Diplomatura en Big Data

Data Warehousing y OLAP

Alejandro VAISMAN
Departmento de Ingeniería Informática
Instituto Tecnológico de Buenos Aires
avaisman@itba.edu.ar

Introduction

Outline

- Introduction - Definition
- A Historical Overview of Data Warehousing
- OLAP vs OLTP
- Data Warehouse Architecture
- The Multidimensional Model

Introduction

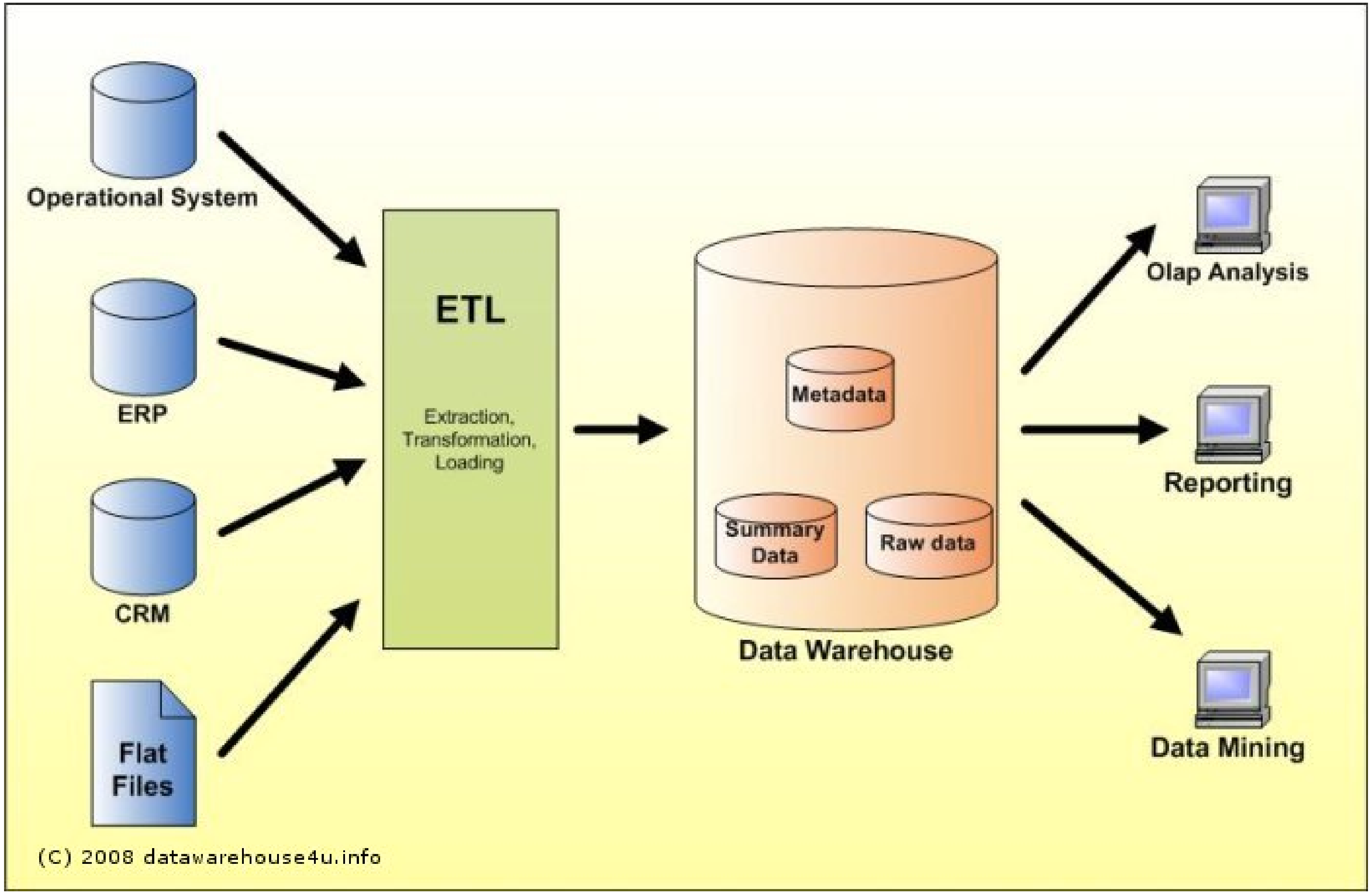
- Organizations facing increasingly complex challenges to achieve operational goals
- Analysis tools required for decision support
- **Business intelligence** (**BI**): Methodologies, processes, architectures, and technologies to transform raw data into useful information for decision making
- Business intelligence and decision-support systems used to analyze strategic information
- These systems collect and summarize vast amounts of data
- Extraction, transformation, integration, and cleansing processes take data from sources, and store them in a common repository called a **data warehouse**
  - Data warehouses: Integral part of decision-support systems
  - Provide an infrastructure that enables users to get efficient, accurate responses to complex queries

Introduction

- A wide variety of systems and tools to exploit the data in a warehouse
- Typical mechanism: **Online Analytical Processing** (**OLAP**)
- OLAP allow users to interactively query and aggregate the data in a warehouse
- Decision makers can analyze information at various levels of detail
- **Data mining** used since the 1990s to extract interesting knowledge hidden in data warehouses
- A large number of new BI techniques have been developed and used to assist decision making
- BI is shifting beyond the OLAP paradigm, toward **data analytics** (Cohen, 2009 - MAD Skills - Magnetic, Agile, Deep)
- Typical techniques that exploit a data warehouse (non-comprehensive list)
  - Reporting techniques: dashboards, alerts
  - Performance management: metrics, key performance indicators (KPIs), dashboards
  - Analytics: OLAP, data mining, time series analysis, text mining, web analytics, data visualization

BI Lifecycle

DW General Scheme



Introduction

Outline

- Introduction - Definition
- **A Historical Overview of Data Warehousing**
- OLAP vs OLTP
- Data Warehouse Architecture
- The Multidimensional Model

A Historical Overview of Data Warehousing

- Traditional **operational** or **transactional databases** do not satisfy the requirements for data analysis
- Designed and optimized to support daily business operations, their primary concern: concurrent access and recovery techniques to guarantee data consistency
- Operational databases contain detailed data, do not include historical data, and perform poorly for complex queries that involve many tables or aggregate large volumes of data
- To analyze the behavior of an organization, data from several operational systems must be integrated
- Difficult task to accomplish because of the differences in data definition and content
- **Data warehouses** were proposed as a solution to the growing demands of decision-making users

Data Warehouses

- Collection of subject-oriented, integrated, nonvolatile, and time-varying data to support management decisions (Immon definition)
- **Subject-oriented**: a data warehouse targets one or several subjects of analysis according to the analytical requirements of managers at various levels of the decision-making process
- **Integrated**: the content of a data warehouse result from the integration of data from various operational and external systems
- **Nonvolatile**: a data warehouse accumulates data from operational systems for a long period of time $\rightarrow$ modification and removal are not allowed the only operation is the purging of no longer needed, obsolete data
- **Time-varying**: a data warehouse keeps track of how its data has evolved over time; for instance, it may allow one to know the evolution of sales or inventory over the last months/years

Design of Operational Databases

- Typically performed in four phases
  - **Requirements specification**: needs of users are collected; the specification serves as basis for creating a database schema capable of responding to user queries
  - **Conceptual design**: describes an application without accounting for implementation considerations → entity-relationship (ER) model
  - **Logical design**: implementation paradigm for database applications; the most used logical model → relational model
  - **Physical design**: specific implementation over a DBMS
- Relational databases: highly normalized to guarantee consistency under frequent updates
- Usually achieved at a higher cost of querying (normalization partitions database into multiple tables)
- This paradigm not appropriate for data warehouses
- Data warehouses must deliver good performance for the complex queries needed for analysis tasks
- Lesser degree of normalization required → **Multidimensional modeling**

Design of Analytical Databases

- Multidimensional model
- Views data as consisting of facts linked to dimensions
- A **fact** represents the focus of analysis (e.g., analysis of sales in stores)
- **Measures** quantify facts; usually numeric values, e.g., amount or number of sales
- **Dimensions** used to analyze measures from several perspectives, e.g.:
  - Time dimension to analyze changes in sales over various periods of time
  - Location dimension to analyze sales according to the geographic distribution of stores
- Dimensions include attributes that form **hierarchies** which allow decision-making users to explore measures at various levels of detail, e.g.:
  - month → quarter → year in the time dimension
  - city → state → country in the location dimension
- **Aggregation** of measures occurs when a hierarchy is traversed, e.g., moving from month to year yields aggregated values of sales for the various years

Design of Analytical Databases

- No well-accepted conceptual model for data warehouse applications
- Data warehouse design usually performed at the logical level, leading to schemas that are difficult to understand by a typical user
- We use the **MultiDim model** for **conceptual** modeling
- At the **logical level**, the multidimensional model is usually represented by relational tables organized in **star schemas** and **snowflake schemas**
  - These schemas relate a fact table to several dimension tables
  - Star schemas use a unique table for each dimension, even in the presence of hierarchies (yields denormalized dimension tables)
  - Snowflake schemas use normalized tables for dimensions and their hierarchies
- Over this relational representation of a data warehouse, an OLAP server builds a data cube, which provides a multidimensional view of the data

Querying the Multidimensional Model

- **Query Languages**
- Once a data warehouse has been implemented, analytical queries can be submitted
- Two typical query languages
  - MDX (MultiDimensional eXpressions): de facto standard language for querying a multidimensional database
  - MDX provides a functionality for multidimensional databases similar to the one provided by SQL for traditional relational databases
  - A Cube in the FROM clause instead of tables
  - SQL/OLAP: A standard etension to SQL
  - Operates over relational databases rather than over cubes

Physical Design

- **Physical level**: concerned with implementation issues
- Three techniques are normally used for improving system performance:
  - Materialized views
  - Indexing (bitmap indices used)
  - Data partitioning (e.g., by year or other time periods, so recent data accessed faster)

ETL: Extraction, Transformation, and Loading

- Extracts data from several source systems, transforms data to fit the data warehouse model, and loads transformed data into the data warehouse
- Crucial for the success of a data warehousing project
- About 80% of the total cost
- Still no consensus on a methodology for ETL design, and most problems are solved ad hoc
- Several proposals regarding ETL conceptual design

Exploitation: Data Analytics

- **Data analytics** is the process of exploiting the contents of a data warehouse in order to provide essential information to the decision-making process
- Three main tools:
  - **Data mining**: a series of techniques that analyze the data in a warehouse in order to discover hidden useful knowledge
  - **Key performance indicators** (KPIs) are measurable organizational objectives used for monitoring how an organization is performing
  - **Dashboards** are interactive reports that present the data in a warehouse, including the KPIs, in a visual way, providing an overview of the performance of an organization for decision-support purposes

Introduction

Outline

- Introduction - Definition
- A Historical Overview of Data Warehousing
- **OLAP vs OLTP**
- Data Warehouse Architecture
- The Multidimensional Model

OLAP vs. OLTP

- Traditional database systems designed and tuned to support the day-to-day operation:
  - Ensure fast, concurrent access to data
  - Transaction processing and concurrency control
  - Focus on online update data consistency
  - Known as **operational databases** or **online transaction processing** (**OLTP**)
- OLTP DB data characteristics:
  - Detailed data
  - Do not include historical data
  - Highly normalized
  - Poor performance on complex queries including joins an aggregation
- Data analysis requires a new paradigm: **online analytical processing** (**OLAP**)
  - Typical **OLTP** query: pending orders for customer c1
  - Typical **OLAP** query: total sales amount by product and by customer

OLAP vs. OLTP

- **OLAP** characteristics
  - OLTP paradigm focused on transactions, OLAP focused on analytical queries
  - Normalization not good for analytical queries, reconstructing data requires a high number of joins
  - OLAP databases support a heavy query load
  - OLTP indexing techniques not efficient in OLAP: oriented to access few records
    - OLAP queries typically include aggregation
- The need for a different database model to support OLAP was clear: led to **data warehouses**
- Data warehouse: (usually) large repositories that consolidate data from different sources (internal and external to the organization), are updated offline, follow the **multidimensional data model**, designed and optimized to efficiently support OLAP queries

OLAP vs. OLTP

- No consensus about the relationship between data warehouses and data marts
  - A **bottom-up** approach: a data warehouse is built by first building the smaller data marts and then merging these to obtain the data warehouse
  - A **top-down** (classic data warehouse view): data marts are obtained from the data warehouse, a data mart just a logical view of a data warehouse
- Several **differences between OLTP and OLAP** systems

	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

OLAP vs. OLTP

- OLTP systems users: perform predefined operations through transactional applications (e.g., payroll or ticket reservation systems)
- Data warehouse users employ interactive OLAP tools to perform data analysis, for example, to detect salary inconsistencies or most frequent tourist destinations (Lines 1–2)
- Data:
  - OLTP systems: current and detailed
  - Data analytics require historical, summarized data (Line 3)
- OLTP requires R/W operations, e.g., insert new orders, modify old ones, and delete orders if customers cancel them → small number of records accessed
- OLAP complex aggregation queries, thus requiring **read** access to all the records in one or more tables
- OLAP systems not so frequently accessed as OLTP systems
- OLTP systems: short response time; complex OLAP queries can take long time to complete (Line 9)
- OLTP systems: heavy concurrent accesses (Lines 10–11).
- OLAP systems are read only → queries can be submitted and computed concurrently
- OLTP systems are constantly updated online; OLAP systems are updated offline periodically, then:
  - OLTP: highly normalized schema; OLAP: denormalized schema

Data Warehouse Concepts

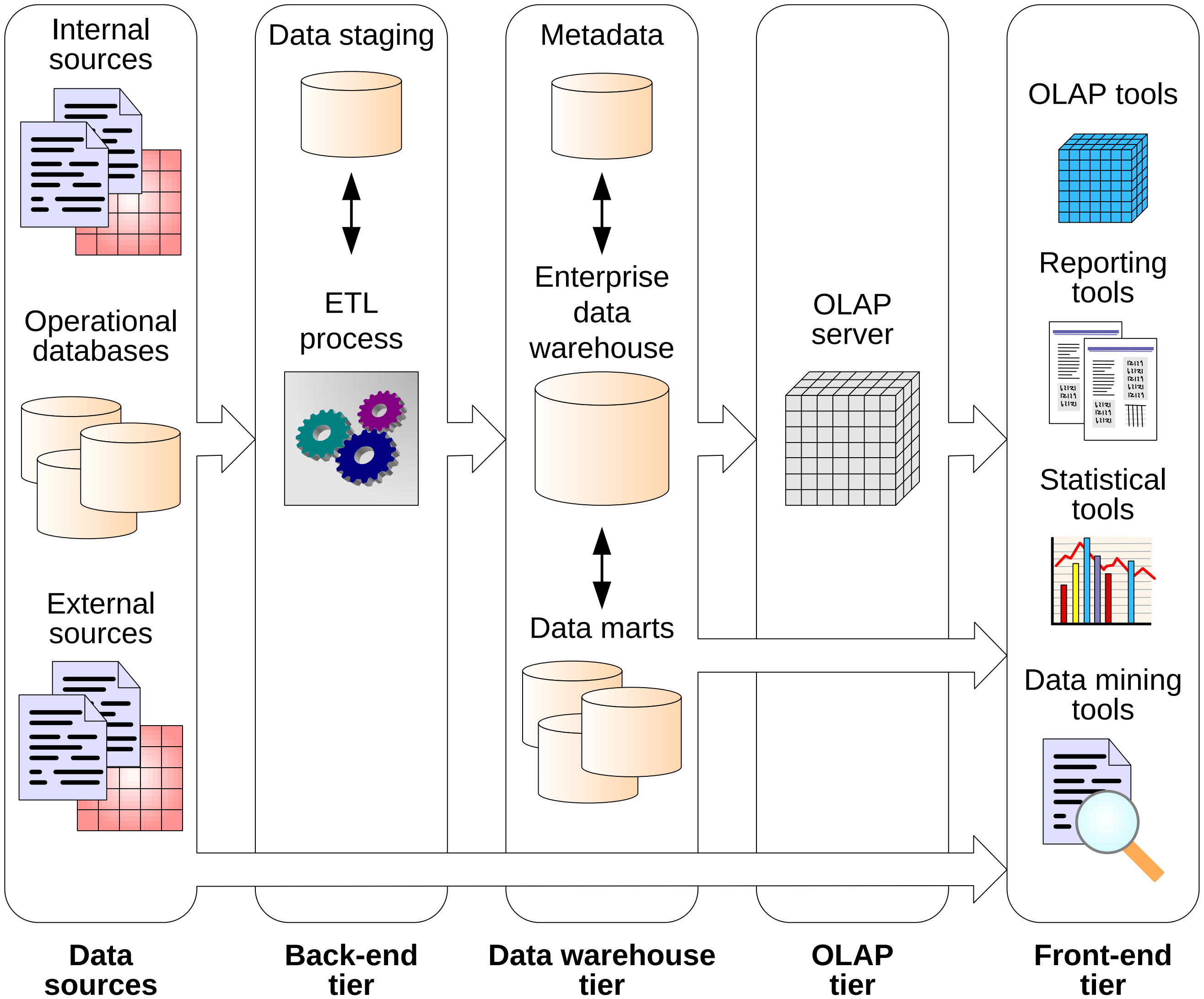
Outline

- Introduction - Definition
- A Historical Overview of Data Warehousing
- OLAP vs OLTP
- **Data Warehouse Architecture**
- The Multidimensional Model

Data Warehouse Architecture

- General data warehouse architecture: **several tiers**
- **Back-end tier** composed of:
  - The extraction, transformation, and loading (ETL) tools : Feed data into the data warehouse from operational databases and internal and external data sources
  - The data staging area: An intermediate database where all the data integration and transformation processes are run prior to the loading of the data into the data warehouse
- **Data warehouse tier** composed of:
  - An enterprise data warehouse and/or several data marts
  - A metadata repository storing information about the data warehouse and its contents
- **OLAP tier** composed of:
  - An OLAP server which provides a multidimensional view of the data, regardless the actual way in which data are stored
- **Front-end tier** is used for data analysis and visualization
  - Contains client tools such as OLAP tools, reporting tools, statistical tools, and data-mining tools

Typical Data Warehouse Architecture

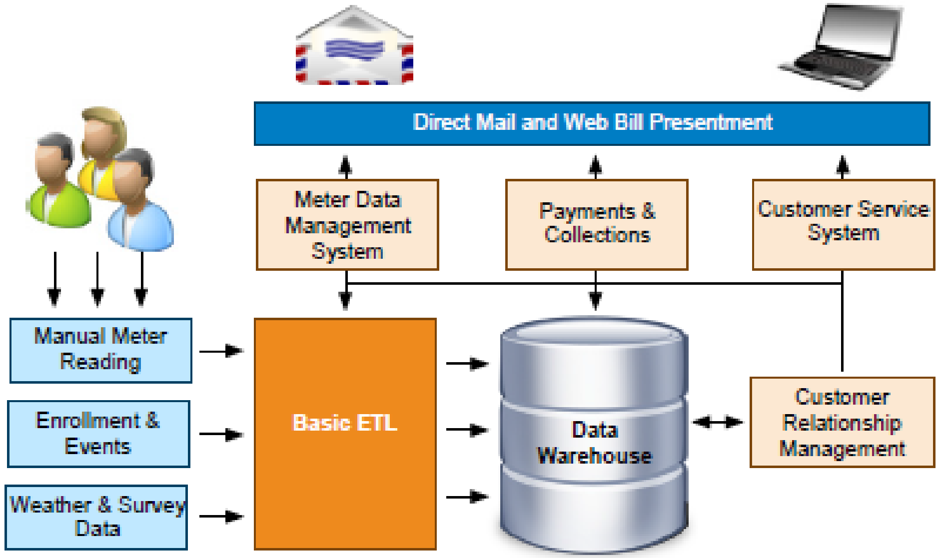


Back-End Tier

- Performs **Extraction, Transformation, and Loading**
- It is a three-step process
- **Extraction** gathers data from multiple, heterogeneous data sources internal or external to the organization
- **Transformation** modifies the data from the format of the data sources to the warehouse format; this includes:
  - 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 granularity of the data warehouse
- **Loading** feeds the data warehouse with the transformed data, including refreshing the data warehouse, that is, propagating updates from the data sources to the data warehouse at a specified frequency
- Data staging area (usually called operational data store): A database where data extracted from the sources undergoes successive modifications before being loaded into the data warehouse

Back-End Tier

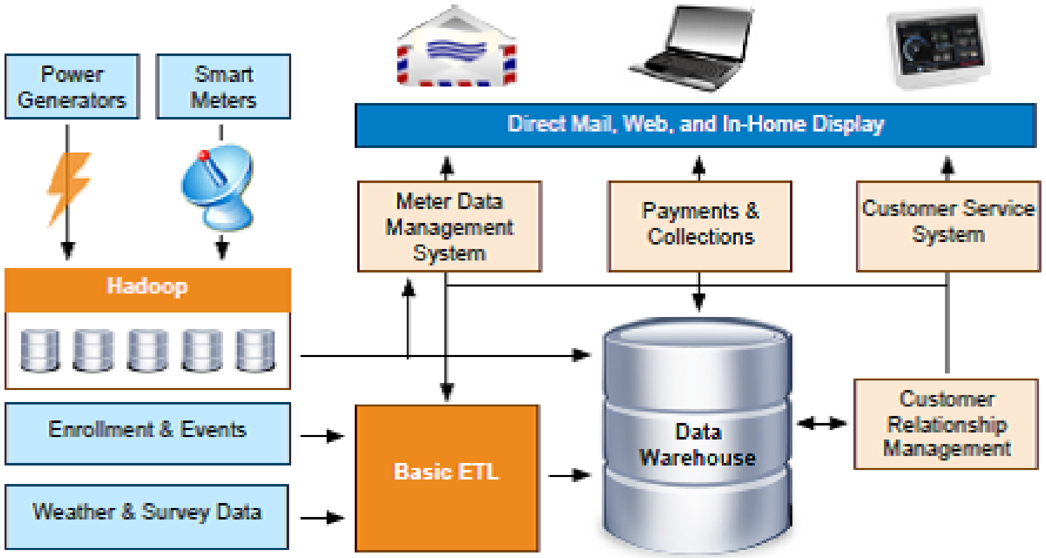
- Example: DW for smart meter analysis: energy saving, offers, etc.
- Traditional solution (see article Hadoop and DW)



- Data loaded from manual processes each quarter: 10 million readings
- Smart meters: a measure each 5 minutes → 100 billion reads each quarter
- A new solution required

Back-End Tier

- Example: DW for smart meter analysis: energy saving, offers, etc.
- Hadoop-based solution (see article Hadoop and DW)



- Raw data moved into Hadoop, processed with Pig and moved to the DW
- Hadoop not an ETL tool, just supports ETL

Data Warehouse Tier

- **Components**:
  - An **enterprise data warehouse**, centralized and encompassing an entire organization
  - Several **data marts**, specialized departmental data warehouses
- **Metadata**
  - **Business metadata** describes the semantics of the data, and organizational rules, policies, and constraints related to the data
  - **Technical metadata** describes how data are structured and stored in a computer system, and the applications and processes that manipulate the data
- The metadata repository may contain information such as:
  - Metadata describing the structure of the data warehouse and the data marts, at the conceptual/logical level (facts, dimensions, hierarchies, ...) and at the physical level (indexes, partitions,...)
  - Security information (user authorization and access control), and monitoring information (usage statistics, error reports, audit trails)
  - Metadata describing data sources: schemas, ownership, update frequencies, legal limitations, access methods
  - Metadata describing the ETL: data lineage, data extraction, cleaning, transformation rules, etc.

OLAP Tier

- Composed of an OLAP server, which presents business users with multidimensional data from data warehouses or data marts
- Products include OLAP extensions and tools allowing building, querying, and navigating cubes, analysis, and reporting
- Not yet a standardized language for defining and manipulating data cubes
- MDX (MultiDimensional eXpressions): query language for OLAP databases, a de facto standard for querying OLAP systems
- SQL extended for providing analytical capabilities: SQL/OLAP

Front-End Tier

- Client tools that allow users to exploit the content of the data warehouse
- **OLAP tools**: allow interactive exploration and manipulation of the warehouse data and formulation of complex **ad hoc queries**
- **Reporting tools** enable the production, delivery, and management of reports, which can be paper-based, interactive, or web-based
- Reports use **predefined queries** queries asking for specific information in a specific format, performed on a regular basis
- **Statistical tools**: used to analyze and visualize the cube data using statistical methods
- **Data-mining tools** allow users to analyze data in order to discover valuable knowledge such as patterns and trends, and also allow to make predictions based on current data

Variations of the Architecture

- Sometimes: only an enterprise data warehouse without data marts or, alternatively, an enterprise data warehouse does not exist
- In other situations, an 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)
- Extreme situation: neither a data warehouse nor an OLAP server → **virtual data warehouse**, which defines a set of views over operational databases that are materialized for efficient access
- Virtual data warehouse is easy to build but does not provide a real data warehouse solution (does not contain historical data, centralized metadata, etc.)
- Data staging area may not be needed when the data in the source systems conforms very closely to the data in the warehouse

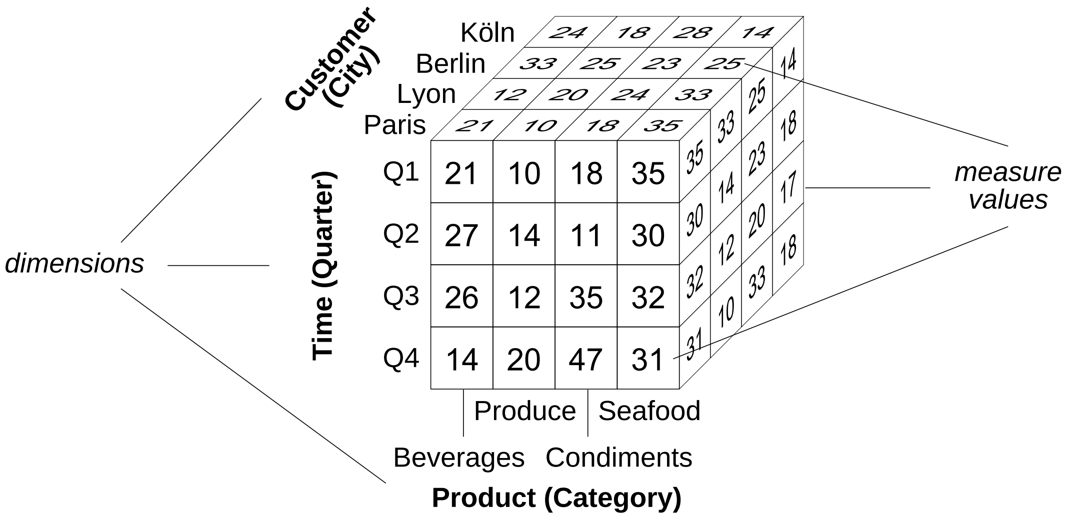
Data Warehouse Concepts

Outline

- Introduction - Definition
- A Historical Overview of Data Warehousing
- OLAP vs OLTP
- Data Warehouse Architecture
- **The Multidimensional Model**

The Multidimensional Model

- Views data in an n-dimensional space: A **data cube**
- A data cube is composed of dimensions and facts
- **Dimensions**: Perspectives used to analyze the data
  - Example: A three-dimensional cube for sales data with dimensions Product, Time, and Customer, and a measure Quantity



- **Attributes** describe dimensions
  - Product dimension may have attributes ProductNumber and UnitPrice (not shown in the figure)
- The **cells** or **facts** of a data cube have associated numeric values called **measures**
- Each cell of the data cube represents Quantity of units sold by category, quarter, and customer's city

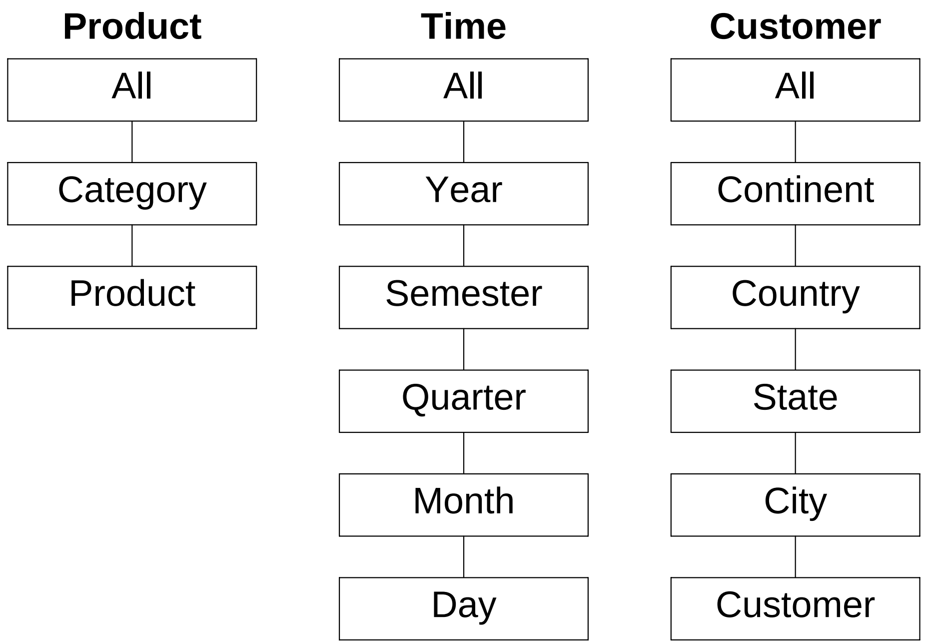
Instituto Tecnológico de Buenos Aires

The Multidimensional Model

- **Data granularity**: level of detail at which measures are represented for each dimension of the cube
  - Example: sales figures aggregated to granularities Category, Quarter, and City
  - We may want sales figures at a finer granularity (Month), or at a coarser granularity (Country)
- Elements of a dimension at a certain granularity are called **members**
  - Example: Seafood and Beverages are members of Product at granularity Category
- A dimension **instance** comprises all members at all granularity levels in a dimension
- A data cube contains several measures, e.g. Amount, indicating the total sales amount (not shown)
- A data cube may be **sparse** (typical case) or **dense**
  - Example: not all customers may have ordered products of all categories during all quarters
- **Hierarchies**: allow viewing data at several granularities
  - Define a sequence of mappings relating lower-level, detailed concepts to higher-level ones
  - 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**

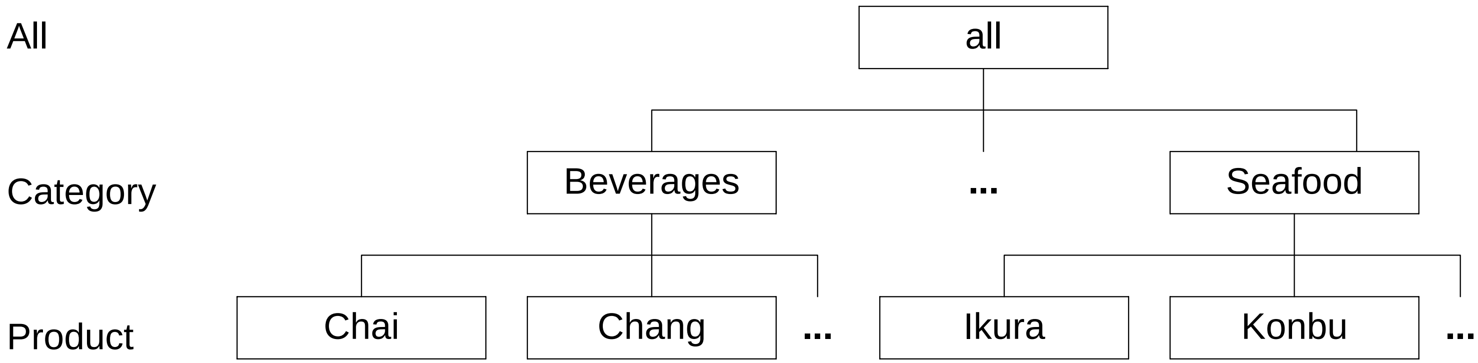
The Multidimensional Model

- Hierarchies of the Product, Time, and Customer dimensions



The Multidimensional Model

- Members of a hierarchy Product → Category



The Multidimensional Model: Measures

- Aggregation of measures changes the abstraction level at which data in a cube are visualized
- Measures can be:
  - **Additive**: can be meaningfully summarized along all the dimensions, using addition
    - The most common type of measures
  - **Semiadditive**: can be meaningfully summarized using addition along *some* dimensions
    - Example: inventory quantities, which cannot be added along the Time dimension
  - **Nonadditive measures** cannot be meaningfully summarized using addition across any dimension
    - Example: item price, cost per unit, and exchange rate
- Another classification of measures:
  - **Distributive**: defined by an aggregation function that can be computed in a distributed way
    - Functions count, sum, minimum, and maximum are distributive, distinct count is not
    - Example: $S = \{3, 3, 4, 5, 8, 4, 7, 3, 8\}$ partitioned in subsets $\{3, 3, 4\}$, $\{5, 8, 4\}$, $\{7, 3, 8\}$ gives a result of 8, while the answer over the original set is 5
  - **Algebraic measures** are defined by an aggregation function that can be expressed as a scalar function of distributive ones; example: average, computed by dividing the sum by the count
  - **Holistic measures** cannot be computed from other subaggregates (e.g., median, rank)

The Multidimensional Model: Measures

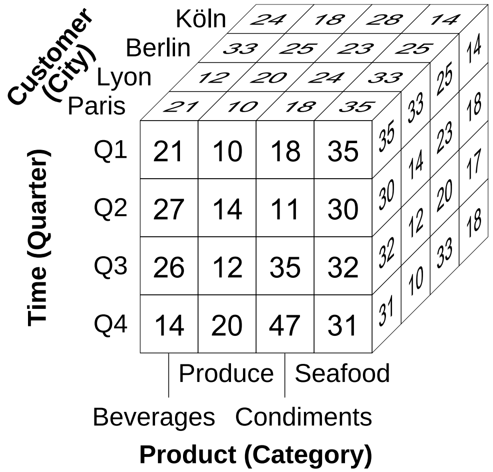
- When defining a measure we must determine the associated aggregation functions
  - For example, a semiadditive measure representing inventory quantities can be aggregated using average along the Time dimension, and using addition along other dimensions
- **Summarizability** refers to the correct aggregation of cube measures along dimension hierarchies
- Summarizability conditions:
  - **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 are included in the hierarchy and each instance is related to one parent in the next level
  - **Correctness:** refers to the correct use of the aggregation functions (more on this next)

Data Warehouse Concepts

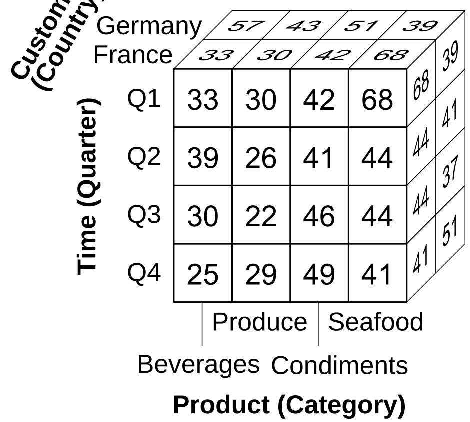
Outline

- Data Warehouses
- Data Warehouse Architecture
- The Multidimensional Model
- **OLAP Operations**

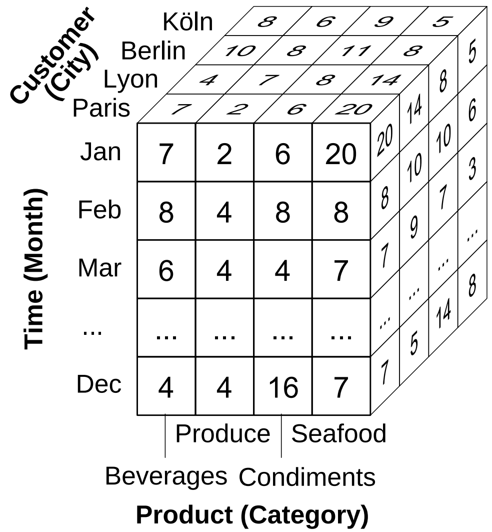
OLAP Operations



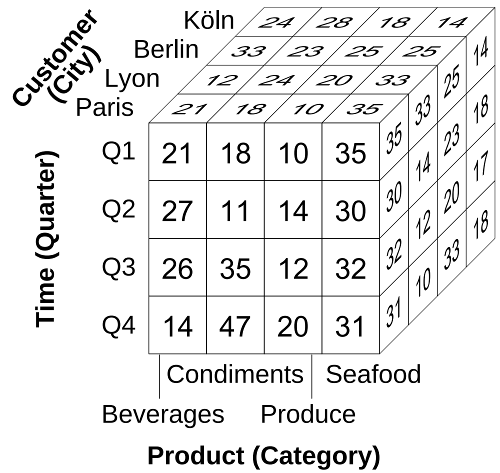
Original cube



Roll-up to the Country level



Drill-down to the Month level

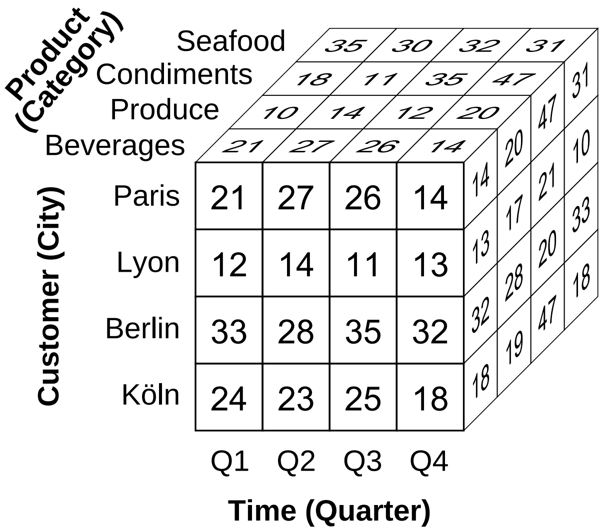


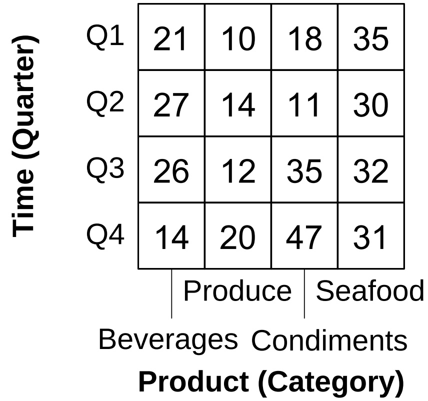
Sort product by name

OLAP Operations

- Starting cube: quarterly sales (in thousands) by product category and customer cities for 2012
- We first compute the sales quantities by country: a **roll-up** operation to the Country level along the Customer dimension
- Sales of category Seafood in France significantly higher in the first quarter
  - To find out if this occurred during a particular month, we take cube back to City aggregation level, and **drill-down** along Time to the Month level
- To explore alternative visualizations, we **sort** products by name
- To see the cube with the Time dimension on the x axis, we rotate the axes of the original cube, without changing granularities → **pivoting** (see next slide)
- To visualize the data only for Paris → **slice** operation, results in a 2-dimensional subcube, basically a collection of time series (see next slide)
- To obtain a 3-dimensional subcube containing only sales for the first two quarters and for the cities Lyon and Paris, we go back to the original cube and apply a **dice** operation

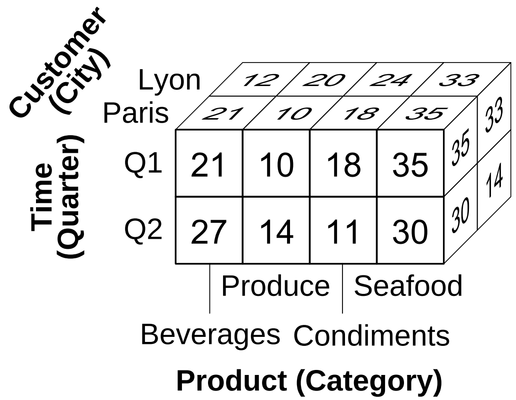
OLAP Operations





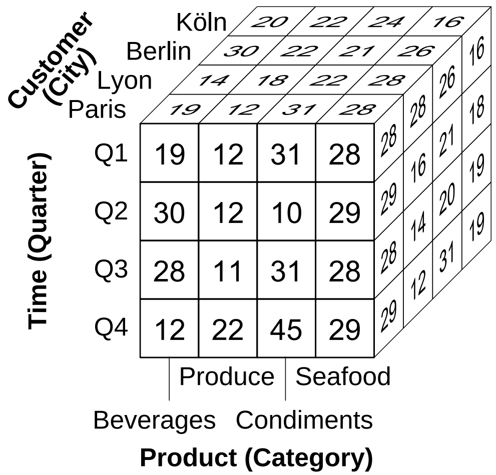
Pivot

Slice on City='Paris'

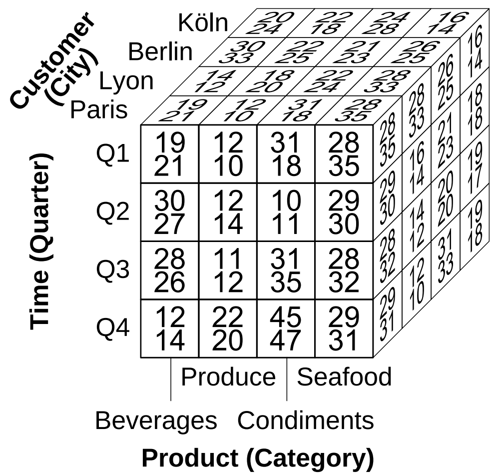


Dice on City='Paris' or 'Lyon' and Quarter='Q1' or 'Q2'

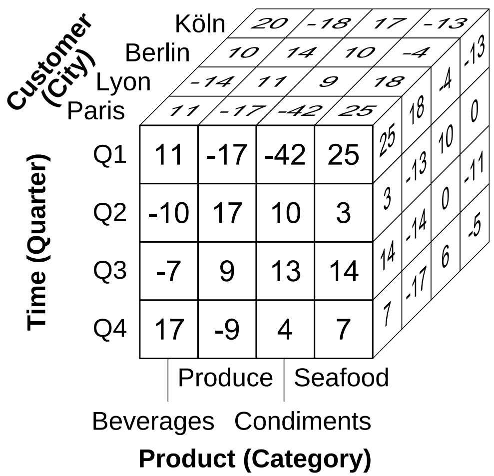
Advanced OLAP Operations



Cube for 2011



Drill-across operator



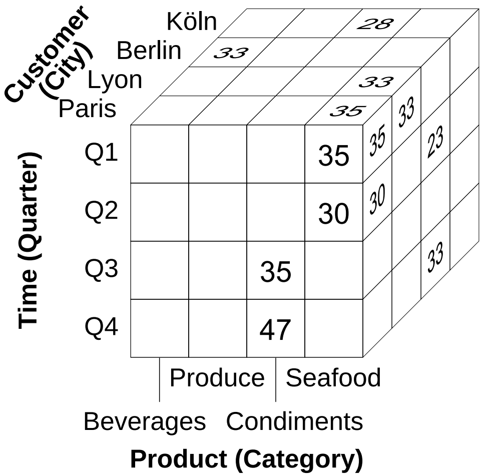
Percentage change

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

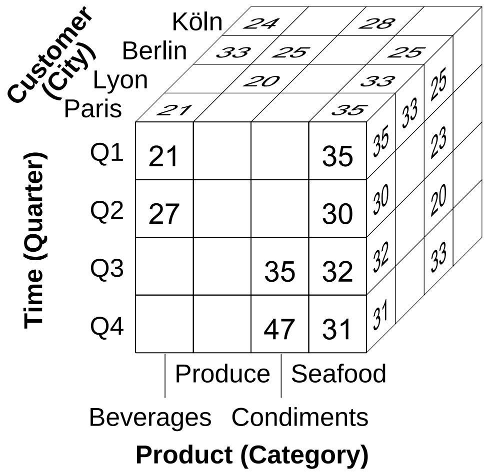
Customer (City)

Total sales by quarter and city

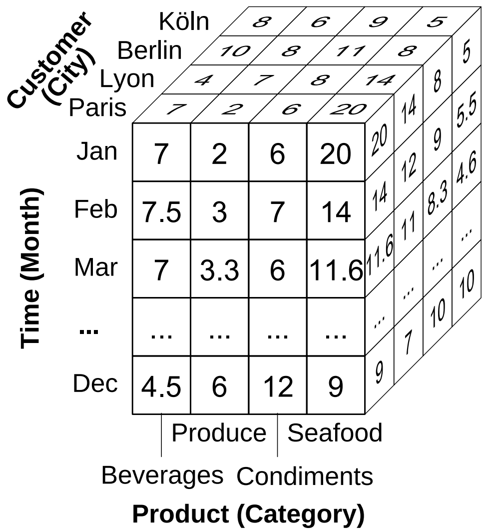
Advanced OLAP Operations



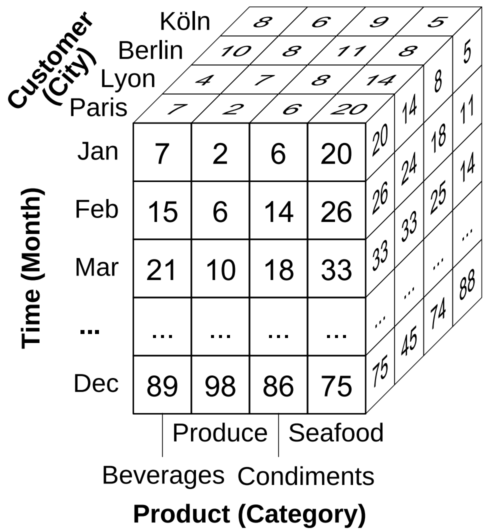
Maximum sales by quarter and city



Top two sales by quarter and city



Three-months moving average



Year-to-date sum

Summarizing OLAP Operations

Operator	Purpose
Add measure	Adds a new measure to a cube computed from other measures or dimensions.
Aggregation operators	Aggregates the cells of a cube, possibly after performing a grouping of cells.
Dice	Keeps the cells that satisfy a Boolean condition over dimension levels, attributes, and measures.
Difference	Removes the cells of a cube that are in another cube. Both cubes must have the same schema.
Drill-across	Merges two cubes that have the same schema and instances using a join condition.
Drill-down	Disaggregates measures along a dimension hierarchy to obtain data at a finer granularity. It is the opposite of the roll-up operation.
Drill-through	Shows data in the operational systems from which the cube was derived. This operation does not formally belong to the OLAP algebra since the result is not a cube.
Drop measure	Removes one or several measures from a cube.
Pivot	Rotates the axes of a cube to provide an alternative presentation of its data.
Recursive roll-up	Performs an iteration of roll-ups over a recursive hierarchy until the top level is reached.
Rename	Renames one or several schema elements of a cube.
Roll-up	Aggregates measures along a dimension hierarchy to obtain data at a coarser granularity. It is the opposite of the drill-down operation.
Roll-up*	Shorthand notation for a sequence of roll-up operations.
Slice	Removes a dimension by fixing a single value in a level of the dimension.
Sort	Orders the members of a dimension according to an expression.
Union	Combines the cells of two cubes that have the same schema but disjoint members.