Chapter -1

Data Warehousing

Why Reporting & Analyzing Data?

- The amount of data stored in databases is growing exponentially & databases are now measured in gigabytes(GBs) and terabytes(TBs).
- However row data does not provide useful information.
- In today’s highly competitive business environment, companies need to turn these terabytes of raw data into some **useful information**.
- The general methods of analysis/reporting can be broadly classified into two categories: **non-parametric** analysis & **parametric** analysis
- **Example**
    - Managers will generally be more interested in **actual data** and non-parametric analysis results, while engineers will be more concerned with parametric analysis.

Introduction to Data Warehouse

- Collections of databases that work together are called **data warehouses.**
- This makes it possible to **integrate data from multiple databases** & it is used to help individuals and organizations **make better decisions.**
- A database consists of one or more files that need to be stored on a computer.
- In large organizations, databases are typically not stored on the individual computers of employees but in a **central system (server).**

Data Warehouse (Cont..)

- A **server** is a computer system that provides a **service over a network**. The server is often located in a specific place with controlled access, so only authorized person can get physical access for it.
- In a typical setting, the database files reside on the server, but it can be accessed from many different computers in the organization.
- As the number and complexity of databases grows, we start referring to them together as a data warehouse.
- **The ultimate goal of a database is not just to store data, but to help businesses make decisions based on that data.**
- A data warehouse supports this goal by providing an architecture and tools to systematically organize and understand data from multiple databases.

Data Warehouse (Cont..)

- According to William H. Inmon, a leading architect in the 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”.
- **Features of Data Warehousing**
  - Subject-oriented
  - Integrated
  - Time-variant
  - Nonvolatile

Features of Data Warehouse

- **Subject-oriented:**
  - A data warehouse is organized around major subjects, such as **customer, supplier, product, and sales.**
  - Rather than concentrating on the day-to-day operations and transaction processing of an organization, a **data warehouse focuses on the modeling and analysis of data for decision makers.**
  - Data warehouses **typically provide a simple and concise view around particular subject** issues by excluding data that are not useful in the decision support process.

Features of Data Warehouse (Cont..)

- **Integrated:**
  - A data warehouse is usually constructed by **integrating multiple heterogeneous sources**, such as **relational databases, flat files, and on-line transaction records.**
  - Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, and so on.
- **Time-variant:**
  - Data are stored to provide information from a historical perspective (e.g., the past 5–10 years).
  - Every key structure in the data warehouse contains, either implicitly or explicitly, an element of time.

Features of Data Warehouse (Cont..)

- **Nonvolatile:**
  - A data warehouse is always a physically separate store of data transformed from the application data found in the operational environment.
  - Due to this separation, a **data warehouse does not require transaction processing, recovery, and concurrency control mechanisms**.
  - It usually requires only two operations in data accessing: **initial loading of data and access of data**.

Data Warehouse Design Process

- A data warehouse can be built using a **top-down** approach, a **bottom-up** approach, or a **combination of both**.
- **Top Down Approach**
  - The top-down approach starts with the overall design and planning.
  - It is useful in cases where the technology is mature and well known, and where the business problems that must be solved are clear and well understood.
- **Bottom up Approach**
  - The bottom-up approach starts with experiments and prototypes.
  - This is useful in the early stage of business modeling and technology development.
  - It allows an organization to move forward at considerably less expense and to evaluate the benefits of the technology before making significant commitments.
- **Combined Approach**
  - In the combined approach, an organization can exploit the planned and strategic nature of the top-down approach while retaining the rapid implementation and opportunistic application of the bottom-up approach.

Types of Data Warehouse

- The three main types of data warehouses are:
  - **Enterprise Data Warehouse**
  - **Operational Data Store**
  - **Data Mart**

Data Warehouse Types (Cont..)

- **Enterprise Data Warehouse**:
  - Enterprise Data Warehouse is a **centralized warehouse**, which provides **decision support service across the enterprise**.
  - It offers a **unified approach to organizing and representing dat**a.
  - It also provides the **ability to classify data according to the subject** and give access according to those divisions.
- **Operational Data Store**:
  - Operational Data Store, also called ODS, is data store required when neither data warehouse nor OLTP systems support organizations reporting needs.
  - It is widely preferred for **routine activities like storing records.**.
  - In ODS, Data warehouse is refreshed in real time.
- **Data Mart**:
  - A Data Mart is a subset of the data warehouse.
  - It specially designed for specific segments like sales, finance, sales, or finance.
  - In an independent data mart, data can collect directly from sources.

Introduction to Data Marts

- A data mart is a **simple form of a data warehouse** that is focused on a single subject (or functional area), such as **Sales or Finance or Marketing**.
- Data marts are often built and controlled by a **single department within an organization,** given their single-subject focus, data marts usually draw data from only a few sources.
- The sources could be internal operational systems, a central data warehouse, or external data.

Introduction to Data Marts (Cont..)

- A data mart is a repository of data that is **designed to serve a particular community of knowledge workers**.
- The difference between a data warehouse and a data mart can be confusing because the two terms are sometimes used incorrectly as synonyms.
- A data warehouse is a **central repository for all an organization's data**.
- The goal of a data mart, however, is to meet the particular demands of a specific group of users within the organization, such as human resource management (HRM).
- Generally, an organization's data marts are subsets of the organization's data warehouse.

Reasons for Creating a Data Marts

- Easy access to frequently needed data
- Creates collective view by a group of users
- Improves end-user response time
- Ease of creation
- Lower cost than implementing a full data warehouse
- Potential users are more clearly defined than in a full data warehouse
- Contains only business essential data and is less cluttered

Data Warehouse v/s Data Mart

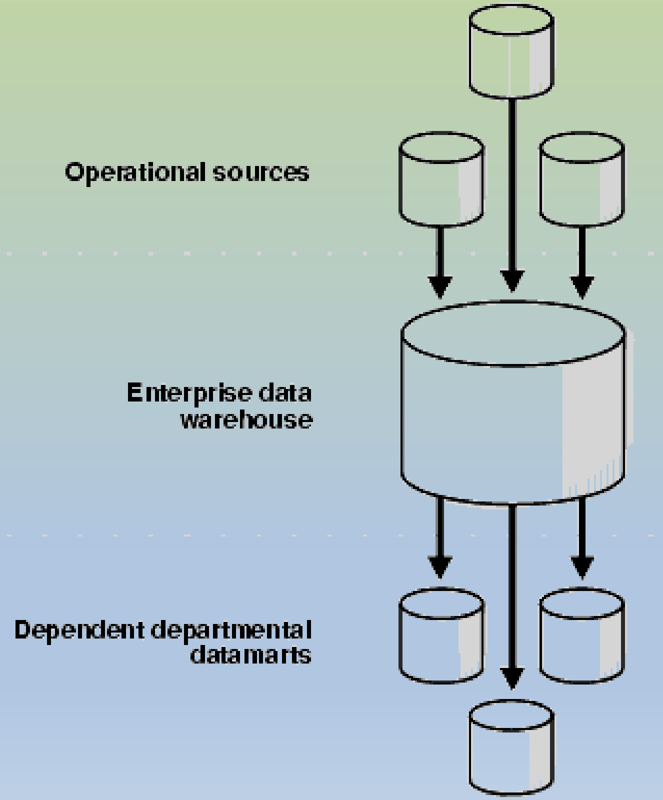
- **Data warehouse**:
  - Holds multiple subject areas
  - Holds very detailed information
  - Works to integrate all data sources
  - Size (typical) 100 GB-TB+
  - Implementation Time : Months to Years
- **Data mart**:
  - Often holds only one subject area- for example, Finance, or Sales
  - May hold more summarized data
  - Concentrates on integrating information from a given subject area or set of source systems
  - Size (typical) < 100GB
  - Implementation Time : Months

Types of Data Marts

- There are three kinds of Data-Marts (DMs), which are as follows:
  1) **Dependent DM**: Created from a data warehouse to a separate physical data-store. (build over data warehouse physically)
  2) **Independent DM**: Created from operational systems and have separate physical data-store.
  3) **Logical or Hybrid DM**: Exists as a subset of data warehouse. (build over data warehouse logically)

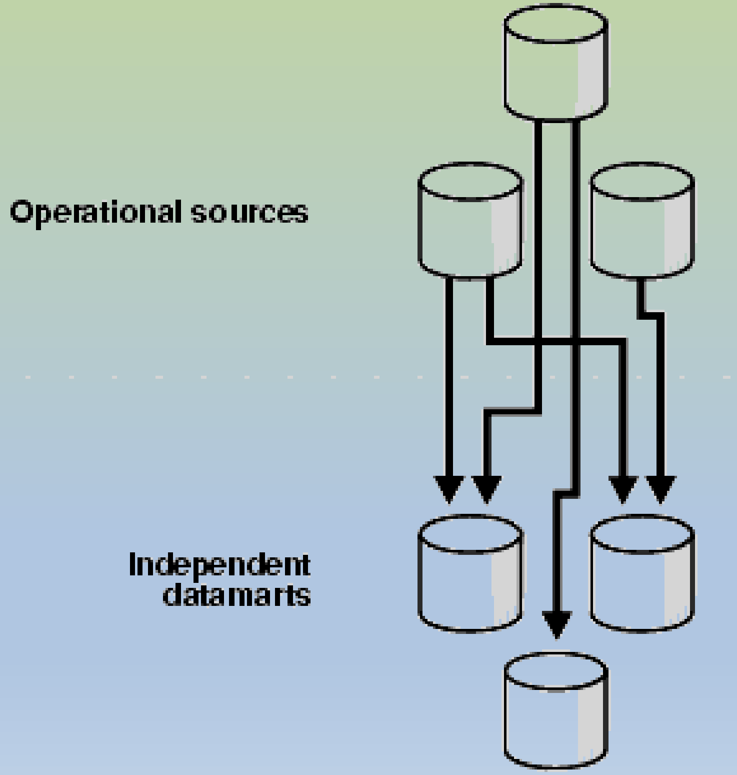
1) Dependent Data Marts

- A dependent data mart allows you to unite your organization's data in one data warehouse.
- This gives you the usual advantages of centralization.



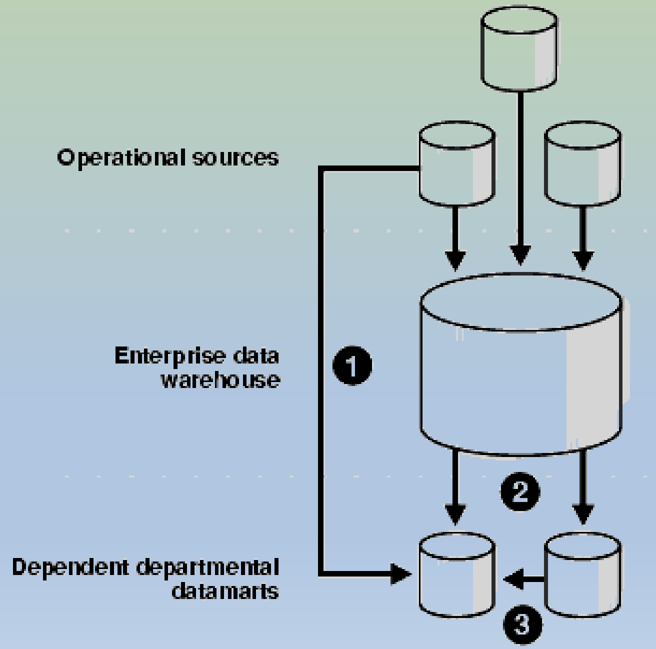
2) Independent Data Marts

- An independent data mart is created without the use of a central data warehouse.
- This could be desirable for smaller groups within an organization.



3) Hybrid Data Mart

- A hybrid data mart allows you to combine input from sources other than a data warehouse.
- This could be useful for many situations, especially when you need ad hoc integration, such as after a new group or product is added to the organization.



Meta data

- Metadata are **data about data**.
- When meta data is used in a data warehouse, that defines warehouse objects.
- Metadata are created for the data names and definitions of the given warehouse.
- Additional metadata are created and captured for time stamping any extracted data, the source of the extracted data, and missing fields that have been added by data cleaning or integration processes.

Metadata – Example

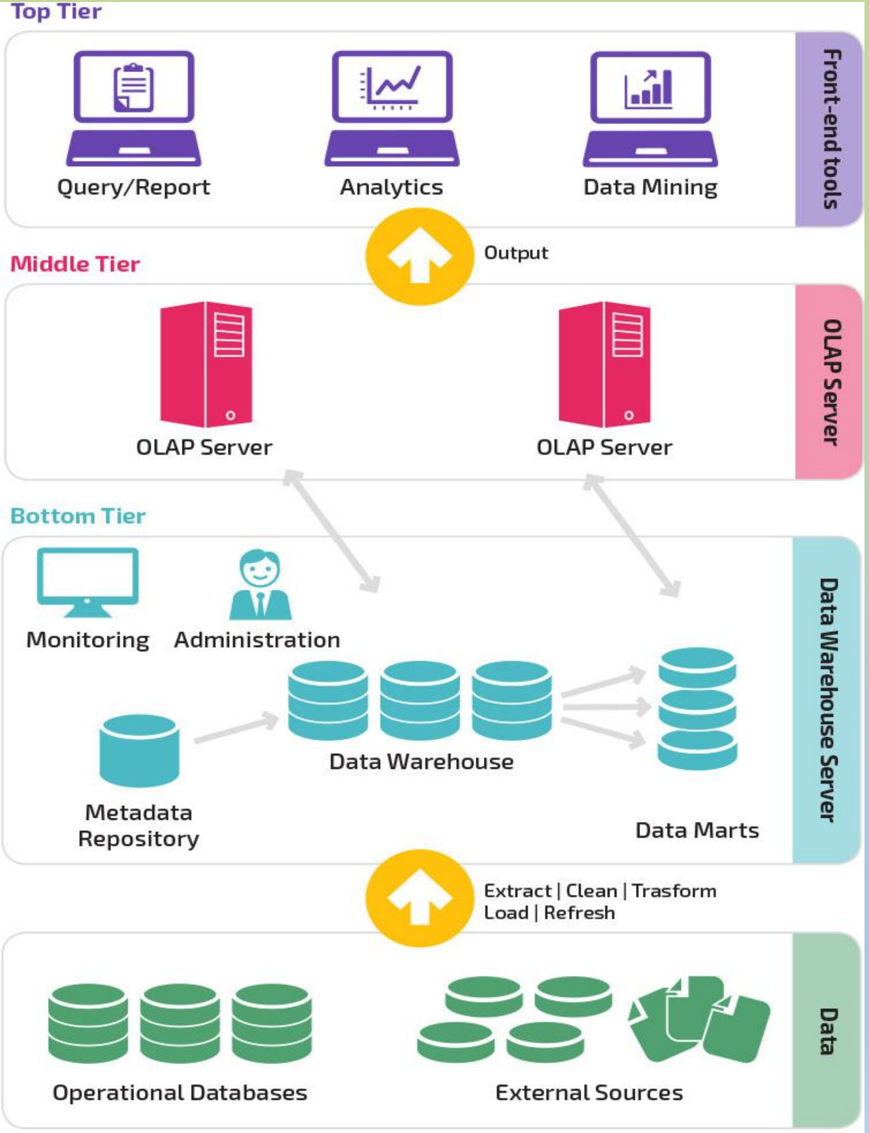
- **To Describe Meta Data of a Book Store:**
  - Name of Book
  - Summary of the Book
  - The Date of publication
  - High level description of what it contains
  - How you can find the book
  - Author of the book
  - Whether the book is available OR not
- **The information helps you to:**
  - Search for the book
  - Access the book
  - Understand the book before you access OR buy it.

Data Warehouse Architecture

Top Tier

Middle Tier

Bottom Tier



Data Warehouse Architecture

- **Bottom tier:**
  - The **bottom tier** is a warehouse **database server** that is almost always a relational database system.
  - **Back-end tools and utilities are used to feed data** into the bottom tier from operational databases or other external sources.
  - These tools and utilities **perform data extraction, cleaning, and transformation**, as well as load and refresh functions to update the data warehouse.
  - The data are extracted using application program interfaces known as **gateways**.
  - A gateway is supported by the underlying DBMS and allows client programs to generate SQL code to be executed at a server.
  - Examples of gateways include ODBC (Open Database Connection) and OLEDB (Open Linking and Embedding for Databases) by Microsoft and JDBC (Java Database Connection).
  - This tier also **contains a metadata repository**, which stores information about the data warehouse and its contents.

Data Warehouse Architecture

- **Middle tier:**
  - The middle tier is an OLAP (Online Analytical Processing Server) that is typically implemented using either
    - A **relational OLAP** **(ROLAP**) model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations or,
    - A **multidimensional OLAP** **(MOLAP)** model, that is, a special-purpose server that directly implements multidimensional data and operations.
- **Top tier:**
  - The top tier is a front-end client layer, which contains **query and reporting tools, analysis tools, and/or data mining tools**.

OLAP (**O**n-**L**ine **A**nalytical **P**rocessing)

- OLAP is characterized by relatively **low volume of transactions**.
- Queries are often **very complex and involve aggregations**.
- For OLAP systems a **response time is an effectiveness measure**.
- OLAP applications are widely used by Data Mining techniques.
- In OLAP database there is **aggregated, historical data, stored in multi-dimensional** schemas (usually star schema).

OLTP (**O**n-**L**ine **T**ransaction **P**rocessing)

- It is characterized by a **large number of short on-line transactions** (INSERT, UPDATE, DELETE).
- The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second.
- In OLTP database, **there is detailed and current data**, and schema used to store transactional databases is the entity model (usually 3NF).

OLTP v/s OLAP (Understanding)

OLTP	OLAP
Many Short Transactions (Queries + Updates)	Long Transactions (Complex Queries)
Examples • Update account balance • Enroll in course • Add book to shopping cart	**Examples** • Report total sales for each department in each month • Identify top-selling books • Count classes with fewer than 10 students
Queries touch small amount of data (one record or few records)	Queries touch large amount of data
Updates are frequent	Updates are infrequent

OLTP v/s OLAP

Functionality	OLTP	OLAP
Characteristic	Operational processing informational processing	Transaction Analysis
Orientation	Transaction	Analysis
User	Clerk, DBA, database professional	Knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements, decision support
DB design	ER based, **application-oriented**	Star/snowflake, **subject-oriented**
Data	Current; guaranteed up-to-date	Historical; accuracy maintained over time
Summarization	Primitive, highly detailed	Summarized, consolidated
View	Detailed, flat relational	Summarized, multidimensional
Unit of work	Short, simple transaction	Complex query
Access	Read/write	Mostly read

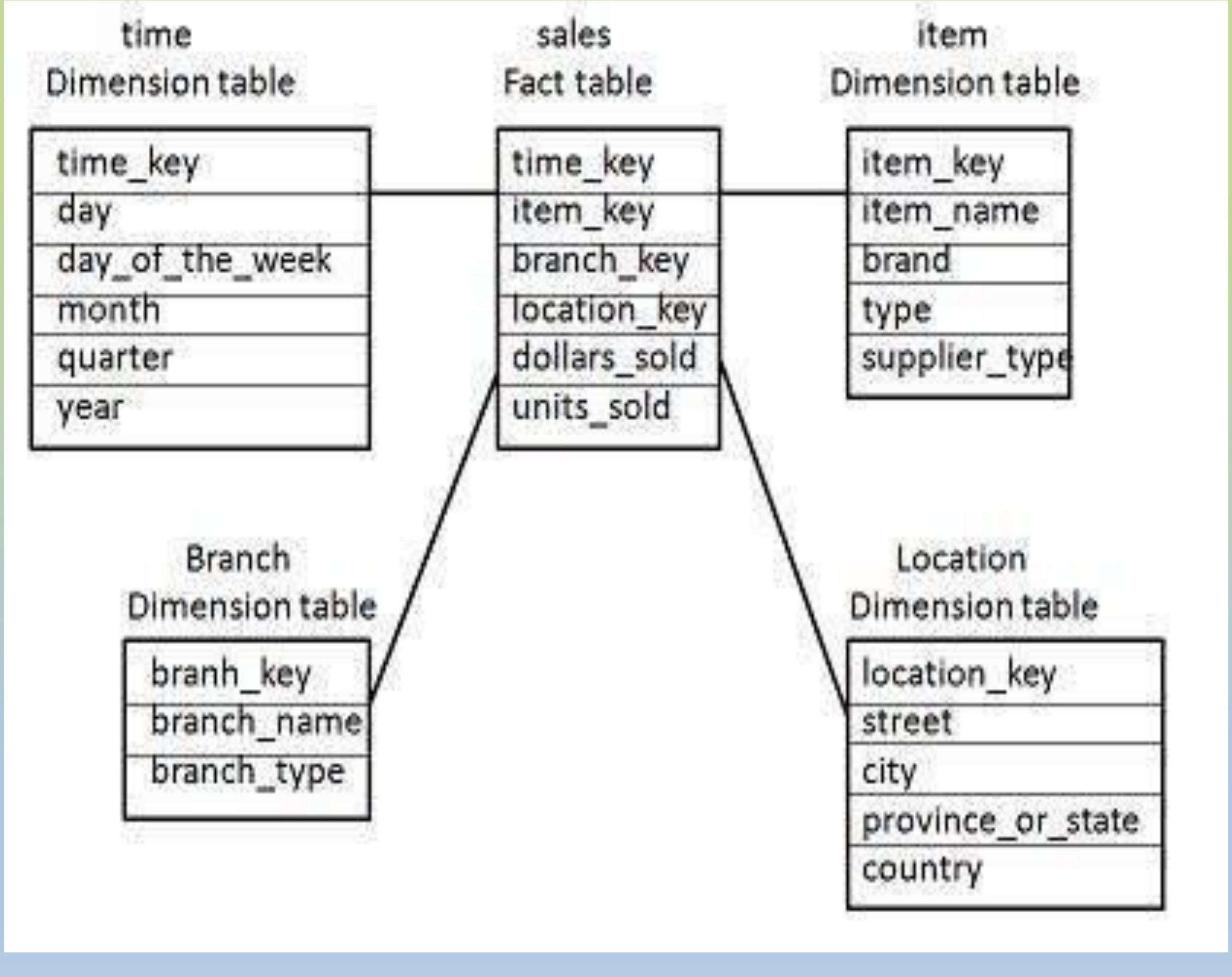
Data Warehouse Schema Architecture

- Data Warehouse environment usually transforms the relational data model into some special architectures.
- There are many schema models designed for data warehousing but the most commonly used are:
  - **Star Schema**
  - **Snowflake Schema**
  - **Fact constellation(Group of star, Collection of fact tables) Schema**
- The determination of which schema model should be used for a data warehouse based upon the analysis of project requirements, accessible tools and project team preferences.

Star Schema

- The star schema architecture is the **simplest data warehouse schema.**
- It is called a star schema because the diagram resembles a **star**, with points radiating from a center.
- The center of the star consists of **fact table** and the **points of the star are the dimension tables**.
- Usually the fact tables in a star schema are in third normal form (3NF) whereas dimensional tables are de-normalized.
- Despite the fact that the star schema is the simplest architecture, it is **most commonly used nowadays** and is recommended by Oracle.

Star Schema - Example



time
Dimension table

time_key
day
day_of_the_week
month
quarter
year

sales
Fact table

time_key
item_key
branch_key
location_key
dollars_sold
units_sold

item
Dimension table

item_key
item_name
brand
type
supplier_type

Branch
Dimension table

branh_key
branch_name
branch_type

Location
Dimension table

location_key
street
city
province_or_state
country

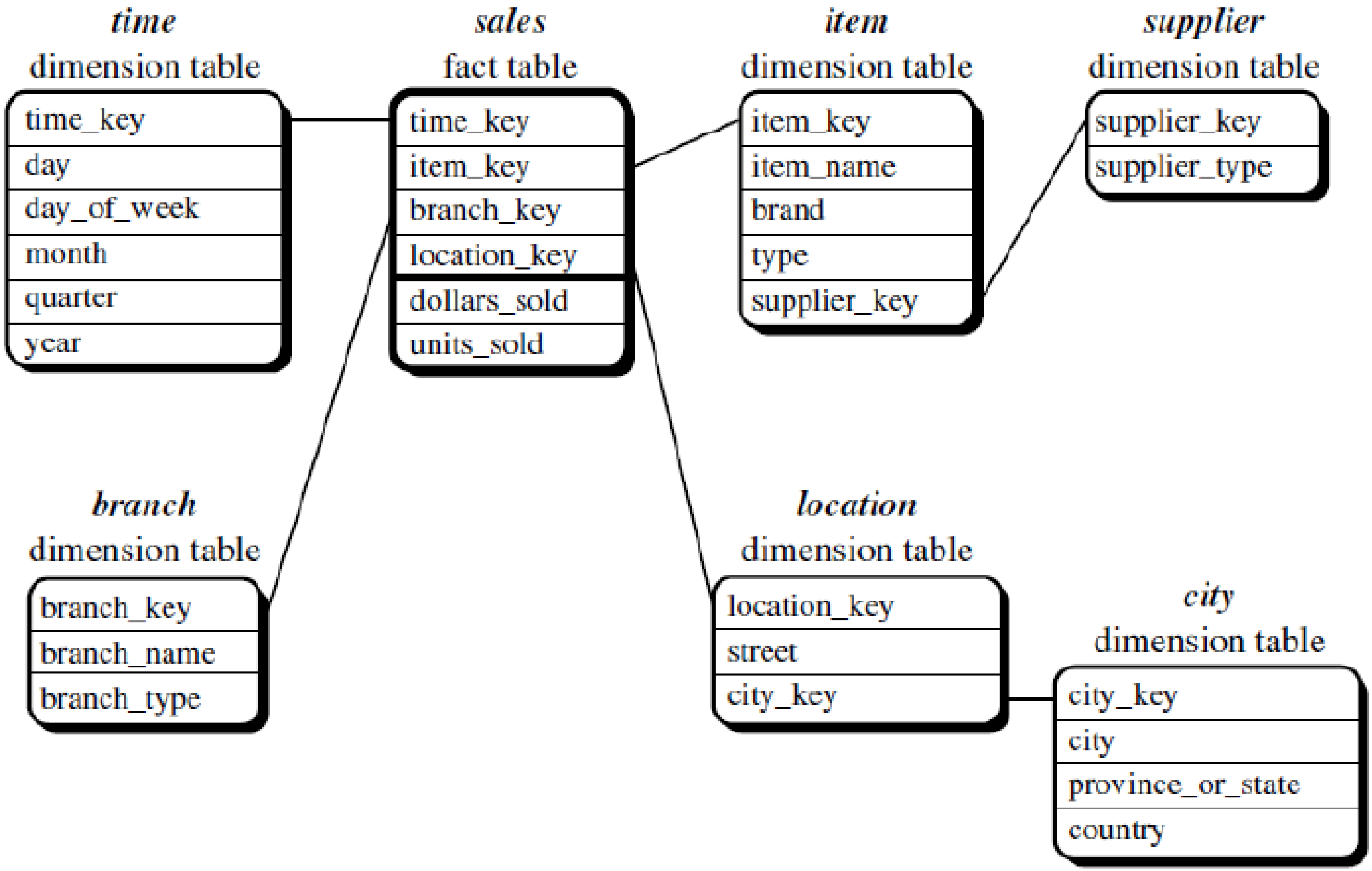
Continue…

- Given diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.
- There is a fact table at the center. It contains the keys to each of four dimensions.
- The fact table also contains the attributes, namely dollars sold and units sold.

Snowflake Schema

- The snowflake schema architecture is a **more complex variation of the star schema** used in a data warehouse, because the tables which describe the dimensions are normalized.
- This table is easy to maintain and saves storage space.
- However, this saving of space is negligible in comparison to the typical size of the fact table.
- Furthermore, the snowflake structure can reduce the effectiveness of browsing, since **more joins** will be needed to execute a query.
- Hence, although the **snowflake schema reduces redundancy**, it is not as popular as the star schema in data warehouse design.

Snowflake Schema - Example



time
dimension table

- time_key
- day
- day_of_week
- month
- quarter
- year

sales
fact table

- time_key
- item_key
- branch_key
- location_key
- dollars_sold
- units_sold

item
dimension table

- item_key
- item_name
- brand
- type
- supplier_key

supplier
dimension table

- supplier_key
- supplier_type

branch
dimension table

- branch_key
- branch_name
- branch_type

location
dimension table

- location_key
- street
- city_key

city
dimension table

- city_key
- city
- province_or_state
- country

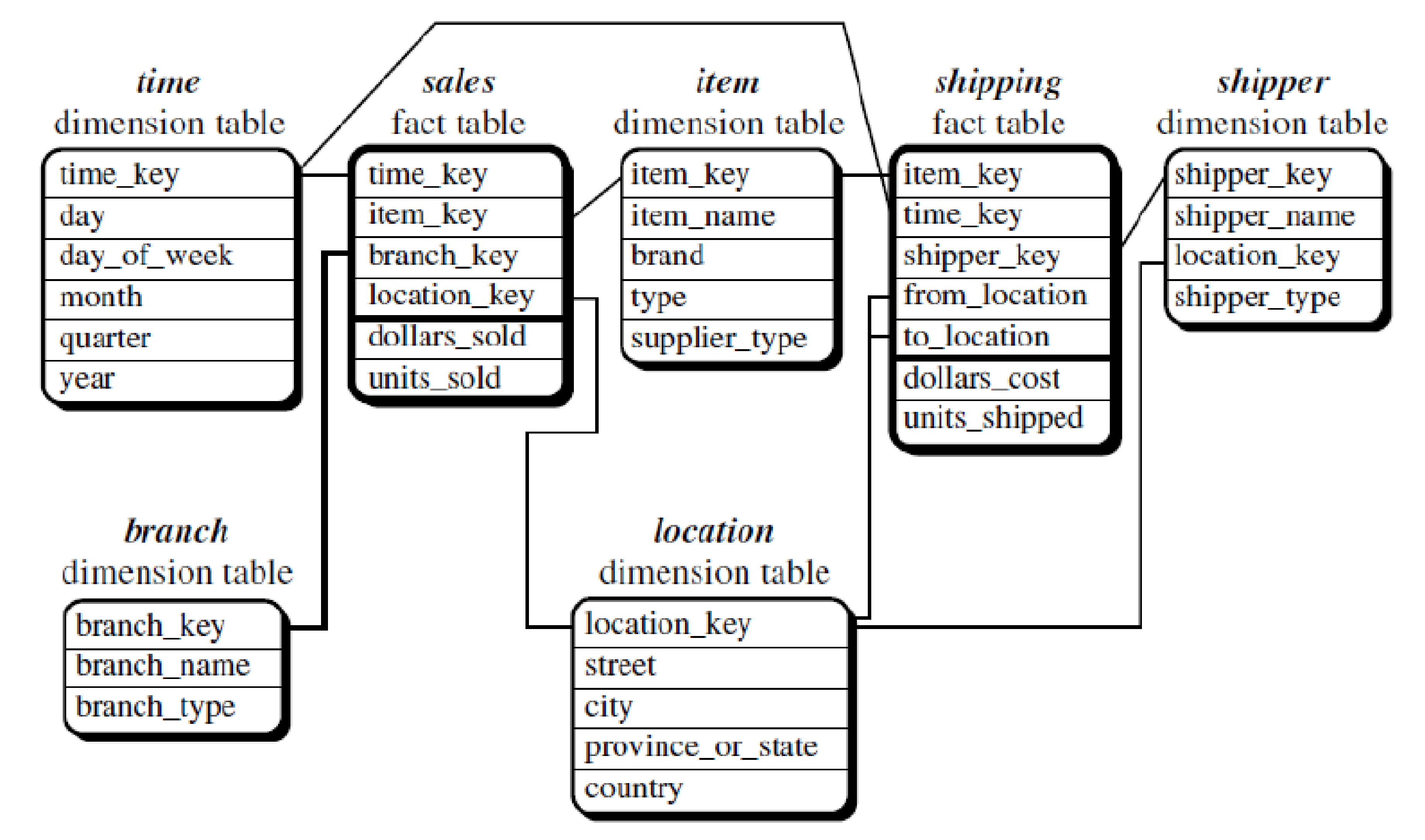
Snowflake Schema - Example

- DMQL(Data Mining Query Language) code for Snowflake Schema can be written as follows:
  - Define cube sales snowflake [**time, item, branch, location**]:
  - **Dollars sold** = sum(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, city, province or state, country))

Fact Constellation 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**.
- A fact constellation schema allows dimension tables to be shared between fact tables.
- For example, the dimensions tables for ***time, item***, and ***location*** are shared between both the **sales and shipping** fact tables.
- The main shortcoming of the fact constellation schema is a more complicated design because many variants for particular kinds of aggregation must be considered and selected.

Fact Constellation Schema

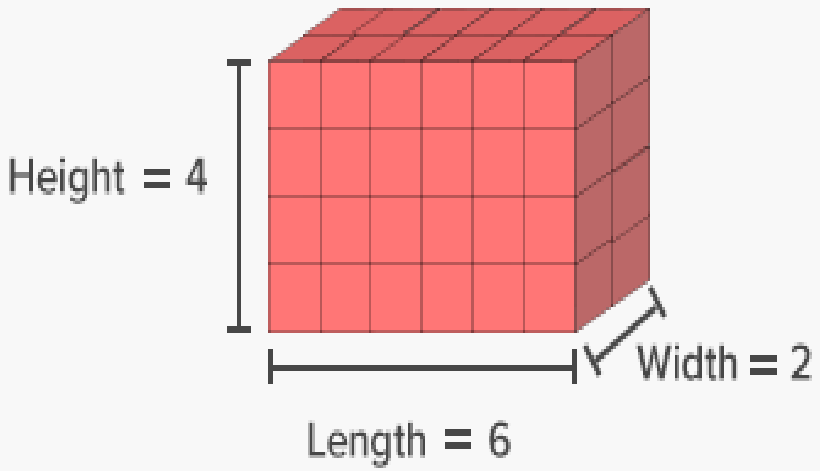


Fact Constellation Schema

- DMQL code for Fact Constellation schema can be written as follows:
  - Define cube sales [time, item, branch, location]:
  - Dollars sold = sum(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]:
  - Dollars cost = sum(cost in dollars), units 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

OLAP Operations

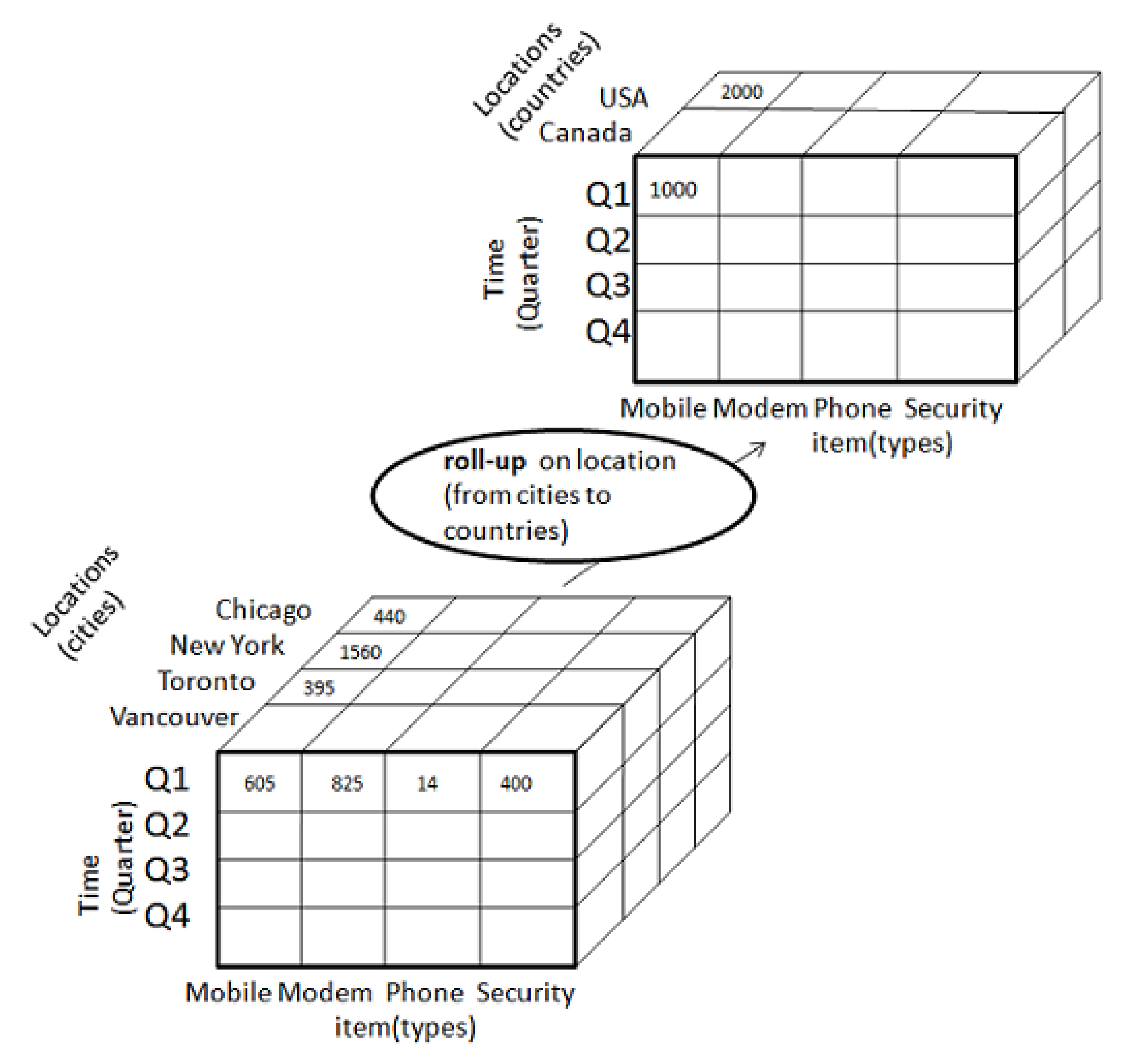
- Roll up
- Drill Down
- Slice
- Dice
- Pivot (Rotate)



Roll up – OLAP Operation

- The roll-up operation (also called drill-up or aggregation operation) **performs aggregation on a data cube** by following ways:
  - By climbing up a concept hierarchy for a dimension
  - By dimension reduction
- Roll-up is performed by **climbing up** a concept hierarchy for the dimension location.
- Initially the concept hierarchy was "street < city < province < country".
- On rolling up, the **data is aggregated by ascending the location hierarchy from the level of city to the level of country**.
- The data is grouped into cities rather than countries.
- When roll-up is performed, one or more dimensions from the data cube are removed.

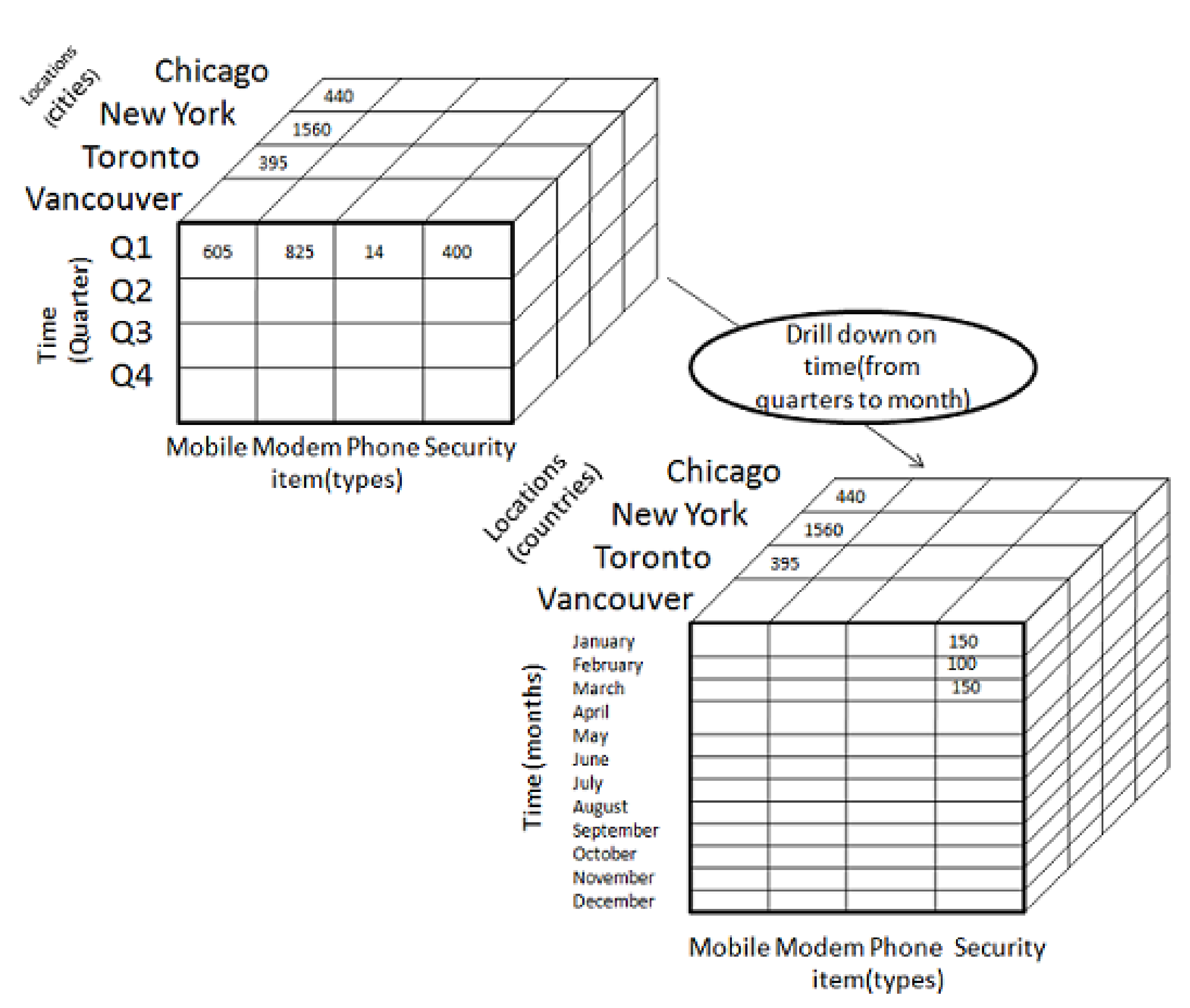
Roll up – OLAP Operation



Drill Down – OLAP Operation

- Drill-down is the reverse operation of roll-up. It is performed by either of the following ways:
  - By stepping down a concept hierarchy for a dimension
  - By introducing a new dimension
- Drill-down is performed by **stepping down** a concept hierarchy for the dimension time.
- Initially the concept hierarchy was "day < month < quarter < year."
- On drilling down, **the time dimension is descended from the level of quarter to the level of month**.
- When drill-down is performed, **one or more dimensions from the data cube are added**.
- It navigates the data from less detailed data to highly detailed data.

Drill Down – OLAP Operation

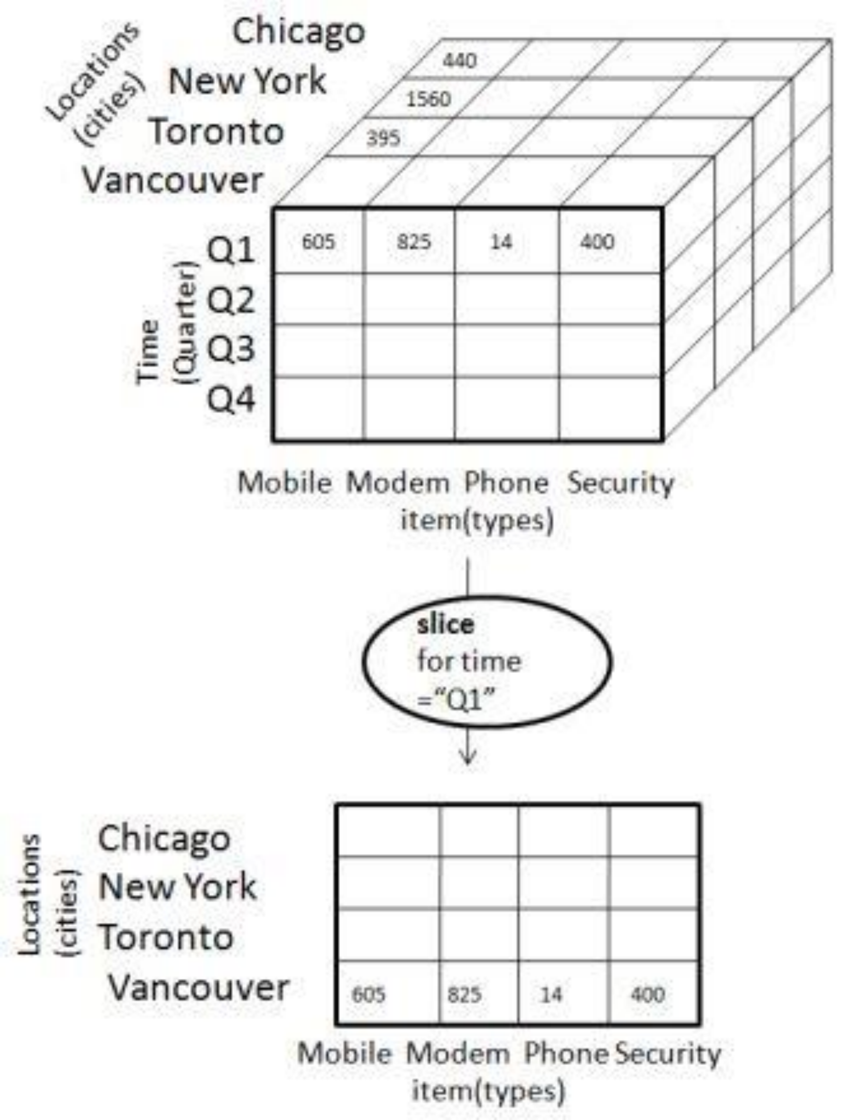


Drill down on time(from quarters to month)

Slice – OLAP Operation

- The slice operation **selects one particular dimension from a given cube and provides a new sub cube**.
- Here Slice is performed for the dimension "time" using the criterion time = "Q1“, time = "Q2“, time = "Q3“ etc.
- It will form a new sub-cube by selecting one or more dimensions.

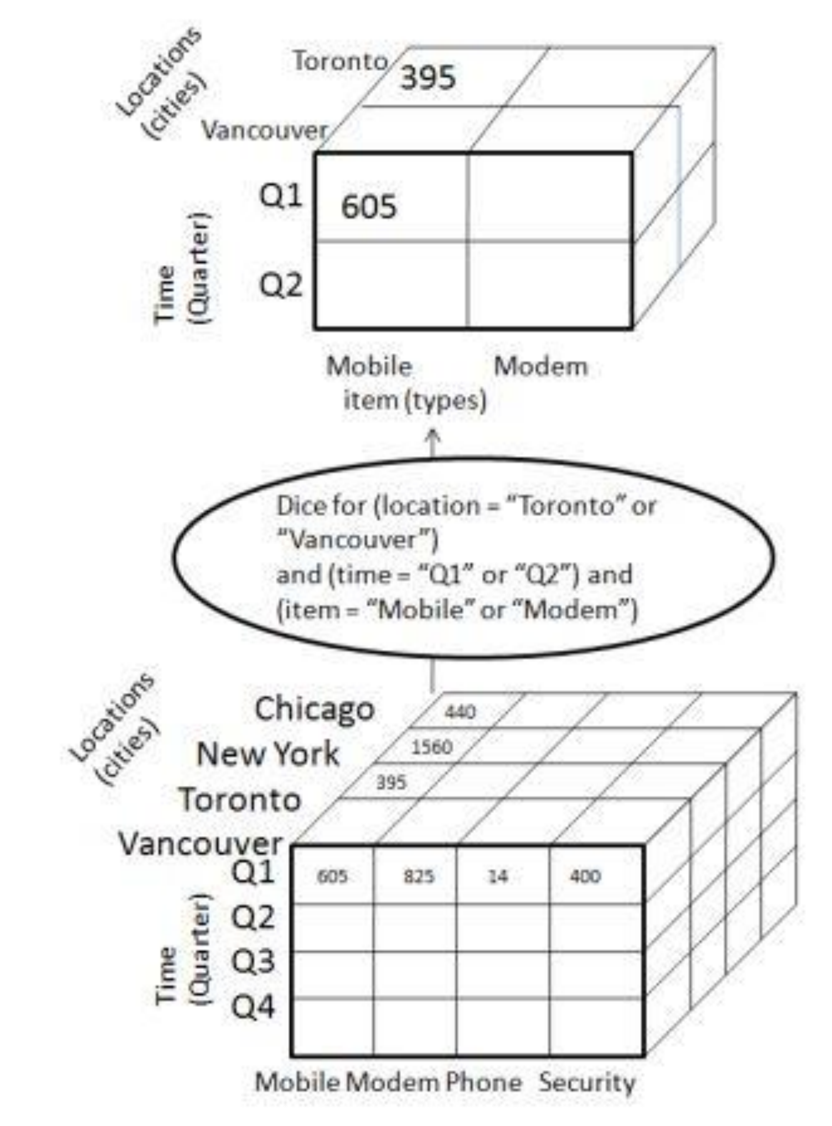
Slice – OLAP Operation



Dice – OLAP Operation

- Dice **selects two or more dimensions** from a given cube and provides a new sub cube.
- The dice operation on the cube based on the following selection criteria involves three dimensions.
  - (location = "Toronto" or "Vancouver")
  - (time = "Q1" or "Q2")
  - (item =" Mobile" or "Modem")

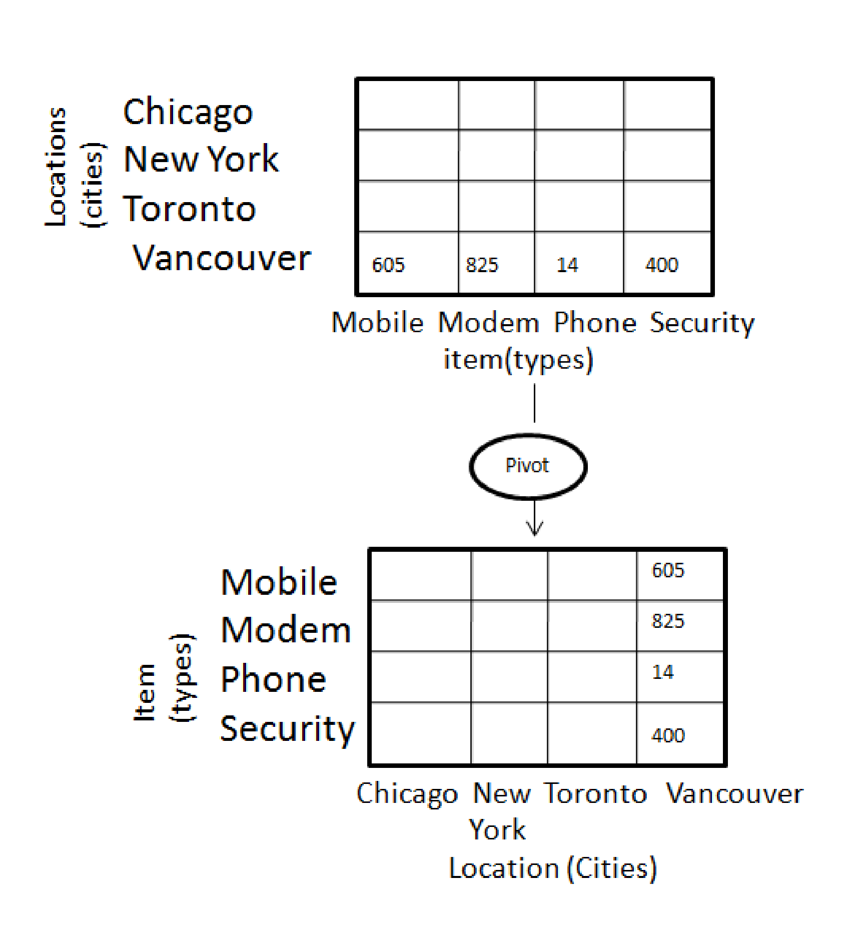
Dice – OLAP Operation



Dice for (location = "Toronto" or "Vancouver") and (time = "Q1" or "Q2") and (item = "Mobile" or "Modem")

Pivot – OLAP Operation

- It is a technique of changing one dimension operation to another.
- The pivot operation is also known as **rotation**.
- It rotates the data axes in rnative presentation of data.



Locations (cities)	Mobile	Modem	Phone	Security
Chicago				
New York				
Toronto				
Vancouver	605	825	14	400

Item (types)	Chicago	New York	Toronto	Vancouver
Mobile				605
Modem				825
Phone				14
Security				400

OLAP Servers

- Relational OLAP (ROLAP)
- Multidimensional OLAP (MOLAP)
- Hybrid OLAP (HOLAP)

Relational OLAP (ROLAP)

- Relational On-Line Analytical Processing (ROLAP) work mainly for the data that resides in a **relational database**, where the base data and dimension tables are **stored as relational tables**.
- ROLAP servers are placed between the relational back-end server and client front-end tools.
- ROLAP servers use RDBMS to store and manage warehouse data, and OLAP middleware to support missing pieces.
  - **Advantages of ROLAP**
    - ROLAP can handle large amounts of data.
    - Can be used with data warehouse and OLTP systems.
  - **Disadvantages of ROLAP**
    - Limited by SQL functionalities.
    - Hard to maintain aggregate tables.

Multidimensional OLAP (MOLAP)

- Multidimensional On-Line Analytical Processing (MOLAP) support **multidimensional views of data** through array-based multidimensional storage engines.
- With multidimensional data stores, the storage utilization may be low if the data set is sparse.
  - **Advantages of MOLAP**
    - Optimal for slice and dice operations.
    - Performs better than ROLAP when data is dense(heavy).
    - Can perform complex calculations.
  - **Disadvantages of MOLAP**
    - Difficult to change dimension without re-aggregation.
    - MOLAP can handle limited amount of data.

Hybrid OLAP (HOLAP)

- Hybrid On-Line Analytical Processing (HOLAP) is a **combination of ROLAP and MOLAP**.
- HOLAP provide greater scalability of ROLAP and the faster computation of MOLAP.
  - **Advantages of HOLAP**
    - HOLAP provide advantages of both MOLAP and ROLAP.
    - Provide fast access at all levels of aggregation.
  - **Disadvantages of HOLAP**
    - HOLAP **architecture is very complex** because it support both MOLAP and ROLAP servers.

What is ETL ?

- ETL (or Extract, Transform, Load) is a process of data integration that encompasses three steps — extraction, transformation, and loading.
- ETL systems take large volumes of raw data from multiple sources, converts it for analysis, and loads that data into your warehouse. Let's cover the three primary ETL steps.

    1. Extraction
    2. Transform
    3. Loading

Extraction

- In the first step, extracted data sets come from a source (e.g., Salesforce, Google AdWords, etc.) into a staging area.
- The staging area acts as a buffer between the data warehouse and the source data.
- Since data may be coming from multiple different sources, it's likely in various formats, and directly transferring the data to the warehouse may result in corrupted data.
- The staging area is used for data cleansing and organization.
- A big challenge during the data extraction process is how your ETL tool handles structured and unstructured data.
- All of those unstructured items (e.g., emails, web pages, etc.) can be difficult to extract without the right tool, and you may have to create a custom solution to assist you in transferring unstructured data if you chose a tool with poor unstructured data capabilities.

Transformation

- The data cleaning and organization stage is the transformation stage.
- All of that data from multiple source systems will be normalized and converted to a single system format — improving data quality and compliance. ETL yields transformed data through these methods:

- Cleaning
- Filtering
- Joining
- Sorting
- Splitting
- Summarization

Loading

- Finally, data that has been extracted to a staging area and transformed is loaded into your data warehouse. Depending upon your business needs, data can be loaded in batches or all at once. The exact nature of the loading will depend upon the data source, ETL tools, and various other factors.