UNIT - 1
DATA WAREHOUSING & OLAP

PREPARED BY S NOORTAJ

Data warehouse

In 1980s William Inmon defines data warehouse to be a set of data that supports decision support system[DSS].

I. Data warehouse market supports such diverse industries as manufacturing, retail ,telecommunications and health care.
II. A personal database contains information about the current set of employees is sufficient information can be add and substract in data bases.
III. The data must and should be cleaned , refirmatted, integrated and sunmarised before being place in the warehouse.
IV. Warehouase is stored as a dtaabase , it can access by traaditional querry language .
V. SQL is a high level querry language , so it is used for reques the data in datawarehouse.
VI. Changes can be occure in database but not in datawarehouse.

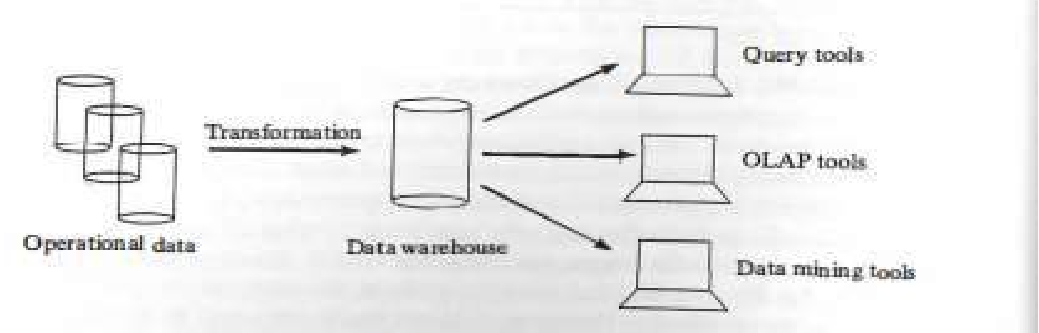


FIGURE 2.14: Data warehouse.

Several ways to improve performance of the data warehouse applications.
1. Summarization
2. De-Normalization
3. Partitioning

Data warehouse collaborate data from several sources and ensure data accuracy, quality, and consistency. System execution is boosted by differentiating the process of analytics from traditional databases. In a data warehouse, data is sorted into a formatted pattern by type and as needed. The data is examined by query tools using several patterns.

FEATURES OF DATA WAREHOUSES:

Subject Oriented: It provides you with important data about a specific subject like suppliers, products, promotion, customers, etc. Data warehousing usually handles the analysis and modeling of data that assist any organization to make data-driven decisions.

Integrated: Different heterogeneous sources are put together to build a data warehouse, such as level documents or social databases.

Time-Variant: The data collected in a data warehouse is identified with a specific period.

Nonvolatile: This means the earlier data is not deleted when new data is added to the data warehouse. The operational database and data warehouse are kept separate and thus continuous changes in the operational database are not shown in the data warehouse.

APPLICATIONS OF DATA WAREHOUSES:

Data warehouses help analysts or senior executives analyze, organize, and use data for decision making.It is used in the following fields:

Consumer goods
Banking services
Financial services
Manufacturing
Retail sectors

ADVANTAGES OF DATA WAREHOUSING:

Cost-efficient and provides quality of data.

Performance and productivity are improved.

Accurate data access and consistency.

Data Warehouse Modeling

Data warehouse modeling is the process of designing the schemas of the detailed and summarized information of the data warehouse. The goal of data warehouse modeling is to develop a schema describing the reality, or at least a part of the fact, which the data warehouse is needed to support.

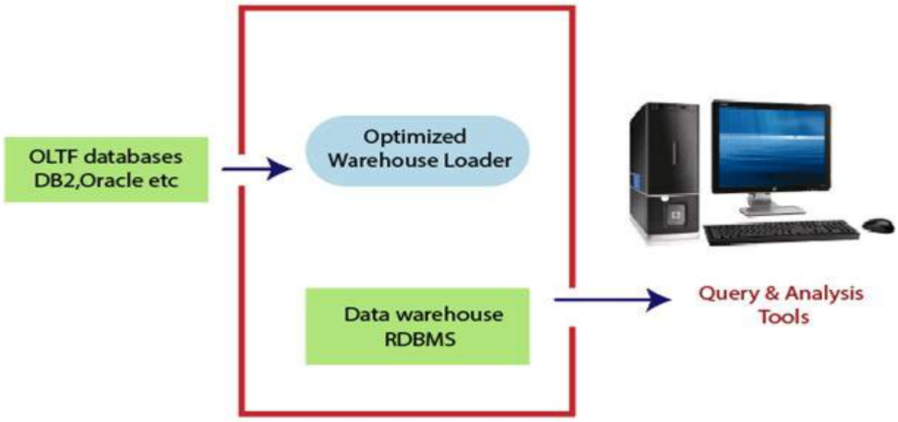
Data warehouse modeling is an essential stage of building a data warehouse for two main reasons.

1. Firstly, through the schema, data warehouse clients can visualize the relationships among the warehouse data, to use them with greater ease.
2. Secondly, a well-designed schema allows an effective data warehouse structure to emerge, to help decrease the cost of implementing the warehouse and improve the efficiency of using it.

Data modeling in data warehouses is different from data modeling in operational database systems. The primary function of data warehouses is to support DSS processes. Thus, the objective of data warehouse modeling is to make the data warehouse efficiently support complex queries on long term information.

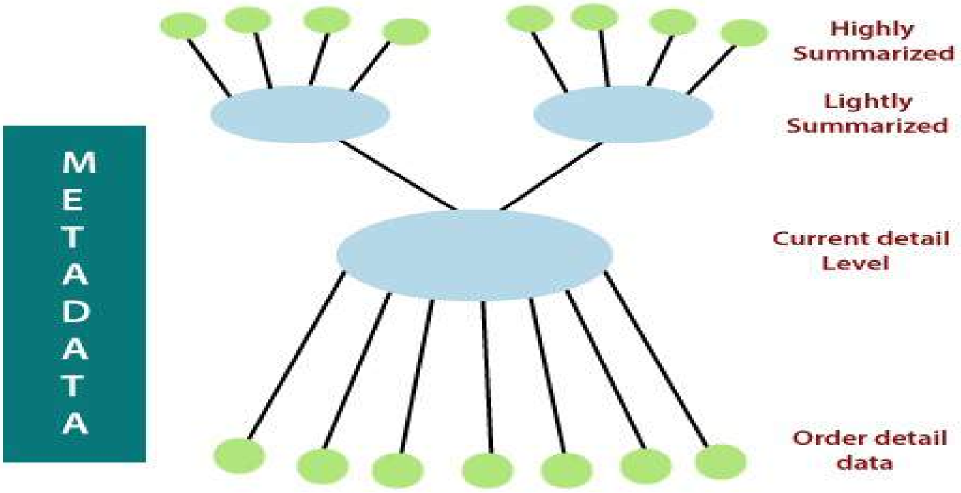
In contrast, data modeling in operational database systems targets efficiently supporting simple transactions in the database such as retrieving, inserting, deleting, and changing data. Moreover, data warehouses are designed for the customer with general information knowledge about the enterprise, whereas operational database systems are more oriented toward use by software specialists for creating distinct applications.

Data Warehouse model is illustrated in the given diagram.



Data Warehouse Model

The data within the specific warehouse itself has a particular architecture with the emphasis on various levels of summarization, as shown in figure:



The Structure of data inside the data warehouse

The current detail record is central in importance as it:

- Reflects the most current happenings, which are commonly the most stimulating.
- It is numerous as it is saved at the lowest method of the Granularity.
- It is always (almost) saved on disk storage, which is fast to access but expensive and difficult to manage.

Older detail data is stored in some form of mass storage, and it is infrequently accessed and kept at a level detail consistent with current detailed data.

Lightly summarized data is data extract from the low level of detail found at the current, detailed level and usually is stored on disk storage. When building the data warehouse have to remember what unit of time is summarization done over and also the components or what attributes the summarized data will contain.

Highly summarized data is compact and directly available and can even be found outside the warehouse.

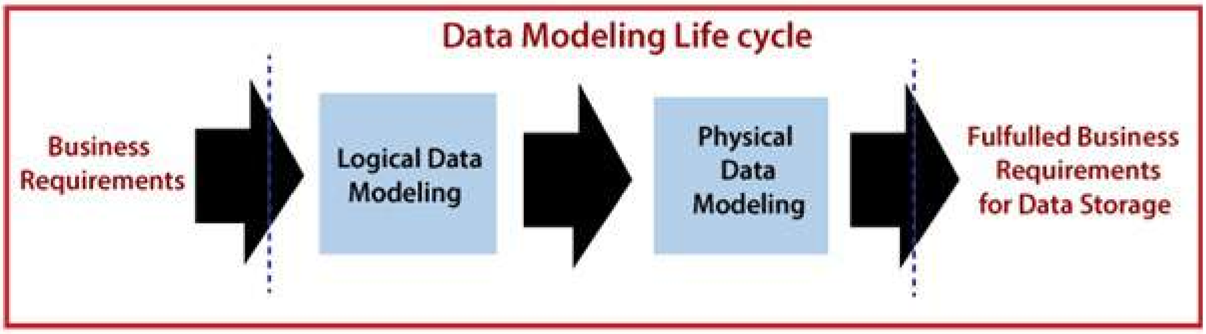
Metadata is the final element of the data warehouses and is really of various dimensions in which it is not the same as file drawn from the operational data, but it is used as:-

- A directory to help the DSS investigator locate the items of the data warehouse.
- A guide to the mapping of record as the data is changed from the operational data to the data warehouse environment.
- A guide to the method used for summarization between the current, accurate data and the lightly summarized information and the highly summarized data, etc.

Data Modeling Life Cycle

It is a straight forward process of transforming the business requirements to fulfill the goals for storing, maintaining, and accessing the data within IT systems. The result is a logical and physical data model for an enterprise data warehouse.

The objective of the data modeling life cycle is primarily the creation of a storage area for business information. That area comes from the logical and physical data modeling stages, as shown in Figure:



A generic data modeling life cycle

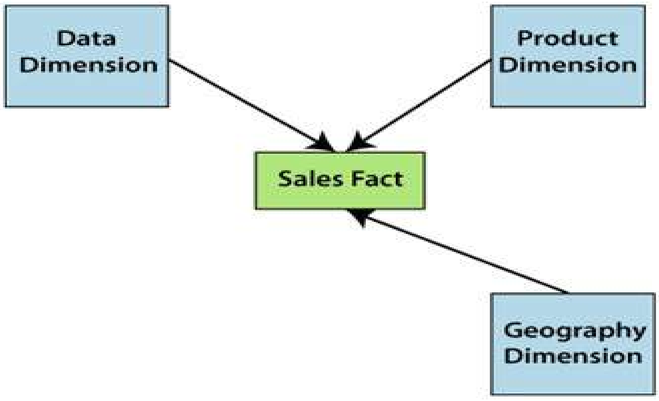
Conceptual Data Model

A conceptual data model recognizes **the highest-level relationships between the different entities.**

Characteristics of the conceptual data model

- It contains the essential entities and the relationships among them.
- No attribute is specified.
- No primary key is specified.

We can see that the only data shown via the conceptual data model is the entities that define the data and the relationships between those entities. No other data, as shown through the conceptual data model.



Example of Conceptual Data Model

Logical Data Model

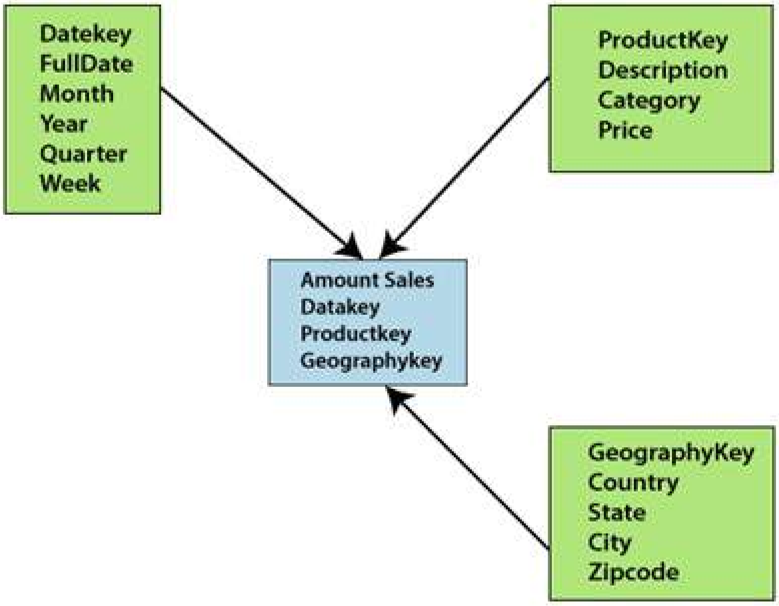
A logical data model defines the information in as much **structure** as possible, without observing how they will be physically achieved in the database. The primary objective of logical data modeling is to document the business data structures, processes, rules, and relationships by a single view - the logical data model.

Features of a logical data model

- It involves all entities and relationships among them.
- All attributes for each entity are specified.
- The primary key for each entity is stated.
- Referential Integrity is specified (FK Relation).

The phase for designing the logical data model which are as follows:

- Specify primary keys for all entities.
- List the relationships between different entities.
- List all attributes for each entity.
- Normalization.
- No data types are listed



Example of Logical Data Model

Physical Data Model

Physical data model describes **how the model will be presented in the database.** A physical database model demonstrates all table structures, column names, data types, constraints, primary key, foreign key, and relationships between tables. The purpose of physical data modeling is the mapping of the logical data model to the physical structures of the RDBMS system hosting the data warehouse. This contains defining physical RDBMS structures, such as tables and data types to use when storing the information. It may also include the definition of new data structures for enhancing query performance.

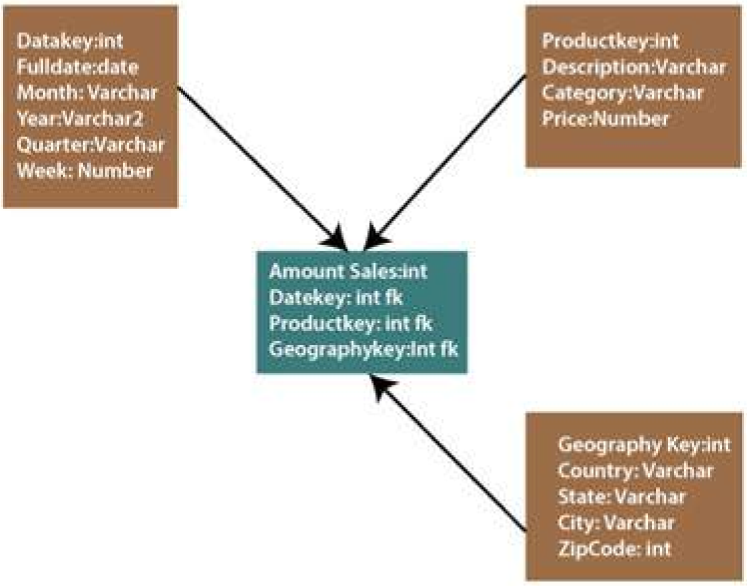
Characteristics of a physical data model

- Specification all tables and columns.
- Foreign keys are used to recognize relationships between tables.

The steps for physical data model design which are as follows:

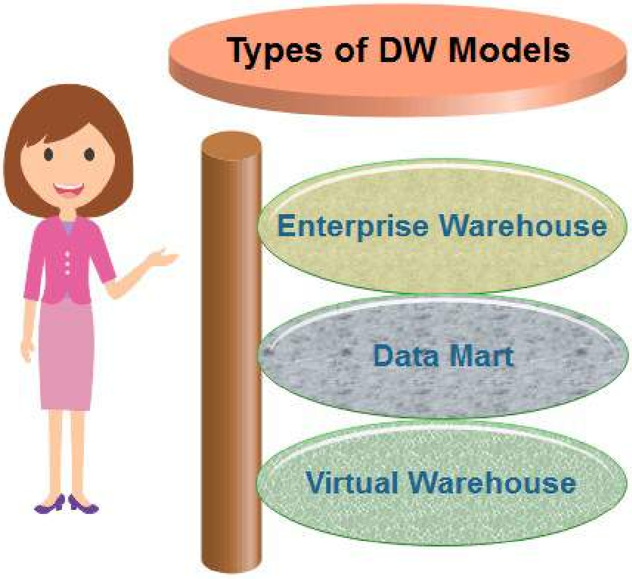
- Convert entities to tables.
- Convert relationships to foreign keys.

- Convert attributes to columns.



Example of Physical Data Model

Types of Data Warehouse Models:



Enterprise Warehouse

An Enterprise warehouse collects all of the records about subjects spanning the entire organization. It supports corporate-wide data integration, usually from one or more operational systems or external data providers, and it's cross-functional in scope. It generally contains detailed information as well as summarized information and can range in estimate from a few gigabyte to hundreds of gigabytes, terabytes, or beyond.

An enterprise data warehouse may be accomplished on traditional mainframes, UNIX super servers, or parallel architecture platforms. It required extensive business modeling and may take years to develop and build.

Data Mart

A data mart includes a subset of corporate-wide data that is of value to a specific collection of users. The scope is confined to particular selected subjects. For example, a marketing data mart may restrict its subjects to the customer, items, and sales. The data contained in the data marts tend to be summarized.

Data Marts is divided into two parts:

Independent Data Mart: Independent data mart is sourced from data captured from one or more operational systems or external data providers, or data generally locally within a different department or geographic area.

Dependent Data Mart: Dependent data marts are sourced exactly from enterprise data-warehouses.

Virtual Warehouses

Virtual Data Warehouses is a set of perception over the operational database. For effective query processing, only some of the possible summary vision may be materialized. A virtual warehouse is simple to build but required excess capacity on operational database servers.

Data Warehouse Design

A data warehouse is a single data repository where a record from multiple data sources is integrated for online business analytical processing (OLAP). This implies a data warehouse needs to meet the requirements from all the business stages within the entire organization. Thus, data warehouse design is a hugely complex, lengthy, and hence error-prone process. Furthermore, business analytical functions change over time, which results in changes in the requirements for the systems. Therefore, data warehouse and OLAP systems are dynamic, and the design process is continuous.

Data warehouse design takes a method different from view materialization in the industries. It sees data warehouses as database systems with particular needs such as answering management related queries. The target of the design becomes how the record from multiple data sources should be extracted, transformed, and loaded (ETL) to be organized in a database as the data warehouse.

There are two approaches

1. "top-down" approach
2. "bottom-up" approach

Top-down Design Approach

In the "Top-Down" design approach, a data warehouse is described as a subject-oriented, time-variant, non-volatile and integrated data repository for the entire enterprise data from different sources are validated, reformatted and saved in a normalized (up to 3NF) database as the data warehouse. The data warehouse stores "atomic" information, the data at the lowest level of granularity, from where dimensional data marts can be built by selecting the data required for specific business subjects or particular departments. An approach is a data-driven approach as the information is gathered and integrated first and then business requirements by subjects for building data marts are formulated. The advantage of this method is which it supports a single integrated data source. Thus data marts built from it will have consistency when they overlap.

Advantages of top-down design

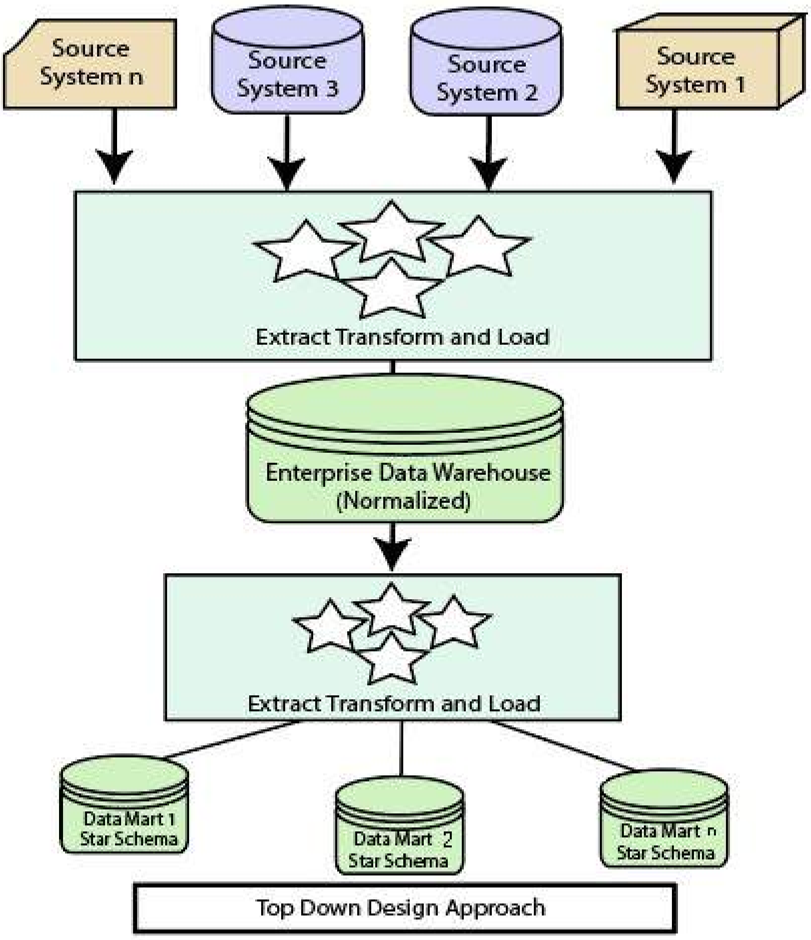
Data Marts are loaded from the data warehouses.

Developing new data mart from the data warehouse is very easy.

Disadvantages of top-down design

This technique is inflexible to changing departmental needs.

The cost of implementing the project is high.



Top Down Design Approach

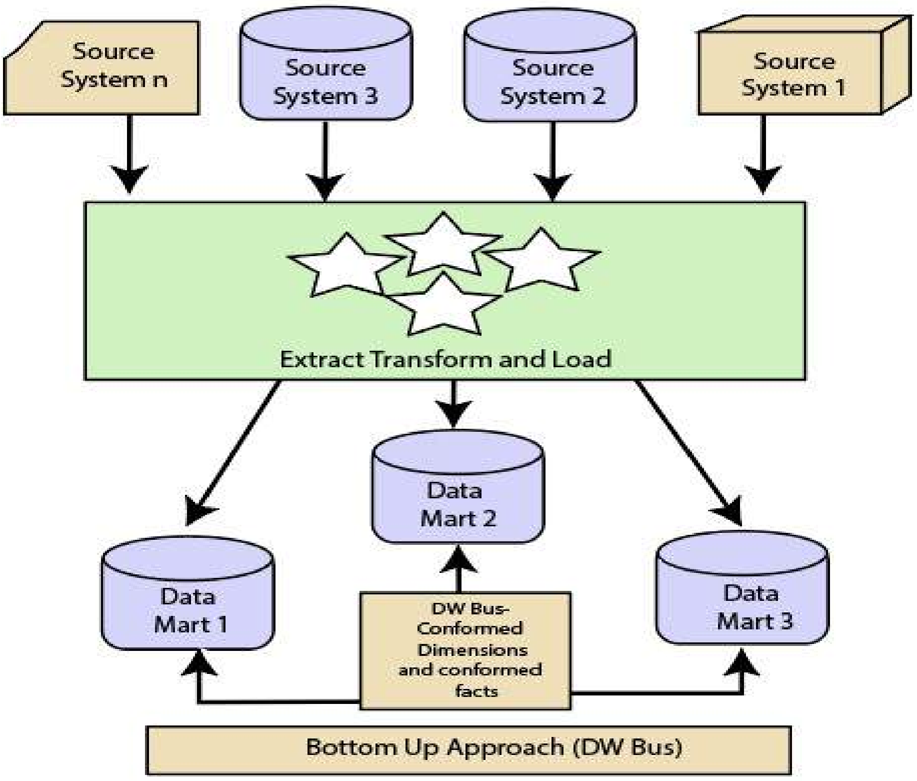
Bottom-Up Design Approach

In the "Bottom-Up" approach, a data warehouse is described as "a copy of transaction data specifical architecture for query and analysis," term the star schema. In this approach, a data mart is created first to necessary reporting and analytical capabilities for particular business processes (or subjects). Thus it is needed to be a business-driven approach in contrast to Inmon's data-driven approach.

Data marts include the lowest grain data and, if needed, aggregated data too. Instead of a normalized database for the data warehouse, a denormalized dimensional database is adapted to meet the data delivery requirements of data warehouses. Using this method, to use the set of data marts as the enterprise data warehouse, data marts should be built with conformed dimensions in mind, defining that ordinary objects are represented the same in different data marts.

The conformed dimensions connected the data marts to form a data warehouse, which is generally called a virtual data warehouse.

The advantage of the "bottom-up" design approach is that it has quick ROI, as developing a data mart, a data warehouse for a single subject, takes far less time and effort than developing an enterprise-wide data warehouse. Also, the risk of failure is even less. This method is inherently incremental. This method allows the project team to learn and grow.


Source System n
Source System 3
Source System 2
Source System 1
Extract Transform and Load
Data Mart 2
Data Mart 1
Data Mart 3
DW Bus- Conformed Dimensions and conformed facts
Bottom Up Approach (DW Bus)

Bottom Up Design Approach

Advantages of bottom-up design

Documents can be generated quickly.

The data warehouse can be extended to accommodate new business units.

It is just developing new data marts and then integrating with other data marts.

Disadvantages of bottom-up design

the locations of the data warehouse and the data marts are reversed in the bottom-up approach design.

Differentiate between Top-Down Design Approach and Bottom-Up Design Approach

Top-Down Design Approach	**Bottom-Up Design Approach**
Breaks the vast problem into smaller sub problems.	Solves the essential low-level problem and integrates them into a higher one.
Inherently architected- not a union of several data marts.	Inherently incremental; can schedule essential data marts first.
Single, central storage of information about the content.	Departmental information stored.
Centralized rules and control.	Departmental rules and control.
It includes redundant information.	Redundancy can be removed.
It may see quick results if implemented with repetitions.	Less risk of failure, favorable return on investment, and proof of techniques.

What is Dimensional Modeling?

Dimensional modeling represents data with a cube operation, making more suitable logical data representation with OLAP data management. The perception of Dimensional Modeling was developed by **Ralph Kimball** and is consist of **"fact"** and **"dimension"** tables.

In dimensional modeling, the transaction record is divided into either **"facts,"** which are frequently numerical transaction data, or **"dimensions,"** which are the reference information that gives context to the facts. For example, a sale transaction can be damage into facts such as the number of products ordered and the price paid for the products, and into dimensions such as order date, user name, product number, order ship-to, and bill-to locations, and salesman responsible for receiving the order.

Objectives of Dimensional Modeling:

The purposes of dimensional modeling are:

1. To produce database architecture that is easy for end-clients to understand and write queries.
2. To maximize the efficiency of queries. It achieves these goals by minimizing the number of tables and relationships between them.

Advantages of Dimensional Modeling

Following are the benefits of dimensional modeling are:

Dimensional modeling is simple: Dimensional modeling methods make it possible for warehouse designers to create database schemas that business customers can easily hold and comprehend. There is no need for vast training on how to read diagrams, and there is no complicated relationship between different data elements.

Dimensional modeling promotes data quality: The star schema enable warehouse administrators to enforce referential integrity checks on the data warehouse. Since the fact information key is a concatenation of the essentials of its associated dimensions, a factual record is actively loaded if the corresponding dimensions records are duly described and also exist in the database.

By enforcing foreign key constraints as a form of referential integrity check, data warehouse DBAs add a line of defense against corrupted warehouses data.

Performance optimization is possible through aggregates: As the size of the data warehouse increases, performance optimization develops into a pressing concern. Customers who have to wait for hours to get a response to a query will quickly become discouraged with the warehouses. Aggregates are one of the easiest methods by which query performance can be optimized.

Disadvantages of Dimensional Modeling

1. To maintain the integrity of fact and dimensions, loading the data warehouses with a record from various operational systems is complicated.
2. It is severe to modify the data warehouse operation if the organization adopting the dimensional technique changes the method in which it does business.

Elements of Dimensional Modeling

Fact: It is a collection of associated data items, consisting of measures and context data. It typically represents business items or business transactions.

Dimensions: It is a collection of data which describe one business dimension. Dimensions decide the contextual background for the facts, and they are the framework over which OLAP is performed.

Measure: It is a numeric attribute of a fact, representing the performance or behavior of the business relative to the dimensions.

Considering the relational context, there are three basic models which are used in dimensional modeling:

- Star Model
- Snowflake Model
- Fact Constellation\Galaxy schema

The star model is the underlying structure for a dimensional model. It has one broad central table (fact table) and a set of smaller tables (dimensions) arranged in a radial design around the primary table. The snowflake model is the conclusion of decomposing one or more of the dimensions.fact constellation model is the

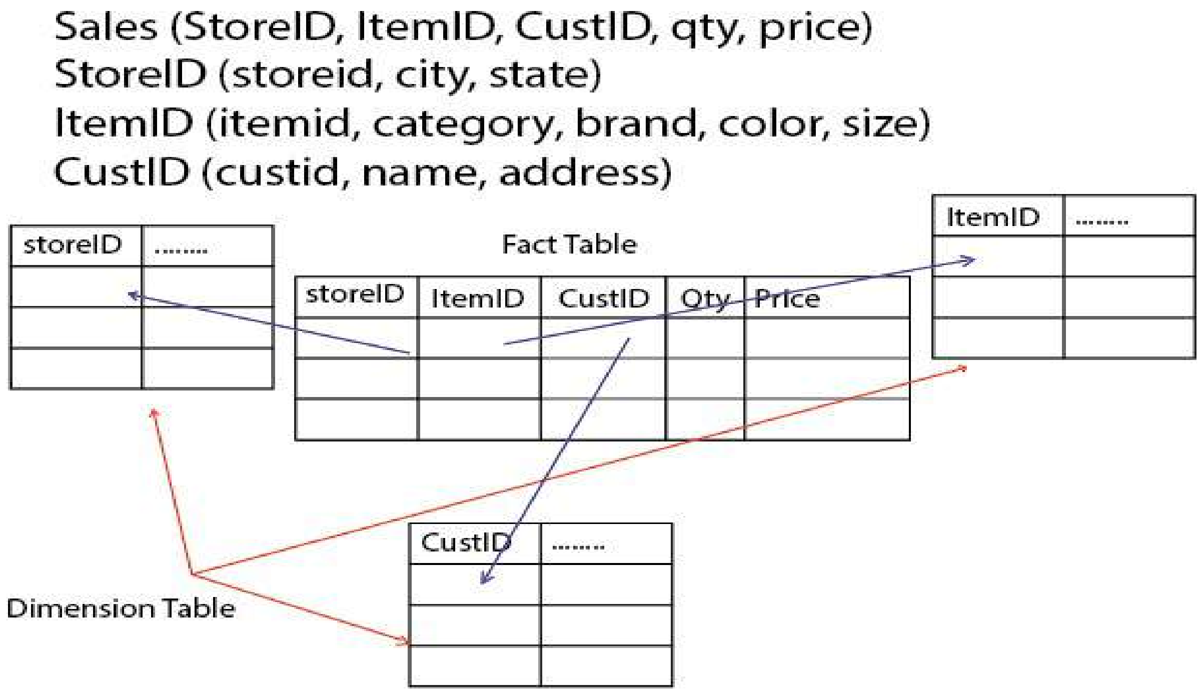
Fact Table

Fact tables are used to data facts or measures in the business. Facts are the numeric data elements that are of interest to the company.

Dimension Table

Dimension tables establish the context of the facts. Dimensional tables store fields that describe the facts.

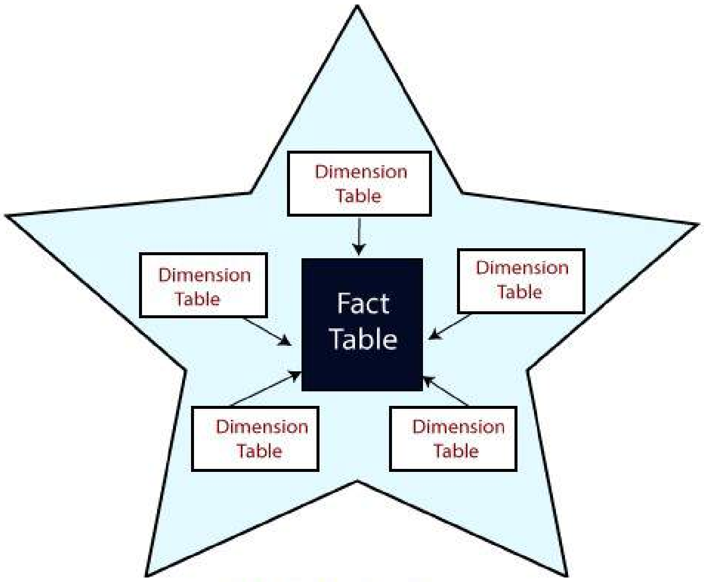
Example: A city and state can view a store summary in a fact table. Item summary can be viewed by brand, color, etc. Customer information can be viewed by name and address.



Sales (StoreID, ItemID, CustID, qty, price)
StoreID (storeid, city, state)
ItemID (itemid, category, brand, color, size)
CustID (custid, name, address)

What is Star Schema?

A star schema is the elementary form of a dimensional model, in which data are organized into **facts** and **dimensions**. A fact is an event that is counted or measured, such as a sale or log in. A dimension includes reference data about the fact, such as date, item, or customer.



Star Schema

Fact Tables

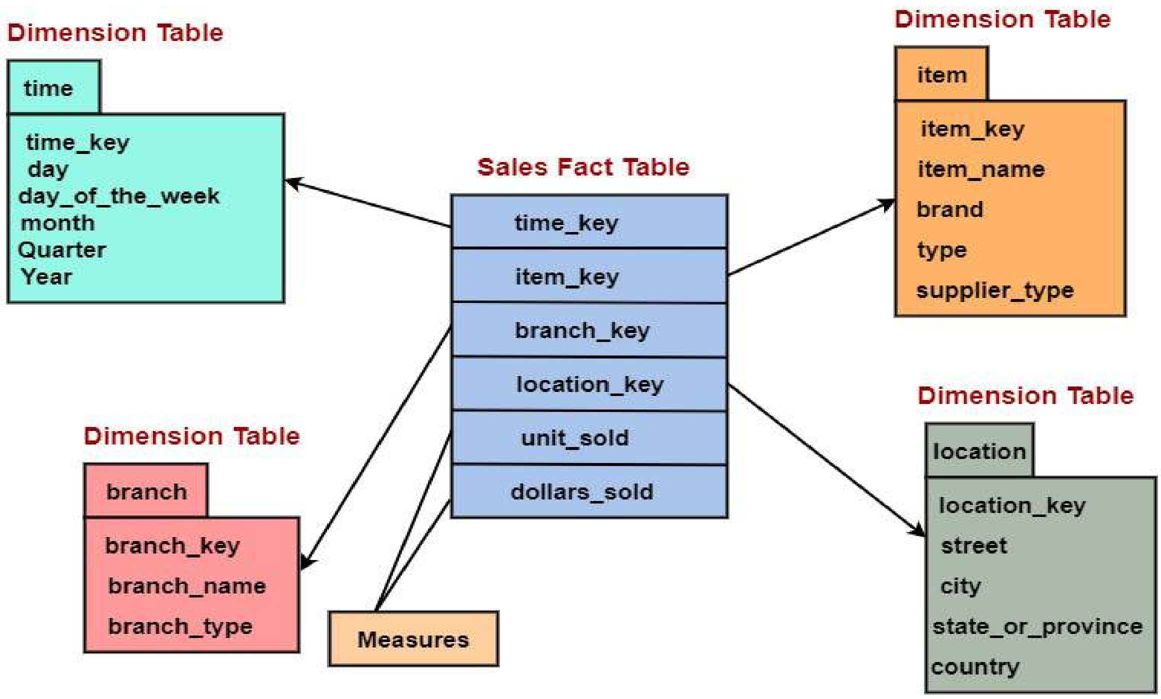
A table in a star schema which contains facts and connected to dimensions. A fact table has two types of columns: those that include fact and those that are foreign keys to the dimension table. The primary key of the fact tables is generally a composite key that is made up of all of its foreign keys.

A fact table might involve either detail level fact or fact that have been aggregated (fact tables that include aggregated fact are often instead called summary tables). A fact table generally contains facts with the same level of aggregation.

Dimension Tables

A dimension is an architecture usually composed of one or more hierarchies that categorize data. If a dimension has not got hierarchies and levels, it is called a **flat dimension** or **list**. The primary keys of each of the dimensions table are part of the composite primary keys of the fact table. Dimensional attributes help to define the dimensional value. They are generally descriptive, textual values. Dimensional tables are usually small in size than fact table.

Fact tables store data about sales while dimension tables data about the geographic region (markets, cities), clients, products, times, channels.



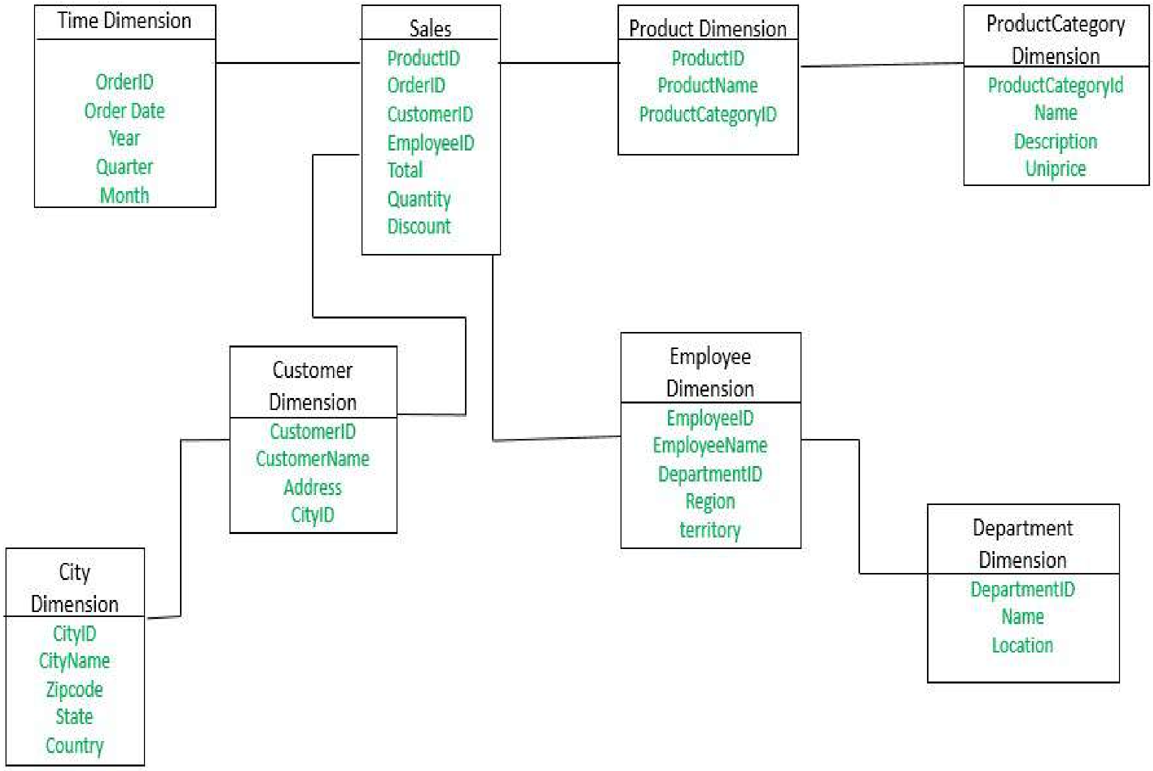
In the above example, the SALES table contains only four columns with IDs from the dimension tables, TIME, ITEM, BRANCH, and LOCATION, instead of four columns for time data, four columns for ITEM data, three columns for BRANCH data, and four columns for LOCATION data. Thus, the size of the fact table is significantly reduced. When we need to change an item, we need only make a single change in the dimension table, instead of making many changes in the fact table.

We can create even more complex star schemas by normalizing a dimension table into several tables. The normalized dimension table is called a **Snowflake**.

Snowflake Schema

The snowflake schema is a variant of the star schema. Here, the centralized fact table is connected to multiple dimensions. In the snowflake schema, dimensions are present in a normalized form in multiple related tables. The snowflake structure materialized when the dimensions of a star schema are detailed and highly structured, having several levels of relationship.

Example:

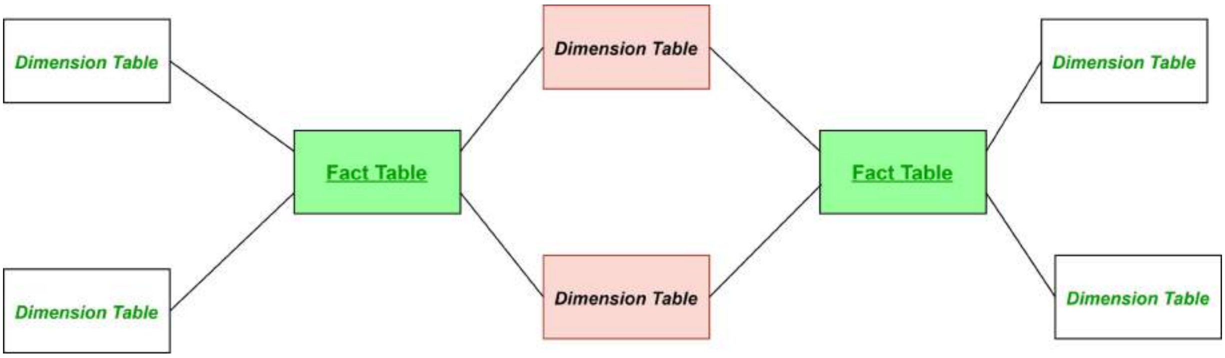


The **Employee** dimension table now contains the attributes: EmployeeID, EmployeeName, DepartmentID, Region, Territory. The DepartmentID attribute links with the **Employee** table with the **Department** dimension table. The **Department** dimension is used to provide detail about each department, such as the Name and Location of the department. The **Customer** dimension table now contains the attributes: CustomerID, CustomerName, Address, CityID. The CityID attributes link the **Customer** dimension table with the **City** dimension table. The **City** dimension table has details about each city such as CityName, Zipcode, State, and Country.

The main difference between star schema and snowflake schema is that the dimension table of the snowflake schema is maintained in the normalized form to reduce redundancy. The advantage here is that such tables (normalized) are easy to maintain and save storage space.

Fact Constellation\Galaxy schema

The fact constellation is a schema for representing multidimensional model. It is a collection of multiple fact tables having some common dimension tables. It can be viewed as a collection of several star schemas and hence, also known as Galaxy schema. It is one of the widely used schema for Data warehouse designing and it is much more complex than star and snowflake schema. For complex systems, we require fact constellations.

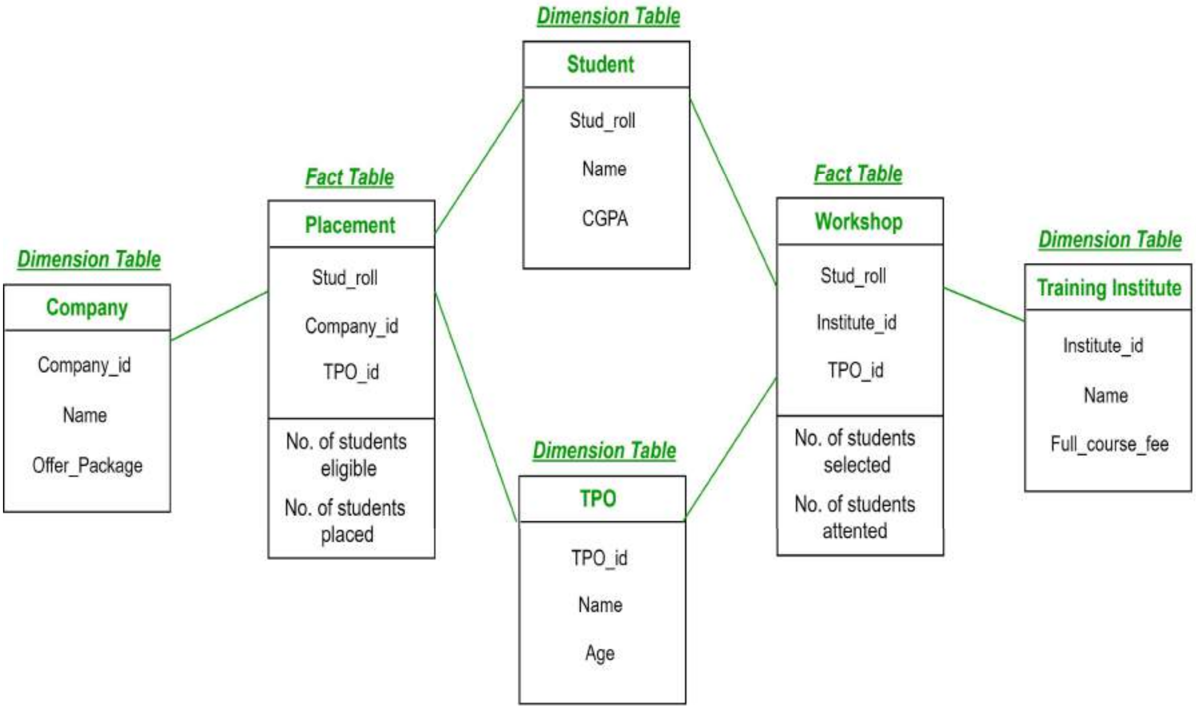


Fact Constellation in Data Warehouse modelling:

Figure – General structure of Fact Constellation

Here, the pink coloured Dimension tables are the common ones among both the star schemas. Green coloured fact tables are the fact tables of their respective star schemas.

Example:



Placement is a fact table having attributes: (Stud_roll, Company_id, TPO_id) with facts: (Number of students eligible, Number of students placed).
Workshop is a fact table having attributes: (Stud_roll, Institute_id, TPO_id) with facts: (Number of students selected, Number of students attended the workshop).
Company is a dimension table having attributes: (Company_id, Name, Offer_package).
Student is a dimension table having attributes: (Student_roll, Name, CGPA).
TPO is a dimension table having attributes: (TPO_id, Name, Age).
Training Institute is a dimension table having attributes: (Institute_id, Name, Full_course_fee).

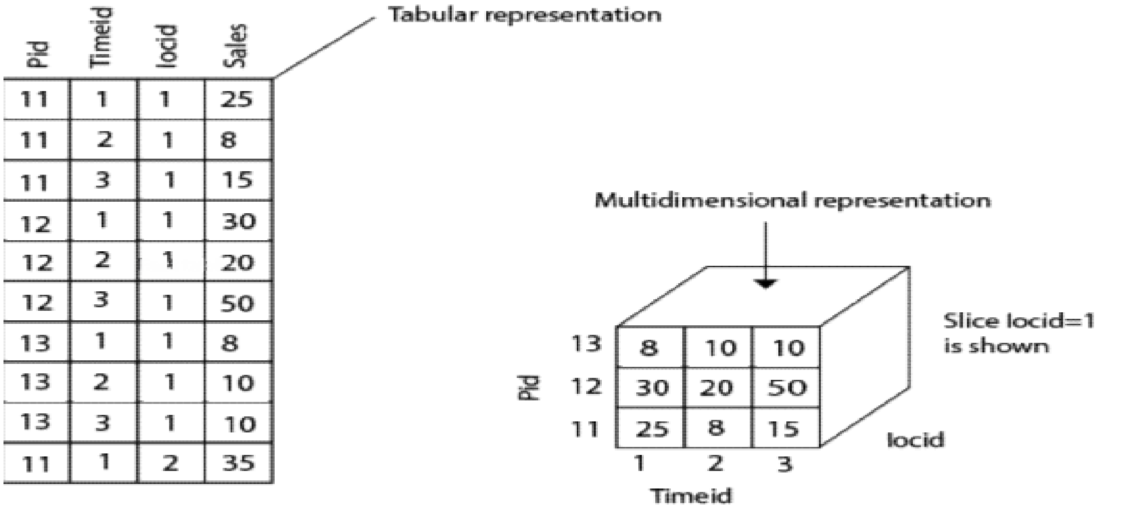
Advantage: Provides a flexible schema.
Disadvantage: It is much more complex and hence, hard to implement and maintain.

What is Multi-Dimensional Data Model?

A multidimensional model views data in the form of a data-cube. A data cube enables data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.

The dimensions are the perspectives or entities concerning which an organization keeps records. For example, a shop may create a sales data warehouse to keep records of the store's sales for the dimension time, item, and location. These dimensions allow the save to keep track of things, for example, monthly sales of items and the locations at which the items were sold. Each dimension has a table related to it, called a dimensional table, which describes the dimension further. For example, a dimensional table for an item may contain the attributes item_name, brand, and type.

A multidimensional data model is organized around a central theme, for example, sales. This theme is represented by a fact table. Facts are numerical measures. The fact table contains the names of the facts or measures of the related dimensional tables.



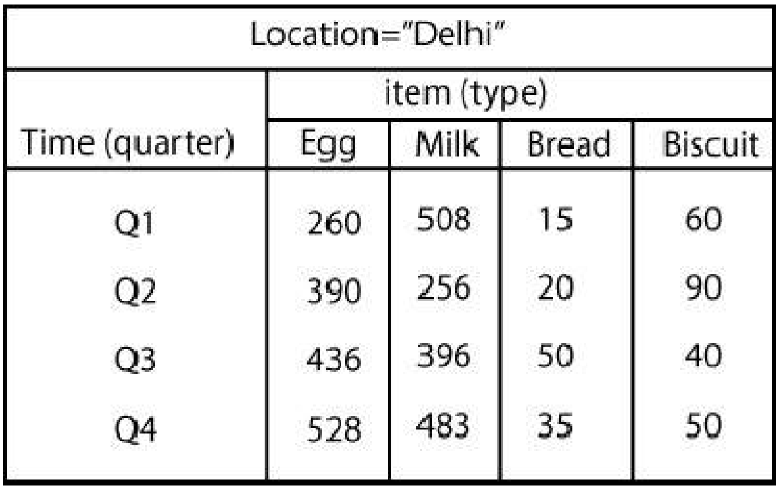
Tabular representation

Pid	Timeid	locid	Sales
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35

Multidimensional representation

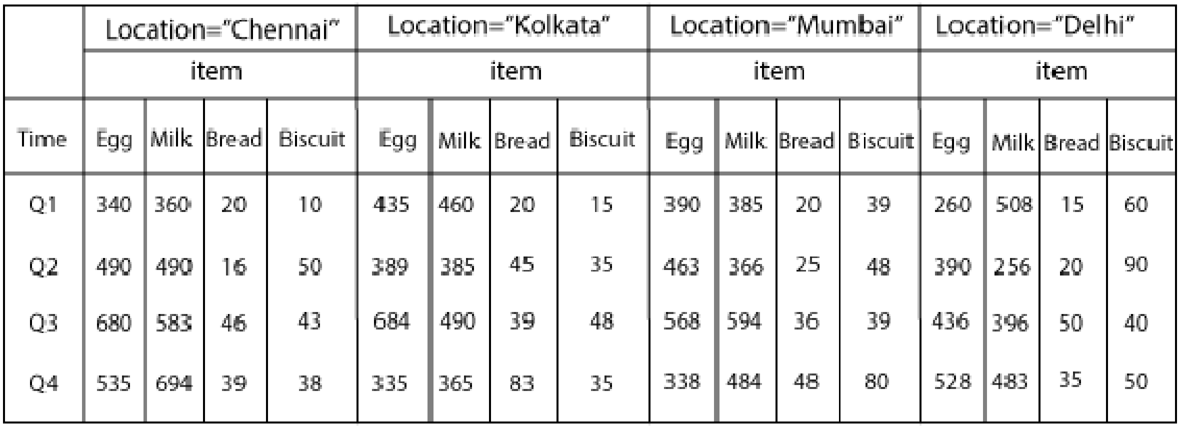
Pid	1	2	3
13	8	10	10
12	30	20	50
11	25	8	15

Consider the data of a shop for items sold per quarter in the city of Delhi. The data is shown in the table. In this 2D representation, the sales for Delhi are shown for the time dimension (organized in quarters) and the item dimension (classified according to the types of an item sold). The fact or measure displayed in rupee_sold (in thousands).



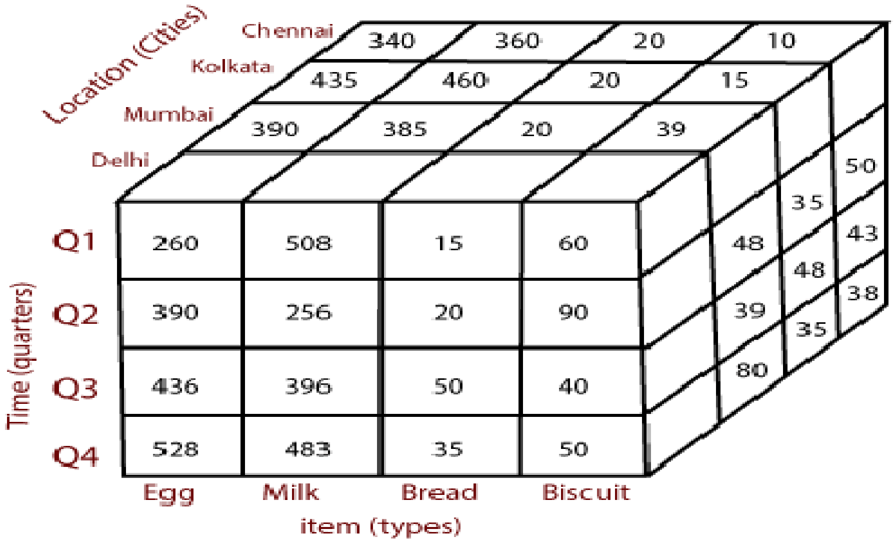
Location="Delhi"				
	item (type)			
Time (quarter)	Egg	Milk	Bread	Biscuit
Q1	260	508	15	60
Q2	390	256	20	90
Q3	436	396	50	40
Q4	528	483	35	50

Now, if we want to view the sales data with a third dimension, For example, suppose the data according to time and item, as well as the location is considered for the cities Chennai, Kolkata, Mumbai, and Delhi. These 3D data are shown in the table. The 3D data of the table are represented as a series of 2D tables.



	Location="Chennai"				Location="Kolkata"				Location="Mumbai"				Location="Delhi"			
	item				item				item				item			
Time	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit
Q1	340	360	20	10	435	460	20	15	390	385	20	39	260	508	15	60
Q2	490	490	16	50	389	385	45	35	463	366	25	48	390	256	20	90
Q3	680	583	46	43	684	490	39	48	568	594	36	39	436	396	50	40
Q4	535	694	39	38	335	365	83	35	338	484	48	80	528	483	35	50

Conceptually, it may also be represented by the same data in the form of a 3D data cube, as shown in fig:



What is Data Cube?

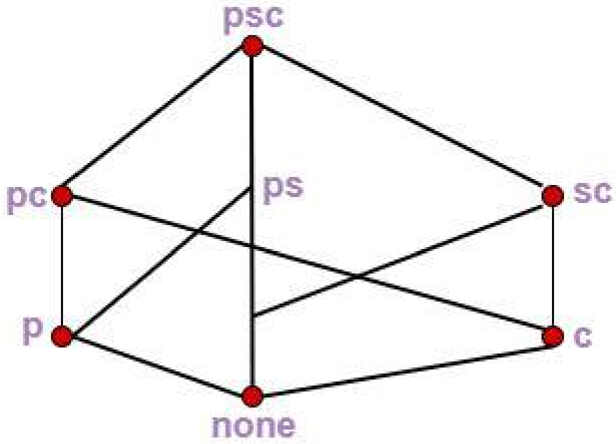
data cube is a **multidimensional data model** that store the optimized, summarized or aggregated data which eases the OLAP tools for the quick and easy analysis. Data cube stores the precomputed data and eases online analytical processing.

When it comes to cube, we, all think it as a three-dimensional structure but in data warehousing, we can implement an n-dimensional data cube.

When data is grouped or combined in multidimensional matrices called Data Cubes. The data cube method has a few alternative names or a few variants, such as "Multidimensional databases," "materialized views," and "OLAP (On-Line Analytical Processing)."

The general idea of this approach is to materialize certain expensive computations that are frequently inquired.

For example, a relation with the schema sales (part, supplier, customer, and sale-price) can be materialized into a set of eight views as shown in fig, where **psc** indicates a view consisting of aggregate function value (such as total-sales) computed by grouping three attributes part, supplier, and customer, **p** indicates a view composed of the corresponding aggregate function values calculated by grouping part alone, etc.

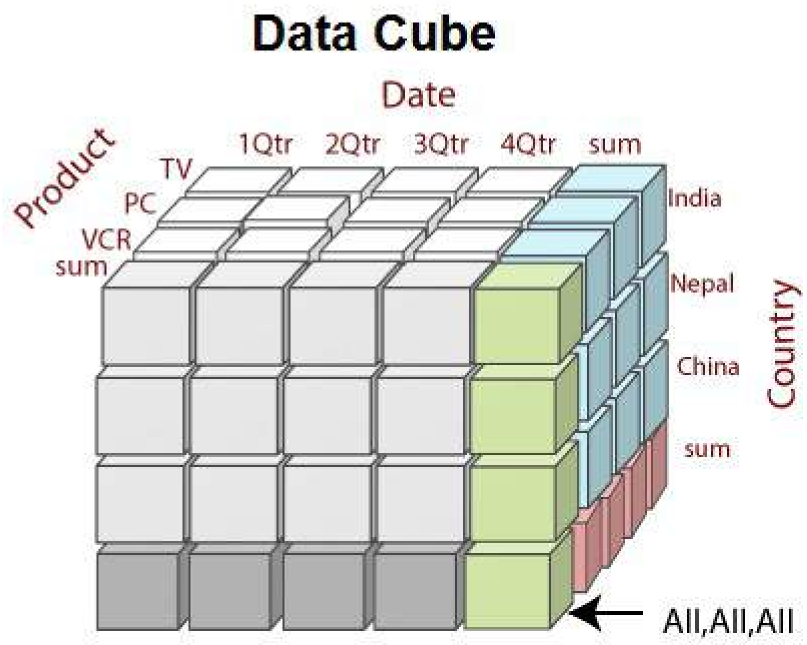


Eight views of data cubes for sales information.

A data cube is created from a subset of attributes in the database. Specific attributes are chosen to be measure attributes, i.e., the attributes whose values are of interest. Another attributes are selected as dimensions or functional attributes. The measure attributes are aggregated according to the dimensions.

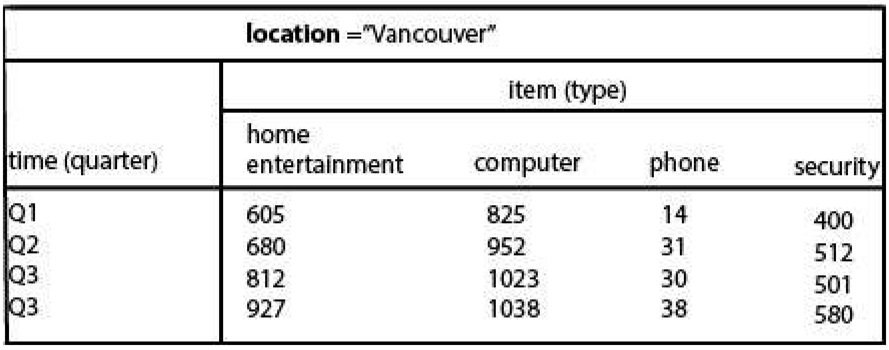
A data cube refers is a three-dimensional (3D) (or higher) range of values that are generally used to explain the time sequence of an image's data. It is a data abstraction to evaluate aggregated data from a variety of viewpoints. It is also useful for imaging spectroscopy as a spectrally-resolved image is depicted as a 3-D volume.

A data cube can also be described as the multidimensional extensions of two-dimensional tables. It can be viewed as a collection of identical 2-D tables stacked upon one another. Data cubes are used to represent data that is too complex to be described by a table of columns and rows. As such, data cubes can go far beyond 3-D to include many more dimensions.



Example: In the **2-D representation**, we will look at the All Electronics sales data for **items sold per quarter** in the city of Vancouver. The measured display in dollars sold (in thousands).

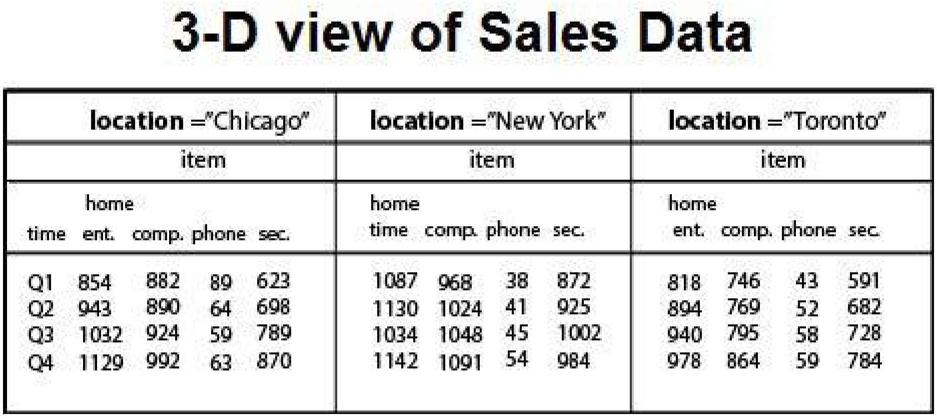
2-D view of Sales Data



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

3-Dimensional Cuboids

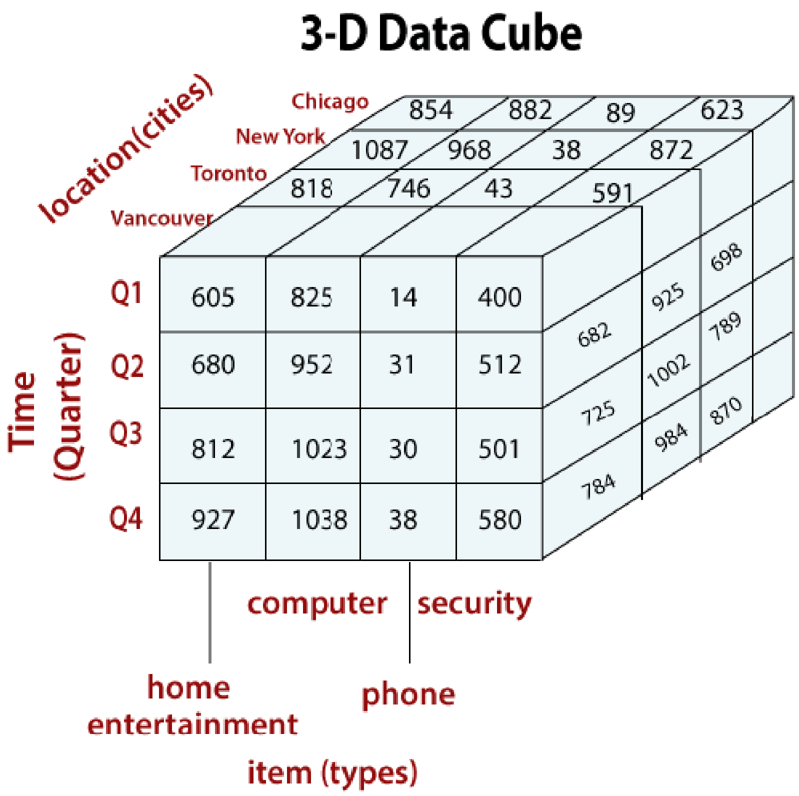
Let suppose we would like to view the sales data with a third dimension. For example, suppose we would like to view the data according to time, item as well as the location for the cities Chicago, New York, Toronto, and Vancouver. The measured display in dollars sold (in thousands). These 3-D data are shown in the table. The 3-D data of the table are represented as a series of 2-D tables.



3-D view of Sales Data

	location ="Chicago"				location ="New York"				location ="Toronto"			
	item				item				item			
time	home ent.	comp.	phone	sec.	home	comp.	phone	sec.	home ent.	comp.	phone	sec.
Q1	854	882	89	623	1087	968	38	872	818	746	43	591
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784

Conceptually, we may represent the same data in the form of 3-D data cubes, as shown in fig:



3-D Data Cube

Let us suppose that we would like to view our sales data with an additional fourth dimension, such as a supplier.

In data warehousing, the data cubes are n-dimensional. The cuboid which holds the lowest level of summarization is called a **base cuboid**.

For example, the **4-D cuboid** in the figure is the base cuboid for the given time, item, location, and supplier dimensions.

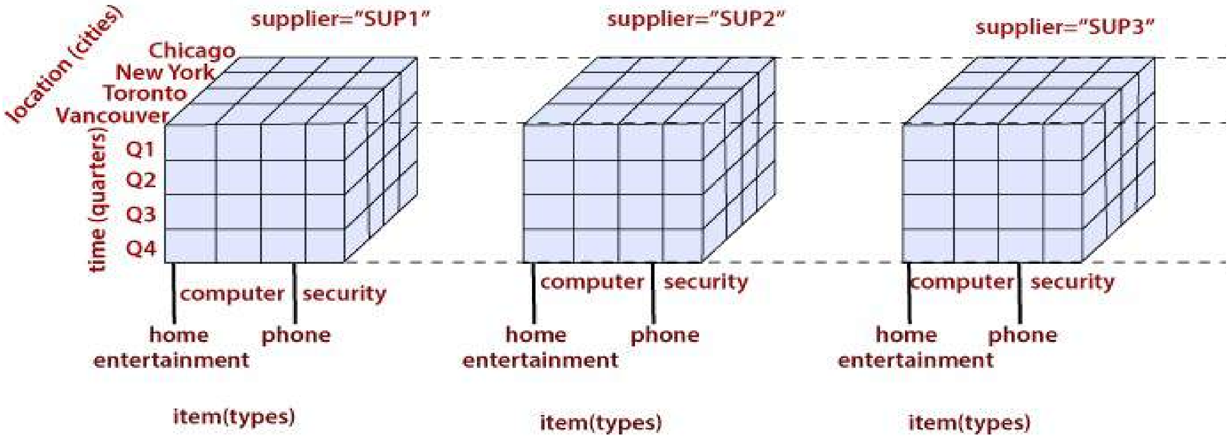
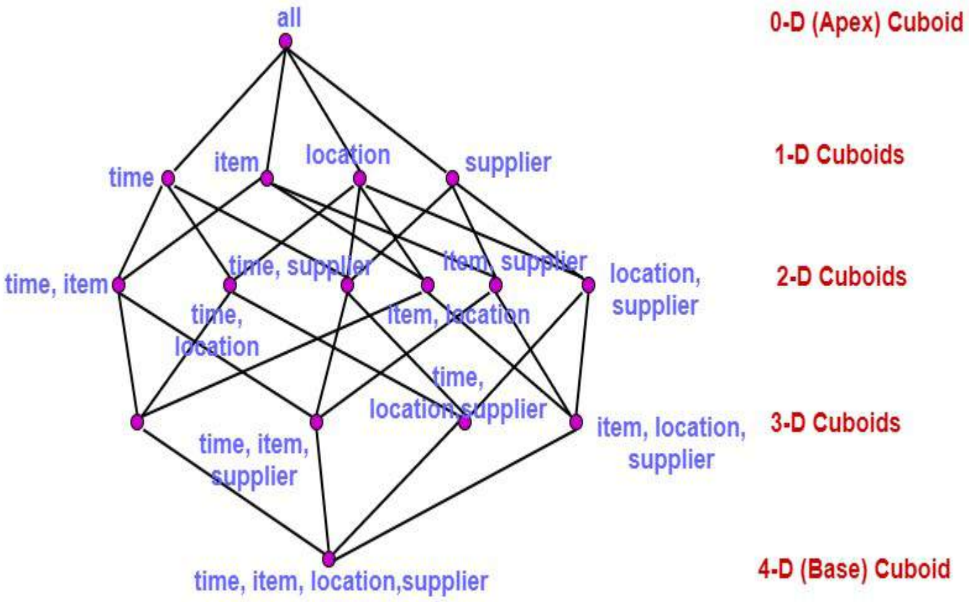


Figure is shown a **4-D data cube** representation of sales data, according to the dimensions time, item, location, and supplier. The measure displayed is dollars sold (in thousands).

The topmost **0-D cuboid**, which holds the highest level of summarization, is known as the apex cuboid. In this example, this is the total sales, or dollars sold, summarized over all four dimensions.

The lattice of cuboid forms a data cube. The figure shows the lattice of cuboids creating 4-D data cubes for the dimension time, item, location, and supplier. Each cuboid represents a different degree of summarization.

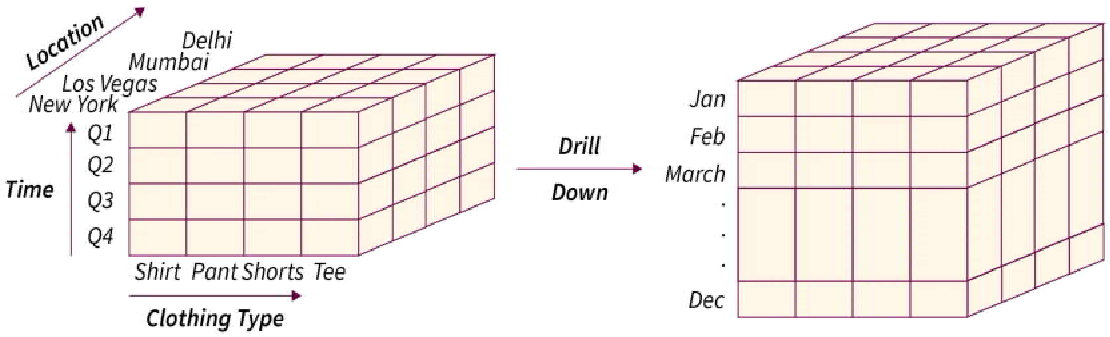


Data cube operations:

Drill Down

Drill down operation allows a user to zoom in on the data cube i.e., the less detailed data is converted into highly detailed data. It can be implemented by either stepping down a concept hierarchy for a dimension or adding additional dimensions to the hypercube.

Example: Consider a cube that represents the annual sales (4 Quarters: Q1, Q2, Q3, Q4) of various kinds of clothes (Shirt, Pant, Shorts, Tees) of a company in 4 cities (Delhi, Mumbai, Las Vegas, New York) as shown below:

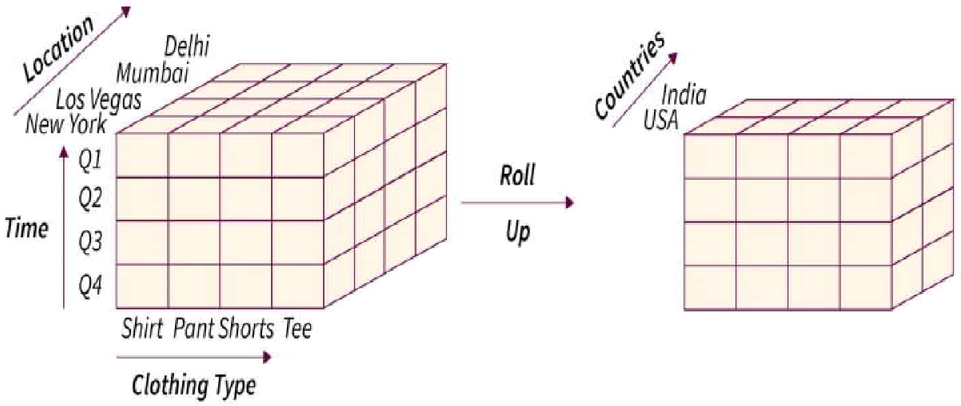


Here, the drill-down operation is applied on the time dimension and the quarter Q1 is drilled down to January, February, and March. Hence, by applying the drill-down operation, we can move down from quarterly sales in a year to monthly or weekly records.

Roll up

It is the opposite of the drill-down operation and is also known as a drill-up or aggregation operation. It is a dimension reduction technique that performs aggregation on a data cube. It makes the data less detailed and it can be performed by combining similar dimensions across any axis.

Example: Considering the above-mentioned clothing company sales example:

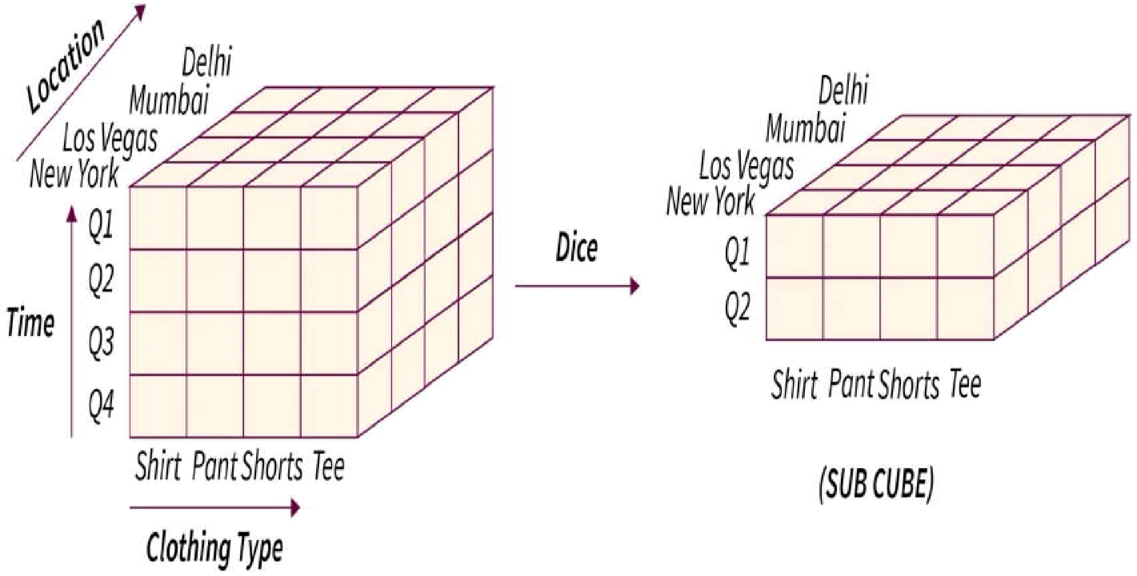


Here, we are performing the Roll-up operation on the given data cube by combining categorizing the sales based on the countries instead of cities.

Dice

Dice operation is used to generate a new sub-cube from the existing hypercube. It selects two or more dimensions from the hypercube to generate a new sub-cube for the given data.

Example: Considering our clothing company sales example:

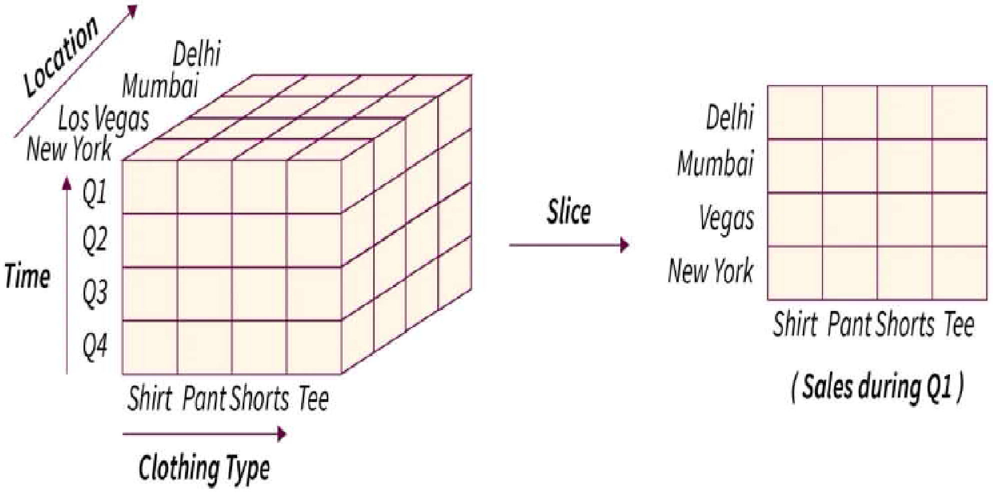


Here, we are using the dice operation to retrieve the sales done by the company in the first half of the year i.e., the sales in the first two quarters.

Slice

Slice operation is used to select a single dimension from the given cube to generate a new sub-cube. It represents the information from another point of view.

Example: Considering our clothing company sales example:

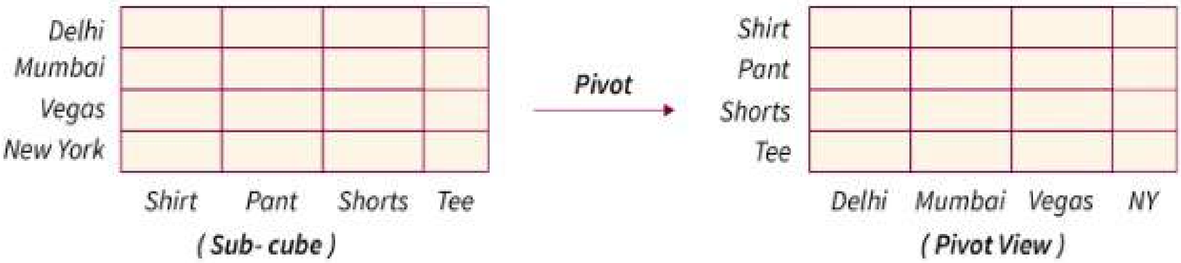


Here, the sales done by the company during the first quarter are retrieved by performing the slice operation on the given hypercube.

Pivot

It is used to provide an alternate view of the data available to the users. It is also known as Rotate operation as it rotates the cube's orientation to view the data from different perspectives.

Example: Considering our clothing company sales example:



Advantages of data cubes:

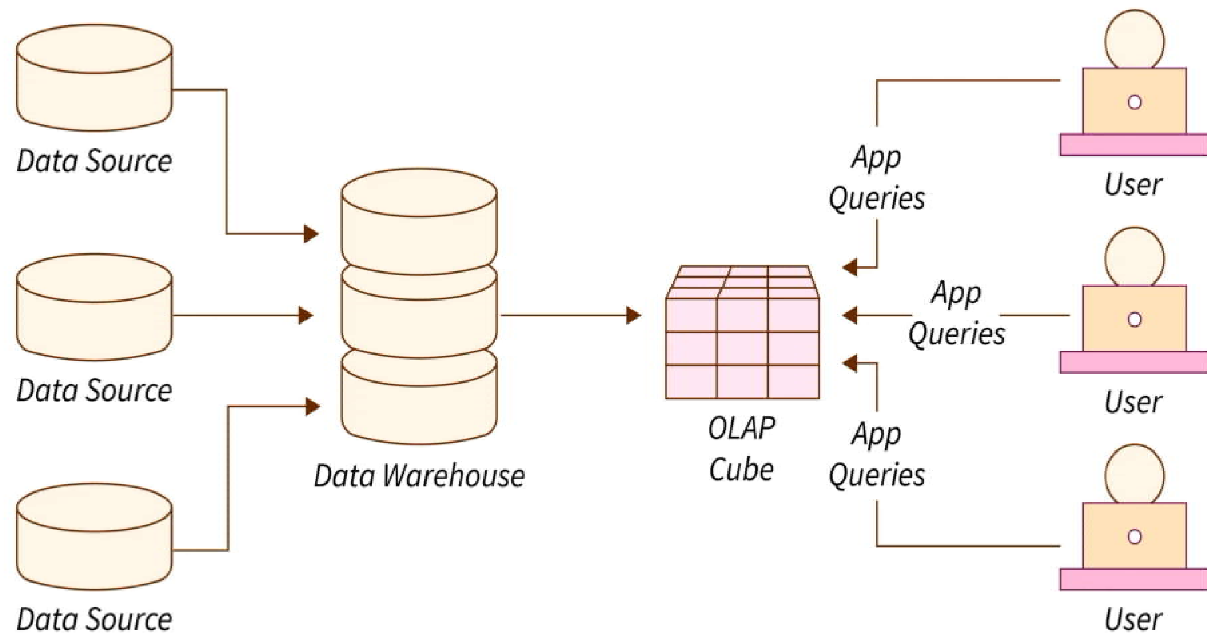
- Helps in giving a summarized view of data.
- Data cubes store large data in a simple way.
- Data cube operation provides quick and better analysis,
- Improve performance of data.

What is OLAP (Online Analytical Processing)?

OLAP implement the multidimensional analysis of business information and support the capability for complex estimations, trend analysis, and sophisticated data modeling. It is rapidly enhancing the essential foundation for Intelligent Solutions containing Business Performance Management, Planning, Budgeting, Forecasting, Financial Documenting, Analysis, Simulation-Models, Knowledge Discovery, and Data Warehouses Reporting. OLAP enables end-clients to perform ad hoc analysis of record in multiple dimensions, providing the insight and understanding they require for better decision making.

The OLAP Process

How data is prepared for online analytical processing (OLAP)



Who uses OLAP and Why?

OLAP applications are used by a variety of the functions of an organization.

Finance and accounting:

- Budgeting
- Activity-based costing
- Financial performance analysis
- And financial modeling

Sales and Marketing

- Sales analysis and forecasting
- Market research analysis
- Promotion analysis
- Customer analysis
- Market and customer segmentation

Production

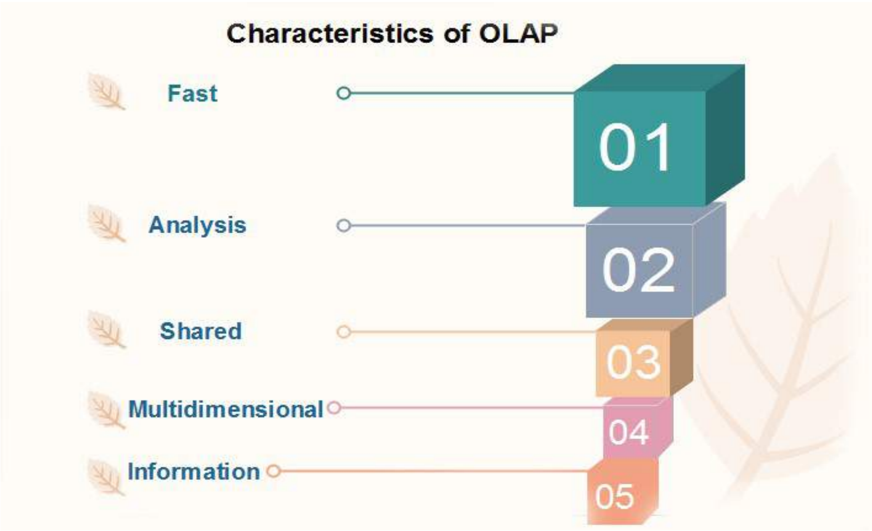
- Production planning
- Defect analysis

OLAP cubes have two main purposes. The first is to provide business users with a data model more intuitive to them than a tabular model. This model is called a Dimensional Model.

The second purpose is to enable fast query response that is usually difficult to achieve using tabular models.

Characteristics of OLAP

In the **FASMI characteristics of OLAP methods**, the term derived from the first letters of the characteristics are:



Fast

It defines which the system targeted to deliver the most feedback to the client within about five seconds, with the elementary analysis taking no more than one second and very few taking more than 20 seconds.

Analysis

It defines which the method can cope with any business logic and statistical analysis that is relevant for the function and the user, keep it easy enough for the target client. Although some preprogramming may be needed we do not think it acceptable if all application definitions have to be allow the user to define new Ad hoc calculations as part of the analysis and to document on the data in any desired method, without having to program so we excludes products (like Oracle Discoverer) that do not allow the user to define new Ad hoc calculation as part of the analysis and to document on the data in any desired product that do not allow adequate end user-oriented calculation flexibility.

Share

It defines which the system tools all the security requirements for understanding and, if multiple write connection is needed, concurrent update location at an appropriated level, not all functions need customer to write data back, but for the increasing number which does, the system should be able to manage multiple updates in a timely, secure manner.

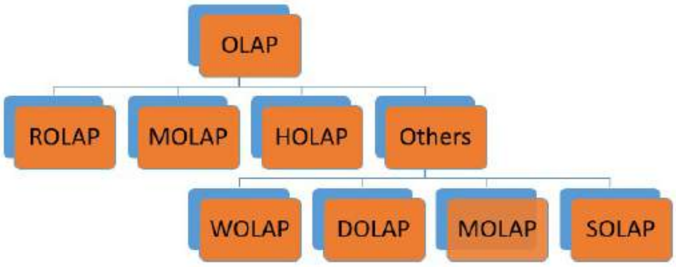
Multidimensional

This is the basic requirement. OLAP system must provide a multidimensional conceptual view of the data, including full support for hierarchies, as this is certainly the most logical method to analyze business and organizations.

Information

The system should be able to hold all the data needed by the applications. Data sparsity should be handled in an efficient manner.

OLAP cube classification: There are 3 main types of OLAP servers are as following:



Relational OLAP (ROLAP) – Star Schema based –
The ROLAP is based on the premise that data need not to be stored multidimensionally in order to viewed multidimensionally, and that it is possible to exploit the well-proven relational database technology to handle multidimensionality of data.In ROLAP data is stored in a relational database.In essence, each action of slicing and dicing is equivalent to adding a “WHERE” clause in SQL statement. ROLAP can handle large amounts of data. ROLAP can leverage functionalities inherent in the relational database.

Multidimensional OLAP (MOLAP) – Cube based –

MOLAP stores data on disks in a specialized multidimensional array structure. OLAP is performed on it relying on the random access capability of the arrays. Arrays element are determined by dimension instances, and the fact data or measured value associated with each cell is usually stored in the corresponding array element. In MOLAP, the multidimensional array is usually stored in a linear allocation according to nested traversal of the axes in some predetermined order.

MOLAP cubes are fast data retrieval, optimal for slicing and dicing and they can perform complex calculation. All calculation are pre-generated when the cube is created.

Hybrid OLAP (HOLAP) –

HOLAP is a combination of ROLAP and MOLAP. HOLAP servers allows storing the large data volumes of detail data.On the one hand, HOLAP leverages the greater scalability of ROLAP. On the other hand, HOLAP leverages the cube technology for faster performance and for summary-type information. Cubes are smaller than MOLAP since detail data is kept in the relational database. The database are used to stores data in the most functional way possible.

The OLAP cube or Hypercube is a special kind of data structure that is optimized for very quick multidimensional data analysis and storage. It is a screenshot of data at a specific point in time.

The main characteristics of OLAP are as follows:

1. **Multidimensional conceptual view:** OLAP systems let business users have a dimensional and logical view of the data in the data warehouse. It helps in carrying slice and dice operations.
2. **Multi-User Support:** Since the OLAP techniques are shared, the OLAP operation should provide normal database operations, containing retrieval, update, adequacy control, integrity, and security.
3. **Accessibility:** OLAP acts as a mediator between data warehouses and front-end. The OLAP operations should be sitting between data sources (e.g., data warehouses) and an OLAP front-end.
4. **Storing OLAP results:** OLAP results are kept separate from data sources.
5. **Uniform documenting performance:** Increasing the number of dimensions or database size should not significantly degrade the reporting performance of the OLAP system.

6. OLAP provides for distinguishing between zero values and missing values so that aggregates are computed correctly.
7. OLAP system should ignore all missing values and compute correct aggregate values.
8. OLAP facilitate interactive query and complex analysis for the users.
9. OLAP allows users to drill down for greater details or roll up for aggregations of metrics along a single business dimension or across multiple dimension.
10. OLAP provides the ability to perform intricate calculations and comparisons.
11. OLAP presents results in a number of meaningful ways, including charts and graphs.

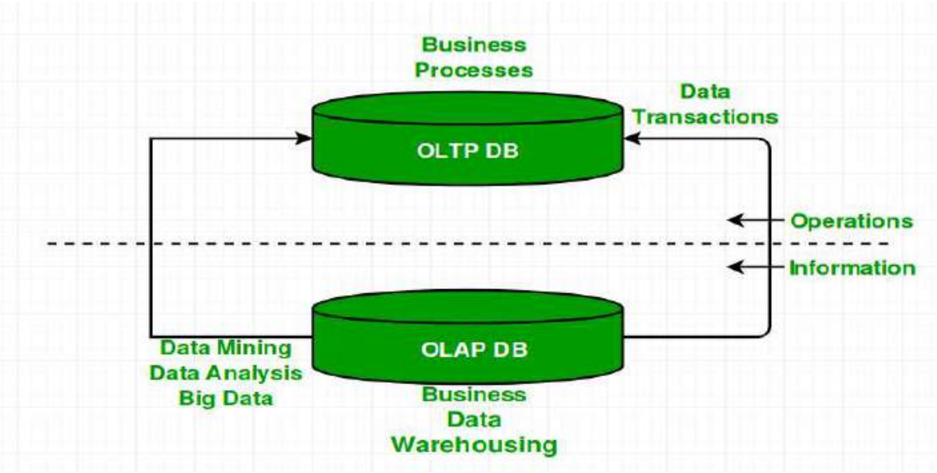
Difference between OLTP and OLAP

OLTP (On-Line Transaction Processing) is featured by a large number of short on-line transactions (INSERT, UPDATE, and DELETE). The primary significance of OLTP operations is put on very rapid query processing, maintaining record integrity in multi-access environments, and effectiveness consistent by the number of transactions per second. In the OLTP database, there is an accurate and current record, and schema used to save transnational database is the entity model (usually 3NF).

Online transaction processing Online transaction processing provides transaction-oriented applications in a 3-tier architecture. OLTP administers the day-to-day transactions of an organization.

Examples: Uses of OLTP are as follows:

- ATM center is an OLTP application.
- OLTP handles the ACID properties during data transactions via the application.
- It’s also used for Online banking, Online airline ticket booking, sending a text message, add a book to the shopping cart.



Implementation of data cube /OLAP:

Implementation steps: follow final guidelines for OLAP implementation:

Step one: Arrange the data in dimensional modeling
Step two: select the data required for removing into OLAP system
Step three: data extraction for the OLAP system
Step four: loading data to the OLAP server
Step five: data aggregation and derived data computation
Step six: implementation of OLAP application on desktop
Step seven: user's training organization

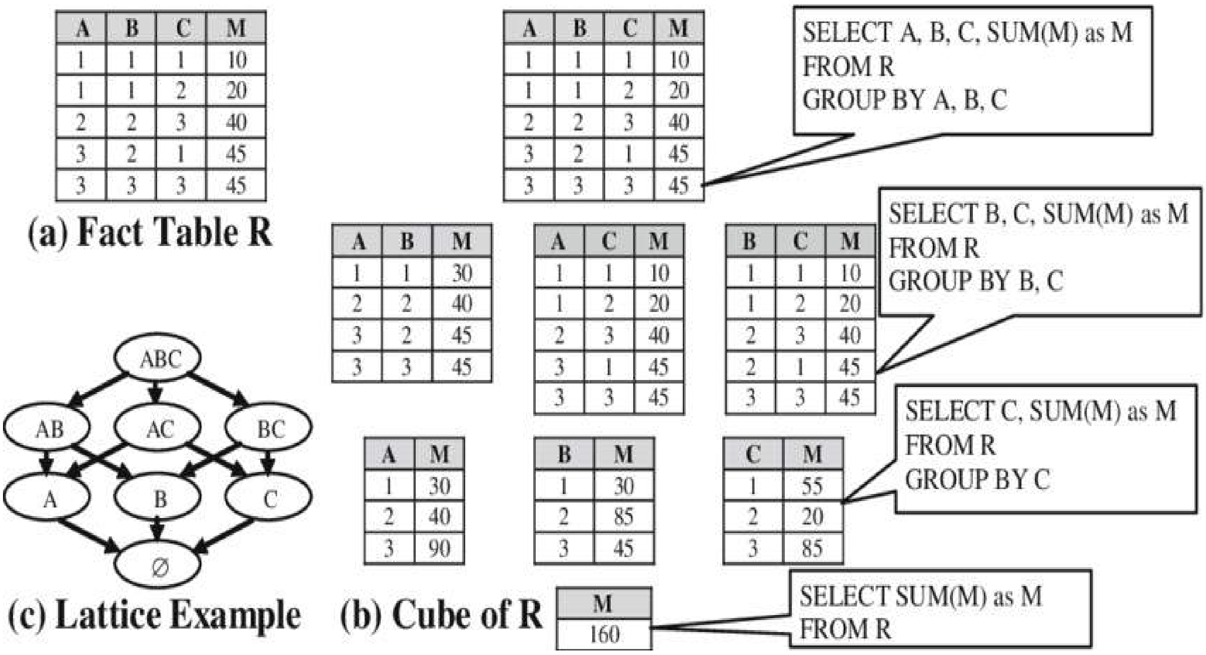
What challenges can OLAP implementation bring?
Some users can consider the following issues as challenges:

- Lack of standardization: every OLAP vendor has its own client interface;
- Scalability: OLAP system is good for processing summary data, but not for large amounts of detailed data.

Data Cube Implementations:

Cube implementation involves the procedures of computation, storage, and manipulation of a data cube, which is a disk structure that stores the results of the aggregate queries that group the tuples of a fact table on all possible combinations of its dimension attributes.
Example: For example in Fig. 1a, assuming that R is a fact table that consists of three dimensions (A, B, C) and one measure M (see definitional entry for *Measure*), the corresponding cube of R appears in Fig. 1b. Each cube node (i.e., view that belongs to the data cube) stores the results of a particular

aggregate query as shown in Fig. 1b. Clearly, if D denotes the number of dimensions of a fact table, the number of all possible aggregate queries is 2^D; hence, in the worst case, the size of the data cube is exponentially larger with respect to D than the size of the original fact table. In typical applications, this may be in the order of gigabytes or even...



A	B	C	M
1	1	1	10
1	1	2	20
2	2	3	40
3	2	1	45
3	3	3	45

(a) Fact Table R

A	B	C	M
1	1	1	10
1	1	2	20
2	2	3	40
3	2	1	45
3	3	3	45

SELECT A, B, C, SUM(M) as M
FROM R
GROUP BY A, B, C

A	B	M
1	1	30
2	2	40
3	2	45
3	3	45

A	C	M
1	1	10
1	2	20
2	3	40
3	1	45
3	3	45

B	C	M
1	1	10
1	2	20
2	3	40
2	1	45
3	3	45

SELECT B, C, SUM(M) as M
FROM R
GROUP BY B, C

A	M
1	30
2	40
3	90

B	M
1	30
2	85
3	45

C	M
1	55
2	20
3	85

SELECT C, SUM(M) as M
FROM R
GROUP BY C

M
160

SELECT SUM(M) as M
FROM R

(c) Lattice Example (b) Cube of R