



ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN  
VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY

한-베정보통신기술대학교

Nhân bản – Phụng sự – Khai phóng

# Data Warehouse

Faculty of Computer Science



ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN  
VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY

한-베정보통신기술대학교

Nhân bản – Phụng sự – Khai phóng

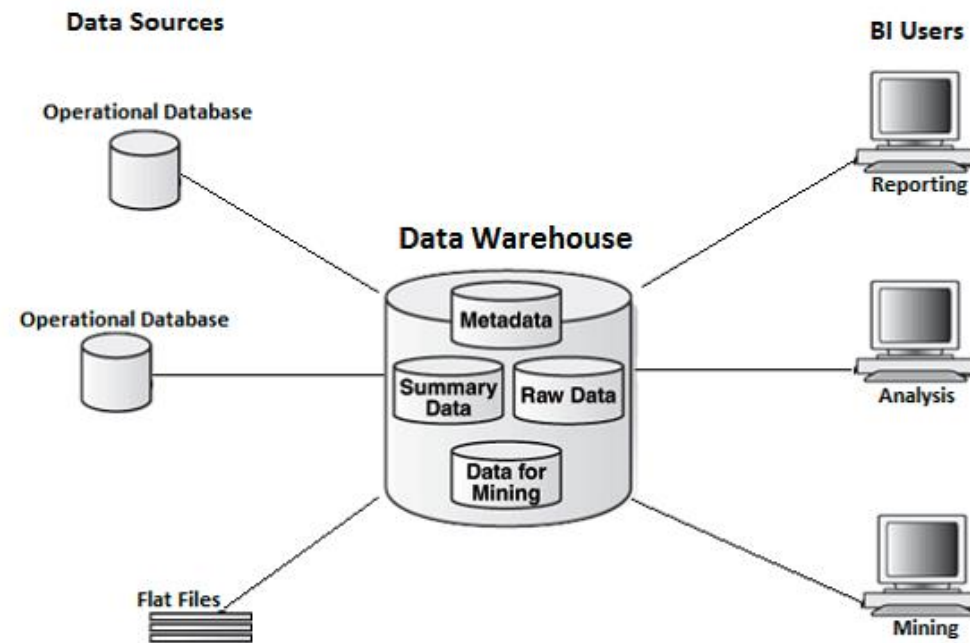
# Chapter 1: Data Warehouse Concepts

- **Data Warehouses**
- **Data warehouse architectures**
- **Multidimensional Model**
- **OLAP Operations**
- **Data warehouse Design**
- **Business Intelligence Tools**



- only designed and optimized to support **daily business operations** focused on transactions known as **operational databases** or **online transaction processing (OLTP)** systems.
- refers to techniques that **guarantee data consistency** (normalization)
- **did not satisfy the requirements for data analysis** to support the decision making processes.
- contain detailed data, do **not include historical data**
- **perform poorly when executing complex queries** that involve many tables or aggregate large volumes of data.
- users need to analyze the behavior of an organization as a whole, data from **several different operational systems must be integrated**.

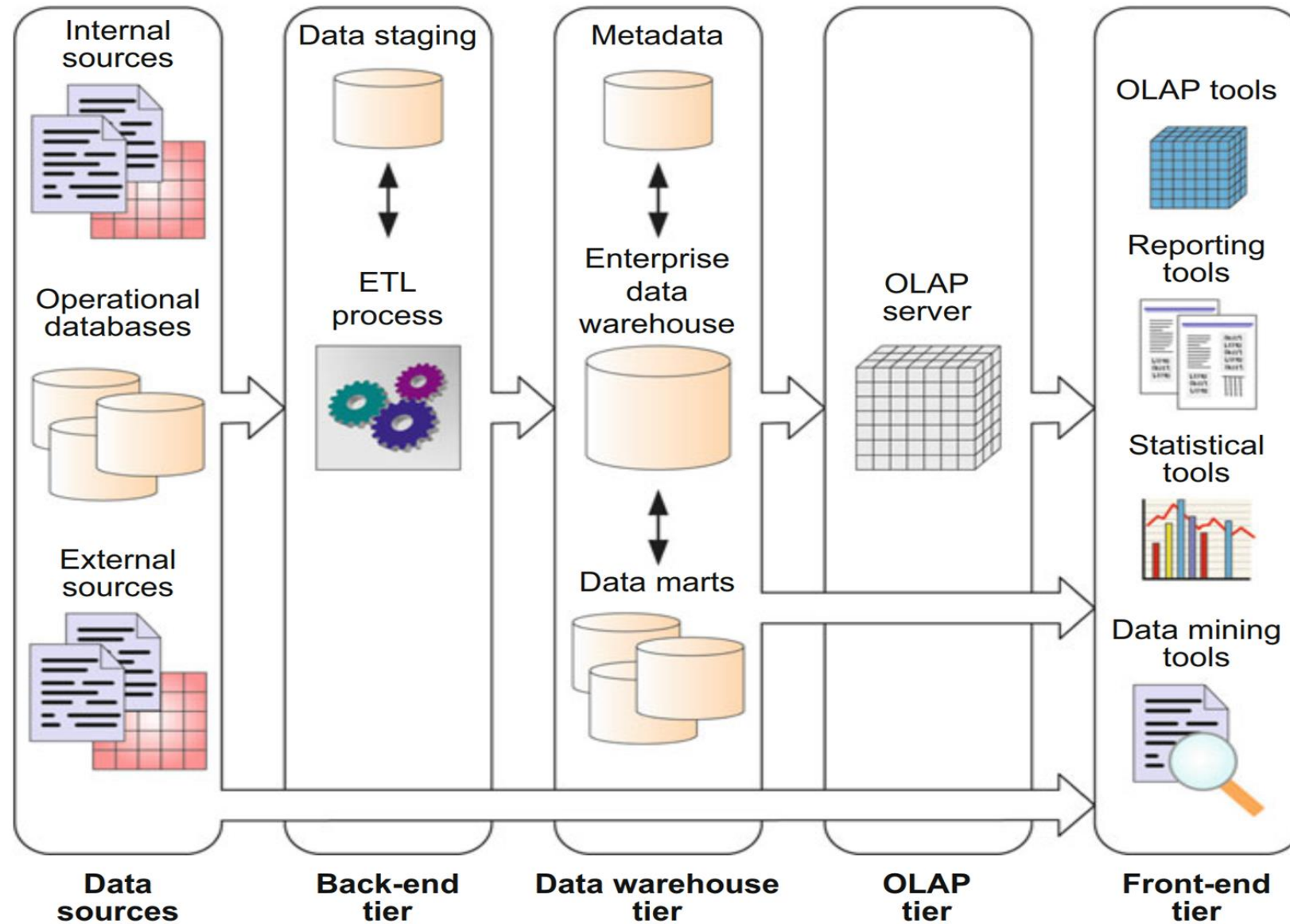
→ A **DW** is a particular database targeted toward decision support that takes data from various operational databases and other data sources and transforms it into new structures that fit better for the task of performing business analysis.



DATABASE	DATA WAREHOUSE
An organized collection of related data which stores data in a tabular format	A central location which stores consolidated data from multiple databases
Contains detailed data	Contains summarized data
Uses Online Transactional Processing (OLTP)	Uses Online Analytical Processing (OLAP)
Helps to perform fundamental operations of a business	Helps to analyze the business
Less fast and less accurate	Faster and accurate
Application oriented	Subject oriented
Tables and joins are complex because they are normalized	Tables and joins are simple because they are denormalized
Design is helped by entity relationship modelling	Design is helped by data modelling technique

→ A **data warehouse (DW)** is a collection of **subject-oriented, integrated, nonvolatile, and time-varying** data to support management decisions. (*W.H. Inmon, Building the Data Warehouse, 1992*).

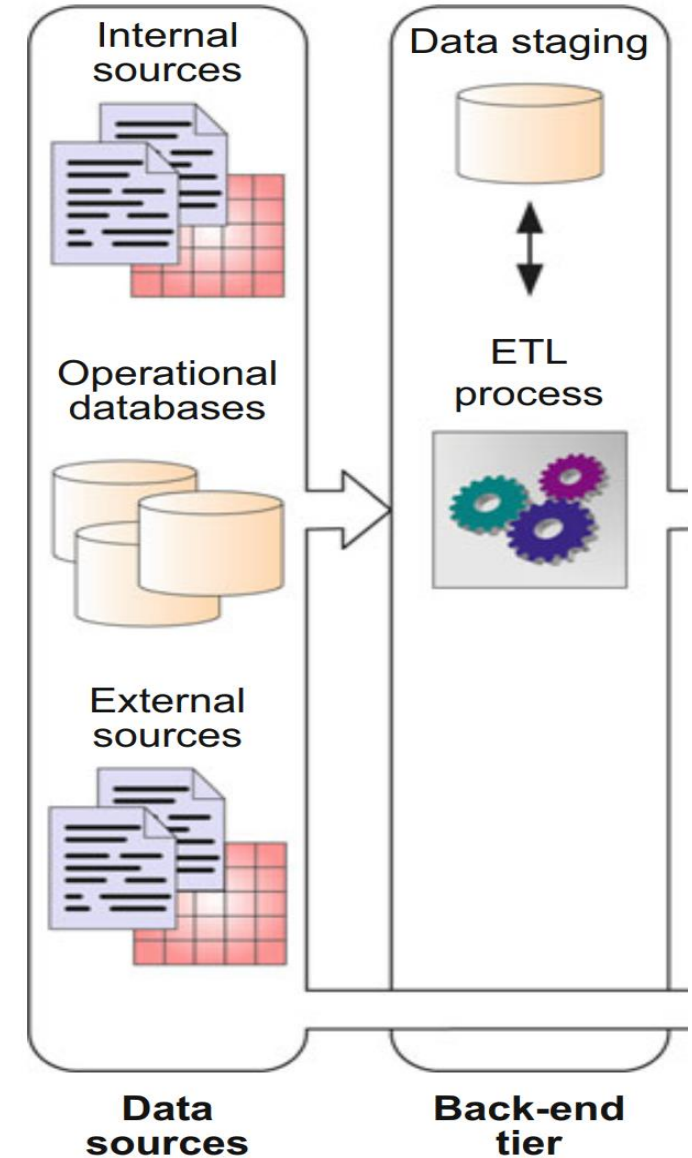
- **Subject oriented:** data warehouses focus on the analytical needs of different areas depending on the kind of activities performed by the organization.
- **Integrated:** data obtained from several operational and external systems must be joined together.
- **Nonvolatile:** durability of data is ensured by disallowing data modification and removal.
- **Time varying:** the possibility of retaining different values for the same information, as well as the time when changes to these values occurred.

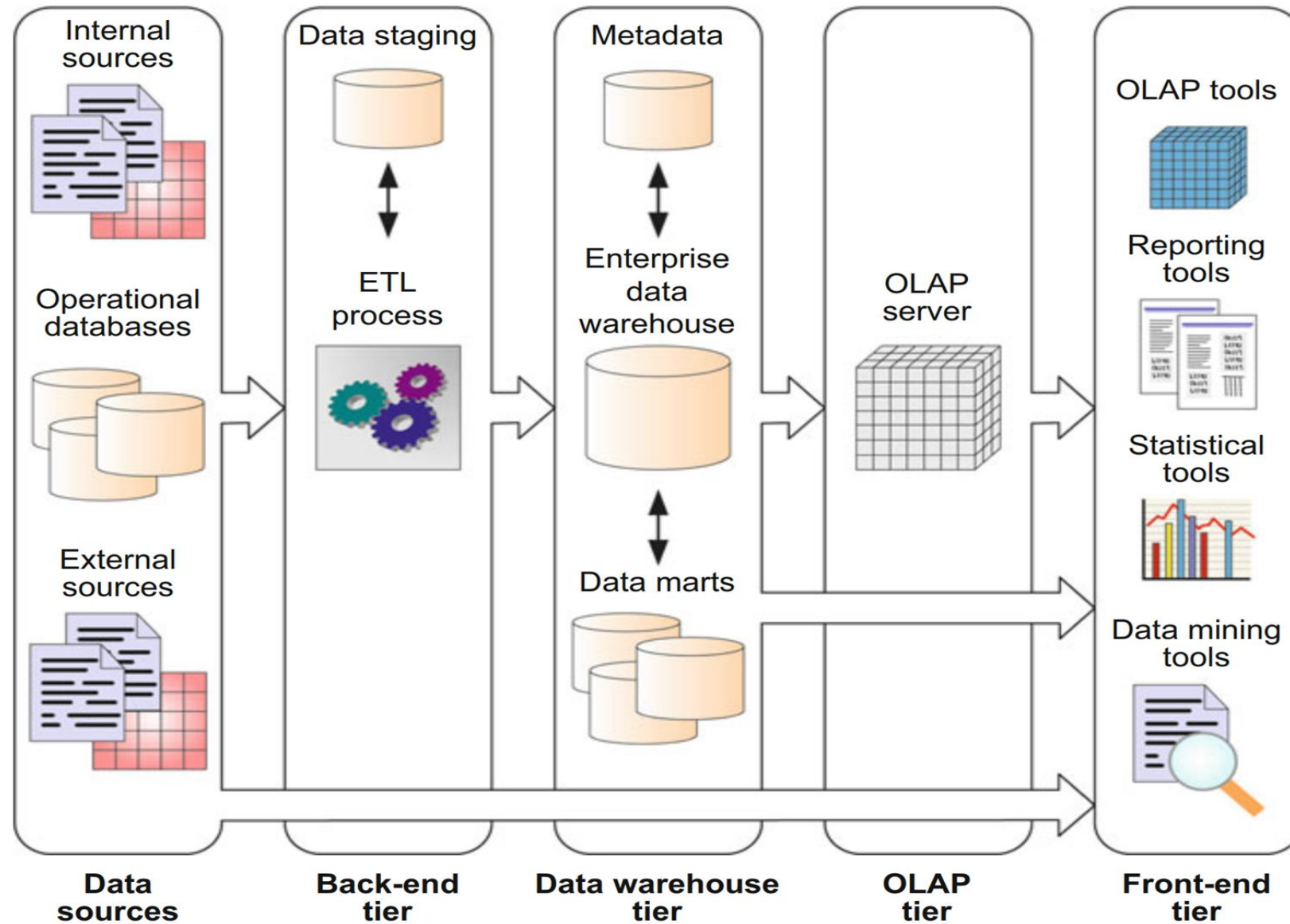




- **Back-end tier** is composed of ***extraction***, ***transformation***, and ***loading (ETL)*** tools used to feed data into the data warehouse
- ❑ **Extraction** gathers data from multiple, heterogeneous data sources from operational databases and other data sources (internal or external) from the organizations.
  - ❑ **Transformation** modifies the data from the format of the data sources to the warehouse format.
    - ✓ **Cleaning**: removes errors and inconsistencies in the data and converts it into a standardized format.
    - ✓ **Integration**: reconciles data from different data sources, both at the schema and at the data level.
    - ✓ **Aggregation**: summarizes the data obtained from data sources according to the level of detail, or granularity, of the data warehouse.
  - ❑ **Loading** feeds the data warehouse with the transformed data includes refreshing the data warehouse, that is, propagating updates from the data sources to the data warehouse at a specified frequency (e.g., monthly or several times a day)

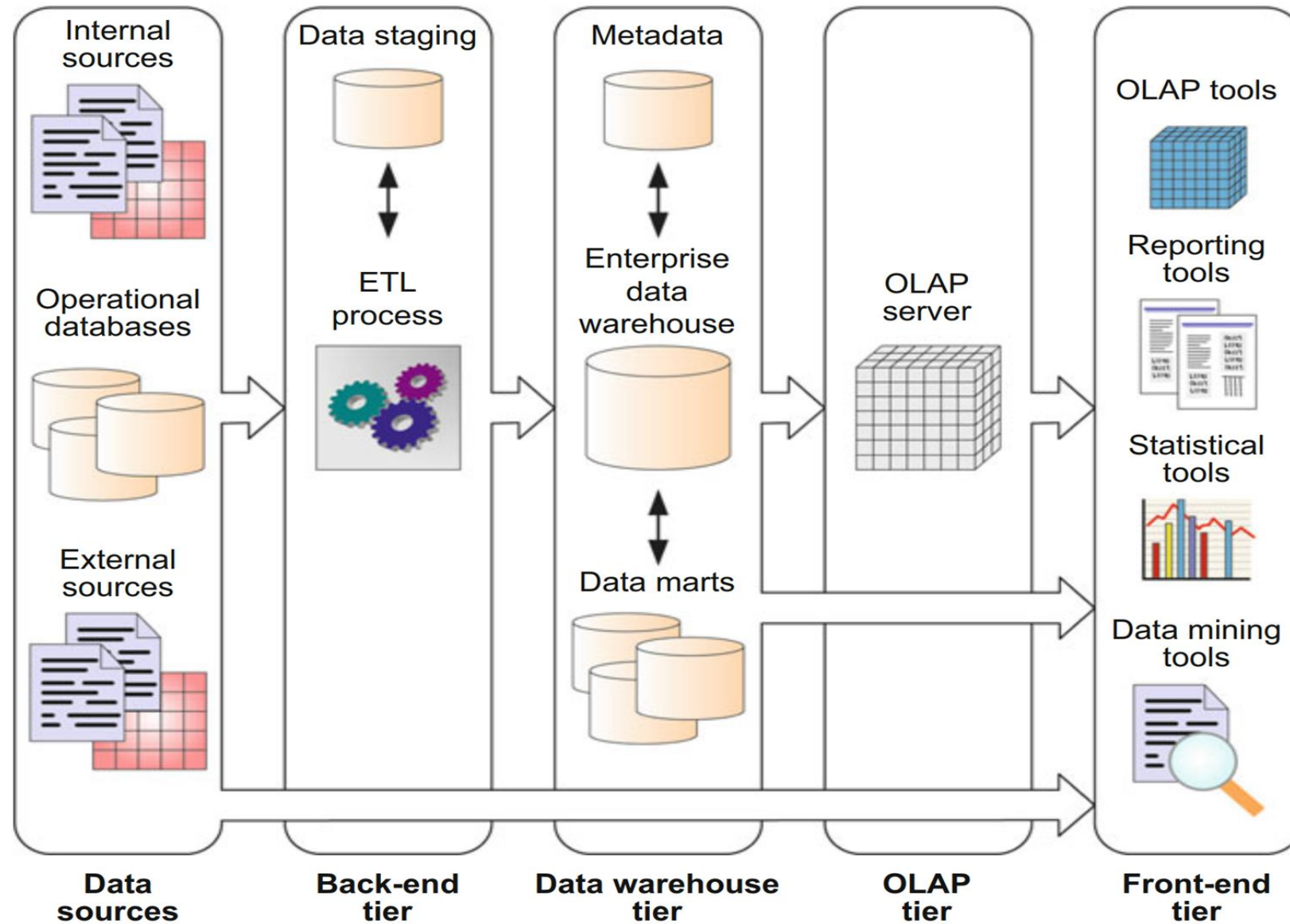
- **ETL processes** usually require a **data staging area**.
- It is an intermediate database usually called **operational data store** in which the data extracted from the sources undergoes successive modifications to eventually be ready to be loaded into the data warehouse.



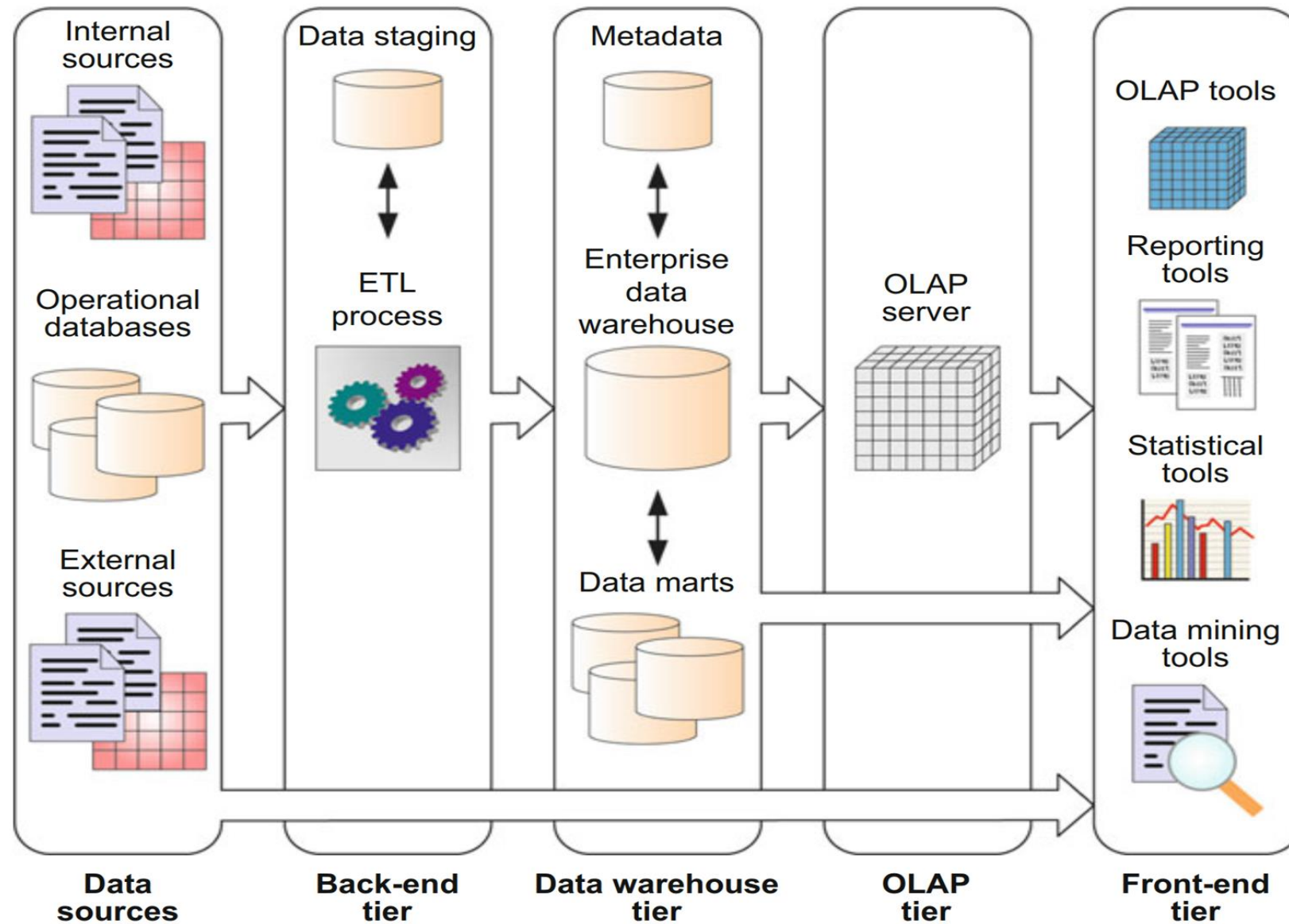


- **Data warehouse tier** stores information about the data warehouse and its contents composed of:
- An **enterprise data warehouse** is centralized and encompasses an entire organization.
  - A **data mart** is a specialized data warehouse targeted toward a particular functional or departmental area in an organization as a small local data warehouse either derived from an enterprise data warehouse or collected directly from data sources.
  - A **metadata repository** contain metadata (defined as “data about data”) describing the *data warehouse system* (e.g. its structure and the data marts), the *data sources* (e.g., their schemas, update frequencies, legal limitations, and access methods...), and the *ETL process*.





- **OLAP tier** is composed of an *OLAP server*, which presents business users with multidimensional data from data warehouses or data marts.
- Most database products provide OLAP extensions and related tools allowing the construction and querying of cubes, as well as navigation, analysis, and reporting.
- **MDX (MultiDimensional eXpressions)** is a query language for OLAP databases.
- The SQL standard has also been extended for providing analytical capabilities referred to as **SQL/OLAP**.

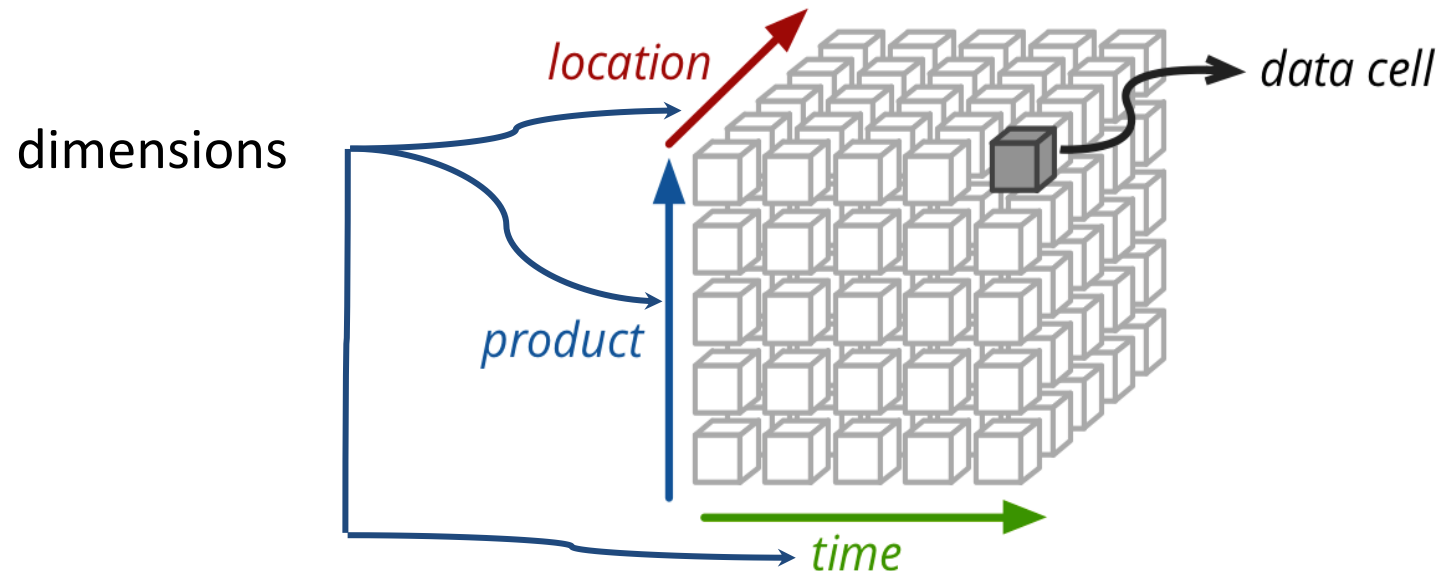


- **Front-end tier** contains client tools that allow users to exploit the contents of the data warehouse including:
- ◆ **OLAP tools** allow interactive exploration and manipulation of the warehouse data that facilitate the formulation of complex queries (called ***ad hoc queries***) involving a large amounts of data.
  - ◆ **Reporting tools** enable the production, delivery, and management of reports, which can be paper-based reports or interactive, web-based reports by using ***predefined queries*** (queries asking for specific information in a specific format that are performed on a regular basis).
  - ◆ **Statistical tools** are used to analyze and visualize the cube data using statistical methods.
  - ◆ **Data mining tools** allow users to analyze data in order to discover or predict valuable knowledge such as patterns and trends on the basis of current data.

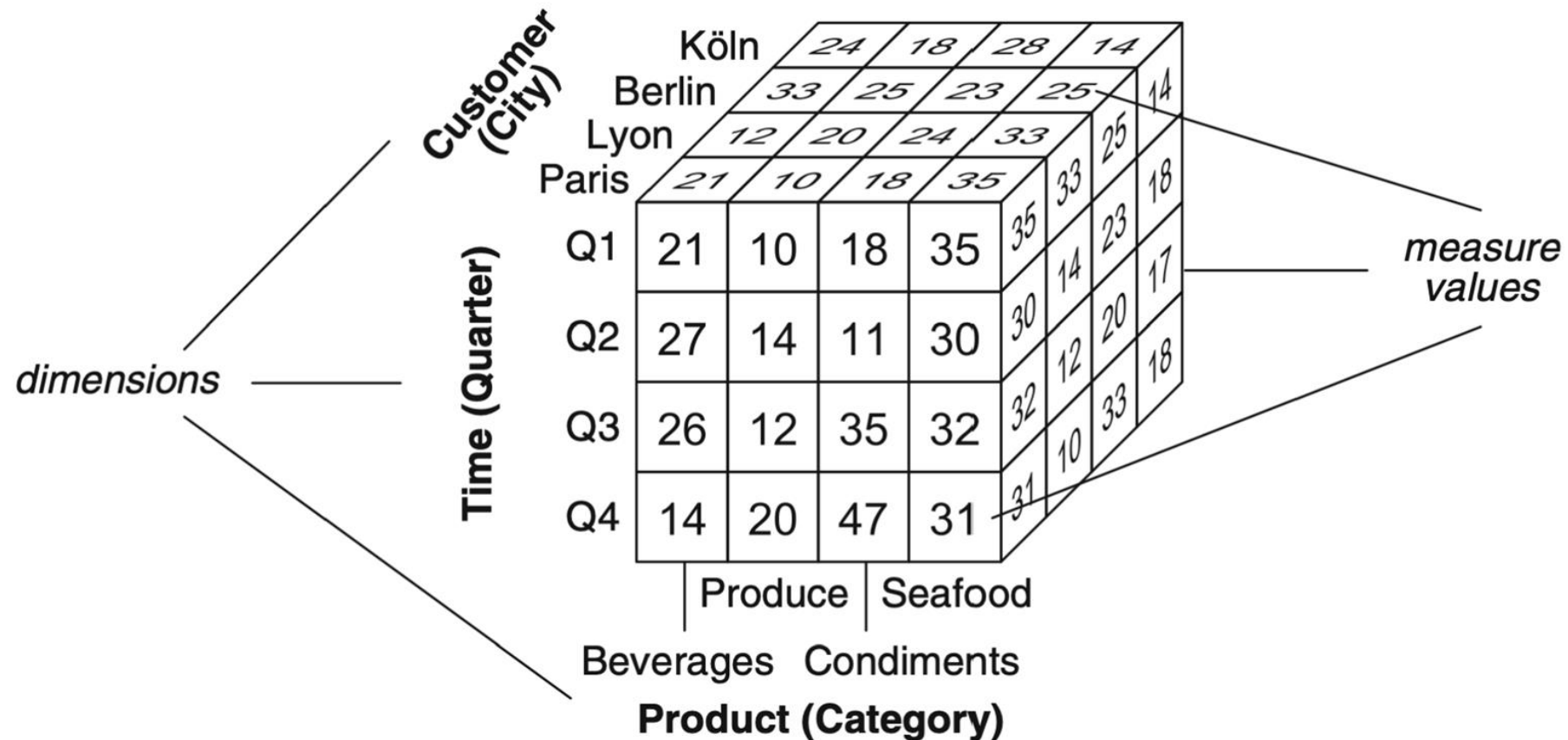


- Some of the components of DW can be missing in a real environment
- ***Only an enterprise data warehouse without datamarts***, or alternatively, an enterprise data warehouse does not exist.
- ***OLAP server does not exist*** and/or the client tools directly access the data warehouse indicated by the arrow ***connecting the data warehouse tier to the front-end tier***.
- ***Neither a data warehouse nor an OLAP server*** called a ***virtual data warehouse (the data sources to the front-end tier)***.

- DWs are based on a **multidimensional model**, where data are represented in an *n-dimensional space* as **a data cube** or **hypercubes** defined by data **cells** (or **facts**) and **dimensions**.
- ◆ data **cells** ( or **facts**) contain the **measures** (usually numeric values) to be analyzed.
  - ◆ **dimensions** used to see the measures from several perspectives correspond to the various business perspectives



A *three-dimensional* cube for sales data with dimensions **Product**, **Time**, and **Customer**, and a measure **Quantity**



→ A ***dimension level*** represents the *granularity*, or *level of detail*, at which measures are represented for each dimension of the cube corresponding to each axis of the cube.

*Example: Sales figures are aggregated to the levels Category, Quarter, and City, respectively.*

→ ***Instances of a dimension*** are called *members*.

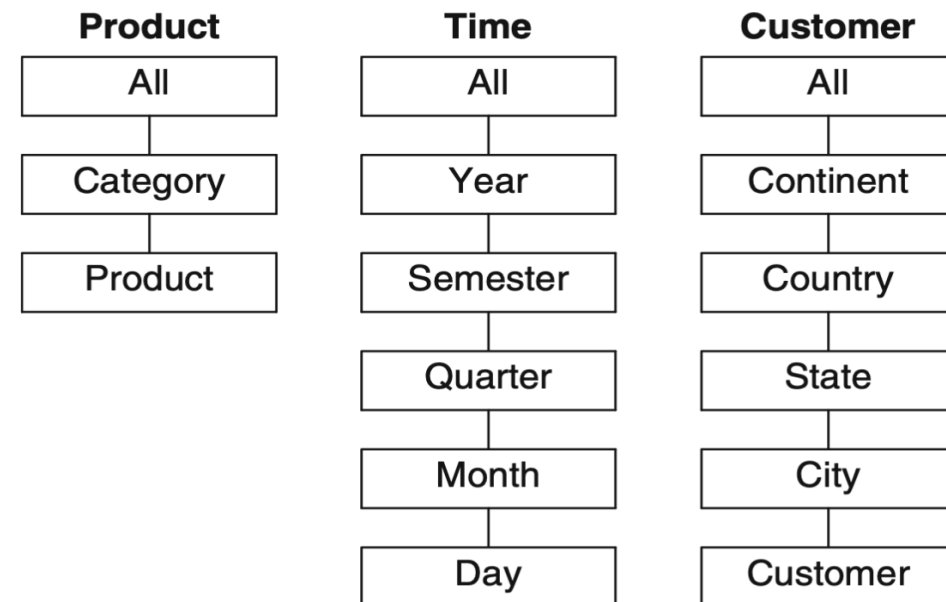
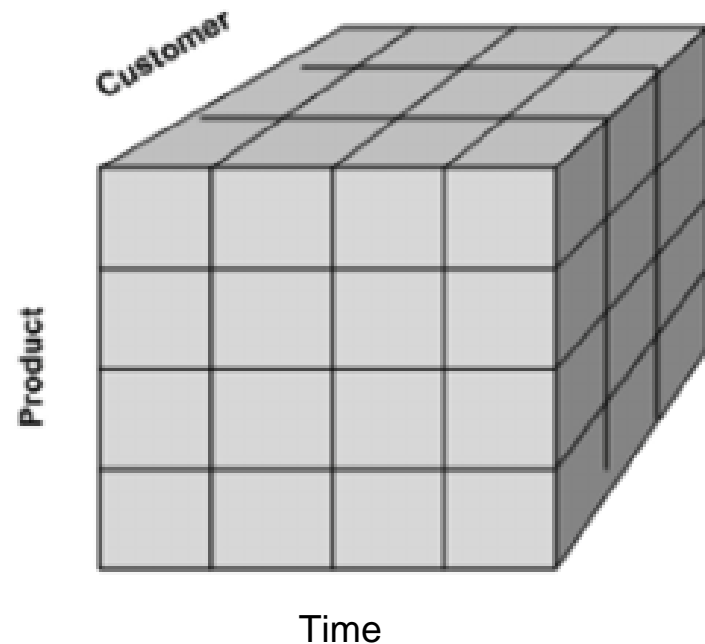
*Example: Seafood and Beverages are members of the Product dimension at the Category level.*

→ Dimensions also have associated ***attributes*** describing them.

*Example: the Product dimension could contain attributes such as ProductNumber and UnitPrice,*

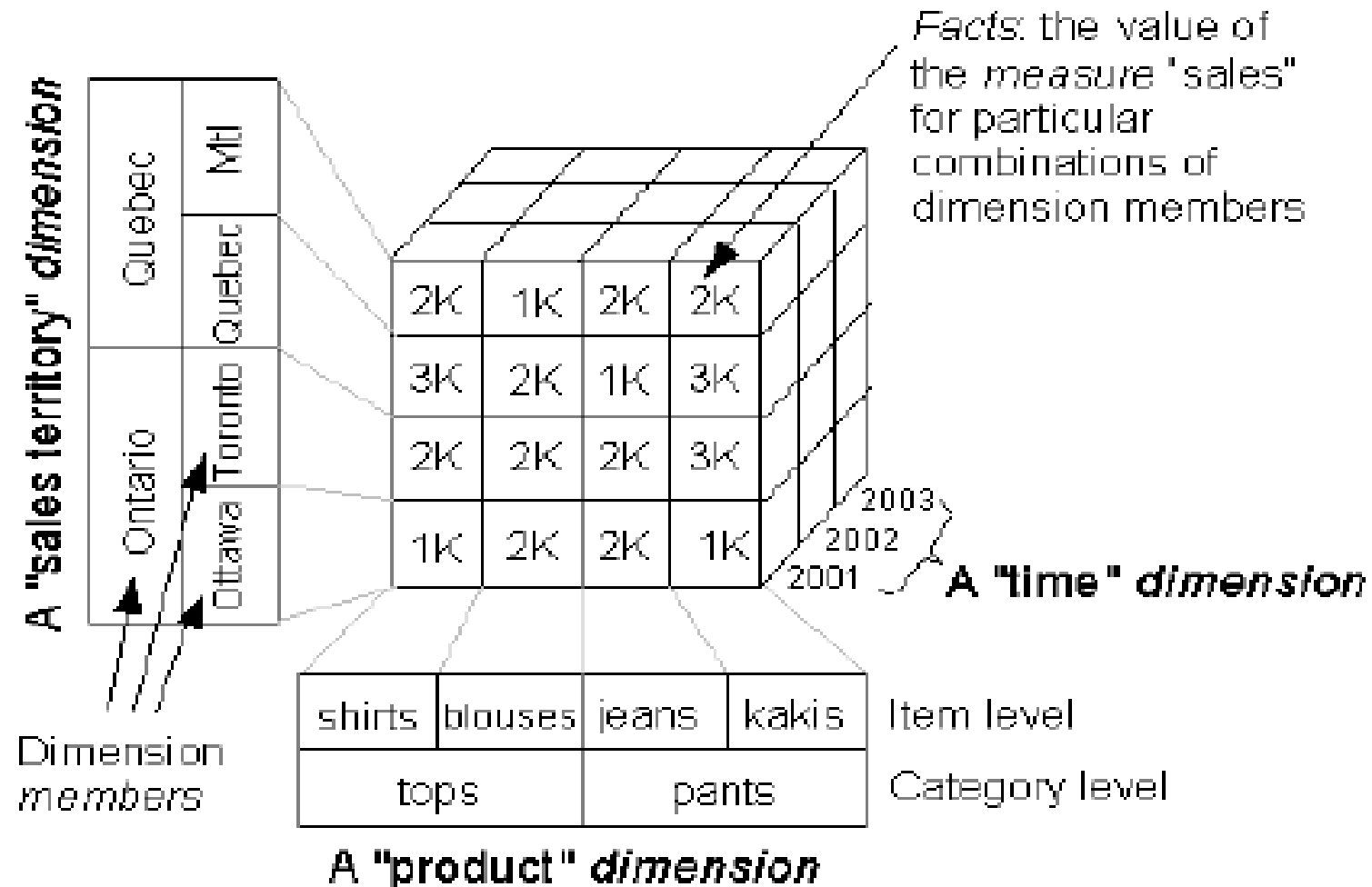


- **Hierarchies** allow viewing the data at several levels of detail by defining a sequence of mappings relating lower-level, detailed concepts to higher-level, more general concepts.
- The *lower level* is called the **child** and the *higher level* is called the **parent**.
- The hierarchical structure of a dimension is called the **dimension schema**
- A **dimension instance** comprises the members at all levels in a dimension



Hierarchies of the Product, Time, and Customer dimensions

## A "sales" data cube



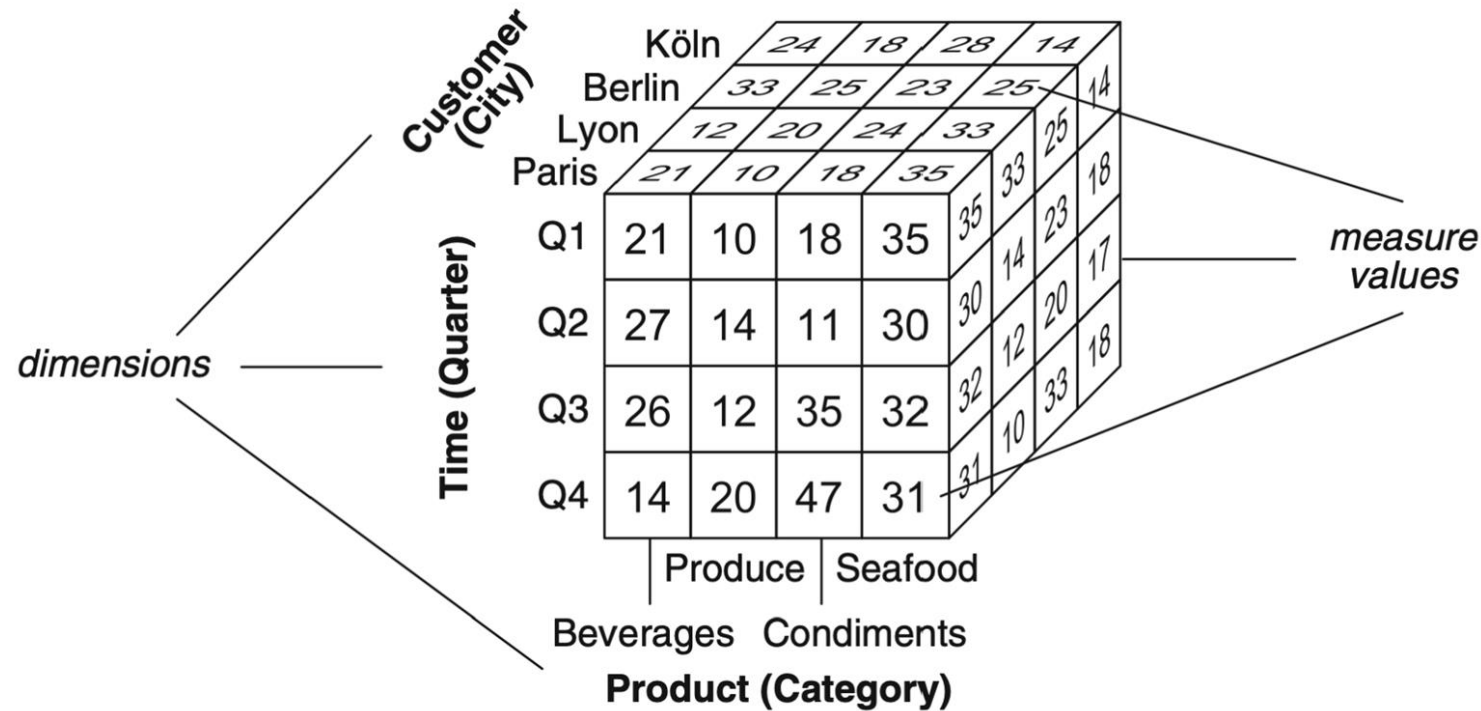
→ **Data cells** (**Data cube** or **facts**), have often associated *numeric values*, called **measures** which are used to evaluate quantitatively various aspects of the analysis at hand.

*Example:* a measure *Quantity*, indicating the number of units sold (in thousands) by category, quarter, and customer's city that might help to analyze sales activities in various stores

→ A data cell typically can contain several measures.

*Example:* another measure *Amount* indicates the total sales amount.

→ A data cube may be **sparse** or **dense** depending on whether it has measures associated with each combination of dimension values.



This depends on whether all products are bought by all customers during the period of time considered. For example, not all customers may have ordered products of all categories during all quarters of the year.



- Each measure in a cube is associated with an aggregation function that combines several measure values into a single one.
- Aggregation of measures takes place when one changes the level of detail at which data in a cube are visualized.
- This is performed by traversing the hierarchies of the dimensions.

*Example:*

if the *Customer* hierarchy is used for changing the granularity of the data cube from *City* to *Country*, then the sales figures for all customers in the same country will be aggregated using the ***SUM*** operation.

- **Summarizability** refers to the correct aggregation of cube measures along dimension hierarchies, in order to obtain consistent aggregation results.
- A set of conditions may hold to ensure summarizability:
  - ◆ **Disjointness of instances:** The grouping of instances in a level with respect to their parent in the next level must result in disjoint subsets.
  - ◆ **Completeness:** All instances must be included in the hierarchy and each instance must be related to one parent in the next level.
  - ◆ **Correctness:** It refers to the correct use of the aggregation functions.
- In order to define a measure, it is necessary to determine the aggregation functions that will be used in the various dimensions.

→ **Additive measures** can be meaningfully summarized along all the dimensions, using addition.

*Example:* the measure Quantity in the cube is additive: it can be summarized when the hierarchies in the Product, Time, and Customer dimensions are traversed.

→ **Semi-additive measures** can be meaningfully summarized using addition along some, but not all, dimensions.

*Example:* inventory quantities, which cannot be meaningfully aggregated in the Time dimension, for instance, by adding the inventory quantities for two different quarters

→ **Nonadditive measures** cannot be meaningfully summarized using addition across any dimension.

*Example:* item price, cost per unit, and exchange rate.

→ ***Distributive measures*** are defined by an aggregation function that can be computed in a distributed way. The data are partitioned into  $n$  sets and that the aggregate function is applied to each set, giving  $n$  aggregated values.

*Example:* The set of measure values {3, 3, 4, 5, 8, 4, 7, 3, 8} is divided into the subsets {3, 3, 4}, {5, 8, 4}, and {7, 3, 8}. Summing up the result of the distinct count function applied to each subset gives us a result of 8

→ ***Algebraic measures*** are defined by an aggregation function that can be expressed as a scalar function of distributive ones.

*Example:* The average, which can be computed by dividing the sum by the count, the latter two functions being distributive.

→ ***Holistic measures*** are measures that cannot be computed from other subaggregates.

*Example:* the median, the mode, and the rank.

- OLAP provides an interactive data analysis environment.
- **OLAP operations** allow *multiple perspectives* and *levels of detail* to be materialized by exploiting the dimensions and their hierarchies.
- OLAP Operations involves tasks to operate over a data cube in order to analyze data in different ways.
  - ◆ Roll-up
  - ◆ Drill-down
  - ◆ Slice and dice
  - ◆ Pivot (rotate)
  - ◆ Add measure
  - ◆ Sum Aggregation
  - ◆ Max/Min
  - ◆ .....

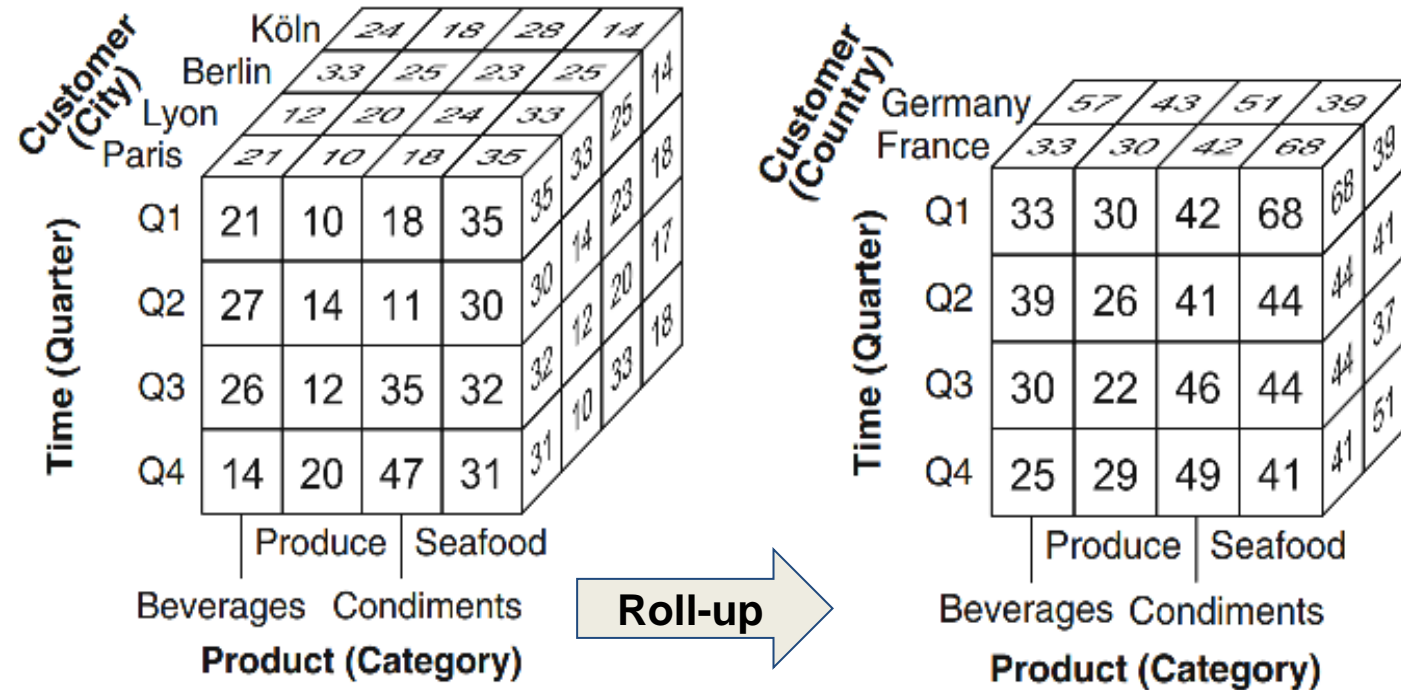


→ **Roll-up** performs aggregation on a data cube by climbing up a concept hierarchy for a dimension

**ROLLUP(CubeName, (Dimension → Level)\*, AggFunction(Measure)\*)**

*Example:* A roll-up operation is applied from the *City* level to the *Country* level along the *Customer* dimension to compute the sales quantities by country.

**ROLLUP(Sales, Customer → Country, SUM(Quantity))**

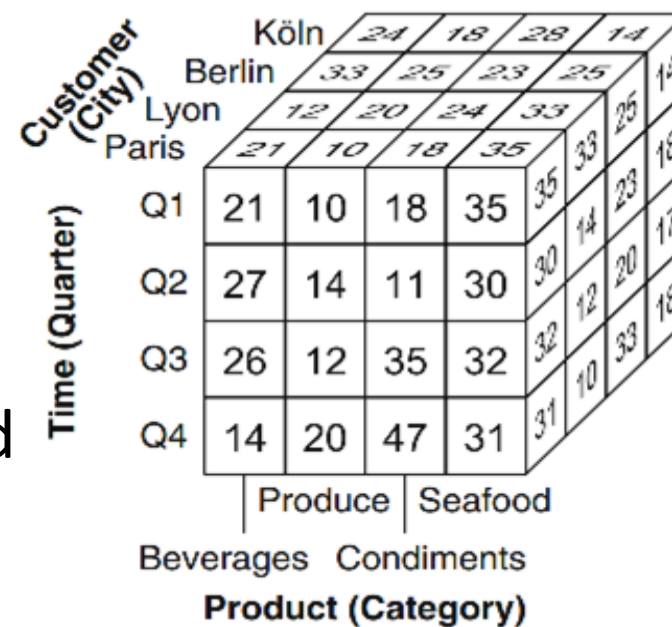


→ **Drill-up** is the *reverse* operation of roll-up. It is performed by stepping down a concept hierarchy for a dimension.

**DRILLDOWN(CubeName, (Dimension → Level)\*)**

*Example:* A drill-up operation is applied by stepping down from the *Quarter* level to the *Month* level along the *Time* dimension to find out whether the high value occurred during a particular month.

**DRILLDOWN(Sales, Time → Month)**

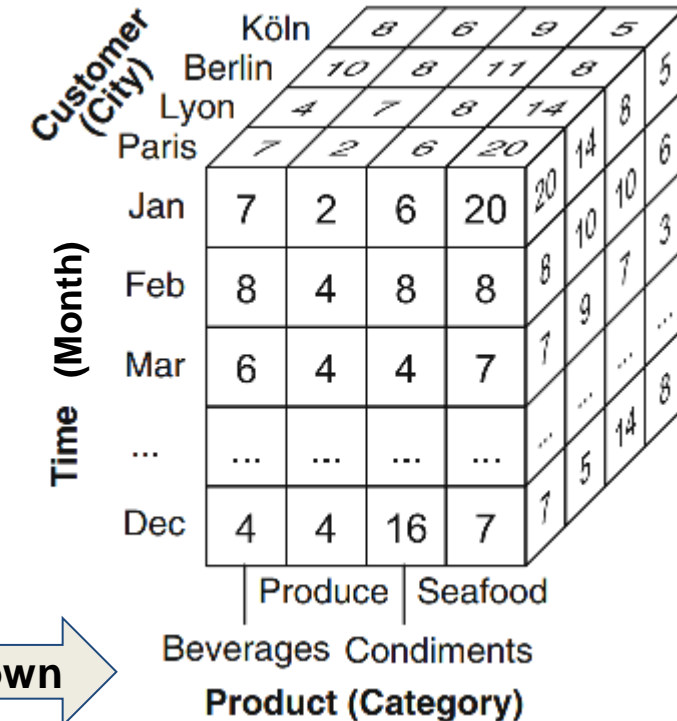


Customer (City): Köln, Berlin, Lyon, Paris

Time (Quarter): Q1, Q2, Q3, Q4

Product (Category): Produce, Seafood, Beverages, Condiments

City	Produce	Seafood	Beverages	Condiments
Köln	24	18	28	14
Berlin	33	25	23	25
Lyon	12	20	24	33
Paris	21	10	18	35
Q1	21	10	18	35
Q2	27	14	11	30
Q3	26	12	35	32
Q4	14	20	47	31



Customer (City): Köln, Berlin, Lyon, Paris

Time (Month): Jan, Feb, Mar, ..., Dec

Product (Category): Produce, Seafood, Beverages, Condiments

City	Produce	Seafood	Beverages	Condiments
Köln	8	6	9	5
Berlin	10	8	11	8
Lyon	4	7	8	14
Paris	7	2	6	20
Jan	7	2	6	20
Feb	8	4	8	8
Mar	6	4	4	7
...	...	...	...	...
Dec	4	4	16	7

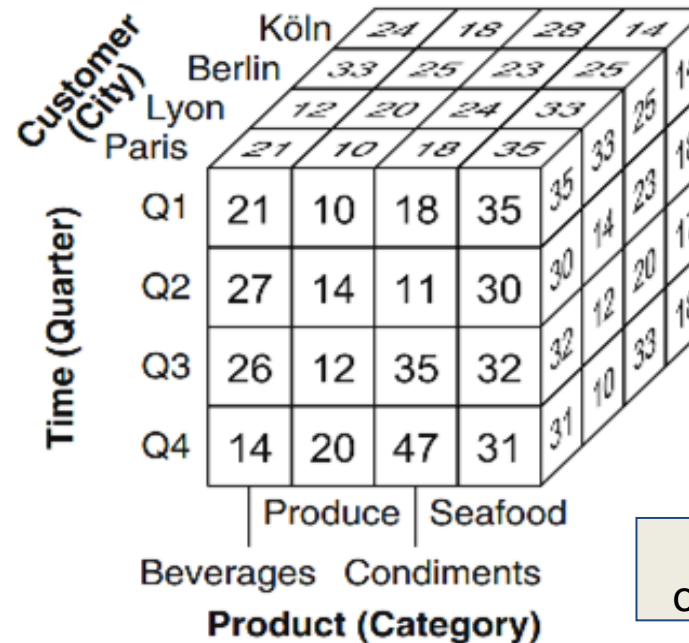
Drill-down

→ **Slice** operation performs to **select one particular dimension** from a given cube and provides a new sub-cube.

**SLICE(CubeName, Dimension, Level = Value)**

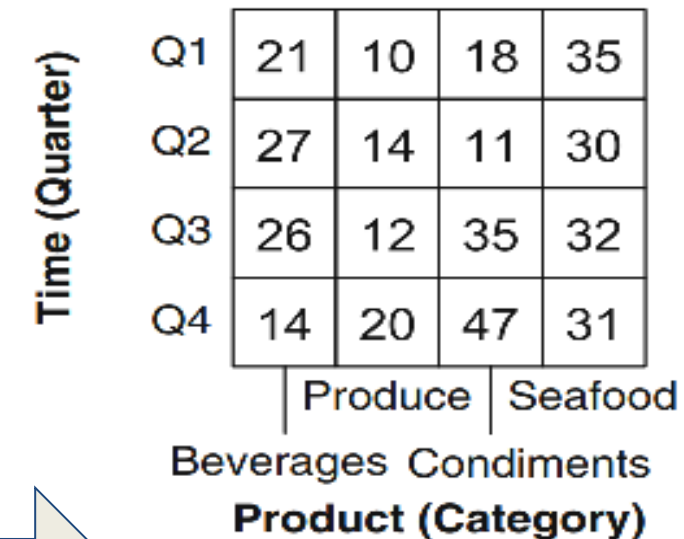
*Example:* A slice operation is applied to visualize the data only for Paris

**SLICE(Sales, Customer, City = 'Paris')**



Time (Quarter)	Customer (City)				Product (Category)			
	Köln	Berlin	Lyon	Paris	Produce	Seafood	Beverages	Condiments
	24	18	28	14	21	10	18	35
	33	25	23	25	27	14	11	30
	12	20	24	33	26	12	35	32
	21	10	18	35	14	20	47	31

**Slice**  
on City= 'Paris'



Time (Quarter)	Produce	Seafood	Beverages	Condiments
Q1	21	10	18	35
Q2	27	14	11	30
Q3	26	12	35	32
Q4	14	20	47	31

→ **Dice** operation performs to **select two and more particular dimension** from a given cube based on a Boolean condition  $\Phi$  and provides a new sub-cube.

**DICE(CubeName,  $\Phi$ )**

*Example:* A dice operation is applied to visualize the data only for sales figures for the first two quarters and for the cities *Lyon* and *Paris*

Time (Quarter)	Customer (City)				Product (Category)			
					Produce		Seafood	
					Beverages	Condiments	Beverages	Condiments
	Q1	Q2	Q3	Q4				
	21	27	26	14	10	14	18	20
	18	11	12	20	35	32	47	31
	35	30	32	31	23	12	10	33
	25	20	33	18	14	33	25	14
	14	17	18	14	28	23	25	14
	24	33	12	21	18	20	10	33
	28	25	33	35	14	33	25	14
	14	14	18	31	28	23	25	14

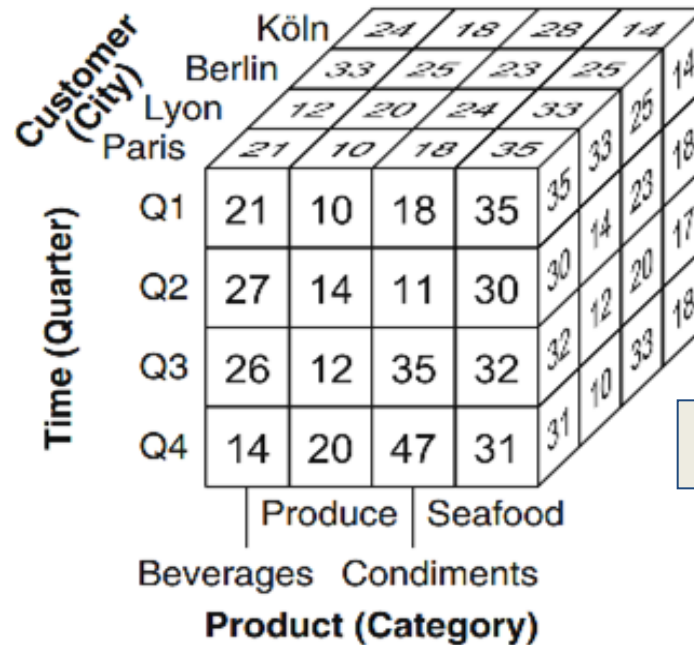
Time (Quarter)	Customer (City)				Product (Category)			
					Produce		Seafood	
					Beverages	Condiments	Beverages	Condiments
Q1	21	27	26	14	10	14	18	20
Q2	18	11	12	20	35	32	47	31

**Dice**  
on City= 'Paris' or 'Lyon'  
and Quarter='Q1' or 'Q2'

- **Pivot** operation performs to rotate the data axes in view in order to provide an alternative presentation of cube to better understand the data contained in it.

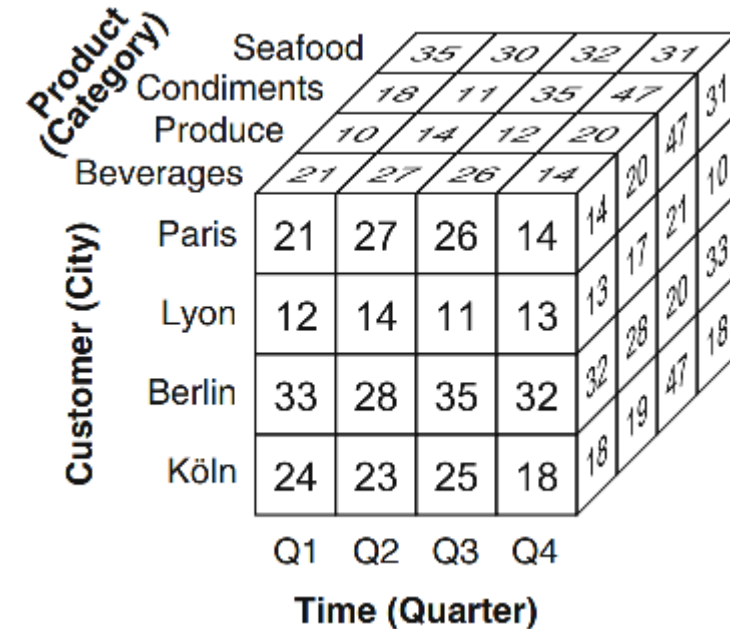
**PIVOT(CubeName, (Dimension → Axis)\*)**

*Example:* A pivot operation is applied to rotate the cube with the *Time* dimension on the *x-axis* without changing granularities.



Time (Quarter)	Customer (City)				Product (Category)			
	Köln	Berlin	Lyon	Paris	Produce	Seafood	Beverages	Condiments
Q1	21	10	18	35	35	14	23	17
Q2	27	14	11	30	30	12	20	18
Q3	26	12	35	32	32	10	33	18
Q4	14	20	47	31	31	10	33	18

Pivot

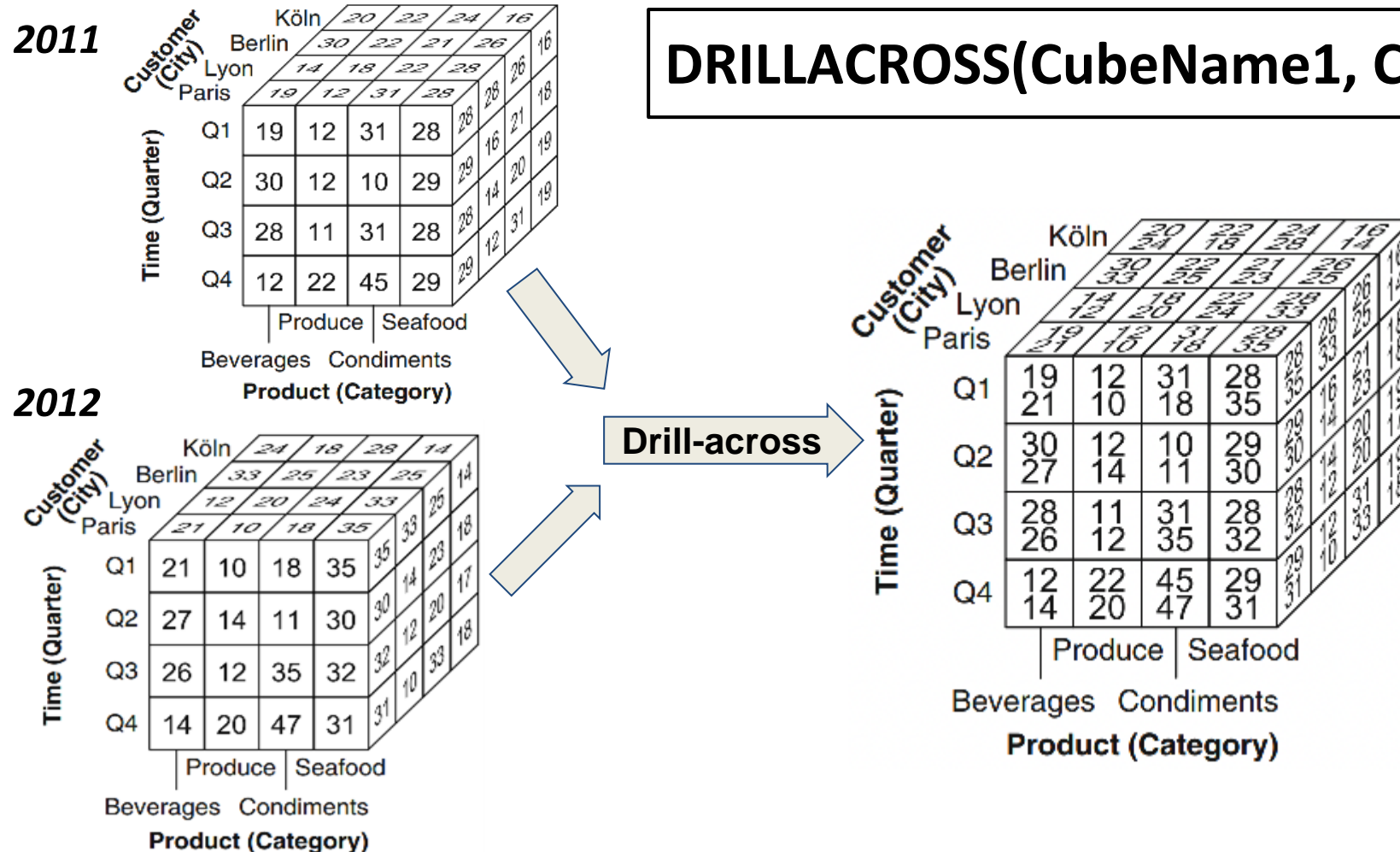


Customer (City)	Product (Category)				Time (Quarter)			
	Seafood	Condiments	Produce	Beverages	Q1	Q2	Q3	Q4
Paris	21	27	26	14	14	17	21	33
Lyon	12	14	11	13	13	28	20	18
Berlin	33	28	35	32	32	19	47	18
Köln	24	23	25	18	18	19	47	18



→ **Drill-across** operation combines cells from two data cubes that have the same schema and instances.

**DRILLACROSS(CubeName1, CubeName2, [Condition])**

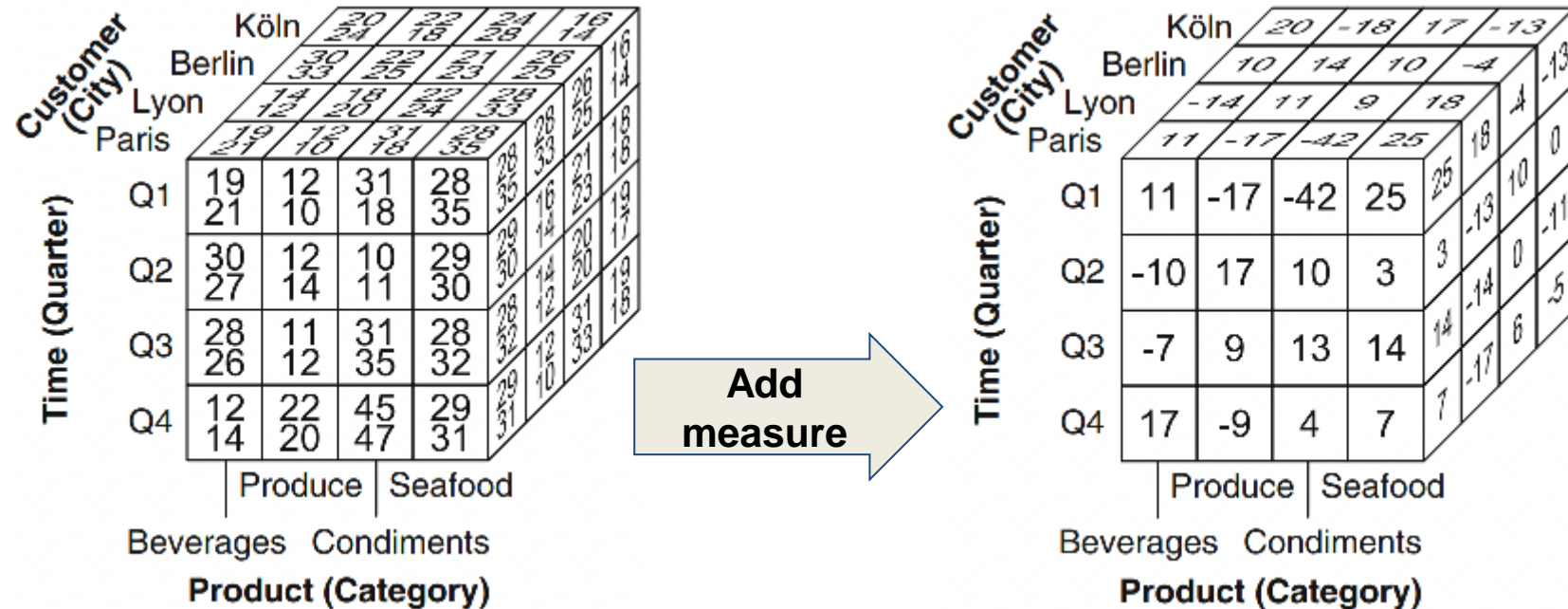


*Example:* A drill-across operation is applied to compare the sales quantities in 2012 with those in 2011.

→ **Add Measure** operation adds new measures to the cube computed from other measures or dimensions.

**ADDMEASURE(CubeName, (NewMeasure = Expression, [AggFct])\* )**

*Example:* An add measure operation is applied to compute the percentage change for each cell from the values in the cube



→ **Aggregation** operations perform to aggregate measures of a cube at the current granularity, that is, without performing a roll-up operation.

**AggFunction(CubeName, Measure) [BY Dimension\*]**

- ◆ SUM
- ◆ MAX/MIN
- ◆ AVG
- ◆ COUNT
- ◆ TOPPERCENT
- ◆ BOTTOM PERCENT
- ◆ RANK
- ◆ DENSERANK

→ *Example:* An Sum operation is applied to compute to total sales by *quarter* and *city*.

Time (Quarter)	Customer (City)	Product (Category)			
		Produce		Seafood	
		Beverages	Condiments		
	Köln	24	18	28	14
	Berlin	33	25	23	25
	Lyon	12	20	24	33
	Paris	21	10	18	35
Q1		21	10	18	35
Q2		27	14	11	30
Q3		26	12	35	32
Q4		14	20	47	31

Sum  
on Quarter and City

Time (Quarter)	Customer (City)			
	Paris	Lyon	Berlin	Köln
Q1	84	89	106	84
Q2	82	77	93	79
Q3	105	72	65	88
Q4	112	61	96	102

→ *Example:* An MAX/MIN operation is applied to measures to obtain the maximum/minimum sales by *quarter* and *city*

Time (Quarter)	Customer (City)	Product (Category)			
		Produce		Seafood	
		Beverages	Condiments		
	Köln	24	18	28	14
	Berlin	33	25	23	25
	Lyon	12	20	24	33
	Paris	21	10	18	35
Q1		21	10	18	35
Q2		27	14	11	30
Q3		26	12	35	32
Q4		14	20	47	31

**Max**  
on Quarter and City

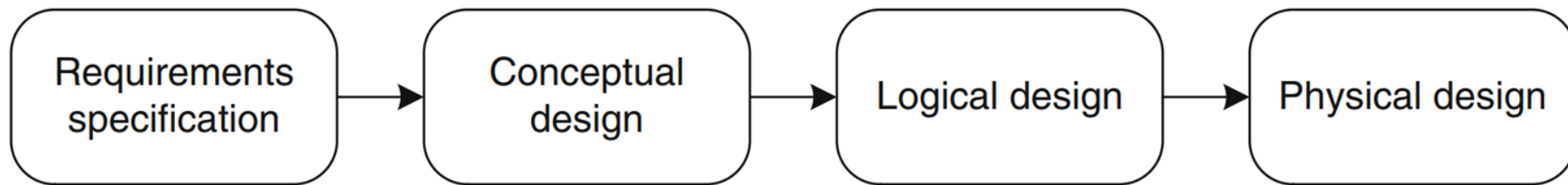
Time (Quarter)	Customer (City)	Product (Category)			
		Produce		Seafood	
		Beverages	Condiments		
	Köln			28	
	Berlin	33			
	Lyon			33	
	Paris			35	
Q1				35	23
Q2				30	
Q3			35		33
Q4			47		



- **Top-down approach:** The requirements of users at different organizational levels are merged before the design process begins, and one schema for the entire data warehouse is built. Then, separate data marts are tailored according to the characteristics of each business area or process.
- **Bottom-up approach:** A separate schema is built for each data mart, taking into account the requirements of the decision-making users responsible for the corresponding specific business area or process. Later, these schemas are merged in a global schema for the entire data warehouse.

- The **choice between the top-down and the bottom-up** approach depends on many factors, such as the *professional skills* of the development team, the *size of the data warehouse*, the users' *motivation* for having a data warehouse, and the *financial support*,...
- **Top down approach**: may be overwhelming for many organizations in terms of cost and duration and also for designers because of its size and complexity.
- **Bottom-up approach**: may deliver a data mart faster and at less cost, allowing users to quickly interact with OLAP tools and create new reports. It can improve the motivation for having a data warehouse. However, it lack of the global framework which can make integration difficult and costly in the long term.

→ **Model-based approach** follows the traditional phases for designing operational databases to define the schema of the overall schema of the organizational data warehouse or the schemas of individual data marts.

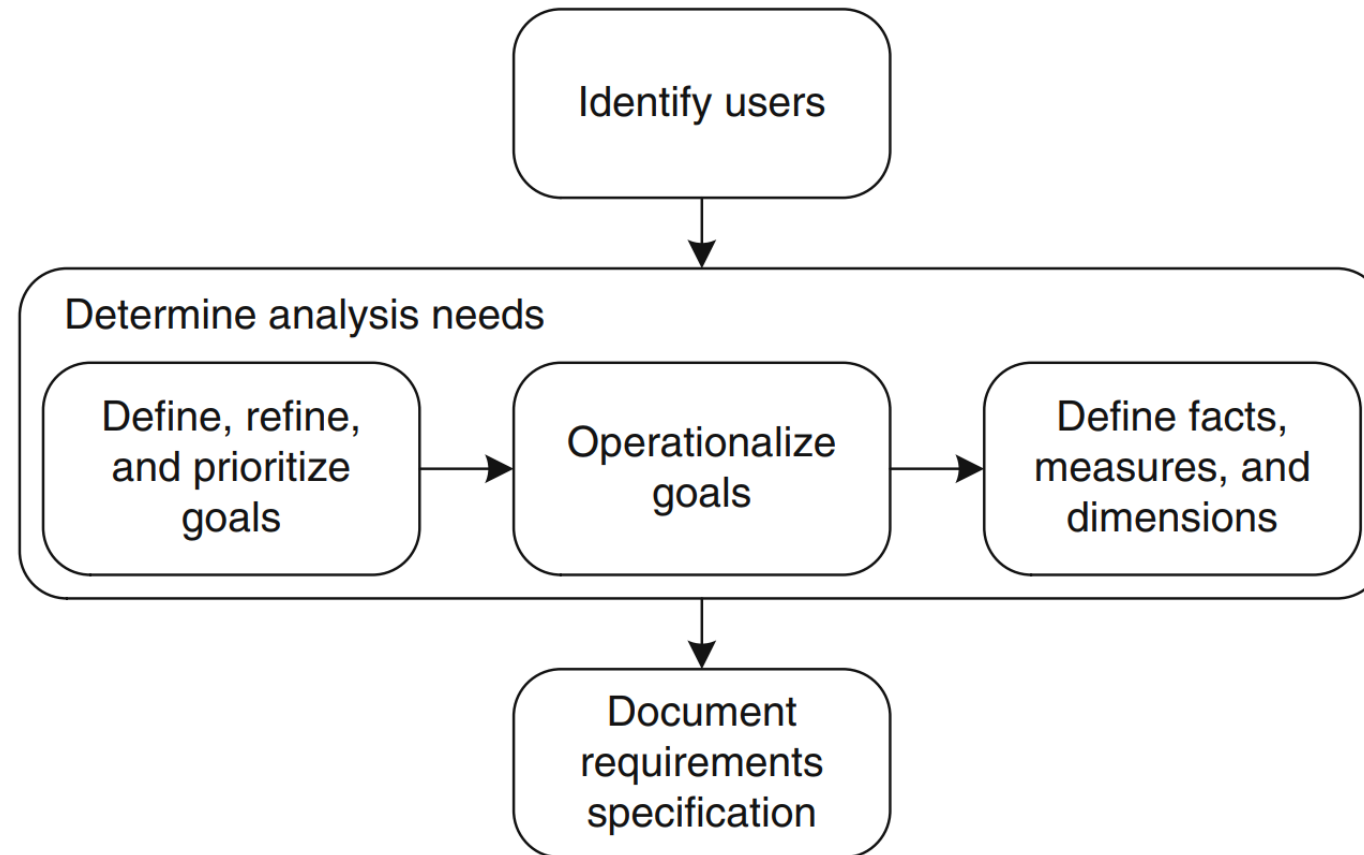


- is one of the earliest steps in system development in which it determines, among other things, which data should be available and how these data should be organized.
- The queries of interest for the users are also determined.
- lead the designer to discover the essential elements of a multidimensional schema, like the facts and their associated dimensions, which are required to facilitate future data manipulation and calculations in the DW development. it has a major impact on the success of data.
- impact on the success of warehouse projects since it directly affects the technical aspects, as well as the data warehouse structures and applications.

- Analysis-driven approach
- Source-driven approach
- Analysis/Source-driven approach



→ is based on whether the analysis goals that requires the identification of key users that can provide useful input about the organizational goals.



- Users at various hierarchical levels in the organization should be considered when analyzing requirements.
  - ◆ **Executive users** at the top organizational level typically require global, summarized information who help in understanding high-level objectives and goals and the overall business vision.
  - ◆ **Management users** may require more detailed information pertaining to a specific area of the organization who provide more insight into the business processes or the tactics used for achieving the business goals
  - ◆ **Professional users** are responsible for a specific section or set of services and may demand specific information related to their area of interest
- The identification of potential users should also consider different entities in a horizontal division of the organization (e.g., departments). This will help in providing an overall view of the project and its scope.

- **Analysis needs** help developers understand what data should be available to respond to the users' expectations on the data warehouse system.
- discover a collection of facts, measures, dimensions, and hierarchies.
- includes several steps:
  - ◆ **Define, Refine, and Prioritize Goals:** considering the business goals (both general and specific), the list of goals are analyzed to detect redundancies and dependencies, the interaction with the users is required to establish the final list of goals.
  - ◆ **Operationalize Goals:** making the goals concrete. A collection of representative queries must be defined through interviews with the users (functional and non functional requirements)
  - ◆ **Define Facts, Measures, and Dimensions:** the analyst tries manually to identify the underlying facts and dimensions from the queries.

- Documentize the obtained information which is the starting point to transfer into technical and business *metadata*.
- Documents can include all elements required by the designers and also a dictionary of the terminology, organizational structure, policies, and constraints of the business,...

*Example:* the document could express in business terms what the candidate measures or dimensions actually represent, who has access to them, and what operations can be done.

- Documents will not be final since additional interactions could be necessary during the conceptual design phase in order to refine or clarify some aspects

Apply the analysis-driven approach to produce a requirements specification for the Northwind data warehouse.

Identify Users:

Three groups of users were identified:

1. Executive: the members of the board of directors of the Northwind company who define the overall company goals.
2. Management: managers at departmental levels, for example, marketing, regional sales, and human resources.
3. Professional: professional personnel who implement the indications of the management. Examples are marketing executive officers.

### Determine Analysis Needs:

The general goal will be addressed: increase the overall company sales by 10% percent yearly. This goal can be decomposed into subgoals:

1. Increase sales in underperforming regions.
2. For customers buying below their potential, increase their orders (in number of orders and individual order amount).
3. Increase sales of products selling below the company expectations.
4. Take action on employees performing below their expected quota.



Determine Analysis Needs:

some examples of the queries that operationalize the mentioned goals

1. Increase sales in underperforming regions:

(a) Five best- and worst-selling (measured as total sales amount) pairs of customer and supplier countries.

(b) Countries, states, and cities whose customers have the highest total sales amount.

(c) Five best- and worst-selling (measured as total sales amount) products by customer country, state, and city

Determine Analysis Needs:

2. For customers buying below their potential, increase their orders (in number of orders and individual order amount):

(a) Monthly sales by customer compared to the corresponding sales (for the same customer) of the previous year.

(b) Total number of orders by customer, time period (e.g., year), and product.

(c) Average unit price per customer.

3. Increase sales of products selling below the company expectations:

(a) Monthly sales for each product category for the current year.

(b) Average discount percentage per product and month.

(c) Average quantity ordered per product.

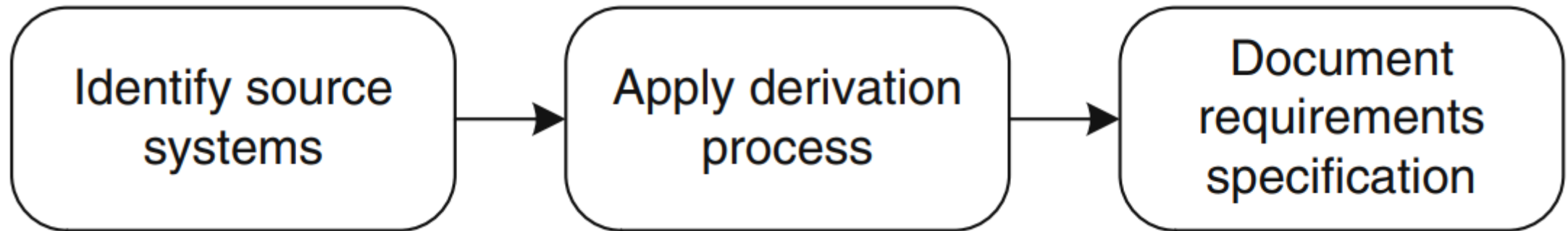
Determine Analysis Needs:

4. Take action on employees performing below their expected quota:
  - (a) Best-selling employee per product per year with respect to sales amount.
  - (b) Average monthly sales by employee and year.
  - (c) Total sales by an employee and her subordinates during a certain time period.

Multidimensional elements of the Northwind case study obtained using analysis-driven approach

Dimensions /measures	Hierarchies and levels	Analysis scenarios											
		1a	1b	1c	2a	2b	2c	3a	3b	3c	4a	4b	4c
Employee	<b>Supervision</b> Subordinate → Supervisor												
	<b>Territories</b> Employee ↔ City → State → Country → Continent	–	–	–	–	–	–	–	–	–	✓	✓	✓
Time	<b>Calendar</b> Day → Month →												
	Quarter → Semester → Year	–	–	–	✓	✓	✓	✓	✓	–	✓	✓	✓
Product	<b>Categories</b> Product → Category												
		–	–	✓	–	✓	–	✓	✓	✓	✓	–	–
Customer	<b>Geography</b> Customer → City →												
	State → Country → Continent	✓	✓	✓	✓	✓	✓	–	–	–	–	–	–
Supplier	<b>Geography</b> Supplier → City →												
	State → Country → Continent	✓	–	–	–	–	–	–	–	–	–	–	–
Quantity	–	–	–	–	–	–	–	–	–	✓	–	–	–
Discount	–	–	–	–	–	–	–	–	✓	–	–	–	–
SalesAmount	–	✓	✓	✓	✓	–	–	✓	–	–	✓	✓	✓
UnitPrice	–	–	–	–	–	–	✓	–	–	–	–	–	–

- The source-driven approach is based on the data available at the source systems aiming at identifying all multidimensional schemas that can be implemented starting from the available operational databases.
- These databases are analyzed exhaustively in order to discover the elements that can represent facts with associated dimensions, hierarchies, and measures leading to an initial data warehouse schema.



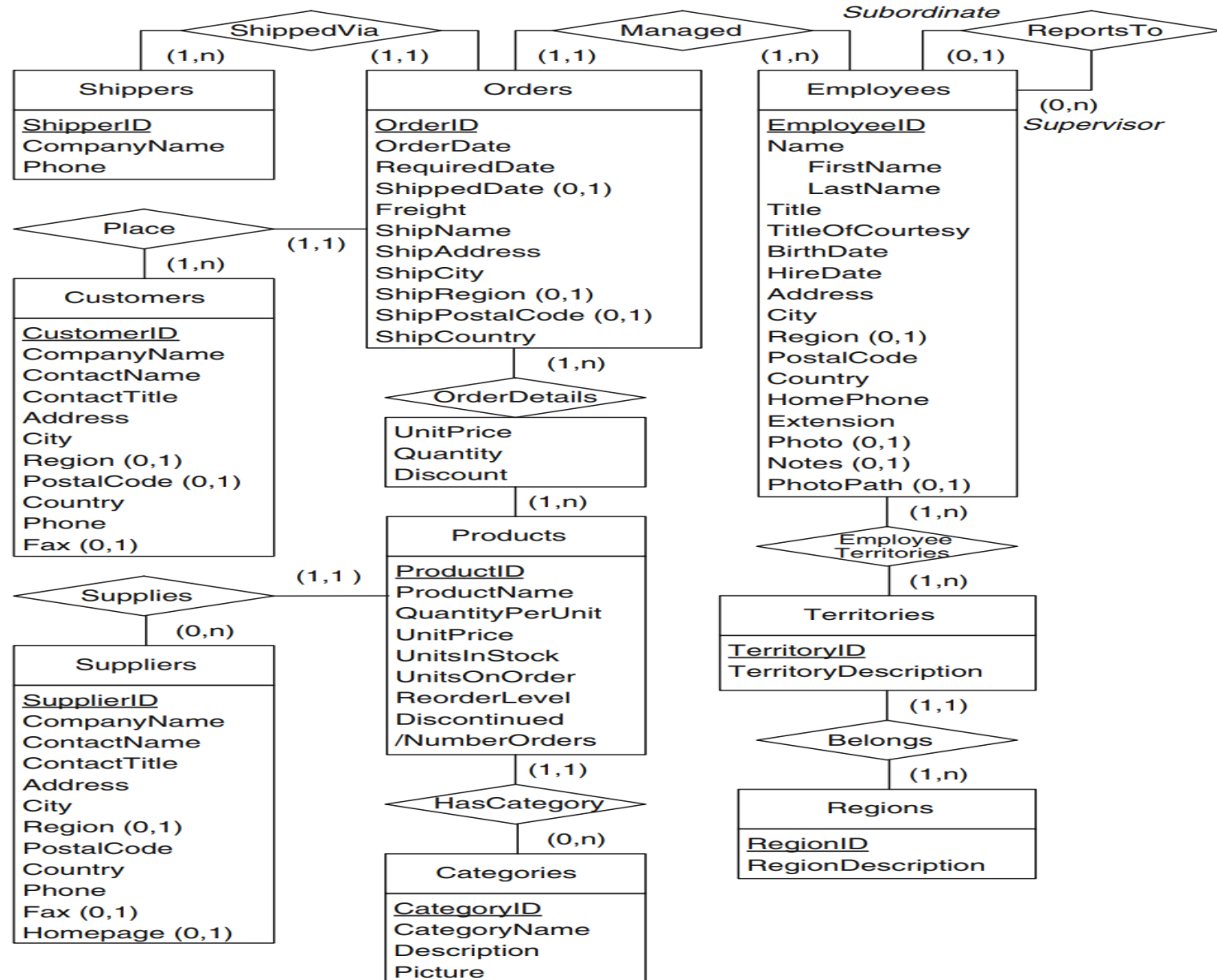
- The aim of this step is to determine the existing operational systems that can be data providers for the warehouse and to assess their quality.
- External sources are not considered at this stage but they can be included later on when the need for additional information has been identified
- It relies on system documentation, preferably represented using the entity-relationship model or relational tables but it may be difficult to obtain, e.g., when the data sources include implicit structures that are not declared through the data definition language of the database,
- Reverse engineering processes can be applied to rebuild the logical and conceptual schemas of source systems whose documentation is missing or outdated.



- require that the operational databases.
- are represented using either the entity-relationship or the relational model.
- External sources are not considered at this stage; they can be included later on when the need for additional information has been identified
- It relies on system documentation, preferably represented using the entity-relationship model or relational tables but it may be difficult to obtain, e.g., when the data sources include implicit structures that are not declared through the data definition language of the database,
- Reverse engineering processes can be applied to rebuild the logical and conceptual schemas of source systems whose documentation is missing or outdated.

- Like in the analysis-driven approach, the requirements specification phase should be documented
- The documents describe those elements of the source systems that can be considered as facts, measures, dimensions, and hierarchies contained in the technical metadata.
- The document involve at this stage a domain expert who can help in defining business terminology for these elements and in indicating, for example, whether measures are additive, semiadditive, or nonadditive.

Given entity-relationship schema of the operational database in the *Northwind* case study



## Apply Derivation Process

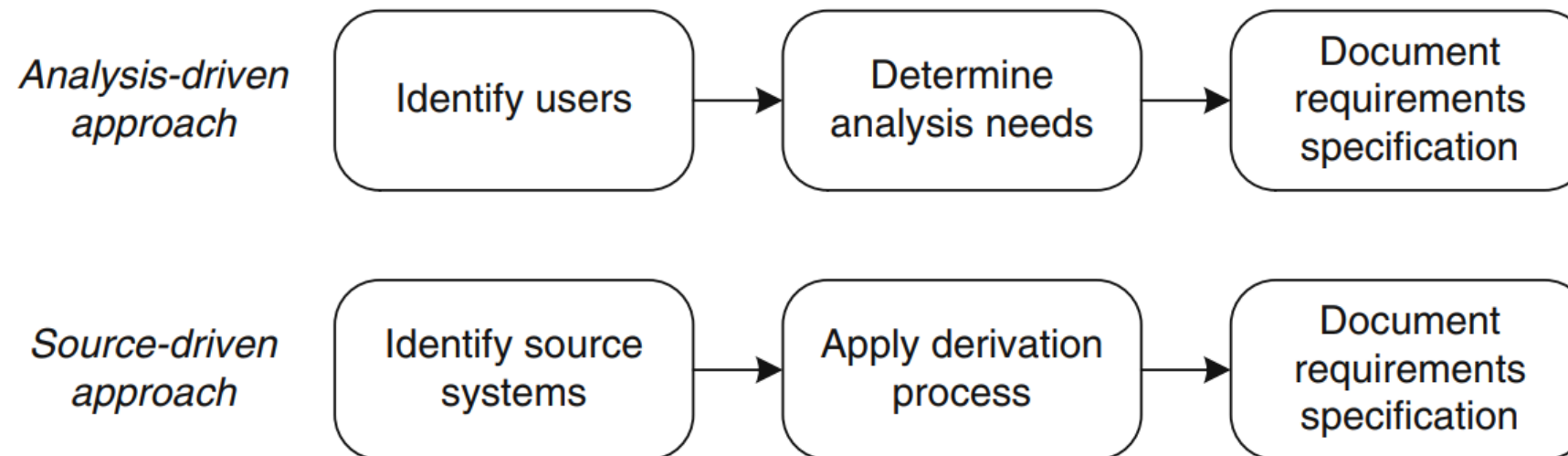
Multidimensional elements  
in the *Northwind* case  
study obtained using the  
source-driven approach

Facts	Measures	Dimensions and cardinalities		Hierarchies and levels
Sales	UnitPrice Quantity Discount	Product	1:n	<b>Categories</b> Product → Category
		Supplier	1:n	<b>Geography</b> Supplier → City → State → Region → Country
		Customer	1:n	<b>Geography</b> Supplier → City → State → Region → Country
		Employee	1:n	<b>Supervision</b> Subordinate → Supervisor <b>Territories</b> Employee ↔ City → State → Region → Country
		OrderDate	1:n	<b>Calendar</b> Date → Month → Quarter → Semester → Year
		DueDate	1:n	<b>Calendar</b> (as above)
		ShippedDate	1:n	<b>Calendar</b> (as above)
		Order	1:1	

### Document requirement specification

- all information specified in the previous steps is documented
- includes a detailed description of the source schemas that serve as a basis for identifying the elements in the multidimensional schema
- also contain elements in the source schema for which it is not clear whether they can be used as attributes or hierarchies in a dimension
- consider that the address of employees will not be used as a hierarchy.
- If the source schemas use attributes or relation names with unclear semantics, the corresponding elements of the multidimensional schema must be renamed, specifying clearly the correspondences between the old and new names.

- It is a combination of the analysis and source-driven approaches, which can be used in parallel to achieve an optimal design
- It takes into account two types of activities including what are the analysis needs from the users and what the source systems can provide involved in creating a multidimensional schema from operational databases.
- Each type of activity results in the identification of elements for the initial multidimensional schema.





- **Microsoft's SQL Server tools**
- **Pentaho Business Analytics**

- It provides an integrated platform for building analytical applications including three main components: **Analysis Services, Integration Services and Reporting Services.**
- SQL Server provides two tools for developing and managing these components:
  - ◆ SQL Server Data Tools (SSDT) is a development platform integrated with Microsoft Visual Studio supporting Analysis Services, Reporting Services, and Integration Services projects.
  - ◆ SQL Server Management Studio (SSMS) provides integrated management of all SQL Server components.
- The underlying model across these tools is called the Business Intelligence Semantic Model (BISM) which comes in two modes, the multidimensional and tabular modes.

- **Analysis Services** is an OLAP tool that provides analytical and data mining capabilities. It is used to define, query, update, and manage OLAP database.
- The MDX (MultiDimensional eXpressions) language is used to retrieve data.
- Users may work with OLAP data via client tools (Excel or other OLAP clients) that interact with Analysis Services' server component.
- Analysis Services provides several data mining algorithms and uses the DMX (Data Mining eXtensions) language for creating and querying data mining models and obtaining predictions

- **Integration Services** supports ETL processes, which are used for loading and refreshing data warehouses on a periodic basis.
- It is used to extract data from a variety of data sources; to combine, clean, and summarize this data; and, finally, to populate a data warehouse with the resulting data.

- **Reporting Services** is used to define, generate, store, and manage reports.
- Reports can be built from various types of data sources, including data warehouses and OLAP cubes.
- Reports can be personalized and delivered in a variety of formats.
- Users can view reports with a variety of clients, such as web browsers or other reporting clients
- Clients access reports via Reporting Services' server component

**Pentaho Business Analytics** is a suite of business intelligence products with two versions: an enterprise edition that is commercial and a community edition that is open source.

The main components include:

- Pentaho Business Analytics Platform
- Pentaho Analysis Services
- Pentaho Data Mining
- Pentaho Data Integration
- Pentaho Report Designer

Several design tools are provided:

- Pentaho Schema Workbench
- Pentaho Aggregation Designer
- Pentaho Metadata Editor



**Nhân bản – Phụng sự – Khai phóng**

**Enjoy the Course...!**