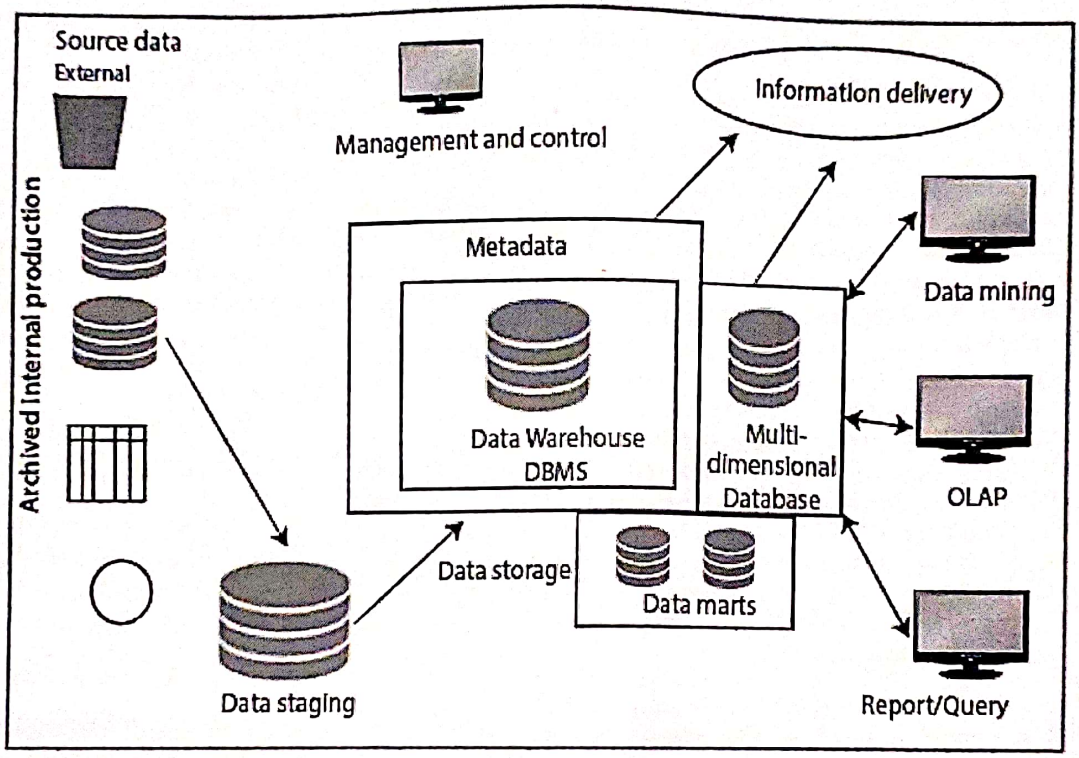
# UNIT 2

## DATA WAREHOUSES AND DATA MARTS

### COMPONENTS OR BUILDING BLOCKS OF DATA WAREHOUSE

Architecture is the proper arrangement of the elements. We build a data warehouse with software and hardware components. To suit the requirements of our organizations, we arrange these buildings we may want to boost up another part with extra tools and services. All of these depend on our circumstances.




**Components or Building Blocks of Data Warehouse**

The figure shows the essential elements of a typical warehouse. We see the Source Data component shows on the left. The Data staging element serves as the next building block. In the middle, we see the Data Storage component that handles the data warehouse, data. This element not only stores and manages the data; it also keeps track of data using the metadata repository. The Information Delivery component shown on the right consists of all the different ways of making the information from the data warehouses available to the users.

**1. Source Data Component**

Source data coming into the data warehouses may be grouped into four broad categories:

- **Production Data:** This type of data comes from the different operating systems of the enterprise. Based on the data requirements in the data warehouse, we choose segments of the data from the various operational modes.
- **Internal Data:** In each organization, the client keeps their "private" spreadsheets, reports, customer profiles, and sometimes even department databases. This is the internal data, part of which could be useful in a data warehouse.

- **Archived Data:** Operational systems are mainly intended to run the current business. In every operational system, we periodically take the old data and store it in achieved files.
- **External Data:** Most executives depend on information from external sources for a large percentage of the information they use. They use statistics associated with their industry produced by the external department.

### 2. Data Staging Component

After we have extracted data from various operational systems and external sources, we have to prepare the files for storage in the data warehouse. The extracted data coming from several different sources need to be changed, converted, and made ready in a format that is relevant to be saved for querying and analysis.

The three primary functions that take place in the staging area.

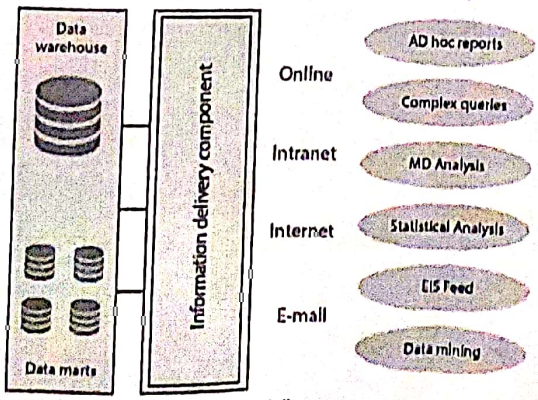
- **Data Extraction:** This method has to deal with numerous data sources. We have to employ the appropriate techniques for each data source.
- **Data Transformation:** As we know, data for a data warehouse comes from many different sources. If data extraction for a data warehouse posture big challenge, data transformation presents even more significant challenges. We perform several individual tasks as part of data transformation. First, we clean the data extracted from each source. Cleaning may be the correction of misspellings or may deal with providing default values for missing data elements, or elimination of duplicates when we bring in the same data from various source systems. Standardization of data components forms a large part of data transformation. Data transformation contains many forms of combining pieces of data from different sources. We combine data from a single source record or related data parts from many source records. On the other hand, data transformation also contains purging source data that is not useful and separating outsourced records into new combinations. Sorting and merging of data take place on a large scale in the data staging area. When the data transformation function ends, we have a collection of integrated data that is cleaned, standardized, and summarized.
- **Data Loading:** Two distinct categories of tasks form data loading functions. When we complete the structure and construction of the data warehouse and go live for the first time, we do the initial loading of the information into the data warehouse storage. The initial load moves high volumes of data using up a substantial amount of time.

### 3. Data Storage Components

Data storage for data warehousing is a split repository. The data repositories for the operational systems generally include only the current data. Also, these data repositories include the data structured in highly normalized for fast and efficient processing.

### 4. Information Delivery Component

The information delivery element is used to enable the process of subscribing for data warehouse files and having it transferred to one or more destinations according to some customer-specified scheduling algorithm.




Information delivery component

### 5. Metadata Component

Metadata in a data warehouse is equal to the data dictionary or the data catalog in a database management system. In the data dictionary, we keep the data about the logical data structures, the data about the records and addresses, the information about the indexes, and so on.

### 6. Data Marts

It includes a subset of corporate-wide data that is of value to a specific group of users. The scope is confined to particular selected subjects. Data in a data warehouse should be fairly current, but not mainly up to the minute, although development in the data warehouse industry has made standard and incremental data dumps more achievable. Data marts are lower than data warehouses and usually contain organization. The current trends in data warehousing are to develop a data warehouse with several smaller related data marts for particular kinds of queries and reports.

### 7. Management and Control Component

The management and control elements coordinate the services and functions within the data warehouse. These components control the data transformation and the data transfer into the data warehouse storage. On the other hand, it moderates the data delivery to the clients. Its work with the database management systems and authorizes data to be correctly saved in the repositories. It monitors the movement of information into the staging method and from there into the data warehouse storage itself.

## METADATA

Metadata is data about data. In data warehouse is equal to the data dictionary or the data catalog in a database management system. In the data dictionary, we keep the data about the logical data structures, the data about the records and addresses, the information about the indexes, and so on. Metadata can be stored in various forms, such as text, XML, or RDF, and can be organized using metadata standards and schemas. There are many metadata standards that have been developed to facilitate the creation and management of metadata, such as Dublin Core, schema.org, and the Metadata Encoding and Transmission Standard (METS). Metadata schemas define the structure and format of metadata and provide a consistent framework for organizing and describing data.

### Types of Metadata

There are many types of metadata that can be used to describe different aspects of data, such as its content, format, structure, and provenance. Some common types of metadata include:

1. **Descriptive metadata:** This type of metadata provides information about the content, structure, and format of data, and may include elements such as title, author, subject, and keywords. Descriptive metadata helps to identify and describe the content of data and can be used to improve the discoverability of data through search engines and other tools.
2. **Administrative metadata:** This type of metadata provides information about the management and technical characteristics of data, and may include elements such as file format, size, and creation date. Administrative metadata helps to manage and maintain data over time and can be used to support data governance and preservation.
3. **Structural metadata:** This type of metadata provides information about the relationships and organization of data, and may include elements such as links, tables of contents, and indices. Structural metadata helps to organize and connect data and can be used to facilitate the navigation and discovery of data.
4. **Provenance metadata:** This type of metadata provides information about the history and origin of data, and may include elements such as the creator, date of creation, and sources of data. Provenance metadata helps to provide context and credibility to data and can be used to support data governance and preservation.
5. **Rights metadata:** This type of metadata provides information about the ownership, licensing, and access controls of data, and may include elements such as copyright, permissions, and terms of use. Rights metadata

helps to manage and protect the intellectual property rights of data and can be used to support data governance and compliance.

6. **Educational metadata:** This type of metadata provides information about the educational value and learning objectives of data, and may include elements such as learning outcomes, educational levels, and competencies. Educational metadata can be used to support the discovery and use of educational resources and to support the design and evaluation of learning environments.

### Examples of Metadata

Metadata is data that provides information about other data. Here are a few examples of metadata:

1. **File metadata:** This includes information about a file, such as its name, size, type, and creation date.
2. **Image metadata:** This includes information about an image, such as its resolution, color depth, and camera settings.
3. **Music metadata:** This includes information about a piece of music, such as its title, artist, album, and genre.
4. **Video metadata:** This includes information about a video, such as its length, resolution, and frame rate.
5. **Document metadata:** This includes information about a document, such as its author, title, and creation date.
6. **Database metadata:** This includes information about a database, such as its structure, tables, and fields.
7. **Web metadata:** This includes information about a web page, such as its title, keywords, and description.

### Metadata Repository

A metadata repository should contain the following:

1. **A description of the data warehouse structure:** It includes the warehouse schema, view, dimensions, hierarchies, and derived data definitions, as well as data mart locations and contents.
2. **Operational metadata:** It includes data lineage (history of migrated data and the sequence of transformations applied to it), currency of data (active, archived, or purged), and monitoring information (warehouse usage statistics, error reports, and audit trails).
3. **The algorithm used for summarization:** It includes measure and dimension definition algorithms, data on granularity, partitions, subject area, aggregation, summarization, and predefined queries and reports.
4. **Mapping from the operational environment to the data warehouse:** It includes source databases and their contents, gateway descriptions, data partitions, data extraction, cleaning, transformation rules, and defaults, data refresh and purging rules, and security (user authorization and access control).
5. **Data related to system performance:** It includes indices and profiles that improve data access and retrival performance, in addition to rules for the timing and scheduling of refresh, update, and replication cycles.
6. **Business metadata:** It includes business terms and definitions, data ownership information and charging policies.

### Benefits of Metadata Repository

A metadata repository is a centralized database or system that is used to store and manage metadata. Some of the benefits of using a metadata repository include:

1. **Improved data quality:** A metadata repository can help ensure that metadata is consistently structured and accurate, which can improve the overall quality of the data.
2. **Increased data accessibility:** A metadata repository can make it easier for users to access and understand the data, by providing context and information about the data.
3. **Enhanced data integration:** A metadata repository can facilitate data integration by providing a common place to store and manage metadata from multiple sources.

4. **Improved data governance:** A metadata repository can help enforce metadata standards and policies, making it easier to ensure that data is being used and managed appropriately.
5. **Enhanced data security:** A metadata repository can help protect the privacy and security of metadata, by providing controls to restrict access to sensitive or confidential information.

Metadata repositories can provide many benefits in terms of improving the quality, accessibility, and management of data.

**Challenges for Metadata Management**

There are several challenges that can arise when managing metadata:

1. **Lack of standardization:** Different organizations or systems may use different standards or conventions for metadata, which can make it difficult to effectively manage metadata across different sources.
2. **Data quality:** Poorly structured or incorrect metadata can lead to problems with data quality, making it more difficult to use and understand the data.
3. **Data integration:** When integrating data from multiple sources, it can be challenging to ensure that the metadata is consistent and aligned across the different sources.
4. **Data governance:** Establishing and enforcing metadata standards and policies can be difficult, especially in large organizations with multiple stakeholders.
5. **Data security:** Ensuring the security and privacy of metadata can be a challenge, especially when working with sensitive or confidential information.

**Metadata Management Software:**

Software for managing metadata makes it easier to assess, curate, collect, and store metadata. In order to enable data monitoring and accountability, organizations should automate data management. Examples of this kind of software include the following:

- **SAP Power Designer by SAP:** This data management system has a good level of stability. It is recognised for its ability to serve as a platform for model testing.
- **SAP Information Steward by SAP:** This solution's data insights make it valuable.
- **IBM InfoSphere Information Governance Catalog by IBM:** The ability to use Open IGC to build unique assets and data lineages is a key feature of this system.
- **Alation Data Catalog by Alation:** This provides a user-friendly, intuitive interface. It is valued for the queries it can publish in Standard Query Language (SQL).
- **Informatica Enterprise Data Catalog by Informatica:** The technology used by this solution, which can both scan and gather information from diverse sources, is highly respected.

# MULTIDIMENSIONAL DATA MODEL



Data cube Data warehouses and OLAP tools are based on a multidimensional data model.

## What is a data Cube

A data cube allows data to be modelled 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. Eg. AllElectronics may create a sales data warehouse in order to keep records of store's sales with respect to dimensions time, item, branch and location.

→ These dimensions allow the store to keep track of things like monthly sales of the items and branch and locations at which the items were sold.

→ Each dimension may have a table associated with it called dimension table.

→ Dimension table can be specified by users or experts, or automatically generated and adjusted based on data distribution

→ Facts are numeric measure.

→ They are like quantities by which we want to analyze relationship between dimensions.

→ Eg. data sold (sales dollars sold (sales amount in dollars), units sold (no. of units sold), amount

budgeted.

→ The fact table contains the names of the facts, or measures, as well as keys to each of the related dimension tables.

→ Although it is assumed that the cube is a 3-D geometric structure, but in data warehousing the data cube is n-dimensional.

Eg.

A spreadsheet for sales data from AllElectronics

2-D View (According to Time and item)

location = "Vancouver"

| Time (quarter) | Item (type) (dollar_sold). | | | |
|---|---|---|---|---|
| | 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 |

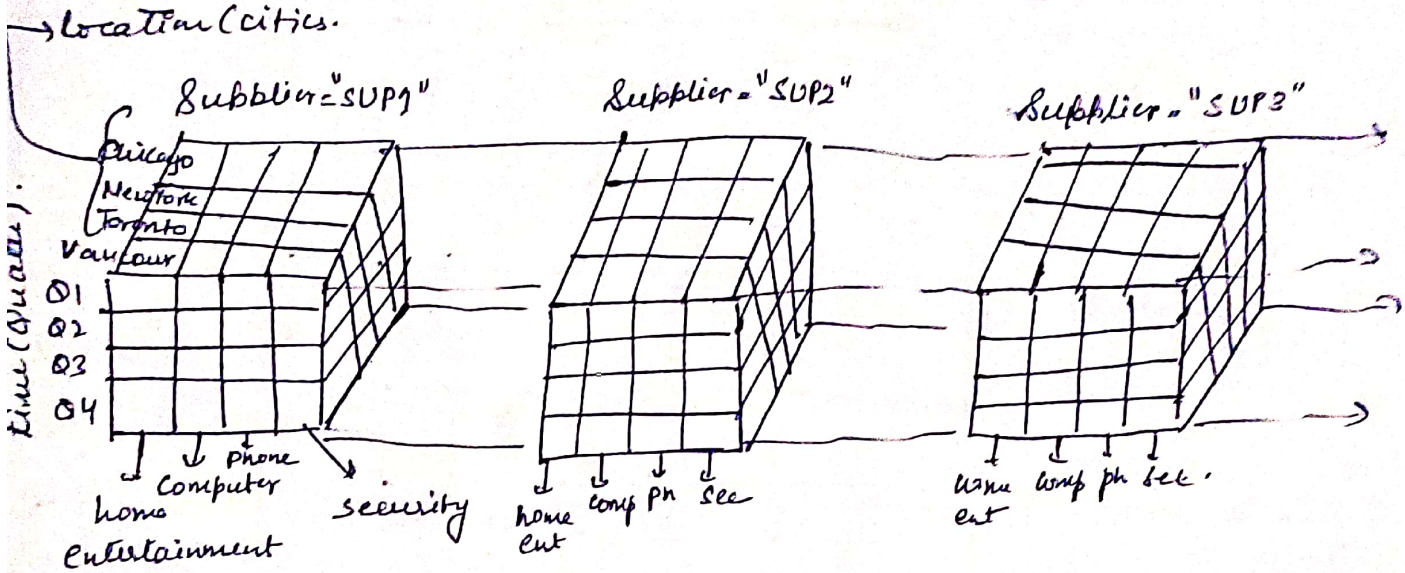
→ The above table represents AllElectronics sales data for items sold per quarter in the city of Vancouver.

→ The 2-D table represents the sales for Vancouver with respect to the time dimension (organized in quarters) and item dimension (organized according to the type of items sold). The fact or measure displayed is dollar_sold (in thousands).

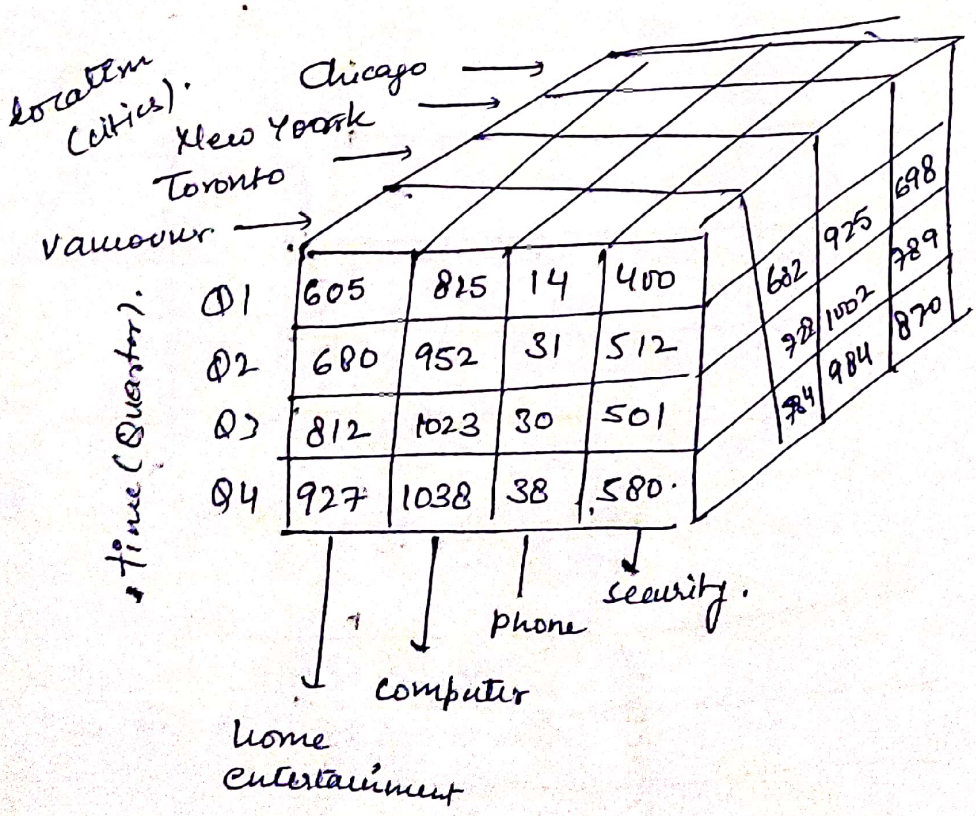
→ If the data we want to view is of 3D i.e. according to time and item as well as the location, for the cities of Chicago, NewYork, Toronto and Vancouver.

→ The table above represents a series of 2-D table.

## Representation of 3-D Table in cube form




## Representation of 3-D Table in cube form



# 3-D VIEW OF SALES DATA FOR ALLELECTRONICS, According to time, item and location.

location = "Chicago"

| Time | Home ent | Comp. | phone | sec. |
|---|---|---|---|---|
| Q1 | 854 | 882 | 89 | 623 |
| Q2 | 943 | 890 | 64 | 698 |
| Q3 | 1032 | 924 | 59 | 789 |
| Q4 | 1129 | 992 | 63 | 870 |

location = "NewYork"

| Time | Home ent. | Comp | phone | sec. |
|---|---|---|---|---|
| Q1 | 1087 | 968 | 38 | 872 |
| Q2 | 1130 | 1024 | 41 | 925 |
| Q3 | 1002 | 1048 | 45 | 1002 |
| Q4 | 984 | 1091 | 54 | 984 |

location = "Toronto"

| Time | Home ent. | Comp | pho. | Sec |
|---|---|---|---|---|
| Q1 | 818 | 746 | 43 | 591 |
| Q2 | 894 | 769 | 52 | 682 |
| Q3 | 940 | 795 | 58 | 728 |
| Q4 | 978 | 864 | 59 | 784 |

location = "Vancouver"

| Time | Hom ent. | Comb | Ph | Se |
|---|---|---|---|---|
| Q1 | 605 | 825 | 14 | 400 |
| Q2 | 680 | 952 | 31 | 512 |
| Q3 | 812 | 1023 | 30 | 501 |
| Q4 | 927 | 1038 | 38 | 580 |

Suppose we want to view sales data with and additional fourth dimension such as supplier.

→ Viewing in 4-D becomes tricky. However, we can think of a 4-D cube as being a series of 3-D cube shown in above figure.

→ If we continue this way, we may display n-dimensional data as a series of (n-1) dimensional "cubes".

→ The data cube is a metaphor for multidimensional data storage.

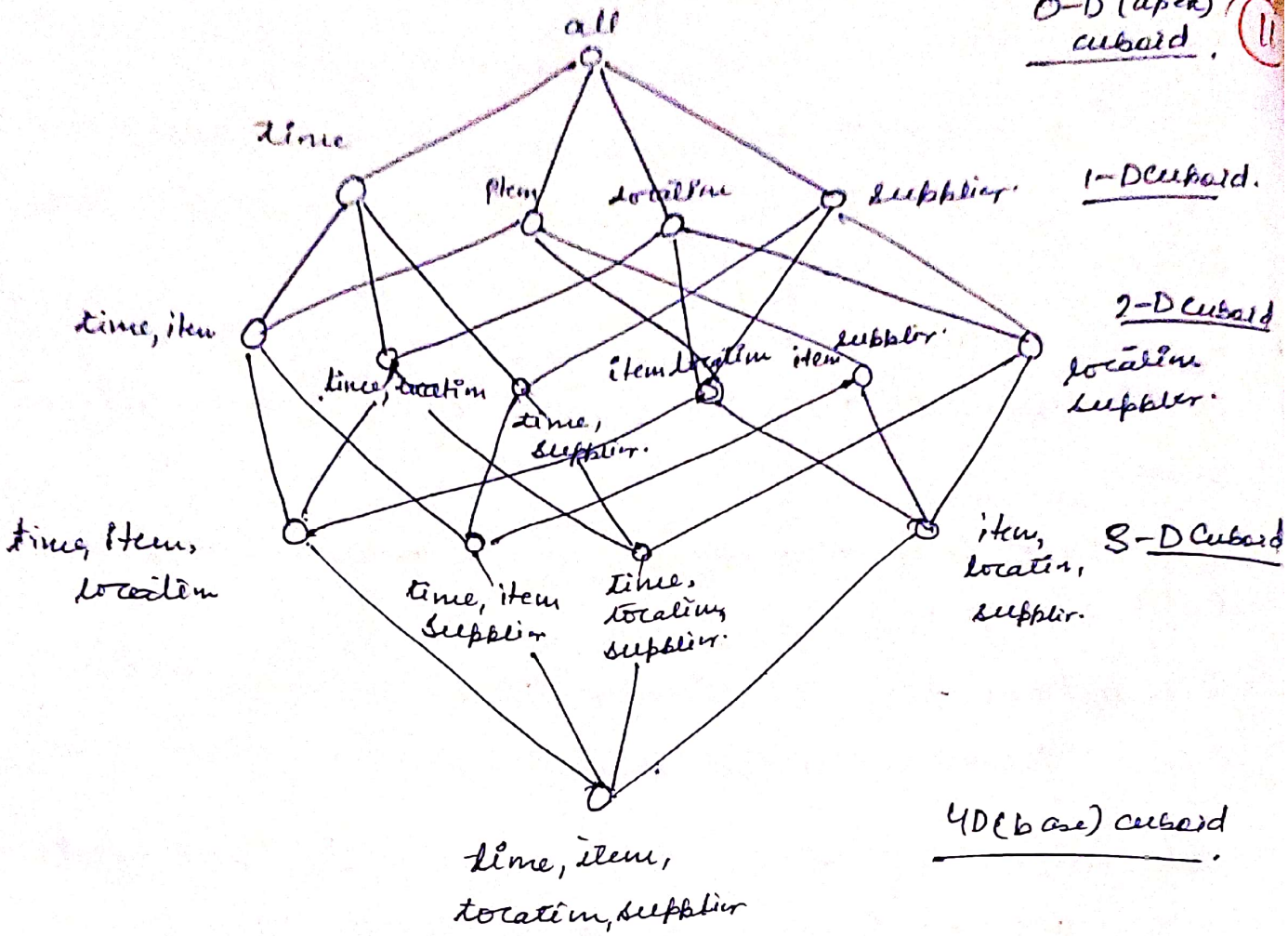
→ The actual physical storage of such data may differ from its logical representation.

→ The important thing to remember is that data cubes are n-dimensional and do not confine data to 3-D.

→ In data warehousing, a data cube are often referred to as cuboid.

→ Given a set of dimensions, we can generate a cuboid for each of the possible subsets of the given dimensions.

→ The result would form a lattice of cuboids is then referred to as a data cube.




Lattice of cuboids, making up a 4-D data cube for time, item, location and supplier

(Each cuboid represents and different degree of ~~subtraction~~ summarization).

→ Here, the cuboid that holds the lowest level of summarization is called the base cuboid.
(Eg. From fig → time, item location supplier).

→ The 0-D cuboid, hold the highest level of summarization is called the apex cuboid.

(Eg → Total sales, or dollar-sold, summarized over all four dimensions.

# SCHEMAS FOR MULTIDIMENSIONAL DATA MODELS



→ The entity - relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them.

→ Such data model is appropriate for online transaction processing.

→ Thus schema is a logical description of the entire database. It includes the name and description of all records types including all associated data-items and aggregates.

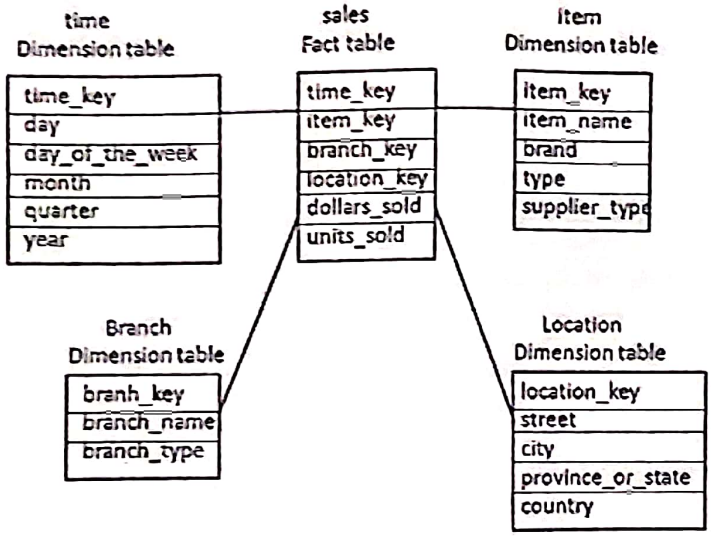
→ A data warehouse, however requires a concise, subject-oriented schema that facilitates online data analysis.

→ The most popular data model for a data warehouse is a multidimensional model, which can exist in the form of a star schema, a snowflake schema or a fact constellation schema.

## SCHEMAS FOR MULTIDIMENSIONAL DATABASES

### 1. Star Schema

- The most common modeling paradigm is the star schema, in which the data warehouse contains, (1) a large central table **(fact table)** containing the bulk of the data with no redundancy and (2) 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.
- Each dimension in a star schema is represented with only one-dimension table.
- This dimension table contains the set of attributes.
- The following 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.

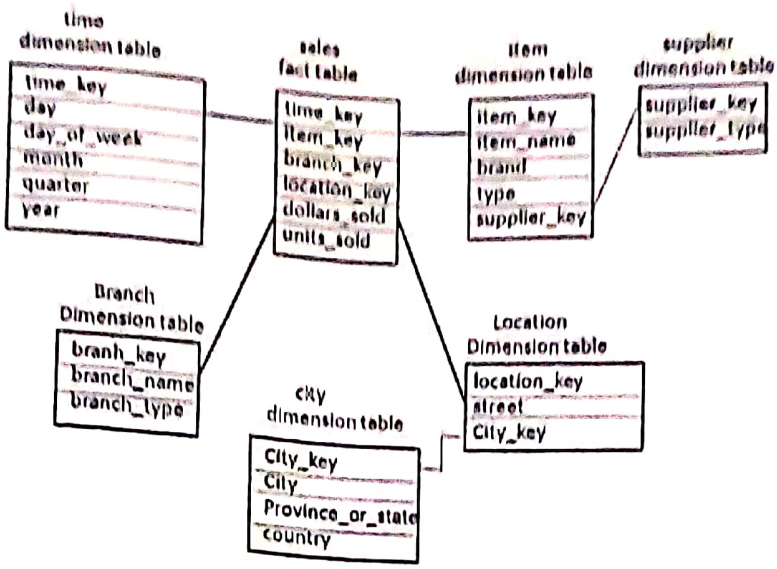



**Note** – Each dimension has only one dimension table and each table holds a set of attributes. For example, the location dimension table contains the attribute set {location_key, street, city, province_or_state,country}. This constraint may cause data redundancy. For example, "Vancouver" and "Victoria" both the cities are in the Canadian province of British Columbia. The entries for such cities may cause data redundancy along the attributes province_or_state and country.

### 2. Snowflake Schema

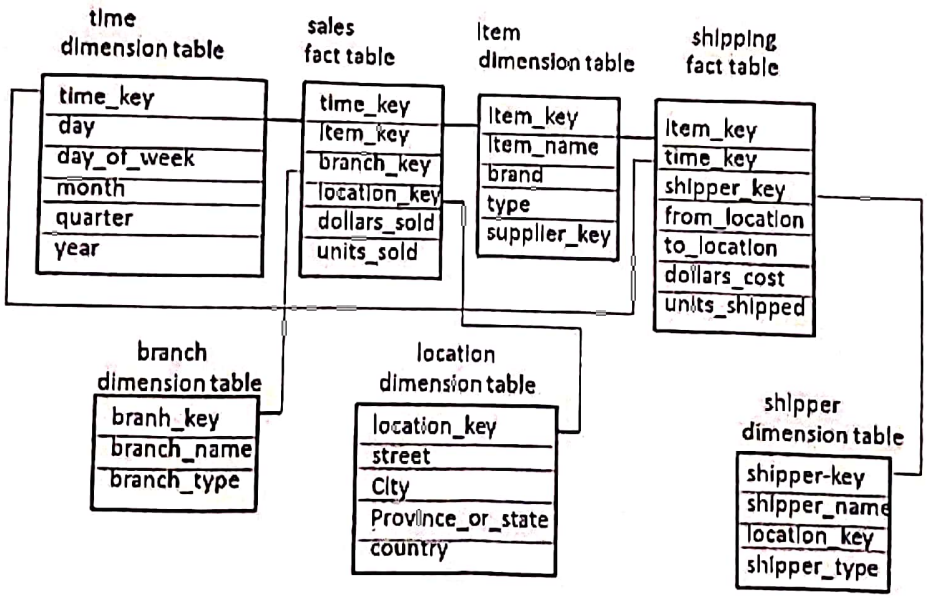
- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema is normalized. For example, the item dimension table in a star schema is normalized and split into two dimension tables, namely item and supplier table. The resulting schema graph forms a shape similar to a snowflake.
- Now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key.
- The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier_key and supplier_type.

**Note** – Due to normalization in the Snowflake schema, the redundancy is reduced and therefore, it becomes easy to maintain and save storage space.




### 3. 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.
- The following diagram shows two fact tables, namely sales and shipping.
- The sales fact table is the same as that in the star schema.
- The shipping fact table has the five dimensions, namely item_key, time_key, shipper_key, from_location, to_location.
- The shipping fact table also contains two measures, namely dollars sold and units sold.
- It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.



**NOTE**

- In data warehousing, there is a distinction between a data warehouse and a data mart.
- A data warehouse collects information about subjects that span the entire organization, such as customers, items, sales, assets, and personnel, and thus its scope is enterprise-wide.
- For data warehouses, the fact constellation schema is commonly used, since it can model multiple, interrelated subjects.
- A **data mart**, on the other hand, is a department subset of the data warehouse that focuses on selected subjects, and thus its scope is department-wise.
- For data marts the star and snowflake schema is commonly used, since both are geared towards modeling single subjects, although the star schema is more popular and efficient.

## THE ROLE OF CONCEPT HIERARCHIES: DIMENSIONS

→ The concept hierarchy defines a sequence of mapping 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.

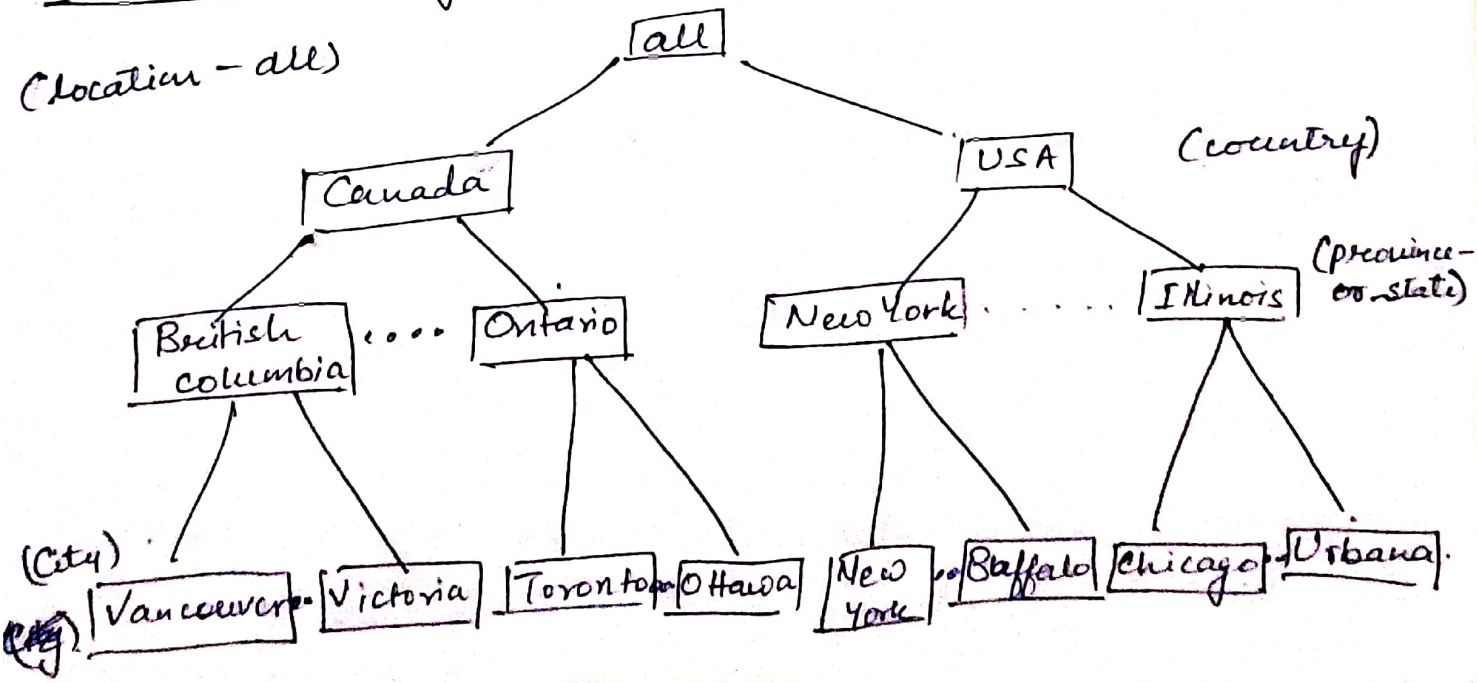
→ Each city, however, can be ~~viewed~~ mapped to the province or state to which it belongs.

→ For example, Vancouver can be mapped to British Columbia, and Chicago to Illinois.

→ The provinces and state 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 location, mapping a set of low-level concepts (ie. cities) to higher-level, more general concepts (ie. ~~concepts~~ countries)

A concept hierarchy for location (Eg).

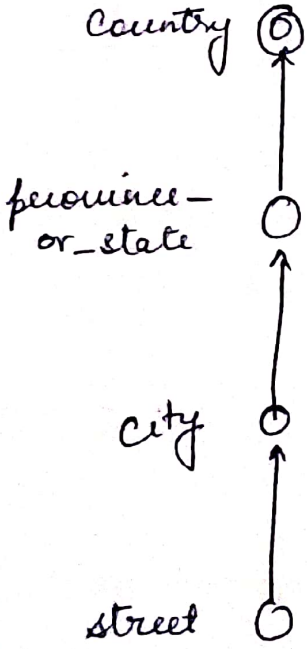


→ Many concept hierarchies are implicit within the database schema.

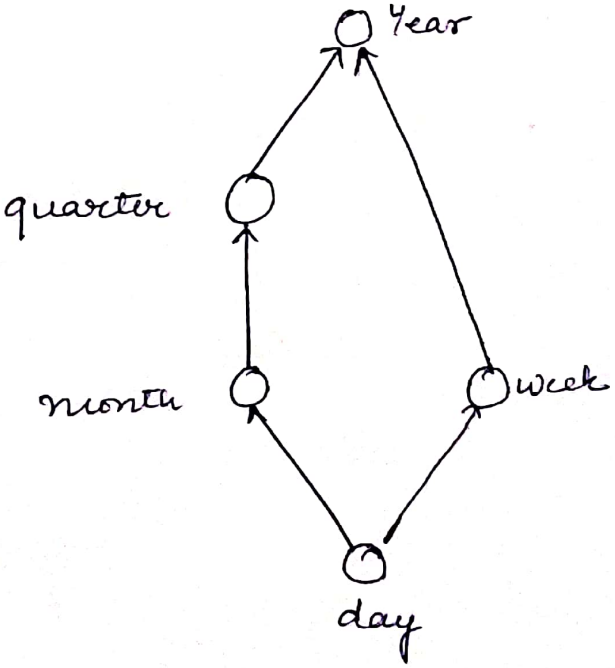
→ For example, suppose that the dimension location is described by the attributes number, street, city, province-or-state, zip-code and country.

→ These attributes are related by a total order, forming a concept hierarchy such as "street < city < province_or_state < country".

## Hierarchical and lattice structures of attributes in warehouse dimensions:




a hierarchy for location




a lattice for time

→ Alternatively, the attributes of a dimension may be organized in a partial order, forming a lattice. 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".

→ A concept hierarchy that is a total or partial order among attributes in a database schema is called a schema hierarchy (12)

→ Concept hierarchies that are common to many applications (e.g. for time) may be predefined in the data mining system.

→ Concept hierarchies may also be defined by discretizing a grouping values for a given dimension or attribute, resulting in a set-grouping hierarchy. Eg.

→ A total or partial order can be defined among groups of values. Eg. for dimension price, where and interval ($X...$Y) denotes the range from $X(exclusive) to $Y(inclusive)

→ There may be more than one concept hierarchy for a given attribute or dimension, based on different user viewpoints. For instance, a user prefer to organize price by defining range 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.

## OLAP OPERATIONS IN THE MULTIDIMENSIONAL DATA MODEL

In the multidimensional model, the records are organized into various dimensions, and each dimension includes multiple levels of abstraction described by concept hierarchies.

This organization supports users with the flexibility to view data from various perspectives. A number of OLAP data cube operations exist to demonstrate these different views, allowing interactive queries and a search of the record at hand. Hence, OLAP supports a user-friendly environment for interactive data analysis.

Consider the OLAP operations which are to be performed on multidimensional data. The figure shows data cubes for sales of a shop. The cube contains the dimensions, location, and time and item, where the **location** is aggregated with regard to city values, **time** is aggregated with respect to quarters, and an **item** is aggregated with respect to item types.

### 1. Roll-Up

The roll-up operation **(also known as drill-up or aggregation operation)** performs aggregation on a data cube, by climbing down concept hierarchies, i.e., dimension reduction. Roll-up is like **zooming-out** on the data cubes. Figure shows the result of roll-up operations performed on the dimension location. The hierarchy for the location is defined as the Order Street, city, province, or state, country. The roll-up operation aggregates the data by ascending the location hierarchy from the level of the city to the level of the country.

When a roll-up is performed by dimensions reduction, one or more dimensions are removed from the cube. For example, consider a sales data cube having two dimensions, location and time. Roll-up may be performed by removing, the time dimensions, appearing in an aggregation of the total sales by location, relatively than by location and by time.

Example,

**Consider the following cubes illustrating the temperature of certain days recorded weekly:**

| **Temperature** | 64 | 65 | 68 | 69 | 70 | 71 | 72 | 75 | 80 | 81 | 83 | 85 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Week1** | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **Week2** | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 0 | 1 | 0 | 0 |

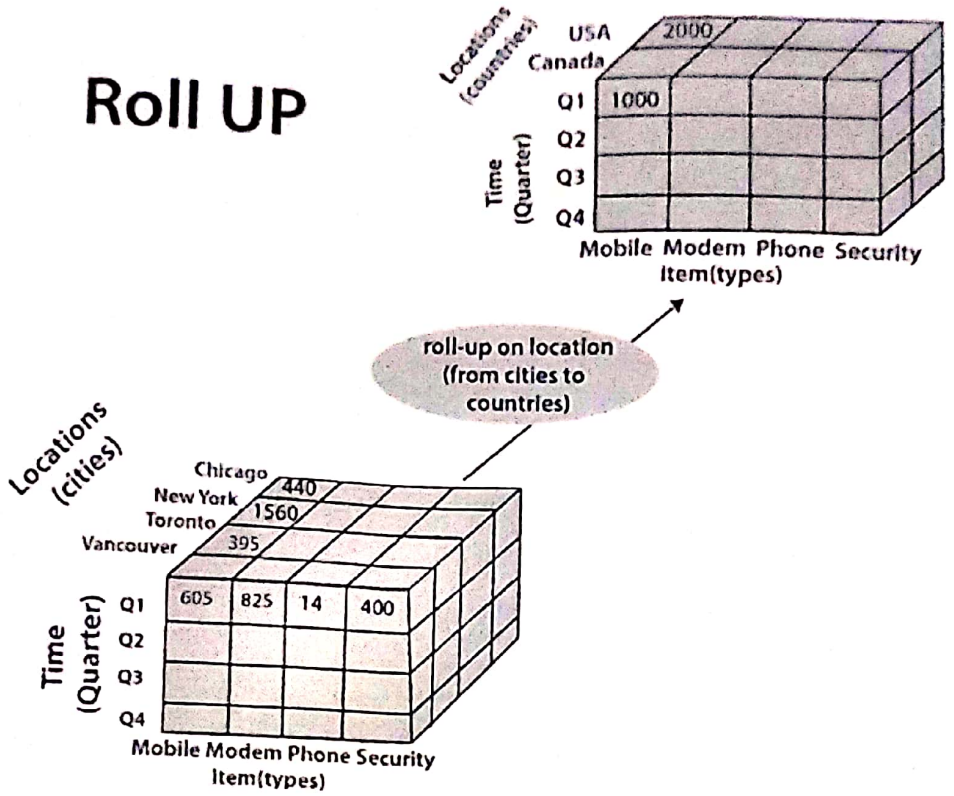
Consider that we want to set up levels (hot (80-85), mild (70-75), cool (64-69)) in temperature from the above cubes. To do this, we have to group columns and add up the values according to the concept hierarchies. This operation is known as a roll-up.

By doing this, we contain the following cube:

| **Temperature** | **cool** | **mild** | **hot** |
|---|---|---|---|
| **Week1** | 2 | 1 | 1 |
| **Week2** | 2 | 1 | 1 |

**The roll-up operation groups the information by levels of temperature.**

The following diagram illustrates how roll-up works.




**2. Drill-Down**

The drill-down operation **(also called roll-down)** is the reverse operation of **roll-up**. Drill-down is like **zooming in** on the data cube. It navigates from less detailed records to more detailed data. Drill-down can be performed by either **stepping down** a concept hierarchy for a dimension or adding additional dimensions.

Figure shows a drill-down operation performed on the dimension time by stepping down a concept hierarchy which is defined as day, month, quarter, and year. Drill-down appears by descending the time hierarchy from the level of the quarter to a more detailed level of the month.

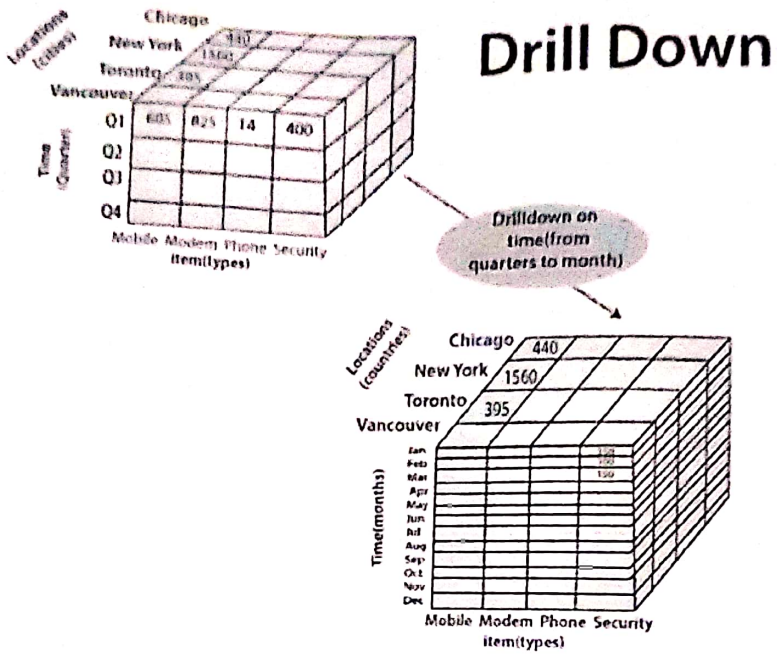
Because a drill-down adds more details to the given data, it can also be performed by adding a new dimension to a cube. For example, a drill-down on the central cubes of the figure can occur by introducing an additional dimension, such as a customer group.

**Example**

Drill-down adds more details to the given data

| Temperature | cool | mild | hot |
|---|---|---|---|
| Day 1 | 0 | 0 | 0 |
| Day 2 | 0 | 0 | 0 |
| Day 3 | 0 | 0 | 1 |
| Day 4 | 0 | 1 | 0 |
| Day 5 | 1 | 0 | 0 |
| Day 6 | 0 | 0 | 0 |
| Day 7 | 1 | 0 | 0 |
| Day 8 | 0 | 0 | 0 |
| Day 9 | 1 | 0 | 0 |
| Day 10 | 0 | 1 | 0 |
| Day 11 | 0 | 1 | 0 |
| Day 12 | 0 | 1 | 0 |
| Day 13 | 0 | 0 | 1 |
| Day 14 | 0 | 0 | 0 |

The following diagram illustrates how Drill-down works.




### 3. Slice

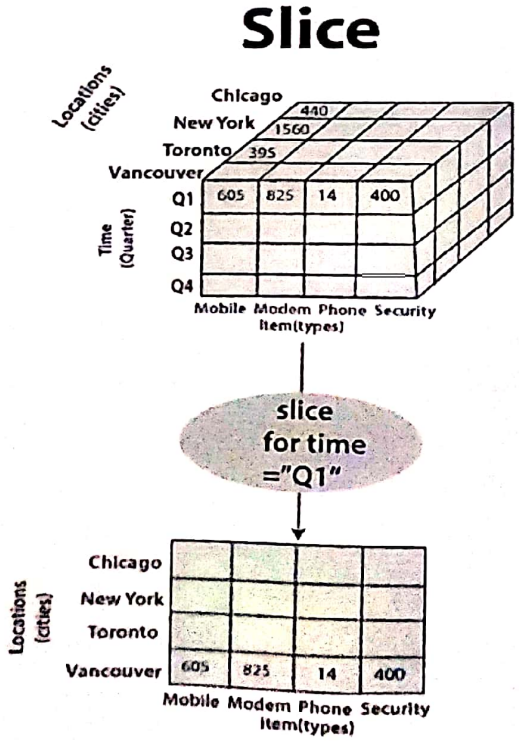
A **slice** is a subset of the cubes corresponding to a single value for one or more members of the dimension. For example, a slice operation is executed when the customer wants a selection on one dimension of a three-dimensional cube resulting in a two-dimensional site. So, the Slice operations perform a selection on one dimension of the given cube, thus resulting in a subcube.

For example, if we make the selection, temperature=cool we will obtain the following cube:

| Temperature | cool |
|---|---|
| Day 1 | 0 |
| Day 2 | 0 |
| Day 3 | 0 |
| Day 4 | 0 |
| Day 5 | 1 |

| | |
|---|---|
| Day 6 | 1 |
| Day 7 | 1 |
| Day 8 | 1 |
| Day 9 | 1 |
| Day 11 | 0 |
| Day 12 | 0 |
| Day 13 | 0 |
| Day 14 | 0 |

**The following diagram illustrates how Slice works.**




Here Slice is functioning for the dimensions "time" using the criterion time = "Q1".

It will form a new sub-cubes by selecting one or more dimensions.

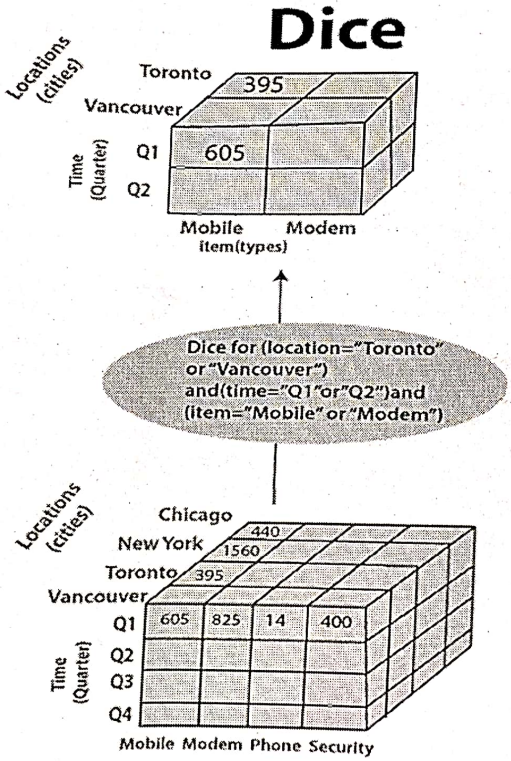
**4. Dice**

The dice operation describes a subcube by operating a selection on two or more dimensions.

**For example**, Implement the selection (time = day 3 OR time = day 4) AND (temperature = cool OR temperature = hot) to the original cubes we get the following subcube (still two-dimensional)

| Temperature | cool | hot |
|---|---|---|
| Day 3 | 0 | 1 |
| Day 4 | 0 | 0 |

Consider the following diagram, which shows the dice operations.

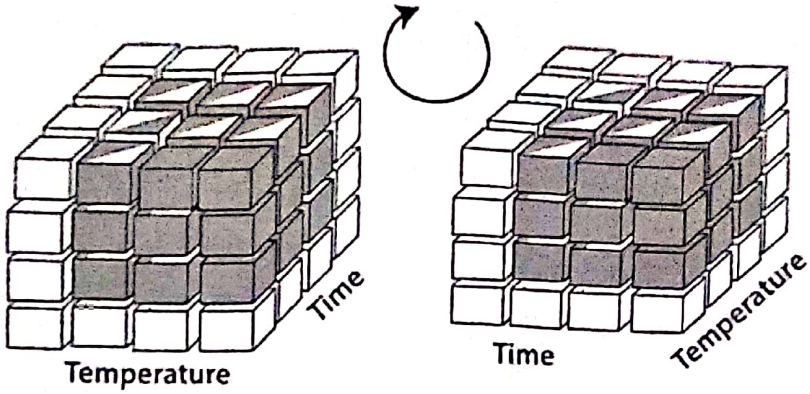



The dice operation on the cubes based on the following selection criteria involves three dimensions.

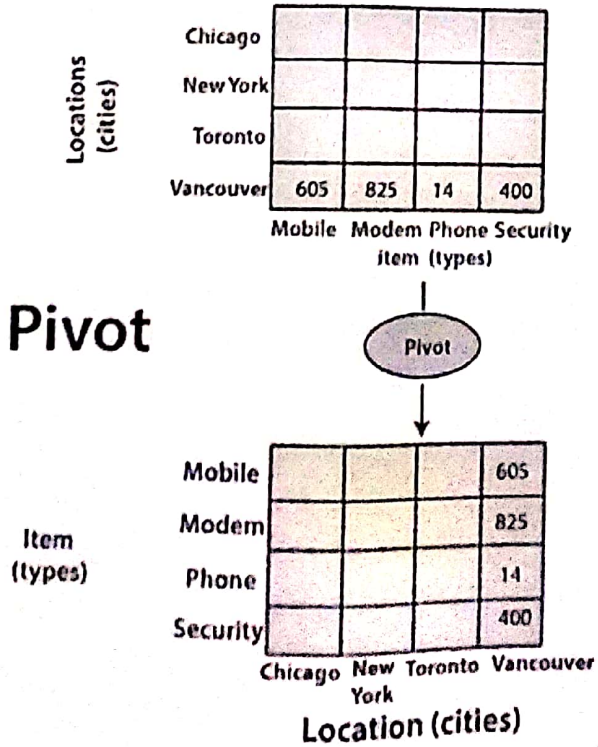
- (location = "Toronto" or "Vancouver")
- (time = "Q1" or "Q2")
- (item =" Mobile" or "Modem")

### 5. Pivot

The pivot operation is also called a rotation. Pivot is a visualization operation that rotates the data axes in view to provide an alternative presentation of the data. It may contain swapping the rows and columns or moving one of the row-dimensions into the column dimensions.




Consider the following diagram, which shows the pivot operation.



**Other OLAP Operations**

executes queries containing more than one fact table. The drill-through operations make use of relational SQL facilitate drilling through the bottom level of a data cube down to its back-end relational tables.

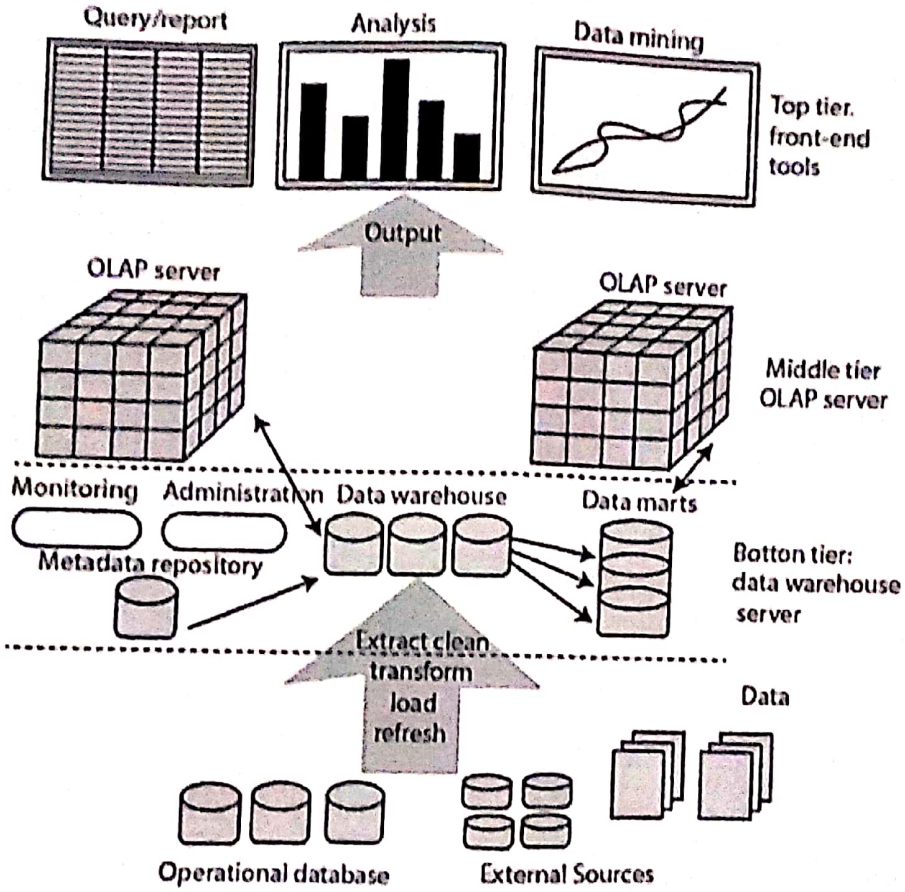
Other OLAP operations may contain ranking the top-N or bottom-N elements in lists, as well as calculating moving averages, growth rates, and interests, internal rates of returns, depreciation, currency conversions, and statistical tasks.

OLAP offers analytical modeling capabilities, containing a calculation engine for determining ratios, variance, etc., and for computing measures across various dimensions. It can generate summarization, aggregation, and hierarchies at each granularity level and at every dimensions intersection. OLAP also provide functional models for forecasting, trend analysis, and statistical analysis. In this context, the OLAP engine is a powerful data analysis tool.

## DATA WAREHOUSE ARCHITECTURE: 3 – TIER ARCHITECTURE

Data Warehouse is referred to the data repository that is maintained separately from the organization's operational data. **Multi-Tier Data Warehouse Architecture consists of the following components:**

1. Bottom Tier
2. Middle Tier
3. Top Tier




Three-Tier Data Warehouse Architecture

**Bottom Tier(Data sources and data storage) :**

1. The bottom Tier usually consists of Data Sources and Data Storage.
2. It is a warehouse database server. For Example RDBMS.
3. In Bottom Tier, using the application program interface(called gateways), data is extracted from operational and external sources.
4. Application Program Interface likes ODBC(Open Database Connection), OLE-DB(Open-Linking and Embedding for Database), JDBC(Java Database Connection) is supported.
5. ETL stands for Extract, Transform, and Load.
   Several popular ETL tools include:
   I. IBM Infosphere
   II. Informatica
   III. Confluent

IV. Microsoft SSIS
V. Snaplogic
VI. Alooma

**Middle Tier :**

The middle tier is an OLAP server that is typically implemented using either :
A relational OLAP (ROLAP) model (i.e., an extended relational DBMS that maps operations from standard data to standard data); or A multidimensional OLAP (MOLAP) model (ie, a special purpose server that directly implements multidimensional data and operations).
OLAP server models come in three different categories, including:

1. **ROLAP:** A relational database is not converted into a multidimensional database; rather, a relational database is actively broken down into several dimensions as part of relational online analytical processing(ROLAP). This is used when everything that is contained in the repository is a relational database system.
2. **MOLAP:** A different type of online analytical processing called multidimensional online analytical processing(MOLAP) includes directories and catalogs that are immediately integrated into its multidimensional database system. This is used when all that is contained in the repository is the multidimensional database system.
3. **HOLAP:** A combination of relational and multidimensional online analytical processing paradigms is hybrid online analytical processing(HOLAP). HOLAP is the ideal option for a seamless functional flow across the database systems when the repository houses both the relational database management system and the multidimensional database management system.

**Top Tier :**

The top tier is a front-end client layer, which includes query and reporting tools, analysis tools, and/or data mining tools (eg, trend analysis, prediction, etc.).
Here are a few Top Tier tools that are often used:

- SAP BW
- SAS Business Intelligence
- IBM Cognos
- Crystal Reports
- Microsoft BI Platform

**Advantages of Multi-Tier Architecture of Data warehouse**

1. **Scalability**: Various components can be added, deleted, or updated in accordance with the data warehouse's shifting needs and specifications.
2. **Better Performance**: The several layers enable parallel and efficient processing, which enhances performance and reaction times.
3. **Modularity**: The architecture supports modular design, which facilitates the creation, testing, and deployment of separate components.
4. **Security**: The data warehouse's overall security can be improved by applying various security measures to various layers.
5. **Improved Resource Management**: Different tiers can be tuned to use the proper hardware resources, cutting expenses overall and increasing effectiveness.
6. **Easier Maintenance**: Maintenance is simpler because individual components can be updated or maintained without affecting the data warehouse as a whole.
7. **Improved Reliability**: Using many tiers can offer redundancy and failover capabilities, enhancing the data warehouse's overall reliability.

## DATA WAREHOUSE MODELS

From the perspective of data warehouse architecture, we have the following data warehouse models –

- Virtual Warehouse
- Data mart
- Enterprise Warehouse

**Enterprise Warehouse:-**

- An enterprise warehouse collects all information topics spread throughout the organization.
- It provides corporate-wide data integration, typically from one or several operational systems or external information providers, and is cross-functional in scope.
- It usually contains detailed data as well as summarized data and can range in size from a few gigabytes to hundreds of gigabytes, terabytes, or beyond. Can be an enterprise data warehouse.
- The traditional mainframe, computer super server, or parallel architecture has been implemented on platforms. This requires extensive commercial modeling and may take years to design and manufacture.

**Data Mart:-**

- A data mart contains a subset of corporate-wide data that is important to a specific group of users.
- The scope is limited to specific selected subjects.
- For example, a marketing data mart may limit its topics to customers, goods, and sales.
- The data contained in the data marts are summarized. Data marts are typically applied to low-cost departmental servers that are Unix/Linux or Windows-based.
- The implementation cycle of a data mart is more likely to be measured in weeks rather than months or years. However, it can be in the long run, complex integration is involved in its design and planning were not enterprise-wide.

**Virtual Warehouse:-**

- A virtual warehouse is a group of views on an operational database.
- For efficient query processing, only a few possible summary views can be physical.
- Creating a virtual warehouse is easy, but requires additional capacity on operational database servers.

## WHAT ARE THE PROS AND CONS OF THE TOP-DOWN AND BOTTOM-UP APPROACHES TO DATA WAREHOUSE DEVELOPMENT:



→ The top-down development of an enterprise warehouse serves as a systematic solution and minimizes integration problems.

→ However, it is expensive, takes a long time to develop, and lacks flexibility due to the difficulty in achieving consistency and consensus for a common data model for the entire organization.

→ The bottom-up approach to the design, development and deployment of independent data marts provides flexibility, low cost, and rapid return of investment.

→ It, however, can lead to problems when integrating various disparate data marts into a consistent enterprise data warehouse.

→ A recommended method for the development of data warehouse systems is to implement the warehouses in an incremental and evolutionary manner.

→ First, a high-level corporate data model is defined within a reasonably short period (such as one or two months) that provides a corporate-wide, consistent, integrated view of data among different subjects and potential usage.

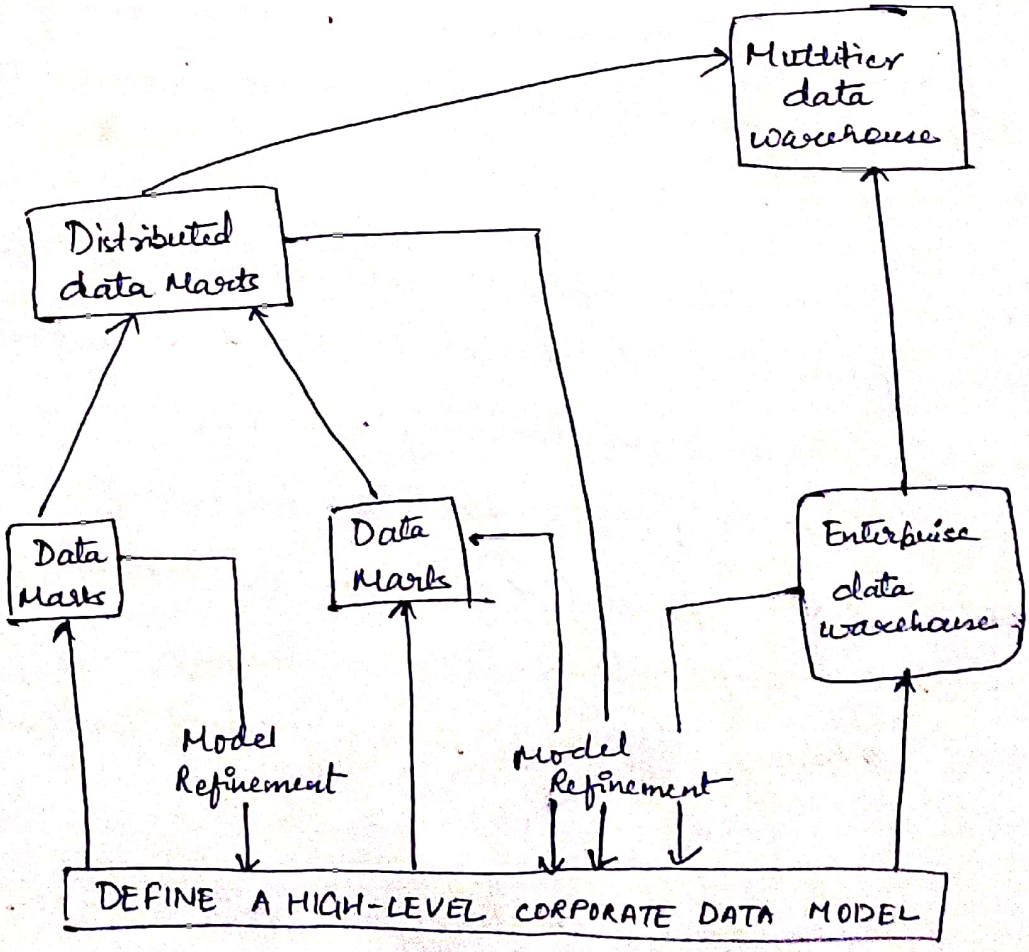
→ This high-level model, although it will need to be refined in the further development of enterprise data warehouses and departmental data marts, will greatly reduce future integration problems.

→ second, independent datamarts can be implemented in parallel with the enterprise based on the same corporate data model set noted before.

→ Third, distributed data marts can be constructed to integrate different data marts via hub servers.

→ Finally, a multitier data warehouse is constructed where the enterprise warehouse is the sole custodian of all warehouse data, which is then distributed to the various dependent data marts.

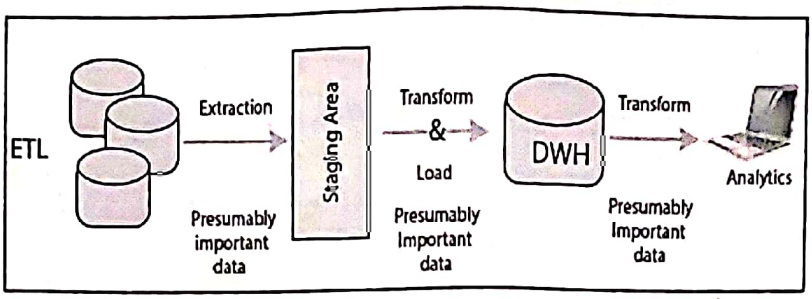
A recommended approach for data warehouse development



## ETL (EXTRACT, TRANSFORM, AND LOAD) PROCESS

### What is ETL?

- The mechanism of extracting information from source systems and bringing it into the data warehouse is commonly called ETL, which stands for **Extraction, Transformation, and Loading.**
- The ETL process requires active inputs from various stakeholders, including developers, analysts, testers, top executives, and is technically challenging.
- To maintain its value as a tool for decision-makers, Data warehouse technique needs to change with business changes. ETL is a recurring method (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.
- ETL consumes about 50% of the effort in data warehouse.




### How ETL work?

ETL consists of three separate phases:

1. **Extraction**

- Extraction is the operation of extracting information from a source system for further use in a data warehouse environment. This is the first stage of the ETL process.
- The extraction process is often one of the most time-consuming tasks in the ETL.
- The source systems might be complicated and poorly documented, and thus determining which data needs to be extracted can be difficult.
- The data has to be extracted several times in a periodic manner to supply all changed data to the warehouse and keep it up-to-date.

### Cleansing

The cleansing stage is crucial in a data warehouse technique because it is supposed to improve data quality. This is done before the transformation phase. The primary data cleansing features found in ETL tools are rectification and homogenization. They use specific dictionaries to rectify typing mistakes and to recognize synonyms, as well as rule-based cleansing to enforce domain-specific rules and define appropriate associations between values.

The following examples show the essentials of data cleaning:

- If an enterprise wishes to contact its users or its suppliers, a complete, accurate, and up-to-date list of contact addresses, email addresses, and telephone numbers must be available.
- If a client or supplier calls, the staff responding should be quickly able to find the person in the enterprise database, but this needs the caller's name or his/her company name is listed in the database.
- If a user appears in the databases with two or more slightly different names or different account numbers, it becomes difficult to update the customer's information.

## 2. Transformation

Transformation is the core of the reconciliation phase. It converts records from its operational source format into a particular data warehouse format. If we implement a three-layer architecture, this phase outputs our reconciled data layer.

Basic Steps:

1. Selection: Identify tables, and records on which transformation is to be made
2. Splitting/Joining: Slipping the table to generalize the data. Split to transform (parallelization).
3. Conversion: Data format, platform, and representation are different so, this process converts them into a consistent format.

Example 1: Consistent format

| Feild Format | Field data |
|---|---|
| First Middle Surname ⟶ | Mohd Ibraheem Khan |
| Surname Middle first ⟶ | Khan Ibraheem Mohd |
| Surname First Middle. ⟶ | Khan Mohd. Ibraheem. |

Example 2: Transition of dissimilar code into a standered code

F/No-2
F-2
FL.NO.2
FL.2
PL/NO-2
FLAT #2
} FLAT NO.2

4. Summarization: One aspect of transformation is to summarize data and then store it.
5. Enrichment: Data from different sources once made into a consistent format are then enriched by adding the missing data.

The following points must be rectified in this phase:

- Loose texts may hide valuable information. For example, XYZ PVT Ltd does not explicitly show that this is a Limited Partnership company.
- Different formats can be used for individual data. For example, data can be saved as a string or as three integers.

The following are the main transformation processes aimed at populating the reconciled data layer:

- Conversion and normalization that operate on both storage formats and units of measure to make data uniform.
- Matching that associates equivalent fields in different sources.
- Selection that reduces the number of source fields and records.

Cleansing and Transformation processes are often closely linked in ETL tools.

## 3. Loading

The Load is the process of writing the data into the target database. During the load step, it is necessary to ensure that the load is performed correctly and with as few resources as possible.

Loading can be carried out in two ways:

1. **Refresh:** Data Warehouse data is completely rewritten. This means that older file is replaced. Refresh is usually used in combination with static extraction to populate a data warehouse initially.
2. **Update:** Only those changes applied to source information are added to the Data Warehouse. An update is typically carried out without deleting or modifying preexisting data. This method is used in combination with incremental extraction to update data warehouses regularly.

## EXTRACT

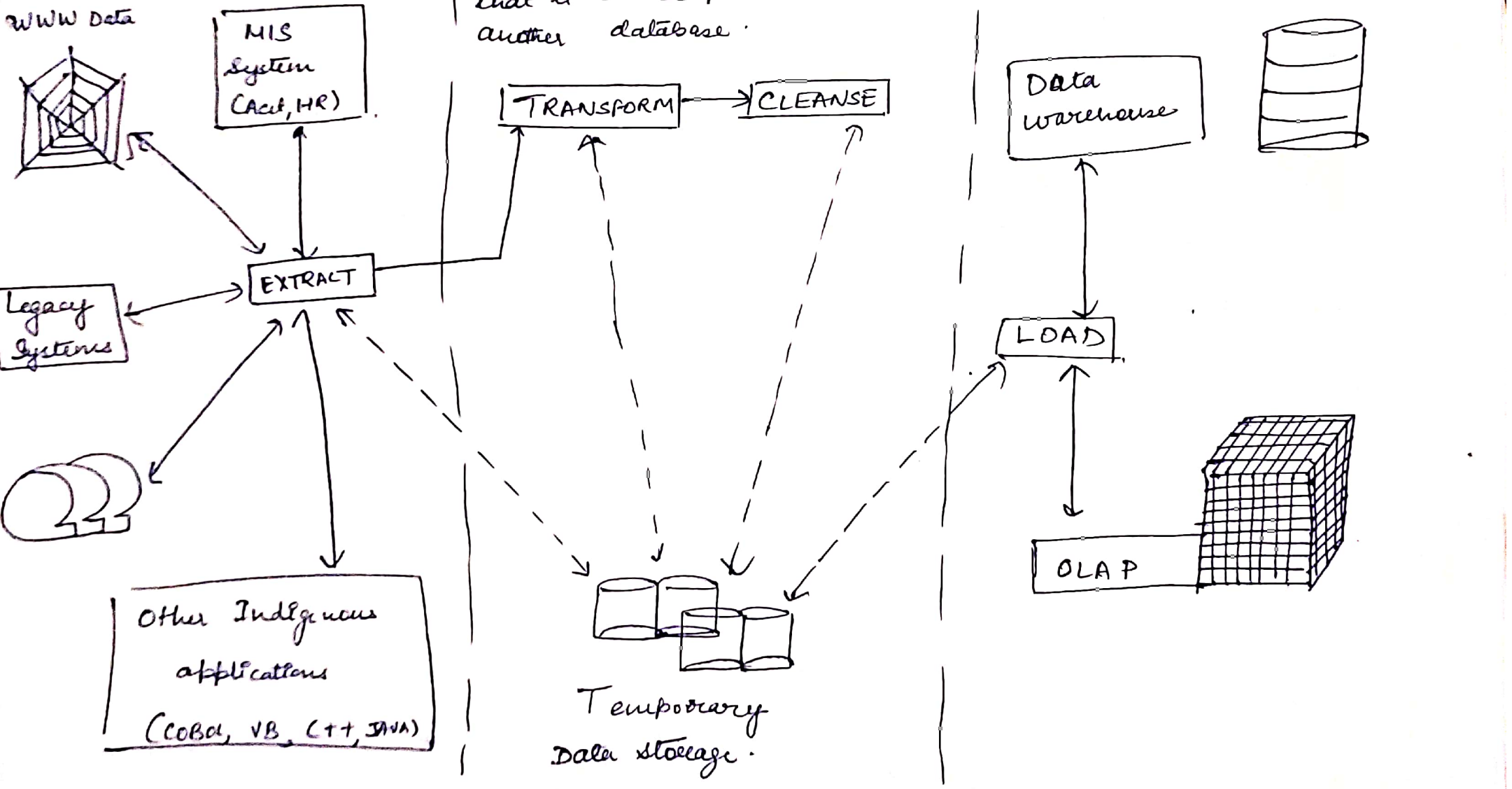
The process of reading data from different sources.

## TRANSFORM

The process of transforming the extracted data from its original state into a consistent state so that it can be placed into another database.

## LOAD

The process sorts, summarizes, consolidates, computes views, check integrity and builds indices and partitions.



## DATA QUALITY

### What is Data Quality?

Data quality is defined as:

*the degree to which data meets a company's expectations of accuracy, validity, completeness, and consistency.*

By tracking data quality, a business can pinpoint potential issues harming quality, and ensure that shared data is fit to be used for a given purpose.

When collected data fails to meet the company's expectations of accuracy, validity, completeness, and consistency, it can have massive negative impacts on customer service, employee productivity, and key strategies.

### Why Is Data Quality Important?

Quality data is key to making accurate, informed decisions. While all data has some level of "quality," a variety of characteristics and factors determines the degree of data quality (high-quality versus low-quality). Furthermore, different data quality characteristics will likely be more important to various stakeholders across the organization.

A list of popular data quality characteristics and dimensions include:

- Accuracy
- Completeness
- Consistency
- Integrity
- Reasonability
- Timeliness
- Uniqueness/Deduplication
- Validity
- Accessibility

Because data accuracy is a key attribute of high-quality data, a single inaccurate data point can wreak havoc across the entire system.

Without accuracy and reliability in data quality, executives cannot trust the data or make informed decisions. This can, in turn, increase operational costs and wreak havoc for downstream users. Analysts wind up relying on imperfect reports and making misguided conclusions based on those findings. And the productivity of end-users will diminish due to flawed guidelines and practices being in place.

Poorly maintained data can lead to a variety of other problems, too. For example, out-of-date customer information may result in missed opportunities for up- or cross-selling products and services.

Low-quality data might also cause a company to ship their products to the wrong addresses, resulting in lowered customer satisfaction ratings, decreases in repeat sales, and higher costs due to reshipments.

And in more highly regulated industries, bad data can result in the company receiving fines for improper financial or regulatory compliance reporting.

## Characteristics of Data Quality

① **Accuracy** → The ~~value~~ data must conform to actual, real-world scenarios and reflect real-world objects and events. Analysts should use verifiable sources to confirm the measure of accuracy, determined by how close the values jibe with the verified correct information sources.

② **Completeness** → Completeness measures the data's ability to deliver all the mandatory values that are available successfully.

③ **Consistency** → Data consistency describe the data's uniformity as it moves across applications and networks and when it comes from multiple sources. Consistency also means that the same datasets stored in different locations should be the same and not conflict. Note that consistent data can still be wrong.

④ **Timeliness** → Timely data is information that is readily available whenever it's needed. This dimension also covers keeping the data current; data should undergo real-time updates to ensure that it is always available and accessible.

⑤ **Uniqueness** → Uniqueness means that no duplications or redundant are overlapping across all the dataset. No records in the datasets exists multiple times. Analysts use data cleasing and deduplication to help address a low uniqueness score.

⑥ Validity → Data must be collected according to the organization's defined business rules and parameters. The information should also conform to the correct, accepted formats, and all dataset values should fall within the proper range.

⑦ Redundancy → The same data must not be stored in more than one place in a system.

⑧ Accessibility → The extent to which data is available or easily and quickly retrievable

⑨ Objectivity → The extent to which the data is unbiased.

⑩ Clarity → clarity is obtained by proper naming conventions. It helps to make the data element well understood by users.

## DATA QUALITY CHALLENGES

1. **Managing the data structure and optimization**
   The correct way to process data is to structure it in a way that will aid your future operations. As you add more and more data to your warehouse, structuring becomes increasingly difficult and can slow down the ETL process. Also, it becomes increasingly difficult for system managers to qualify the data for advanced analytics.
   In terms of system optimization, it's important to carefully design and configure data analysis tools that are better suited to business needs.
2. **Managing user expectations**
   As more information gets loaded into a data warehouse, management systems struggle more to find and analyze it. This means that business users expect refined and relevant results from any analysis they run. However, data warehouse performance can decrease as the data volume increases, which inevitably leads to reduced speed and efficiency. It's your job to manage the expectations of your team so that they aren't frustrated when the buffering occurs.
3. **The costs of data warehousing**
   A common problem with traditional data warehouses is the high failure rate. According to a Gartner report, more than 50% of data warehouses fail at one point — not only because of the technical challenges and complex architecture but also because the projects fail to meet user requirements.
   Organizations then face the same challenges when trying to update a data warehouse to accommodate new reporting requirements or data models.
   Even if such projects don't fail, they have high costs and timelines. All these factors make traditional data warehouses inadequate for real-time data requirements and scalability.
   On the other hand, if you go with a cloud-based data warehouse, all the maintenance rests on the cloud provider, while the cost is formed by the used GBs per month.
   Snowflake, for example, even has a flat rate of $23/TB/month.
   Google BigQuery's active storage costs $0.02 per GB per month, with the first 10 GB free each month.
4. **Data quality**
   Maintaining quality data is difficult in a traditional data warehouse where manual errors and missed updates lead to corrupt or obsolete data. This inevitably impacts business decisions and causes inaccurate data processing.
   As businesses increasingly adopt digital transformation initiatives, they often run into the problem of unintended data silos.
   This occurs when departments heavily rely on cloud tools accompanied by the democratization of technology — where each department is more likely to be responsible for purchasing and developing technologies for its use.
   Each of these silos represents another source system from which users need to pull, integrate, and analyze data to use it correctly in decision-making. To make matters worse, silos often don't follow the same set of businesswide standards, making data integration even more difficult.
   And due to the democratization of cloud technologies, your organization might even have valuable data silos that IT doesn't know about.
   Modern warehousing solutions can automate the data quality process, preventing data silos, outliers, manual errors, redundancy, and other data inconsistencies from occurring.
   With an automated data warehousing solution, you are able to provide high-quality data that brings the most value to your organization.
5. **Data Accuracy**
   If you want your data insights and business intelligence to be reliable, the data that is analyzed in a warehouse needs to be accurate. Traditional data warehouses often suffer from inconsistencies that lead to inaccurate data as a result of manual processing and other errors.
   There are several ways to go around this challenge, but the first and most important is to ensure that a data collection and storing process is accurate and that the new data is transformed correctly before it enters the warehouse.
   Data accuracy can also be improved through regular testing.

However, with the right data warehousing solution that supports automated transfers, the chance for human error is minimal. If you use an ETL tool, not only can you prevent inaccurate data from entering your data warehouse, but also flag errors so that you can optimize your data accuracy at the source.

6. **Adjusting to non-technical users**

Traditional data warehouses are often complex for non-technical teams to use. Sure, everyone can master data analysis enough to be able to query data from any source and know how to use the data provided this way. But the reality is different.

Non-technical users often need to interact with company data, which is not very efficient if you use a traditional data warehouse — submitting a request to the data team, waiting for the data team to fulfill the request, and using the data once delivered to them.

The process might work in small teams, but for larger teams, it's time-consuming and inefficient, as data teams can quickly become saturated with requests, leading to frustration and bottlenecks.

However, with modern, self-managed data warehouses and automated ETL tools, this challenge is easy to overcome.

Data transfer tools like What graph allow any user to move data from disparate sources to Google BigQuery without enlisting any help from the data or developer team. With point-and-click solutions, even non-technical users can operate a data warehouse without slowing down the workflow.

7. **Data pollution**

Sometimes the data gets corrupted in the source systems. Some of the common sources of data pollution is:

- System Conversions
- Data Aging
- Heterogeneous System Integration
- Poor Database Design
- Incomplete information at data entry
- Input errors
- Internationalization and Localization of data
- Fraud
- Lack of policy