1

INTRODUCTION TO DATA WAREHOUSING

CHAPTER OUTLINE

- Lifecycle of data, Types of data,
- Data warehouse and data warehousing,
- Differences between operational database and data warehouse,
- A multidimensional data model,
- OLAP operation in multidimensional data model,
- Conceptual modeling of data warehouse,
- Architecture of data ware house,
- Data warehouse implementation,
- Data marts
- Components of data warehouse
- Need for data ware housing
- Trends in data warehousing

LIFECYCLE OF DATA

The data life cycle provides a high-level overview of the stages involved in successful management and preservation of data for use and reuse. Multiple versions of a data life cycle exist with differences attributable to variation in practices across domains or communities. The data life cycle is often described as a cycle because the lessons learned and insights gleaned from one data project typically inform the next. In this way, the final step of the process feeds back into the first.

No two data projects are identical; each brings its own challenges, opportunities, and potential solutions that impact its trajectory. Nearly all data projects, however, follow the same basic life cycle from start to finish. This life cycle can be split into eight common stages or steps, or phases: *Generation, Collection, Processing, Storage, Management, Analysis, Visualization, and Interpretation*

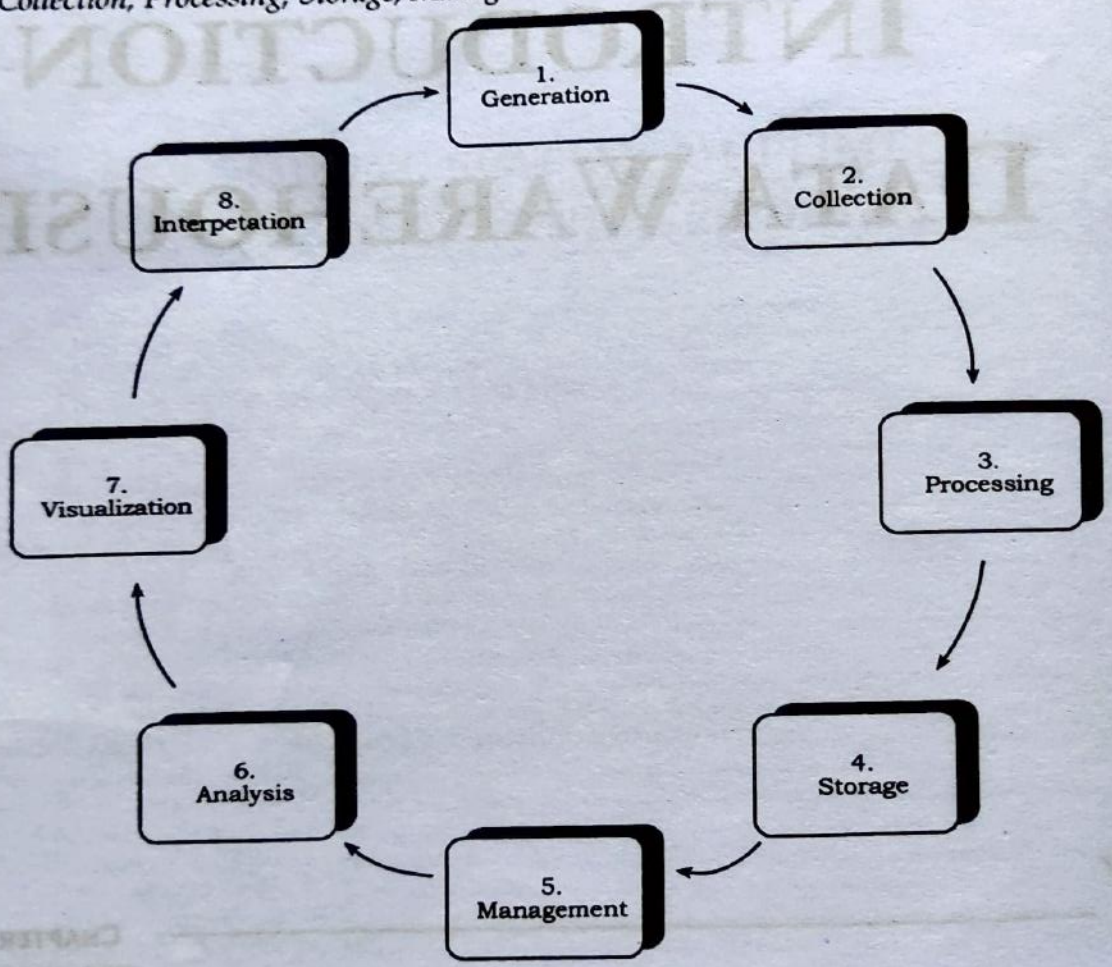

Figure 1.1: 8 steps in data life cycle.

1. Generation

For the data life cycle to begin, data must first be generated. Otherwise, the following steps can't be initiated. Data generation occurs regardless of whether you're aware of it, especially in our increasingly online world. Some of this data is generated by your organization, some by your customers, and some by third parties you may or may not be aware of. Every sale, purchase, hire, communication, interaction—everything generates data. Given the proper attention, this data can often lead to powerful insights that allow you to better serve your customers and become more effective in your role.

2. **Collection**

Not all of the data that's generated every day is collected or used. It's up to your data team to identify what information should be captured and the best means for doing so, and what data is unnecessary or irrelevant to the project at hand. We can collect data in a variety of ways, including:

- **Forms:** Web forms, client or customer intake forms, vendor forms, and human resources applications are some of the most common ways businesses generate data.
- **Surveys:** Surveys can be an effective way to gather vast amounts of information from a large number of respondents.
- **Interviews:** Interviews and focus groups conducted with customers, users, or job applicants offer opportunities to gather qualitative and subjective data that may be difficult to capture through other means.
- **Direct Observation:** Observing how a customer interacts with your website, application, or product can be an effective way to gather data that may not be offered through the methods above.

It's important to note that many organizations take a broad approach to data collection, capturing as much data as possible from each interaction and storing it for potential use.

3. **Processing**

Once data has been collected, it must be processed. Data processing can refer to various activities, including:

- **Data wrangling,** in which a data set is cleaned and transformed from its raw form into something more accessible and usable. This is also known as data cleaning or data remediation.
- **Data compression,** in which data is transformed into a format that can be more efficiently stored.
- **Data encryption,** in which data is translated into another form of code to protect it from privacy concerns.

Even the simple act of taking a printed form and digitizing it can be considered a form of data processing.

4. **Storage**

After data has been collected and processed, it must be stored for future use. This is most commonly achieved through the creation of databases or datasets. These datasets may then be stored in the cloud, on servers, or using another form of physical storage like a hard drive, CD, cassette, or floppy disk.

When determining how to best store data for your organization, it's important to build in a certain level of redundancy to ensure that a copy of your data will be protected and accessible, even if the original source becomes corrupted or compromised.

5. **Management**

Data management, also called database management, involves organizing, storing, and retrieving data as necessary over the life of a data project. While referred to here as a step, it's an ongoing process that takes place from the beginning through the end of a project. Data management includes everything from storage and encryption to implementing access logs and change logs that track who have accessed data and what changes they may have made.

6. **Analysis**
Data analysis refers to processes that attempt to glean meaningful insights from raw data. Analysts and data scientists use different tools and strategies to conduct these analyses. Some of the more commonly used methods include statistical modeling, algorithms, artificial intelligence, data mining, and machine learning. Exactly who performs an analysis depends on the specific challenge being addressed, as well as the size of your organization's data team. Business analysts, data analysts, and data scientists can all play a role.

7. **Visualization**
Data visualization refers to the process of creating graphical representations of your information, typically through the use of one or more visualization tools. Visualizing data makes it easier to quickly communicate your analysis to a wider audience both inside and outside your organization. The form your visualization takes depends on the data you're working with, as well as the story you want to communicate. While technically not a required step for all data projects, data visualization has become an increasingly important part of the data life cycle.

8. **Interpretation**
Finally, the interpretation phase of the data life cycle provides the opportunity to make sense of your analysis and visualization. Beyond simply presenting the data, this is when you investigate it through the lens of your expertise and understanding. Your interpretation may not only include a description or explanation of what the data shows but, more importantly, what the implications may be.

TYPES OF DATA

Data is a collection of facts, such as numbers, words, measurements, observations or just descriptions of things. We define data into three categories namely, **Record Data, Graph-based Data,** and **Ordered Data.**

1. **Record Data**
   a) Data Matrix
   b) Document Data
   c)Transaction Data
2. **Graph based data**
   a) Linked web pages
   b) Benzene Molecular Structures
3. **Ordered data**
   a) Sequential Data
   b) Genetic Sequence Data
   c) Temporal Data
   d) Spatial Data

Record Data

Data that consists of a collection of records, each of which consists of a fixed set of attributes is called record data. The most basic form of record data has no explicit relationship among records or data fields, and every record (object) has the same set of attributes. Record data is usually stored either in flat files or in relational databases.

Example:

Tid	Refund	Marital status	Taxable income	Cheat
1	Yes	Single	200000	No
2	No	Married	400000	Yes
3	No	Single	340000	No
4	Yes	Single	150000	No
5	No	Married	340000	Yes
6	No	Single	550000	No
7	Yes	Divorced	540000	Yes

There are a few variations of Record Data, which have some characteristic properties.

Transaction or Market Basket Data

It is a special type of record data, in which each record contains a set of items. For example, shopping in a supermarket or a grocery store. For any particular customer, a record will contain a set of items purchased by the customer in that respective visit to the supermarket or the grocery store. This type of data is called Market Basket Data. Transaction data is a collection of sets of items, but it can be viewed as a set of records whose fields are asymmetric attributes. Most often, the attributes are binary, indicating whether or not an item was purchased or not.

Tid	Item
1	Pencil, Paper
2	Pencil, Book, Rubber, Ink
3	Paper, Book, Rubber, Ruler
4	Pencil, Paper, Book, Rubber
5	Pencil, Paper, Book, Ruler

The Data Matrix

If the data objects in a collection of data all have the same fixed set of numeric attributes, then the data objects can be thought of as points (vectors) in a multidimensional space, where each dimension represents a distinct attribute describing the object. A set of such data objects can be interpreted as an m × n matrix, where there are n rows, one for each object, and n columns, one for each attribute. Standard matrix operation can be applied to transform and manipulate the data. Therefore, the data matrix is the standard data format for most statistical data. Some examples are shown below:

Point	Attribute1	Attribute2
x_1	1	2
x_2	3	5
x_3	2	0

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

The Sparse Data Matrix /Document-data Matrix

A sparse data matrix (sometimes also called document-data matrix) is a special case of a data matrix in which the attributes are of the same type and are asymmetric; i.e., only non-zero values are important.

	Team	Coach	Play	Ball	Score	Game	Win	Lost	Timeout	Season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	3	2	0	4	0

Graph Based Data

In computing, a graph database (GDB) is a database that uses graph structures for semantic queries with nodes, edges, and properties to represent and store data. A key concept of the system is the graph (or edge or relationship). The graph relates the data items in the store to a collection of nodes and edges, the edges representing the relationships between the nodes. The relationships allow data in the store to be linked together directly and, in many cases, retrieved with one operation. Graph databases hold the relationships between data as a priority. This can be further divided into types:

Data with Relationships Among Objects (Linked Web Pages Data)

Linked data is an approach to publishing and sharing data on the web. The Semantic Web isn't just about putting data on the web. It is about making links, so that a person or machine can explore the web of data. With linked data, when we have some of it, we can find other, related, data. Like the web of hypertext, the web of data is constructed with documents on the web. However, unlike the web of hypertext, where links are relationships anchors in hypertext documents written in HTML, for data they link between arbitrary things described by Resource Description Framework (RDF).

The data objects are mapped to nodes of the graph, while the relationships among objects are captured by the links between objects and link properties, such as direction and weight. Consider Web pages on the World Wide Web, which contain both text and links to other pages. In order to process search queries, Web search engines collect and process Web pages to extract their contents.

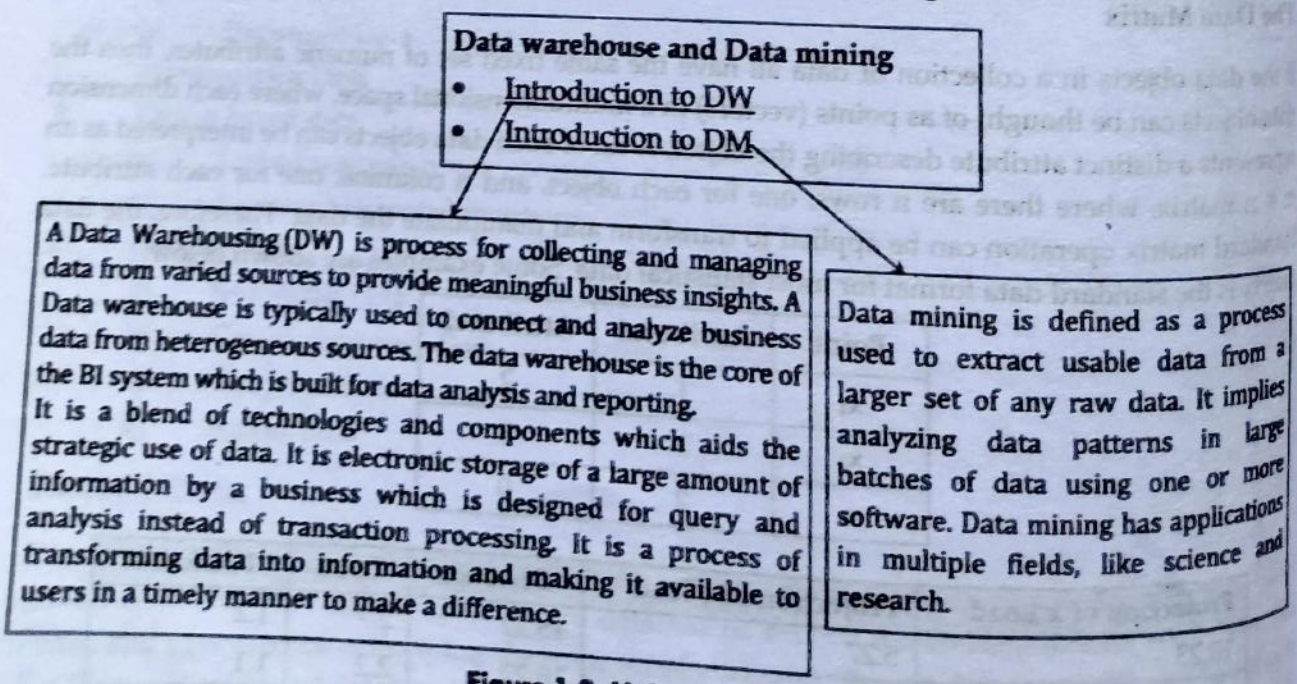

Figure 1.2: Linked web pages

Data with Objects That Are Graphs (Benzene(C_6H_6)Molecular Structures)

If objects have structure, that is, the objects contain sub objects that have relationships, then such objects are frequently represented as graphs. For example, the structure of chemical compounds can be represented by a graph, where the nodes are atoms and the links between nodes are chemical bonds.

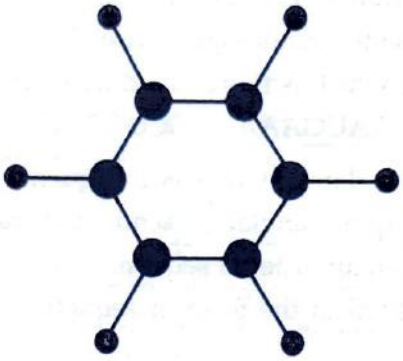

Figure 1.3: Graphical representation of Benzine molecular structure

Ordered Data

Ordered data set records are kept in a physical sequence based on a user-specified key without the necessity of utilizing a set. Ordered data sets can be either disjoint or embedded, but are normally embedded. For some types of data, the attributes have relationships that involve order in time or space. It can be segregated into four types:

Sequential Data

Whenever the points in the dataset are dependent on the other points in the dataset the data is said to be Sequential data. A common example of this is a Time series such as a stock price or a sensor data where each point represents an observation at a certain point in time.

Sequential Data is any kind of data where the order matters as you said. So, we can assume that time series is a kind of sequential data, because the order matters. A time series is a sequence taken at successive equally spaced points in time and it is not the only case of sequential data. Consider a retail transaction data set that also stores the time at which the transaction took place

Time	Customer	Item Purchased
T1	Aarav	Bag, book
T2	Umesh	Bag, pen
T2	Aarav	Pen, Copy
T3	Aadesh	Bag, Copy
T4	Aadesh	Doll
T5	Aarav	Bag, Doll

customer	Time and Item Purchased
Aarav	(T1: Bag, book) (T2: Pen, Copy)(T5: Bag, Doll)
Umesh	(T2: Bag, pen)
Aadesh	(T3: Bag, Copy) (T4: Doll)

Genetic Sequence Data

Organisms are built, and their functions are determined, by their genetic code. This code is contained in DNA molecules, which are found in human, animal and plant cells, as well as in microorganisms like bacteria and viruses. DNA has four components, or building blocks, called C (cytosine), G (guanine), A (adenine), or T (thymine). Laboratories can determine the genetic sequence of a particular organism, using sequencing technologies. The data generated through this process is called genetic sequence data (GSD), which is represented by listing the nucleotides in order (e.g., an RNA sequence might look like AGAAAUGAAAUGGCUCCUGUCAA).

Genetic Sequence data consists of a data set that is a sequence of individual entities, such as a sequence of words or letters. It is quite similar to sequential data, except that there are no time stamps; instead, there are positions in an ordered sequence. For example, the genetic information of plants and animals can be represented in the form of sequences of nucleotides that are known as genes.

GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG

Figure 1.4: Genomic sequence data

Time Series Data (Temporal Data)

Time series data, also referred to as time-stamped data, is a sequence of data points indexed in time order. Time-stamped is data collected at different points in time. These data points typically consist of successive measurements made from the same source over a time interval and are used to track change over time. Time series data is a special type of sequential data in which each record is a time series, i.e., a series of measurements taken over time. For example, a financial data set might contain objects that are time series of the daily prices of various stocks.

Fiscal Year	NEPSE Index (Mid-July)
2053/54	176.3
2054/55	163.3
2055/56	216.9
2056/57	360.7
2057/58	348.4
2058/59	227.5
2059/60	204.86
2060/61	22.04
2061/62	286.67
2062/63	386.86
2063/64	683.95
2064/65	963.36
2065/66	749.1
2066/67	477.73
2067/68	362.85
2068.69	389.74
2069/70	518.33
2070/71	1036.1
2071/72	961.2
2072/73	1718.2
1073/74	1582.67
2074/75	1200.09
2075/76*	1102.64

*Nepse Index is of Falgun 21, 2072

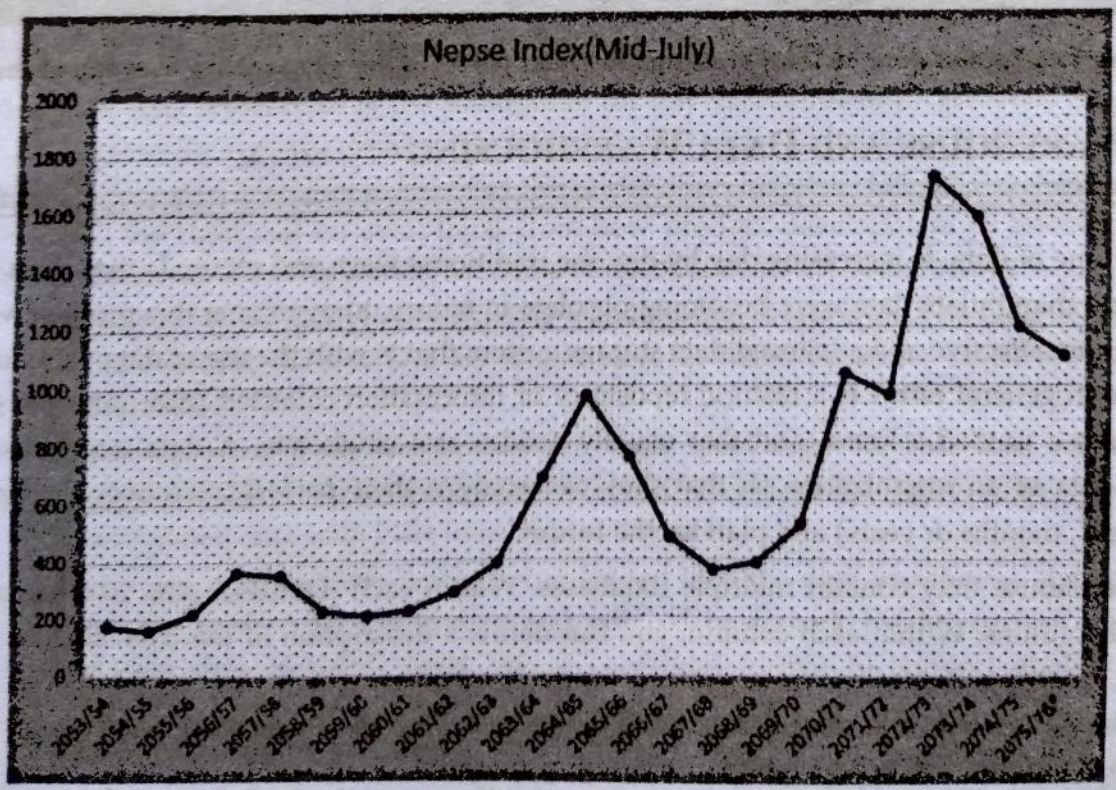

Figure 1.5: NEPSE index (Mid July) over 23 years

Spatial Data

Spatial data, also known as geospatial data, is information about a physical object that can be represented by numerical values in a geographic coordinate system. Generally speaking, spatial data represents the location, size and shape of an object on planet Earth such as a building, lake, mountain or township.

Some objects have spatial attributes, such as positions or areas, as well as other types of attributes. An example of spatial data is weather data (precipitation, temperature, pressure) that is collected for a variety of geographical locations.

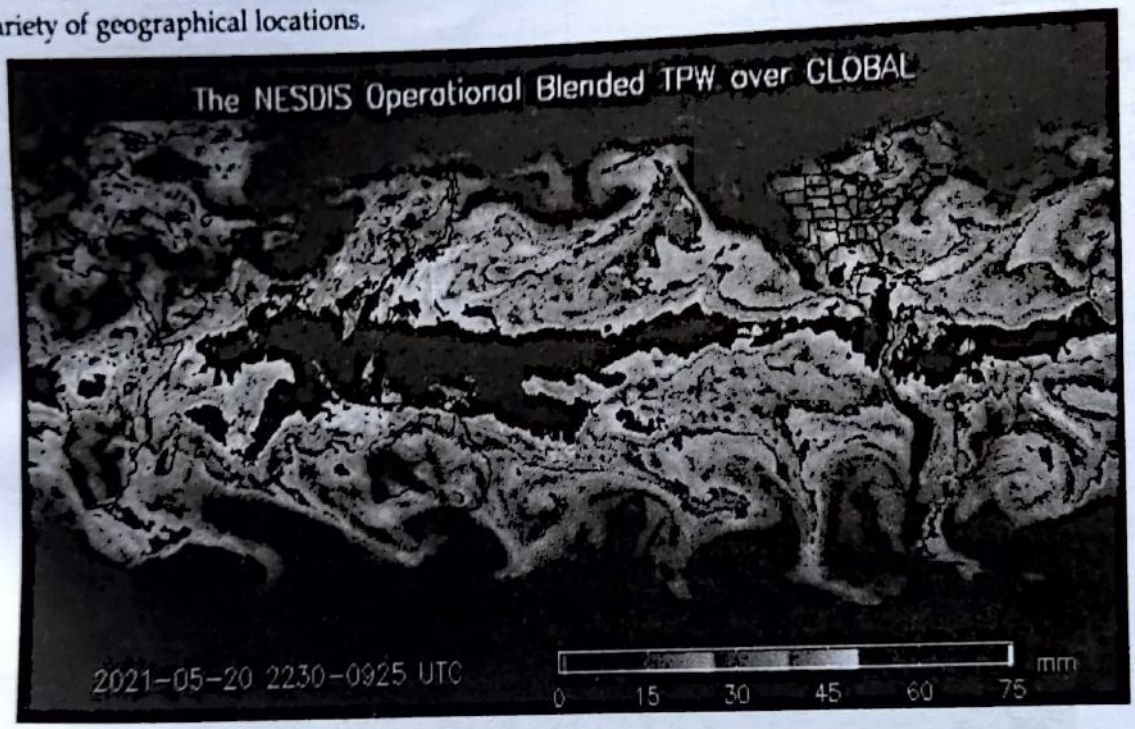

Figure 1.6: Spatial data of Total Precipitable Water (TPW) in the atmosphere over the globe.

DATA WAREHOUSE AND DATA WAREHOUSING

As the volume of data, is increasing day by day the traditional ways and methods that were used to manage and manipulate data were becoming obsolete in nature, to overcome this problem we need to have a more effective and advanced data storage system that is with the use of data warehouses. A warehouse in general terms is a historic repository of information collected from multiple sources, stored under a unified schema, and that usually resides at a single site. A data warehouse stores historical data of an organization so that they can analyze their performance over the past time (days, weeks, months or years) and plan for the future.

A data warehouse may contain multiple databases. Within each database, data is organized into tables and columns. Within each column, you can define a description of the data, such as integer, data field, or string. Tables can be organized inside of schemas, which you can think of as folders. When data is ingested, it is stored in various tables described by the schema. Query tools use the schema to determine which data tables to access and analyze.

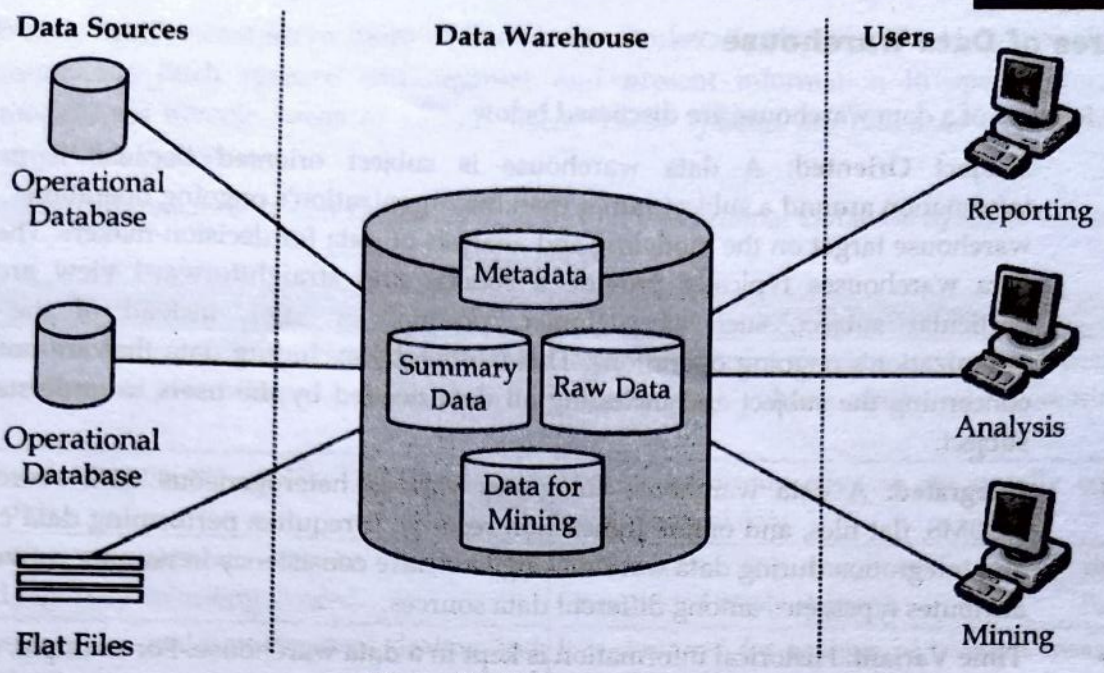

Figure 1.7: Data warehouse

A data warehouse (DW) is a digital storage system that connects and harmonizes large amounts of data from many different sources. Its purpose is to feed business intelligence (BI), reporting, and analytics, and support regulatory requirements – so companies can turn their data into insight and make smart, data-driven decisions. Data warehouses store current and historical data in one place and act as the single source of truth for an organization.

Data warehousing is the process of constructing and using data warehouses. It is the process of extracting & transferring operational data into informational data & loading it into a central data store (warehouse).

Benefits of Data Warehousing

A well-designed data warehouse is the foundation for any successful BI or analytics program. Its main job is to power the reports, dashboards, and analytical tools that have become indispensable to businesses today. A data warehouse provides the information for your data-driven decisions – and helps you make the right call on everything from new product development to inventory levels. There are many benefits of a data warehouse some of major benefits are listed below:

- **Better business analytics**: With data warehousing, decision-makers have access to data from multiple sources and no longer have to make decisions based on incomplete information.
- **Faster queries:** Data warehouses are built specifically for fast data retrieval and analysis. With a data warehouse, we can very rapidly query large amounts of consolidated data with little to no support from IT.
- **Improved data quality:** Before being loaded into the data warehouse, data cleansing cases are created by the system and entered in a work list for further processing, ensuring data is transformed into a consistent format to support analytics – and decisions – based on high quality, accurate data.
- **Historical insight:** By storing rich historical data, a data warehouse lets decision-makers learn from past trends and challenges, make predictions, and drive continuous business improvement.

Features of Data Warehouse

The key features of a data warehouse are discussed below:

- **Subject Oriented:** A data warehouse is subject oriented because it provides information around a subject rather than the organization's ongoing operations. A data warehouse target on the modeling and analysis of data for decision-makers. Therefore, data warehouses typically provide a concise and straightforward view around a particular subject, such as customer, product, or sales, instead of the global organization's ongoing operations. This is done by excluding data that are not useful concerning the subject and including all data needed by the users to understand the subject.
- **Integrated:** A data warehouse integrates various heterogeneous data sources like RDBMS, flat files, and online transaction records. It requires performing data cleaning and integration during data warehousing to ensure consistency in naming conventions, attributes types, etc., among different data sources.
- **Time Variant:** Historical information is kept in a data warehouse. For example, one can retrieve files from 3 months, 6 months, 12 months, or even previous data from a data warehouse. These variations with a transactions system, where often only the most current file is kept.
- **Non-volatile:** Non-volatile means the previous data is not erased when new data is added to it. A data warehouse is kept separate from the operational database and therefore frequent changes in operational database are not reflected in the data warehouse.

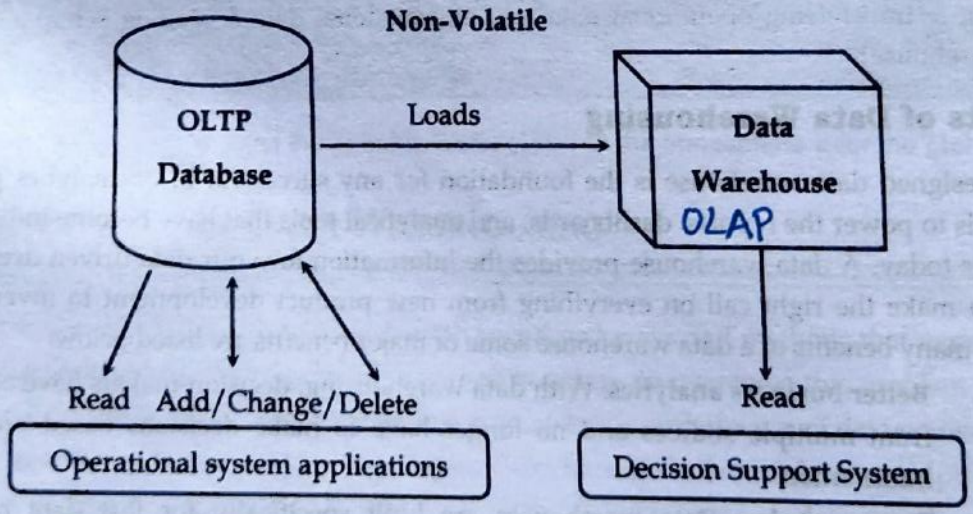

Figure 1.8: OLTP versus data warehouse and both are non-volatile

Differences Between Operational Database and Data Warehouse

The Operational Database is the source of information for the data warehouse. It includes detailed information used to run the day-to-day operations of the business. The data frequently changes as updates are made and reflect the current value of the last transactions. Operational Database Management Systems also called as OLTP (Online Transactions Processing Databases), are used to manage dynamic data in real-time.

Data Warehouse Systems serve users or knowledge workers in the purpose of data analysis and decision-making. Such systems can organize and present information in specific formats to accommodate the diverse needs of various users. These systems are called as Online-Analytical Processing (OLAP) Systems.

Some major differences between Data Warehouses and Operational Database Systems are tabulated below:

Operational Database	Data Warehouse
Operational systems are designed to support high-volume transaction processing.	Data warehousing systems are typically designed to support high-volume analytical processing (i.e., OLAP).
Operational systems are usually concerned with current data.	Data warehousing systems are usually concerned with historical data.
Data within operational systems are mainly updated regularly according to need.	Non-volatile, new data may be added regularly. Once Added rarely changed.
It is designed for real-time business dealing and processes.	It is designed for analysis of business measures by subject area, categories, and attributes.
It is optimized for a simple set of transactions, generally adding or retrieving a single row at a time per table.	It is optimized for extent loads and high, complex, unpredictable queries that access many rows per table.
It is optimized for validation of incoming information during transactions, uses validation data tables.	Loaded with consistent, valid information, requires no real-time validation.
It supports thousands of concurrent clients.	It supports a few concurrent clients relative to OLTP.
Operational systems are widely process-oriented.	Data warehousing systems are widely subject-oriented
Operational systems are usually optimized to perform fast inserts and updates of associatively small volumes of data.	Data warehousing systems are usually optimized to perform fast retrievals of relatively high volumes of data.
Less Number of data accessed.	Large Number of data accessed.
Relational databases are created for on-line transactional Processing (OLTP)	Data Warehouse designed for on-line Analytical Processing (OLAP)
It has normalized schema	Data warehouse has de-normalized schema
E-R Model is used for designing	Star or Snow flake Model is used for designing

A MULTIDIMENSIONAL DATA MODEL

Multidimensional data model in data warehouse is a model which represents data in the form of data cubes. It allows to model and view the data in multiple dimensions and it is defined by dimensions and facts. Multidimensional data model is generally categorized around a central theme and represented by a fact table. It is typically used in the organizations for drawing out Analytical results and generation of reports, which can be used as the main source for imperative decision-making processes. This model is typically applied to systems that operate with OLAP techniques (Online Analytical Processing).

The Multi-Dimensional Data Model is a significant improvement amongst various areas of Data Science, like the Data Warehouse system and the Data Management techniques. Multi-Dimensional Models are found to be the competent relational systems, which can serve as a key input for generating Analytical outcomes for the purpose of business decision making processes.

Now, if we want to view the sales data with a third dimension, for example, suppose the data according to time, product and location. Time is considered for four quarters i.e., *Q1, Q2, Q3,* and *Q4,* whereas four products are considered i.e., *Television (TV), Personal Computer (PC), Access Point (AP),* and *Solid-State Drive (SSD),* and the location is considered for the cities *Pokhara, Kawasoti, Dhangadi,* and *Mahendranagar.* These 3D data are shown in the table below. The 3D data of the table are represented as a series of 2D tables.

Table 1.1:3D view of sales data according to time, product and location

	Location = "Pokhara"				Location = "Kawasoti"				Location = "Dhangadi"				Location="Mahendranagar"			
	Product				Product				Product				Product			
Time	TV	PC	AP	SSD	TV	PC	AP	SSD	TV	PC	AP	SSD	TV	PC	AP	SSD
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

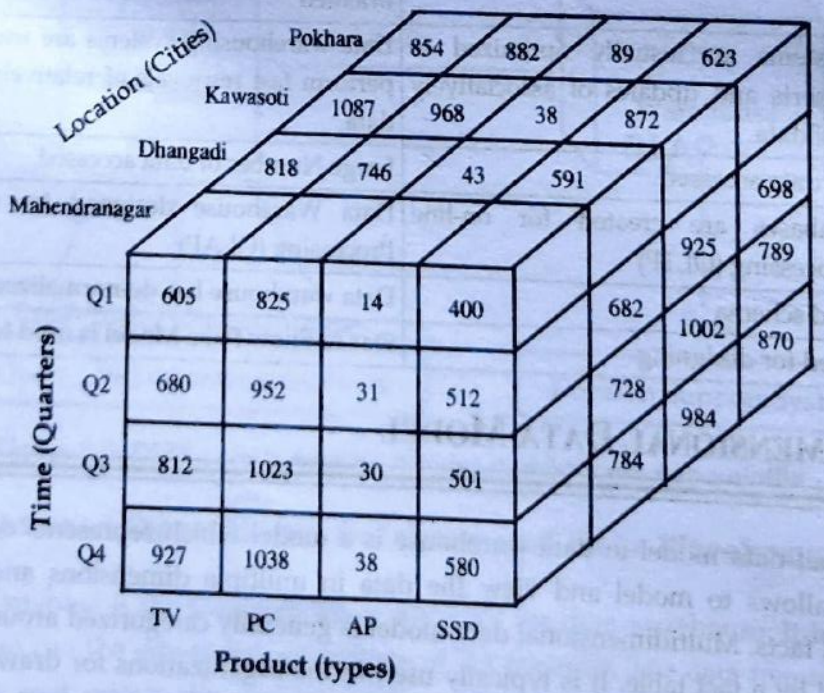

Figure 1.9: Multidimensional Data Model (3D data cube of sales data)

Working Mechanism of Multidimensional Data Model

Like any other system, the Multidimensional Data Model also works based on the predetermined steps, in order to keep the pattern, the same throughout the industry and for enabling the reusability of the already designed or created database systems. For creating a Multidimensional Data Model, every project should go all the way through the below phases,

- **Congregating the requirements from the client**
  Similar to the other software applications, a Data Model also requires the precise requirement from the client. Most of the time, the client might not know what could be accomplished with the selected technology. It is the software professional's duty to provide clarity on to what extent a requirement can be achieved with the selected technology, and elaborately collect the complete requirement.
- **Categorizing the various modules of the system**
  After the process of collecting the entire requirement, the next step is to identify and categorize each of the requirements under the module where they belong. Modularity helps in better management, and also makes it trouble-free to implement, one at a time.
- **Spotting the various dimensions based on which the system needs to be designed**
  Once the separation of various requirements and moving them to the matching modules are completed, the next step is to identify the main factors, from the user's point of view. These factors can be termed as the dimensions, based on which the multidimensional data model can be created.
- **Drafting the real-time dimensions and the corresponding properties**
  As a part of next step, in the process of the Multi-Dimensional Data Model, the dimensions identified in the previous step can be further used for recognizing the related properties. These properties are termed as the 'attributes' in the database systems.
- **Discovering the facts from the already listed dimensions and their properties**
  From the initial requirement gathering, the dimensions can be a mix of dimensions and facts. It is a significant step to distinguish and segregate the facts from the dimensions. These facts play a great role in the structure of the Multi-Dimensional Data Models.
- **Constructing the Schema to place the data, with respect to the information gathered from the above steps:**
  Based on the information collected so far, the elaborate requirements, the dimensions, the facts, and their respective attributes, a Schema can be constructed. There are many types of Schemas, from which the most suitable type of schema can be chosen. A few of the commonly used schema types are the Star Schema, the Galaxy Schema, and the Snowflake Schema.

Advantages and Disadvantages of Multidimensional Data Model

Below are the advantages and disadvantages:

Advantages

- Multi-Dimensional Data Models are workable on complex systems and applications, unlike the simple one-dimensional database systems.
- The Modularity in this type of Database is an encouragement for projects with lower bandwidth for maintenance staff.

- Overall, organizational capacity and structural definition of the Multi-Dimensional Data Models aids in holding cleaner and reliable data in the database.
- Clearly defined construction of the data placements makes it uncomplicated, in situations like one team constructs the database, another team works on it and some other team works on the maintenance. It serves as a self -learning system if and when required.
- As the system is fresh and free of junk, the efficiency of the data and performance of the database system is found to be advanced & elevated.

Disadvantages

- As the Multi-Dimensional Data Model handles complex systems, these types of databases are typically complex in nature.
- Being a complex system means the contents of the database are huge in the amount as well. This makes the system to be highly risky when there is a security breach.
- When the system caches due to the operations on the Multi-Dimensional Data Model, the performance of the system is affected greatly.
- Though the end product in a Multi-Dimensional Data Model is advantageous, the path to achieving it is intricate most of the time.

OLAP Operation in Multidimensional Data Model

Online Analytical Processing Server (OLAP) is based on the multidimensional data model. It allows managers, and analysts to get an insight of the information through fast, consistent, and interactive access to information. Since OLAP servers are based on multidimensional view of data, we will discuss OLAP operations in multidimensional data. Here is the list of OLAP operations:

- Roll-up
- Drill-down
- Slice and dice
- Pivot (rotate)

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.

Roll-Up

The roll-up operation (also known as drill-up or aggregation operation) performs aggregation on a data cube, by climbing up 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 province. 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. The following diagram illustrates how roll-up works when the sales data cube is rolled-up from cities to province:

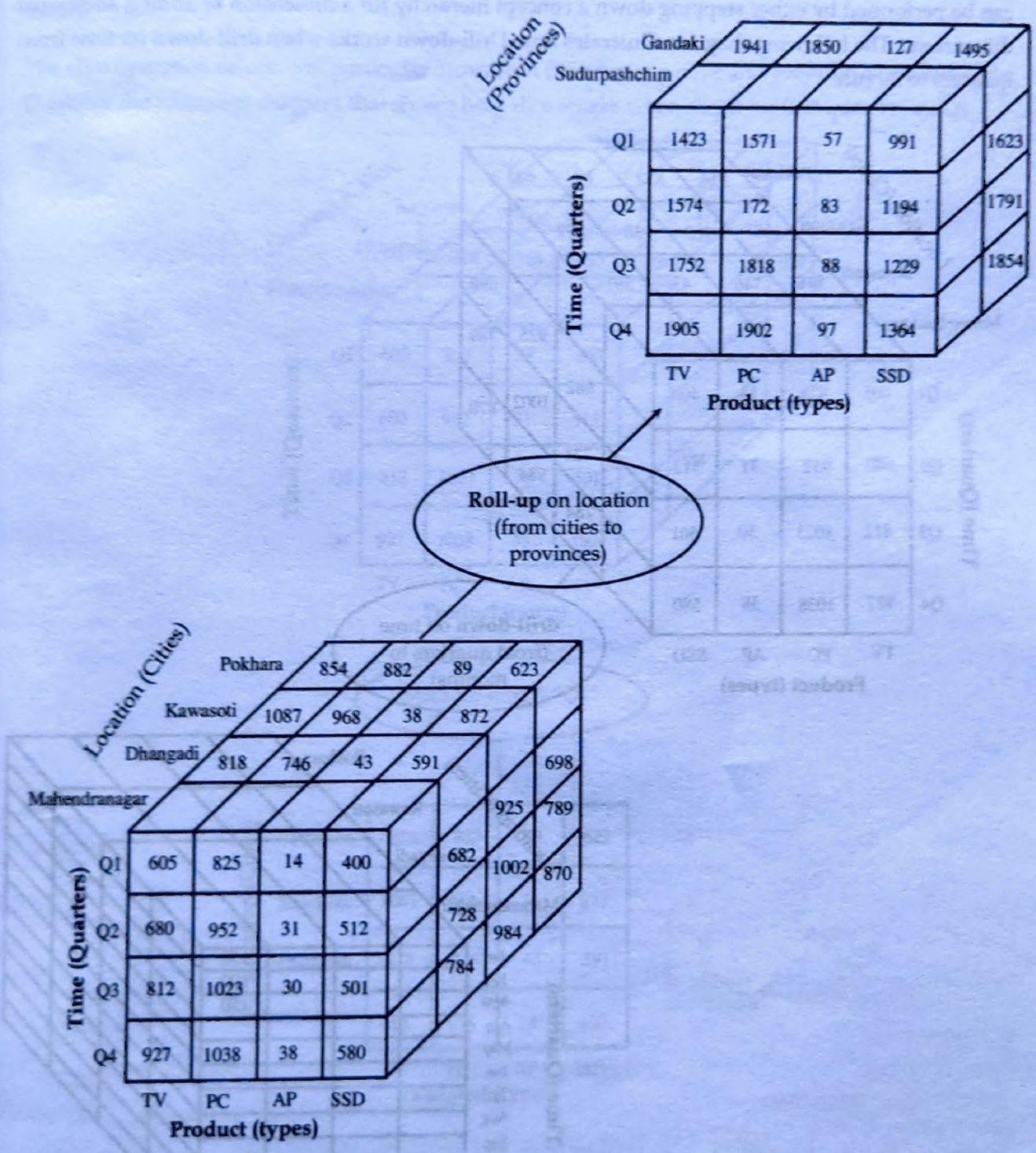

Figure 1.10: Illustration of roll-up operations on sales data

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 province.

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 record to more detailed data. Drill-down can be performed by either stepping down a concept hierarchy for a dimension or adding additional dimensions. The following diagram illustrates how Drill-down works when drill-down on time from quarters to month:

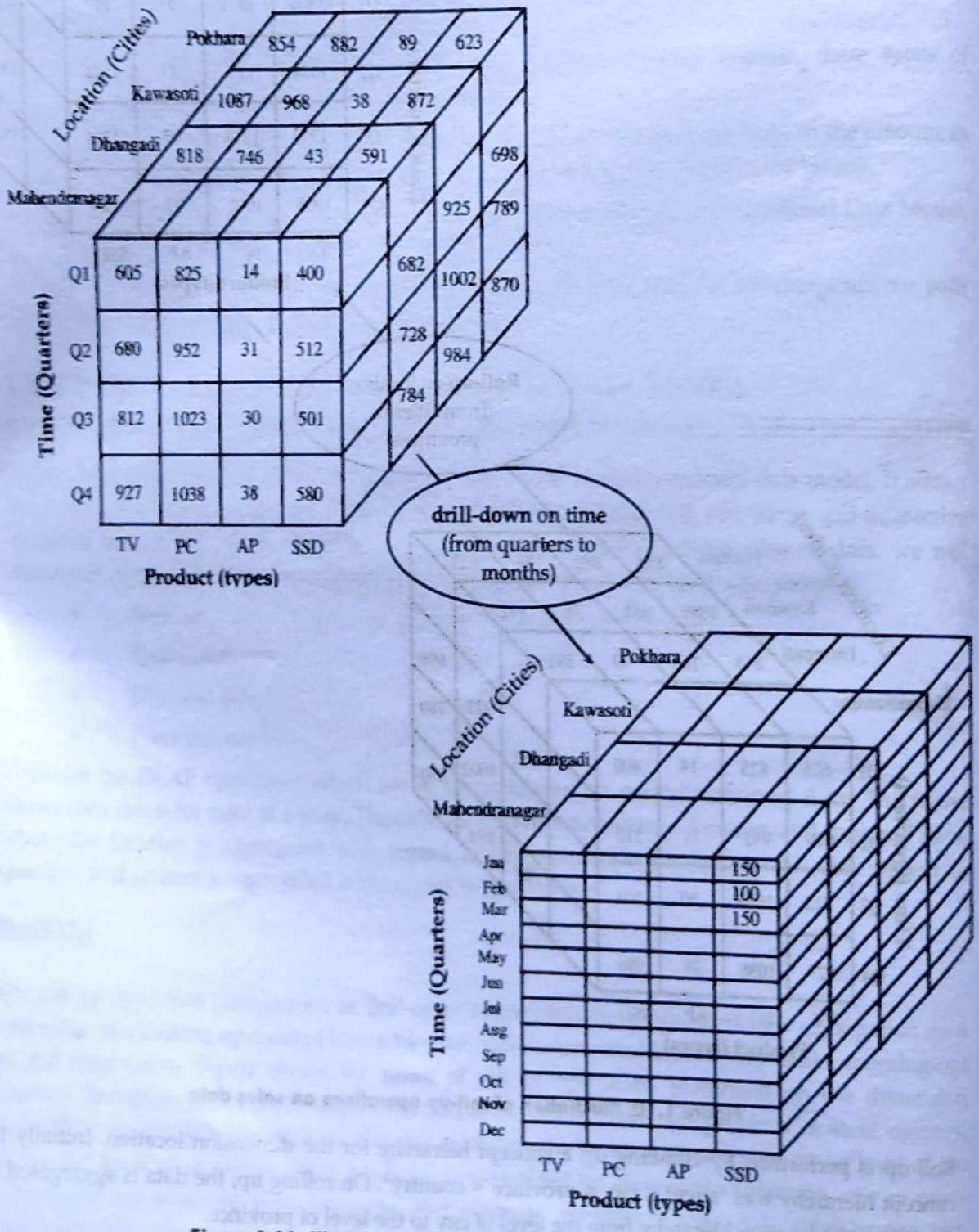

Figure 1.11: Illustration of drill-down operation on sales data

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.

Slice

The slice operation selects one particular dimension from a given cube and provides a new sub-cube. Consider the following diagram that shows how slice works when sliced for first quarter i.e., Q1.

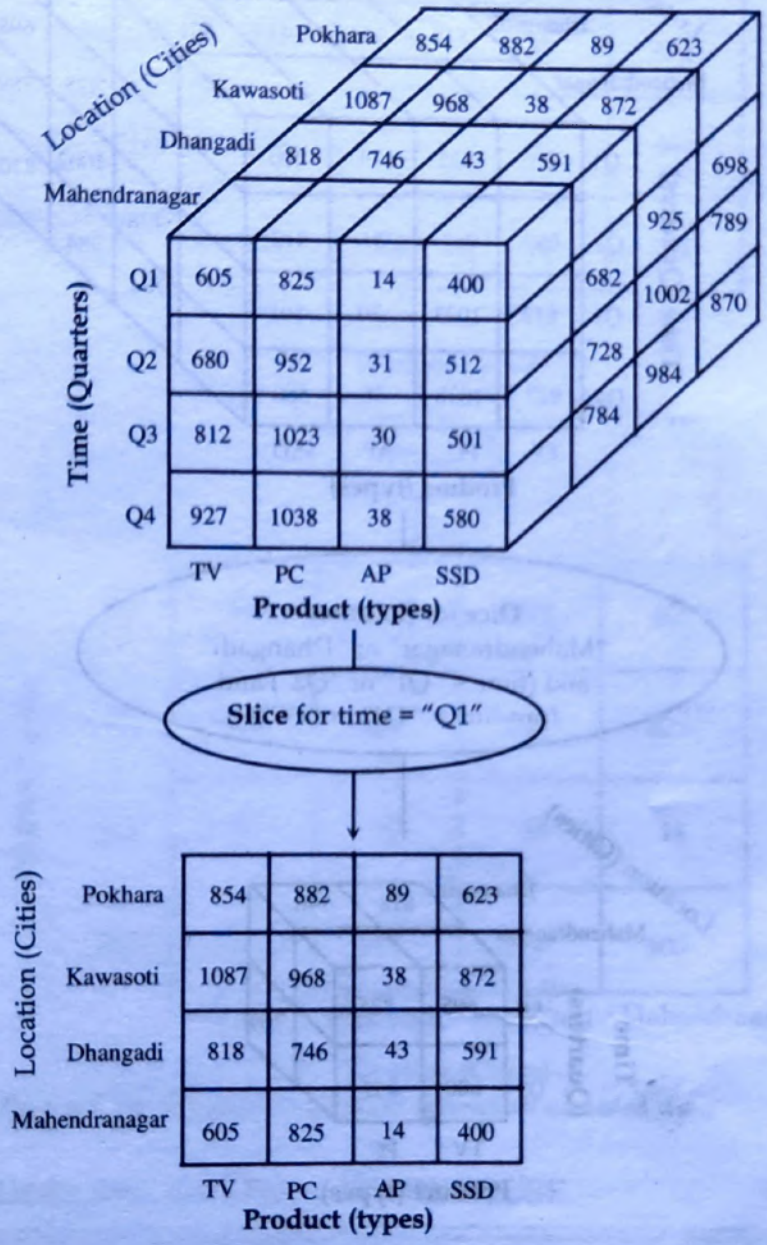

Figure 1.12: Illustration of slice operation on sales data

Here, Slice is performed for the dimension "time" using the criterion time = "Q1". It will form a new sub-cube by selecting one or more dimensions.

Dice

Dice selects two or more dimensions from a given cube and provides a new sub-cube. Consider the following diagram that shows the dice operation.

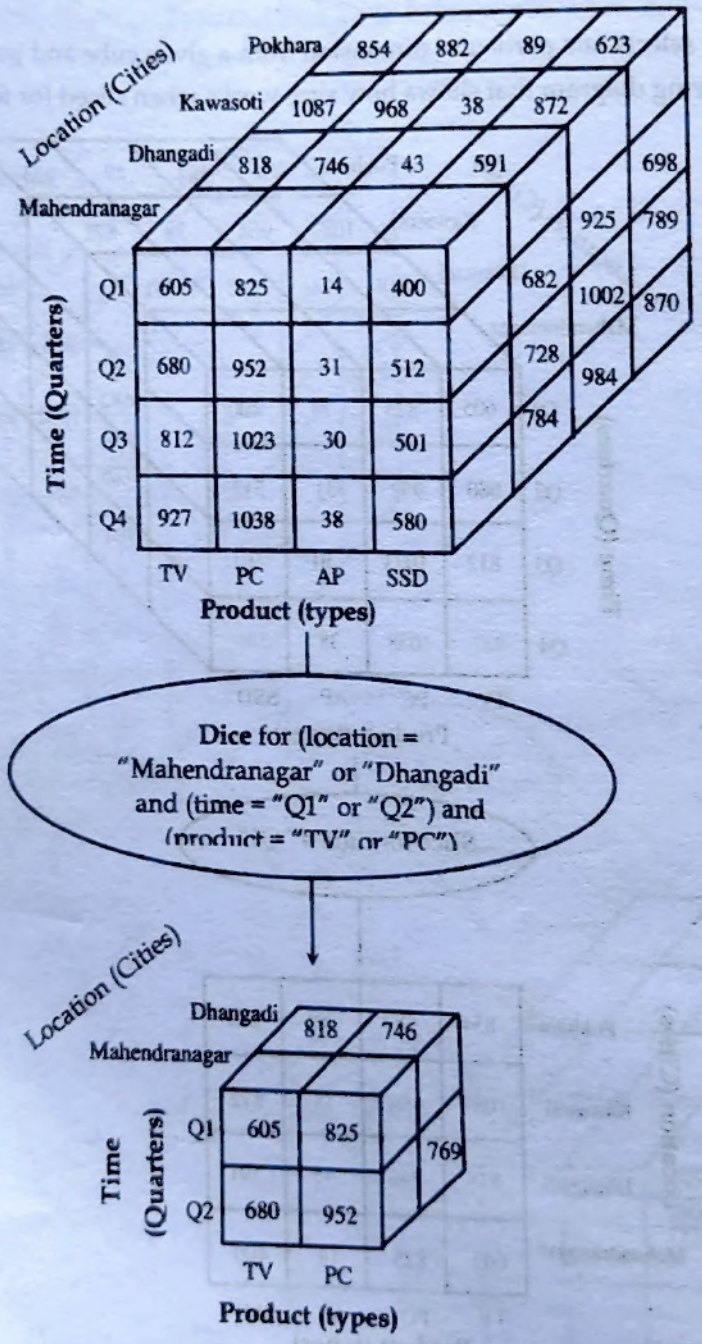

Figure 1.13: Illustration of dice operation on sales data

The dice operation on the cube based on the following selection criteria involved three dimensions are

- (location = "Mahendranagar" or "Dhangadi") and
- (time = "Q1" or "Q2") and
- (item =" TV" or "PC").

Pivot

The pivot operation is also known as rotation. It rotates the data axes in view in order to provide an alternative presentation of data. Consider the following diagram that shows the pivot operation between location and product dimension.

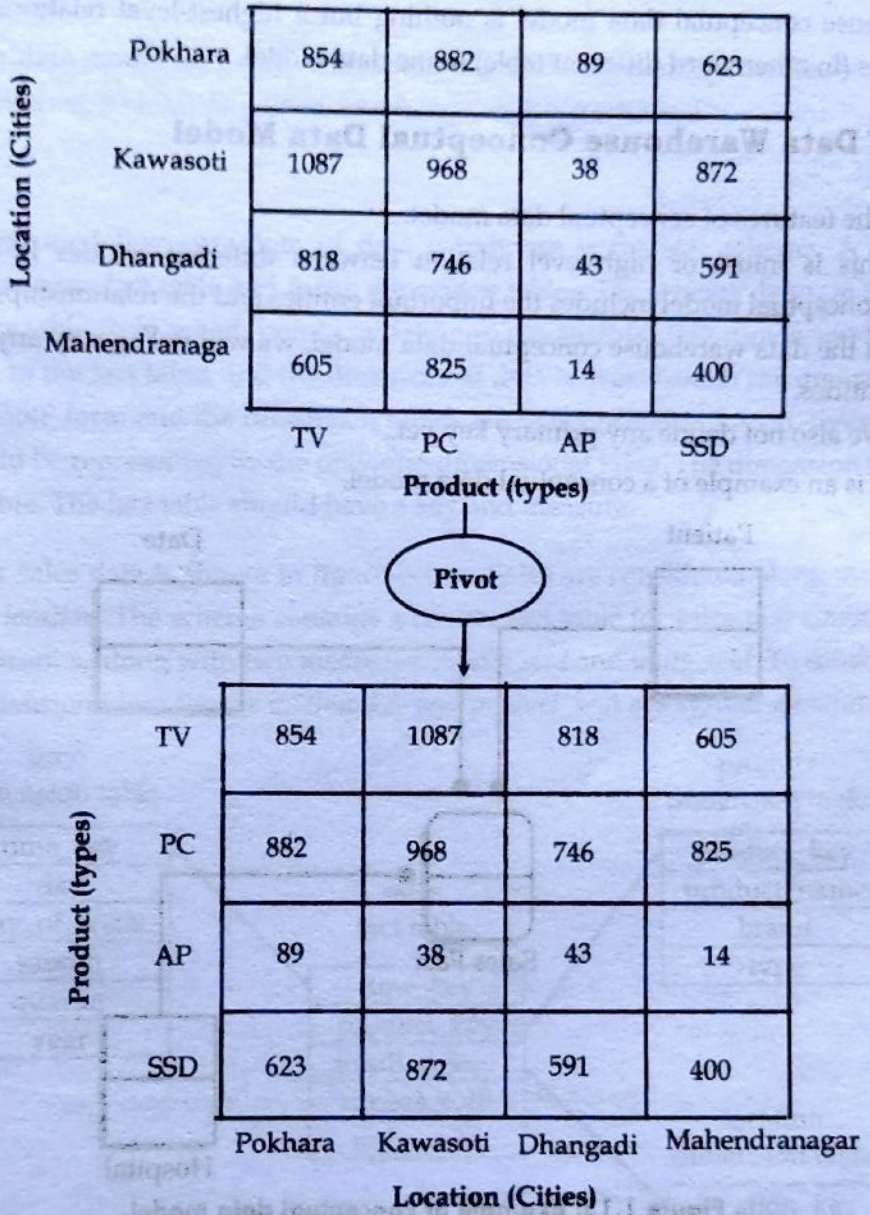

Figure 1.14: Illustration of pivot operation on sales data

CONCEPTUAL MODELING OF DATA WAREHOUSE

The conceptual data model is a structured business view of the data required to support business processes, record business events, and track related performance measures. This model focuses on identifying the data used in the business but not its processing flow or physical characteristics. It is a concise description of the user's data requirements without taking into account implementation details. Conventional databases are generally designed at the conceptual level using some variation of the well-known entity-relationship (ER) model, although the Unified Modeling Language (UML) is being

increasingly used. Conceptual schemas can be easily translated to the relational model by applying a set of mapping rules. Providing extensions to the ER and the UML models for data warehouses is not really a solution to the problem, since ultimately, they represent a reflection and visualization of the underlying relational technology concepts and, in addition, reveal their own problems. Therefore, conceptual data warehousing modeling requires a model that clearly stands on top of the logical level.

A Data warehouse conceptual data model is nothing but a highest-level relationship between the different entities (in other word different table) in the data model.

Features of Data Warehouse Conceptual Data Model

Following are the features of conceptual data model:

- This is initial or high-level relation between different entities in the data model. Conceptual model includes the important entities and the relationships among them.
- In the data warehouse conceptual data model, we will not specify any attributes to the entities.
- We also not define any primary key yet.

The figure 1.15 is an example of a conceptual data model.

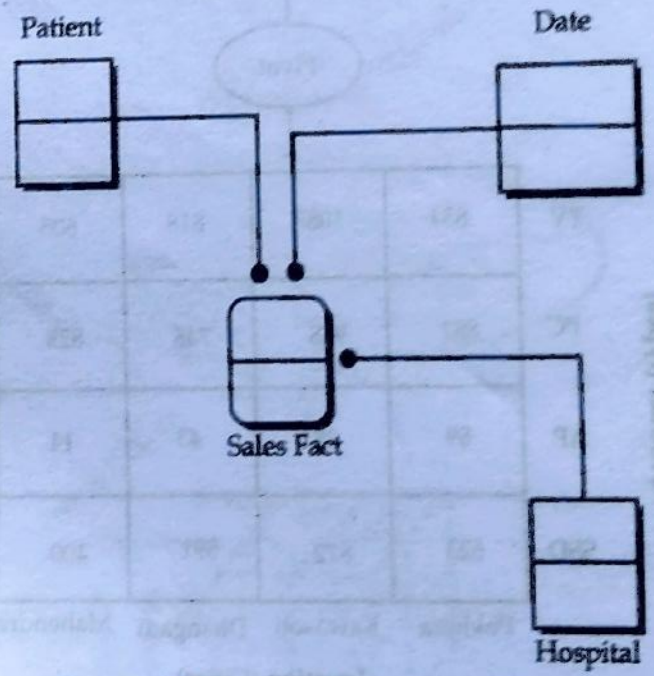

Figure 1.15: Example of conceptual data model.

From the above figure(see figure 1.15) you can see that, data warehouse conceptual model describes only high-level relationship between the entities.

Schemas for Multidimensional Data Models

A schema is a logical description that describes the entire database. In the data warehouse there includes the name and description of records. It has all data items and also different aggregates associated with the data. Like a database has a schema, it is required to maintain a schema for a data warehouse as well. There are different schemas based on the setup and data which are maintained in a data warehouse.

There are fact tables and dimension tables that form the basis of any schema in the data warehouse that are important to be understood. The fact tables should have data corresponding data to any business process. Every row represents any event that can be associated with any process. It stores quantitative information for analysis. A dimension table stores data about how the data in fact table is being analyzed. They facilitate the fact table in gathering different dimensions on the measures which are to be taken.

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.*

Star Schema

The most common modeling paradigm of data warehouse is the star schema. A star schema is represented by one large fact table and many dimension tables. The schema diagram looks like a star with a central fact table from which points radiating to the surrounding dimension tables. The fact data is organized in the fact table, and the dimensional data is organized in the dimension table. The fact tables are in 3NF form and the dimension tables are in denormalized form. Every dimension in star schema should be represented by the only one-dimensional table. The dimension table should be joined to a fact table. The fact table should have a key and measure.

A star schema for sales data is shown in figure below. Sales are considered along three dimensions: *product, time* and *location*. The schema contains a central fact table for sales that contains key to each of the three dimensions, along with two measures: *rupees_sold* and *units_sold*. To minimize the size of the fact table, dimension identifiers (e.g., *time_key* and *product_key*) are system-generated identifiers.

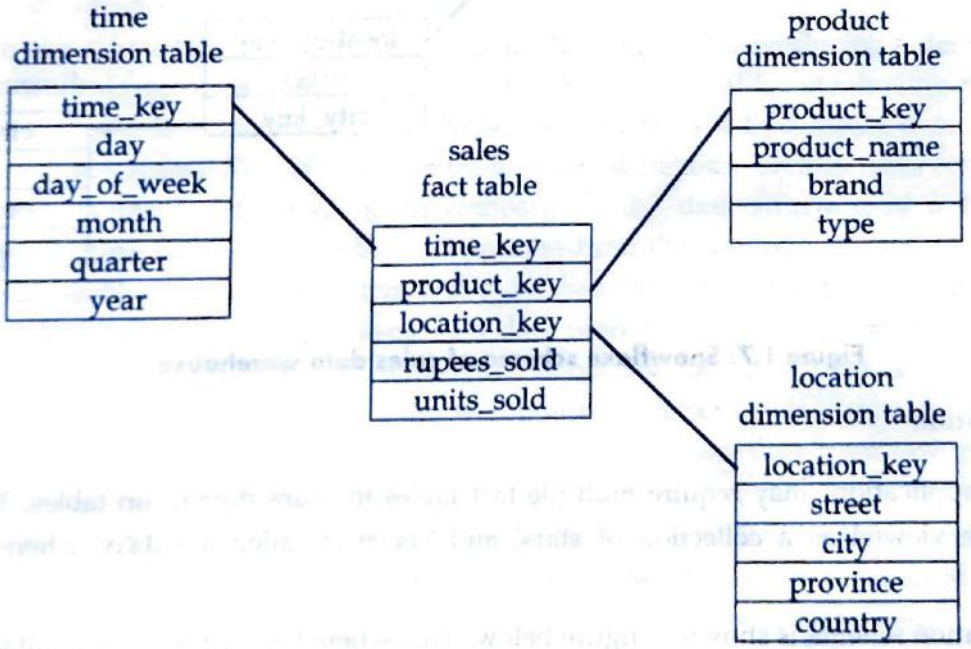

Figure 1.6: Star schema of sales data warehouse.

Snowflake Schema

Snowflake schema can be considered as a variant of the star schema. However, this is a more complex data model compared to the star schema. In a snowflake schema, there is single, large and

central fact table and one or more tables for each dimension. In order to eliminate redundancy, dimension tables split data into different tables. Due to this normalization, often it results in more complex queries and reduced query performance. The advantage of snowflake schema is that it uses small disk space. The implementation of dimensions is easy when they are added to this schema. The same set of attributes are published by different sources.

A snowflake schema for sales data is shown in figure below. Sales are considered along three dimensions: *product, time* and *location*. The fact table is identical to star schema. The main difference between the two schemas is in the definition of dimension tables. The single dimension table for location in the star schema can be normalized into two new tables: *location* and *city*. The city key in the new location table links to the city dimension as shown in figure below.

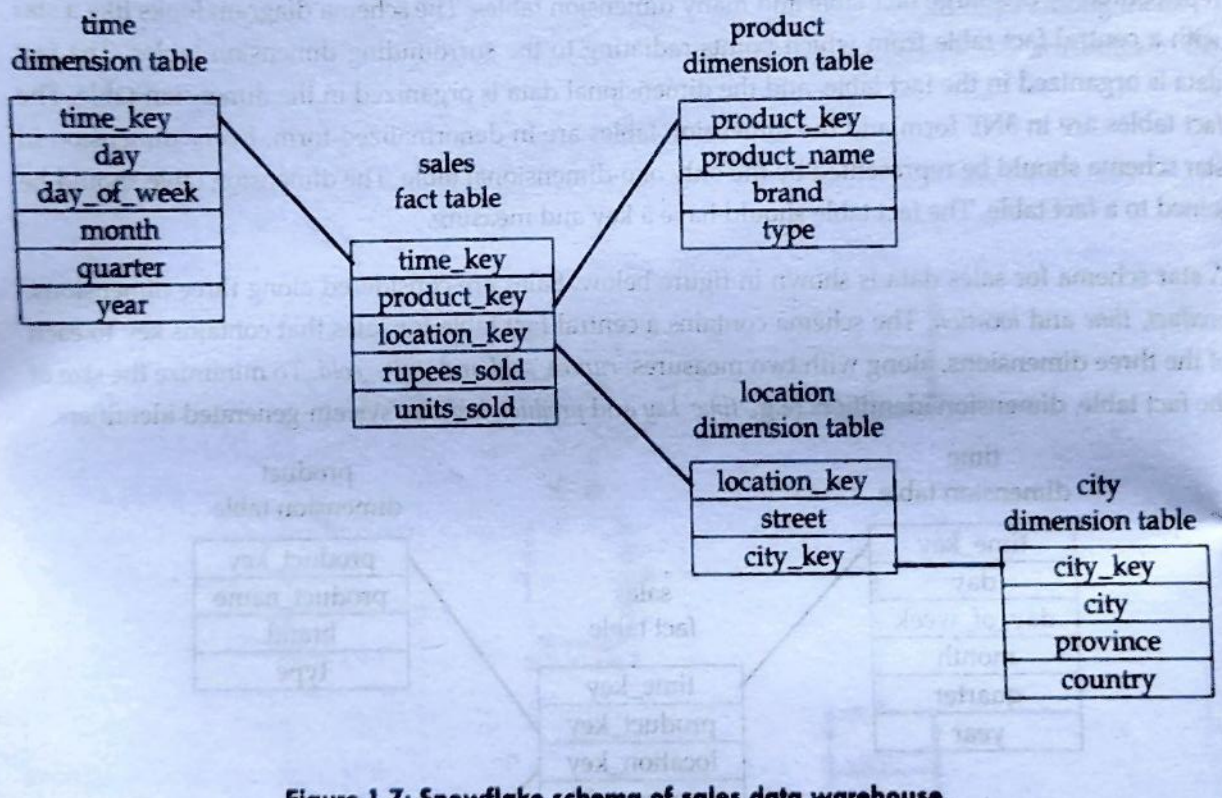

Figure 1.7: Snowflake schema of sales data warehouse.

Fact Constellation

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 is shown in figure below. This schema specifies two fact tables, *sales* and *shipping*. The sales table definition is identical to that of the star schema. The shipping table has five dimensions, or keys: *product_key, time_key, shipper_key, from location, and to location* and two measures: *rupees_cost, and units_shipped*. A fact constellation schema allows dimension tables to be shared between fact tables. For example, the dimensions tables for time, product, and location are shared between the *sales* and *shipping fact tables*.

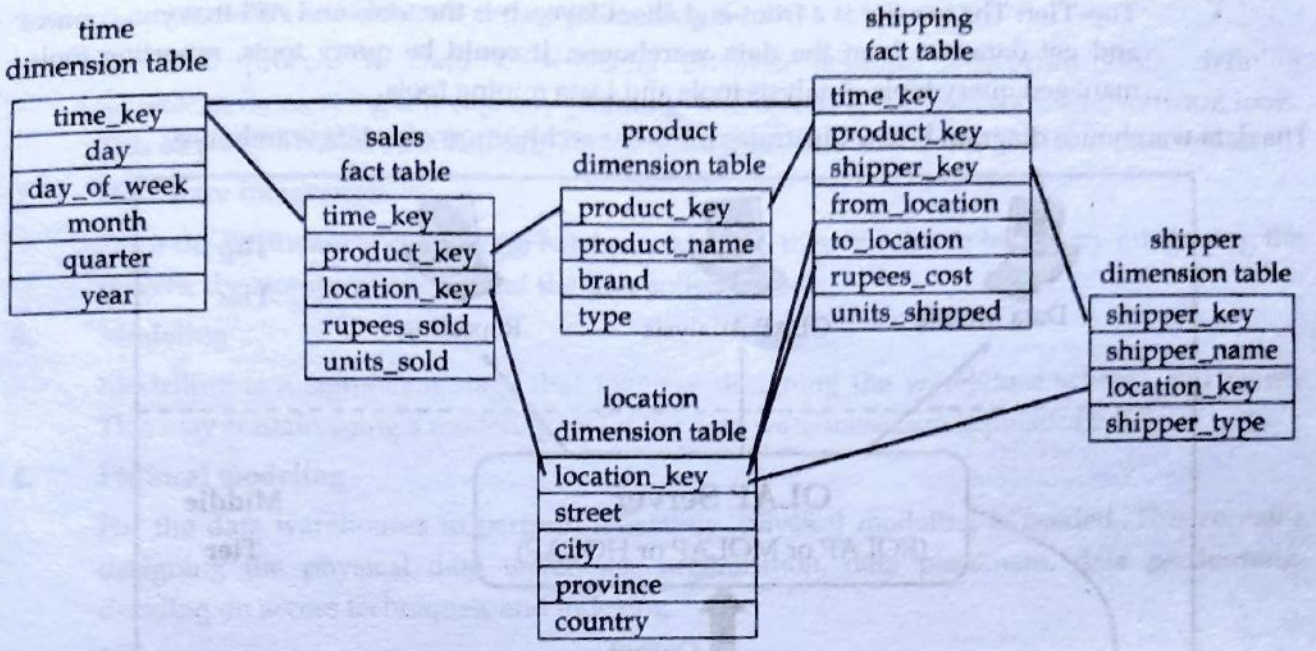

Figure 1.8: Fact constellation schema of sales and shipping data warehouse.

ARCHITECTURE OF DATA WAREHOUSE

Data Warehouse Architecture is complex as it's an information system that contains historical and commutative data from multiple sources. There are 3 approaches for constructing Data Warehouse layers: *Single tier, Two tier* and *Three tier.*

The structure of a **single-tier** data warehouse architecture centers on producing a dense set of data and reducing the volume of data deposited. Although it is beneficial for eliminating redundancies, this type of warehouse architecture is not suitable for businesses with complex data requirements and numerous data streams. This is where multi-tier data warehouse architectures come in as they deal with more complex data streams. In comparison, the data structure of a **two-tier** data warehouse architecture splits the tangible data sources from the warehouse itself. Unlike a single-tier, the two-tier architecture uses a system and a database server. This is most commonly used in small organizations where a server is used as a data mart. Although it is more efficient at data storage and organization, the two-tier architecture is not scalable. Moreover, it only supports a nominal number of users. **Three-tier** data warehouse architecture is the most widely used architecture of data warehouse as it produces a well-organized data flow from raw information to valuable insights. It consists of the **Top, Middle** and **Bottom Tier**.

The top tier is the front-end client that presents results through reporting, analysis, and data mining tools. The middle tier consists of the analytics engine that is used to access and analyze the data. The bottom tier of the architecture is the database server, where data is loaded and stored.

- **Bottom Tier:** The database of the data warehouse servers as the bottom tier. It is usually a relational database system. Data is cleansed, transformed, and loaded into this layer using back-end tools.
- **Middle Tier:** The middle tier in data warehouse is an OLAP server which is implemented using either ROLAP or MOLAP or HOLAP model. For a user, this application tier presents an abstracted view of the database. This layer also acts as a mediator between the end-user and the database.

- **Top-Tier:** The top tier is a front-end client layer. It is the tools and API that you connect and get data out from the data warehouse. It could be query tools, reporting tools, managed query tools, Analysis tools and Data mining tools.

The data warehouse diagram below illustrates the 3-tier architecture of a data warehouse.

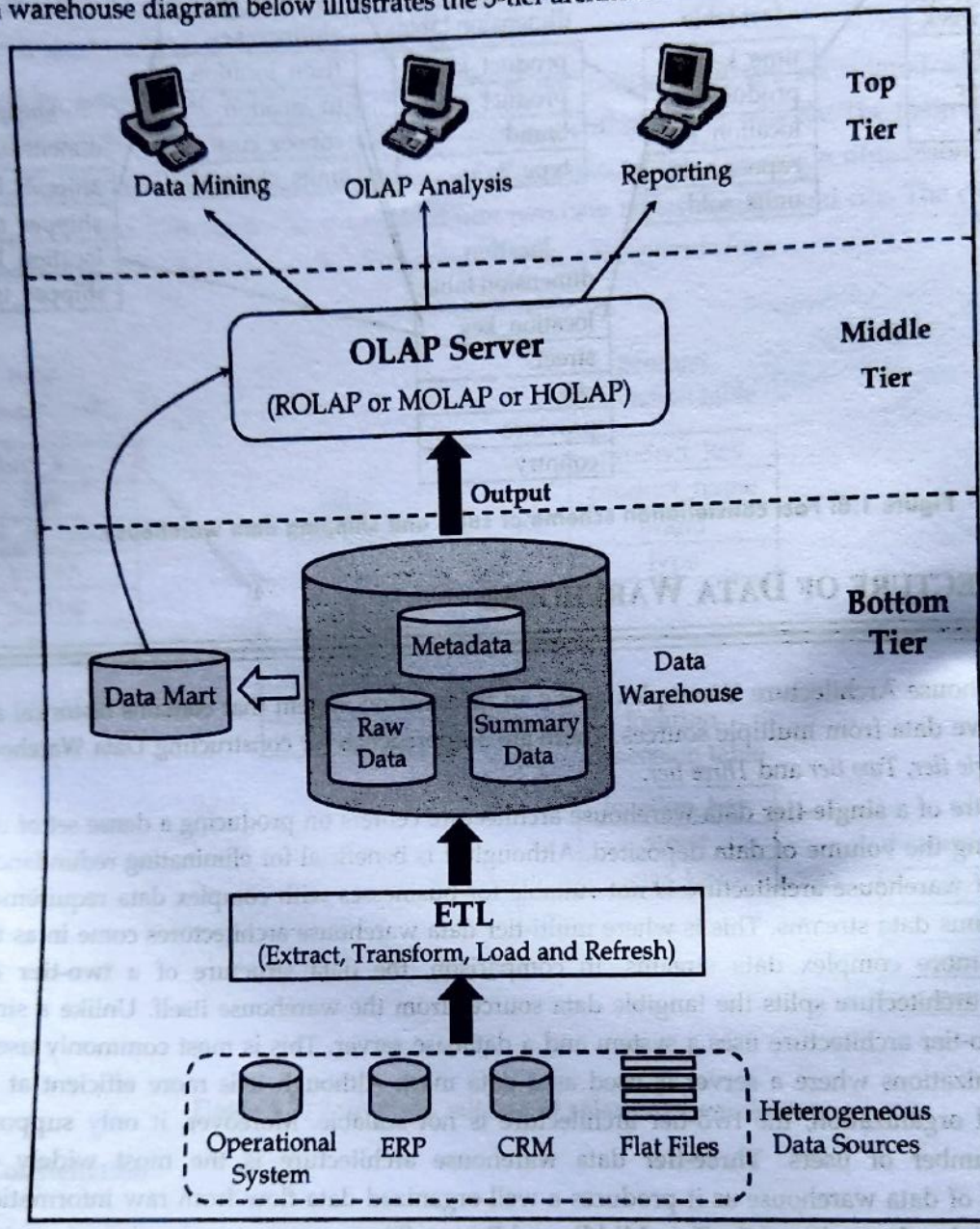

Figure 1.19: Three-Tier data warehouse architecture

DATA WAREHOUSE IMPLEMENTATION

Data Warehouse Implementation is a series of activities that are essential to create a fully functioning Data Warehouse, after classifying, analyzing and designing the Data Warehouse with respect to the requirements provided by the client. The process of establishing and implementing a data warehouse system in an organization is known as data warehouse implementation. Data warehousing is one of the most important components of the business intelligence process for an organization. The data warehousing implementation process requires a series of steps that need to be followed in a very effective manner. The processes are as follows:

1. **Requirement's analysis and capacity planning**

   The first process in data warehousing involves defining enterprise needs, defining architectures, carrying out capacity planning, and selecting the hardware and software tools. This step will contain be consulting senior management as well as the different stakeholder.

2. **Hardware integration**

   Once the hardware and software has been selected, they require to be put by integrating the servers, the storage methods, and the user software tools.

3. **Modeling**

   Modelling is a significant stage that involves designing the warehouse schema and views. This may contain using a modeling tool if the data warehouses are sophisticated.

4. **Physical modeling**

   For the data warehouses to perform efficiently, physical modeling is needed. This contains designing the physical data warehouse organization, data placement, data partitioning, deciding on access techniques, and indexing.

5. **Sources**

   The information for the data warehouse is likely to come from several data sources. This step contains identifying and connecting the sources using the gateway, ODBC drives, or another wrapper.

6. **ETL**

   The data from the source system will require to go through an ETL phase. The process of designing and implementing the ETL phase may contain defining a suitable ETL tool vendor and purchasing and implementing the tools. This may contains customize the tool to suit the need of the enterprises.

7. **Populate the data warehouses**

   Once the ETL tools have been agreed upon, testing the tools will be needed, perhaps using a staging area. Once everything is working adequately, the ETL tools may be used in populating the warehouses given the schema and view definition.

8. **User application**

   For the data warehouses to be helpful, there must be end-user applications. This step contains designing and implementing applications required by the end-users.

9. **Roll-out the warehouses and applications**

   Once the data warehouse has been populated and the end-client applications tested, the warehouse system and the operations may be rolled out for the user's community to use.

DATA MARTS

A data mart is a subset of a data warehouse oriented to a specific business line. Data marts contain repositories of summarized data collected for analysis on a specific section or unit within an organization E.g., Marketing, Sales, HR or finance. It is often controlled by a single department in an organization. Data Mart usually draws data from only a few sources compared to a Data warehouse. Data marts are small in size and are more flexible compared to a Data warehouse.

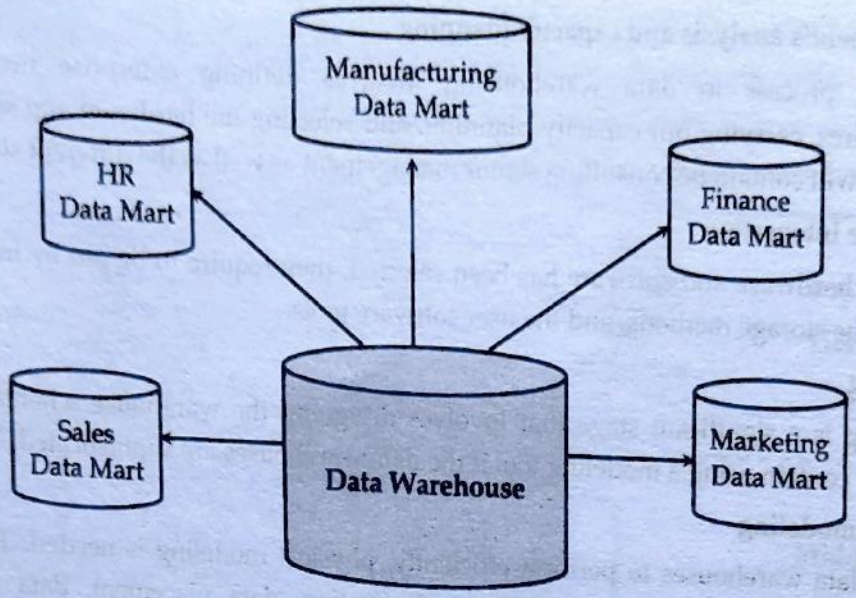

Figure 1.20: Data mart

Why do we need Data Mart?

- Data Mart helps to enhance user's response time due to reduction in volume of data
- It provides easy access to frequently requested data.
- Data mart are simpler to implement when compared to corporate Datawarehouse. At the same time, the cost of implementing Data Mart is certainly lower compared with implementing a full data warehouse.
- Compared to Data Warehouse, a datamart is agile. In case of change in model datamart can be built quicker due to a smaller size.
- A Datamart is defined by a single Subject Matter Expert. On the contrary data warehouse is defined by interdisciplinary SME from a variety of domains. Hence, Data mart is more open to change compared to Datawarehouse.
- Data is partitioned and allows very granular access control privileges.
- Data can be segmented and stored on different hardware/software platforms.

Types of Data Mart

There are three main types of data mart:

- **Dependent:** Dependent data marts are created by drawing data directly from operational, external or both sources.
- **Independent:** Independent data mart is created without the use of a central data warehouse.
- **Hybrid:** This type of data marts can take data from data warehouses or operational systems.

Dependent Data Mart

A dependent data mart allows sourcing organization's data from a single Data Warehouse. It is one of the data marts examples which offers the benefit of centralization. If you need to develop one or more physical data marts, then you need to configure them as dependent data marts. Dependent Data Mart in data warehouse can be built in two different ways. Either where a user can access both

the data mart and data warehouse, depending on need, or where access is limited only to the data mart. The second approach is not optimal as it produces sometimes referred to as a data junkyard. In the data junkyard, all data begins with a common source, but they are scrapped, and mostly junked.

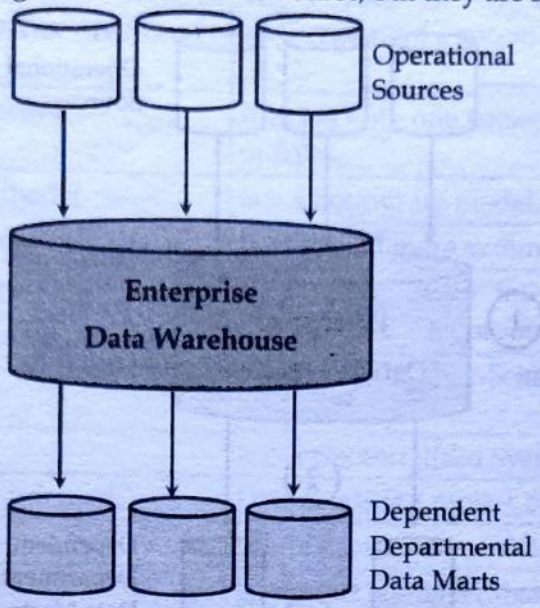

Figure 1.21: Dependent data mart

Independent Data Mart

An independent data mart is created without the use of central Data warehouse. This kind of Data Mart is an ideal option for smaller groups within an organization.

An independent data mart has neither a relationship with the enterprise data warehouse nor with any other data mart. In Independent data mart, the data is input separately, and its analyses are also performed autonomously.

Implementation of independent data marts is antithetical to the motivation for building a data warehouse. First of all, you need a consistent, centralized store of enterprise data which can be analyzed by multiple users with different interests who want widely varying information.

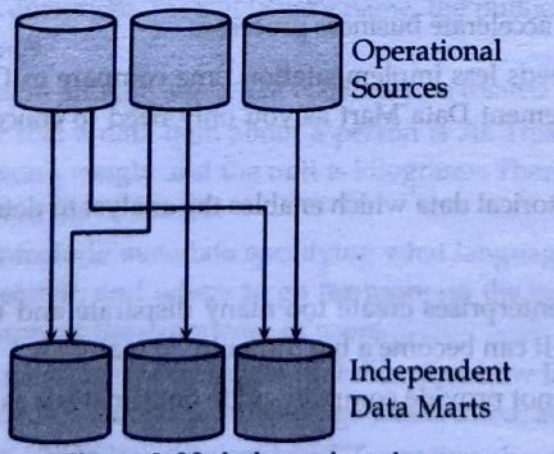

Figure 1.22: Independent data mart

Hybrid Data Mart

A hybrid data mart combines input from sources apart from Data warehouse. This could be helpful when you want ad-hoc integration, like after a new group or product is added to the organization. It is the best data mart example suited for multiple database environments and fast implementation

turnaround for any organization. It also requires least data cleansing effort. Hybrid Data mart also supports large storage structures, and it is best suited for flexible for smaller data-centric applications.

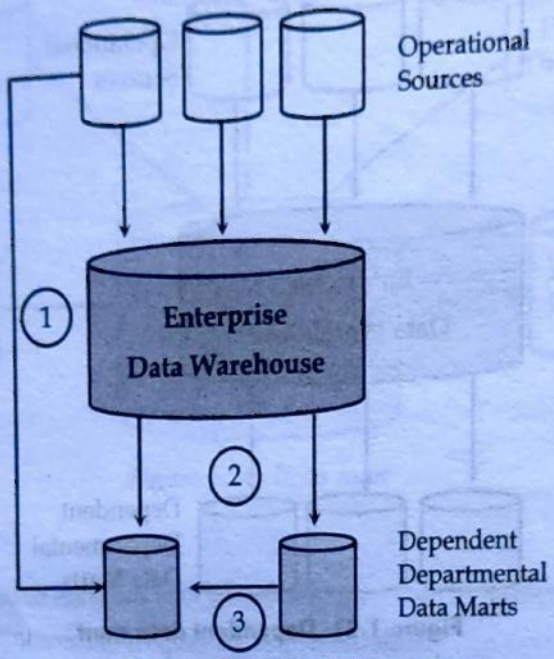

Figure 1.23: Hybrid data mart

Advantages and Disadvantages of a Data Mart

Advantages

- Data marts contain a subset of organization-wide data. This Data is valuable to a specific group of people in an organization.
- It is cost-effective alternatives to a data warehouse, which can take high costs to build.
- Data Mart allows faster access of Data.
- Data Mart is easy to use as it is specifically designed for the needs of its users. Thus, a data mart can accelerate business processes.
- Data Marts needs less implementation time compare to Data Warehouse systems. It is faster to implement Data Mart as you only need to concentrate the only subset of the data.
- It contains historical data which enables the analyst to determine data trends.

Disadvantages

- Many times, enterprises create too many disparate and unrelated data marts without much benefit. It can become a big difficulty to maintain.
- Data Mart cannot provide company-wide data analysis as their data set is limited.

Difference Between Data Warehouse and Data Mart

Data warehouses are built to serve as the central store of data for the entire business, whereas a data mart fulfills the request of a specific division or business function. The major differentiate between data warehouse and data mart are tabulated below:

Data Warehouse	Data Mart
A Data Warehouse is a vast repository of information collected from various organizations or departments within a corporation.	A data mart is an only subtype of a Data Warehouses. It is architecture to meet the requirement of a specific user group.
It may hold multiple subject areas.	It holds only one subject area. For example, Finance or Sales.
Data warehouse is top-down model.	It is a bottom-up model.
It holds very detailed information.	It may hold more summarized data.
Works to integrate all data sources	It concentrates on integrating data from a given subject area or set of source systems.
In data warehousing, Fact constellation is used.	In Data Mart, Star Schema and Snowflake Schema are used.
It is a Centralized System.	It is a Decentralized System.
Data Warehousing is the data-oriented.	Data Marts is a project-oriented.
Data Ware house has long life	While data-mart has short life than warehouse.
Data Warehouse is vast in size.	Data mart is smaller than warehouse.
In data warehouse, Fact constellation schema is used.	While in this, Star schema and snowflake schema are used.

META DATA

Metadata is data about the data or documentation about the information which is required by the users. In data warehousing, metadata is one of the essential aspects. Several examples of Meta data are listed below:

- A library catalog may be considered metadata. The directory metadata consists of several predefined components representing specific attributes of a resource, and each item can have one or more values. These components could be the name of the author, the name of the document, the publisher's name, the publication date, and the methods to which it belongs.
- The table of content and the index in a book may be treated metadata for the book.
- Suppose we say that a data item about a person is 70. This must be defined by noting that it is the person's weight and the unit is kilograms. Therefore, (weight, kilograms) is the metadata about the data is 70.
- A webpage may include metadata specifying what language it is written in, what tools were used to create it, and where to go for more on the subject, allowing browsers to automatically improve the experience of users.
- A digital image may include metadata that describes how large the picture is, the color depth, the image resolution, when the image was created, and other data.
- A text document's metadata may contain information about how long the document is, who the author is, when the document was written, and a short summary of the document.
- Another example of metadata are data about the tables and figures in a report like this book. A table (which is a record) has a name (e.g., table titles), and there are column names of the tables that may be treated metadata. The figures also have titles or names.

Key features of Metadata are described as following:

- The location and descriptions of warehouse systems and components.
- Names, definitions, structures, and content of data-warehouse and end-users views.
- Identification of authoritative data sources.
- Integration and transformation rules used to populate data.
- Integration and transformation rules used to deliver information to end-user analytical tools.
- Subscription information for information delivery to analysis subscribers.
- Metrics used to analyze warehouses usage and performance.
- Security authorizations, access control list, etc.

Metadata is used for building, maintaining, managing, and using the data warehouses. Metadata allow users access to help understand the content and find data.

COMPONENTS OF DATA WAREHOUSE

A typical data warehouse has four main components: a central database, ETL (extract, transform, and load) tools, metadata, and access tools. All of these components are engineered for speed so that we can get results quickly and analyze data on the fly.

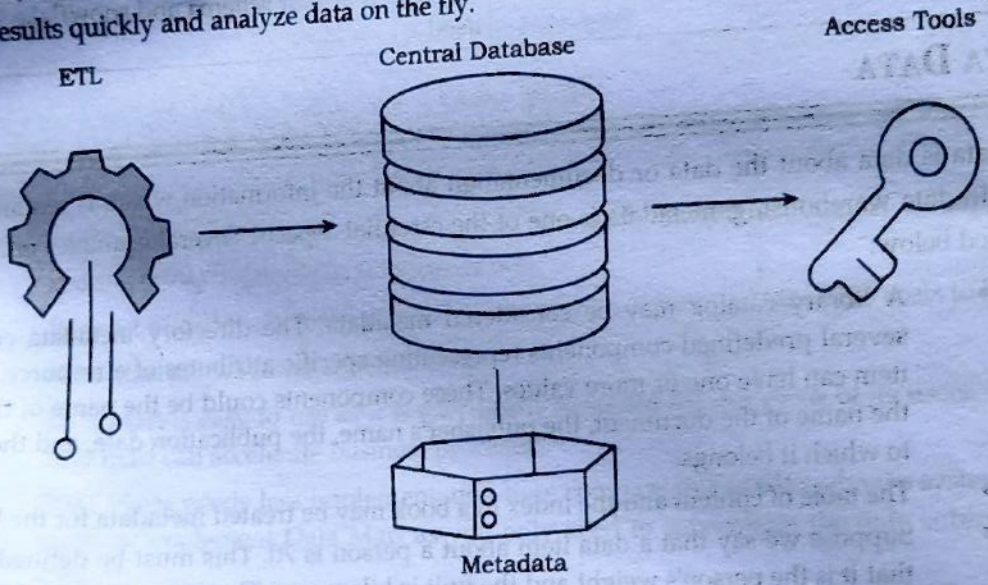

Figure 1.24: Components of data warehouse

The figure 1.24 shows the essential elements of a typical warehouse. We see the ETL 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 warehouses data. This element not only stores and manages the data; it also keeps track of data using the metadata repository. The Information Delivery component shows on the right consists of all the different ways of making the information from the data warehouses available to the users. The major four components of Datawarehouse are listed below:

1. **Central database**

   A database serves as the foundation of your data warehouse. Traditionally, these have been standard relational databases running on premise or in the cloud. But because of Big Data, the need for true, real-time performance, and a drastic reduction in the cost of RAM, in-memory databases are rapidly gaining in popularity.

2. **Data integration**
   Data is pulled from source systems and modified to align the information for rapid analytical consumption using a variety of data integration approaches such as ETL (extract, transform, load) and ELT as well as real-time data replication, bulk-load processing, data transformation, and data quality and enrichment services.
3. **Metadata**
   Metadata is data about your data. It specifies the source, usage, values, and other features of the datasets in your data warehouse. There is business metadata, which adds context to your data, and technical metadata, which describes how to access data – including where it resides and how it is structured.
4. **Data warehouse access tools**
   Access tools allow users to interact with the data in your data warehouse. Examples of access tools include: query and reporting tools, application development tools, data mining tools, and OLAP tools.

NEED FOR DATA WAREHOUSING

A well-designed data warehouse is the foundation for any successful BI or analytics program. Its main job is to power the reports, dashboards, and analytical tools that have become indispensable to businesses today. A data warehouse provides the information for your data-driven decisions – and helps you make the right call on everything from new product development to inventory levels. There are many benefits of a data warehouse. Here are just a few:

- **Better business analytics**
  With data warehousing, decision-makers have access to data from multiple sources and no longer have to make decisions based on incomplete information.
- **Faster queries**
  Data warehouses are built specifically for fast data retrieval and analysis. With a data warehouse, we can very rapidly query large amounts of consolidated data with little to no support from IT.
- **Improved data quality**
  Before being loaded into the data warehouse, data cleansing cases are created by the system and entered in a worklist for further processing, ensuring data is transformed into a consistent format to support analytics – and decisions – based on high quality, accurate data.
- **Historical insight**
  By storing rich historical data, a data warehouse lets decision-makers learn from past trends and challenges, make predictions, and drive continuous business improvement.

TRENDS IN DATA WAREHOUSING

Data warehouses have come a long way since their earliest iterations back in the 1980s. They're now faster, more powerful, and in the cloud. But what hasn't changed is their goal: to unlock the full value of an organization's data. The latest developments are only making this easier with automation, empowerment, and openness. Trends in data warehousing are listed below:

- Continued Growth in Data warehousing
- Data warehouse has become Mainstream
- Industries using Data warehouse
- Vendor Solution & Products
- Status of Data warehouse market
- Significant Trends
- Web Enabled Data warehouse

Continued Growth in Data Warehousing

Data warehousing is no longer a purely novel idea for study and experimentation. It has become mainstream. True, the data warehouse is not in every dentist's office yet, but neither is it confined only to high-end businesses. More than half of all U.S. companies have made a commitment to data warehousing. About 90% of multinational companies have data warehouses or are planning to implement data warehouses in the next few months.

Even during the first few years of data warehousing in the late 1990s, hundreds of vendors had flooded the market with numerous products. Vendor solutions and products run the gamut of data warehousing: data modeling, data acquisition, data quality, data analysis, metadata, and so on. A buyer's guide published by the Data Warehousing Institute at that time featured no fewer than 105 leading products. The market is huge and continues to grow in revenue dollars.

Data Warehouse Has Become Mainstream

In the early stages, four significant factors drove many companies to move into data warehousing:

- Fierce competition
- Government deregulation
- Need to revamp internal processes
- Imperative for customized marketing

Telecommunications, banking, and retail were the first industries to adopt data warehousing. That was largely because of government deregulation in telecommunications and banking. Retail businesses moved into data warehousing because of fiercer competition. Utility companies joined the group as that sector was deregulated. The next wave of businesses to get into data warehousing consisted of companies in financial services, health care, insurance, manufacturing, pharmaceuticals, transportation, and distribution.

Industries Using Data Warehouse

Although earlier data warehouses concentrated on keeping summary data for high-level analysis, we now see larger and larger data warehouses being built by different businesses. Now companies have the ability to capture, cleanse, maintain, and use the vast amounts of data generated by their business transactions. The quantities of data kept in data warehouses continue to swell to the terabyte range. Data warehouses storing several terabytes of data are not uncommon in retail and telecommunications.

Vendor Solution & Products

As an information technology professional, you are familiar with database vendors and database products. In the same way, you are familiar with most of the operating systems and their vendors. How many leading database vendors are there? How many leading vendors of operating systems are there? A handful? The number of database and operating system vendors pales in comparison with data warehousing products and vendors. There are hundreds of data warehousing vendors and thousands of data warehousing products and solutions.

In the beginning, the market was filled with confusion and vendor hype. Every vendor, small or big, that had any product remotely connected to data warehousing jumped on the bandwagon. Data warehousing meant what each vendor defined it to be. Each company positioned its own products as the proper set of data warehousing tools. Data warehousing was a new concept for many of the businesses that adopted it. These businesses were at the mercy of the marketing hype of the vendors.

Status of Data Warehouse Market

With so many vendors and products, how can we classify the vendors and products, and thereby make sense of the market? It is best to separate the market broadly into two distinct groups. The first group consists of data warehouse vendors and products catering to the needs of corporate data warehouses in which all enterprise data is integrated and transformed. This segment has been referred to as the market for strategic data warehouses. This segment accounts for about a quarter of the total market. The second segment is looser and more dispersed, consisting of departmental data marts, fragmented database marketing systems, and a wide range of decision support systems. Specific vendors and products dominate each segment.

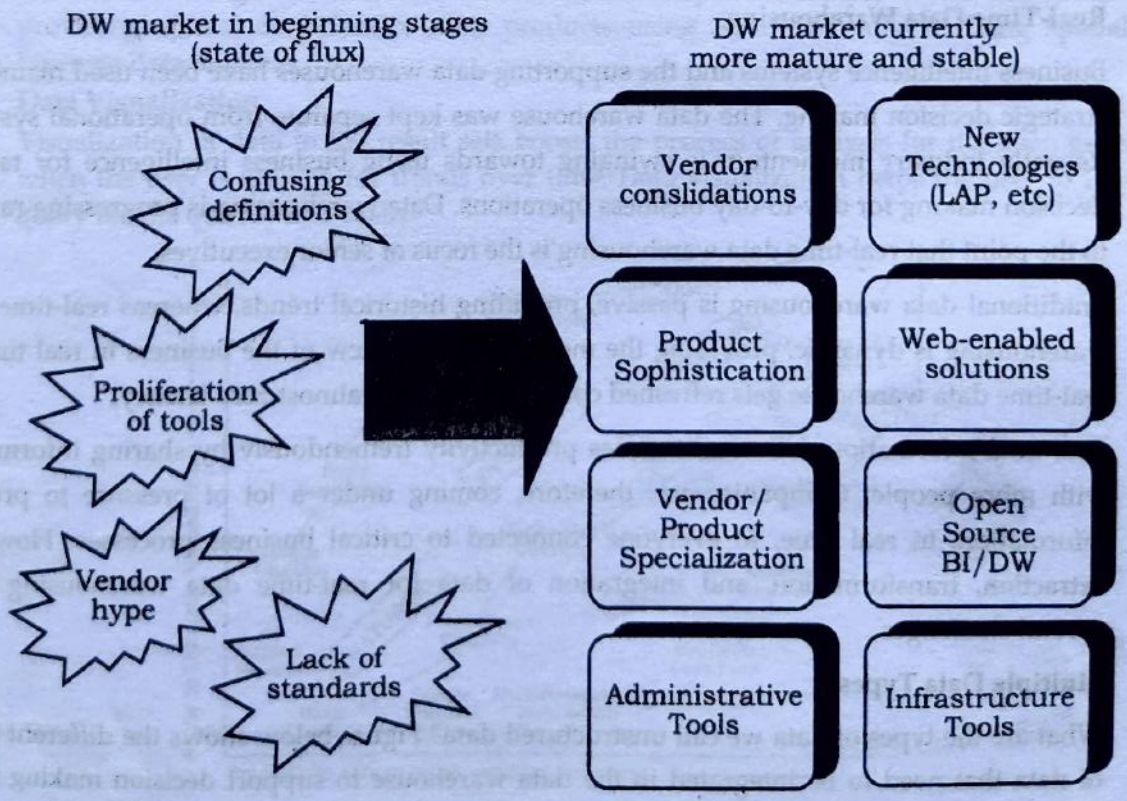

Figure 1.25: Status of the data warehousing market.

Significant Trends

Some experts feel that, until now, technology has been driving data warehousing. These experts declare that we are now beginning to see important progress in software. In the next few years, data warehousing is expected make big strides in software, especially for optimizing queries, indexing very large tables, enhancing SQL, improving data compression, and expanding dimensional modeling.

1. Real-Time Data Warehousing
2. Multiple Data Types
   - Adding Unstructured Data
   - Searching Unstructured Data
   - Spatial Data
3. Data Visualization
   - Major Visualization Trend
   - Visualization Types
   - Advanced Visualization Techniques
     - Chart Manipulation.
     - Drill Down.
     - Advanced Interaction
4. Web Enabled Data warehouse

1. Real-Time Data Warehousing

Business intelligence systems and the supporting data warehouses have been used mainly for strategic decision making. The data warehouse was kept separate from operational systems. Recently industry momentum is swinging towards using business intelligence for tactical decision making for day-to-day business operations. Data warehousing is progressing rapidly to the point that real-time data warehousing is the focus of senior executives.

Traditional data warehousing is passive, providing historical trends, whereas real-time data warehousing is dynamic, providing the most up-to-date view of the business in real time. A real-time data warehouse gets refreshed continuously, with almost zero latency.

Real-time information delivery increases productivity tremendously by sharing information with more people. Companies are, therefore, coming under a lot of pressure to provide information, in real time, to everyone connected to critical business processes. However, extraction, transformation, and integration of data for real-time data warehousing have several challenges.

2. Multiple Data Types

What are the types of data we call unstructured data? Figure below shows the different types of data that need to be integrated in the data warehouse to support decision making more effectively.

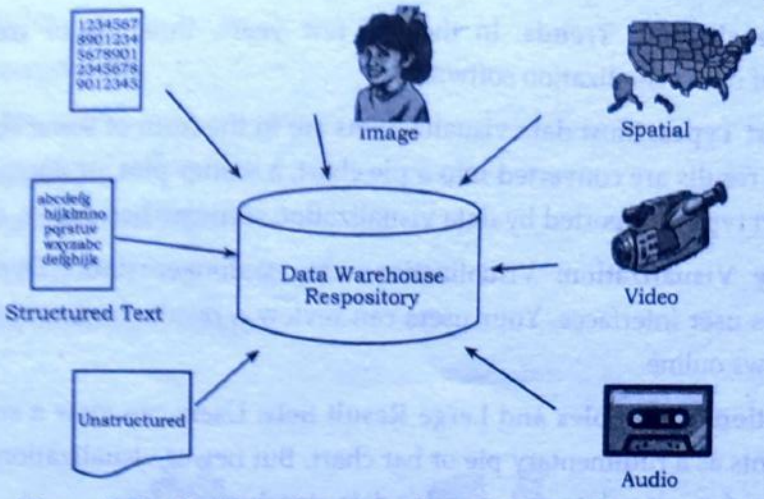

Figure 1.26: Multiple data types in a data warehouse

Adding Unstructured Data: Some vendors are addressing the inclusion of unstructured data, especially text and images, by treating such multimedia data as just another data type. These are defined as part of the relational data and stored as binary large objects (BLOBs) up to 2 GB in size. User-defined functions (UDFs) are used to define these as user defined types (UDTs).

Searching Unstructured Data: For free-form text data, retrieval engines pre index the textual documents to allow searches by words, character strings, phrases, wild cards, proximity operators, and Boolean operators. Some engines are powerful enough to substitute corresponding words and search. A search with a word mouse will also retrieve documents containing the word mice. Searching audio and video data directly is still in the research stage. Usually, these are described with free-form text, and then searched using textual search methods that are currently available.

Spatial Data: Adding spatial data will greatly enhance the value of your data warehouse. Address, street block, city quadrant, county, state, and zone are examples of spatial data. Vendors have begun to address the need to include spatial data. Some database vendors are providing spatial extenders to their products using SQL extensions to bring spatial and business data together.

3. **Data Visualization**

Visualization of data in the result sets boosts the process of analysis for the user, especially when the user is looking for trends over time. Data visualization helps the user to interpret query results quickly and easily.

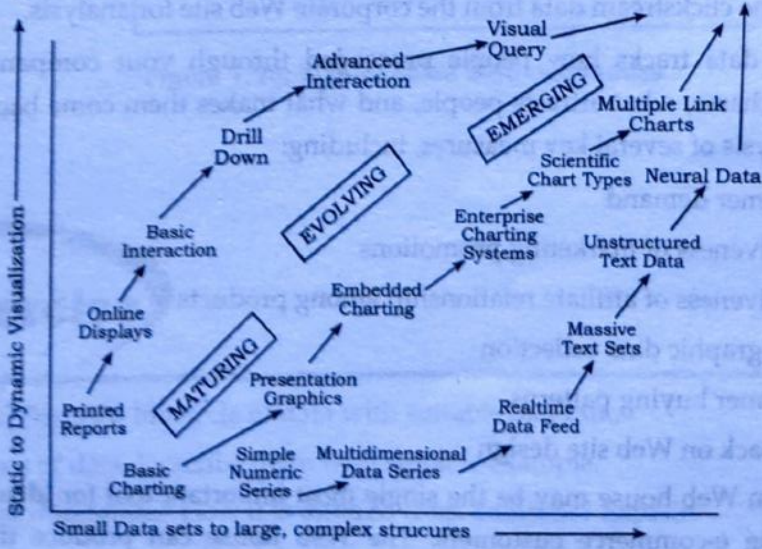

Figure 1.27: Data visualization trends.

Major Visualization Trends: In the last few years, three major trends have shaped the direction of data visualization software.

More Chart Types: Most data visualizations are in the form of some standard chart type. The numerical results are converted into a pie chart, a scatter plot, or another chart type. Now the list of chart types supported by data visualization software has grown much longer.

Interactive Visualization: Visualizations are no longer static. Dynamic chart types are themselves user interfaces. Your users can review a result chart, manipulate it, and then see newer views online.

Visualization of Complex and Large Result Sets: Users can view a simple series of numeric result points as a rudimentary pie or bar chart. But newer visualization software can visualize thousands of result points and complex data structures.

Visualization Types: Visualization software now supports a large array of chart types. Gone are the days of simple line graphs. The current needs of users vary enormously. Business users demand pie and bar charts. Technical and scientific users need scatter plots and constellation graphs. Analysts looking at spatial data need maps and other three-dimensional representations. In the last few years, major trends have shaped the direction of data visualization software.

Advanced Visualization Techniques: The most remarkable advance in visualization techniques is the transition from static charts to dynamic interactive presentations.

- **Chart Manipulation:** A user can rotate a chart or dynamically change the chart type to get a clearer view of the results. With complex visualization types such as constellation and scatter plots, a user can select data points with a mouse and then move the points around to clarify the view.
- **Drill Down:** The visualization first presents the results at the summary level. The user can then drill down the visualization to display further visualizations at subsequent levels of detail.

4. **Web Enabled Data Warehouse**

Web-enabling the data warehouse means using the Web for information delivery and integrating the clickstream data from the corporate Web site for analysis.

Clickstream data tracks how people proceeded through your company's Web site, what triggers purchases, what attracts people, and what makes them come back. Clickstream data enables analysis of several key measures, including:

- Customer demand
- Effectiveness of marketing promotions
- Effectiveness of affiliate relationship among products
- Demographic data collection
- Customer buying patterns
- Feedback on Web site design

A clickstream Web house may be the single most important tool for identifying, prioritizing, and retaining e-commerce customers. The Web house can produce the following useful information:

- Site statistics
- Visitor conversions
- Ad metrics
- Referring partner links
- Site navigation resulting in orders
- Site navigation not resulting in orders
- Pages that are session killers

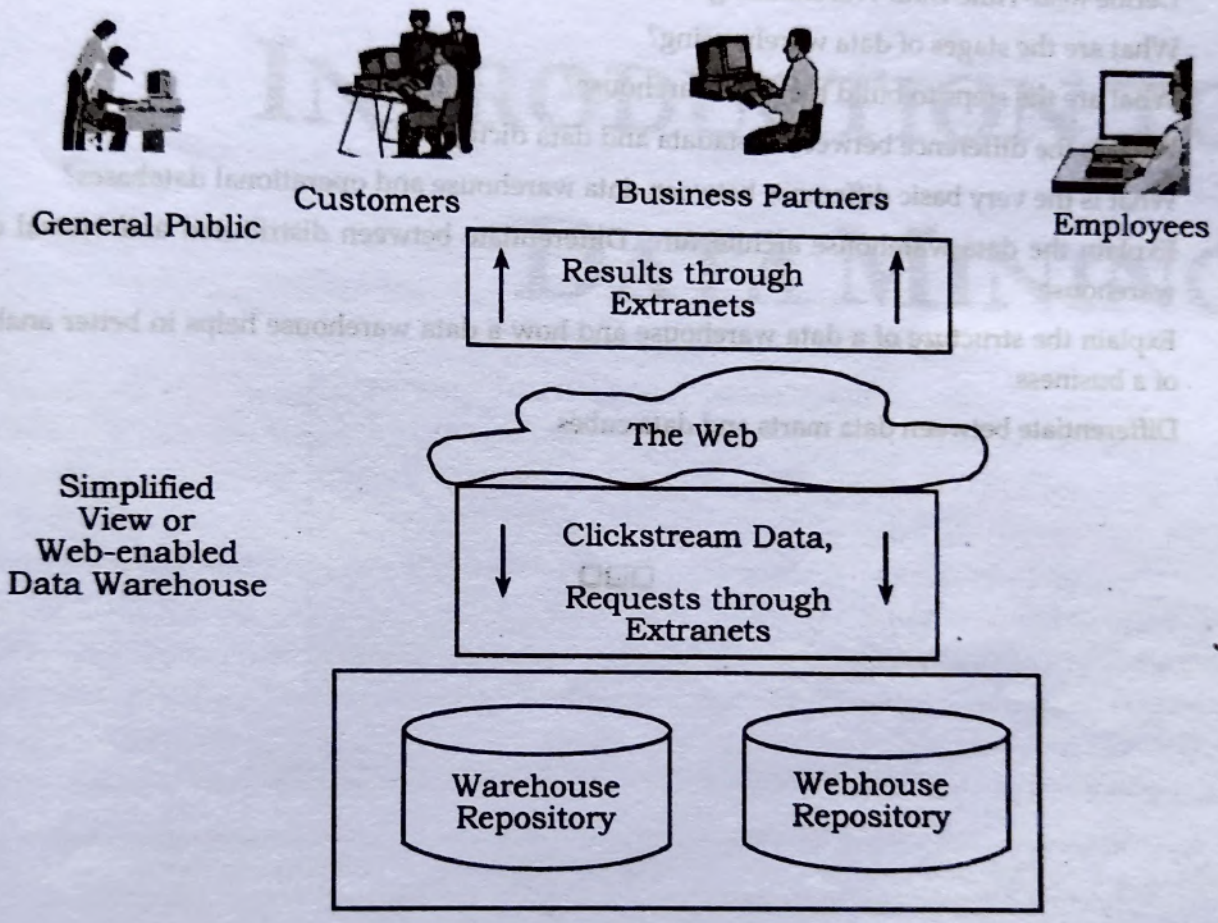

Figure 1.28: Web-enabled data warehouse

Exercise

1. Define data. Describe life cycle of data with suitable diagram.
2. List out types of data. Describe them with suitable example.
3. What is data warehouse? How it is differed from database? Explain
4. Differences between operational database and data warehouse.

5. Define multi-dimensional data model. Explain their uses.
6. Describe OLAP operation in multidimensional data model.
7. What is data warehousing? Describe architecture of data warehouse.
8. What do you mean by conceptual modeling of data warehouse? Explain
9. How to implement data warehouse? Explain. Describe components of data warehouse.
10. What is data mart? How it is differed from data warehouse? Explain
11. Describe needs of data warehousing. Describe trends in data warehousing.
12. Define metadata. How it is differed from database? Explain
13. Define Real-Time Data Warehousing with suitable example.
14. What are the stages of data warehousing?
15. What are the steps to build the data warehouse?
16. What is the difference between metadata and data dictionary?
17. What is the very basic difference between data warehouse and operational databases?
18. Explain the data warehouse architecture. Differentiate between distributed and virtual data warehouse
19. Explain the structure of a data warehouse and how a data warehouse helps in better analysis of a business.
20. Differentiate between data marts and data cubes.

❑❑❑