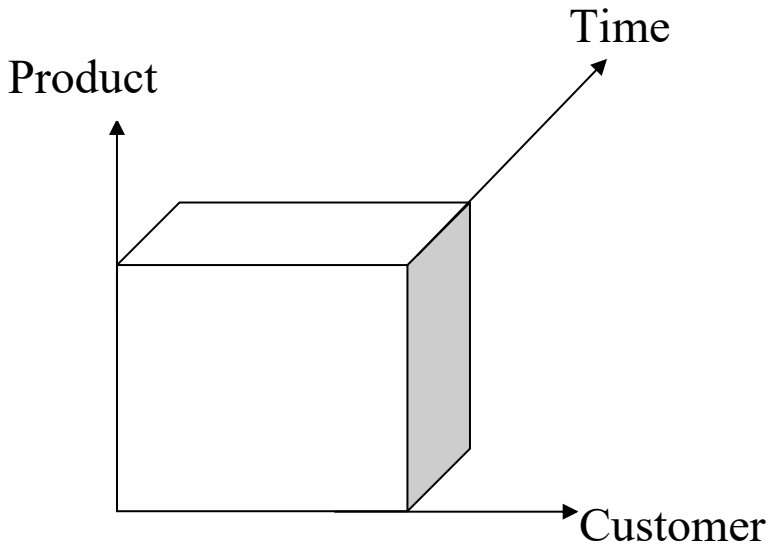
- There is a natural graphical representation widely used by analytical tools, in which the facts are represented by ***multidimensional cubes*** (*data cube*) consisting of atomic elements, called *cells*.
- A multidimensional cube is focused on one fact.
- Every axis of the cube represents a possible dimension of analysis.

Multidimensional Model

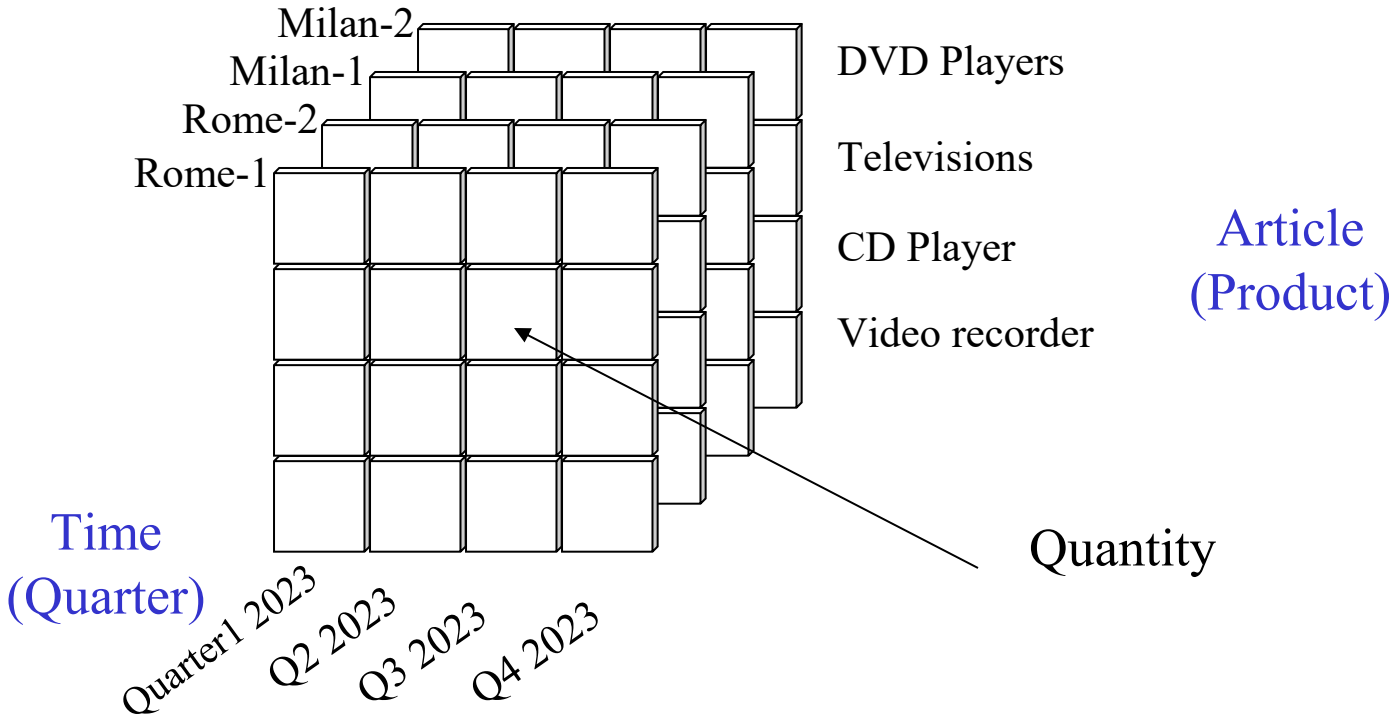
Example: Sales of some products.

The dimensions according to which sales are analyzed are:

- products
- time
- customer



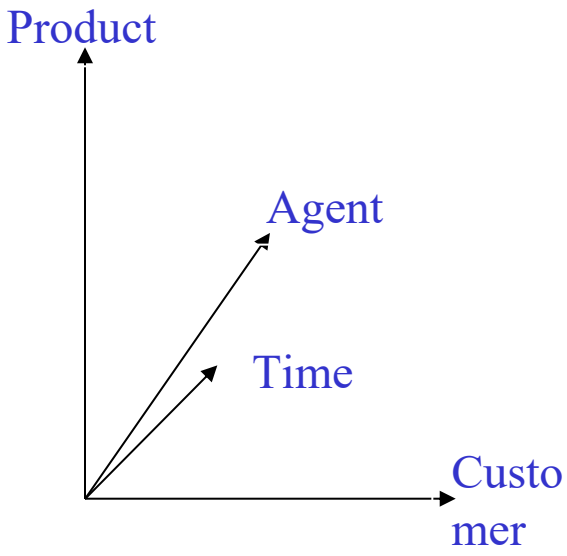
Example

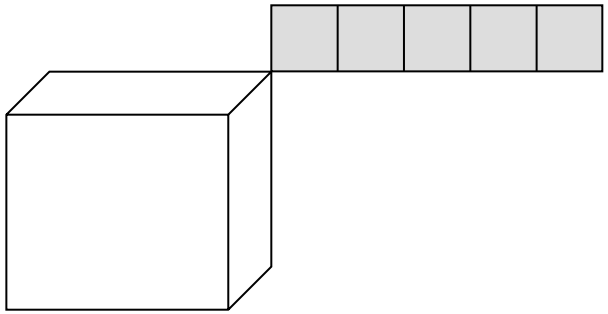


Sales of a commercial company

Multidimensional Model

In fact, the dimensions of analysis may be more than three. For example, sales may be analyzed considering also agents who have taken the deal. In these cases it is necessary to create a *hypercube*.





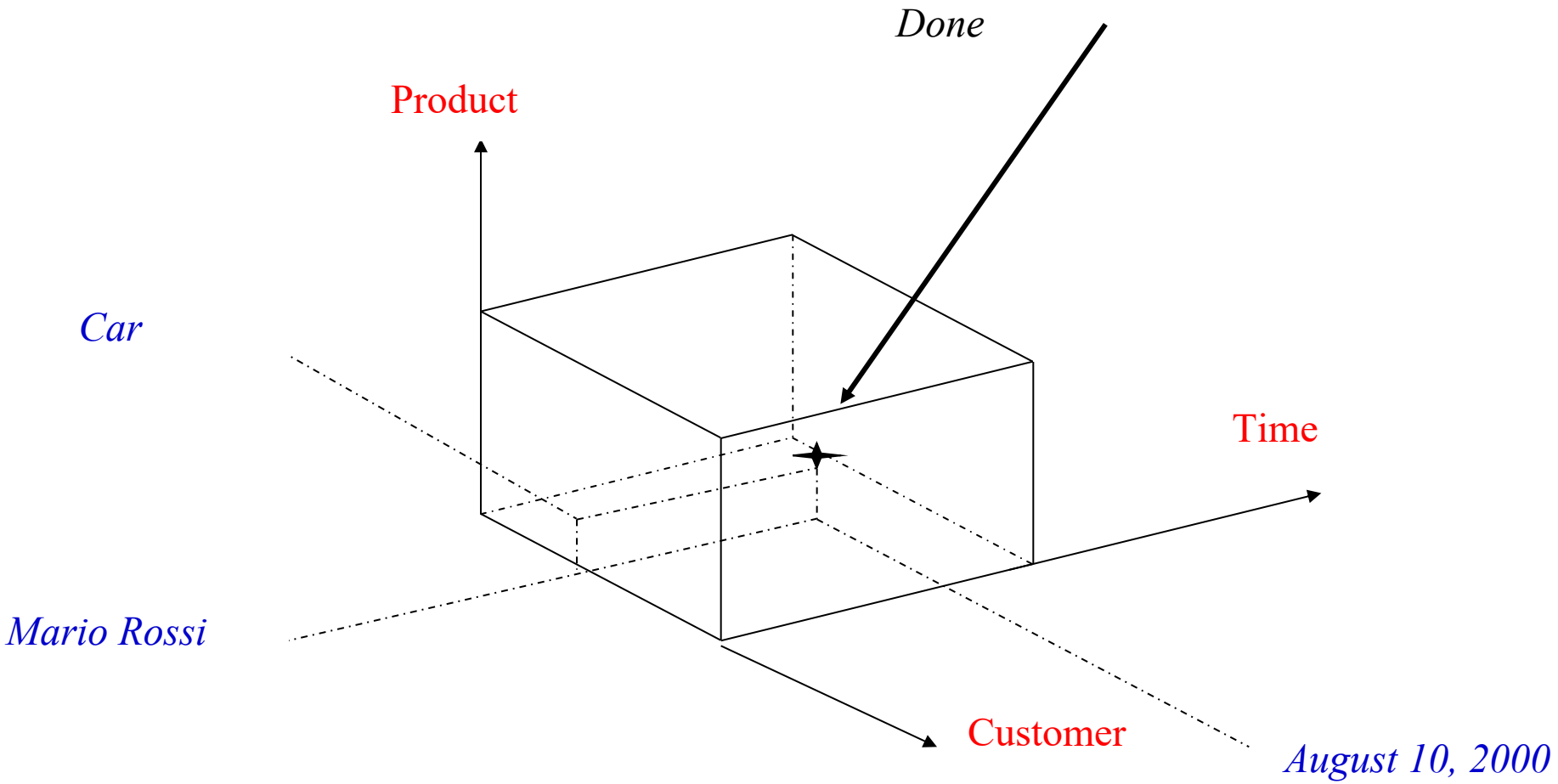
Multidimensional Model

To access the data of a sale, it is necessary to specify the coordinates, or values for the dimensions of analysis.

Example: Referencing sale *August 10, 2000*, of the product *car*, Customer *Mario Rossi* or select all sales for the month of August. In this way it is possible to select from the hypercube only a portion of the data present in it.

If for each of the dimensions it is specified a precise value, then one hypercube ***cell*** (single ***fact***) will be found (in our case, a sale).

Multidimensional Model



Multidimensional Model

If you fix a precise value for only one of the dimensions a ***slice*** of the hypercube is determined

The query operations in a multidimensional logical model are reduced, therefore, to simple selections of portions of a hypercube.

In this sense, it is easier to create queries than in relational databases, for which it is often necessary to resort to complex ***join*** operations, with computational costs that are not negligible.

Multidimensional Model

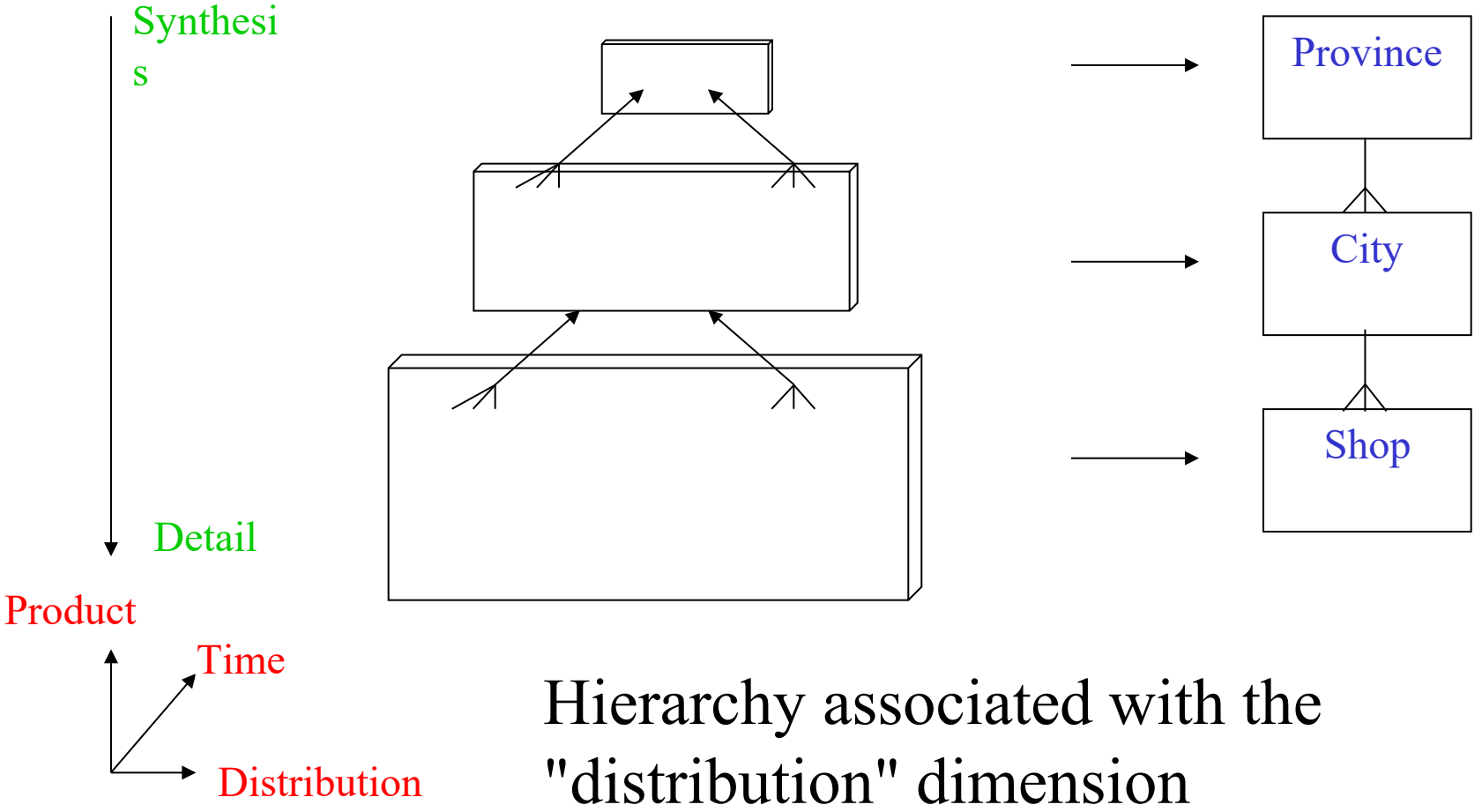
- The multidimensional model appears similar to that of multi-dimensional arrays, but ...
- unlike the classic array, in which the indices are characterized by a linear order (typically the indices are integer values), in a multidimensional model for the indices it could not be defined an order, or in any case, it could be defined a partial ordering (hierarchies of values).

Multidimensional Model

In the multidimensional model the dimensions of analysis can generally be organized in **hierarchies**.

Example: the dimension of analysis distribution, which refers to a certain company distribution network, will have the lowest level of the hierarchy the single retail shop, and at higher levels the cities and the provinces reached by the distribution network.

Multidimensional Model



Hierarchy associated with the "distribution" dimension

Multidimensional Model

The analogy with cubes is not limited to data presentation. Indeed, we can define some analysis operations that are expressed as operations on cubes.

These are operations that apply to cubes and return new multidimensional cubes, not necessarily with the same number of dimensions.

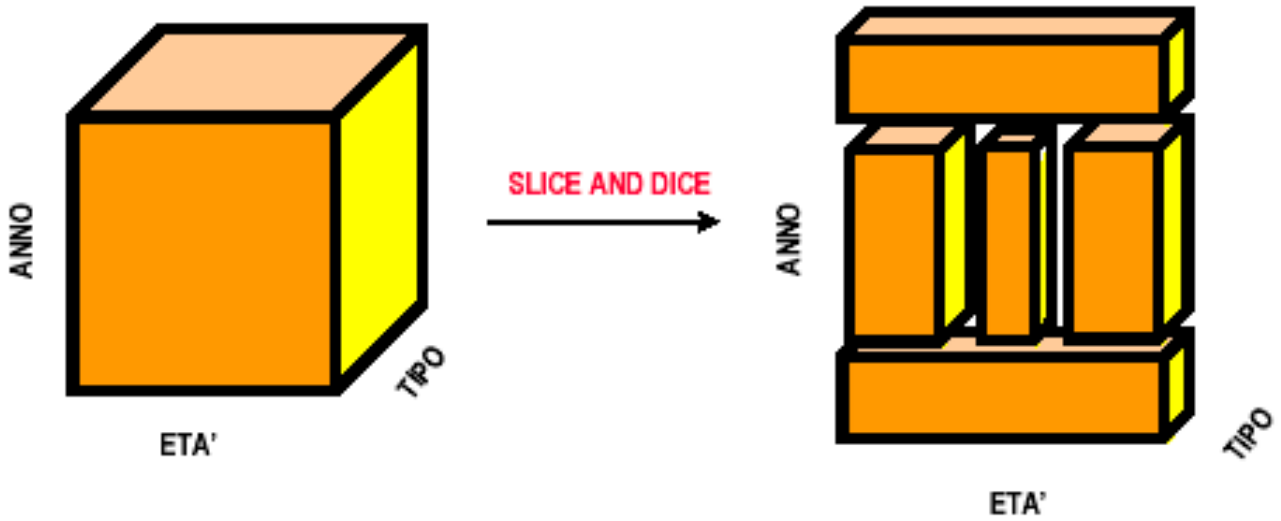
The best known are:

Multidimensional Model

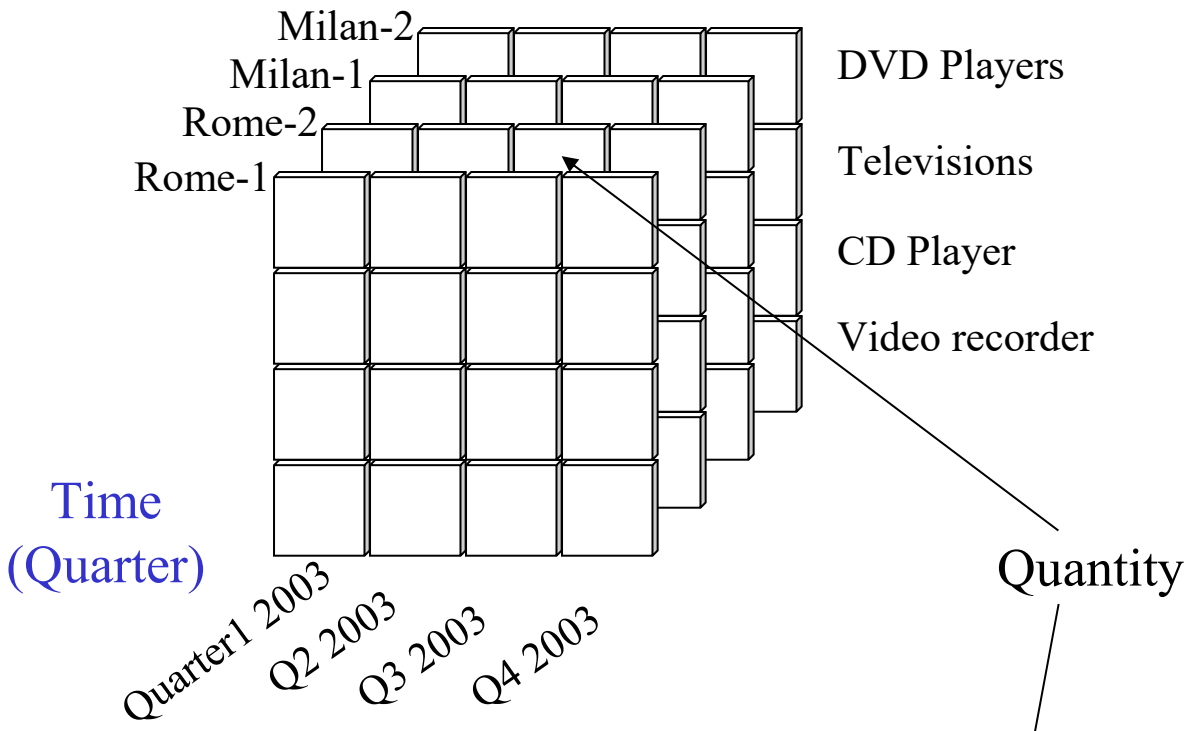
- ***Slice***: Is the operator that let you see the cube *transversely* (literally "sliced"), setting a value for at least one of the dimensions and analyzing the data with respect to all the others, i.e. focusing on a hypercube with (n-1) dimensions from the original n-dimensional cube (*dimensional contraction*)

- ***Dice***: An interval of each dimension is given, so that it produces a *volumetric reduction,* without contractions of the number of dimensions.

Multidimensional Model



Operation of slice-and-dice



DVD PLAYERS	Q1 03	Q2 03	Q3 03	Q4 03
Rome-1	38	91	66	198
Rome-2	155	219	248	265
Milan-1	121	273	266	326
Milan-2	222	122	155	200

Multidimensional Model

- ***Drill down*:** is the operator that allows us to go into detail along one or more hierarchical dimensions.

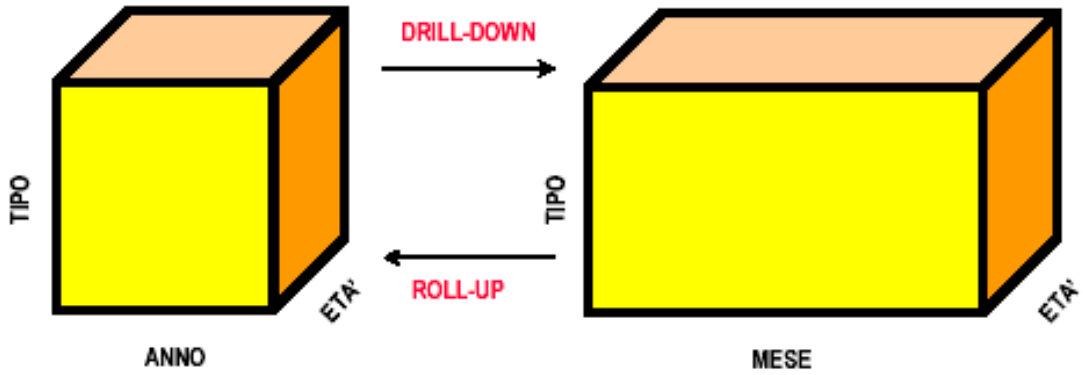
Example: using a drill-down we can go from an analysis of sales by province in a more detailed one, where we distinguish the city.

This operator is useful when we want to analyze a cause or an effect of some phenomena observed on the aggregate data.

Multidimensional Model

- ***Roll-up*** or ***consolidation*** or ***drill up***: It is the dual operator of the drill-down, since it allows to go up along one or more hierarchical dimensions.

Example: starting from the analysis of a particular product we could move to the analysis of a whole range of products.

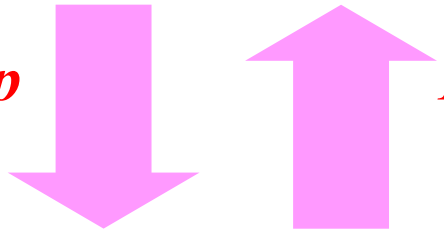


Roll-up an drill down operations

DVD PLAYERS	Q1 03	Q2 03	Q3 03	Q4 03
Rome-1	38	91	66	198
Rome-2	155	219	248	265
Milan-1	121	273	266	326
Milan-2	222	122	155	200

Low granularity

Roll-up



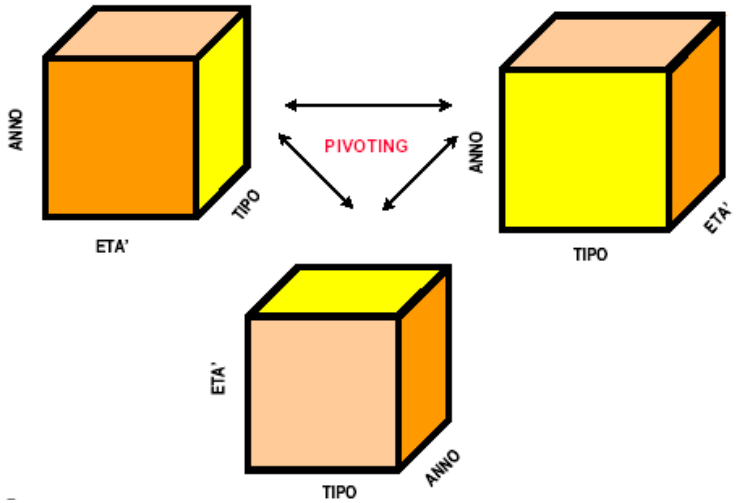
Drill-down

DVD PLAYERS	Q1 03	Q2 03	Q3 03	Q4 03
Rome	193	310	314	463
Milan	343	395	421	526

Higher granularity

Multidimensional Model

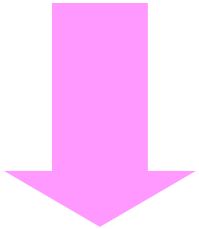
- ***Rolling*** or ***pivoting***: It involves a change in the presentation mode with the aim of analyzing the same information from different points of view. Following the multidimensional metaphor, perform the pivoting means to rotate the cube, bringing in the foreground a different combination of size. If space is m-dimensional analysis, it is possible to have *m*! different perspectives of analysis.



Multidimensional Model

Example: Pivoting on a two-dimensional table

DVD PLAYERS	Q1 03	Q2 03	Q3 03	Q4 03
Rome	193	310	314	463
Milan	343	395	421	526



DVD PLAYERS	Rome	Milan
Q1 03	193	343
Q2 03	310	395
Q3 03	314	421
Q4 03	463	526

Schema of a DW

In a DW application it would be natural translate a multidimensional conceptual model into a *logical* model with similar properties.

This presupposes, however, the use of a DBMS that supports such a logical model, ie a ***MDDBMS*** (***Multidimensional DBMS***), which precisely stores numerical or quantitative data categorized on different qualitative dimensions.

Example: a multidimensional database (***MDDB***) can store (quantitative) data related to sales for different product lines, in different cities, and for each month (three qualitative dimensions).

Schema of a DW

The MDDBMSs are not new. For about twenty years, the EXPRESS software package of IRI Software Inc. (Burlington, MA), now owned by Oracle, has included a MDDB. Since the early 90s, many other software companies have product for MDDB systems.

- *Benefit*: The MDDB are optimized for speeding up and simplifying queries, thanks to pre-processing operations.
- *Downside*: They have scalability problems. A 200MB file, input to a MDDBMS, can take up to 5GB because of the pre-processing carried out

Schema of a DW

In addition to these technical problems, we add the availability in companies of only relational databases, and the competence of people limited to the relational model.

This leads to consider the possibility of transforming a multidimensional conceptual model into a logical relational model.

Of course, the choice of a relational DBMS to model a data warehouse leads to face the problem of how to simulate a multi-dimensional schema with a relational schema.

Star Schema

The *star schema* is the easiest way to simulate, through the use of a relational database, a multidimensional approach.

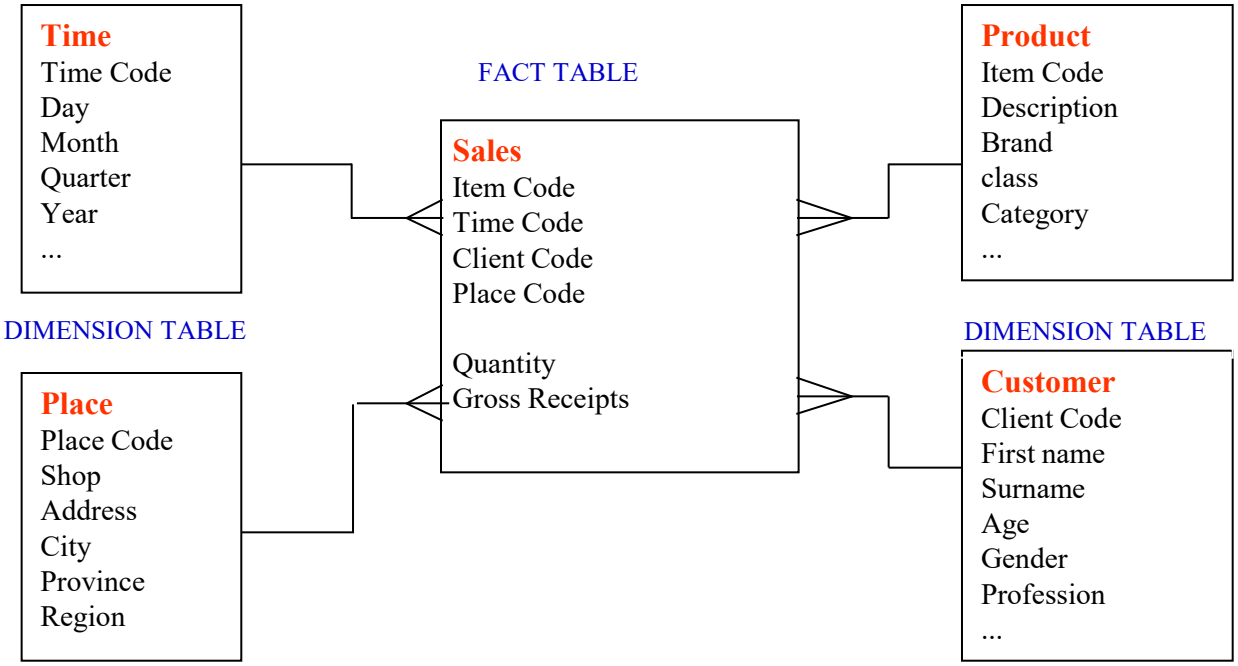
The name comes from the appearance of the data model, with a large central table surrounded by many subordinate tables in a star configuration.

The central table, known as ***fact table*** contains the numerical data of a hypercube cell of the multidimensional model and the keys that link data to their dimensions.

Star Schema

Contingency tables, also known as ***dimensional*** or ***satellite tables***, contain the attributes that describe the data components: they are as many as the dimensions of analysis identified in the conceptual modeling.

Star Schema



- The facts are in (Boyce-Codd) normal form as each non-key attribute is functionally dependent on its unique key
- The dimensions are not normalized. Eg, in Product, the CategoryName depends on Category Code, which is not a key
- There are four foreign key integrity constraints, one per dimension.

Star Schema

The denormalization of each of the dimension tables allows to reduce the number of joins required to solve a query, and then, ultimately, allows to increase efficiency of analysis systems that will transform the queries made by the decision maker, by means of graphical user interfaces, in SQL queries.

The price to pay for the gain in efficiency is paid in terms of waste of memory, due to data redundancy. For example, the category name will be repeated for all the products that fall into the same category.

Star Schema

Fortunately, the anomalies for delete, edit and update operations, which represent typical drawbacks arising from redundancy, do not occur in this case simply because, as previously mentioned, the data in the DW are, by their nature, static.

Star Schema

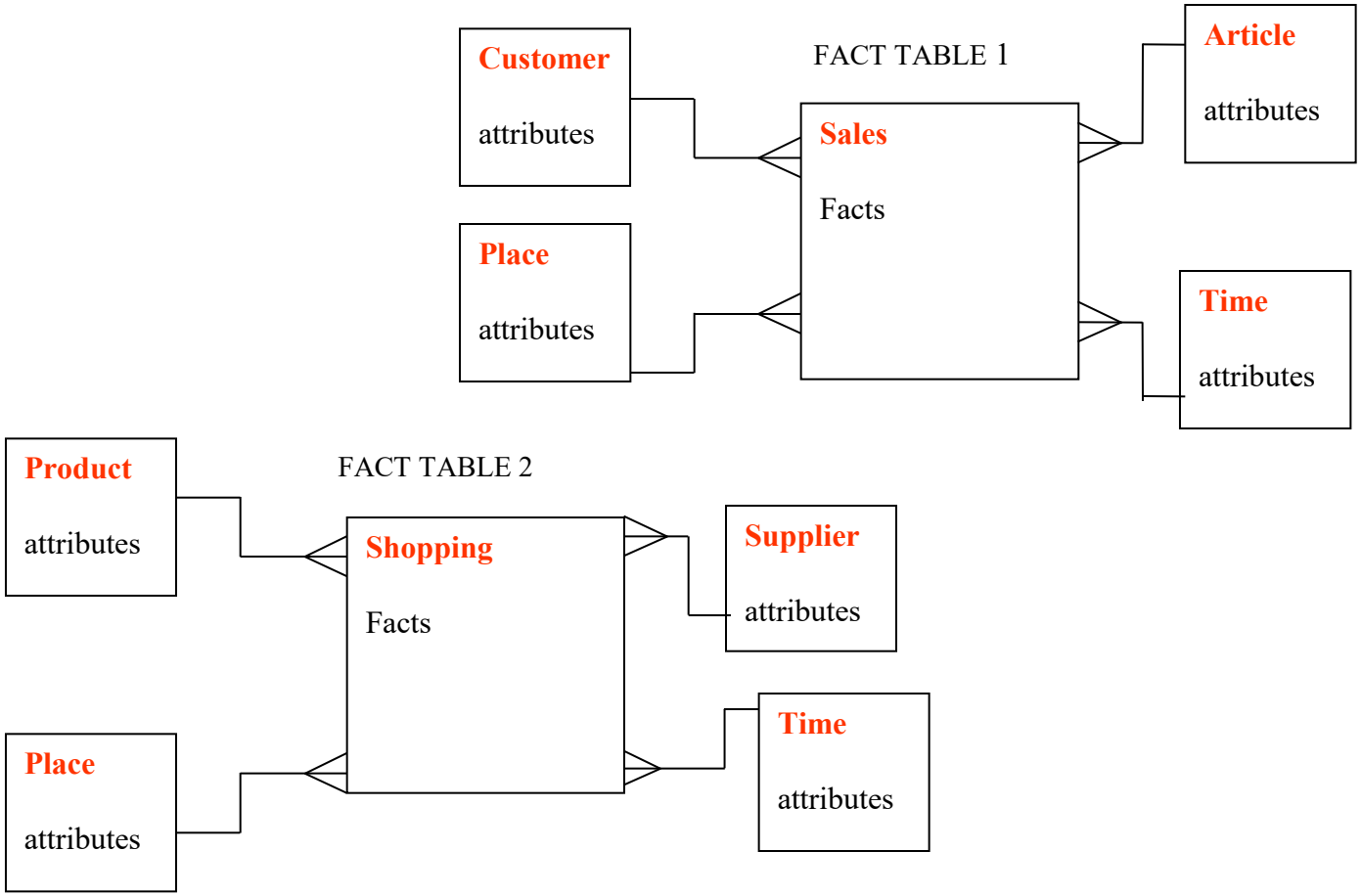
The data present in the fact table, called ***measures***, must necessarily be numeric. In fact, the purpose is always to extrapolate synthetic numeric data, such as "the total gross receipts due to the sales of a particular product in a certain geographical area."

Constellation diagram

- The multi-dimensional conceptual schema of a DW can have more hypercubes. This means that we have several groups of measures, each of which corresponds to a set of dimensions.
- The translation of a multidimensional conceptual schema of its kind in a relational logical schema can lead to multiple fact tables.
- It is also possible that the fact tables share some dimensions. In these cases we need to make a choice.

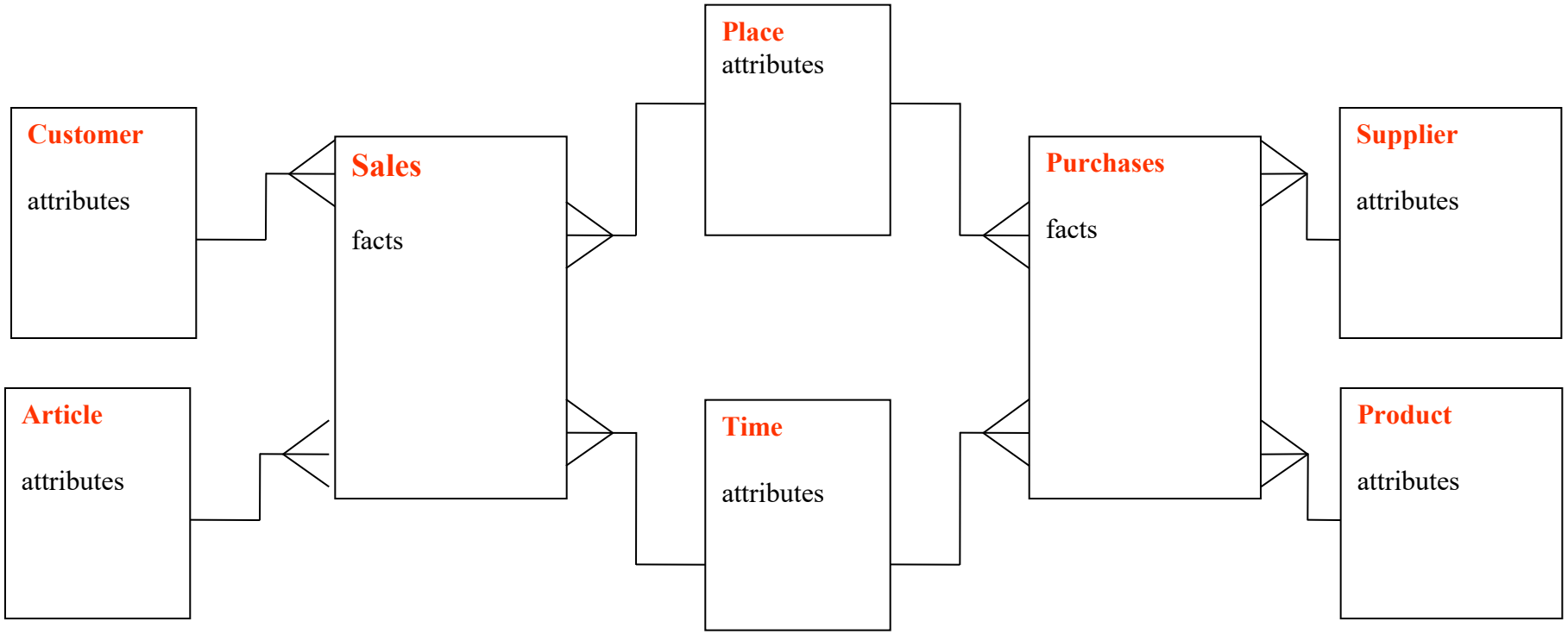
Constellation diagram

Fact tables can be kept separate, together with the relative dimensional tables.



Constellation diagram

Or we can build a constellation diagram in which the shared dimensional tables are inserted only once and connected to all the shared fact tables.



Snowflake Schema

The snowflake schema takes its name by how ER diagrams related to this scheme look like. This aspect is determined by a normalization of dimensions.

Example. The dimension place has three attributes: the city, the province and the region of the shop.

The records in the dimension table place have many redundant information, due, in this case, to the defined hierarchy of cities, regions and states.

To remove this redundancy, we could normalize the place table dividing it into four tables: one for the shops, one for the cities, the other for the provinces and the fourth for the regions.

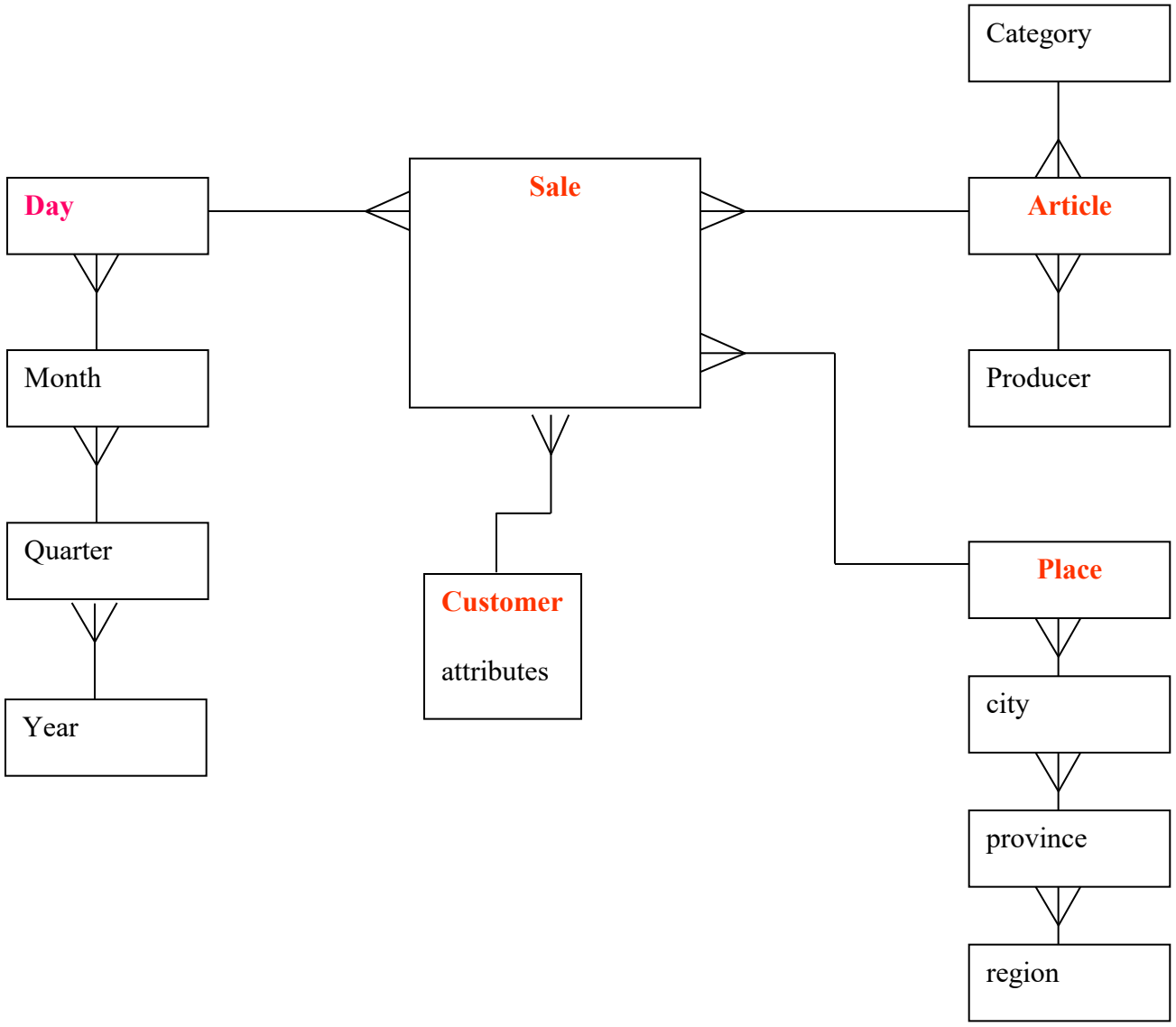
Snowflake schema

Example (Cont.)

The situation is different for the table Article where, in addition to a dependence of the hierarchical nature of the class, there is not really hierarchical nature with the producer (see brand) of that article.

Still different is the situation in the case of date, since there is no functional dependence between the day, month and year. In this case, however, we can define a sort of hierarchy among the attributes day, month and year, and we can still split the table Time in several tables.

Snowflake schema



Snowflake schema

In general, the normalization allows us to divide the data as a function of hierarchies identified for the specific dimension.

Each dimension table contains data related to a single hierarchical level and a link to the next hierarchical level.

The dimension table on the lower hierarchical level is connected to the fact table.

Snowflake schema

The use of a snowflake is generally not recommended, because the gain in terms of memory occupation, due to the normalization, is usually not sufficient to compensate the loss of efficiency, due to the greater number of joins required to answer to queries.

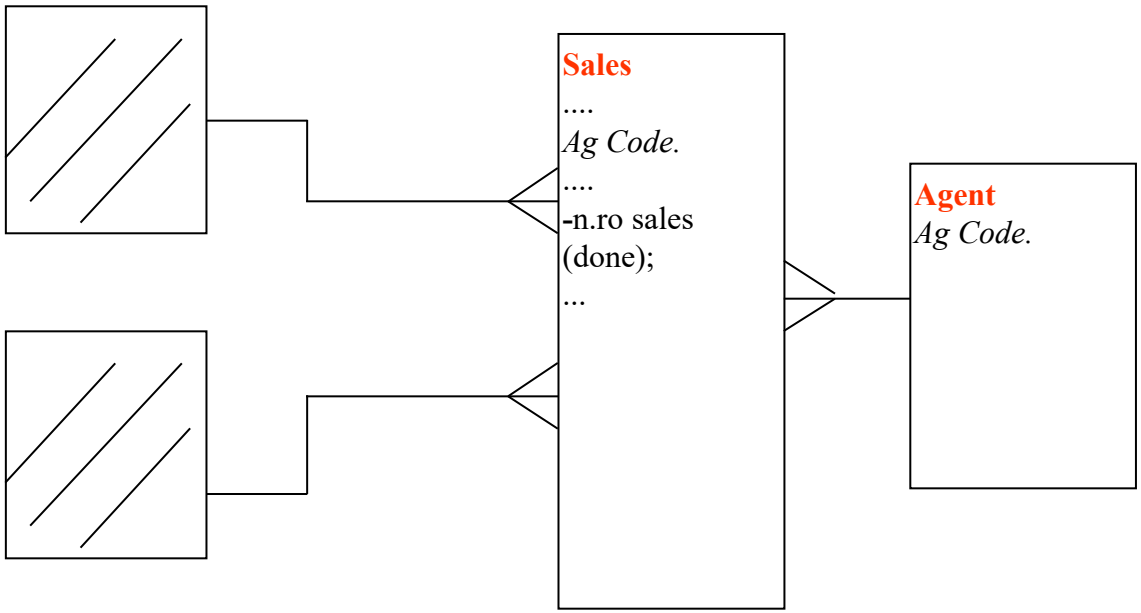
Moreover, in most cases, the memory occupancy gain is insignificant if one considers the proportion with the size of the entire DW, and also it is observed that the size of the dimension tables are negligible when compared with the size of the fact table.

Example: The same SQL query seen previously, made on the normalized schema, requires up to three joins (with tables Category, Month, Quarter), resulting in performance degradation.

Snowflake schema

The snowflake schema also allows us to represent the many-to-many relationships, thus overcoming one of the star schema limitations.

Example: In this star schema, the sales may have been followed by a single agent and an agent can follow different sales.



Snowflake schema

Example (Cont.): However, in some environments, it may be that a sale is followed by multiple agents. The star schema is therefore no longer sufficient.

We can choose between two different alternatives.

Snowflake schema

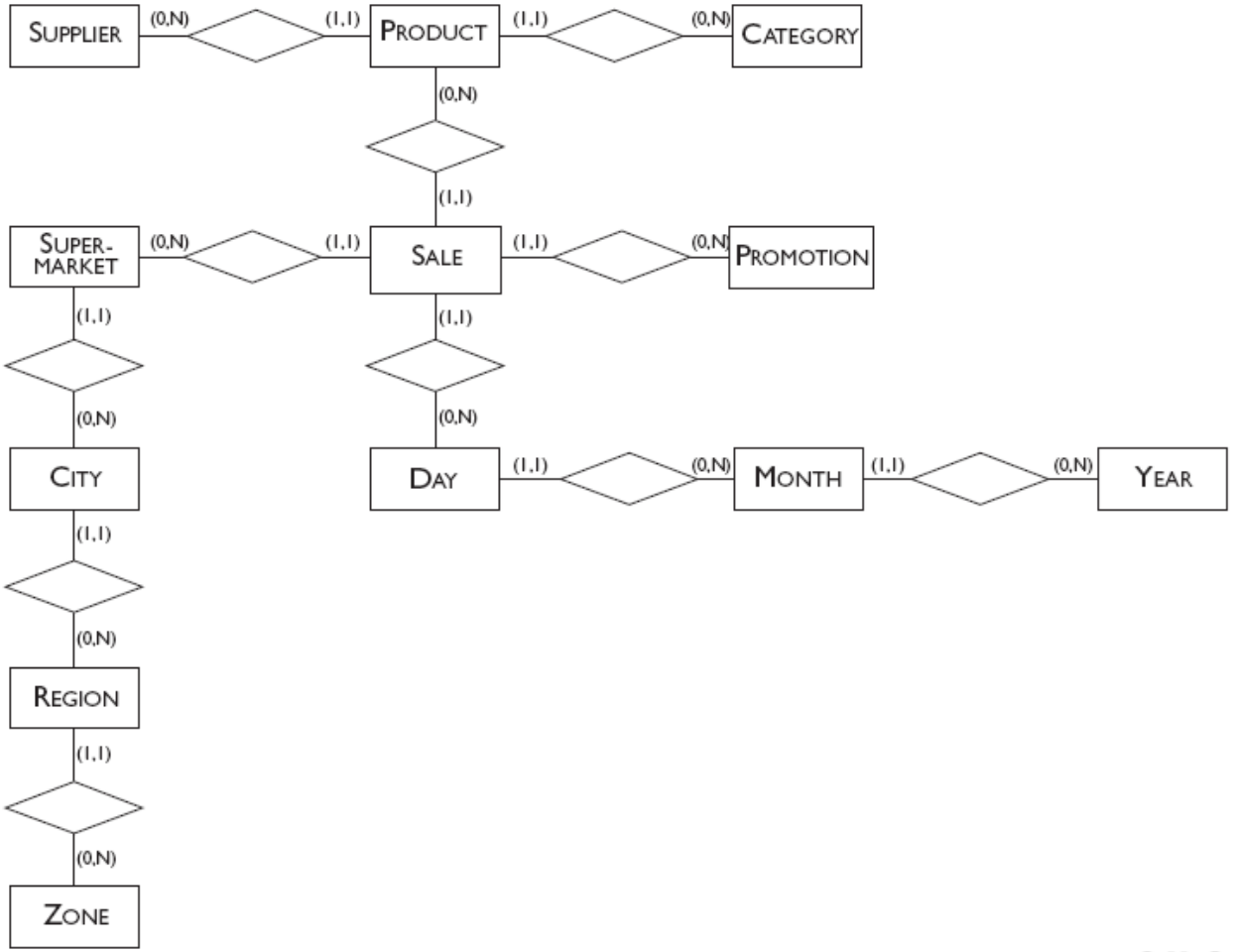
1. The insertion in the fact table of a record for each of the agents involved in the sale, which leads to the insertion, for each sale, of as many identical records as the agents involved in such a sale, with the exception of the agent code, which of course will be different.
   - In this case, however, any query will consider several times the same measures, distorting the results of the sum, average, or any other operation of aggregation.
   - To solve this problem we will need to make sure that, during querying, the formulated query considers only once every sale using controls on the primary key of the sale.

Snowflake schema

2. Using intermediate tables that, as in a classical ER models, make it possible to decompose the many-to-many relationships in two one-to-many.

Snowflake schema

Example of snowflake schema for a chain of supermarkets.



The time dimension

- From the point of view of the dimensional model of the DW, one might ask what the real usefulness of having in the schema an explicit temporal dimension: it would be enough to include the date between the keys of the fact table, and then use the normal semantics of SQL language on data-type to select a month rather than a year.
- This observation is certainly valid as long as the use of the time variable remains bound by day, month and year, in which case the time dimension table would actually be redundant.

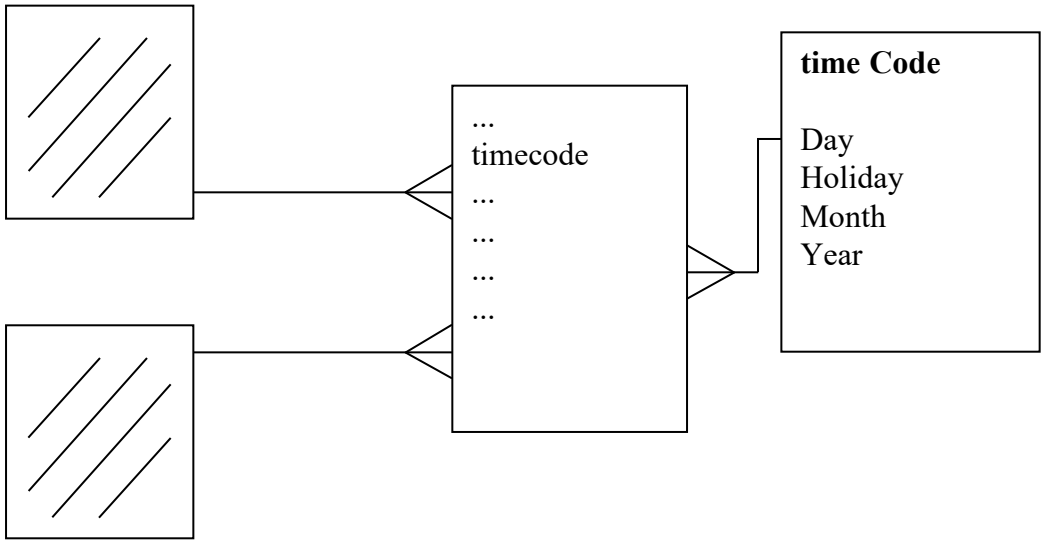
The time dimension

However, queries could be more detailed.

Example: identify all the data on the last Saturday of each month, or consider only weekdays, or only consider the first week of the summer months only.

- In each of these cases the storage of simple date as one of the attributes of the table of the facts is obviously not enough.
- For these reasons, in DW systems it is always used a special dimension table.

The time dimension



Note the separation between years, months and days, as well as the use of appropriate flag to indicate special conditions such as the fact that one day is a public holiday or not. In this way the selection, for example, of only the public holidays is done by selecting in the dimension *Time* only those records with the flag *Holiday* opportunely set.

Exercise 1

Design a data warehouse for the customers of a bank, considering as facts the amounts of the various transactions and as dimensions of analysis the time, the place (assuming that the bank has many branches), the customer, and transaction characteristics (for example, withdrawal from ATM, debit by credit card, payment by check, payment for direct remittance, withdrawal by check, etc.). Create both the star schema and the snowflake schema.

Exercise 2

Design a data warehouse for the Acme Credit Card Company. For every purchase made by a customer with the Acme credit card, we know the data for the product / service purchased, the holder of the credit card, the day of purchase and the store / company that provided the product / customer service which has an agreement with the Acme company. Define the possible attributes that might be used for an analysis for planning marketing strategies and a logical schema taking into account that the reference database is relational.

Implementation of some operators

How can some operators of multi-dimensional conceptual model be implement in the relational logical model?

The operation of *roll-up* translates into an SQL query with a simple structure:

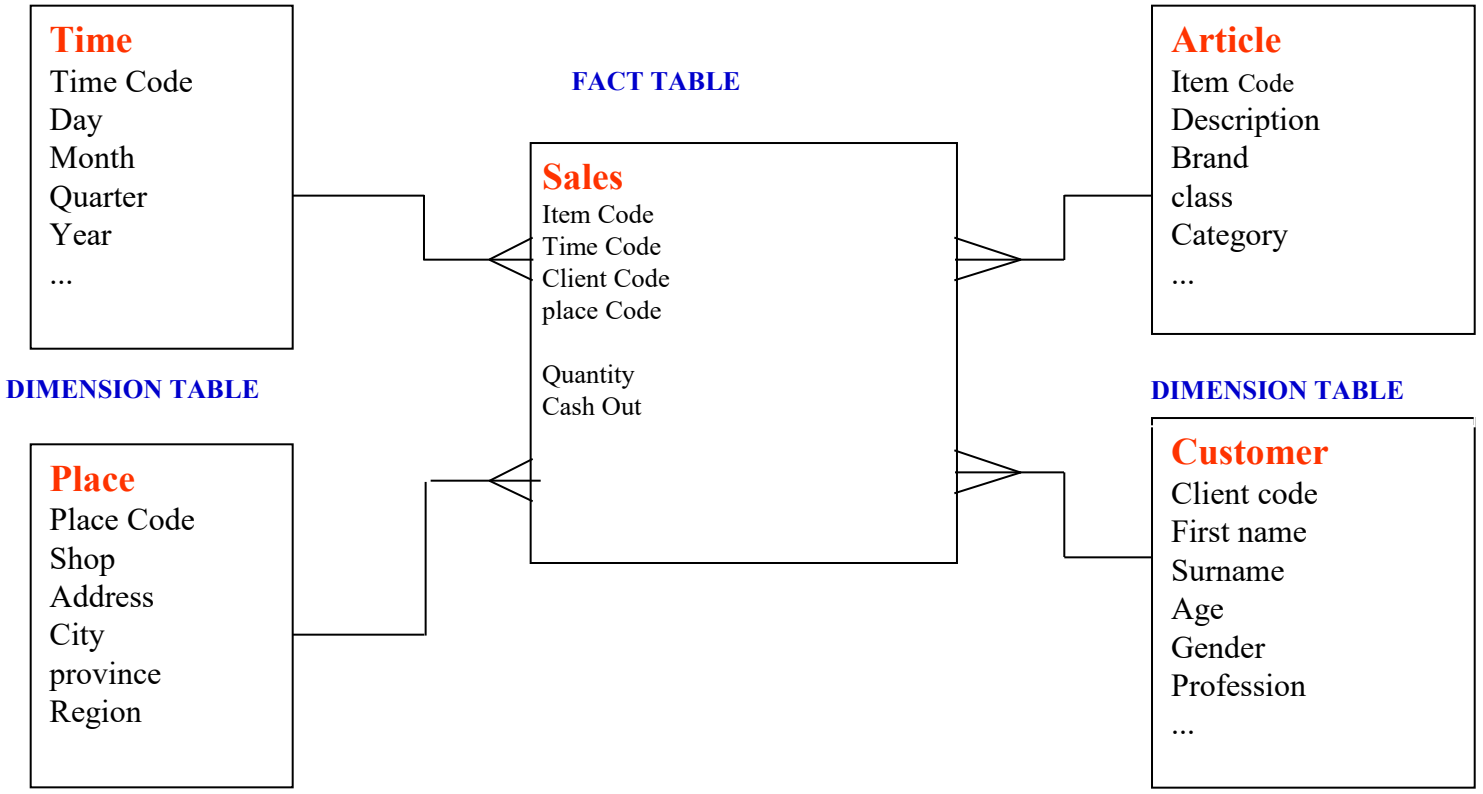
SELECT $D_1.L_1$, ..., $D_n.L_n$, $Aggr_1(F.M_1)$, ..., $Aggr_k(F.M_k)$
FROM Facts AS F,
$Dimension_1$ AS D_1, …, $Dimension_n$ AS D_n,
WHERE Join-predicate (F, D_1) AND ... AND Join-predicate (F, D_n) AND selection-predicate
GROUP BY $D_1.L_1$, ..., $D_n.L_n$
ORDER BY $D_1.L_1$, ..., $D_n.L_n$

Implementation of some operators

- Facts is the name of the table of facts,
- M_j is the j-th measure name of the table Facts,
- D_i is the name of the i-th dimensional table Facts,
- L_i is the name of the level of the i-th dimension table against which we want to perform the roll-up,
- $Aggr_x$ indicates an aggregate function (for example, sum)
- Join-predicate (F, D_i) indicates the join condition between the fact table and the i-th table dimension
- selection-predicate indicates a possible condition of selection on dimension tables.

Implementation of some operators

Example:
Consider the following star schema:



Implementation of some operators

Example (Cont):

A roll-up which, applied to the table Sales eliminates the dimension Place and Customer and returns the total sales in 2003 grouped by quarter and article category:

```
SELECT A.Category, T.Quarter, SUM (V.Quantity)
FROM Sales AS S, Article AS A, Time AS T
WHERE S.TimeCode = T.TimeCode AND
   S.ArticleCode = A.ArticleCode AND  T.Year =2003
GROUP BY A.Category, T.Quarter
ORDER BY A.Category, T.Quarter
```

Implementation of some operators

The high frequency of use of the aggregations in the context of operations on data warehouses has suggested the introduction in the SQL standard of a very powerful operator, called ***cube***, which performs all the possible combinations on a table based on the grouping attributes specified.

Example:

```
SELECT City, Category, SUM (Quantity) AS SalesValues
FROM Sales AS S, Article AS A, Place AS P
WHERE S.ArticleCode = A.ArticleCode AND
   S.PlaceCode= P.PlaceCode
GROUP BY CUBE(City, Category)
```

One possible result of this statement is as follows:

Implementation of some operators

City	Category	SalesValues
Rome	CD Player	361
Rome	DVD Players	1280
Rome	Televisions	1187
Rome	Video recorders	458
Milan	CD Player	393
Milan	DVD Players	1685
Milan	Televisions	1419
Milan	Video recorders	429
Rome	*ALL*	*3286*
Milan	*ALL*	*3926*
ALL	CD Player	754
ALL	DVD Players	2965
ALL	Televisions	2606
ALL	Video recorders	887
ALL	ALL	7212

aggregations were calculated on *all* the possible combinations of attributes present in clause group by cube.

It is also noted that, to represent the aggregation, the polymorphic value ALL is used, which (as NULL) is present in all domains and corresponds to the set of all possible values in that domain.

Implementation of some operators

The complexity of the cube operator grows in combinatorial manner with the increase in the number of grouping attributes. For this reason in the standard SQL there is also an extension where the aggregations are progressive with respect to the order of grouping attributes. This is the clause rollup which replaces the clause cube.

Example:

```
SELECT City, Category, SUM (Quantity) AS SalesValues
FROM Sales AS S, Article AS A, Place AS P
WHERE S.ArticleCode = A.ArticleCode AND S.PlaceCode=
    P.PlaceCode
GROUP BY ROLLUP(City, Category)
```

Implementation of some operators

One possible result of this statement is as follows:

City	Category	SalesValues
Rome	CD Player	361
Rome	DVD Players	1280
Rome	Televisions	1187
Rome	Video recorders	458
Milan	CD Player	393
Milan	DVD Players	1685
Milan	Televisions	1419
Milan	Video recorders	429
ALL	CD Player	754
ALL	DVD Players	2965
ALL	Televisions	2606
ALL	Video recorders	887
ALL	ALL	7212

Sales for the city are not calculated.

Implementation of some operators

The clauses cube and rollup are found in many commercial DBMS (e.g. Oracle) albeit in different forms from what is suggested by the standard.

So with relational systems, we can perform simple operations under the multidimensional model.

Materialization of views

Many questions on DW repeatedly require aggregations and very laborious synthesis. In this case it may be convenient to evaluate views that express the aggregate data "once and for all", and memorize them. This technique is called *materialization of views*.

The materialization saves execution time, but also increases the already large mass storage requirements of a data warehouse.

In the absence of the concept hierarchies, it is possible to create 2^n distinct cuboids through all possible combinations of n dimensions.

The presence of hierarchies along several dimensions makes even greater the number of distinct cuboids.

Materialization of views

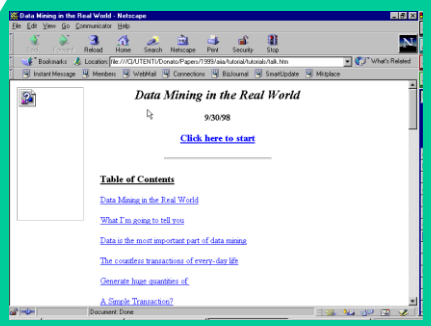
If we denote by L_i the number of hierarchical levels associated with the i-th dimension, for an n-dimensional data cube we can calculate a total number of cuboids equal to:

$$T = \prod_{i=1}^{n} (L_i + 1)$$

So the total materialization of all the possible cuboids corresponding to all data cubes associated with the fact tables of a data warehouse requires an amount of space which is difficult manage, also considering that the DW evolves.

Therefore, it is better to only proceed to a partial materialization that involves only the cuboids with greater frequency of access.

Architecture of a Data Warehouse



external sources

operational databases

metadata

extraction
cleaning
plastic pro.
loading

data warehouse

used by

marts

used by

EIS

DSS

OLAP

data mining

queries

Using data from a DW

The last common level to all data warehouse architectures is the analysis. In fact, once the data has been cleaned up, integrated and processed, we must figure out how to get the highest information advantage

There are, basically, four different approaches, supported by as many categories of business intelligence tools, to query a DW by end users:

1. reporting,
2. OLAP,
3. DSS / EIS
4. data mining.

Reports

This approach (called *query and reporting*) is directed to users who need to access, at predetermined time intervals, to structured information in an almost invariable way.

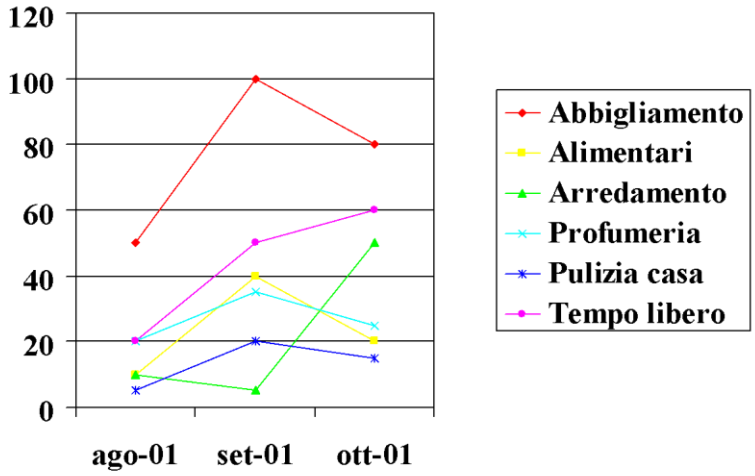
Example: Local health company must deliver to the public regional administrative offices monthly reports on hospitalization costs incurred. Of these reports, it is known a-priori the shape, which changes only as a result of variations in the current regulations. The designer can then design the query that generates the report and "freeze" it within an application so that it can be performed on the current data when the user has the actual need.

Reports

A *report* is defined by a query and a presentation.

- The query generally involves the selection and aggregation of multidimensional data: for example, can take the monthly collections during the last quarter for each product category.
- The presentation can be in tabular or graphical form (graph, histogram, pie-chart, etc.).

incassi (K)	ott-01	set-01	ago-01
Abbigliamento	80	100	50
Alimentari	20	40	10
Arredamento	50	5	10
Profumeria	25	35	20
Pulizia casa	15	20	5
Tempo libero	60	50	20



Reports

A reporting tool is evaluated not only according to the richness of the presentation of the reports, but also considering the flexibility for their distribution mechanisms. The report can be generated upon explicit user requests or automatically and periodically distributed to registered users, for example by email.

Reporting is not born with data warehousing. However, data warehousing has given to reporting two major benefits:

- Reliability and accuracy of results, because the data in the data warehouse are consistent and integrated;
- Timeliness, since the architectural separation between the transaction and the analytical processing, significantly improves the computing performance.

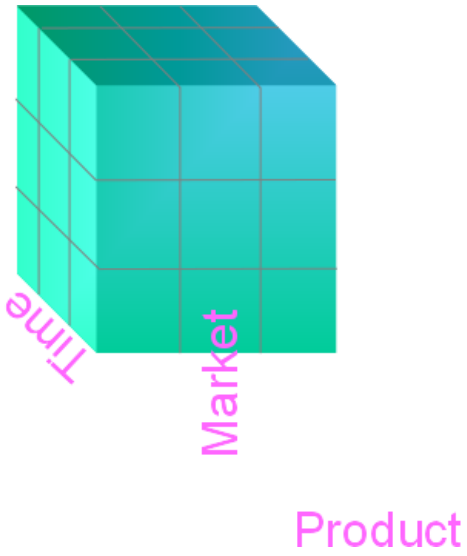
On-Line Analytical Processing (OLAP)

- It is perhaps the main data fruition way of a data warehouse.
- Term introduced by Codd (1993) as opposed to On-Line Transaction Processing (OLTP)
- Definition given by the OLAP Council:

"A category of software technology that enables analysts, managers and executives to gain insight into data through fast, consistent, interactive access to a wide variety of possible views of information that have been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user "

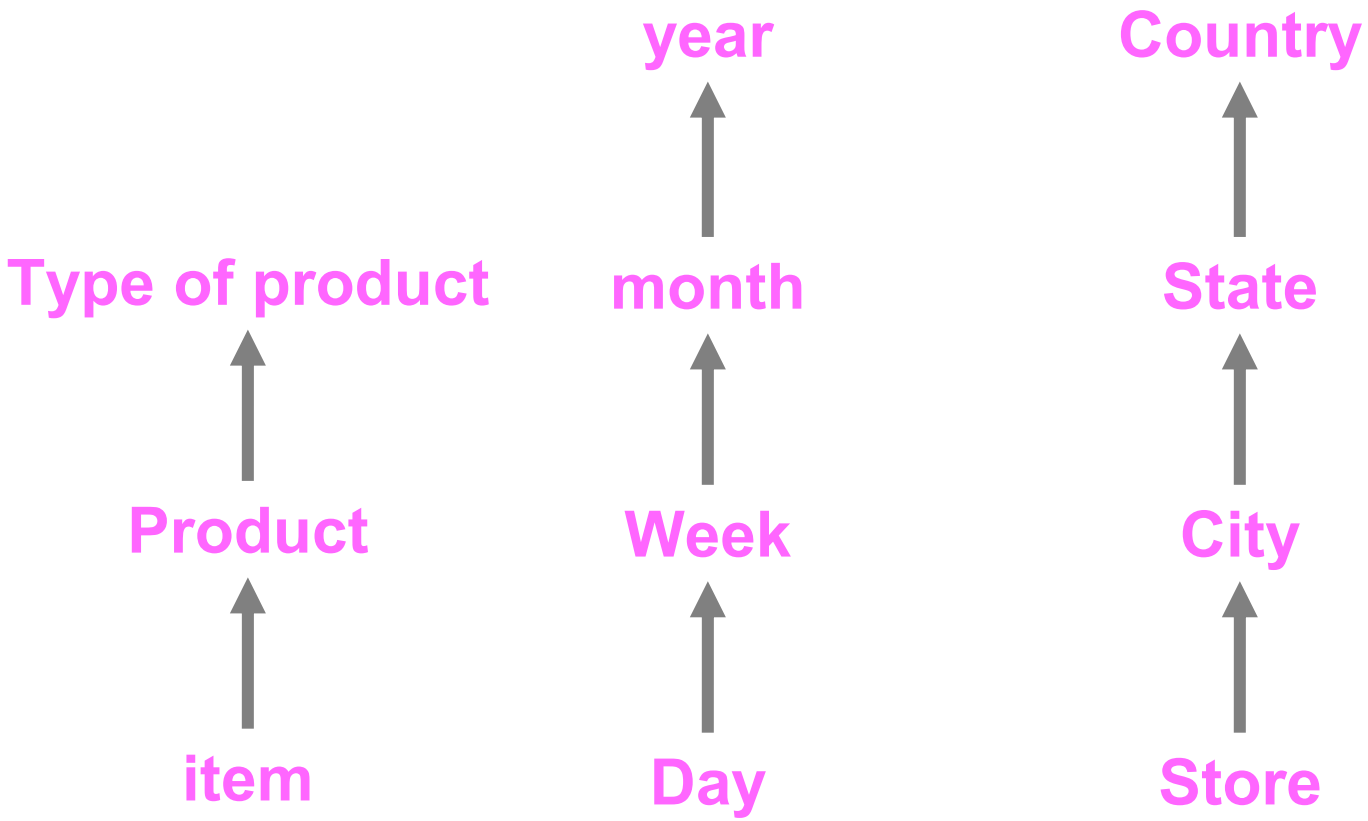
On-Line Analytical Processing (OLAP)

- Idea behind: Users should be able to manipulate the data models of an enterprise through many dimensions of analysis in order to understand the changes that are taking place.
- The data used in OLAP should be in the form of multidimensional cubes.



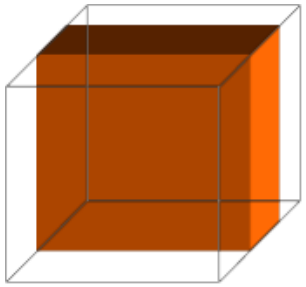
Dimensional hierarchies

Each dimension can be structured hierarchically.

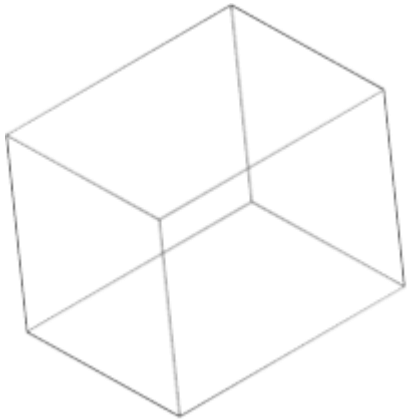


OLAP Operations

- *Rollup*: Decrease the level of detail
- *Drill down*: Increase the level of detail
- *Slice-and-dice*: Selection and projection



- *Pivot*: Re-orientation of the dimensional view of the data



Implement the multi-dimensionality of OLAP

- *Multi-dimensional databases* (MDDB)
- To allow relational DB (RDB) to treat the multidimensionality two types of tables are introduced:
  - *Fact table*: Contains numerical measures of a fact. It is long and narrow.
  - *Dimension Tables*: Contain pointers to the fact table. It has a different table for each dimension. They are short and wide.
- OLAP technology is called *MOLAP / ROLAP* (Multidimensional / relational OLAP) if it uses an MDDB / RDB.

OLAP vs. DSS

- The OLAP technology is considered an *extension* of the original DSS technology.
- The DSSs are tools that access, for the purposes of analysis, to the data in tables of a relational DBMS.
- The OLAP tools access and analyze multidimensional data (typically three, up to ten dimensions per cube).
- These elaborate a specific query made by the user.

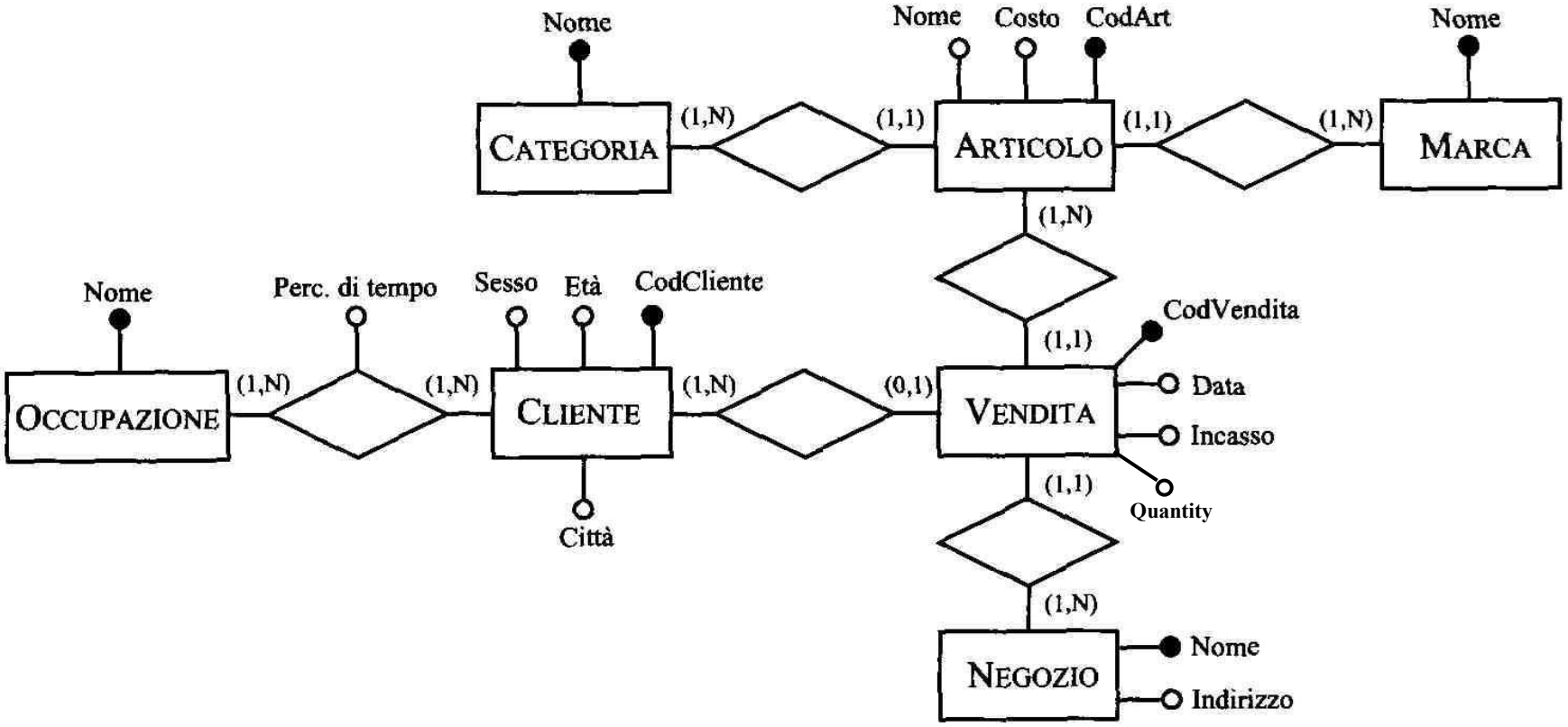
Data Warehouse ⇔ Data Mining

⇒ The transition from the data warehouse to data mining arises from the necessity of increasing the benefit that an organization can obtain from its data warehousing project.

⇐ After implementing a data mining solution, an organization may decide to integrate the solution into a more systematic approach to the use of data in decision making. The data warehouse provides an excellent vehicle for such integration.

Designing a data warehouse

Example: Suppose that the conceptual schema obtained in the process of integration is the following



Designing a data warehouse

Example (Cont):

Assume that the objectives of the analysis are:

1. Identification of sales trends (and corresponding gross receipts)
2. Analysis of temporal variation of the production costs of the items sold.

Designing a data warehouse

Example (Cont):

1. Identification in the input schema of the basic concepts for a multidimensional analysis (facts, measures and dimensions).

The facts are easily identifiable: entity Sale and the attribute Cost of the entity Article.

The measures of the first fact:

- Sales volume (attribute Quantity)
- The relative gross receipts (attribute gross receipts).

The only measurement for the second fact is the value of the cost itself.

Here we will focus on the first fact, for the first objective of the analysis.

Designing a data warehouse

Example (Cont):

2. Identification of the dimensions

A dimension can be identified by navigating the schema from the facts and including concepts that provide a way to group instances of facts, namely: relationships *one-to-many* and *categorical attributes*.

Every sale is related to the same item sold and each item is related to the respective category and brand ➔ sales can be examined on the basis of *type of product* at different levels of aggregation (individual product, product category, brand)

For some sales, we know the customer. Customers can be grouped by age, gender, and employment➔ *type of customer* is another dimension for the analysis of the sales.

Designing a data warehouse

Example (Cont):

2. Identification of the dimensions (cont.)

Based on these considerations we conclude that the place of sales is another possible dimension.

Finally, it is easy to identify the time dimension for sales (attribute Date of the entity Sale).

Designing a data warehouse

Example (Cont.):

3. Restructuring the ER schema

It is advisable to restructure the ER schema to represent facts and dimensions more explicitly.

First, we observe that the facts may be represented by attributes or relations (and not necessarily entities), thus they should be transformed into entities as they are of primary interest in the multidimensional analysis.

Moreover, in any dimension, it is useful to explicitly identify and represent various levels of aggregation.

Designing a data warehouse

Example (Cont.):

3. Restructuring the ER schema

Customers can be grouped by age and sex.

However if we want to aggregate customers for jobs we can not directly use the corresponding entity, because, according to many-to-many relationship between Customer and Occupation, each customer has different occupations, in general.

We can, however, replace this entity with a new entity Principal Occupation, which describes the employment of a customer for most of the time, so that the relationship is transformed from many-to-many to one-to-many.

Designing a data warehouse

Example (Cont.):

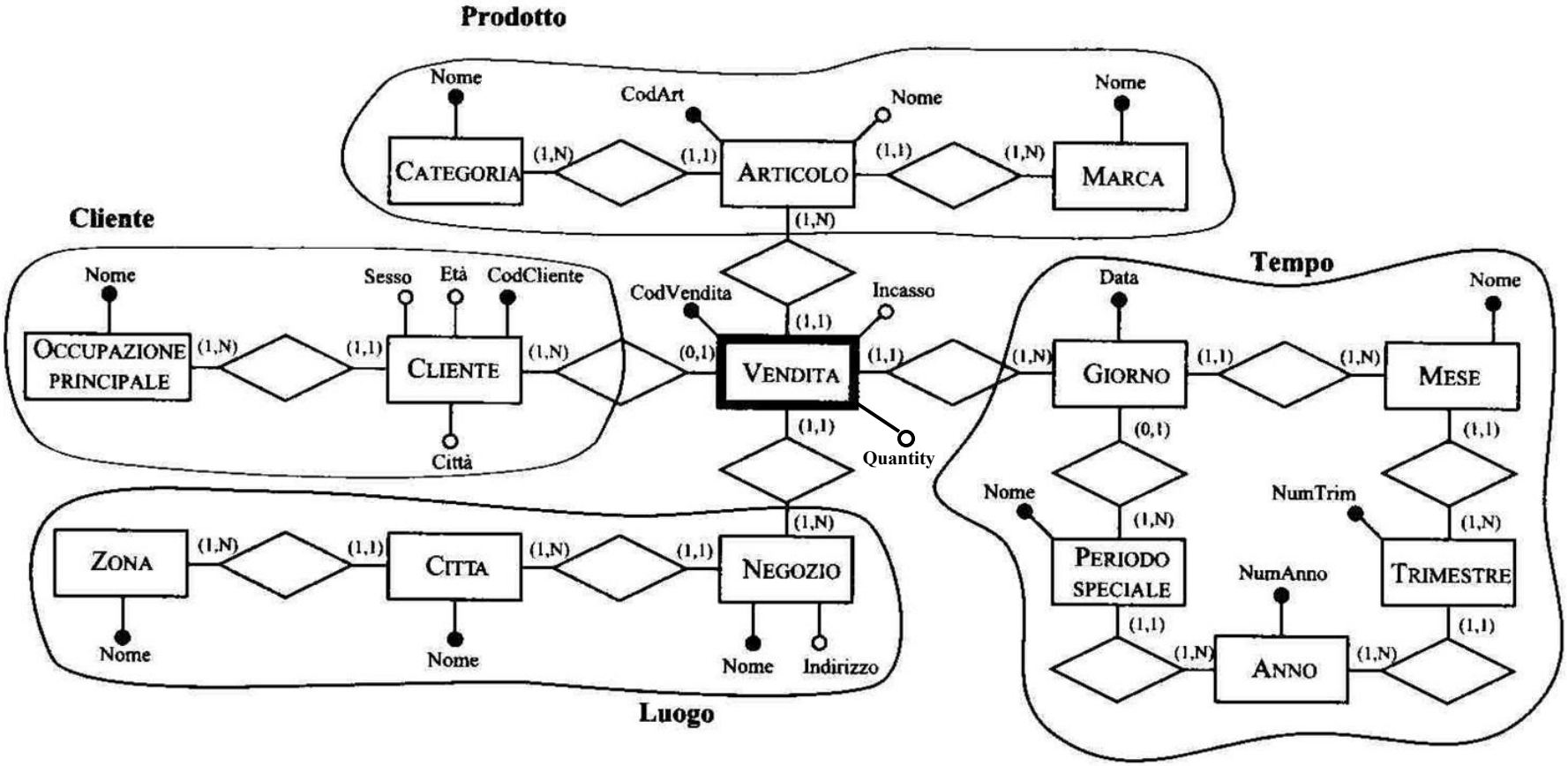
3. Restructuring the ER schema (cont.)

The dimension Place contains only the entity Shop. We may be interested in aggregating the stores based on city and geographical area (this information can be derived by the attribute Address and some geographical databases). Therefore we can introduce the explicit entities City and Area.

For the dimension Time, we may want to aggregate sales on a daily, monthly, quarterly, annually way and for special periods (eg. Easter, Christmas, opening schools, ...). To this end, we can introduce new entities and relationships.

Designing a data warehouse

Example (Cont): Result of restructuring



Designing a data warehouse

Example (Cont):

We can proceed with the logical design to relational database. Using a star schema we obtain:

Sale(SaleCode, ArtCode, CliCod, Date, Store, Amount, GrossReceipt)

Product(ArtCode, Category, Brand)

Customer(CusCode, Age, Gender, City, MainOccupation)

Time(Date, Month, Quarter, Period, Year)

Place(Shop, Address, City, State)

We have chosen to have as key of the fact table the identifier which is "closer" to the entity “sale”.

References

P. Atzeni, S. Ceri, P. Fraternali, S. & R. Paraboschi Torlone
Databases: Architectures and lines of evolution. Second edition.
McGraw-Hill Books Italy, 2007
Chapter 8, Sections 8.1, 8.2, 8.3, 8.4

Carlo Vercellis
Business Intelligence: Mathematical models and systems for decisions
McGraw-Hill Books Italy, 2006.
Part I, Chapters: 1,2,3

For more information:

Golfarelli M., S. Rizzi.
Data Warehouse, theory and practice of design, second edition
McGraw Hill, 2006