Data Warehousing

Unit - 2


WELCOME TO THE DARK SIDE OF SCIENCE
DATA SCIENCE
memegenerator.net

Data Warehouse and OLAP

- Data analytics, also often known as business intelligence, is the strategies and technology that enable enterprises to gain deep and actionable insights into business data.
- Data mining plays the core role in data analytics and business intelligence.
- Data Warehouses generalize and consolidate data in multidimensional space. The construction of data warehouses involves data cleaning, data integration, and data transformation, and can be viewed as an important preparation step for data mining.
- Data Warehouses provide online analytical processing (OLAP) tools for interactive analysis of multidimensional data of varied granularities, which facilitates effective data generalization and data mining.
- Many other data mining functions, such as association, classification, prediction, and clustering, can be integrated with OLAP operations to enhance interactive mining of knowledge at multiple levels of abstraction.

What is A Data Warehouse?

- Data Warehouse refers to a data repository that is specific for analysis and is maintained separately from an organization's operational databases.
- Data warehouse systems support information processing by providing a solid platform of consolidated historic data for analysis.
- According to William H. Inmon, a leading architect in construction of data warehouse systems, "**A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process**".

Data Warehouse Features: Subject Oriented

Four major features of a data warehouse:

- A data warehouse is a **subject-oriented, integrated, time-variant**, and **nonvolatile** collection of data in support of management's decision-making process.

Subject-Oriented:

- A data warehouse is organized around major subjects, such as customer, product, sales.
- A data warehouse focuses on the modeling and analysis of data for decision makers, not on daily operations or transaction processing.
- A data warehouse provides a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.

Data Warehouse Features: Integrated

Integrated:

- A data warehouse is constructed by integrating multiple heterogeneous data sources such as relational databases, flat files, on-line transaction records.
- Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, etc.
  - When data is moved to the warehouse from operational databases, it is converted.

Data Warehouse Features: Time-variant

Time-Variant:

- The time horizon for a data warehouse is significantly longer than that of operational systems
  - Operational database: current value data
  - Data warehouse data: provide information from a historical perspective (e.g., past 5-10 years)
- Every key structure in the data warehouse contains an element of time, explicitly or implicitly.
  - But the key structure of operational data may or may not contain "time element"

DATA WAREHOUSE FEATURES: NONVOLATILE

Nonvolatile:

- A data warehouse is a physically separate store of data transformed from the operational environment.
- Operational update of data does not occur in a data warehouse environment.
  - Does not require transaction processing, recovery, and concurrency control mechanisms
  - Requires only two operations in data accessing:
    - initial loading of data and access of data

Data Warehouse

- In summary, a data warehouse is a semantically consistent and persistent data store that serves as a physical implementation of a decision support data model. It stores the information that an enterprise needs to make strategic decisions.
- A data warehouse is also often viewed as an architecture, constructed by integrating data from multiple heterogeneous sources to support structured and/or ad hoc queries, analytical reporting, and decision making.
- ***Data Warehousing*** *is the process of constructing and using data warehouses. The construction of a data warehouse requires data cleaning, data integration, and data consolidation.*

How do organizations use information from Data Warehouses?

- Many organizations use this information to support business decision-making activities.
- For example, by identifying the groups of most active customers an e-commerce company can design promotion campaigns to retain those customers firmly.
- By analyzing the sales patterns of products in different seasons, a company may design supply chain strategies to reduce the stocking cost of seasonal products.
- Analytic results from data warehouses are often presented to analysts and decision makers through periodic or ad hoc reports, such as daily, weekly, and monthly sales analysis reports analyzing sales patterns on customer groups, regions, products, and promotions.

DATA WAREHOUSE VS OPERATIONAL DATABASE SYSTEMS

- The major task of on-line **operational database systems** is to perform on-line transaction and query processing.
  - These systems are called **on-line transaction processing (OLTP)** systems.
  - They cover most of the day-to-day operations of an organization, such as purchasing, inventory, banking, payroll, registration, and accounting.

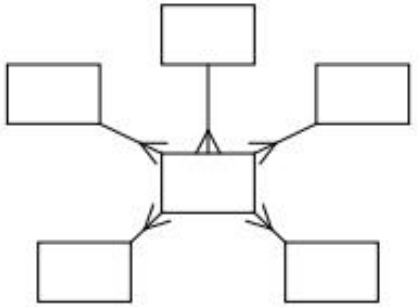
- **Data warehouse systems**, on the other hand, serve users or knowledge workers in the role of data analysis and decision making.
  - Such systems can organize and present data in various formats in order to accommodate the diverse needs of the different users.
  - These systems are known as **on-line analytical processing (OLAP)** systems.

Data Warehouse vs Operational Database System

Key	Data warehouse	Operational Database
Basic	A data warehouse is a repository for structured, filtered data that has already been processed for a specific purpose	Operational Database are those databases where data changes frequently
Data Structure	Data warehouse has denormalized schema	It has normalized schema
Transaction Optimization	Optimized for bulk loads and large complex, unpredictable queries.	Optimized for a common and known set of transactions.
Performance	It is fast for analysis queries	It is slow for analytics queries
Type of Data	It focuses on historical data	It focuses on current transactional data
Uses Case	It is used for OLAP	It is used for OLTP
Data Updates	Batch updates	Continuous updates
Query Handling	Usually very complex queries	Simple to complex queries

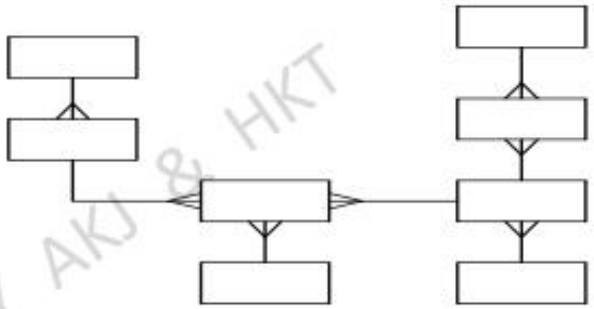
Data Warehouses Vs Operational Database Systems - Design point of view

Data Warehouse

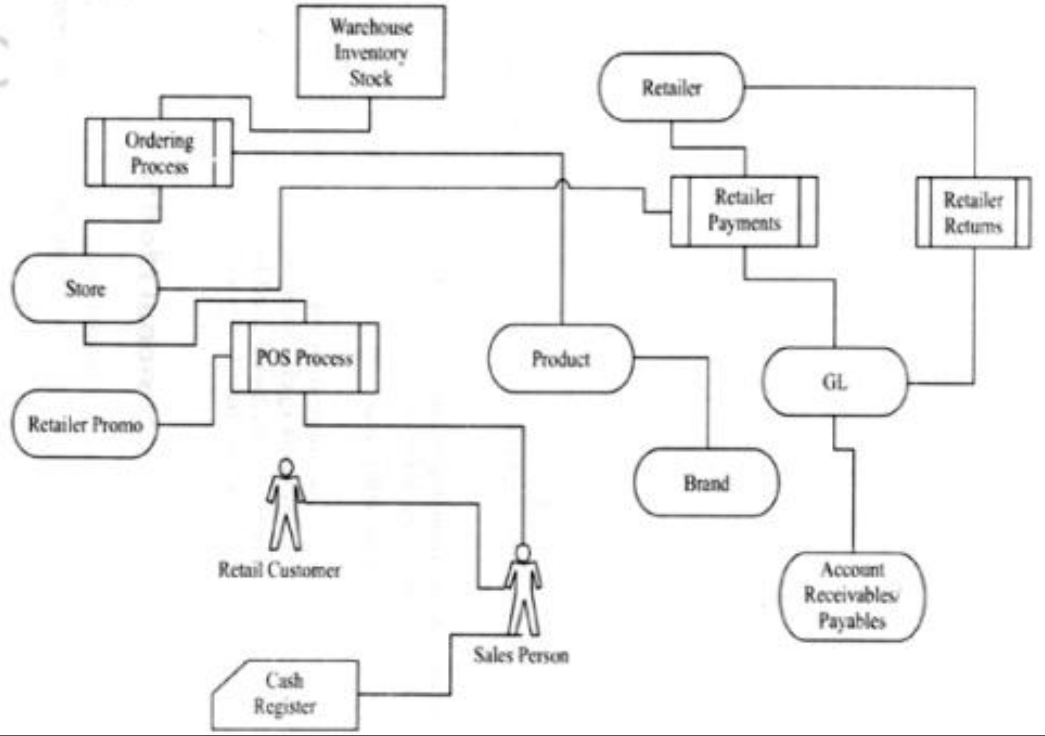


Star Schema

Operational Database System



ER Diagram



OLTP vs OLAP

Users and System Orientation:

- An OLTP system is *transaction-oriented* and is used for operation execution by clerks and clients.
- An OLAP system is business insight-oriented and is used for data summarization and analysis by knowledge workers, including managers, executives, and analysts.

Data Contents:

- An OLTP system manages current data that are typically too detailed to be easily used for business decision making.
- An OLAP system manages large amounts of historic data, provides facilities for summarization and aggregation, and stores and manages information at different levels of granularity, such as weekly-monthly-annually. These features make data easier to be used for informed decision making.

OLTP vs OLAP

- **Database Design:**
  - An OLTP system usually adopts an entity-relationship (ER) data model and an application-oriented database design.
  - An OLAP system typically adopts either a star model or a snowflake model and a subject-oriented database design.
- **View:**
  - An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historic data or data in different organizations.
  - In contrast, an OLAP system often spans multiple versions of a database schema, due to the evolutionary process of an organization. OLAP systems also deal with information that originates from different organizations, integrating information from many data stores.

OLTP vs OLAP

Access Patterns:

- The access patterns of an OLTP system consist mainly of short, atomic transactions, such as transferring an amount from one account to another. Such a system requires concurrency control and recovery mechanisms.
- However, accesses to OLAP systems are mostly read-only operations (because most data warehouses store historic rather than up-to-date information). Many accesses may be complex queries.

Data Warehouse vs Operational DBMS

- OLTP (on-line transaction processing)
  - Major task of traditional relational DBMS
  - Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
- OLAP (on-line analytical processing)
  - Major task of data warehouse system
  - Data analysis and decision making
- Distinct features (OLTP vs. OLAP):
  - User and system orientation: customer vs. market
  - Data contents: current, detailed vs. historical, consolidated
  - Database design: ER + application vs. star + subject
  - View: current, local vs. evolutionary, integrated
  - Access patterns: update vs. read-only but complex queries

Why a separate Data Warehouse?

- Why not perform OLAP directly on operational databases instead of constructing a separate data warehouse?
- **To ensure High performance of both systems -**
  - An operational database is designed and tuned from known tasks and workloads like indexing and hashing using primary keys, searching for particular records, and optimizing "canned" queries, which are preprogrammed and frequently used queries in business.
  - OLAP queries, however, are often complex. They involve the computation of large data groups at summarized levels and may require the use of special data organization, access, and implementation methods based on multidimensional views.
  - Processing OLAP queries directly in operational databases may substantially jeopardize the performance of operational tasks.

WHY A SEPARATE DATA WAREHOUSE?

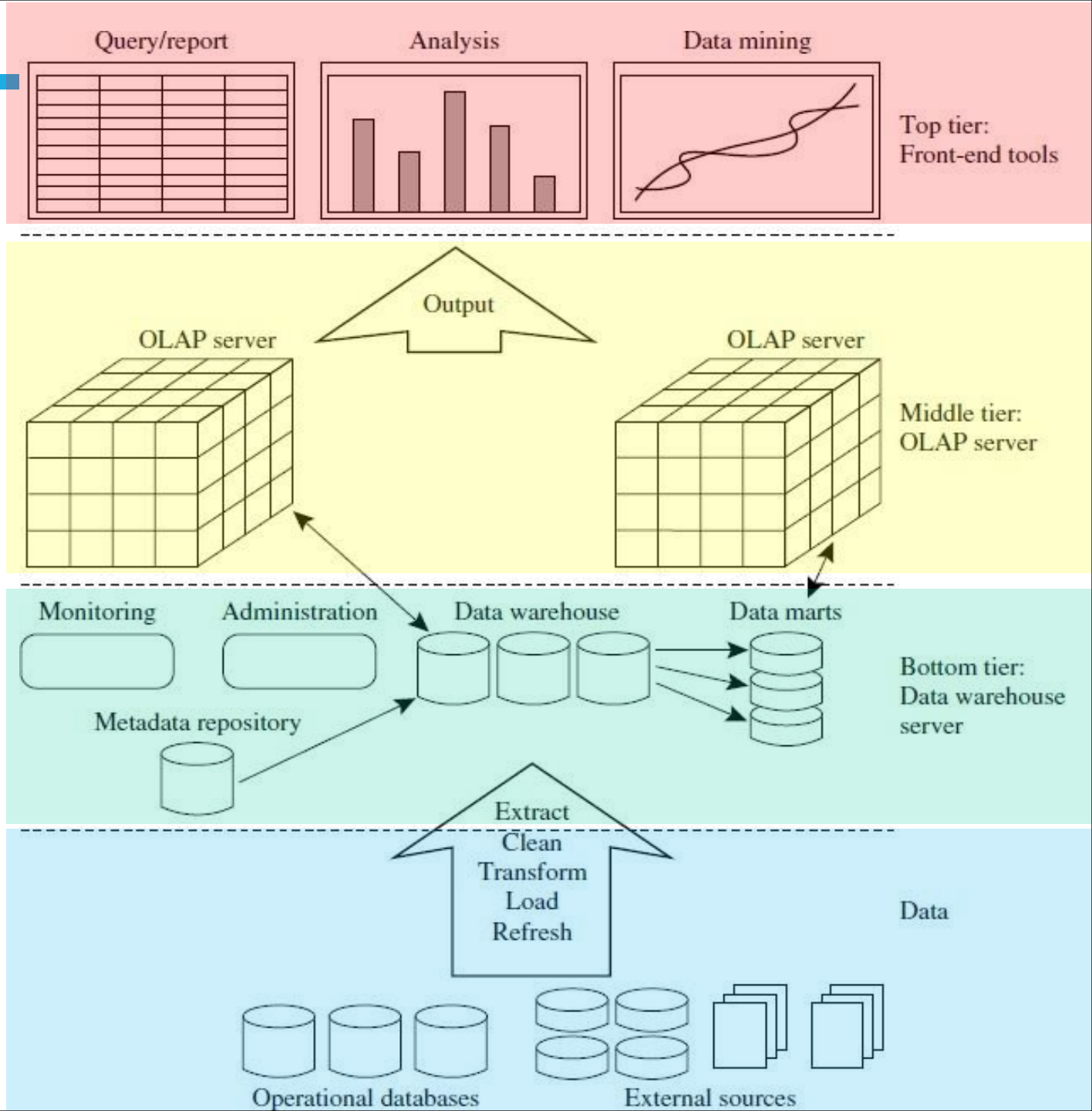
- **To ensure High performance of both systems -**
  - An operational database supports the concurrent processing of multiple transactions. Concurrency control and recovery mechanisms (e.g., locking and logging) are required to ensure the consistency and robustness of transactions.
  - An OLAP query often needs read-only access of massive data records for summarization and aggregation. Concurrency control and recovery mechanisms, if applied for such OLAP operations, may seriously delay the execution of concurrent transactions and thus substantially reduce the throughput of an OLTP system.

Why a separate Data Warehouse?

- **Different functions and Different data -**
  - Decision support requires historic data, whereas operational databases do not typically maintain historic data. In this context, the data in operational databases are usually far from complete for decision making.
  - Decision support requires consolidation (e.g., aggregation and summarization) of data from heterogeneous sources, resulting in high-quality, clean, and integrated data. In contrast, operational databases contain only detailed raw data, such as transactions, which need to be consolidated before analysis.
- Because the two systems provide quite different functionalities and require different kinds of data, it is presently necessary to maintain separate databases.

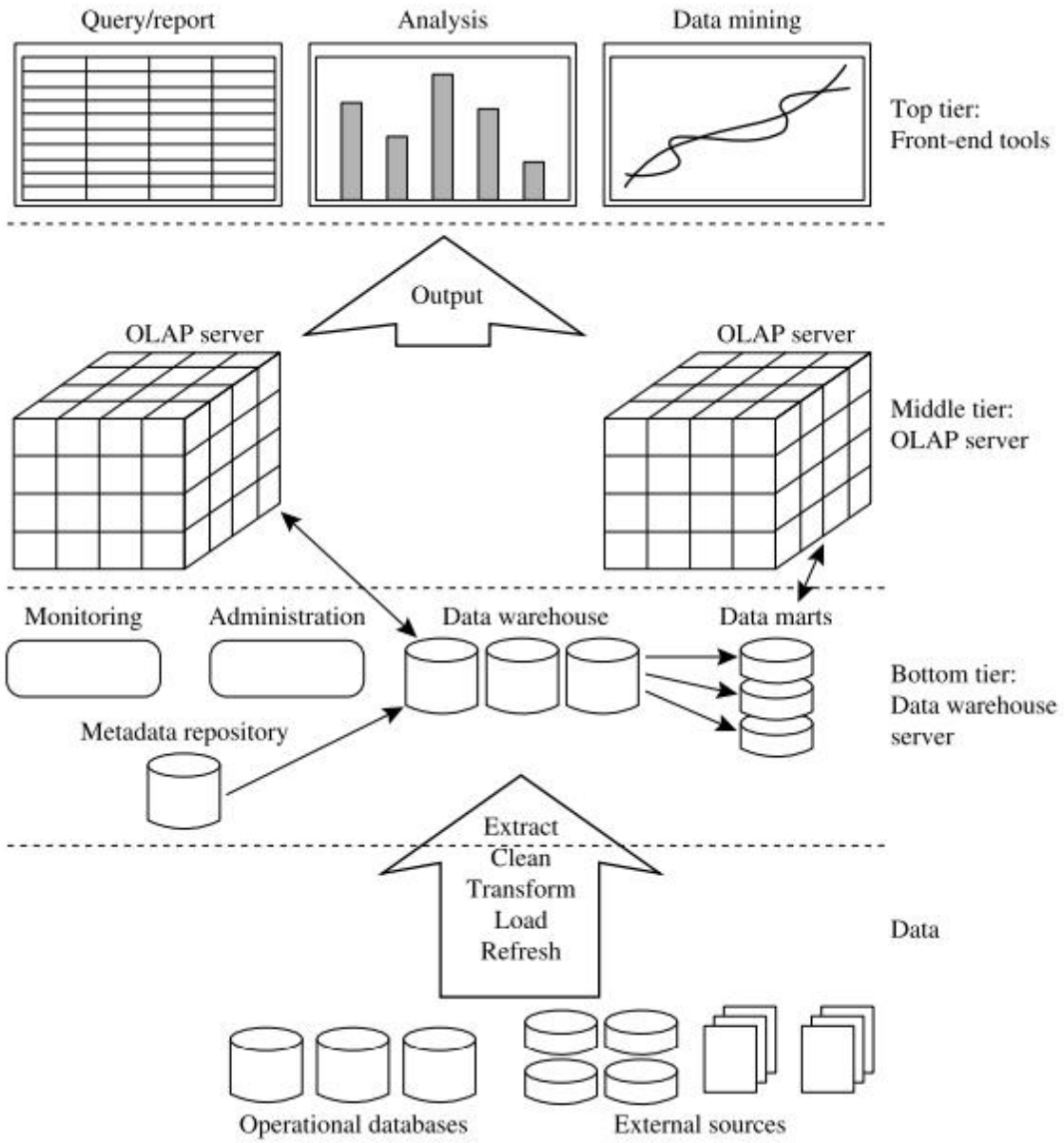
ACHITECTURE OF DATA WAREHOUSE

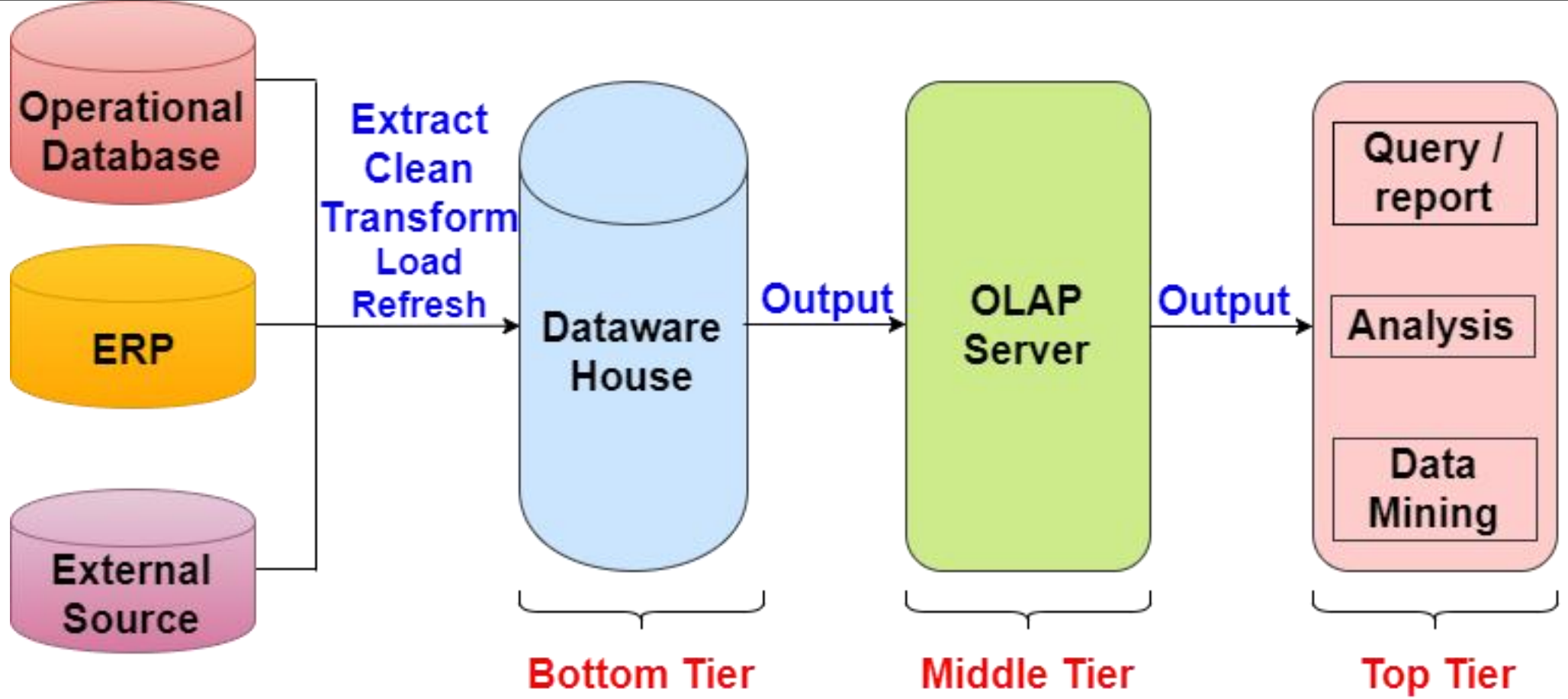
- Data warehouses often adopt ***a three-tier architecture.***



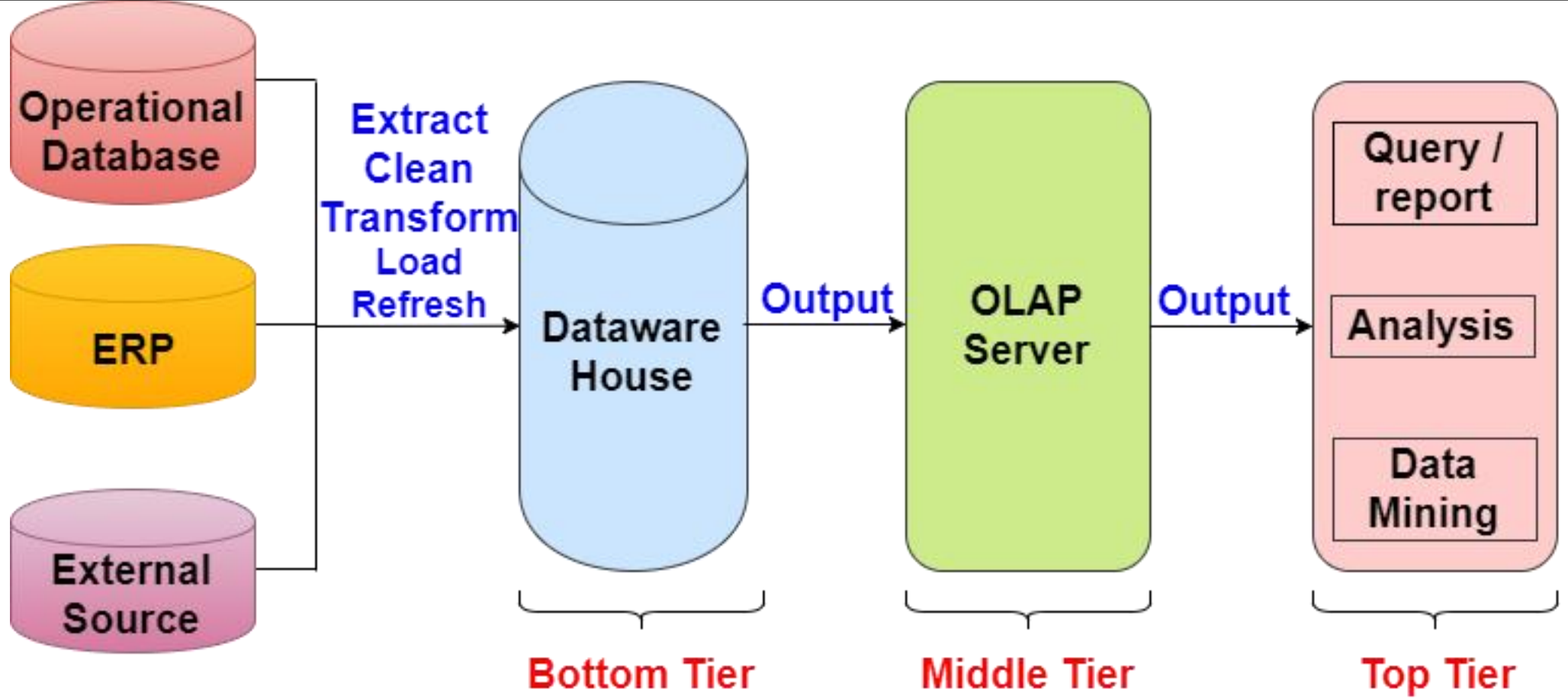
The Three-Tier Architecture

- The **bottom level** is a warehouse database server that is typically a main-stream database system, such as a relational database or a key-value store.
- Back-end tools and data extraction/transformation/loading (ETL) utilities are used to feed data into the bottom tier from operational databases or other external sources (e.g., customer profile information provided by external partners).
- These tools and utilities perform data extraction, cleaning, and transformation, as well as load and refresh functions to update the data warehouse.
- This tier also contains a metadata repository, which stores information about the data warehouse and its contents.





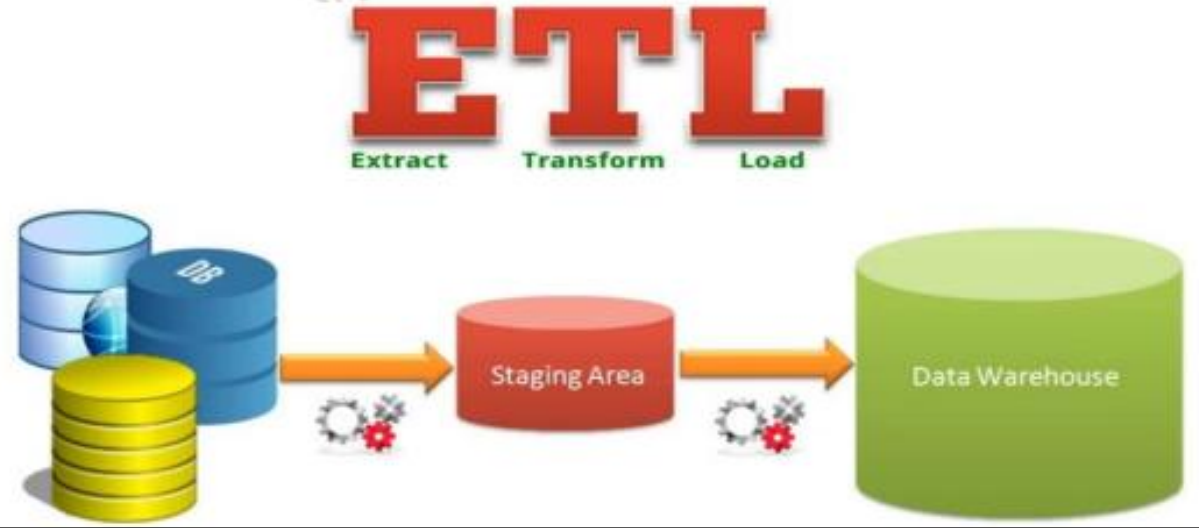
- **Data Sources** -
  - Text files
  - OLTP Databases
  - XML files
  - JSON files
  - Spreadsheets



- **ETL (Extract, Transform, and Load):**
  - **Extraction** - Accessing and extracting the data from the source systems, including database, flat files, spreadsheets, etc.
  - **Transformation** - data cleaning, data normalization, Converting data from source format to Data warehouse format.
  - **Loading -** Loading data to data warehouse and checking for integrity of the data

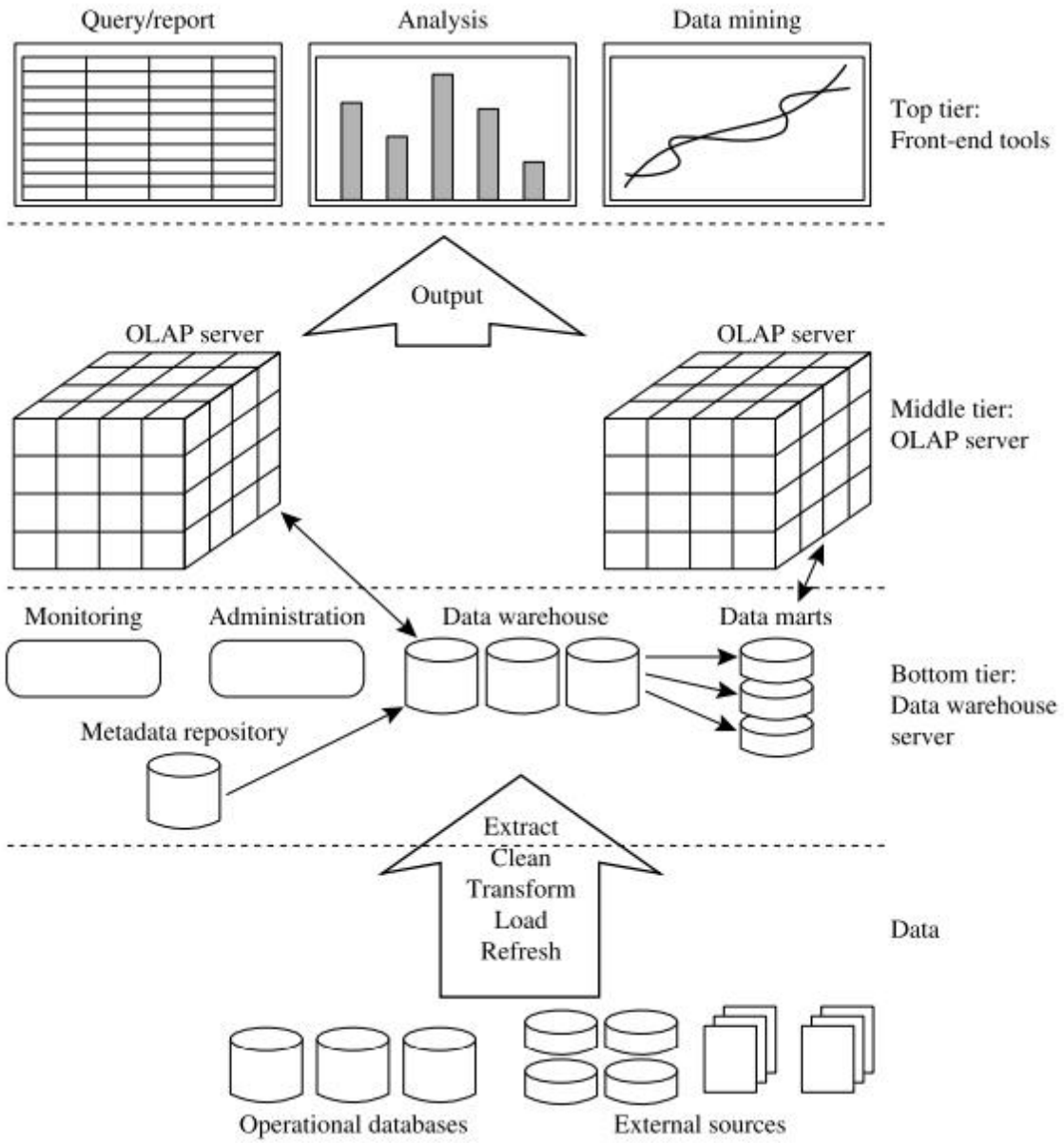
Extraction, Transformation, and Loading (ETL)

- **Data extraction**
  - get data from multiple, heterogeneous, and external sources
- **Data cleaning**
  - detect errors in the data and rectify them when possible
- **Data transformation**
  - convert data from legacy or host format to warehouse format
- **Load**
  - sort, summarize, consolidate, compute views, check integrity, and build indicies and partitions
- **Refresh**
  - propagate the updates from the data sources to the warehouse



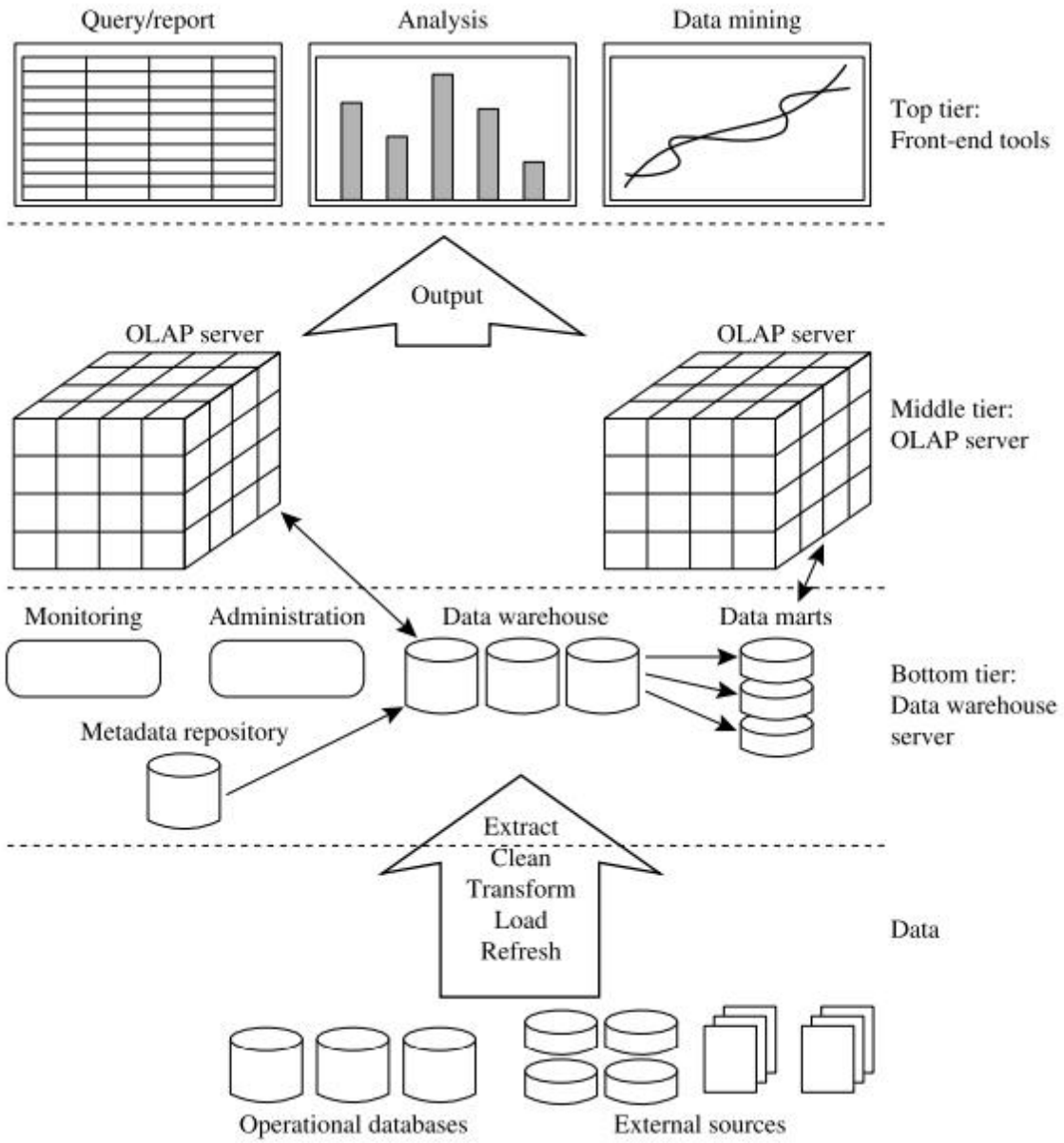
The Three-Tier Architecture

- The middle tier is an **OLAP server** that is typically implemented using -
  - a **relational OLAP(ROLAP)** model which is an extended relational DBMS that maps operations on multidimensional data to standard relational operations.
  - a **multidimensional OLAP (MOLAP)** model which is an special-purpose server that directly implements multidimensional data and operations.



The Three-Tier Architecture

- The top tier is a **front-end client layer** which contains tools for querying, reporting, visualization, analysis and data mining such as trend analysis and prediction.



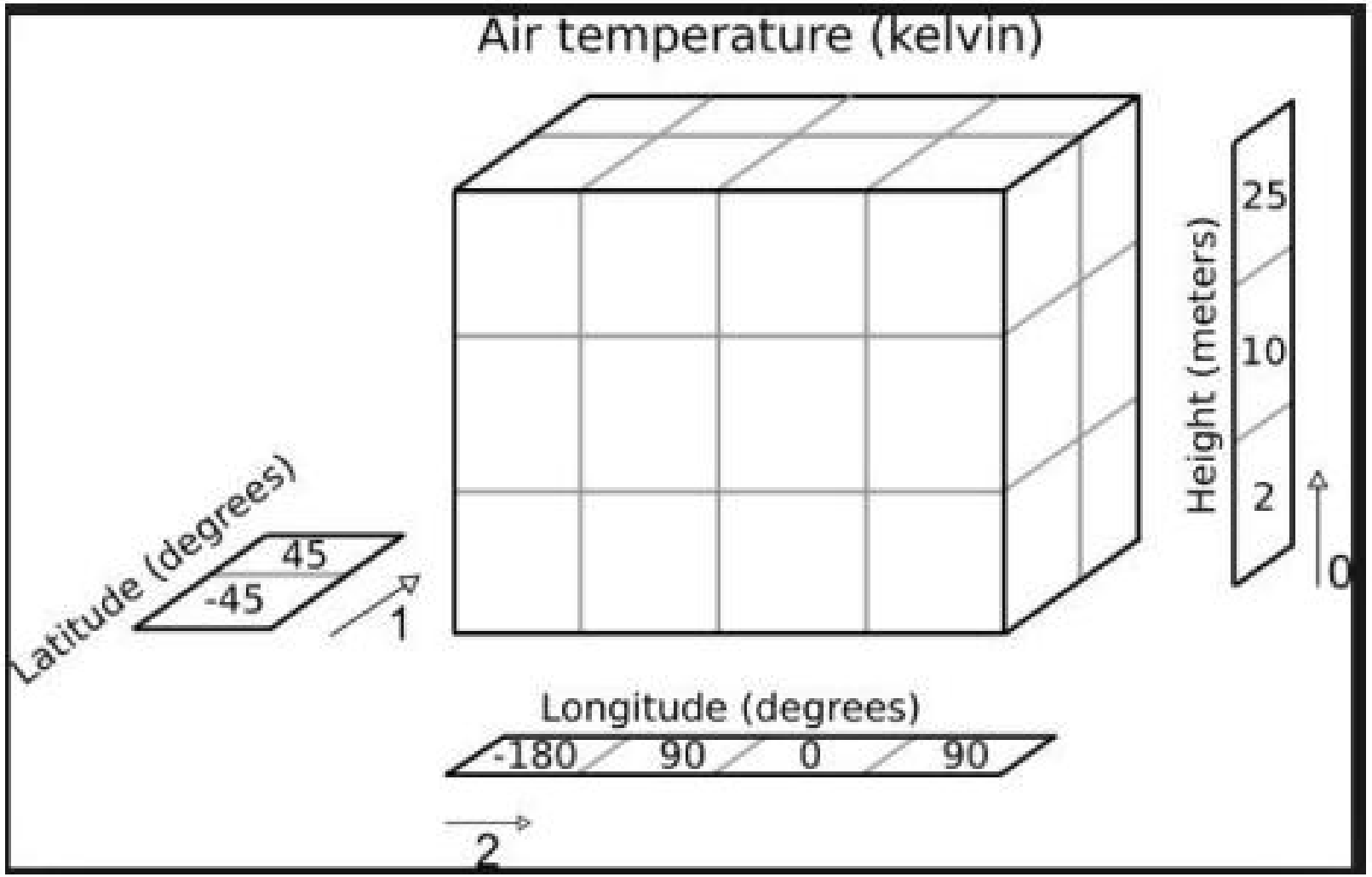
Data Warehouse Models

- **Enterprise Warehouse**
  - An enterprise warehouse collects all information about subjects spanning the entire organization.
  - It typically contains detailed data and summarized data and can range in size from hundreds of gigabytes to terabytes or beyond.
  - It requires extensive business modeling at the enterprise level and may take years to design and build.
- **Data Mart**
  - A data mart contains a subset of corporate-wide data that is of value to a specific group of users, such as those within a business department.
  - For example, a marketing data mart may confine its subjects to customer, item, marketing channel and sales.
  - Two types - *Independent data marts* and *dependent data marts*
- **Virtual Warehouse**
  - A set of views over operational databases.
  - A virtual warehouse is easy to build but puts excess overhead on operational database servers.

MULTIDIMENSIONAL DATA MODEL: DATA CUBE

- Data warehouses and OLAP tools are based on a **multidimensional data model**.
- This model views data in the form of a **data cube**.
- A data cube allows data to be modeled and viewed in **multiple dimensions**.
  - It is defined by *dimensions* and *facts*.
  - Dimensions are the perspectives or entities with respect to which an organization wants to keep records.
    - Each dimension may have a table associated with it, called a dimension table, which further describes the dimension. For example, a dimension table for item may contain the attributes item name, brand, and type.
  - Facts are numerical measures.
    - Examples of facts for a sales data warehouse include dollars sold units sold

Multidimensional Data Model: Data Cube



DATA CUBE: A 2-D DATA CUBE

- Although we usually think of cubes as 3-D geometric structures, in data warehousing the data cube is **n-dimensional**.
- **A 2-D data cube:**
  - **dimensions -** *time* and *item, the measure displayed (fact) is dollars_sold (in thousands)*
  - In this 2-D representation, the sales for Vancouver are shown with respect to the time dimension (organized in quarters) and the item dimension (organized according to the types of items sold).

location = "Vancouver"				
time (quarter)	**item (type)**			
	home entertainment	**computer**	**phone**	**security**
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

Note: The sales are from branches located in the city of Vancouver. The measure displayed is dollars_sold *(in thousands).*

DATA CUBE: A 3-D DATA CUBE

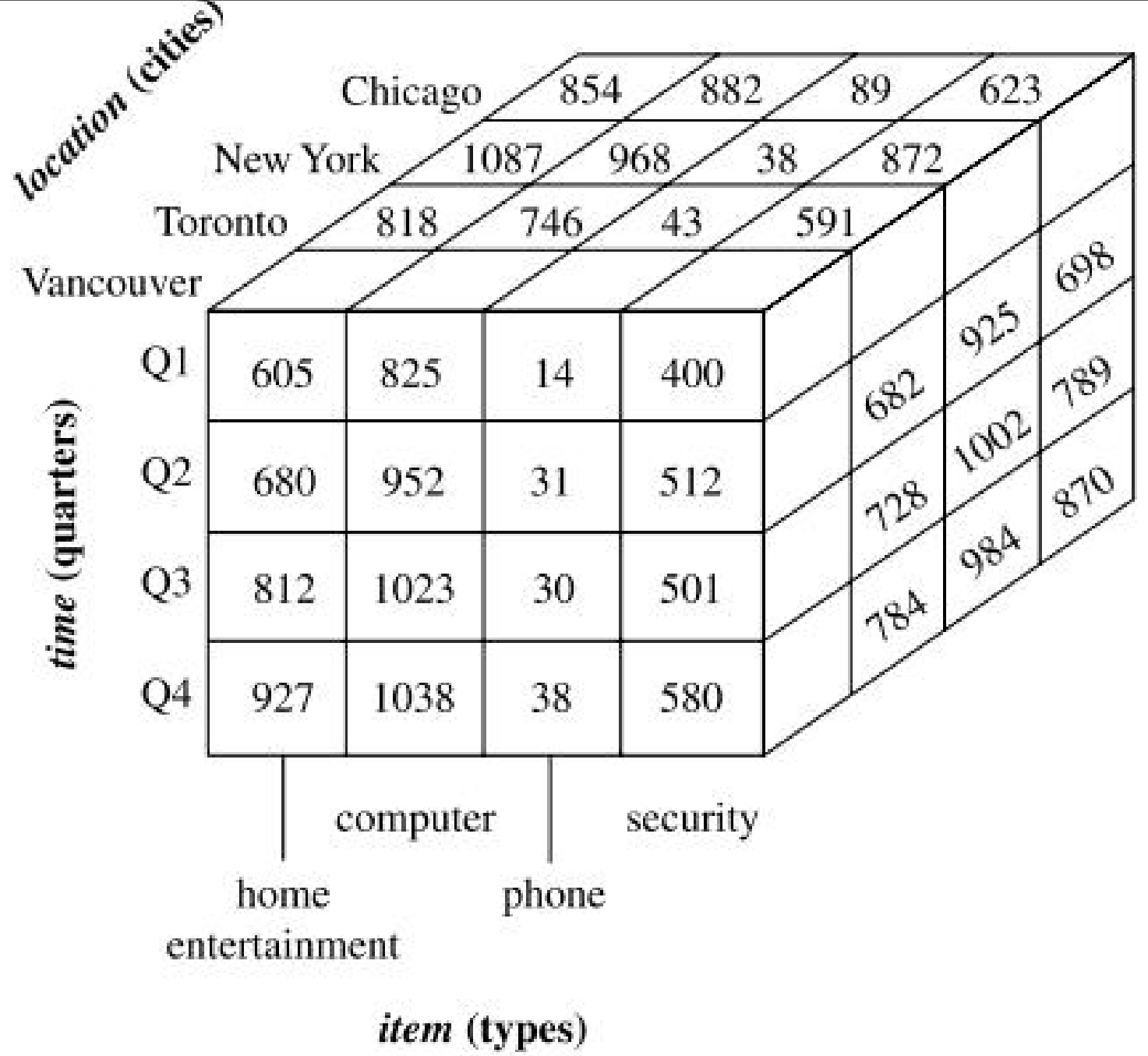
- Now, suppose that we would like to view the sales data with a third dimension. For instance, suppose we would like to view the data according to time and item, as well as location, for the cities Chicago, New York, Toronto, and Vancouver.

	location = "Chicago"				**location** = "New York"				**location** = "Toronto"				**location** = "Vancouver"			
time	**item**				**item**				**item**				**item**			
	home ent.	**comp.**	**phone**	**sec.**	**home ent.**	**comp.**	**phone**	**sec.**	**home ent.**	**comp.**	**phone**	**sec.**	**home ent.**	**comp.**	**phone**	**sec.**
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

Note: The measure displayed is dollars_sold *(in thousands).*

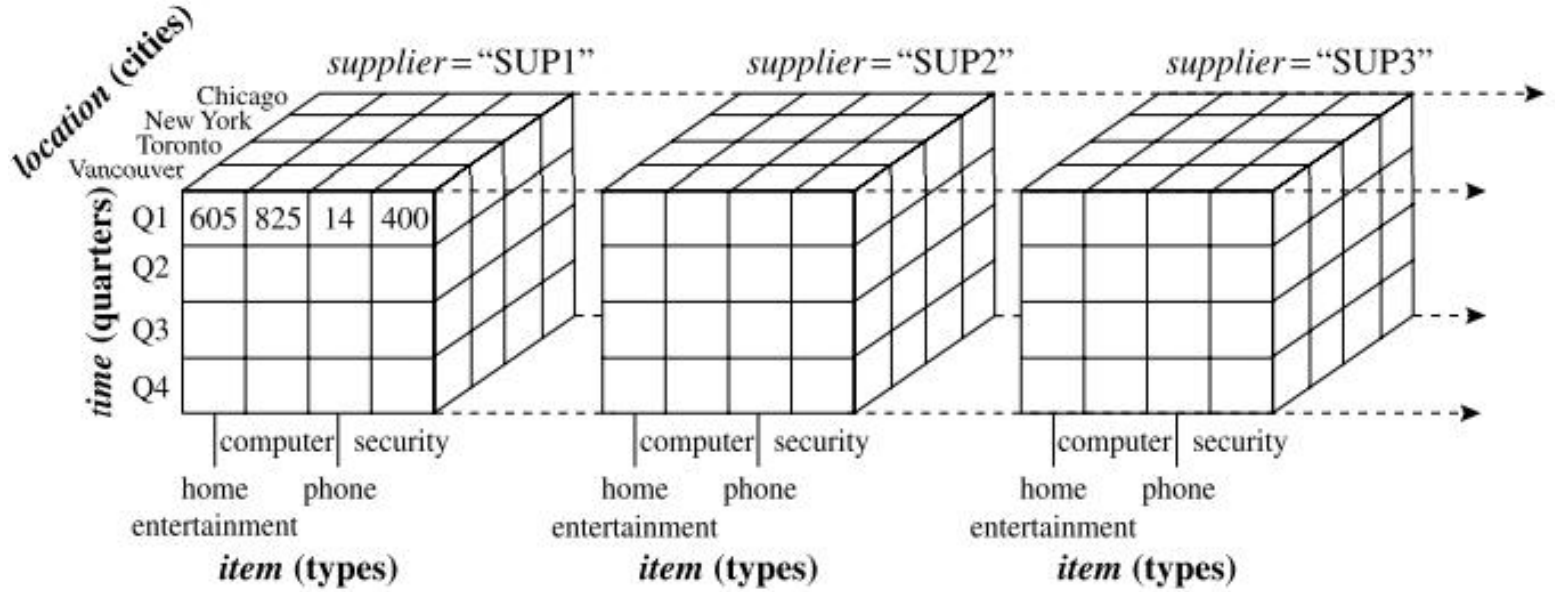
DATA CUBE: A 3-D DATA CUBE

This representation is also called **Cuboid.**



DATA CUBE: A 4-D CUBE

- Suppose that we would now like to view our sales data with an additional fourth dimension, say supplier.
- Visualizing things in 4-D becomes tricky. However, we can think of a 4-D cube as a series of 3-D cubes.



Data Cubes

- The data cube is a metaphor for multidimensional data storage. The actual physical storage of such data may differ from its logical representation.
- In SQL terms, these aggregations are referred to as *group-by*'s. Each *group-by* can be represented by a cuboid.
- Given a set of dimensions, we can generate a cuboid for each of the possible subsets of the given dimensions, including the empty set.
- The result would form a lattice of cuboids, each showing the data at a different level of summarization, or group-by. The lattice of cuboids is then referred to as a data cube.

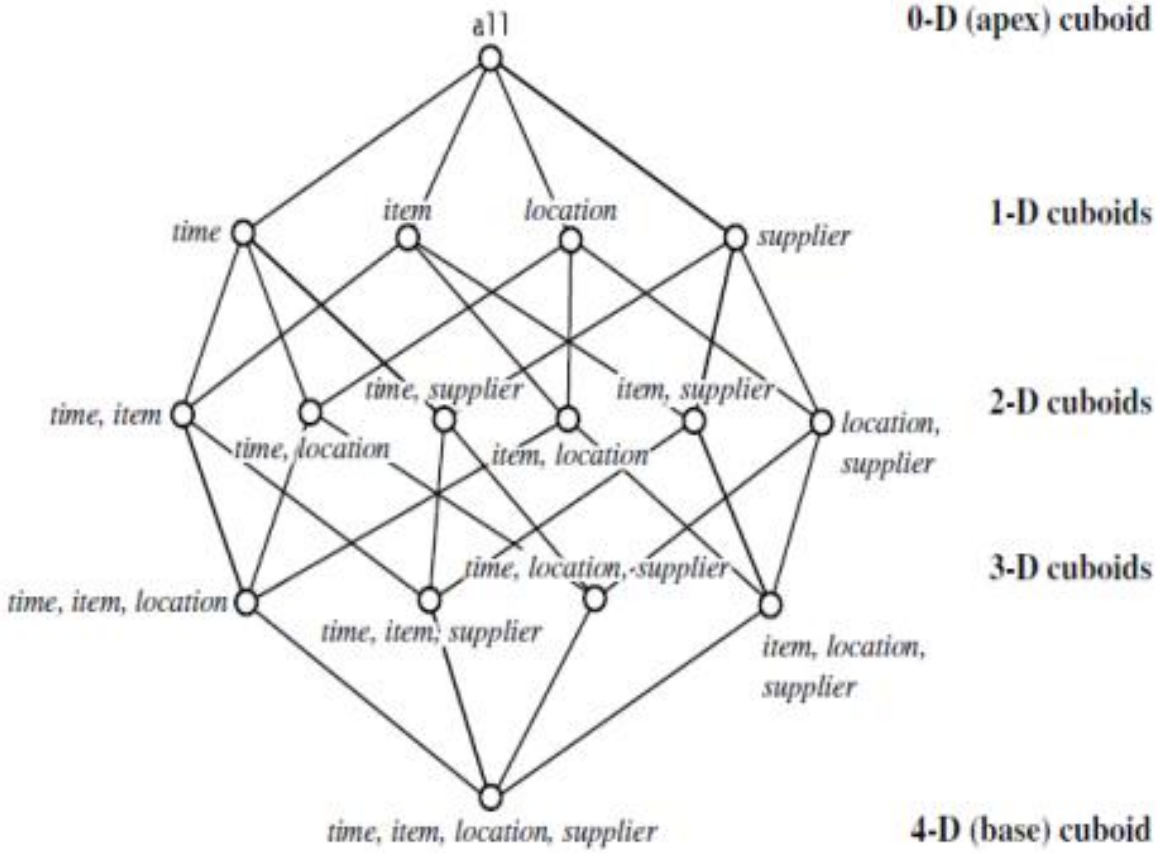
Table: Details

Name	Age	Location	ID
John	35	California	12698
Harry	24	Los Angeles	12699
Smith	32	California	12700
Gary	45	New Jersey	12701

Location	number
California	2
Los Angeles	1
New Jersey	1

Data Cube: A Lattice of Cuboids

- A lattice of cuboids, making up a 4-D data cube for the dimensions *time*, *item*, *location*, and *supplier*.
  - Each cuboid represents a different degree of summarization.



- **0-D cuboid** which holds highest level of summarization, is called **apex cuboid**.
  - This is the total sales summarized over all four dimensions.
  - The **apex cuboid** is typically denoted by **all**.
- A **3-D (nonbase) cuboid** for ***time***, ***item***, ***location***, summarized for all suppliers.
- The cuboid that holds the lowest level of summarization is called the **base cuboid.**
  - base cuboid for ***time***, ***item***, ***location***, and ***supplier*** dimensions

Schemas for Multidimensional Data Models

- The entity-relationship (ER) data model is popular in relational databases where a database schema consists of a set of entities and the relationships among them.
- An online data analysis often has to scan a lot of data. To support online data analysis, a data warehouse requires a concise, subject-oriented schema that facilitates scanning a large amount of data efficiently.

Star Schema

- The most common paradigm of multidimensional model is **star schema** in which a data warehouse contains -
  - a large central table (fact table) containing the bulk of the data with no redundancy.
  - a set of smaller attendant tables (dimension tables), one for each dimension.
- The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table.
- Notice that in the star schema, each dimension is represented by only one table, and each table contains a set of attributes.

Fact Table

- A fact table is the central table in a data warehouse schema that stores quantitative data for analysis. This table contains facts (measurable or quantitative data), such as sales amounts, quantities, or profits, which are typically numeric values used for decision-making.

Sales Fact Table:

SaleID	CustomerID	ProductID	DateID	QuantitySold	Revenue
1	1001	2001	3001	5	500
2	1002	2002	3002	3	300

- **Measures:** `QuantitySold`, `Revenue`
- **Foreign Keys:** `CustomerID`, `ProductID`, `DateID`

Dimensional Table

- A dimension table is a supporting table in a data warehouse schema that contains the descriptive attributes (metadata) that provide context to the facts. Dimension tables describe the who, what, when, where, and how of the data.

Customer Dimension Table:

CustomerID	CustomerName	CustomerCity	CustomerAge
1001	John Doe	New York	30
1002	Jane Smith	San Francisco	27

- **Descriptive Data:** `CustomerName`, `CustomerCity`, `CustomerAge`
- **Primary Key:** `CustomerID` (used to join with the fact table)

Fact Table:

SaleID	CustomerID	ProductID	DateID	QuantitySold	Revenue
1	1001	2001	3001	5	500

Customer Dimension Table:

CustomerID	CustomerName	CustomerCity
1001	John Doe	New York

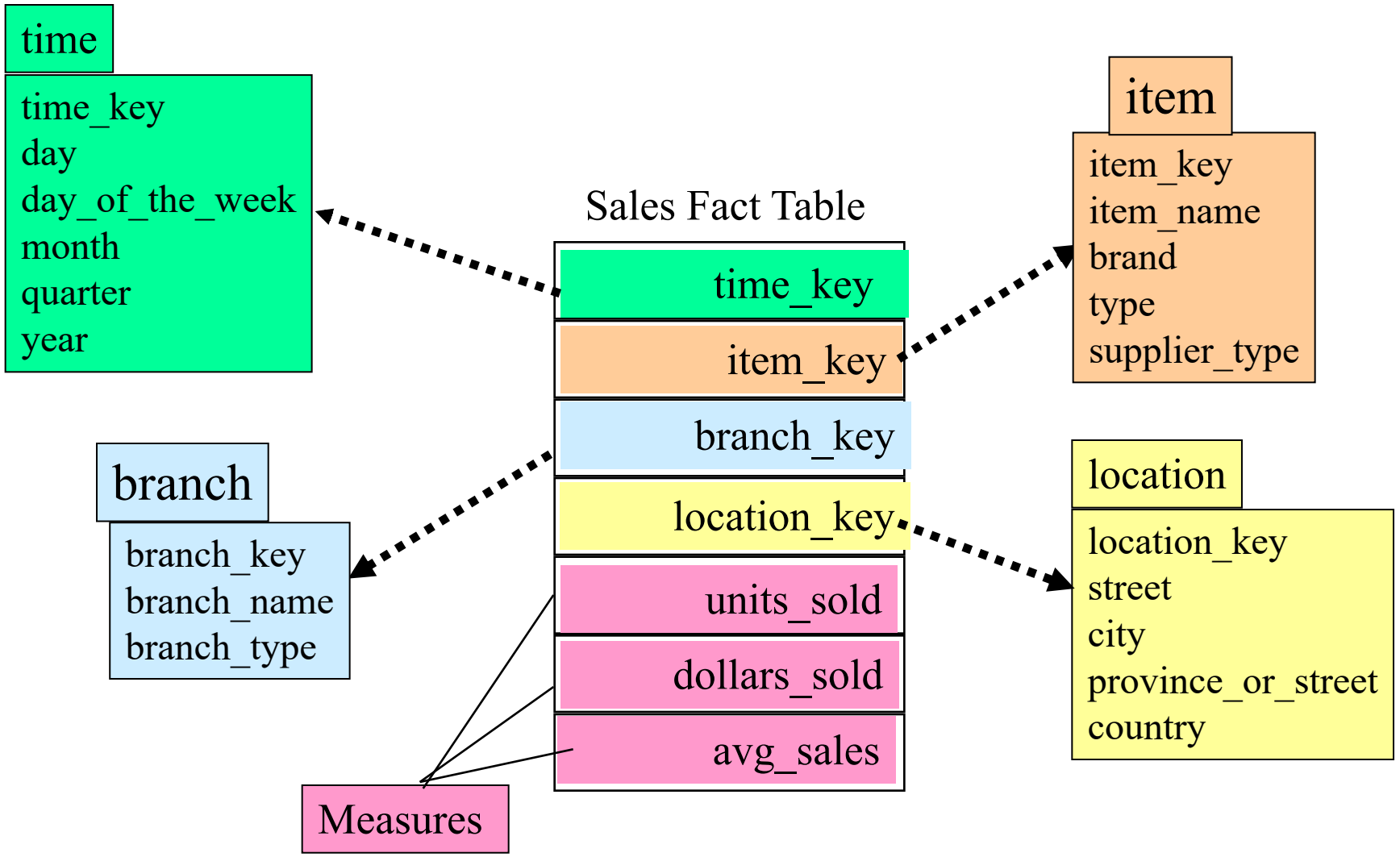
Product Dimension Table:

ProductID	ProductName	ProductCategory
2001	Laptop	Electronics

Date Dimension Table:

DateID	Day	Month	Year
3001	01	Jan	2024

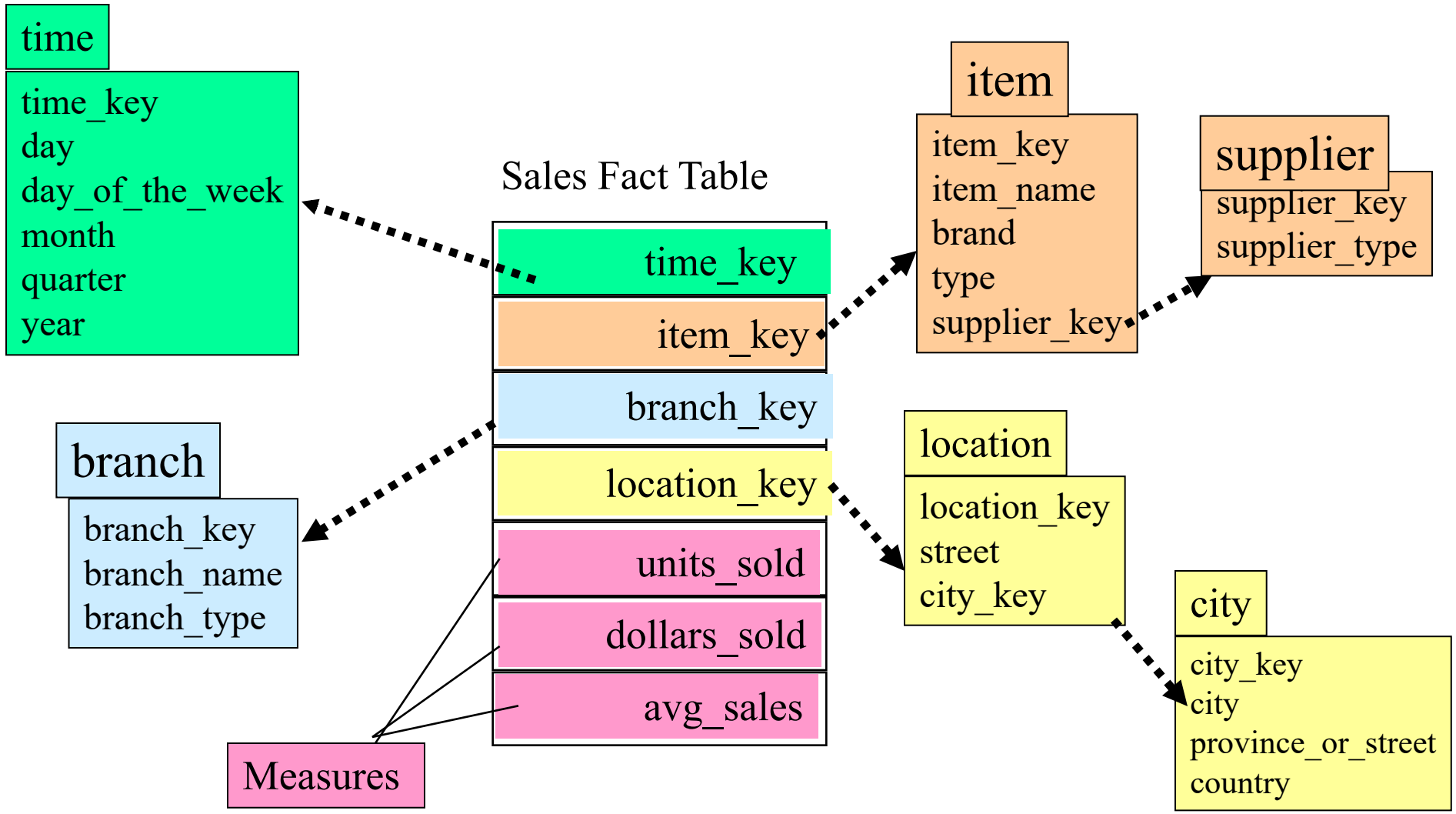
Example of Star Schema



Snowflake Schema

- Snowflake schema is a variant of star schema, where some dimension tables are normalized, thereby further splitting the data into additional tables. The resulting schema graph forms a shape similar to a snowflake.
- The major difference between the snowflake schema and star schema models is that the dimension tables of the snowflake model may be kept in normalized form to reduce redundancies. Such a table is easy to maintain and saves storage space. However, this space savings is negligible in comparison to the typical magnitude of the fact table.

Example of Snowflake Schema



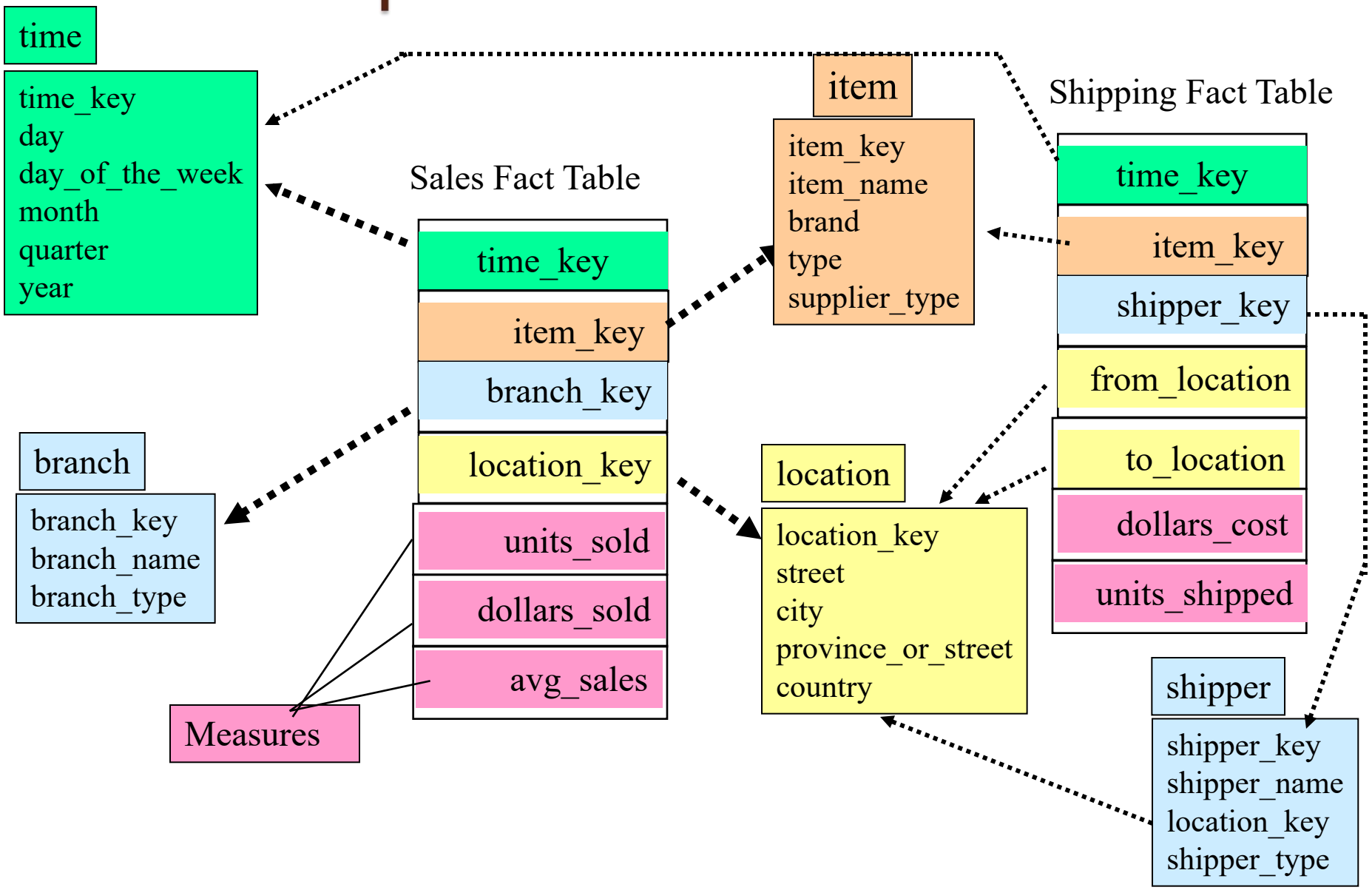
Snowflake Schema

- The snowflake structure may reduce the effectiveness of browsing, since more joins are needed to execute a query.
- The system performance may be adversely impacted. Hence, although the snowflake schema reduces redundancy, it is not as popular as the star schema in data warehouse design.

Fact Constellation Schema or Galaxy Schema

- Sophisticated applications may require multiple fact tables to share dimension tables.
- This kind of schema can be viewed as a collection of stars and hence is called a galaxy schema or a fact constellation.

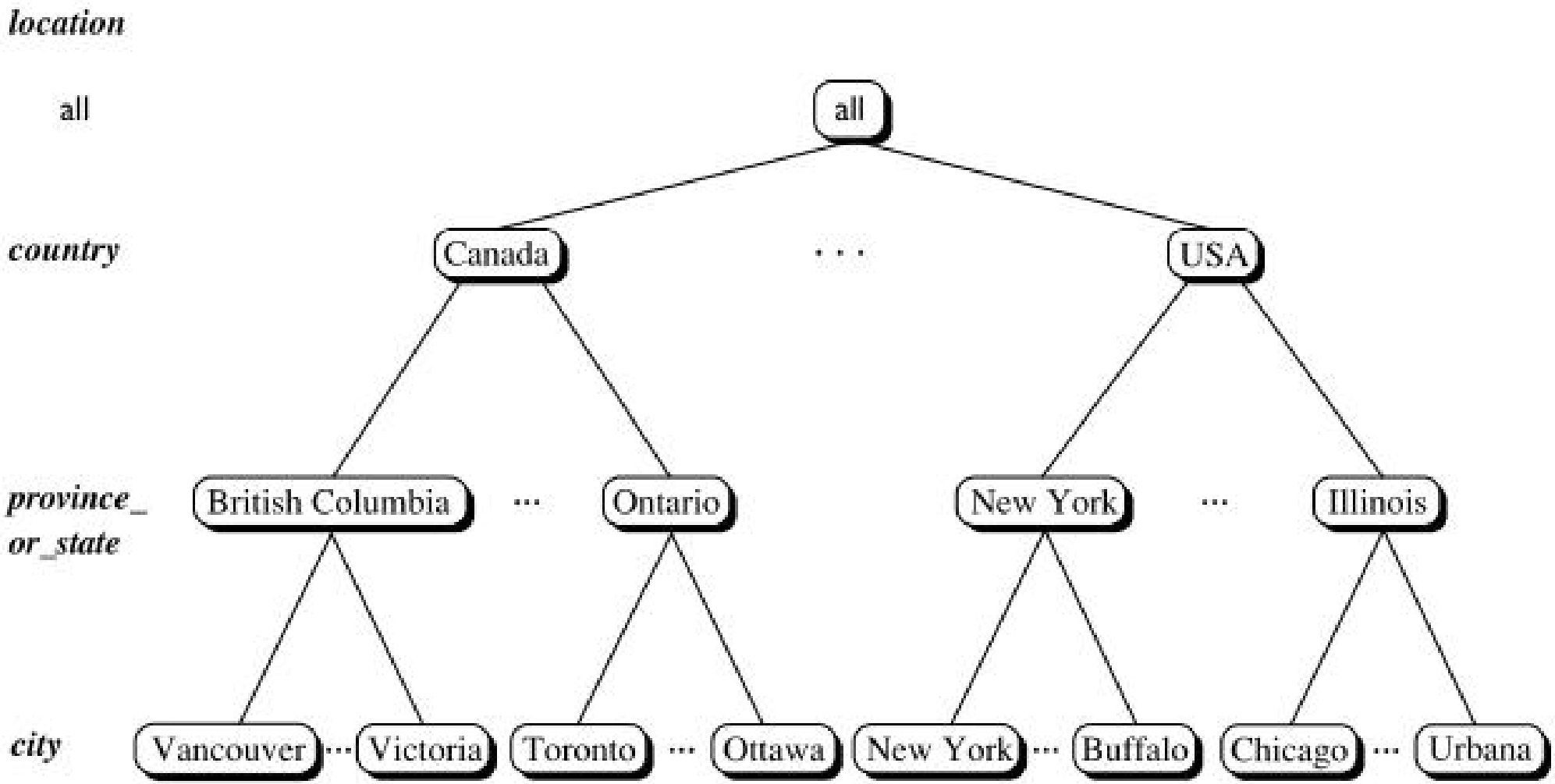
Example of Fact Constellation



Concept Hierarchies

- Dimensions define concept hierarchies.
- A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts.
- Consider a concept hierarchy for the dimension location. City values for location include Vancouver, Toronto, New York, and Chicago. Each city, however, can be mapped to the province or state to which it belongs. For example, Vancouver can be mapped to British Columbia and Chicago to Illinois. The provinces and states can in turn be mapped to the country (e.g., Canada or the United States) to which they belong. These mappings form a concept hierarchy for the dimension location, mapping a set of low-level concepts (i.e., cities) to higher-level, more general concepts (i.e., countries).

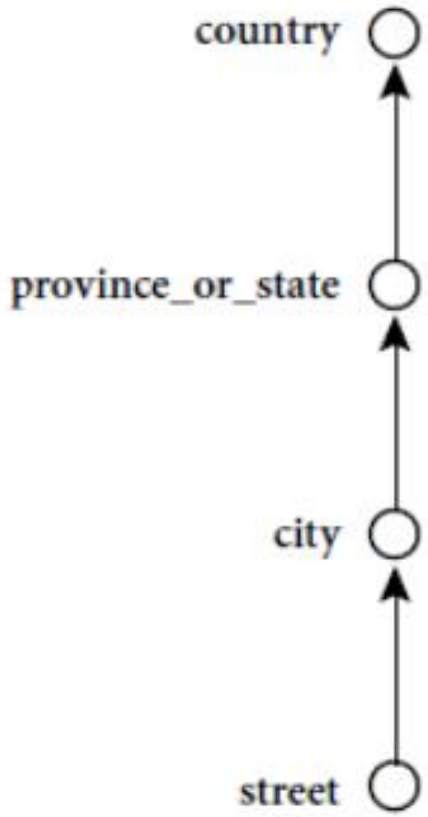
Concept Hierarchy of Dimension Location



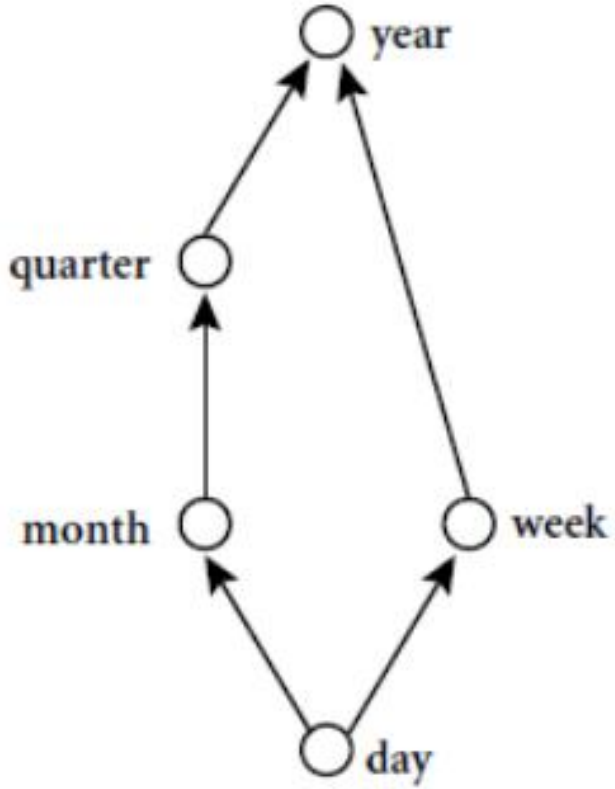
Concept Hierarchies

- Many concept hierarchies are implicit within the database schema.
- For example, suppose that the dimension location is described by the attributes *street, city, province_or_state, zip_code, and country*. These attributes are related by a total order, forming a concept hierarchy such as "*street < city < province_or_state < country*."
- Alternatively, the attributes of a dimension may be organized in a partial order, forming an acyclic directed graph. An example of a partial order for the time dimension based on the attributes *day, week, month, quarter, and year* is "*day < {month < quarter; week} < year*."

Concept Hierarchies



a hierarchy for *location*

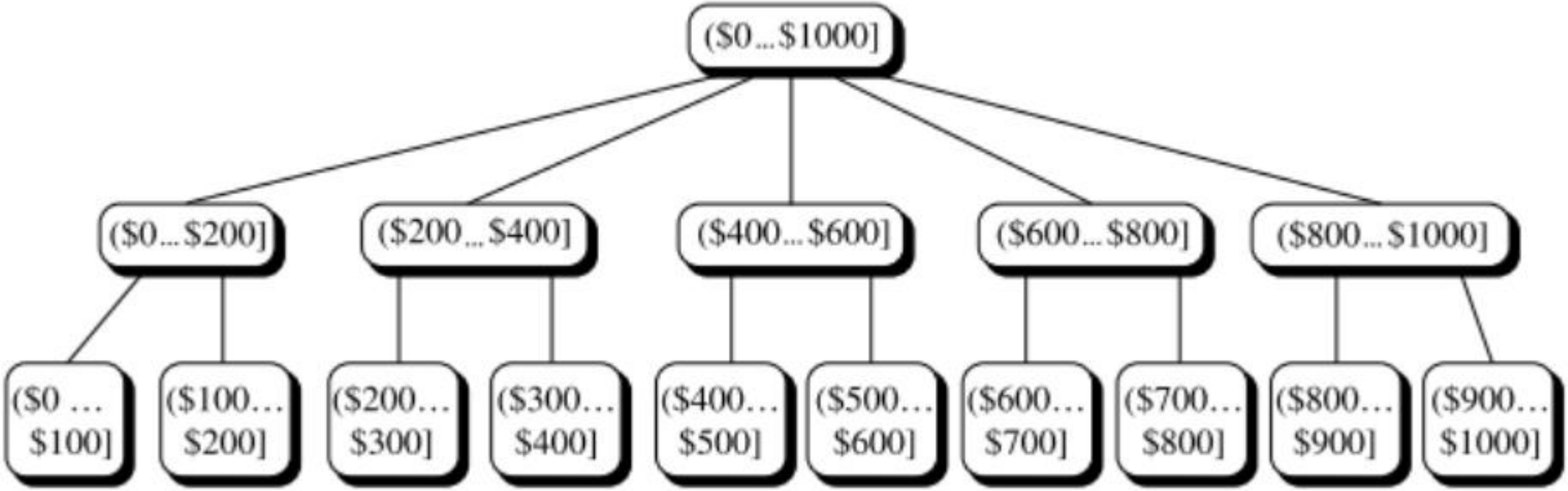


a lattice for *time*

Concept Hierarchies

- A concept hierarchy that is a total or partial order among attributes in a database schema is called a **schema hierarchy**.
- Concept hierarchies may also be defined by discretizing or grouping values for a given dimension or attribute, resulting in a **set-grouping hierarchy**. A total or partial order can be defined among groups of values.
- There may be more than one concept hierarchy for a given attribute or dimension, based on different user viewpoints. For instance, a user may prefer to organize price by defining ranges for inexpensive, moderately_priced, and expensive.
- Concept hierarchies may be provided manually by system users, domain experts or knowledge engineers, or may be automatically generated based on statistical analysis of the data distribution.
- Concept hierarchies allow data to be handled at varying levels of abstraction.

CONCEPT HIERARCHIES



OLAP Operations

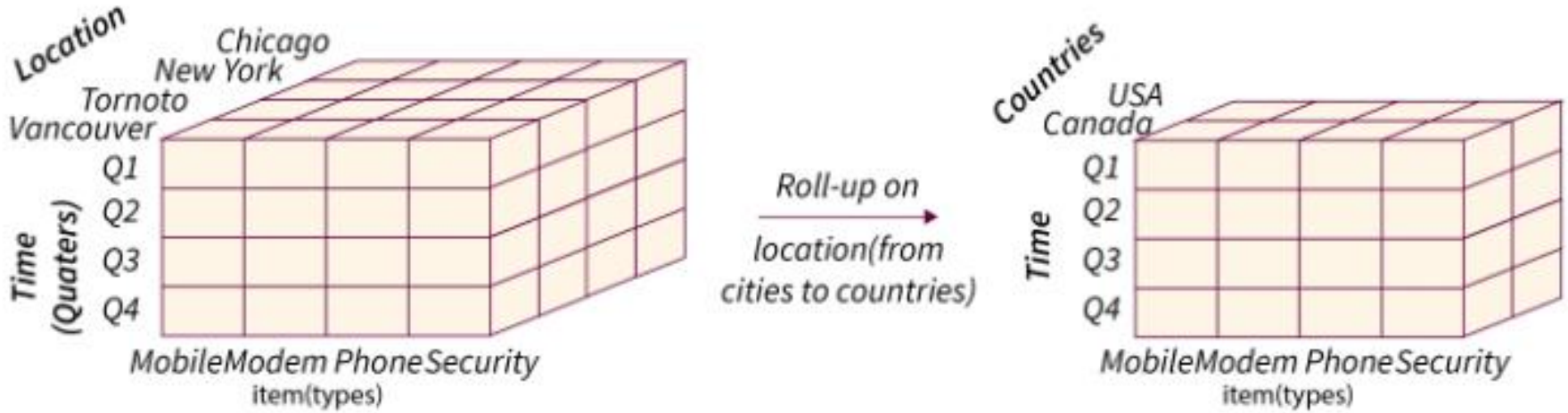
- How are concept hierarchies useful in OLAP?
- In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies.
- This organization provides users with the flexibility to view data from different perspectives.
- A number of OLAP data cube operations exist to materialize these different views, allowing interactive querying and analysis of the data at hand.

Typical OLAP Operations

- **Roll up (drill-up):** summarize data
  - by climbing up hierarchy or by dimension reduction
- **Drill down (roll down):** reverse of roll-up
  - from higher level summary to lower level summary or detailed data, or introducing new dimensions
- **Slice and dice:** project and select
- **Pivot (rotate):**
  - reorient the cube, visualization, 3D to series of 2D planes

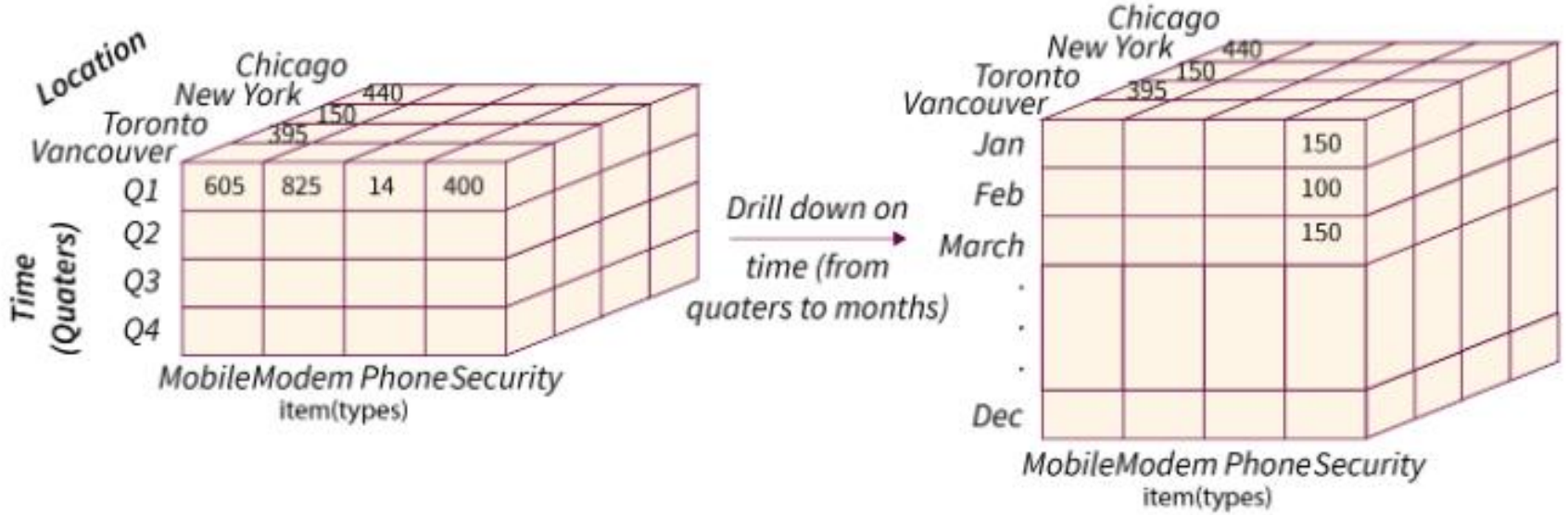
OLAP Operation: Roll-up / Drill-up

- The roll-up operation (also called the drill-up operation) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction.
- The Drill Up OLAP operation allows users to view data at a higher level of aggregation by collapsing a specific dimension in a multidimensional cube.
- By drilling up, users can identify trends and patterns that may not be visible at lower levels of detail, enabling them to make informed decisions based on a complete picture of the data.



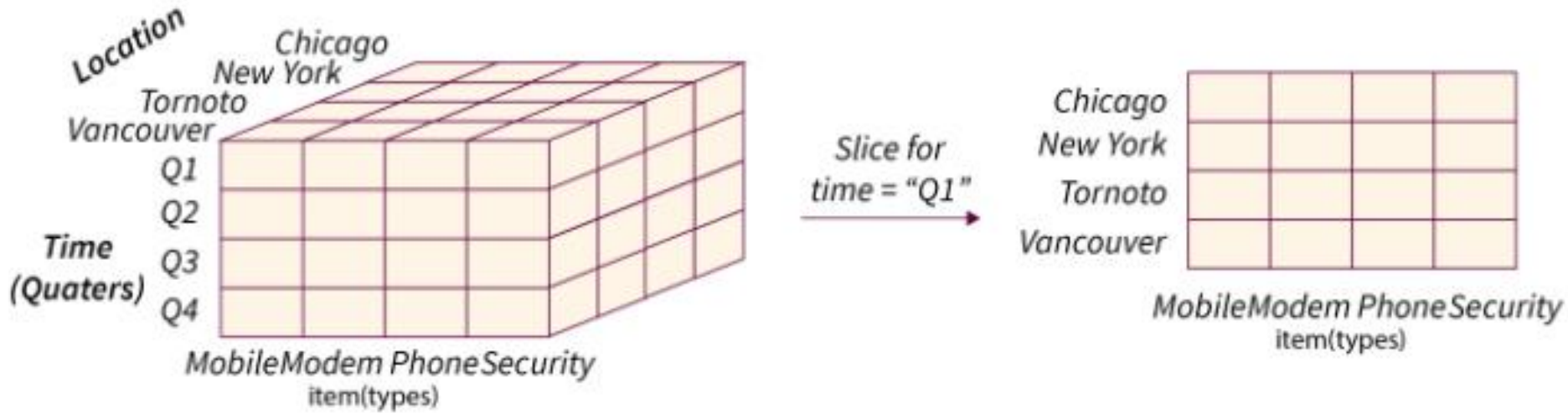
OLAP Operations: Drill-Down

- Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data.
- Drill-down can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions.
- The Drill Down OLAP operation allows users to view more detailed data by expanding a particular dimension in a multidimensional cube.
- By drilling down into the data, users can identify trends and patterns that may not be apparent at higher levels of aggregation, allowing for more targeted analysis and decision-making.



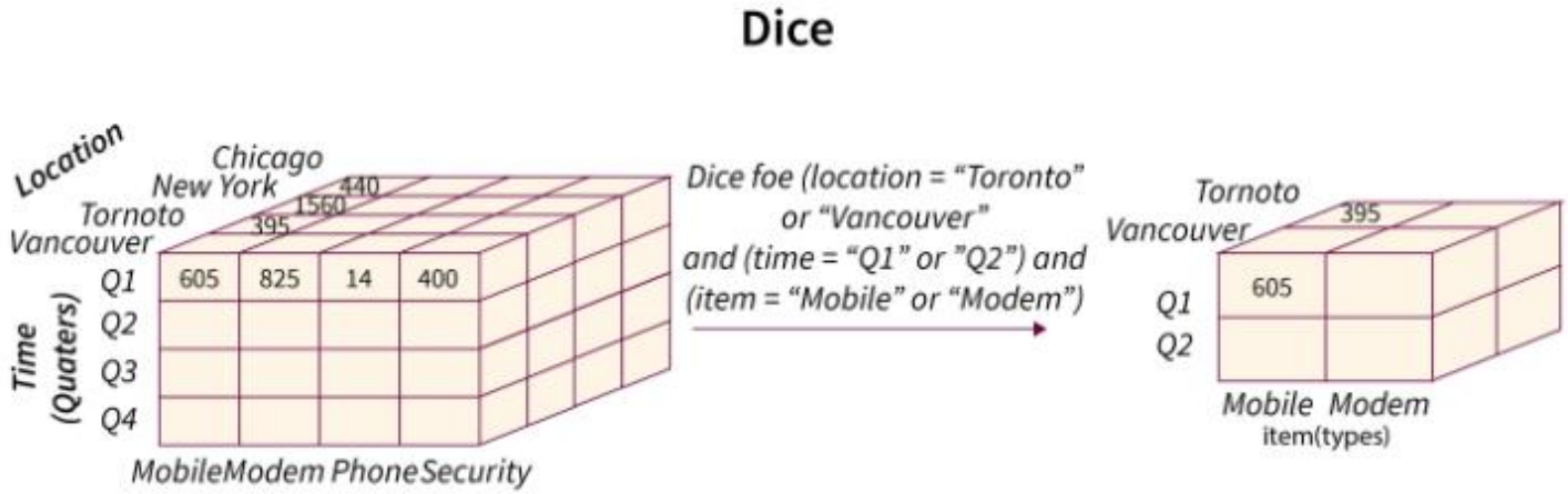
OLAP Operations: Slice

- The slice operation performs a selection on one dimension of the given cube, resulting in a subcube.
- The Slice OLAP operation allows users to extract data from a multidimensional cube by selecting a single dimension and a specific value for that dimension.
- This operation is important in data mining, enabling users to extract data based on specific criteria and allowing for more focused and targeted analysis.



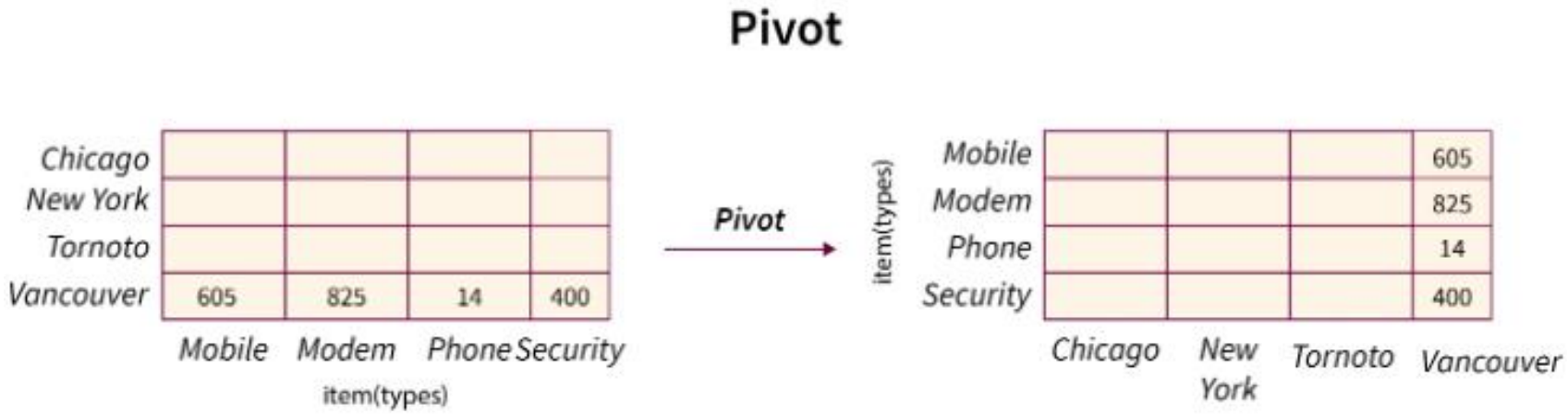
OLAP Operations: Dice

- The dice operation defines a subcube by performing a selection on two or more dimensions.
- The Dice OLAP operation allows users to extract data from a multidimensional cube by selecting multiple dimensions and specific values for each selected dimension.
- This operation is important in data mining, enabling users to extract data based on multiple criteria, allowing for more focused and targeted analysis.



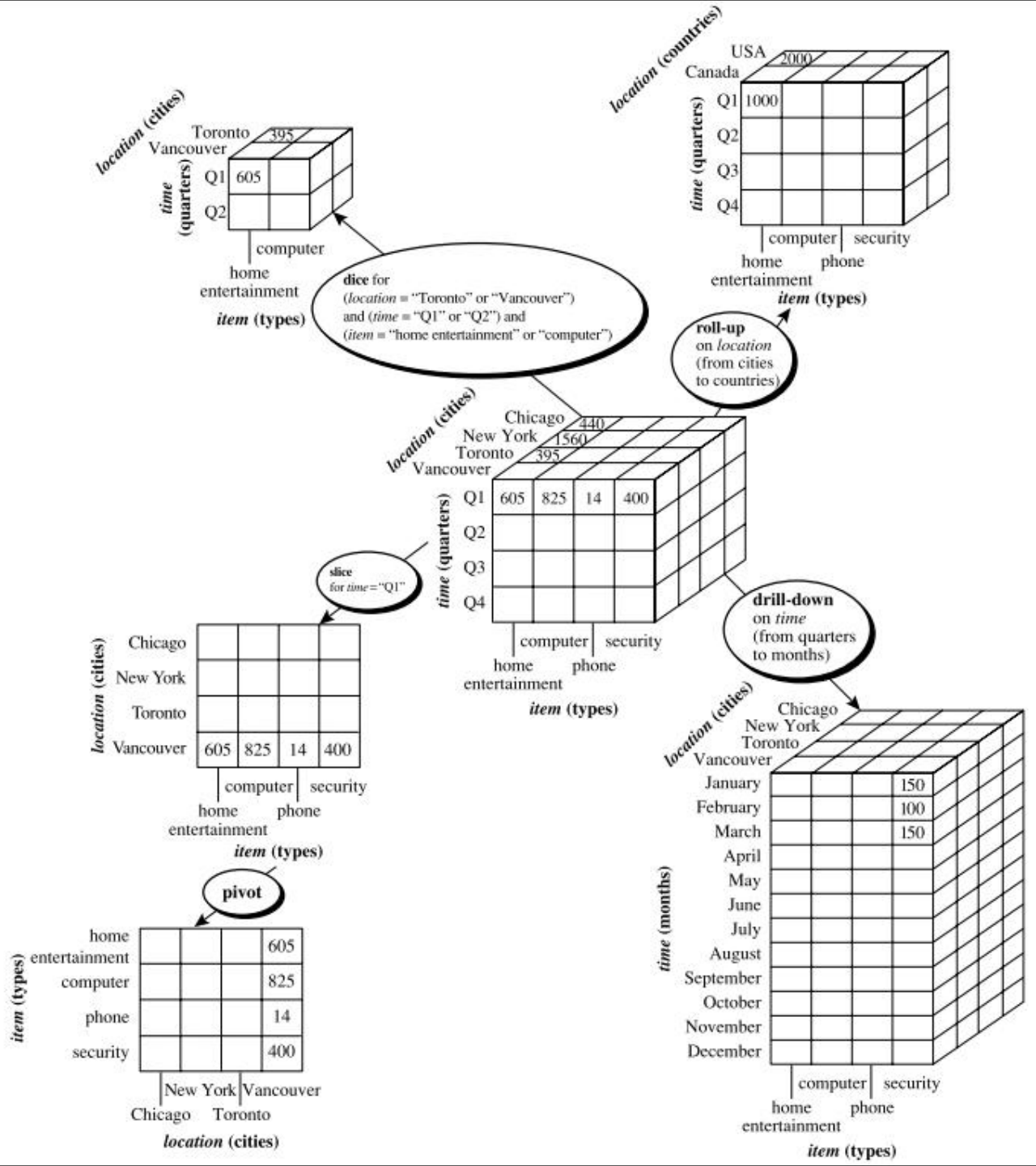
OLAP Operation: Pivot

- Pivot (rotate) is a visualization operation that rotates the data axes in view in order to provide an alternative presentation of the data.
- The Pivot OLAP operation allows users to rotate the orientation of a multidimensional cube to view the data from a different perspective.
- This operation is important in data mining as it enables users to view the same data from different angles and gain new insights into the patterns and relationships in the data.



Other OLAP Operations

- **Drill Across**
  - The Drill Across OLAP operation allows users to view data across multiple dimensions in a multidimensional cube.
  - For example, if a user wants to analyze data for a particular product category and a specific geographic region, they can drill across the cube by selecting the product category and the geographic region dimensions.
  - It involves more than one fact table.
- **Drill Through**
  - The Drill Through OLAP operation allows users to access detailed information about a specific data point in a multidimensional cube.
  - For example, if a user wants to view the sales data for a particular product in a specific region and period, they can drill through the cube to view the underlying transactional data for that product, region, and time period.
  - The drill-through operation uses relational SQL facilities to drill through the bottom level of a data cube down to its back-end relational tables.


location (countries)
USA
Canada
2000
Q1
1000
Q2
Q3
Q4
time (quarters)
computer
security
home entertainment
phone
item (types)
location (cities)
Toronto
Vancouver
395
Q1
605
Q2
time (quarters)
computer
home entertainment
item (types)
dice for
(location = "Toronto" or "Vancouver")
and (time = "Q1" or "Q2") and
(item = "home entertainment" or "computer")
roll-up
on location
(from cities
to countries)
location (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
Q1
605
825
14
400
Q2
Q3
Q4
time (quarters)
computer
security
home entertainment
phone
item (types)
slice
for time = "Q1"
drill-down
on time
(from quarters
to months)
location (cities)
Chicago
New York
Toronto
Vancouver
605
825
14
400
computer
security
home entertainment
phone
item (types)
location (cities)
Chicago
New York
Toronto
Vancouver
January
February
March
April
May
June
July
August
September
October
November
December
150
100
150
time (months)
computer
security
home entertainment
phone
item (types)
pivot
item (types)
home entertainment
computer
phone
security
605
825
14
400
New York
Vancouver
Chicago
Toronto
location (cities)

Measures

- A measure in a data cube is a numeric function that can be evaluated at each point in the data cube space.
- A measure value is computed for a given point by aggregating the data corresponding to the respective dimension–value pairs defining the given point.
- For example, the measure *total-sales* for the cell *time = "Q1," location = "Vancouver," item = "computer"* is computed by summing up all the amounts happened in Q1, at the branch of Vancouver, and about computers from the fact table.
- Based on what kind of aggregate functions used, measures can be organized into three categories -
  - Distributive
  - Algebraic
  - Holistic

Distributive Measures

- Measures where the result can be calculated in a distributed manner (i.e., computed by breaking down the data into subsets, calculating the measure for each subset, and then combining the results).
- Distributive measures allow large datasets to be broken into smaller chunks for distributed processing in parallel computing systems.
- For example - count(), sum(), min(), max(), etc.

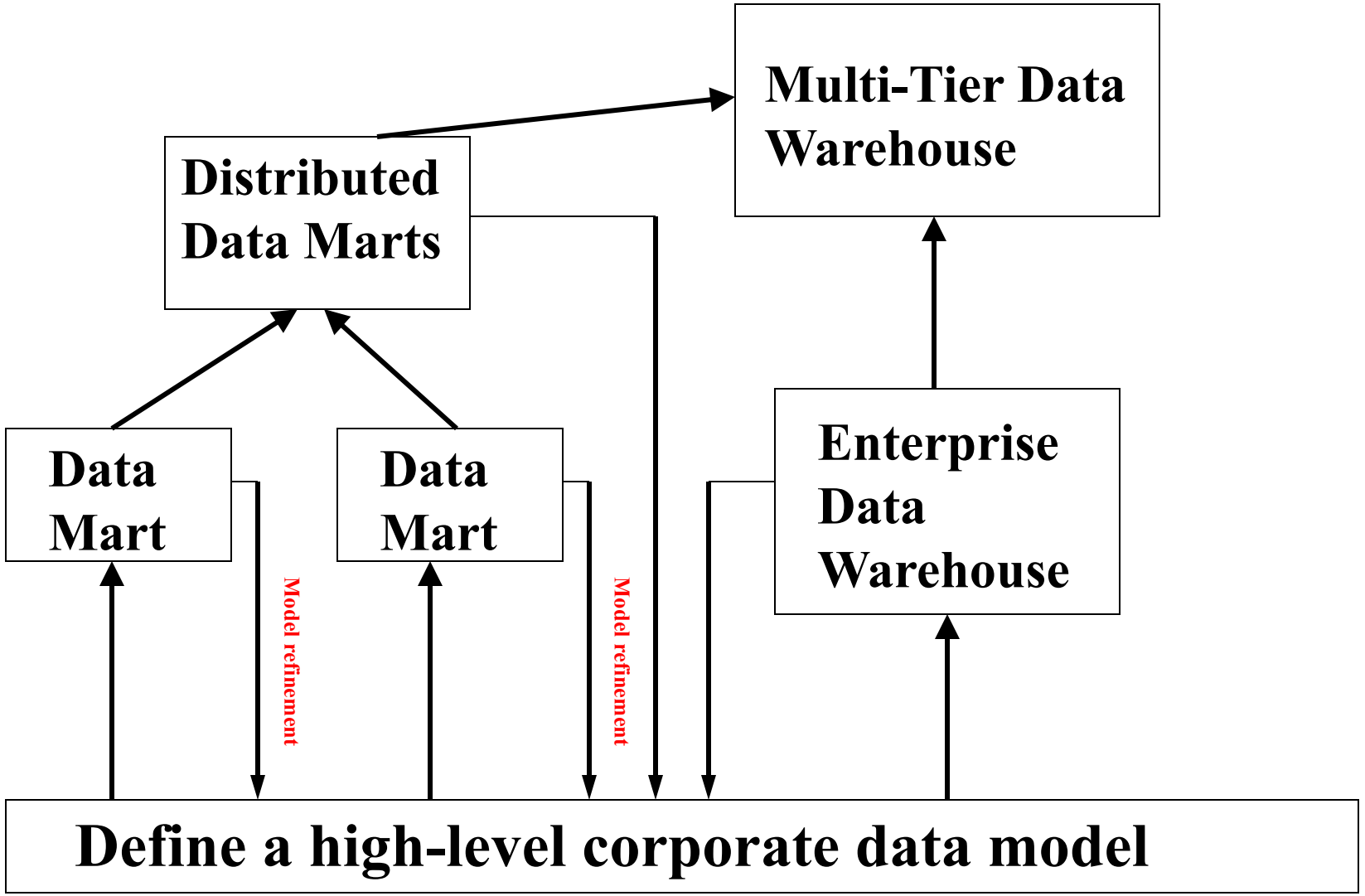
Algebraic Measures

- An aggregate function is ***algebraic*** if it can be computed by an algebraic function with M arguments (where M is a fixed positive integer), each of which is obtained by applying a distributive aggregate function.
- For example, *avg() (average)* can be computed by *sum()/count()* with two arguments, where both sum() and count() are distributive aggregate functions.
- A measure is algebraic if it is obtained by applying an algebraic aggregate function.
- Examples - avg(), standard_deviation(), etc.
- These measures are useful for computing ratios and averages that are frequently used in business reports.

Holistic Measures

- Measures that require access to all the data to compute and cannot be broken down into smaller subsets.
- Example - Median, mode, rank, percentile.
- These measures are typically used in situations where full data access is required to compute the result, such as calculating the 90th percentile in customer satisfaction scores.
- Most large data cube applications require efficient and scalable computation, and thus distributive and algebraic measures are often used.
- Many efficient techniques for computing data cubes using distributive and algebraic measures exist.
- In contrast, it is difficult to compute holistic measures efficiently. Efficient techniques to approximate the computation of some holistic measures, however, do exist.

Data Warehouse Development: A Recommended Approach



Data Warehouse Usage

- Three kinds of data warehouse applications
  - **Information processing**
    - supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs
  - **Analytical processing**
    - multidimensional analysis of data warehouse data
    - supports basic OLAP operations, slice-dice, drilling, pivoting
  - **Data mining**
    - knowledge discovery from hidden patterns
    - supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools

Data Warehouse and OLAP: Summary

- A **data warehouse** is a *subject-oriented, integrated, time-variant*, and *nonvolatile* collection of data organized in support of management decision making.
  - Several factors distinguish data warehouses from operational databases.
  - Because the two systems provide quite different functionalities and require different kinds of data, it is necessary to maintain data warehouses separately from operational databases.
- A ***multidimensional data model*** is typically used for the design of corporate *data warehouses*.
  - A multidimensional data model can adopt a *star schema*, *snowflake schema*, or *fact constellation schema*.
  - The core of the *multidimensional model* is the data cube, which consists of a large set of *facts* (or *measures*) and a number of *dimensions*.
  - Dimensions are the entities or perspectives with respect to which an organization wants to keep records and are hierarchical in nature.
- A **data cube** consists of a lattice of cuboids, each corresponding to a different degree of summarization of the given multidimensional data.

Data Warehouse and OLAP: Summary

- **Concept hierarchies** organize the values of dimensions into gradual levels of abstraction.
- **On-line analytical processing (OLAP)** can be performed in data warehouses using the multidimensional data model.
  - Typical OLAP operations include roll-up, drill-down,, slice-and-dice, pivot (rotate), as well as statistical operations such as ranking and computing moving averages and growth rates.
- Data warehouses often adopt **a three-tier architecture**.
  - The bottom tier is a warehouse database server, which is a relational database system.
  - The middle tier is an OLAP server, and
  - The top tier is a client, containing query and reporting tools.
- A data warehouse contains **back-end tools and utilities** for populating and refreshing the warehouse.
  - data extraction, data cleaning, data transformation, loading, refreshing, and warehouse management.
- Data warehouse **metadata** are data defining the warehouse objects.
  - A metadata repository provides details regarding the warehouse structure, data history, the algorithms used for summarization, mappings from the source data to warehouse form.

A Data Mining Query Language, DMQL: Language Primitives

- Cube Definition (Fact Table)

  define cube <cube_name> [<dimension_list>]:
  <measure_list>

- Dimension Definition (Dimension Table)

  define dimension <dimension_name> as
  (<attribute_or_subdimension_list>)

- Special Case (Shared Dimension Tables)
  - First time as "cube definition"
  - define dimension <dimension_name> as <dimension_name_first_time> in cube <cube_name_first_time>

Defining a Star Schema in DMQL

```
define cube sales_star [time, item, branch, location]:
    dollars_sold = sum(sales_in_dollars), avg_sales =
      avg(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month,
  quarter, year)
define dimension item as (item_key, item_name, brand, type,
  supplier_type)
define dimension branch as (branch_key, branch_name,
  branch_type)
define dimension location as (location_key, street, city,
  province_or_state, country)
```

Defining a Snowflake Schema in DMQL

define cube sales_snowflake [time, item, branch, location]:

dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)

define dimension time as (time_key, day, day_of_week, month, quarter, year)

define dimension item as (item_key, item_name, brand, type, supplier(supplier_key, supplier_type))

define dimension branch as (branch_key, branch_name, branch_type)

define dimension location as (location_key, street, city(city_key, province_or_state, country))

Defining a Fact Constellation in DMQL

```
define cube sales [time, item, branch, location]:
        dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars),
          units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city, province_or_state, country)
define cube shipping [time, item, shipper, from_location, to_location]:
        dollar_cost = sum(cost_in_dollars), unit_shipped = count(*)
define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper_key, shipper_name, location as location in cube
    sales, shipper_type)
define dimension from_location as location in cube sales
define dimension to_location as location in cube sales
```