



CLASS:BCA 5TH SEM
BATCH:2021-24

SUBJECT: Data Warehousing and Mining

Notes as per IKGPTU Syllabus

NAME OF FACULTY: -----Kalash Koul-----

FACULTY OF BUSINESS MANAGEMENT, SBS COLLEGE. LUDHIANA

Data Warehousing and Mining

UNIT-II

Short Questions

Q1 Write a short note on Data summarization.

Ans. Data Summarization is a simple term for a short conclusion of a big theory or a paragraph. This is something where you write the code and, in the end, you declare the result in the form of summarizing data. Data summarization has great importance in data mining. As nowadays a lot of programmers and developers work on big data theory. Earlier, you used to face difficulties declaring the result, but now there are so many relevant tools in the market where you can use in programming or wherever you want in your data.

We are living in a digital world where data transfers in a second and it is much faster than human capability. As a result, data becomes more complex and takes time to summarize information. Always retrieve data in the form of category what type of data you want in the data, or we can say use filtration when you retrieve data. Although, "Data Summarization" technique gives the good amount of quality to summarize the data. Moreover, a customer or user can take benefits in their research.

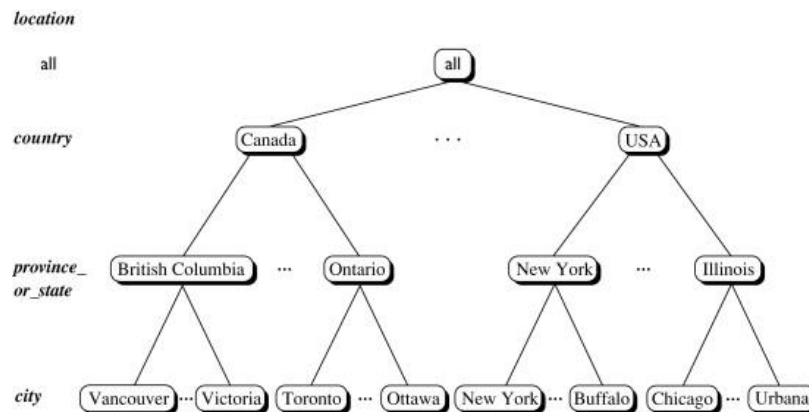
Q2 What are certain methods of Data Transformation?

Ans In data transformation, the data are transformed or consolidated into forms appropriate for mining. Data transformation can involve the following:

1. Normalization, where the attribute data are scaled to fall within a small, specified range, such as -1.0 to 1.0, or 0 to 1.0.
2. Smoothing works to remove noise from data. Such techniques include binning, clustering, and regression.
3. Aggregation, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated to compute monthly and annual total amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.
4. Generalization of the data, where low level or 'primitive' (raw) data are replaced by higher level concepts using concept hierarchies. For example, categorical attributes, like street, can be generalized to higher level concepts, like city or county. Similarly, values for numeric attributes, like age, may be mapped to higher level concepts, like young, middle-aged, and senior.

Q3 What is concept hierarchy?

Ans. A **concept hierarchy** defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts. Consider a concept hierarchy for the dimension location. City values for location include Vancouver, Toronto, New York, and Chicago. Each city, however, can be mapped to the province or state to which it belongs. For example, Vancouver can be mapped to British Columbia, and Chicago to Illinois. The provinces and states can in turn be mapped to the country (e.g., Canada or the United States) to which they belong. These mappings form a concept hierarchy for the dimension location, mapping a set of low-level concepts (i.e., cities) to higher-level, more general concepts (i.e., countries).



Q4 What is a data model and what are its components?

Ans. A data model is a graphical view of data created for analysis and design purposes. Data modeling designing data warehouse databases in detail. It can be defined as an integrated collection of concepts that can be used to describe the structure of the database including data types, relationships between data and constraints that should apply on the data.

A data model comprises of the following three components.

- 1) Structural part:-** It consists of a set of rules according to which database can be constructed.
- 2) Manipulative part:-** It defines the types of operations that are allowed on the data. This includes the operations that are used for updating or retrieving data from the database and for changing the structure of the students.
- 3) Integrity rules:-** Rules which ensure that the data is accurate.

Q5 What are the 3 types of data warehouse models?

Ans. 3 types of data warehouse models are:-

Enterprise Warehouse- An enterprise warehouse collects all details comparing of all information about subjects spanning the entire organization. It provides corporate wide data integration, usually from one or more operational systems and from external information providers. It takes extensive business modeling, and it takes many years to design and build.

Data Marts- A data mart consists of a subset of corporate wide data that is of value to a specific group of users. The scope is restricted to specifically selected subjects. The data contained in a data mart tends to be summarized.

Virtual warehouse- A virtual warehouse is a set of views over operational databases. A virtual warehouse is essentially a business database. The data found in a virtual warehouse is usually copied from multiple sources throughout a production system.

Long Questions

Q1 What are the approaches to building a data warehouse?

Ans. A Data warehouse is a heterogeneous collection of different data sources organized under unified schema. Builders should take a broad view of the anticipated use of the warehouse while constructing a data warehouse. During the design phase, there is no way to anticipate all possible queries or analyses. Some characteristics of Data warehouse are:

- Subject oriented
- Integrated
- Time Variant
- Non-volatile

Building a Data Warehouse –

Some steps that are needed for building any data warehouse are as following below:

- **To extract the data (transnational) from different data sources:** For building a data warehouse, data is extracted from various data sources and that data is stored in central storage area. For extraction of the data Microsoft has come up with an excellent tool. When you purchase Microsoft SQL Server, then this tool will be available free of cost.
- **To transform the transnational data:** There are various DBMS where many of the companies stores their data. Some of them are MS Access, MS SQL Server, Oracle, Sybase etc. Also, these

companies save the data in spreadsheets, flat files, mail systems etc. Relating data from all these sources is done while building a data warehouse.

- **To load the data (transformed) into the dimensional database:** After building a dimensional model, the data is loaded in the dimensional database. This process combines the several columns together or it may split one field into the several columns. There are two stages at which transformation of the data can be performed and they are: while loading the data into the dimensional model or while data extraction from their origins.
- **To purchase a front-end reporting tool:** There are top notch analytical tools available in the market. These tools are provided by the several major vendors. A cost-effective tool and Data Analyzer is released by Microsoft on its own.

For the warehouse there is an acquisition of the data. There must be a use of multiple and heterogeneous sources for the data extraction, for example databases. There is a need for the consistency for which formation of data must be done within the warehouse. Reconciliation of names, meanings and domains of data must be done from unrelated sources. There is also a need for the installation of the data from various sources in the data model of the warehouse.

- To provide the time variant data
- To store the data as per the data model of the warehouse
- Purging the data
- To support the updating of the warehouse data

Q2 What steps need to be implemented to build a data warehouse?

Ans. I) Access Tools-Currently no single tool in the market can handle all possible data warehouse access needs. Therefore, most implementations rely on a suite of tools.

ii) Data Placement Strategies-As Data Warehouse grows, there are at least two options for Data Placement. One is to put some of the data in the data warehouse into another storage medium (WORM, RAID). The second option is to distribute data in data warehouse across multiple servers.

iii) Data Extraction, Cleanup, Transformation, and Migration-As components of the Data Warehouse architecture, proper attention must be given to Data Extraction, which represents a critical success factor for a data warehouse architecture.

1. The ability to identify data in the data source environments that can be read by conversion tools is important.

2. Support for the flat files. (VSAM, ISM, IDMS) is critical, since bulk of the corporate data is still maintained in this type of data storage.

3. The capability to merge data from multiple data stores is required in many installations.
4. The specification interface to indicate the data to be extracted and the conversion criteria is important.
5. The ability to read information from data dictionaries or import information from repository product is desired

iv) Metadata-A frequently occurring problem in Data Warehouse is the problem of communicating to the end user what information resides in the data warehouse and how it can be accessed. The key to providing users and applications with a roadmap to the information stored in the warehouse is the metadata. It can define all data elements and their attributes, data sources and timing, and the rules that govern data use and data transformations. Meta data needs to be collected as the warehouse is designed and built.

Q3 What are technical issues to be considered in building a data warehouse?

Ans. The following technical issues are required to be considered for designing and implementing a data warehouse.

1. Hardware platform for data warehouse
2. DBMS for supporting data warehouse.
3. Communication and network infrastructure for a data warehouse
4. Software tools for building, operating, and using data warehouse.

1. Hardware platforms for data warehouse:- Organization normally likes to utilize the already existing platforms for data warehouse development. However, the disk storage requirements for a data warehouse will be significantly large, especially in comparison with a single application.

Thus, hardware with large data storage capacity is essential for data warehousing. For every data size identified, the disk space provided should be two to three times that of the data to accommodate processing, indexing etc.

2. DBMS for supporting data warehouse:- After hardware selection, a factor most important is the DBMA selection. This determines the speed performance of the data warehousing environment. The requirements of a DBMA for data warehousing environment are scalability, performance in high volume storage and processing and throughput in traffic. All the well-known RDBMA vendors like:- IBM, ORACLE, Sybase support parallel database processing, some of them have even improved their architectures so as to better suit the specialized requirements of a data warehouse.

3. Communication and network infrastructure for a data warehouse:- Data warehouse can be internet or web enabled or intranet enabled as the choice may be. If the web enabled, the networking is taken care of by the internet. If only intranet based, then the appropriate LAN operational environment should be provided to be accessible to all the identified users.

4. Software tools for building, operating, and using data warehouse:- All the data warehouse vendors are not currently providing comprehensive single window software tools capable of handling all aspects of a data warehousing project implementation.

The types of access and reporting are as follows.

- Statistical analysis and forecasting
- Data visualization, graphing, and charting
- Complex textual search
- Ad hoc user specific queries
- Predefined repeatable queries
- Reporting and analysis by drilling down

Q4 What is data pre-processing and what are its objectives?

Ans. Today's real-world databases are highly subject to noisy, missing, and inconsistent data due to their typically huge size, and their likely origin from multiple, heterogeneous source. Incomplete data can occur for several reasons. Attributes of interest may not always be available, such as customer information for sales transaction data. Other data may not be included simply because it was not considered important at the time of entry. Some of the major reason for noisy data is:

- The data collection instruction used may be faulty.
- There may have been human or computer errors occurring at data entry.
- Error in data transmission can also occur.
- There may be technology limitations, such as limited buffer size for coordinating synchronized data transfer and consumption.

Objectives of data Preprocessing

- **Size reduction of the Input space-** Reducing the number of input variables or the size of the input space is a common goal of preprocessing. The objective is to get a reasonable overview of the data set without losing the most important relationship of the data. If the input space is large, one may identify the most important input variables and eliminate the unimportant variables by combining several variables as a single variable.
- **Smoother relationship-** Another commonly used type of preprocessing is problem transformation. The original problem is transformed into a simpler problem.
- **Data normalization-** For many practical problems, the units used to measure each of the input variables can change the data and make the range of values much larger than others. This results in unnecessarily complex relationships by making the nature of the mapping along some dimensions much different from others.

- **Noise Reduction-** A sequence of data may involve useful data, noisy data, and inconsistent data. Preprocessing may reduce the noisy and inconsistent data. The data corrupted with noise can be recovered with preprocessing techniques.
- **Features Extraction-** If the key attribute or features characterizing the data can be extracted, the problem encountered can be easily solved.

Q5 What are certain data pre-processing techniques?

Ans. Data Cleaning

Data Cleaning is a process of cleaning raw data by handling irrelevant and missing tuples. While working on our machine learning projects, the data sets which we take might not be perfect they might have many impurities, Noisy values, and most times the actual data might be missing. the major problems we will be facing during data cleaning are:

1. Missing Values: If it is noted that there are many tuples that have no recorded value for several attributes, then the missing values can be filled in for the attribute by various methods described below:

- **Ignore the tuple:** This is usually done when the class label is missing (assuming the mining task involves classification or description). This method is not very effective, unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably.
- **Fill in the missing value manually:** In general, this approach is time-consuming and may not be feasible given a large data set with many missing values.
- **Use a global constant to fill in the missing value:** Replace all missing attribute values by the same constant, such as a label like \Unknown", or $-\infty$. If missing values are replaced by, say, \Unknown", then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common | that of \Unknown". Hence, although this method is simple, it is not recommended.
- Use the attribute mean to fill in the missing value.
- Use the attribute mean for all samples belonging to the same class as the given tuple.
- Use the most probable value to fill in the missing value: This may be determined with inference-based tools using a Bayesian formalism or decision tree induction.

Methods 3 to 6 bias the data. The filled-in value may not be correct. Method 6, however, is a popular strategy. In comparison to the other methods, it uses the most information from the present data to predict missing values.

2. Noisy Data: Noise is a random error or variance in a measured variable. Given a numeric attribute such as, say, price, how can the data be "smoothed" to remove the noise? The following data smoothing techniques describe this.

- a. **Binning methods:** Binning methods smooth a sorted data value by consulting the "neighborhood" or values around it. The sorted values are distributed into several 'buckets', or bins. Because binning methods consult the neighborhood of values, they perform local smoothing values around it. The sorted values are distributed into several 'buckets', or bins. Because binning methods consult the neighborhood of values, they perform local smoothing.
- b. **Clustering:** Outliers may be detected by clustering, where similar values are organized into groups or "clusters".
- c. **Combined computer and human inspection:** Outliers may be identified through a combination of computer and human inspection. In one application, for example, an information-theoretic measure was used to help identify outlier patterns in a handwritten character database for classification. The measure's value reflected the "surprise" content of the predicted character label with respect to the known label. Outlier patterns may be informative (e.g., identifying useful data exceptions, such as different versions of the characters "0" or "7"), or "garbage" (e.g., mislabeled characters). Patterns whose surprise content is above a threshold are output to a list. A human can then sort through the patterns in the list to identify the actual garbage ones.

This is much faster than having to manually search through the entire database. The garbage patterns can then be removed from the (training) database.
- d. **Regression:** Data can be smoothed by fitting the data to a function, such as with regression. Linear regression involves finding the "best" line to fit two variables, so that one variable can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two variables are involved, and the data are fit to a multidimensional surface. Using regression to find a mathematical equation to fit the data helps smooth out the noise.

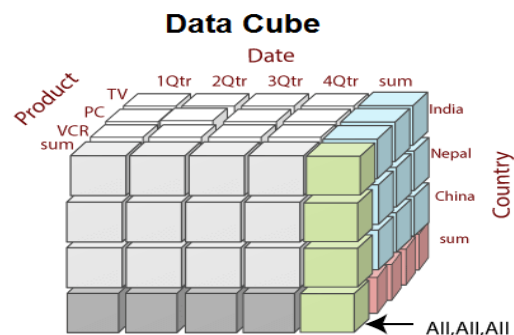
3. Inconsistent data: There may be inconsistencies in the data recorded for some transactions. Some data inconsistencies may be corrected manually using external references. For example, errors made at data entry may be corrected by performing a paper trace. This may be coupled with routines designed to help correct the inconsistent use of codes. Knowledge engineering tools may also be used to detect the violation of known data constraints. For example, known functional dependencies between attributes can be used to find values contradicting the functional constraints.

Q6 What are data cubes and what 2-D and 3-D view of sales data?

Ans. 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.



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"				
time (quarter)	item (type)			
	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

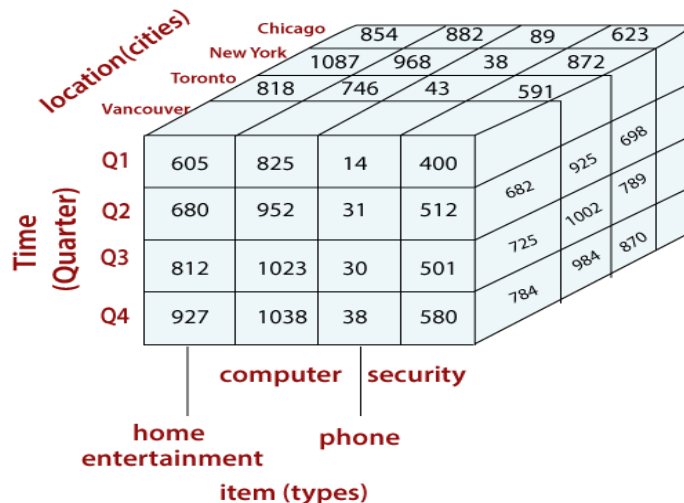
3-Dimensional Cuboids

Let's 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				
home					home					home				
time	ent.	comp.	phone	sec.	time	comp.	phone	sec.	time	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		

3-D Data Cube



Q7 Define different data warehouse schemas with diagram.

Ans. Multidimensional Schema is especially designed to model data warehouse systems. The schemas are designed to address the unique needs of very large databases designed for analytical purposes (OLAP).

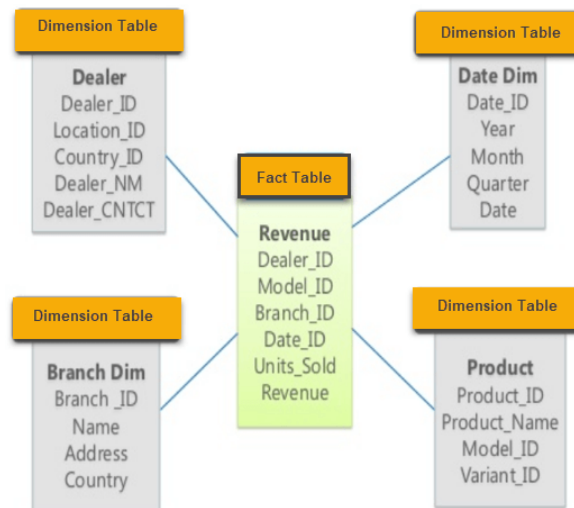
Types of Data Warehouse Schema:

- Star Schema
- Snowflake Schema
- Galaxy Schema

1. Star Schema

In the **STAR Schema**, the center of the star can have one fact table and several associated dimension tables. It is known as star schema as its structure resembles a star. The star schema is the simplest type of Data Warehouse schema. It is also known as Star Join Schema and is optimized for querying large data sets.

In the following example, the fact table is at the center which contains keys to every dimension table like Dealer_ID, Model ID, Date_ID, Product_ID, Branch_ID & other attributes like Units sold and revenue.



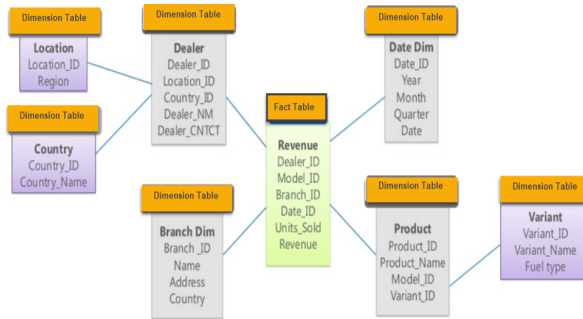
Characteristics of Star Schema:

- Every dimension in a star schema is represented with the only one-dimension table.
- The dimension table should contain the set of attributes.
- The dimension table is joined to the fact table using a foreign key.
- The dimension tables are not joined to each other.
- Fact table would contain key and measure.
- The Star schema is easy to understand and provides optimal disk usage.
- The dimension tables are not normalized. For instance, in the above figure, Country_ID does not have Country lookup table as an OLTP design would have.
- The schema is widely supported by BI Tools

2. Snowflake Schema

SNOWFLAKE SCHEMA is a logical arrangement of tables in a multidimensional database such that the ER diagram resembles a snowflake shape. A Snowflake Schema is an extension of a Star Schema, and it adds additional dimensions. The dimension tables are **normalized** which splits data into additional tables.

In the following example, Country is further normalized into an individual table.

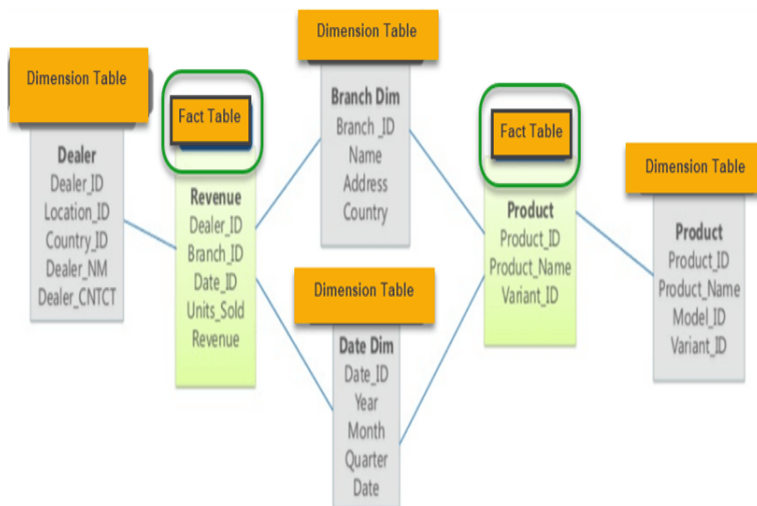


Characteristics of Snowflake Schema:

- The main benefit of the snowflake schema is that it uses smaller disk space.
- Easier to implement a dimension is added to the Schema.
- Due to multiple tables query performance is reduced.
- The primary challenge that you will face while using the snowflake Schema is that you need to perform more maintenance efforts because of the more lookup tables.

3. Fact Constellation Schema(Galaxy Schema)

A **GALAXY SCHEMA** contains two fact tables that share dimension tables between them. It is also called Fact Constellation Schema. The schema is viewed as a collection of stars hence the name Galaxy Schema.



As you can see in the above example, there are two facts table.

1. Revenue
2. Product.

In Galaxy schema shares dimensions are called **Conformed Dimensions**.

Characteristics of Galaxy Schema:

- The dimensions in this schema are separated into separate dimensions based on the various levels of hierarchy.
- For example, if geography has four levels of hierarchy like region, country, state, and city then Galaxy schema should have four dimensions.
- Moreover, it is possible to build this type of schema by splitting the one-star schema into more Star schemes.
- The dimensions are large in this schema which is needed to build based on the levels of hierarchy.
- This schema is helpful for aggregating fact tables for better understanding.

Q8 Explain 3 types of data warehouse architecture?

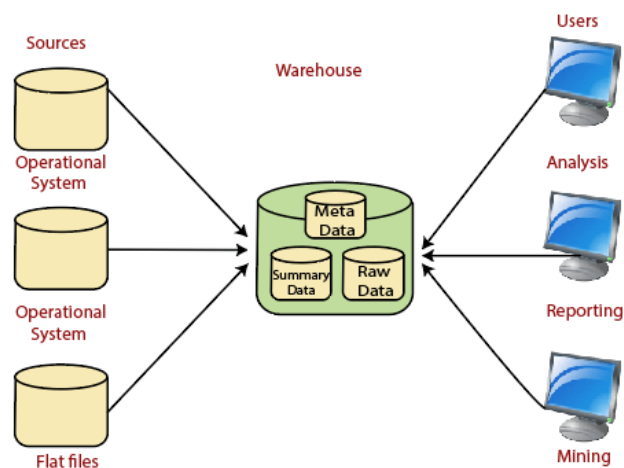
Ans. A data warehouse architecture is a method of defining the overall architecture of data communication processing and presentation that exist for end-clients computing within the enterprise. Each data warehouse is different, but all are characterized by standard vital components.

Data warehouses and their architecture depend upon the elements of an organization's situation.

Three common architectures are:

- Data Warehouse Architecture: Basic
- Data Warehouse Architecture: With Staging Area
- Data Warehouse Architecture: With Staging Area and Data Mart

Architecture of a Data Warehouse



- **Operational System**

An **operational system** is a method used in data warehousing to refer to a **system** that is used to process the day-to-day transactions of an organization.

- **Flat Files**

A **Flat file** system is a system of files in which transactional data is stored, and every file in the system must have a different name.

- **Meta Data**

A set of data that defines and gives information about other data.

Meta Data used in Data Warehouse for a variety of purpose, including:

Meta Data summarizes necessary information about data, which can make finding and working with instances of data more accessible. For example, author, data build, and data changed, and file size are examples of very basic document metadata.

Metadata is used to direct a query to the most appropriate data source.

- **Lightly and highly summarized data**

The area of the data warehouse saves all the predefined lightly and highly summarized (aggregated) data generated by the warehouse manager.

The goal of the summarized information is to speed up query performance. The summarized record is updated continuously as new information is loaded into the warehouse.

- **End-User access Tools**

The principal purpose of a data warehouse is to provide information to the business managers for strategic decision-making. These customers interact with the warehouse using end-client access tools.

The examples of some of the end-user access tools can be:

- Reporting and Query Tools
- Application Development Tools
- Executive Information Systems Tools
- Online Analytical Processing Tools
- Data Mining Tools

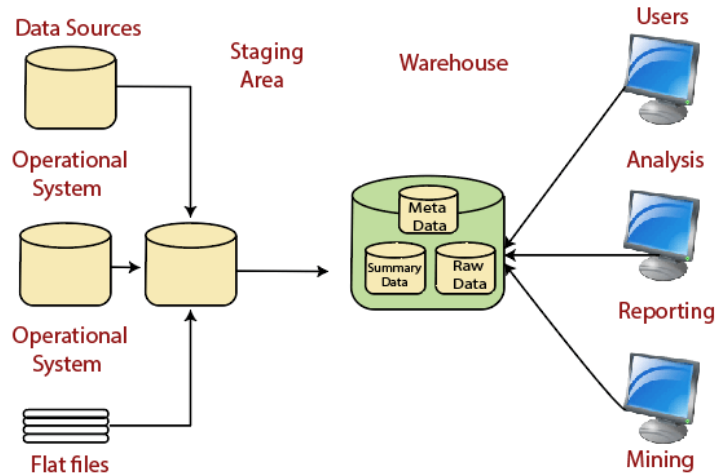
Data Warehouse Architecture: With Staging Area

We must clean and process your operational information before putting it into the warehouse.

We can do this programmatically, although data warehouses use a **staging area** (A place where data is processed before entering the warehouse).

A staging area simplifies data cleansing and consolidation for operational method coming from multiple source systems, especially for enterprise data warehouses where all relevant data of an enterprise is consolidated.

Architecture of a Data Warehouse with a Staging Area



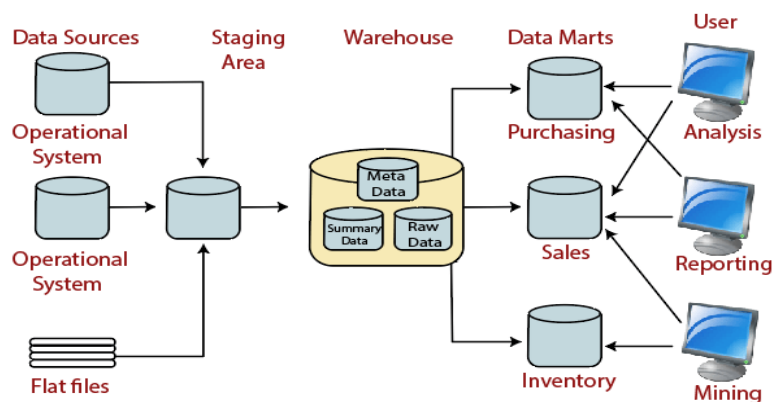
Data Warehouse Architecture: With Staging Area and Data Marts

We may want to customize our warehouse's architecture for multiple groups within our organization.

We can do this by adding **data marts**. A data mart is a segment of a data warehouse that can provide information for reporting and analysis on a section, unit, department, or operation in the company, e.g., sales, payroll, production, etc.

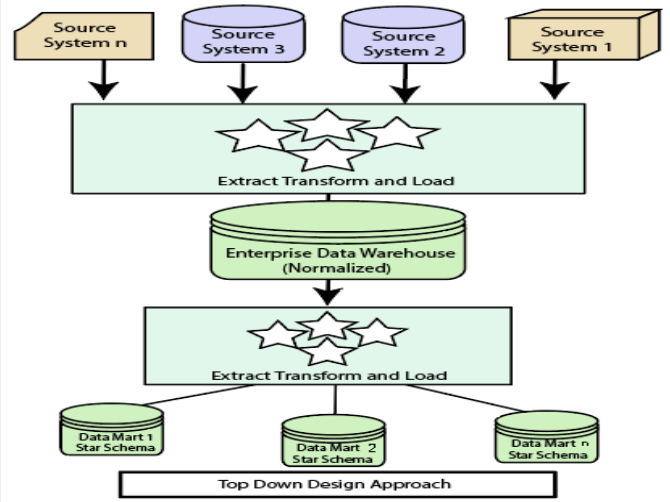
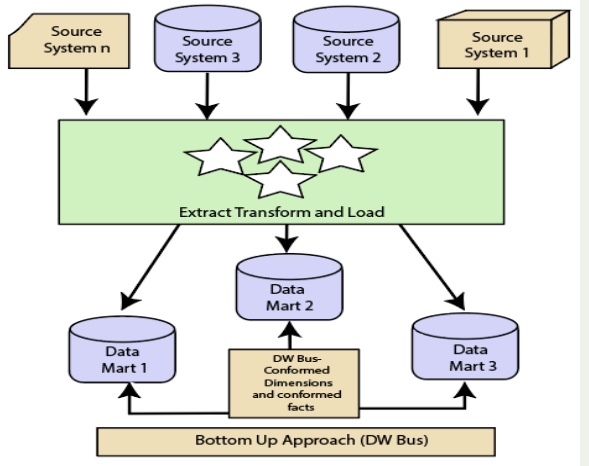
The figure illustrates an example where purchasing, sales, and stocks are separated. In this example, a financial analyst wants to analyze historical data for purchases and sales or mine historical information to make predictions about customer behavior.

Architecture of a Data Warehouse with a Staging Area and Data Marts



Q9 Difference between Top-Down and Bottom-Up approach.

Ans.

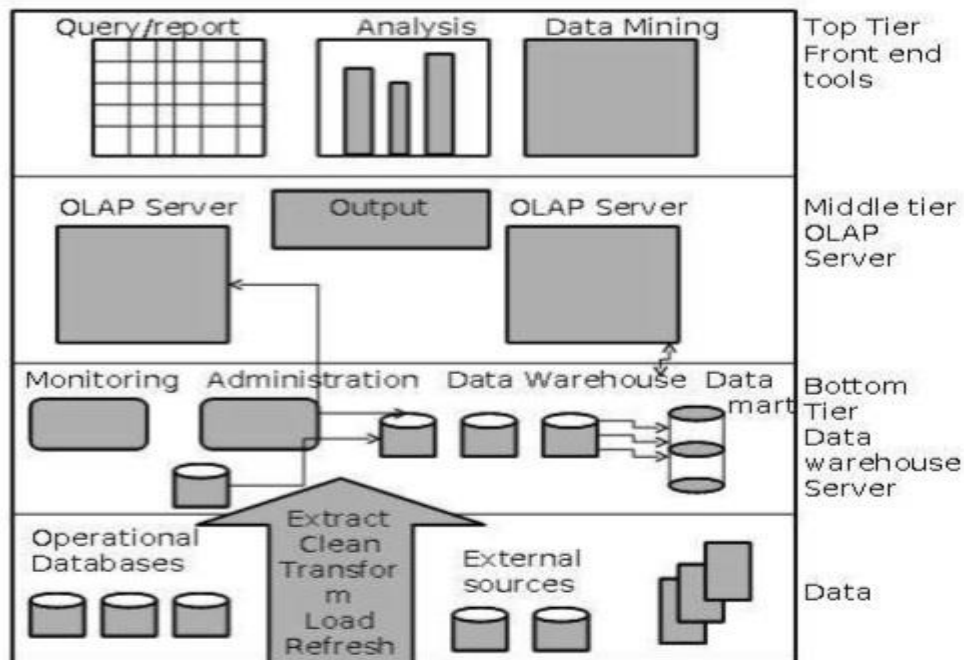
Top-Down Design Approach	Bottom-Up Design Approach
Breaks the vast problem into smaller subproblems.	Solves the essential low-level problem and integrates them into a higher one.
Inherently architected- not a union of several data marts.	Inherently incremental; can schedule essential data marts first.
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.
 <p>Top Down Design Approach</p>	 <p>Bottom Up Design Approach (DW Bus)</p>

Q10 what are the 3 tiers of data warehouse architecture?

Ans. Generally data warehouses adopt a three-tier architecture. Following are the three tiers of the data warehouse architecture.

- **Bottom Tier** – The bottom tier of the architecture is the data warehouse database server. It is the relational database system. We use the back-end tools and utilities to feed data into the bottom tier. These back-end tools and utilities perform the Extract, Clean, Load, and refresh functions.
- **Middle Tier** – In the middle tier, we have the OLAP Server that can be implemented in either of the following ways.
 - By Relational OLAP (ROLAP), which is an extended relational database management system. The ROLAP maps the operations on multidimensional data to standard relational operations.
 - By Multidimensional OLAP (MOLAP) model, which directly implements the multidimensional data and operations.
- **Top-Tier** – This tier is the front-end client layer. This layer holds the query tools and reporting tools, analysis tools and data mining tools.

The following diagram depicts the three-tier architecture of data warehouse.



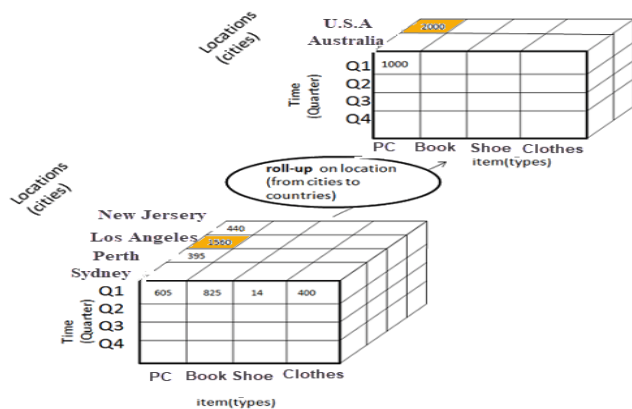
Q11 Explain different OLAP operations.

Ans. 5 types of analytical operations in OLAP are:

1. Roll-up
2. Drill-down
3. Slice
4. dice
5. Pivot (rotate)

1) Roll-up-Roll-up is also known as "consolidation" or "aggregation." The Roll-up operation can be performed in 2 ways.

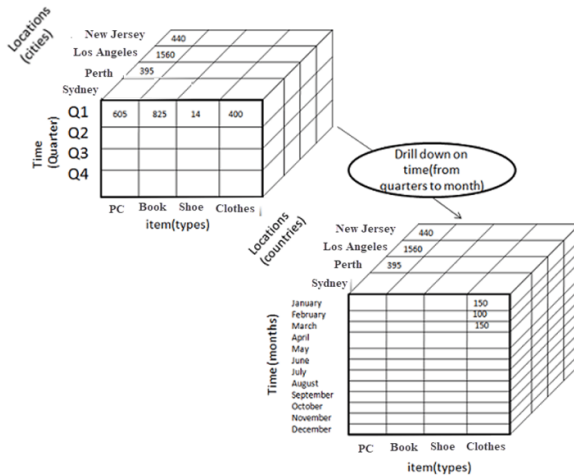
1. Reducing dimensions
2. Climbing up concept hierarchy. Concept hierarchy is a system of grouping things based on their order or level.



- In this example, cities New Jersey and Los Angeles are rolled up into country USA.
- The sales figures of New Jersey and Los Angeles are 440 and 1560 respectively. They become 2000 after roll-up.
- In this aggregation process, data location hierarchy moves up from city to the country.
- In the roll-up process at least one or more dimensions need to be removed. In this example, Quarter dimension is removed.

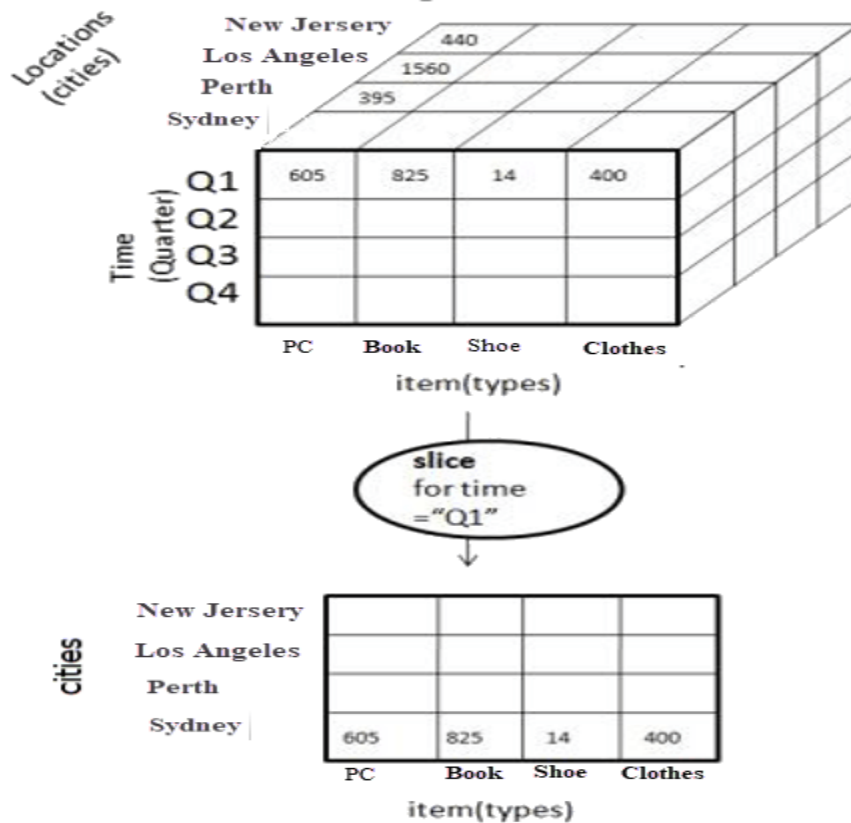
2) Drill-down-In drill-down data is fragmented into smaller parts. It is the opposite of the roll-up process. It can be done via

- Moving down the concept hierarchy
- Increasing a dimension



- Quarter Q1 is drilled down to months January, February, and March. Corresponding sales are also registered.
- In this example, dimension months are added.

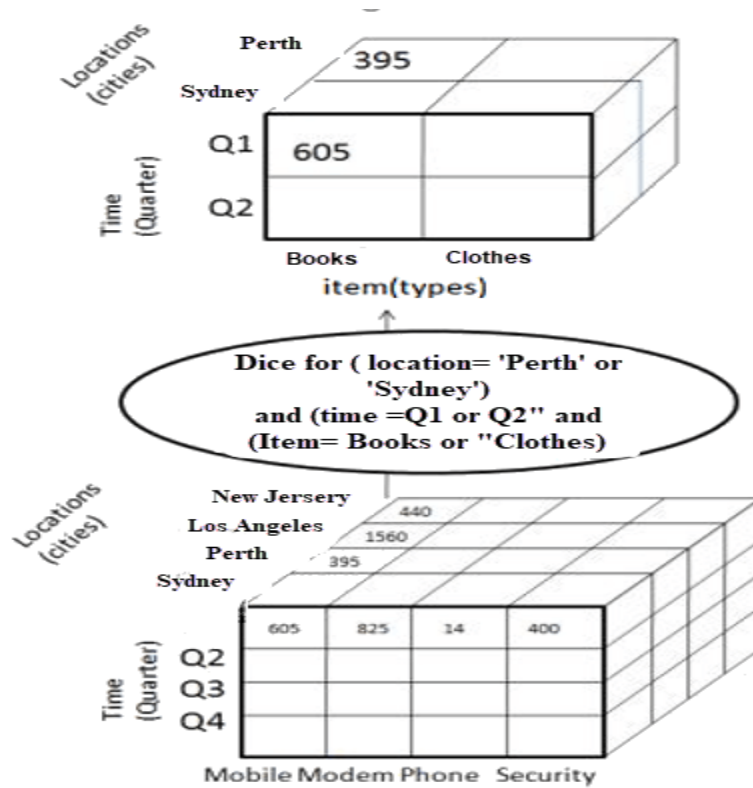
3) Slice-Here, one dimension is selected, and a new sub-cube is created.



- Dimension Time is Sliced with Q1 as the filter.
- A new cube is created altogether.

4) Dice:

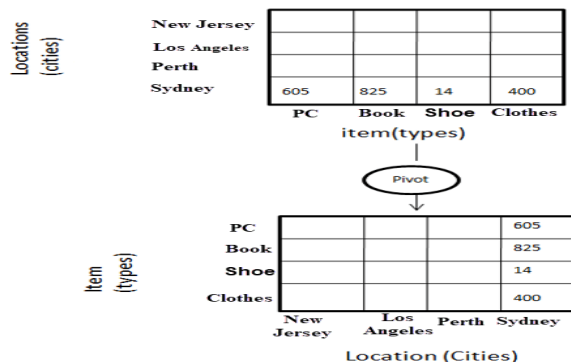
This operation is like a slice. The difference in dice is you select 2 or more dimensions that result in the creation of a sub-cube.



5) Pivot

In Pivot, you rotate the data axes to provide a substitute presentation of data.

In the following example, the pivot is based on item types.



Q12 Explain 2 indexing methods in OLAP.

Ans. In OLAP (Online Analytical Processing), bitmap indexing and join indexing are techniques used to optimize query performance in multidimensional databases. Let's explain both concepts with examples.

Bitmap Indexing:

Bitmap indexing is a technique that uses bitmap vectors to efficiently represent the relationships between values in a column (attribute) and the rows (data records) in a database. Each unique attribute value corresponds to a bitmap, where each bit represents the presence or absence of that value in the data records.

Example:

Suppose you have a sales database with the following dimensions:

- Product (P1, P2, P3)
- Region (R1, R2, R3)
- Time (T1, T2, T3)

Now, let's say you want to create a bitmap index for the "Product" dimension. It might look like this:

- Bitmap for Product P1: 101010
- Bitmap for Product P2: 010101
- Bitmap for Product P3: 110011

In these bitmaps, each row corresponds to a data record, and each bit represents whether a product was sold for that record. For example, in the first row, "101010" means that Product P1 was sold, but P2 and P3 were not sold for that record.

Bitmap indexing makes it fast and efficient to perform operations like filtering, counting, and aggregating data based on attributes. For instance, if you want to find all sales records for Product P1, you can simply perform a bitwise "AND" operation with the bitmap for P1, and it will give you the matching records.

Join Indexing:

Join indexing is a technique used to optimize queries involving multiple fact tables and dimensions by precomputing and storing the results of complex joins.

Example:

Consider a data warehouse with two fact tables: "Sales" and "Inventory." Both tables have a common dimension, "Product."

- Sales Fact Table: Contains sales information (e.g., product sales, date, store).
- Inventory Fact Table: Contains inventory data (e.g., product quantity, date, store).
- Product Dimension: Contains product details.

If you want to find the total sales and total inventory quantity for each product, you will typically join the "Sales" and "Inventory" fact tables using the "Product" dimension. However, this join operation can be resource intensive.

With join indexing, you precompute and store the results of this join. So, when you run a query like "total sales and inventory quantity for each product," you can retrieve the results directly from the join index, avoiding the need to perform the join operation on-the-fly.

Join indexing helps improve query performance significantly when dealing with complex multidimensional datasets involving multiple fact tables and dimensions.

In summary, both bitmap indexing and join indexing are valuable techniques in OLAP for optimizing query performance in multidimensional databases. Bitmap indexing is used for efficient filtering and aggregation based on attribute values, while join indexing is used for precomputing and storing the results of complex joins, especially in scenarios involving multiple fact tables and dimensions.

Q13 Explain OLAM with architecture.

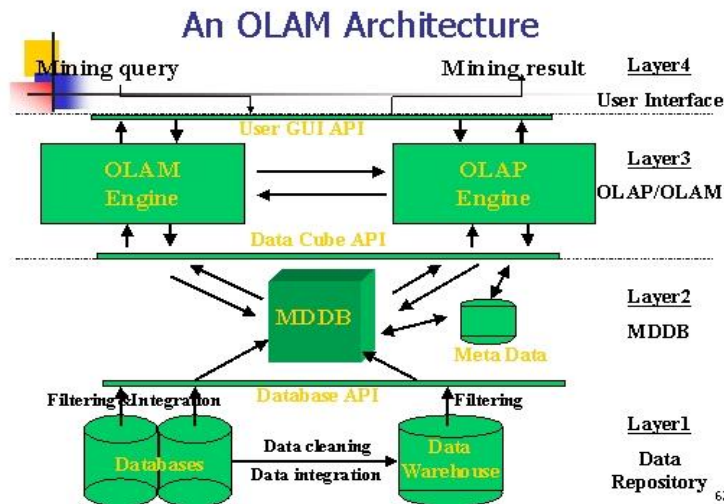
Ans. Online analytical mining integrates online analytical processing (OLAP) and data mining. It represents a promising direction for mining large databases and data warehouses.

Importance of OLAM

OLAM is important for the following reasons –

- **High quality of data in data warehouses** – The data mining tools are required to work on integrated, consistent, and cleaned data. These steps are very costly in the preprocessing of data. The data warehouses constructed by such preprocessing are valuable sources of high-quality data for OLAP and data mining as well.
- **Available information processing infrastructure surrounding data warehouses** – Information processing infrastructure refers to accessing, integration, consolidation, and transformation of multiple heterogeneous databases, web-accessing and service facilities, reporting and OLAP analysis tools.
- **OLAP-based exploratory data analysis** – Exploratory data analysis is required for effective data mining. OLAM provides a facility for data mining on various subsets of data and at different levels of abstraction.
- **Online selection of data mining functions** – Integrating OLAP with multiple data mining functions and online analytical mining provide users with the flexibility to select desired data mining functions and swap data mining tasks dynamically.

❖ OLAM Architecture



An OLAM engine performs analytical mining in data cubes in a similar manner to an OLAP engine performs online analytical processing. Therefore, it is suggested to have an integrated OLAM and OLAP architecture. Where the OLAM and OLAP engines both accept user's online queries via a user graphical user interface.

An OLAM engine can perform multiple data mining tasks, such as concept description, association, classification, prediction, clustering, and time series analysis. Therefore, it usually consists of multiple, integrated data mining modules, making it more sophisticated than an OLAP engine. There is no fundamental difference between the data cube required for OLAP, although OLAM analysis might require more powerful data cube construction and accessing tools.

Q14 Explain different methods of data cube computation.

Ans. Data cube computation is an important task in data warehouse implementation. The pre-computation of all or part of a data cube can greatly reduce the response time and enhance the performance of online analytical processing. However, such computation is challenging since it may require large computational time and storage space. This section explores efficient methods for data cube computation.

Multiway Array Aggregation for full cube computation

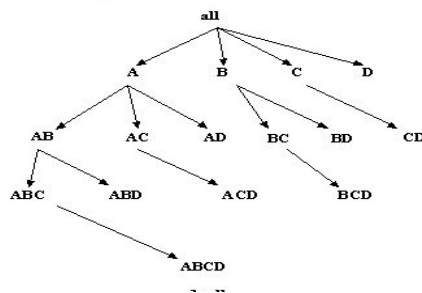
- The multiway array aggregation or simply multiway method computes a full data cube by using multidimensional array as its basic data structure.
- It is a typical MOLAP (Multidimensional online Analytical Processing) approach that uses direct array addressing, where dimension values are accessed via the position or index of their corresponding array locations.

A different approach is developed for the array-based cube construction, as follows:

1. **Partition the array into chunks:-** A chunk is a sub cube that is small enough to fit into the memory available for cube computation. chunking is a method for dividing an N-dimensional array into small N- dimensional chunks, where each chunk is stored as an object on disk. The chunks are compressed to remove wasted space resulting from empty array cells.
2. **Compute aggregates by visiting cube cells:-** The order in which cells are visited can be optimized to minimize the number of times that each cell must be revisited, thereby reducing memory access and storage costs.

BUC (Bottom-up Construction) : Computing Iceberg cubes from the apex cuboid downward

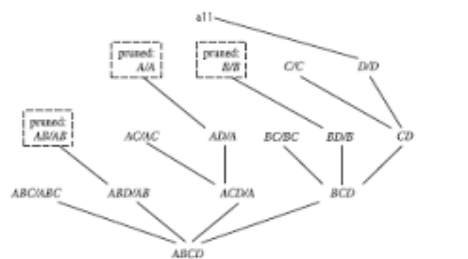
- BUC stands for bottom –up construction and is an algorithm for the computation of sparse and iceberg cubes.
- Unlike multiway ,BUC constructs the cube from the apex cuboid towards the base cuboid. This allows BUC to share data partitioning costs.



- This representation of a lattice of cuboids, with the apex at the top and the base at the bottom, is commonly accepted in data warehousing. It consolidated the notions of drill down and roll up.

Star cubing: computing iceberg cubes using a dynamic star tree structure.

- Star cubing integrates top down and bottom-up cube computation and explores both multidimensional aggregations.
- It operates from a data structure called a star tree, which performs lossless data compression, thereby reducing the computation time and memory requirements.
- A key idea behind star cubing is the concept of shared dimensions . To build up to this notion.
- The order of computation is from the base cuboid, upwards towards the apex cuboid . This order of computation is like that to Multiway.



Star-Cubing: Bottom-up computation with top-down expansion of shared dimensions.

Q15 What is Attribute Oriented Induction(AOI)?

Ans. The Attribute-Oriented Induction (AOI) approach to data generalization and summarization – based characterization was first proposed in 1989 (KDD '89 workshop) a few years before the introduction of the data cube approach.

The data cube approach can be considered as a data warehouse – based, pre computational – oriented, materialized approach.

It performs off-line aggregation before an OLAP or data mining query is submitted for processing.

On the other hand, the attribute oriented induction approach, at least in its initial proposal, a relational database query – oriented, generalized – based, on-line data analysis technique.

However, there is no inherent barrier distinguishing the two approaches based on online aggregation versus offline precomputation.

Some aggregations in the data cube can be computed on-line, while off-line precomputation of multidimensional space can speed up attribute-oriented induction as well.

It was proposed in 1989 (KDD '89 workshop).

It is not confined to categorical data nor particular measures.

Basic Principles of Attribute Oriented Induction

Data focusing:

- Analyzing task-relevant data, including dimensions, and the result is the initial relation.

Attribute-removal:

- To remove attribute A if there is a large set of distinct values for A but (1) there is no generalization operator on A, or (2) A's higher-level concepts are expressed in terms of other attributes.

Attribute-generalization:

- If there is a large set of distinct values for A, and there exists a set of generalization operators on A, then select an operator and generalize A.

Attribute-threshold control:

- Typical 2-8, specified/default.

Generalized relation threshold control (10-30):

- To control the final relation/rule size.

Algorithm for Attribute Oriented Induction

InitialRel-It is nothing but query processing of task-relevant data and deriving the initial relation.

PreGen-It is based on the analysis of the number of distinct values in each attribute and to determine the generalization plan for each attribute: removal? or how high to generalize?

PrimeGen-It is based on the PreGen plan and performing the generalization to the right level to derive a “prime generalized relation” and accumulating the counts.

Presentation-User interaction:

- (1) Adjust levels by drilling,
- (2) Pivoting,
- (3) Mapping into rules, cross tabs, visualization presentations