## **UNIT—III:**

**Data Warehousing and Online Analytical Processing-** DataWarehouse: Basic Concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, Data Generalization by Attribute-Oriented Induction.

# Data Warehouse. How it differs from Operational Database Systems.

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

**Subject-Oriented**: A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject.

**Integrated**: A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product.

**Time-Variant**: Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer.

**Non-volatile**: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered.

**Differences between Operational Database Systems and Data Warehouses**

The major task of online operational database systems is to perform online trans action and query processing. These systems are called online transaction processing (OLTP) systems.

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

**The major distinguishing features of OLTP and OLAP are summarized as follows:**

**Users and system orientation**: An OLTP system is customer-oriented and is used for transaction and query processing by clerks, clients, and information technology professionals. An OLAP system is market-oriented and is used for data analysis by knowledge workers, including managers, executives, and analysts.

**Data contents:** An OLTP system manages current data that, typically, are too detailed to be easily used for decision making. An OLAP system manages large amounts of historic data, provides facilities for summarization and aggregation, and stores and manages information at different levels of granularity.

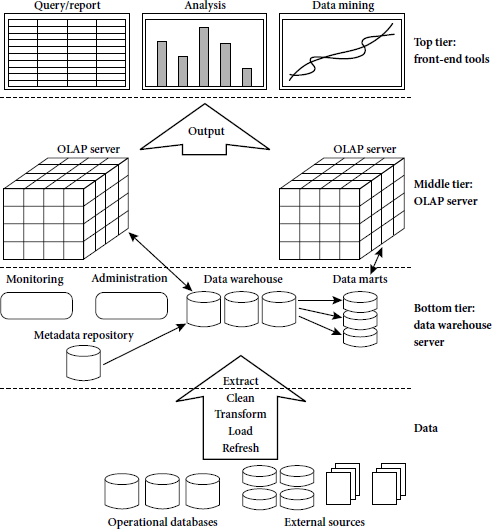
**Database design:** An OLTP system usually adopts an entity-relationship (ER) data model and an application-oriented database design. An OLAP system typically adopts either a star or a snowflake model and a subject-oriented database design.

**View:** An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historic data or data in different organizations. In contrast, an OLAP system often spans multiple versions of a database schema, due to the evolutionary process of an organization.

**Access patterns**: The access patterns of an OLTP system consist mainly of short, atomic transactions. Such a system requires concurrency control and recovery mechanisms. However,accessestoOLAPsystemsaremostlyread-onlyoperations(because most data warehouses store historic rather than up-to-date information), although many could be complex queries.

# Multitiered Architecture of Data Warehousing and the roles played by different tiers.

A Three Tier Data Warehouse Architecture:



#### **Tier-1:**

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

#### **Tier-2:**

The middle tier is an OLAP server that is typically implemented using either a relational OLAP (ROLAP) model or a multidimensional OLAP.

OLAP model is an extended relational DBMS thatmaps operations on multidimensional data to standard relational operations.

A multidimensional OLAP (MOLAP) model, that is, a special-purpose server that directly implements multidimensional data and operations.

#### **Tier-3:**

The top tier is a front-end client layer, which contains query and reporting tools, analysis tools, and/or data mining tools (e.g., trend analysis, prediction, and so on).

# Differentiate between Enterprise Warehouse, Data Mart, and Virtual Warehouse in terms of Data Warehouse Models.

**There are three data warehouse models.**

#### **Enterprise warehouse:**

 An enterprise warehouse collects all of the information about subjects spanning the entire organization.

It provides corporate-wide data integration, usually from one or more operational systems or external information providers, and is cross-functional in scope.



* It typically contains detailed data as well as summarized data, and can range in size from a few gigabytes to hundreds of gigabytes, terabytes, or beyond.

An enterprise data warehouse may be implemented on traditional mainframes, computer super servers, or parallel architecture platforms. It requires extensive business modeling and may take years to design and build.



#### **Data mart:**

 A data mart contains a subset of corporate-wide data that is of value to a specific group of users. The scope is confined to specific selected subjects. For example, a marketing data mart may confine its subjects to customer, item, and sales. The data contained in data marts tend to be summarized.

 Data marts are usually implemented on low-cost departmental servers that are UNIX/LINUX- or Windows-based. The implementation cycle of a data mart is more likely to be measured in weeks rather than months or years. However, it may involve complex integration in the long run if its design and planning were not enterprise-wide.

Depending on the source of data, data marts can be categorized as independent more dependent. Independent data marts are sourced from data captured from one or more operational systems or external information providers, or from data generated locally within a particular department or geographic area. Dependent data marts are source directly from enterprise data warehouses.



#### **Virtual warehouse:**

 A virtual warehouse is a set of views over operational databases. For efficient query processing, only some of the possible summary views may be materialized.

A virtual warehouse is easy to build but requires excess capacity on operational database servers.



# Data Cube, and how does it contribute to Multidimensional Data Modelling.

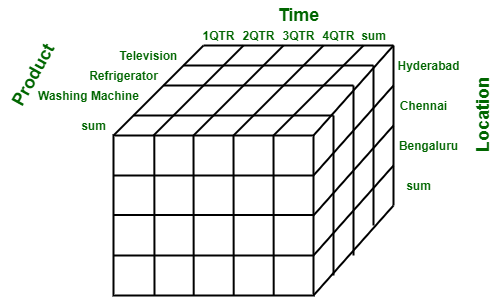
A data cube allows data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.

In general terms, dimensions are the perspectives or entities with respect to which an organization wants to keep records.

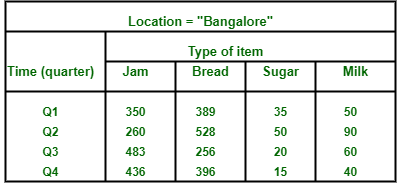
Each dimension may have a table associated with it, called a dimen sion table, which further describes the dimension. For example, a dimension table for item may contain the attributes item name, brand, and type.

OLAP (online analytical processing) and data warehousing uses multi dimensional databases. It is used to show multiple dimensions of the data to users.

It represents data in the form of data cubes. Data cubes allow to model and view the data from many dimensions and perspectives. It is defined by dimensions and facts and is represented by a fact table. Facts are numerical measures and fact tables contain measures of the related dimensional tables or names of the facts.

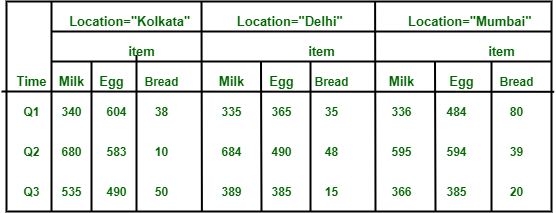


Let us take the example of the data of a factory which sells products per quarter in Bangalore. The data is represented in the table given below :

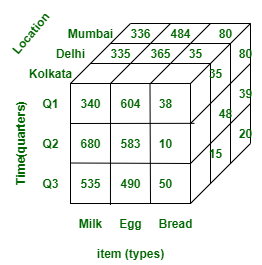


In the above given presentation, the factory’s sales for Bangalore are, for the time dimension, which is organized into quarters and the dimension of items, which is sorted according to the kind of item which is sold. The facts here are represented in rupees (in thousands).

Now, if we desire to view the data of the sales in a three-dimensional**table**, then it is represented in the diagram given below. Here the data of the sales is represented as a two**dimensional table**. Let us consider the data according to item, time and location (like Kolkata, Delhi, Mumbai). Here is the table :



This data can be represented in the form of three dimensions conceptually, which is shown in the image below :



# Compare the Stars, Snowflakes, and Fact Constellations as schemas for Multidimensional Data Models.

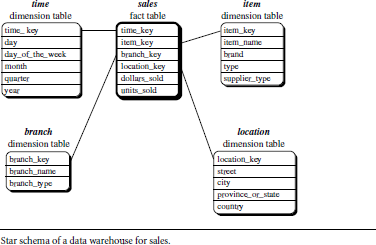
Stars, Snowflakes, and Fact Constellations: Schemas for Multidimensional Databases The entity- relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them. Such a data model is appropriate for on- line transaction processing. A data warehouse, however, requires a concise, subject-oriented schema that facilitates on-line data analysis. The most popular data model for a data warehouse is a multidimensional model. Such a model can exist in the form of a star schema, a snowflake schema, or a fact constellation schema. Let’s look at each of these schema types. Star schema: The most common modeling paradigm is the star schema, in which the data warehouse contains (1) a large central table (fact table) containing the bulk of the data, with no redundancy, and (2) a set of smaller attendant tables (dimension tables), one for each dimension. The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table.

##### **Star schema:**

A star schema for AllElectronics sales is shown in Figure. Sales are considered along four dimensions, namely,time, item, branch, and location. The schema contains a central fact table for sales that contains keys to each of the four dimensions, along with two measures: dollars sold and units sold. To minimize the size of the fact table, dimension identifiers (such as time key and item key) are system-generated identifiers. Notice that in the star schema, each dimension is represented by only one table, and each table contains a set of attributes. For example, the location dimension table contains the attribute set

{location key, street, city, province or state, country}. This constraint may introduce some redundancy.

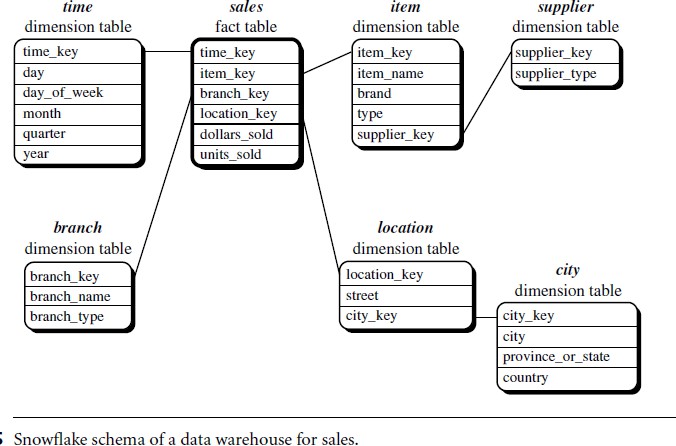
For example, “Vancouver” and “Victoria” are both cities in the Canadian province of British Columbia. Entries for such cities in the location dimension table will create redundancy among the attributes province or state and country, that is, (..., Vancouver, British Columbia, Canada) and (..., Victoria, British Columbia, Canada). Moreover, the attributes within a dimension table may form either a hierarchy (total order) or a lattice (partial order).



##### **Snowflake schema.:**

A snowflake schema for AllElectronics sales is given in Figure Here, the sales fact table is identical to that of the star schema in Figure . The main difference between the two schemas is in the definition of dimension tables.

The single dimension table for item in the star schema is normalized in the snowflake schema, resulting in new item and supplier tables. For example, the item dimension table now contains the attributes item key, item name, brand, type, and supplier key, where supplier key is linked to the supplier dimension table, containing supplier key and supplier type information. Similarly, 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. Notice that further normalization can be performed on province or state and country in the snowflake schema.



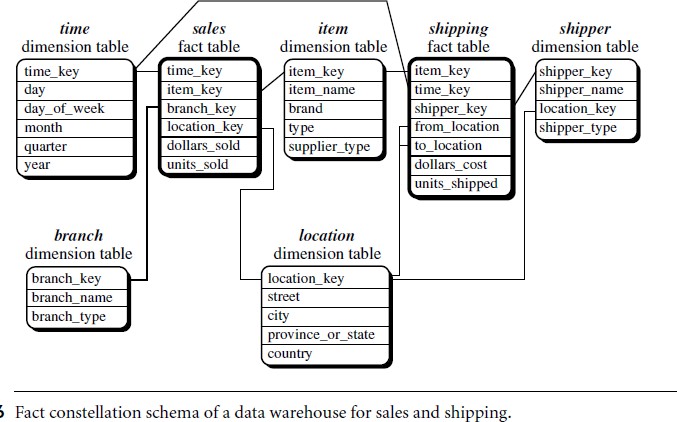
##### **Fact constellation.**

A fact constellation schema is shown in Figure. 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: item key, time key, shipper key, from location, and to location, and two measures: dollars cost and units shipped.

A fact constellation schema allows dimension tables to be shared between fact tables. For example, the dimensions tables for time, item, and location are shared between both the sales and shipping fact tables.

In data warehousing, there is a distinction between a data warehouse and a data mart.

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



# Typical OLAP Operations.

### **OLAP(Online analytical Processing):**

* OLAP is an approach to answering multi-dimensional analytical (MDA) queries swiftly.
* OLAP is part of the broader category of business intelligence, which also encompasses relational database, report writing and data mining.
* OLAP tools enable users to analyze multidimensional data interactively from multiple perspectives.

**OLAP consists of three basic analytical operations:**

* Consolidation (Roll-Up)
* Drill-Down
* Slicing And Dicing
* Pivot (rotate)

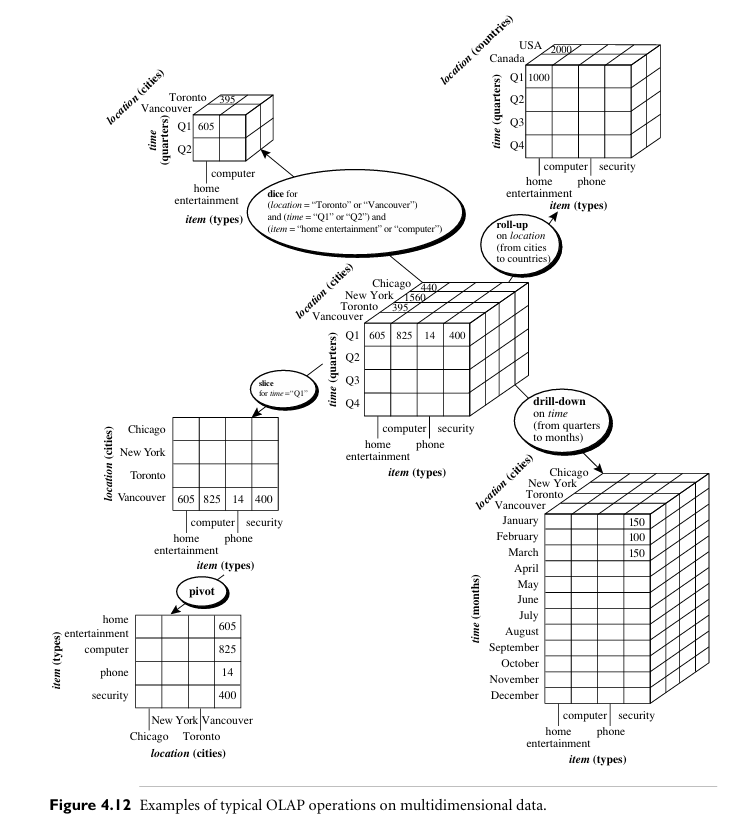
**Roll-up:** The roll-up operation (also called the drill-up operation by some vendors) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction. When roll-up is performed by dimension reduction, one or more dimensions are removed from the given cube.

**Drill-down:** Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data. Drill-down can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions.

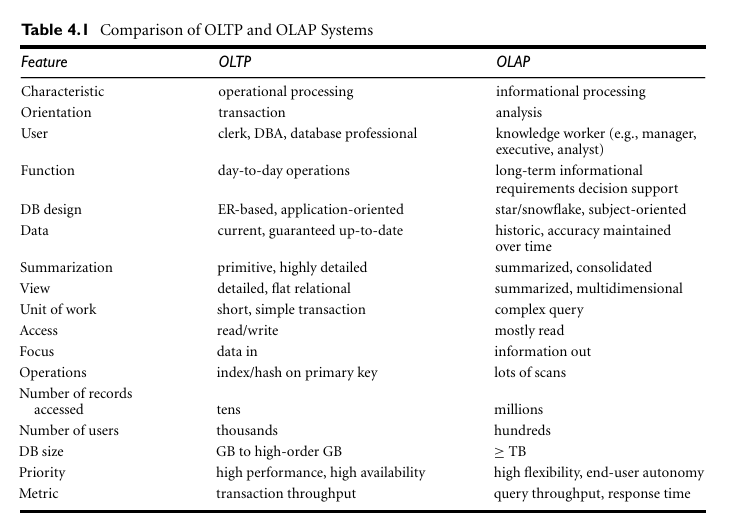
**Slice and dice:** The slice operation performs a selection on one dimension of the given cube, resulting in a sub cube.

**Pivot (rotate):** Pivot (also called rotate) is a visualization operation that rotates the data axes in view to provide an alternative data presentation.

**Other OLAP operations:** Some OLAP systems offer additional drilling operations. For example, drill-across executes queries involving (i.e., across) more than one fact table. The drill-through operation uses relational SQL facilities to drill through the bottom level of a data cube down to its back-end relational tables.



# differences between OLTP and OLAP systems.

****

# Data Warehouse Design Process and usage.

A data warehouse can be built using a top-down approach, a bottom-up approach, or a combination of both.

**The top-down approach** starts with the overall design and planning. It is useful in cases where the technology is mature and well known, and where the business problems that must be solved are clear and well understood.

**The bottom-up approach** starts with experiments and prototypes. This is useful in the early stage of business modeling and technology development. It allows an organization to move forward at considerably less expense and to evaluate the benefits of the technology before making significant commitments.

**In the combined approach**, an organization can exploit the planned and strategic nature of the top-down approach while retaining the rapid implementation and opportunistic application of the bottom-up approach.

**The warehouse design process consists of the following steps:**

* Choose a business process to model, for example, orders, invoices, shipments, inventory, account administration, sales, or the general ledger. If the business process is organizational and involves multiple complex object collections, a data warehouse model should be followed. However, if the process is departmental and focuses on the analysis of one kind of business process, a data mart model should be chosen.
* Choose the grain of the business process. The grain is the fundamental, atomic level of data to be represented in the fact table for this process, for example, individual transactions, individual daily snapshots, and so on.
* Choose the dimensions that will apply to each fact table record. Typical dimensions are time, item, customer, supplier, warehouse, transaction type, and status.
* Choose the measures that will populate each fact table record. Typical measures are numeric additive quantities like dollars sold and units sold.

# Implementation of A Data Warehouse

Data warehouses contain huge volumes of data. OLAP servers demand that decision support queries be answered in the order of seconds. Therefore, it is crucial for data warehouse systems to support highly efficient cube computation techniques, access methods, and query processing techniques.

1. **Efficient Data Cube Computation:**

Efficient data cube computation is crucial for quickly analyzing and summarizing large datasets. A data cube is a multi-dimensional array that represents pre-aggregated measures for various combinations of dimensions. The process of efficiently computing a data cube involves optimizing the storage and computation of aggregated values to facilitate fast querying.

**Here are some techniques and considerations for efficient data cube computation:**

**Roll-Up and Drill-Down:**

Implement roll-up (aggregating data at higher levels) and drill-down (breaking down data into finer levels) operations efficiently. These operations are fundamental to data cube computation and help in navigating between different levels of granularity.

**Incremental Cube Updates:**

Design mechanisms for incremental updates to the data cube. Instead of recomputing the entire cube when new data arrives, update only the affected portions to minimize computational costs.

**Use of Aggregate Functions:**

Utilize database management system (DBMS) aggregate functions for cube computations whenever possible. These functions are optimized for performance and can efficiently compute sums, averages, counts, etc.

**Query Optimization:**

Optimize query processing by employing techniques such as indexing, caching, and query rewriting. This ensures that the system selects the most efficient execution plans for responding to user queries.

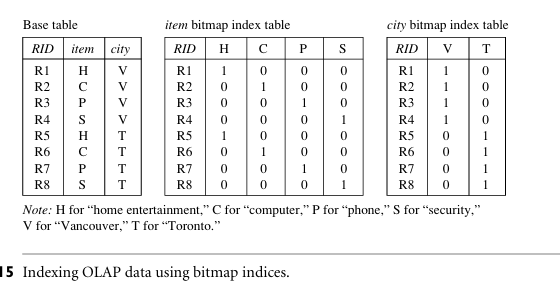
1. **Accessing Methods:**

Accessing data quickly from a data warehouse is crucial for supporting fast and efficient decision-making processes. Several methods and techniques are employed to optimize data access in a data warehouse environment. Indexing is a common strategy:

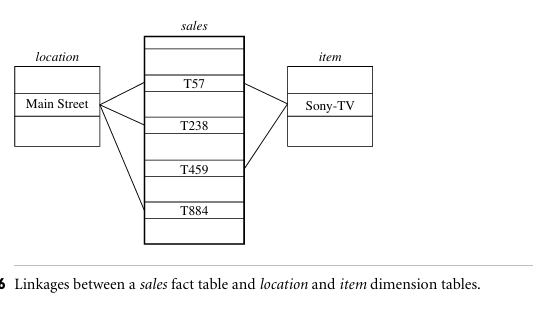
**Indexing:**

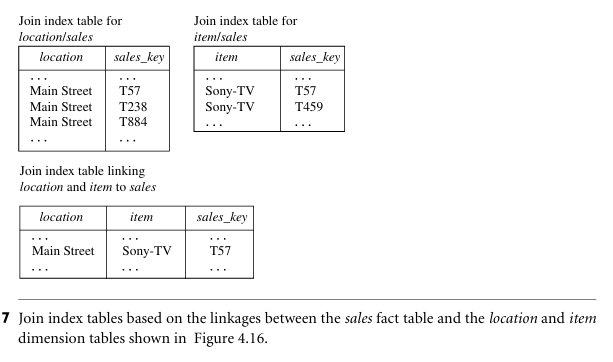
Implementing indexes on key columns can significantly improve query performance. Indexes allow the database engine to locate and retrieve data more quickly. However, it's essential to carefully choose which columns to index, as indexing comes with additional storage overhead. we examine how to index OLAP data by bitmap indexing and join indexing.

**2.1 The bitmap indexing** method is popular in OLAP products because it allows quick searching in data cubes. In the bitmap index for a given attribute, there is a distinct bit vector, Bv, for each value v in the attribute’s domain. If a given attribute’s domain con sists of n values, then n bits are needed for each entry in the bitmap index (i.e., there are n bit vectors). If the attribute has the value v for a given row in the data table, then the bit representing that value is set to 1 in the corresponding row of the bitmap index. All other bits for that row are set to 0.

****

* 1. **Join indexing** refers to the use of indexes to optimize the performance of join operations in a relational database. Join operations involve combining rows from two or more tables based on a related column. Indexes play a crucial role in enhancing the speed of joins by allowing the database engine to quickly locate and match rows from the joined tables.

****

****

1. **Query Processing Techniques(**Efficient Processing of OLAP Queries):

OLAP (Online Analytical Processing) queries involve complex analysis of large volumes of data to provide insights and support decision-making. Efficient processing of OLAP queries is crucial for delivering timely and responsive results. Here are some strategies for optimizing the processing of OLAP queries:

**Multidimensional Data Modeling:**

Design the OLAP database using a multidimensional model, where data is organized into dimensions and hierarchies. This model aligns with the analytical nature of OLAP queries and supports efficient slicing, dicing, and drilling operations.

**Materialized Views:**

Create materialized views that store precomputed results of frequently executed OLAP queries. Materialized views can significantly reduce query response times by eliminating the need to recompute aggregations.

**Indexed Views:**

Use indexed views to store the results of aggregations and complex queries. Indexed views can provide fast access to precomputed data, especially when the underlying data is not frequently updated.

# basic approaches for Data generalization

Data generalization, also known as data summarization or data compression, is the process of reducing the complexity of large datasets by identifying and representing patterns in the data in a more simplified form. This is typically done in order to make the data more manageable and easier to analyze and interpret.

## **Introduction to Data Generalization**

Data generalization is a crucial step in the data analysis process, as it allows us to make sense of large and complex datasets by identifying patterns and trends that may not be immediately apparent. By simplifying the data, we can more easily identify relationships, classify data points, and draw conclusions about the underlying data.

There are a number of different approaches that can be used to generalize data, each with its own strengths and limitations.

**Basic Approaches of Data Generalization:**

There are two main approaches to data generalization − the data cube approach and attribute orientation induction.

### **Data Cube Approach:**

The data cube approach is a method of data generalization that involves creating a multi-dimensional data structure, known as a data cube, to represent the data. The data cube is formed by aggregating the data along different dimensions or attributes, such as time, location, or product type. This allows users to easily slice and dice the data to view and analyze it from different perspectives.

One of the main benefits of the data cube approach is that it allows users to quickly and easily perform ad-hoc queries and drill down into the data to identify patterns and trends. It is particularly well-suited for use in data warehousing and business intelligence applications.

**Attribute-Oriented Induction (AOI):**

It is a technique used in data mining and knowledge discovery to generate generalized patterns or rules from a set of data. The process involves identifying relevant attributes and their generalizations to extract meaningful knowledge from the data. Data generalization refers to the creation of higher-level, more abstract representations of the data, facilitating the discovery of patterns and trends.

**Here's a general overview of the process of data generalization by Attribute-Oriented Induction:**

* Let's consider a simplified example to illustrate Attribute-Oriented Induction for data generalization. Suppose we have a dataset representing customer purchases at an online store.
* The dataset includes the following attributes: **CustomerID, ProductCategory, PurchaseAmount, and DayOfWeek.**

1. **Attribute Selection:**

* The process starts by selecting relevant attributes from the dataset. These attributes are chosen based on their importance to the analysis or the problem at hand.

**Example:** We select the attributes ProductCategory and DayOfWeek as they are relevant to understanding customer purchasing patterns.

1. **Attribute-Oriented Induction:**

* Attribute-Oriented Induction involves the systematic generation of generalized patterns or rules based on the selected attributes.

**Example:**

We create a hierarchy of generalized patterns:

Generalization Level 1: Overall purchasing patterns.

Generalization Level 2: Patterns based on DayOfWeek.

Generalization Level 3: Patterns based on ProductCategory.

1. **Rule Generation:**

* Based on the generalized patterns, rules are generated to describe relationships and associations within the data. These rules provide actionable insights and can be used for decision-making.

**Example:**

we generate rules such as:

Rule 1: If DayOfWeek is 'Saturday' or 'Sunday', then the PurchaseAmount tends to be higher.

Rule 2: If ProductCategory is 'Electronics' and DayOfWeek is not 'Saturday' or 'Sunday', then the PurchaseAmount tends to be higher.