Chapter Name

Data Warehousing and Business Intelligence Solutions

Traditional Databases and Data Analysis

- The importance of data analysis has been steadily increasing from the early 1990s
  - Organizations in all sectors are being required to improve their decision-making processes in order to maintain their competitive advantage.
- Traditional database systems do not satisfy the requirements of data analysis.
  - They are designed and tuned to support the daily operations of an organization, and their primary concern is to ensure fast, concurrent access to data.
  - This requires transaction processing and concurrency control capabilities, as well as recovery techniques that guarantee data consistency.
  - These systems are known as operational databases or online transaction processing (OLTP) systems.
  - The OLTP paradigm is focused on transactions.

OLAP – A New Paradigm Oriented to Analyze Data

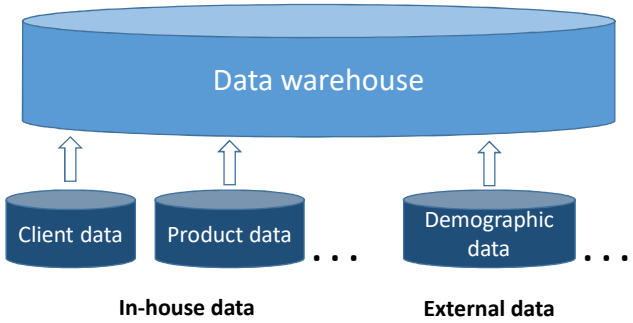
- Since OLTP systems must support heavy transaction loads, their design should prevent update anomalies, and thus, OLTP databases are highly normalized.
  - Thus, they perform poorly when executing complex queries that need to join many relational tables together or to aggregate large volumes of data.
- Besides, typical operational databases contain detailed data and do not include historical data.
- The above needs called for a new paradigm specifically oriented to analyze the data in organizational databases to support decision making.
  - This paradigm is called online analytical processing (OLAP).
  - OLAP is focused on queries, in particular, analytical queries.
  - OLAP-oriented databases should support a heavy query load.

The Need for Data Warehouses

- Typical OLAP queries would ask, for example, for the total sales amount by product and by customer or for the most ordered products by customer.
  - These kinds of queries involve aggregation, and thus, processing them will require, most of the time, traversing all the records in a database table.
  - Indexing techniques aimed at OLTP are not efficient in this case: new indexing and query optimization techniques are required for OLAP.
  - It is easy to see that normalization is not good for these queries, since it partitions the database into many tables. Reconstructing the data would require a high number of joins.
- Therefore, the need for a different database model to support OLAP led to the notion of data warehouses, which
  - are (usually) large repositories that consolidate data from different sources (internal and external to the organization), and
  - follow the multidimensional data model.

The Concept of Data Warehousing

- The concept of data warehousing has evolved out of the need for easy access to a structured store of quality data that can be used for decision making.
  - Organizations have vast amounts of data but have found it increasingly difficult to access it and make use of it.
- A data warehouse is a database for the specific purpose of data analysis.
  - Data warehouses are constructed with use of database technology.
  - Data could be obtained from several sources.

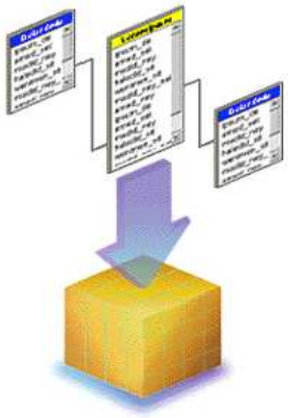


Data warehouses vs. operational databases:

1. Contain more data
2. Data is only for reading
3. Time is particularly respected.

A Database Model to Support OLAP

- Being dedicated analysis databases, data warehouses can be designed and optimized to efficiently support OLAP queries.
  - In addition, data warehouses are also used to support other kinds of analysis tasks, like reporting, data mining, and statistical analysis.
- OLAP is a method of organizing the large data stores, where data are well suited for analysis.
  - Data stored in relational databases are turned into meaningful, easy to navigate business information by creating a data cube.
- Data warehouses and OLAP systems are based on the multidimensional model, which
  - views data in an *n*-dimensional space, usually called a data cube or a hypercube.
- A data cube is defined by dimensions and facts.

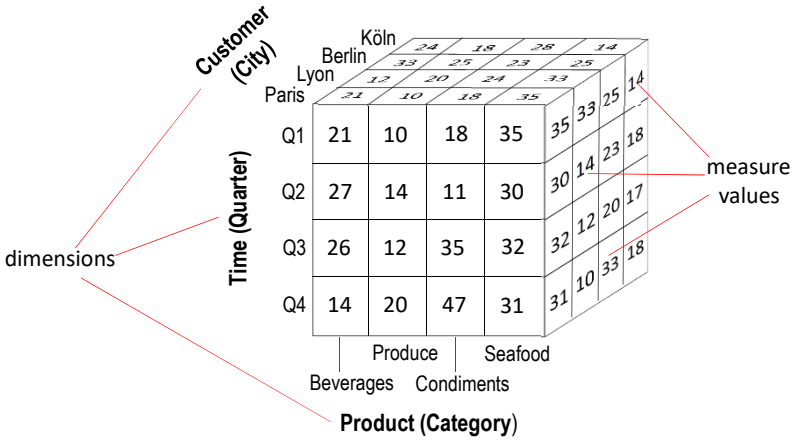


2024-03-18

Data Cube

☞ Let us consider a data cube, which can be used to analyze sales figures. Dimensions correspond to the various business perspectives and cube cells contain the measures to be analyzed.

- The picture shows 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.
- In the example, sales figures are aggregated to the levels `Category`, `Quarter`, and `City`, respectively.



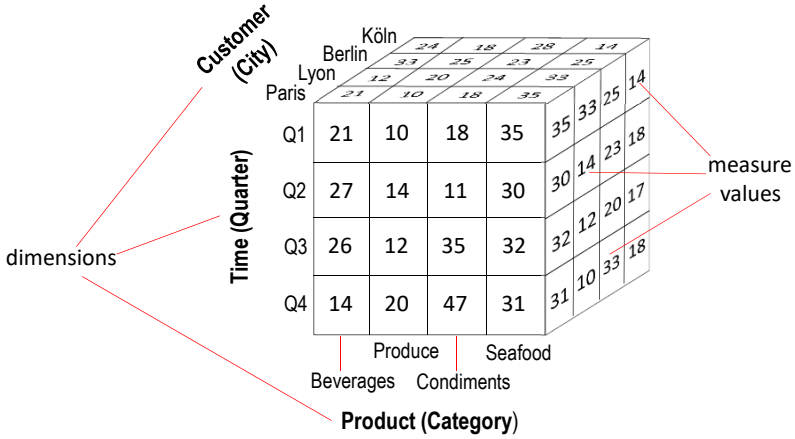
Members and Attributes

☞ Instances of a dimension are called members.

- For example, `Seafood` and `Beverages` are members of the `Product` dimension at the `Category` level.

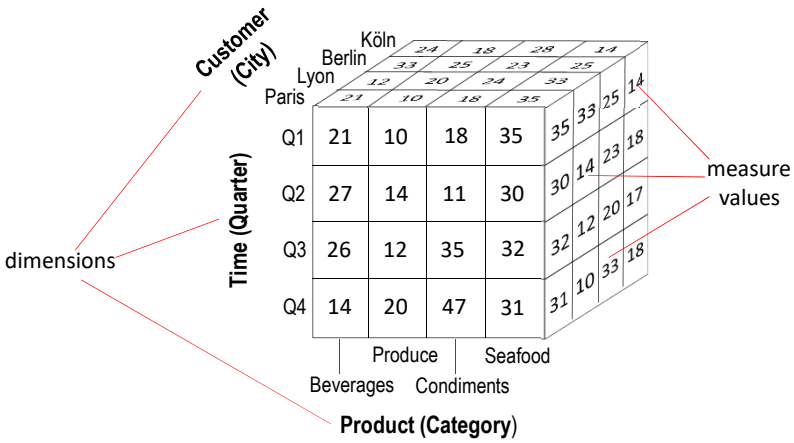
☞ Dimensions also have associated attributes describing them.

- For example, the `Product` dimension could contain attributes such as ProductNumber and UnitPrice, which are not shown in the figure.



Facts and Measures

- The cells of a data cube, or facts, have associated numeric values called measures.
  - These measures are used to evaluate quantitatively various aspects of the analysis at hand.
  - For example, each number shown in a cell of the data cube in our figure represents a measure `Quantity`, indicating the number of units sold (in thousands) by category, quarter, and customer's city.
- A data cube typically contains several measures.
  - For example, another measure, not shown in the figure, could be `Amount`, indicating the total sales amount.



Hierarchies (1/3)

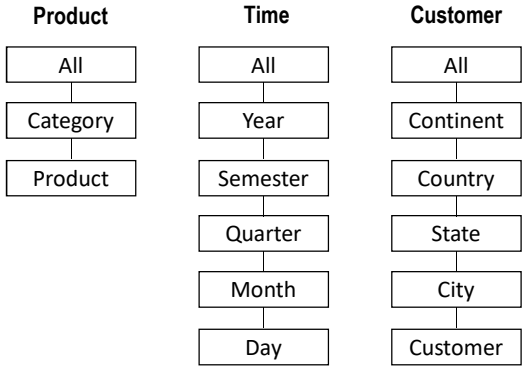
- The granularity of a data cube is determined by the combination of the levels corresponding to each axis of the cube.
  - In our example, the dimension levels are indicated between parentheses: `Category` for the `Product` dimension, `Quarter` for the `Time` dimension, and `City` for the `Customer` dimension.
- In order to extract strategic knowledge from a cube, it is necessary to view its data at several levels of detail.
  - In the current example, an analyst may want to see the sales figures at a finer granularity, such as at the month level, or at a coarser granularity, such as at the customer's country level.
  - Hierarchies allow this possibility by defining a sequence of mappings relating lower-level, detailed concepts to higher-level, more general concepts.
- Given two related levels in a hierarchy, 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, while a dimension instance comprises the members at all levels in a dimension.

Hierarchies (2/3)

☞ The figure shows the simplified hierarchies for our cube example.

- In the `Product` dimension, products are grouped in categories.
- For the `Time` dimension, the lowest granularity is `Day`, which aggregates into `Month`, which in turn aggregates into `Quarter`, `Semester`, and `Year`.
- Similarly, for the `Customer` dimension, the lowest granularity is `Customer`, which aggregates into `City`, `State`, `Country`, and `Continent`.

☞ It is usual to represent the top of the hierarchy with a distinguished level called `All`.



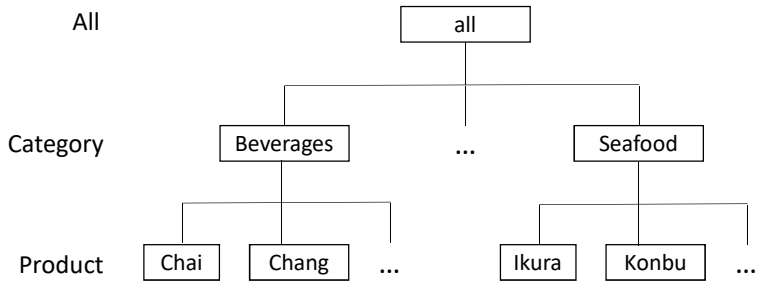
Hierarchies (3/3)

☞ At the instance level, the figure shows an example of the `Product` dimension.

- Each product at the lowest level of the hierarchy can be mapped to a corresponding category.

☞ All categories are grouped under a member called `all`, which is the only member of the distinguished level `All`.

- This member is used for obtaining the aggregation of measures for the whole hierarchy, that is, for obtaining the total sales for all products.



Aggregating Measures

- 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.
  - For example, if we use the `Customer` hierarchy 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, for example, the `SUM` operation.
  - Similarly, total sales figures will result in a cube containing one cell with the total sum of the quantities of all products, that is, this corresponds to visualizing the cube at the `All` level of all dimension hierarchies.

Summarizability (1/2)

- Summarizability refers to the correct aggregation of cube measures along dimension hierarchies, in order to obtain consistent aggregation results.
  - To ensure summarizability, a set of conditions may hold.
- Disjointness of instances
  - The grouping of instances in a level with respect to their parent in the next level must result in disjoint subsets.
  - For example, in the hierarchy of our example, a product cannot belong to two categories. If this were the case, each product sales would be counted twice, one for each category.

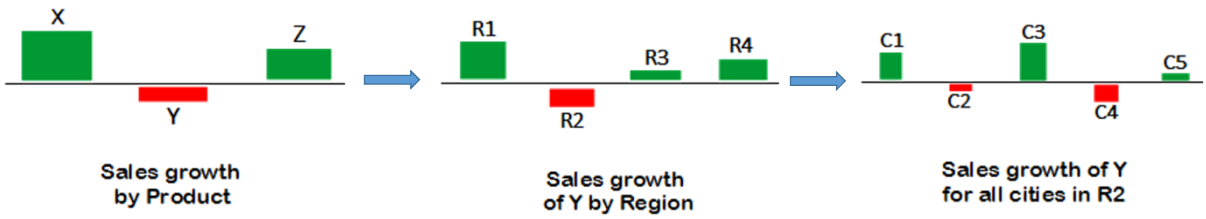
Summarizability (2/2)

- Completeness
  - All instances must be included in the hierarchy and each instance must be related to one parent in the next level.
  - For example, the instances of the `Time` hierarchy must contain all days in the period of interest, and each day must be assigned to a month.
  - If this condition were not satisfied, the aggregation of the results would be incorrect, since there would be dates for which sales will not be counted.
- Correctness
  - It refers to the correct use of the aggregation functions.
  - As explained next, measures can be of various types, and this determines the kind of aggregation function that can be applied to them.

Additive, Semiadditive and Nonadditive Measures

- According to the way in which they can be aggregated, measures can be classified as follows.
- Additive measures can be meaningfully summarized along all the dimensions, using addition.
  - These are the most common type of measures.
  - For example, the measure `Quantity` in our cube is additive: it can be summarized when the hierarchies in the `Product`, `Time`, and `Customer` dimensions are traversed.
- Semiadditive measures can be meaningfully summarized using addition along some, but not all, dimensions.
  - A typical example is that of 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.
  - Typical examples are item price, cost per unit, and exchange rate.

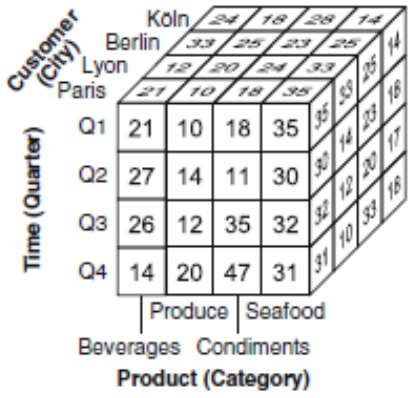
Using OLAP Application (To Remember)



Using an OLAP tool the analyst can browse his data on multiple levels and dimensions in order to explain system's behavior.

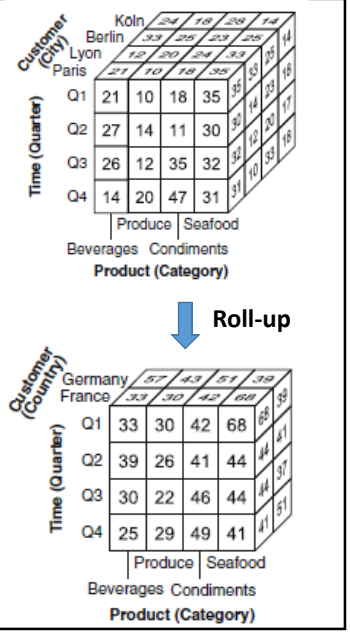
Illustrative Examples of OLAP Operations

- A fundamental characteristic of the multidimensional model is that it allows one to view data from multiple perspectives and at several levels of detail.
  - The OLAP operations allow these perspectives and levels of detail to be materialized by exploiting the dimensions and their hierarchies, thus providing an interactive data analysis environment.
- On the following slides we will consider a possible scenario that shows how an end user can operate over a data cube in order to analyze data in different ways.
- Our user starts from a cube containing quarterly sales quantities (in thousands) by product categories and customer cities for the year 2012.



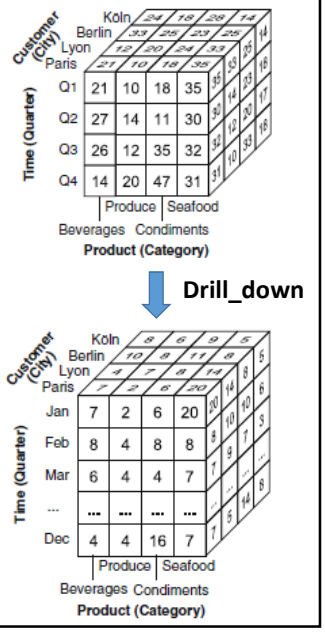
Roll-up to the Country Level

- The user first wants to compute the sales quantities by country.
  - For this, she applies a roll-up operation to the Country level along the Customer dimension. The result is shown in the figure on the right.
- While the original cube contained four values in the Customer dimension, one for each city, the new cube contains two values, each one corresponding to one country.
  - The remaining dimensions are not affected.
- Thus, the values in cells pertaining to Paris and Lyon in a given quarter and category contribute to the aggregation of the corresponding values for France.
  - The computation of the cells pertaining to Germany proceeds analogously.



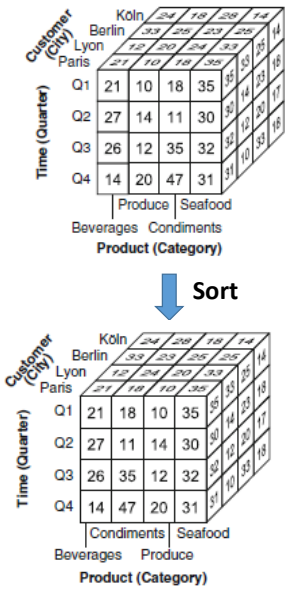
Drill-down to the Month Level

- Our user then notices that sales of the category Seafood in France are significantly higher in the first quarter compared to the other ones.
- Thus, she first takes the cube back to the City aggregation level and then applies a drill-down along the Time dimension to the Month level to find out whether this high value occurred during a particular month. The result is shown in the figure.
- In this way, she discovers that, for some reason yet unknown, sales in January soared both in Paris and in Lyon.



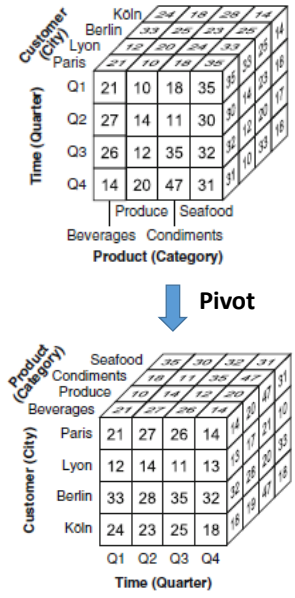
Sort `Product` by Name

- Our user now wants to explore alternative visualizations of the cube to better understand the data contained in it.
  - For this, she sorts the products by name.



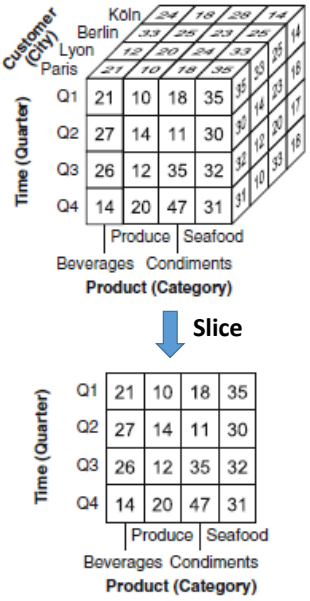
Pivot

- Then, she wants to see the cube with the `Time` dimension on the x-axis.
- Therefore, she takes the original cube and rotates the axes of the cube without changing granularities.
- This is called pivoting.



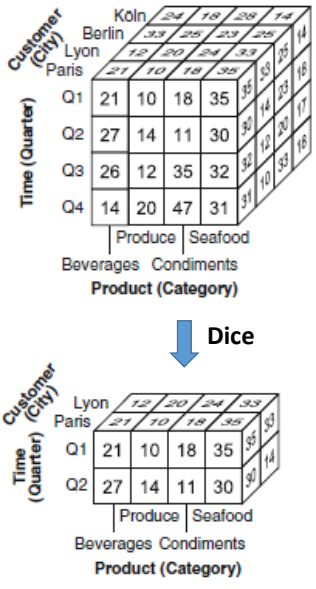
Slice on `City = 'Paris'`

- Continuing her browsing of the original cube, the user then wants to visualize the data only for `Paris`.
- For this, she applies a slice operation that results in the data view depicted in the bottom of the figure.
  - Here, she obtained a two dimensional matrix, where each column represents the evolution of the sales quantity by category and quarter, that is, a collection of time series.



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

- As her next operation, our user goes back to the original cube and builds a subcube, with the same dimensionality, but only containing sales figures for the first two quarters and for the cities `Lyon` and `Paris`.
- This is done with a dice operation.



Data Warehouse Characteristics (1/3)

- A data warehouse is a repository of integrated data obtained from several sources for the specific purpose of multidimensional data analysis.
- More technically, a data warehouse is defined as a collection of subject-oriented, integrated, nonvolatile, and time-varying data to support management decisions.

- Subject oriented means that data warehouses focus on the analytical needs of different areas of an organization.
  - These areas vary depending on the kind of activities performed by the organization.
  - For example, in the case of a retail company, the analysis may focus on product sales or inventory management.
  - In operational databases, on the contrary, the focus is on specific functions that applications must perform, for example, registering sales of products or inventory replenishment.

Data Warehouse Characteristics (2/3)

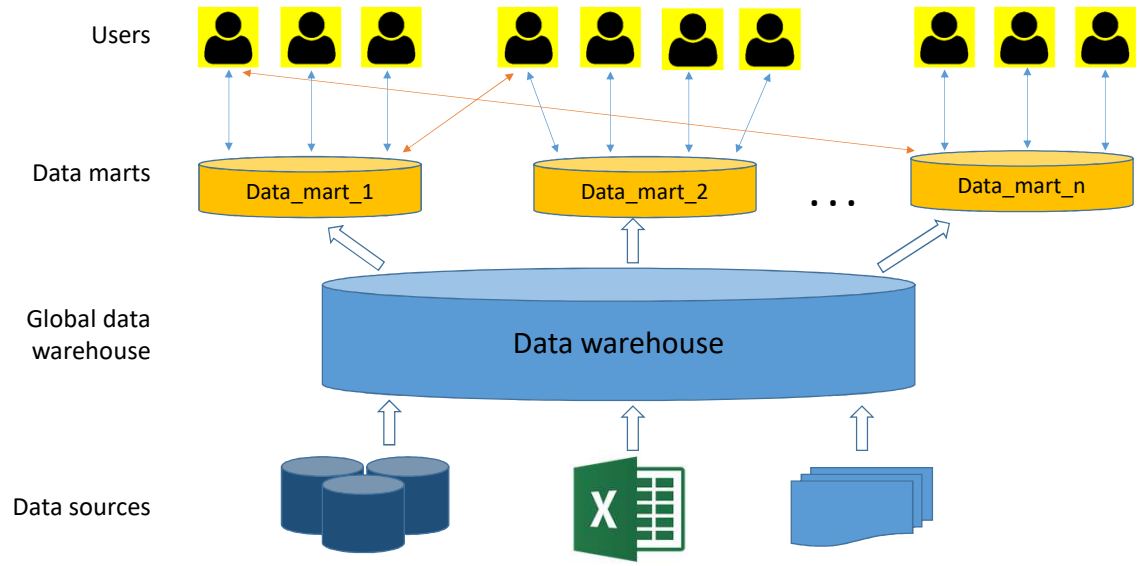
- Integrated means that data obtained from several operational and external systems must be joined together,
  - which implies solving problems due to differences in data definition and content, such as differences in data format and data codification, synonyms (fields with different names but the same data), homonyms (fields with the same name but different meanings), multiplicity of occurrences of data, and many others.
  - In operational databases these problems are typically solved in the design phase.
- Nonvolatile means that durability of data is ensured by disallowing data modification and removal
  - This expands the scope of the data to a longer period of time than operational systems usually offer.
  - A data warehouse gathers data encompassing several years, typically 5–10 years or beyond, while data in operational databases is often kept for only a short period of time, for example, from 2 to 6 months, as required for daily operations, and it may be overwritten when necessary.

Data Warehouse Characteristics (3/3)

☞ Time varying indicates the possibility of retaining different values for the same information, as well as the time when changes to these values occurred.

- For example, a data warehouse in a bank might store information about the average monthly balance of clients' accounts for a period covering several years.
- In contrast, an operational database may not have explicit temporal support, since sometimes it is not necessary for day-to-day operations and it is also difficult to implement.

Data Warehouse and Data Mart



Comparison Between Operational Databases and Data Warehouses

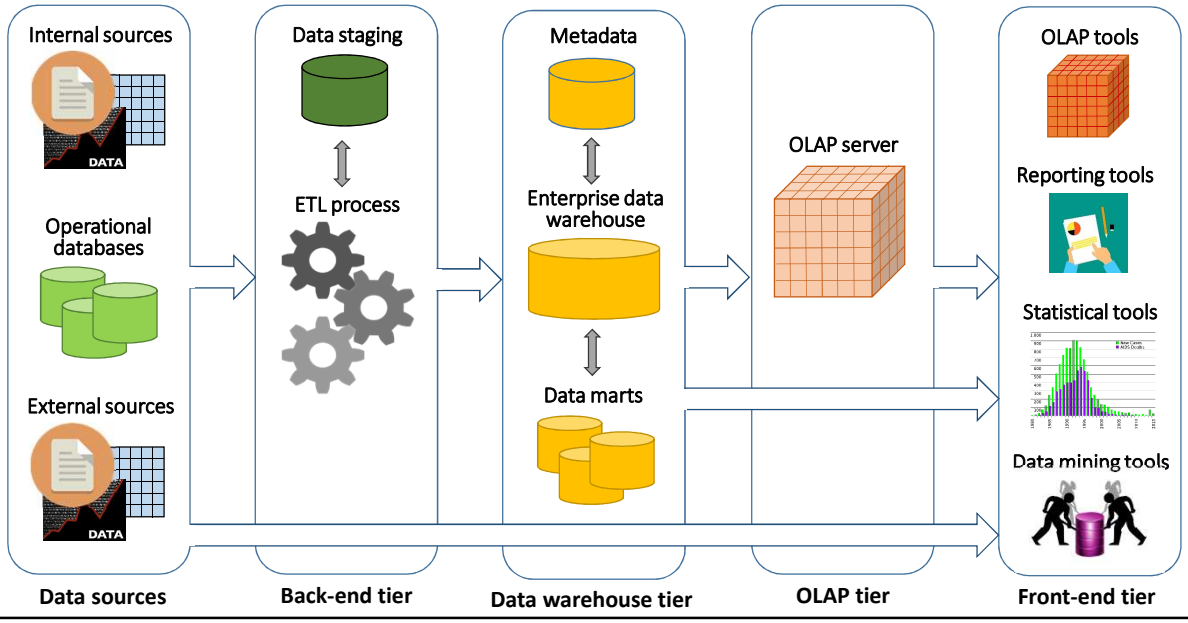
	Aspect	Operational databases	Data Warehouses
1	**User type**	Operators, office employees	Managers, executives
2	**Usage**	Predictable, repetitive	Ad hoc, nonstructured
3	**Data content**	Current, detailed data	Historical, summarized data
4	**Data organization**	According to operational needs	According to analysis needs
5	**Data structures**	Optimized for small transactions	Optimized for complex queries
6	**Access frequency**	High	From medium to low
7	**Access type**	Read, insert, update, delete	Read, append only
8	**Number of records per access**	Few	Many
9	**Response time**	Short	Can be long
10	**Concurrency level**	High	Low
11	**Lock utilization**	Needed	Not Needed
12	**Update frequency**	High	None
13	**Data redundancy**	Low (normalized tables)	High (denormalized tables)
14	**Data modeling**	UML, ER model	Multidimensional model

Data Warehouse Architecture

☞ A general data warehouse architecture consists of several tiers:

- The back-end tier is composed of extraction, transformation, and loading (ETL) tools, used to feed data into the data warehouse from operational databases and other data sources, which can be internal or external from the organization, and a data staging area, which is an intermediate database where all the data integration and transformation processes are run prior to the loading of the data into the data warehouse.
- The data warehouse tier is composed of an enterprise data warehouse and/or several data marts and a metadata repository storing information about the data warehouse and its contents.
- The OLAP tier is composed of an OLAP server, which provides a multidimensional view of the data, regardless of the actual way in which data are stored in the underlying system.
- The front-end tier is used for data analysis and visualization. It contains client tools such as OLAP tools, reporting tools, statistical tools, and data mining tools.

Typical Data Warehouse Architecture

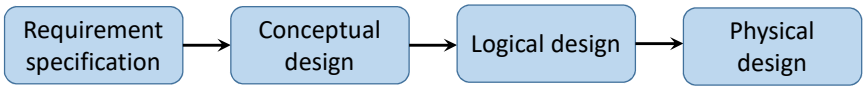


Data Warehouse Design

- Like in operational databases, there are two major methods for the design of data warehouses and data marts.
- In the top-down approach,
  - the requirements of users at different organizational levels are merged before the design process starts, and
  - one schema for the entire data warehouse is built, from which data marts can be obtained.
- In the bottom-up approach,
  - a schema is built for each data mart, according to the requirements of the users of each business area.
  - The data mart schemas produced are then merged in a global warehouse schema.

Phases in Data Warehouse Design

- One possible strategy is to use a model-based approach for data warehouse design, which follows the traditional phases for designing operational databases, as shown in the figure:



- In fact there are important differences between the design phases for databases and data warehouses, arising from their different nature.
- Although, for simplicity, the phases in our figure are depicted consecutively, actually there are multiple interactions between them.
- The phases in figure may be applied to define either the schema of the overall schema of the organizational data warehouse or the schemas of individual data marts.

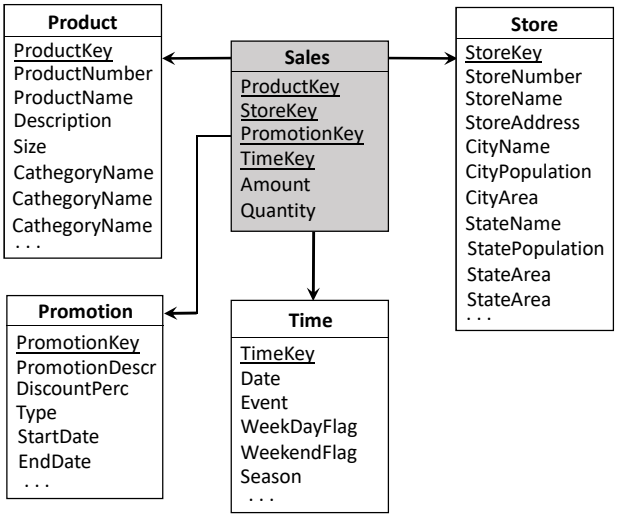
Conceptual Modeling of Data Warehouses

The conventional database design process includes the creation of database schemas at three different levels: conceptual, logical, and physical.

- A conceptual schema is a concise description of the users' data requirements without taking into account implementation details.
- The advantages of using conceptual models for designing databases are well known.
  - Conceptual models facilitate communication between users and designers since they do not require knowledge about specific features of the underlying implementation platform.
  - Schemas developed using conceptual models can be mapped to various logical models, such as relational, object-relational, or object-oriented models, thus simplifying responses to changes in the technology used.
  - Conceptual models facilitate the database maintenance and evolution, since they focus on users' requirements; as a consequence, they provide better support for subsequent changes in the logical and physical schemas.

Logical Modeling of Data Warehouses

- One possible logical (relational) representation of the multidimensional model is based on the star schema, where there is one central fact table, and a set of dimension tables, one for each dimension.
  - An example is given in figure, where the fact table is depicted in gray and the dimension tables are depicted in white.
  - The fact table contains the foreign keys of the related dimension tables, namely, ProductKey, StoreKey, PromotionKey, and TimeKey, and the measures, namely, Amount and Quantity.
  - As shown in the figure, referential integrity constraints are specified between the fact table and each of the dimension tables.



Physical Modeling of Data Warehouses

- The physical design of data warehouses is crucial to ensure adequate query response time.
- There are typically three common techniques for improving performance in data warehouse systems: materialized views, indexing, and partitioning.
- A materialized view is a view that is physically stored in a database, which enhances query performance by precalculating costly operations such as joins and aggregations.
- With respect to indexing, traditional techniques used in OLTP systems are not appropriate for multidimensional data. Thus, alternative indexing mechanisms are used in data warehouses, typically bitmap and join indexes.
  - A bitmap index is a special kind of database index that uses bitmaps. They are particularly useful for columns with a low number of distinct values.
  - Join indexes materialize a relational join between two tables by keeping pairs of row identifiers that participate in the join.
- Finally, partitioning or fragmentation divides the contents of a relation into several files, typically based on a range of values of an attribute.

Evolution of DSS into Business Intelligence

- In the early days of Decision Support Systems, manager let their staff do some supportive analysis by using DSS tools.
- As PC technology advanced, a new generation of managers evolved – one that was comfortable with computing and knew that technology can directly help make intelligent business decision faster.
- New tools such as OLAP, data warehousing, data mining and intelligent systems, delivered via Web technology added promised capabilities and easy access to tools, models and data for computer-aided decision making.
- These tools started to appear under the names Business Intelligence (BI) and business analytics in the mid-1990s.
- Next we will try to relate the DSS, data warehouses and Business Intelligence concepts.

Business Intelligence, Decision Support Systems and Data Warehousing

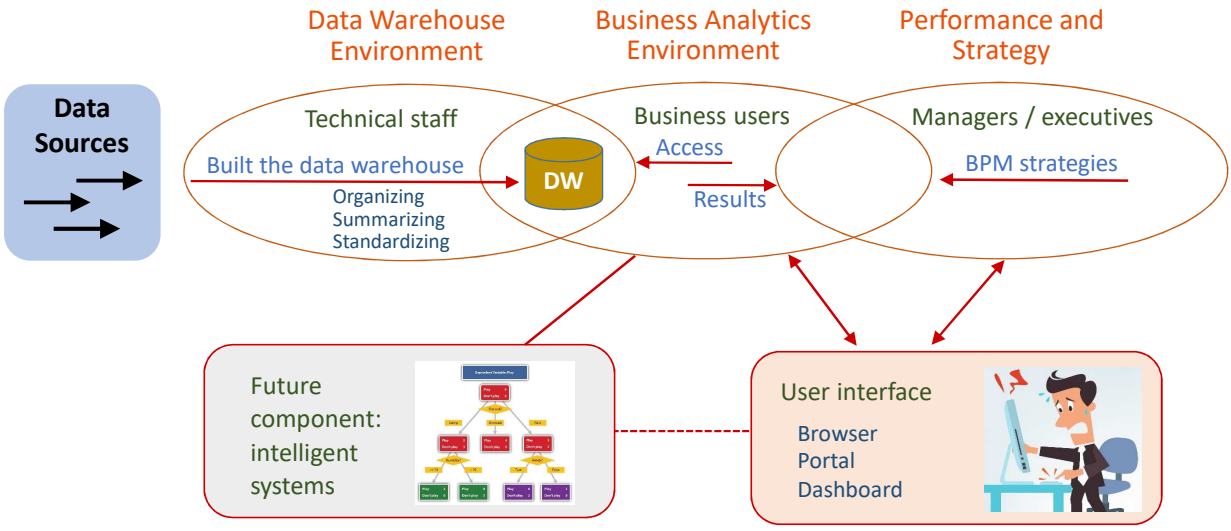
- Business intelligence comprises a collection of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information for decision making.
- Business intelligence and decision-support systems provide assistance to managers at various organizational levels for analyzing strategic information.
  - These systems collect vast amounts of data and reduce them to a form that can be used to analyze organizational behavior.
  - This data transformation comprises a set of tasks that take the data from the sources and, through extraction, transformation, integration, and cleansing processes, store the data in a common repository called a data warehouse.
- Data warehouses have been developed and deployed as an integral part of decision support systems to provide an infrastructure that enables users to obtain efficient and accurate responses to complex queries.

The High Level Architecture of BI (1/2)

☞ A Business Intelligence system has four major components:

- a data warehouse, with its source data
- business analytics, a collection of tools for manipulating, mining, and analyzing the data in the data warehouse
- business performance management (BPM) for monitoring and analyzing performance
- a user interface (e.g. dashboard).

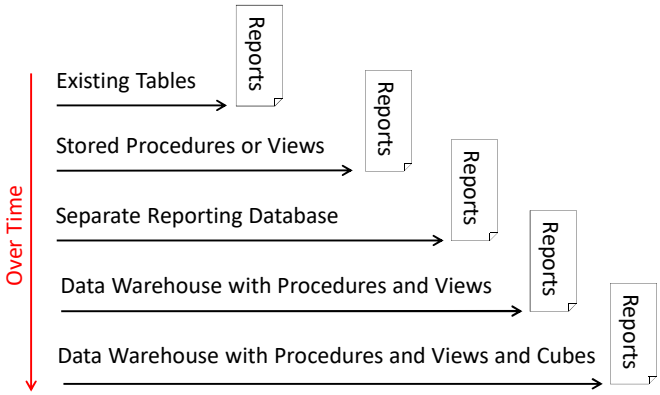
The High Level Architecture of BI (2/2)



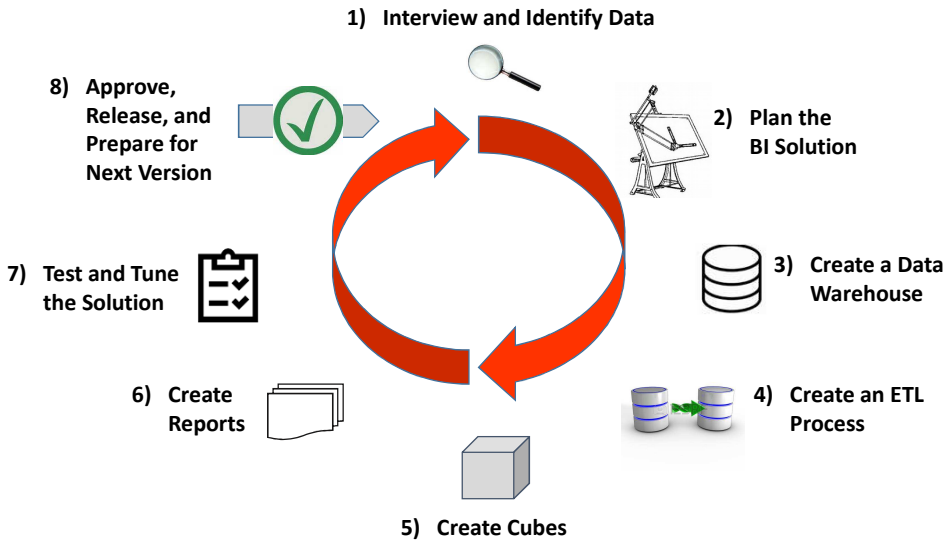
Business Intelligence Reporting Solutions

- The end goal of a BI solution is to convert data into usable information, and that information is routinely represented within reports.
- We will treat here the term business intelligence solutions as business reporting solutions.
- A company may begin by selecting report data directly from OLTP relational tables.
- The use of data warehouses and data cubes is an appropriate way to provide efficient reporting.

Reporting data sources change over time



The BI solution life cycle



Business Intelligence Tools

- Nowadays, the offer in business intelligence tools is quite large.
- The major database providers, such as Microsoft, Oracle, IBM, and Teradata, have their own suite of business intelligence tools.
  - Other popular tools include SAP, MicroStrategy, and Targit.
  - In addition to the above commercial tools, there are also open-source tools, of which Pentaho is the most popular one.
- For example, Microsoft SQL Server provides an integrated platform for building analytical applications. It is composed of three main components:
  - Analysis Services is an OLAP tool that provides analytical and data mining capabilities.
  - Integration Services supports ETL processes, which are used for loading and refreshing data warehouses on a periodic basis.
  - Reporting Services is used to define, generate, store, and manage reports.

The Microsoft BI Suite