1. **Describe in detail Data warehouse architecture and ETL process.**

# **Ans.** Data Warehouse Architecture

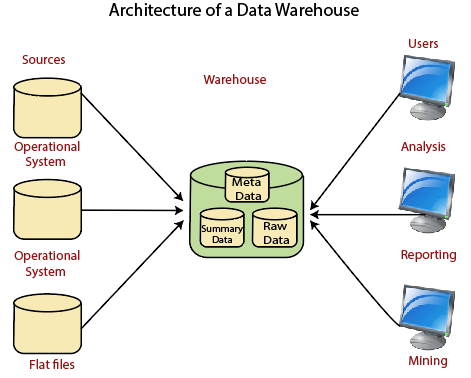
A data warehouse architecture is a method of defining the overall architecture of data communication processing and presentation that exist for end-clients computing within the enterprise. Each data warehouse is different, but all are characterized by standard vital components.

Production applications such as payroll accounts payable product purchasing and inventory control are designed for online transaction processing **(OLTP)**. Such applications gather detailed data from day to day operations.

Data Warehouse applications are designed to support the user ad-hoc data requirements, an activity recently dubbed online analytical processing (OLAP). These include applications such as forecasting, profiling, summary reporting, and trend analysis.

Production databases are updated continuously by either by hand or via OLTP applications. In contrast, a warehouse database is updated from operational systems periodically, usually during off-hours. As OLTP data accumulates in production databases, it is regularly extracted, filtered, and then loaded into a dedicated warehouse server that is accessible to users. As the warehouse is populated, it must be restructured tables de-normalized, data cleansed of errors and redundancies and new fields and keys added to reflect the needs to the user for sorting, combining, and summarizing data.

Data Warehouse Architecture:



**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 work with particular 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 goals of the summarized information are 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

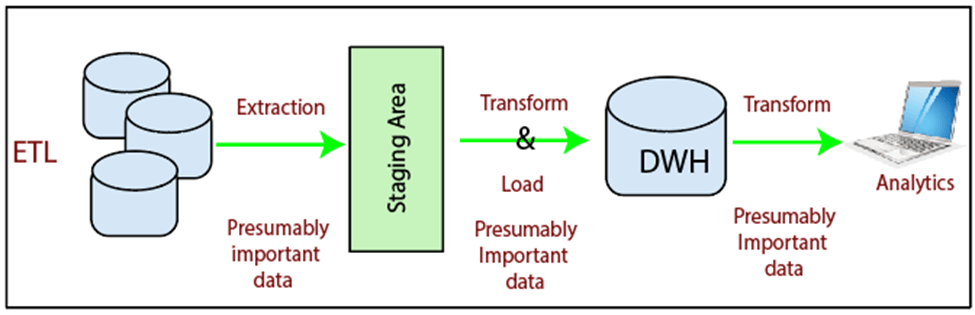
# **ETL (Extract, Transform, and Load) Process**

## **What is ETL?**

The mechanism of extracting information from source systems and bringing it into the data warehouse is commonly called **ETL**, which stands for **Extraction, Transformation and Loading**.

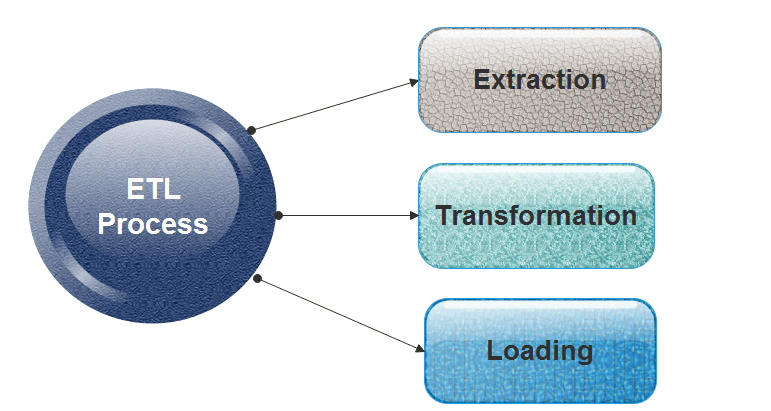
The ETL process requires active inputs from various stakeholders, including developers, analysts, testers, top executives and is technically challenging.

To maintain its value as a tool for decision-makers, Data warehouse technique needs to change with business changes. ETL is a recurring method (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.



## **How ETL Works?**

ETL consists of three separate phases:



### **Extraction**

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

### **Transformation**

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

The following points must be rectified in this phase:

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

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

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

### **Loading**

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

Loading can be carried in two ways:

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

**Ans.**

| **S.NO** | **ROLAP** | **MOLAP** |
| --- | --- | --- |
| 1. | ROLAP stands for **Relational Online Analytical Processing.** | While MOLAP stands for **Multidimensional Online Analytical Processing.** |
| 2. | ROLAP is used for large data volumes. | While it is used for limited data volumes. |
| 3. | The access of ROLAP is slow. | While the access of MOLAP is fast. |
| 4. | In ROLAP, Data is stored in relation tables. | While in MOLAP, Data is stored in multidimensional array. |
| 5. | In ROLAP, Data is fetched from data-warehouse. | While in MOLAP, Data is fetched from MDDBs database. |
| 6. | In ROLAP, Complicated sql queries are used. | While in MOLAP, Sparse matrix is used. |
| 7. | In ROLAP, Static multidimensional view of data is created. | While in MOLAP, Dynamic multidimensional view of data is created. |

**Difference between Operational Systems and Informational Systems :**

| **S.No** | **Operational Systems** | **Informational Systems** |
| --- | --- | --- |
| 1. | Operational systems are designed to deal with the running values of data. | Informational Systems deals with the collection, compilation and deriving information from data. |
| 2. | In operational systems, optimization of data structure is done for transactions. | In informational systems, optimization of data structure is done for complex queries. |
| 3. | Operational systems have response time of sub-seconds. | While informational systems have a response time of few seconds to minutes. |
| 4. | Operational systems are generally suited for small volumes of data. | Informational Systems are mainly designed for large volumes of data and hence convenient to use. |
| 5. | Operational systems are process oriented. | While informational systems are subject oriented. |
| 6. | Operational systems supports various data access operations such as read, update and delete. | Informational systems only supports read operation for data access. |

1. **What is classification? Explain in detail CART method.**

**Ans.** Classification is a machine learning task that involves assigning predefined labels or categories to input data based on its characteristics or features. The goal is to develop a model that can learn from existing labeled data (training data) and then make predictions or classifications on new, unseen data.

CART, which stands for Classification and Regression Trees, is a popular method used for classification tasks. It was introduced by Leo Breiman in 1986 and has since become widely adopted due to its simplicity and effectiveness. CART is a type of decision tree algorithm that recursively splits the data into subsets based on the values of input features. Let's break down the CART method in detail:

**1. Tree Construction:**

**a. Root Node:**

The algorithm starts with a root node that contains the entire dataset.

It selects the feature and a corresponding threshold that best separates the data into distinct classes. The selection is based on criteria such as Gini impurity or entropy.

**b. Splitting:**

The dataset is split into subsets based on the chosen feature and threshold.

This process is repeated recursively for each subset, creating a tree structure.

**c. Leaf Nodes:**

The process continues until a stopping condition is met, such as a predefined depth of the tree, a minimum number of samples in a node, or until further splitting does not significantly improve the purity of the subsets.

Each terminal node or leaf node represents a class label.

**2. Decision Criteria:**

CART uses one of the following criteria to evaluate the quality of a split:

**a. Gini Impurity:**

Gini impurity measures the degree of disorder or impurity in a set of data.

It ranges from 0 (pure node, all samples belong to one class) to 1 (impure node, samples are evenly distributed among classes).

**b. Entropy:**

Entropy is a measure of information disorder.

It ranges from 0 (pure node) to 1 (maximum disorder).

**3. Pruning:**

After the tree is constructed, pruning may be applied to avoid overfitting. Pruning involves removing nodes or branches that do not contribute significantly to the model's predictive accuracy on new, unseen data.

**4. Prediction:**

To classify a new instance, it traverses the decision tree from the root to a leaf, and the class associated with the leaf node becomes the predicted class for the input.

**Advantages of CART:**

**Interpretability:** Decision trees are easy to understand and interpret, making them useful for explaining the model to non-experts.

**Handling Non-linearity:** CART can capture complex relationships and non-linear decision boundaries in the data.

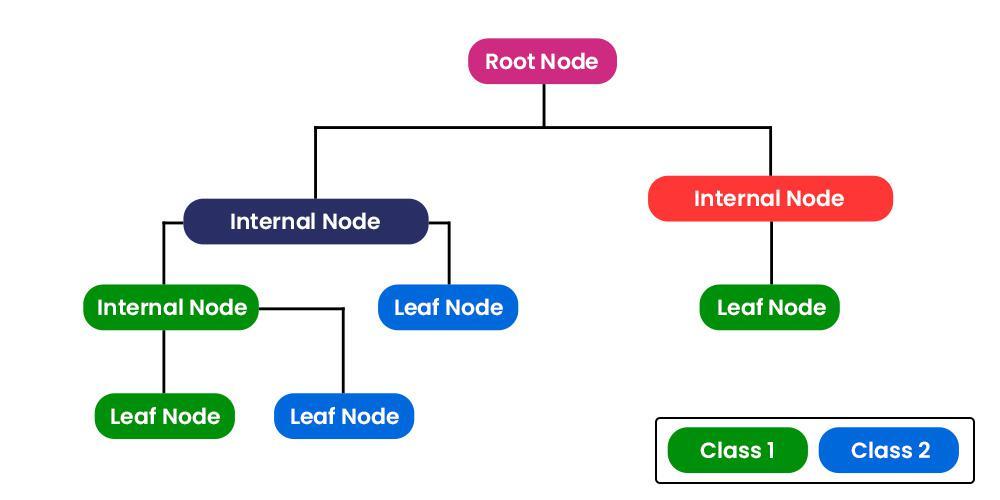
**Feature Importance:** It can provide insights into the importance of different features in making predictions.

**Limitations of CART:**

**Overfitting:** Decision trees can be prone to overfitting, especially if the tree is deep and complex.

**Sensitive to Noisy Data:** They can be sensitive to noise in the data.

**Bias towards Dominant Classes:** If a dataset is imbalanced, CART may have a bias towards the dominant classes.



1. **Discuss agglomerative and divisive hierarchical clustering using suitable example.**

**Ans.**

1. **Explain the concept of a data-cube as a multi-dimensional data model. What is the role of concept hierarchies in defining the dimensions of a data-cube? Illustrate, with an example.**

**Ans.**

1. **What is regression? How can linear regression be used for prediction?**

**Ans.**

1. **Explain text mining and discuss in brief the information retrieval methods.**

**Ans.**

1. **Write a short note on any two the following: a) KDD process**

**Ans.**

1. **Define data mining. State different data mining techniques. Explain KDD process in detail?**

**Ans.**

1. **Describe the major functional components of a data warehouse and illustrate its consequent multi-tier architecture with a diagram.**

**Ans.**

1. **Define classification. Explain decision tree with suitable example**

**Ans.**

1. **What is Market Basket Analysis? Illustrate with examples of its real-world applications**

**Ans.** Market Basket Analysis (MBA) is a data mining technique that examines customer purchasing behavior by analyzing the associations or relationships between different products/items in their shopping baskets. The primary objective is to discover patterns, associations, and dependencies in customer transactions to understand what products are frequently bought together. This analysis helps businesses make informed decisions about product placement, promotions, and inventory management. The most common metric used in Market Basket Analysis is the Association Rule, often expressed in terms of "If a customer buys X, then they are likely to buy Y."

**How Market Basket Analysis Works:**

1. **Frequent Itemset Generation:**
   * Identify sets of items that are frequently purchased together. These sets are known as frequent itemsets.
2. **Association Rule Generation:**
   * Based on frequent itemsets, generate association rules that describe the relationships between items.
3. **Rule Evaluation:**
   * Evaluate the generated rules using metrics such as support, confidence, and lift to determine the strength and significance of the associations.

**Real-World Applications of Market Basket Analysis:**

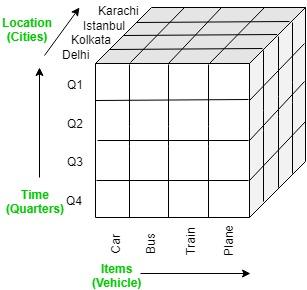
1. **Retail and E-Commerce:**
   * **Example:** In a grocery store, if customers frequently buy bread and butter together, the store might place these items close to each other or run promotions to encourage joint purchases.
2. **Online Recommendations:**
   * **Example:** E-commerce websites like Amazon use Market Basket Analysis to recommend products to customers based on their browsing and purchase history. For instance, if customers who bought a camera also frequently bought camera accessories, the website may suggest those accessories to the current shopper.
3. **Cross-Selling and Upselling:**
   * **Example:** In a fast-food restaurant, if customers often order a burger, the staff might ask if they want to add fries or a drink (cross-selling). Alternatively, if customers order a regular-sized item, the staff might suggest upgrading to a larger size (upselling).
4. **Inventory Management:**
   * **Example:** A bookstore might use Market Basket Analysis to optimize its inventory by ensuring that books frequently bought together are stocked on the same shelves or are part of the same promotions.
5. **Fraud Detection:**
   * **Example:** In the banking industry, Market Basket Analysis can be used to detect unusual patterns of transactions. For instance, if a customer typically makes small purchases but suddenly buys expensive items, it might raise a red flag for potential fraudulent activity.
6. **Healthcare:**
   * **Example:** In a pharmacy, if customers frequently purchase certain over-the-counter medications together, the store can use this information to optimize product placement and promotions.
7. **Telecommunications:**
   * **Example:** Market Basket Analysis can be applied to analyze the combinations of services that customers subscribe to (e.g., cable TV, internet, and phone services) to tailor bundled offerings.
8. **Supply Chain Optimization:**
   * **Example:** Manufacturers and distributors can use Market Basket Analysis to optimize the packaging and shipping of products that are often ordered together, reducing shipping costs and improving efficiency.
9. **Beer and Diapers Phenomenon:**
   * **Example:** One classic example involves a supermarket chain discovering a correlation between the purchase of diapers and beer. This insight led to strategic placement, such as positioning beer near the diaper aisle, as it was found that young fathers often bought both items on their way home.

Market Basket Analysis is a valuable tool in various industries, helping businesses understand customer behavior, optimize their operations, and enhance the overall customer experience. The insights gained from this analysis can lead to more effective marketing strategies and improved decision-making.

1. **Illustrate with an example OLAP operations: Roll-Up, Drill-Down, Slice, Dice.**

# **Ans. OLAP Operations in DBMS**

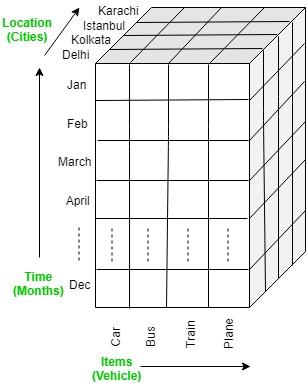
**OLAP** stands for ***Online Analytical Processing*** Server. It is a software technology that allows users to analyze information from multiple database systems at the same time. It is based on multidimensional data model and allows the user to query on multi-dimensional data (eg. Delhi -> 2018 -> Sales data). OLAP databases are divided into one or more cubes and these cubes are known as *Hyper-cubes*.



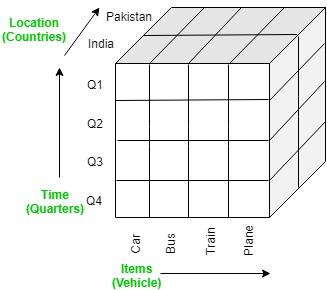
**OLAP operations:**

There are five basic analytical operations that can be performed on an OLAP cube:

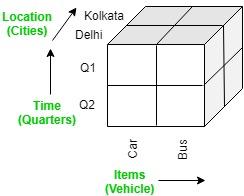
1. **Drill down:**In drill-down operation, the less detailed data is converted into highly detailed data. It can be done by:
   * Moving down in the concept hierarchy
   * Adding a new dimension

In the cube given in overview section, the drill down operation is performed by moving down in the concept hierarchy of *Time*dimension (Quarter -> Month).  


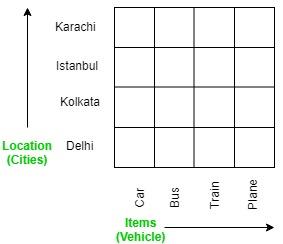
1. **Roll up:**It is just opposite of the drill-down operation. It performs aggregation on the OLAP cube. It can be done by:
   * Climbing up in the concept hierarchy
   * Reducing the dimensions

In the cube given in the overview section, the roll-up operation is performed by climbing up in the concept hierarchy of *Location*dimension (City -> Country).  


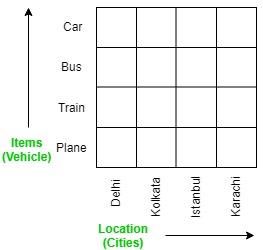
1. **Dice:**It selects a sub-cube from the OLAP cube by selecting two or more dimensions. In the cube given in the overview section, a sub-cube is selected by selecting following dimensions with criteria:
   * Location = “Delhi” or “Kolkata”
   * Time = “Q1” or “Q2”
   * Item = “Car” or “Bus”



1. **Slice:**It selects a single dimension from the OLAP cube which results in a new sub-cube creation. In the cube given in the overview section, Slice is performed on the dimension Time = “Q1”.



1. **Pivot:**It is also known as *rotation* operation as it rotates the current view to get a new view of the representation. In the sub-cube obtained after the slice operation, performing pivot operation gives a new view of it.

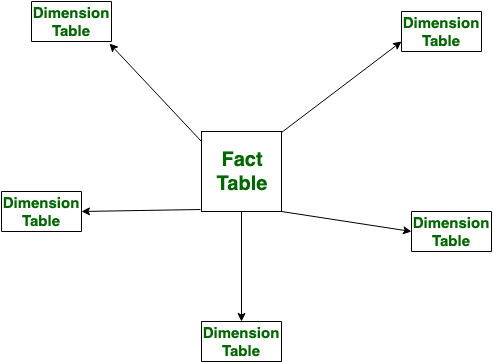


1. **Write a short note on the following: a. Data Mart b. ETL Process c. Abstract Data Types d. Regression e. Web Mining f. Decision Tree.**

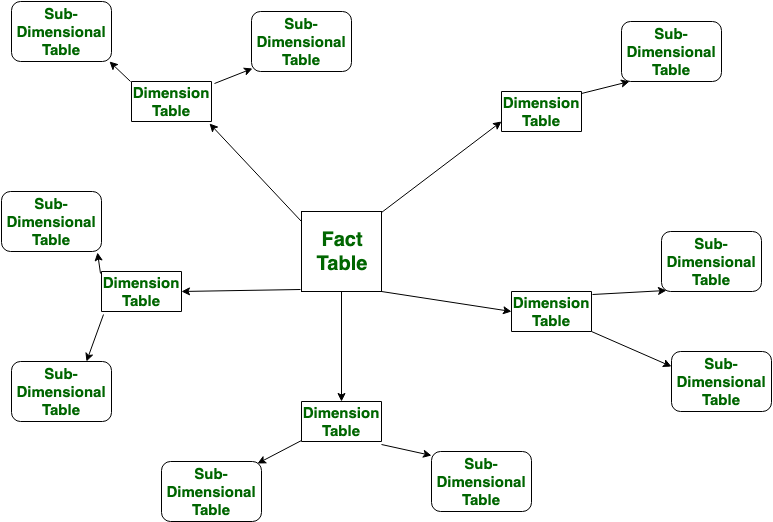
**Ans.**

1. Discuss Star and snow flake schema with a suitable example.

**Ans. Star Schema:** Star schema is the type of multidimensional model which is used for data warehouse. In star schema, The fact tables and the dimension tables are contained. In this schema fewer foreign-key join is used. This schema forms a star with fact table and dimension tables.



**Snowflake Schema:** Snowflake Schema is also the type of multidimensional model which is used for [data warehouse](https://www.geeksforgeeks.org/data-warehousing/). In snowflake schema, The fact tables, dimension tables as well as sub dimension tables are contained. This schema forms a snowflake with fact tables, dimension tables as well as sub-dimension tables.

Let’s see the difference between Star and Snowflake Schema:

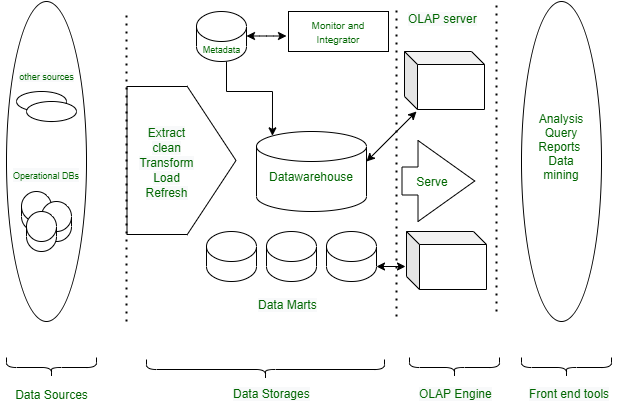
| **S.NO** | **Star Schema** | **Snowflake Schema** |
| --- | --- | --- |
| 1. | In [star schema](https://www.geeksforgeeks.org/star-schema-in-data-warehouse-modeling/), The fact tables and the dimension tables are contained. | While in [snowflake schema](https://www.geeksforgeeks.org/snowflake-schema-in-data-warehouse-model/), The fact tables, dimension tables as well as sub dimension tables are contained. |
| 2. | Star schema is a top-down model. | While it is a bottom-up model. |
| 3. | Star schema uses more space. | While it uses less space. |
| 4. | It takes less time for the execution of queries. | While it takes more time than star schema for the execution of queries. |
| 5. | In star schema, Normalization is not used. | While in this, Both normalization and denormalization are used. |
| 6. | It’s design is very simple. | While it’s design is complex. |
| 7. | The query complexity of star schema is low. | While the query complexity of snowflake schema is higher than star schema. |
| 8. | It’s understanding is very simple. | While it’s understanding is difficult. |
| 9. | It has less number of foreign keys. | While it has more number of foreign keys. |
| 10. | It has high data redundancy. | While it has low data redundancy. |

1. **Describe the major functional components of a data warehouse and illustrate its consequent multitier architecture with a diagram.**

**Ans.** A [data warehouse](https://www.geeksforgeeks.org/data-warehousing/) is Representable by data integration from multiple heterogeneous sources. It was defined by **Bill Inmon**in 1990. The data warehouse is an integrated, subject-oriented, time-variant, and non-volatile collection of data. A Data Warehouse is structured by data integration from multiple heterogeneous sources. It is a system used for data analysis and reporting. A data warehouse is deliberate a core factor of business intelligence. BI technology provides a historical, current, and predictive view of business operations without data mining many businesses may not be able to perform effective market analysis, the strength and weakness of their competitors, profitable decisions, etc.

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

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



*Three/Multi-tier Architecture of Data Warehouse*

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

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

#### **Middle Tier :**

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

OLAP server models come in three different categories, including:

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

#### **Top Tier :**

The top tier is a front-end client layer, which includes query and reporting tools, analysis tools, and/or data mining tools (eg, trend analysis, prediction, etc.).

Here are a few Top Tier tools that are often used:  
   **I.** SAP BW  
    **II.**SAS Business Intelligence  
  **III.** IBM Cognos  
   **IV.**Crystal Reports  
    **V.**Microsoft BI Platform

#### **Data Warehouse Models :**

From the architecture point of view, there are three warehouse models-

**Enterprise Warehouse:-**

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

**Data Mart:-**

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

**Virtual Warehouse:-**

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

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

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

**Ans.**   
Data marts are subsets of data warehouses that are designed to support the reporting and analytical needs of specific business units, departments, or user groups within an organization. They are smaller, more focused databases that contain relevant data for a particular set of users. Here are some characteristics and benefits of data marts:

**Characteristics of Data Marts:**

1. **Subject-Oriented:**
   * Data marts are organized around specific business subjects or functional areas, such as sales, marketing, finance, or human resources.
   * They provide a tailored view of data related to a particular domain.
2. **Subset of Data Warehouse:**
   * Data marts are subsets of larger data warehouses. While a data warehouse serves the entire organization, data marts are more targeted and cater to the needs of specific departments or business units.
3. **User-Focused:**
   * Data marts are designed to meet the specific reporting and analytical requirements of a particular group of users.
   * They often contain pre-aggregated and summarized data, making it easier for users to derive insights without the need for complex queries.
4. **Dimensional Modeling:**
   * They often use dimensional modeling techniques, such as star or snowflake schemas, which facilitate easy navigation and querying for end-users.
5. **Data Quality:**
   * Due to their focused nature, data marts can maintain higher data quality, as the data is curated and refined to meet the specific needs of the user group.
6. **Fast Query Performance:**
   * Data marts are optimized for quick query performance. Since they contain a subset of data, queries can be executed more rapidly compared to querying the entire data warehouse.
7. **Independent Development:**
   * Different data marts can be developed independently, allowing individual business units to create and manage their own data mart without affecting others.

**Benefits of Data Marts:**

1. **Improved Performance:**
   * By focusing on a specific business area, data marts can provide faster query performance, as users are working with a smaller and more relevant dataset.
2. **Increased Accessibility:**
   * Data marts make data more accessible to end-users, as they contain information specifically tailored to the needs of a particular group or department.
3. **Customization:**
   * Each data mart can be customized to cater to the unique requirements of its users, allowing for a more personalized and effective analytical experience.
4. **Cost Efficiency:**
   * Developing and maintaining smaller data marts can be more cost-effective than managing an entire enterprise-wide data warehouse, especially when considering hardware, software, and personnel costs.
5. **Easier Implementation:**
   * Implementing a data mart is often less complex than deploying a full-scale data warehouse. This can lead to quicker development cycles and faster time-to-value.
6. **Business Agility:**
   * Data marts support the agility of individual business units, allowing them to adapt quickly to changing requirements and make data-driven decisions more effectively.
7. **Reduced Data Redundancy:**
   * Since data marts are specific to certain business functions, there is a reduction in redundant data, leading to more efficient storage and maintenance.
8. **Explain the Page Ranking Algorithm with respect to web mining.**

**Ans.** The PageRank algorithm is a crucial component of web mining and is used by search engines to rank web pages based on their importance or relevance to a given query. Developed by Larry Page and Sergey Brin at Google, PageRank assigns a numerical weighting to each element of a hyperlinked set of web pages, with the purpose of measuring the relative importance of the pages within the network. The algorithm assumes that more important pages are likely to receive more links from other pages.

Here's a simplified explanation of the PageRank algorithm in the context of web mining:

**1. Link Analysis:**

* PageRank is based on the concept of link analysis, which considers the structure of the web in terms of hyperlinks between pages.
* Each hyperlink from one page to another is treated as a vote or endorsement for the linked page.

**2. Random Surfer Model:**

* PageRank envisions a "random surfer" navigating the web by clicking on links at random.
* The surfer has an equal probability of clicking on any link on a given page, including links leading to external pages or within the same website.

**3. Ranking Calculation:**

* The PageRank of a page is calculated iteratively. Initially, each page is assigned an equal probability (PageRank value).
* In subsequent iterations, the PageRank value of each page is updated based on the incoming links and their associated PageRank values.

**4. Damping Factor:**

* To model the behavior of a random surfer potentially navigating away from the current page, a damping factor (usually denoted as �*d*) is introduced.
* The damping factor typically has a value of 0.85. It represents the probability that the surfer will continue clicking on links rather than jumping to a random page.

**5. Link Weighting:**

* The PageRank value passed from one page to another is divided by the number of outgoing links on the source page. This helps to distribute the PageRank value evenly among the linked pages.
* A link from a page with higher PageRank is more valuable than a link from a page with lower PageRank.

**6. Convergence:**

* The iterative process continues until the PageRank values converge to stable values. The algorithm repeats the calculation until the changes in PageRank values become negligible.

**Example:**

Consider three web pages (A, B, and C) with initial PageRank values of 1.0. If Page A has links to pages B and C, the updated PageRank values might look like this:

* Initial: A(1.0), B(1.0), C(1.0)
* Iteration 1: A(0.15), B(1.0), C(0.85)
* Iteration 2: A(0.25), B(0.75), C(1.0)
* ...

After several iterations, the PageRank values stabilize, providing a ranking of the pages based on their importance within the web graph.

**Benefits of PageRank in Web Mining:**

1. **Relevance Ranking:**
   * PageRank helps search engines rank pages based on their relevance to user queries, improving the overall quality of search results.
2. **Combatting Spam:**
   * The algorithm's link analysis discourages manipulative practices like link farms and keyword stuffing, promoting more genuine and useful content.
3. **Prioritizing Important Pages:**
   * PageRank prioritizes pages that are linked to by other important pages, helping users discover high-quality content more easily.
4. **Adaptability:**
   * The algorithm is adaptable to changes in the web structure, making it effective in dynamic and evolving environments.
5. **Write a short note on any two the following:**

**i) K-means clustering**

**ii) text mining**

**iii) OLAP operations**

# **Ans.** K-Means Clustering Algorithm

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

What is K-Means Algorithm?

K-Means Clustering is an [Unsupervised Learning algorithm](https://www.javatpoint.com/unsupervised-machine-learning), which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

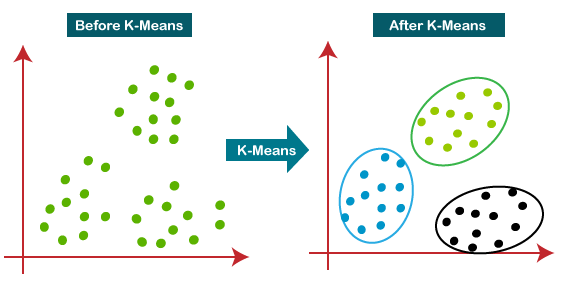
The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means [clustering](https://www.javatpoint.com/clustering-in-machine-learning) algorithm mainly performs two tasks:

* Determines the best value for K center points or centroids by an iterative process.
* Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:



How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

**Step-1:** Select the number K to decide the number of clusters.

**Step-2:** Select random K points or centroids. (It can be other from the input dataset).

**Step-3:** Assign each data point to their closest centroid, which will form the predefined K clusters.

**Step-4:** Calculate the variance and place a new centroid of each cluster.

**Step-5:** Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

**Step-6:** If any reassignment occurs, then go to step-4 else go to FINISH.

**Step-7**: The model is ready.

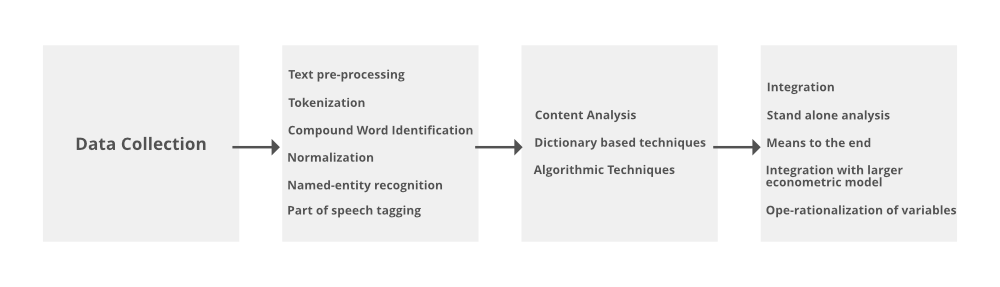
**What is Text Mining?**

Text mining is a component of [data mining](https://www.geeksforgeeks.org/data-mining/) that deals specifically with unstructured text data. It involves the use of natural language processing (NLP) techniques to extract useful information and insights from large amounts of unstructured text data. Text mining can be used as a preprocessing step for data mining or as a standalone process for specific tasks.

By using text mining, the unstructured text data can be transformed into structured data that can be used for data mining tasks such as classification, [clustering](https://www.geeksforgeeks.org/clustering-in-machine-learning/), and association rule mining. This allows organizations to gain insights from a wide range of data sources, such as customer feedback, social media posts, and news articles.

### **Conventional Process of Text Mining**

* Gathering unstructured information from various sources accessible in various document organizations, for example, plain text, web pages, PDF records, etc.
* Pre-processing and data cleansing tasks are performed to distinguish and eliminate inconsistency in the data. The data cleansing process makes sure to capture the genuine text, and it is performed to eliminate stop words [stemming](https://www.geeksforgeeks.org/introduction-to-stemming/) (the process of identifying the root of a certain word and indexing the data.
* Processing and controlling tasks are applied to review and further clean the data set.
* Pattern analysis is implemented in Management Information System.
* Information processed in the above steps is utilized to extract important and applicable data for a powerful and convenient decision-making process and trend analysis.



1. **Discuss issues to consider during data integration.**

**Ans. Issues in Data Integration:**

 There are several issues that can arise when integrating data from multiple sources, including:

1. **Data Quality:** Inconsistencies and errors in the data can make it difficult to combine and analyze.
2. **Data Semantics:** Different sources may use different terms or definitions for the same data, making it difficult to combine and understand the data.
3. **Data Heterogeneity:** Different sources may use different data formats, structures, or schemas, making it difficult to combine and analyze the data.
4. **Data Privacy and Security:** Protecting sensitive information and maintaining security can be difficult when integrating data from multiple sources.
5. **Scalability:** Integrating large amounts of data from multiple sources can be computationally expensive and time-consuming.
6. **Data Governance:** Managing and maintaining the integration of data from multiple sources can be difficult, especially when it comes to ensuring data accuracy, consistency, and timeliness.
7. **Performance:** Integrating data from multiple sources can also affect the performance of the system.
8. **Integration with existing systems:**Integrating new data sources with existing systems can be a complex task, requiring significant effort and resources.
9. **Complexity:** The complexity of integrating data from multiple sources can be high, requiring specialized skills and knowledge.

**There are three issues to consider during data integration: Schema Integration, Redundancy Detection, and resolution of data value conflicts. These are explained in brief below.**

**1. Schema Integration:**

* Integrate metadata from different sources.
* The real-world entities from multiple sources are referred to as the entity identification problem.ER

**2. Redundancy Detection:**

* An attribute may be redundant if it can be derived or obtained from another attribute or set of attributes.
* Inconsistencies in attributes can also cause redundancies in the resulting data set.
* Some redundancies can be detected by correlation analysis.

**3. Resolution of**data value**conflicts:**

* This is the third critical issue in data integration.
* Attribute values from different sources may differ for the same real-world entity.
* An attribute in one system may be recorded at a lower level of abstraction than the “same” attribute in another.

1. **What is classification? Explain in detail Associative Classification method.**

**Ans. How does Association Rule Learning work?**

Association rule learning is a type of unsupervised learning technique that checks for the dependency of one data item on another data item and maps accordingly so that it can be more profitable. It is based on different rules to discover the interesting relations between variables in the database. The association rule learning is one of the very important concepts of machine learning, and it is employed in Market Basket analysis, Web usage mining, continuous production, etc. Here market basket analysis is a technique used by the various big retailer to discover the associations between items.

Association rule learning works on the concept of If and Else Statement, such as if A then B.

Here the If element is called antecedent, and then statement is called as Consequent.

These types of relationships where we can find out some association or relation between two items is known as single cardinality. It is all about creating rules, and if the number of items increases, then cardinality also increases accordingly. So, to measure the associations between thousands of data items, there are several metrics. These metrics are given below:

* Support
* Confidence
* Lift

**1.Support :**

Support is the frequency of A or how frequently an item appears in the dataset. It is defined as the fraction of the transaction T that contains the itemset X. If there are X datasets, then for transactions T, it can be written as:

**Supp(X) = Freq(X) / T**

**2.Confidence:**

Confidence indicates how often the rule has been found to be true. Or how often the items X and Y occur together in the dataset when the occurrence of X is already given. It is the ratio of the transaction that contains X and Y to the number of records that contain X.

**Confidence = Freq(X,Y) / Freq(X)**

**3.Lift:**

It is the strength of any rule, which can be defined as below formula: It is the ratio of the observed support measure and expected support if X and Y are independent of each other. It has three possible values:

**Lift = Supp(X,Y) / Supp(X)\*Supp(Y)**

* If Lift= 1: The probability of occurrence of antecedent and consequent is independent of each other.
* Lift>1: It determines the degree to which the two itemsets are dependent to each other.
* Lift<1: It tells us that one item is a substitute for other items, which means one item has a negative effect on another.

**Types of Association Rule Learning:**

Association rule learning can be divided into three algorithms:

**1.Apriori Algorithm:**

This algorithm uses frequent datasets to generate association rules. It is designed to work on the databases that contain transactions. This algorithm uses a breadth-first search and Hash Tree to calculate the itemset efficiently. It is mainly used for market basket analysis and helps to understand the products that can be bought together. It can also be used in the healthcare field to find drug reactions for patients.

**2.Eclat Algorithm:**

Eclat algorithm stands for Equivalence Class Transformation. This algorithm uses a depthfirst search technique to find frequent itemsets in a transaction database. It performs faster execution than Apriori Algorithm.

**3.F-P Growth Algorithm:**

The F-P growth algorithm stands for Frequent Pattern, and it is the improved version of the Apriori Algorithm. It represents the database in the form of a tree structure that is known as a frequent pattern or tree. The purpose of this frequent tree is to extract the most frequent pattern.

**Applications of Association Rule Learning:**

It has various applications in machine learning and data mining. Below are some popular applications of association rule learning:

* **Market Basket Analysis:** It is one of the popular examples and applications of association rule mining. This technique is commonly used by big retailers to determine the association between items.
* **Medical Diagnosis:** With the help of association rules, patients can be cured easily, as it helps in identifying the probability of illness for a particular disease.
* **Protein Sequence:**The association rules help in determining the synthesis of artificial Proteins.
* It is also used for the **Catalog Design** and **Loss-leader Analysis** and many more other applications.

**Associative Classification in Data Mining:**

Bing Liu Et Al was the first to propose associative classification, in which he defined a model whose rule is “the right-hand side is constrained to be the attribute of the classification class”.An associative classifier is a supervised learning model that uses association rules to assign a target value.

The model generated by the association classifier and used to label new records consists of association rules that produce class labels. Therefore, they can also be thought of as a list of “if-then” clauses: if a record meets certain criteria (specified on the left side of the rule, also known as antecedents), it is marked (or scored) according to the rule’s category on the right. Most associative classifiers read the list of rules sequentially and apply the first matching rule to mark new records. Association classifier rules inherit some metrics from association rules, such as Support or Confidence, which can be used to rank or filter the rules in the model and evaluate their quality.

**Types of Associative Classification:**

There are different types of Associative Classification Methods, Some of them are given below.

**1. CBA (Classification Based on Associations):** It uses association rule techniques to classify data, which proves to be more accurate than traditional classification techniques. It has to face the sensitivity of the minimum support threshold. When a lower minimum support threshold is specified, a large number of rules are generated.

**2. CMAR (Classification based on Multiple Association Rules):**It uses an efficient FP-tree, which consumes less memory and space compared to Classification Based on Associations. The FP-tree will not always fit in the main memory, especially when the number of attributes is large.

**3. CPAR (Classification based on Predictive Association Rules):** Classification based on predictive association rules combines the advantages of association classification and traditional rule-based classification. Classification based on predictive association rules uses a greedy algorithm to generate rules directly from training data. Furthermore, classification based on predictive association rules generates and tests more rules than traditional rule-based classifiers to avoid missing important rules.

1. **What is data preprocessing? How it is performed? Why data preprocessing is important?**

**Ans.** Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.

**Some common steps in data preprocessing include:**

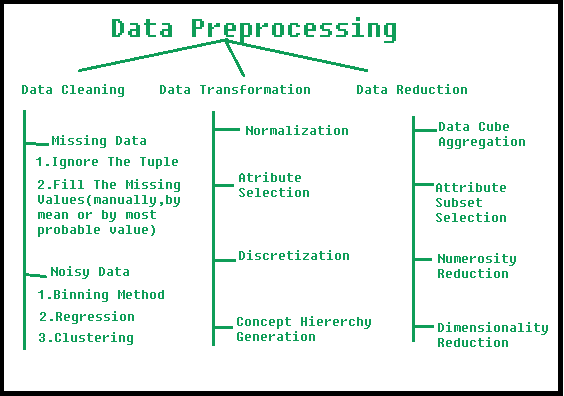
Data preprocessing is an important step in the data mining process that involves cleaning and transforming raw data to make it suitable for analysis. Some common steps in data preprocessing include:

* **Data Cleaning:**This involves identifying and correcting errors or inconsistencies in the data, such as missing values, outliers, and duplicates. Various techniques can be used for data cleaning, such as imputation, removal, and transformation.
* **Data Integration:**This involves combining data from multiple sources to create a unified dataset. Data integration can be challenging as it requires handling data with different formats, structures, and semantics. Techniques such as record linkage and data fusion can be used for data integration.
* **Data Transformation:**This involves converting the data into a suitable format for analysis. Common techniques used in data transformation include normalization, standardization, and discretization. Normalization is used to scale the data to a common range, while standardization is used to transform the data to have zero mean and unit variance. Discretization is used to convert continuous data into discrete categories.
* **Data Reduction:**This involves reducing the size of the dataset while preserving the important information. Data reduction can be achieved through techniques such as feature selection and feature extraction. Feature selection involves selecting a subset of relevant features from the dataset, while feature extraction involves transforming the data into a lower-dimensional space while preserving the important information.
* **Data Discretization:**This involves dividing continuous data into discrete categories or intervals. Discretization is often used in data mining and machine learning algorithms that require categorical data. Discretization can be achieved through techniques such as equal width binning, equal frequency binning, and clustering.
* **Data Normalization:**This involves scaling the data to a common range, such as between 0 and 1 or -1 and 1. Normalization is often used to handle data with different units and scales. Common normalization techniques include min-max normalization, z-score normalization, and decimal scaling.

Data preprocessing plays a crucial role in ensuring the quality of data and the accuracy of the analysis results. The specific steps involved in data preprocessing may vary depending on the nature of the data and the analysis goals.

By performing these steps, the data mining process becomes more efficient and the results become more accurate.

**Preprocessing in Data Mining:**   
Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.



**Steps Involved in Data Preprocessing:**

**1. Data Cleaning:**   
The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc. 

* **(a). Missing Data:**   
  This situation arises when some data is missing in the data. It can be handled in various ways.   
  Some of them are:
  1. **Ignore the tuples:**   
     This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.
  2. **Fill the Missing values:**   
     There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.
* **(b). Noisy Data:**   
  Noisy data is a meaningless data that can’t be interpreted by machines.It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways :
  1. **Binning Method:**   
     This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.
  2. **Regression:**   
     Here data can be made smooth by fitting it to a regression function.The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).
  3. **Clustering:**   
     This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

**2. Data Transformation:**   
This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

1. **Normalization:**   
   It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)
2. **Attribute Selection:**   
   In this strategy, new attributes are constructed from the given set of attributes to help the mining process.
3. **Discretization:**   
   This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.
4. **Concept Hierarchy Generation:**   
   Here attributes are converted from lower level to higher level in hierarchy. For Example-The attribute “city” can be converted to “country”.

**3. Data Reduction:**   
Data reduction is a crucial step in the data mining process that involves reducing the size of the dataset while preserving the important information. This is done to improve the efficiency of data analysis and to avoid overfitting of the model. Some common steps involved in data reduction are:

* **Feature Selection:** This involves selecting a subset of relevant features from the dataset. Feature selection is often performed to remove irrelevant or redundant features from the dataset. It can be done using various techniques such as correlation analysis, mutual information, and principal component analysis (PCA).
* **Feature Extraction:**This involves transforming the data into a lower-dimensional space while preserving the important information. Feature extraction is often used when the original features are high-dimensional and complex. It can be done using techniques such as PCA, linear discriminant analysis (LDA), and non-negative matrix factorization (NMF).
* **Sampling:**This involves selecting a subset of data points from the dataset. Sampling is often used to reduce the size of the dataset while preserving the important information. It can be done using techniques such as random sampling, stratified sampling, and systematic sampling.
* **Clustering:**This involves grouping similar data points together into clusters. Clustering is often used to reduce the size of the dataset by replacing similar data points with a representative centroid. It can be done using techniques such as k-means, hierarchical clustering, and density-based clustering.
* **Compression:** This involves compressing the dataset while preserving the important information. Compression is often used to reduce the size of the dataset for storage and transmission purposes. It can be done using techniques such as wavelet compression, JPEG compression, and gzip compression.

1. **What is clustering? Explain K-mean Clustering in detail.**

**Ans.**

1. **What is Distributed Database? Explain Architecture for Distributed database.**

## **Ans.** Distributed Database System:

A Distributed Database System is a kind of database that is present or divided in more than one location, which means it is not limited to any single computer system. It is divided over the network of various systems. The Distributed Database System is physically present on the different systems in different locations. This can be necessary when different users from all over the world need to access a specific database. For a user, it should be handled in such a way that it seems like a single database.

Parameters of Distributed Database Systems:

* **Distribution:**

It describes how data is physically distributed among the several sites.

* **Autonomy:**

It reveals the division of power inside the Database System and the degree of autonomy enjoyed by each individual DBMS.

* **Heterogeneity:**

It speaks of the similarity or differences between the databases, system parts, and data models.

Common Architecture Models of Distributed Database Systems:

* **Client-Server Architecture of DDBMS:**

This architecture is two level architecture where clients and servers are the points or levels where the main functionality is divided. There is various functionality provided by the server, like managing the transaction, managing the data, processing the queries, and optimization.

* **Peer-to-peer Architecture of DDBMS:**

In this architecture, each node or peer is considered as a server as well as a client, and it performs its database services as both (server and client). The peers coordinate their efforts and share their resources with one another.

* **Multi DBMS Architecture of DDBMS:**

This is an amalgam of two or more independent Database Systems that functions as a single integrated Database System.

Types of Distributed Database Systems:

* **Homogeneous Database System:**

Each site stores the same database in a Homogenous Database. Since each site has the same database stored, so all the data management schemes, operating system, and data structures will be the same across all sites. They are, therefore, simple to handle.

* **Heterogeneous Database System:**

In this type of Database System, different sites are used to store the data and relational tables, which makes it difficult for database administrators to do the transactions and run the queries into the database. Additionally, one site might not even be aware of the existence of the other sites. Different operating systems and database applications may be used by various computers. Since each system has its own database model to store the data, therefore it is required there should be translation schemes to establish the connections between different sites to transfer the data.

Distributed Data Storage:

There are two methods by which we can store the data on different sites:

* **Replication:**

This method involves redundantly storing the full relationship at two or more locations. Since a complete database can be accessed from each site, it becomes a redundant database. Systems preserve copies of the data as a result of replication.

This has advantages because it makes more data accessible at many locations. Additionally, query requests can now be handled in parallel.

However, there are some drawbacks as well. Data must be updated frequently. Any changes performed at one site must be documented at every site where that relation is stored in order to avoid inconsistent results. There is a tonne of overhead here. Additionally, since concurrent access must now be monitored across several sites, concurrency control becomes far more complicated.

* **Fragmentation:**

According to this method, the relationships are divided (i.e., broken up into smaller pieces), and each fragment is stored at the many locations where it is needed. To ensure there is no data loss, the pieces must be created in a way that allows for the reconstruction of the original relation.

Since Fragmentation doesn't result in duplicate data, consistency is not a concern.

Ways of fragmentation:

* **Horizontal Fragmentation:**

In Horizontal Fragmentation, the relational table or schema is broken down into a group of one and more rows, and each row gets one fragment of the schema. It is also called **splitting by rows**.

* **Vertical Fragmentation:**

In this fragmentation, a relational table or schema is divided into some more schemas of smaller sizes. A common candidate key must be present in each fragment in order to guarantee a lossless join. This is also called **splitting by columns**.

Application of Distributed Database Systems:

* Multimedia apps use it.
* The manufacturing control system also makes use of it.
* Another application is by corporate management for the information system.
* It is used in hotel chains, military command systems, etc.

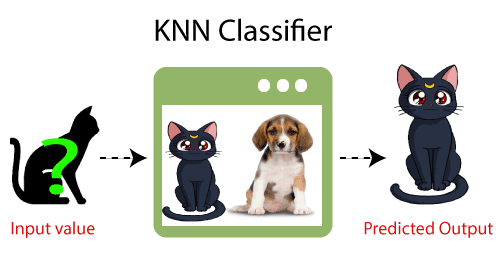
1. **Explain parallel database architecture in detail.**

**Ans.**

1. **Discuss K Nearest Neighbor algorithm with a suitable example**

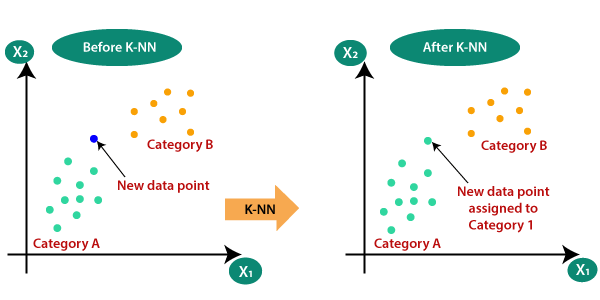
# **Ans.** K-Nearest Neighbor(KNN) Algorithm for Machine Learning

* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
* It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
* **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

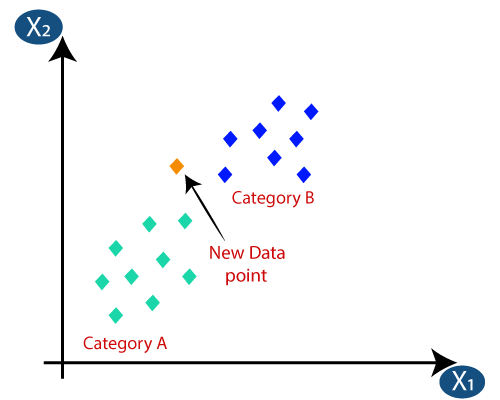


How does K-NN work?

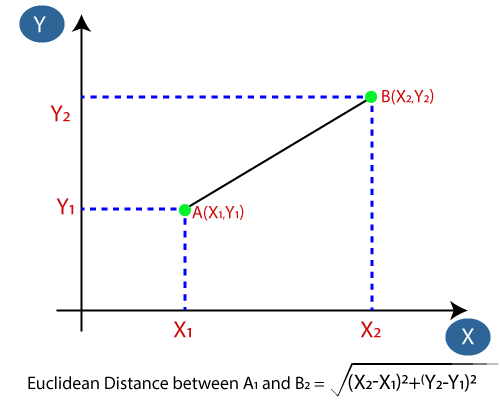
The K-NN working can be explained on the basis of the below algorithm:

* **Step-1:** Select the number K of the neighbors
* **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
* **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
* **Step-4:** Among these k neighbors, count the number of the data points in each category.
* **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
* **Step-6:** Our model is ready.

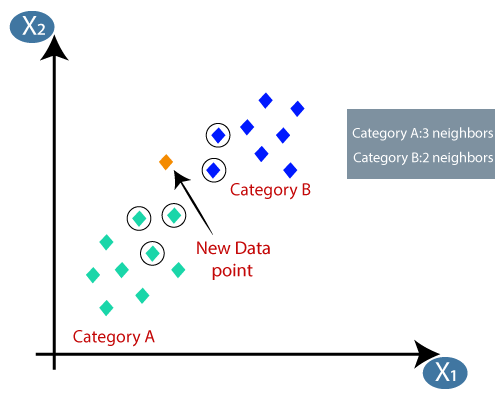
Suppose we have a new data point and we need to put it in the required category. Consider the below image:



* Firstly, we will choose the number of neighbors, so we will choose the k=5.
* Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



* By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



* As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

**How to select the value of K in the K-NN Algorithm?**

Below are some points to remember while selecting the value of K in the K-NN algorithm:

* There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
* A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
* Large values for K are good, but it may find some difficulties.

**Advantages of KNN Algorithm:**

* It is simple to implement.
* It is robust to the noisy training data
* It can be more effective if the training data is large.

**Disadvantages of KNN Algorithm:**

* Always needs to determine the value of K which may be complex some time.
* The computation cost is high because of calculating the distance between the data points for all the training samples.

1. **Explain the terms Entropy and Information Gain with an example.**

**Ans.**

1. **Define distributed database. Also explain the architecture of distributed database in detail.**

**Ans.**

1. **Define OLAP. Explain the different OLAP models with a suitable diagram.**

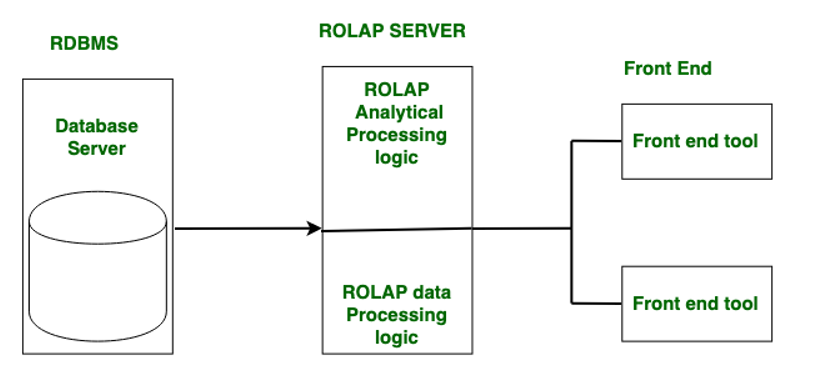
**Ans. OLAP**is considered **(Online Analytical Processing)**whichis a type of software that helps in analyzing information from multiple [databases](https://www.geeksforgeeks.org/types-of-databases/) at a particular time. OLAP is simply a [multidimensional data model](https://www.geeksforgeeks.org/multidimensional-data-model/)and also applies querying to it.

## **Types of OLAP Servers**

* Relational OLAP
* Multi-Dimensional OLAP
* Hybrid OLAP
* Transparent OLAP

### **Relational OLAP (ROLAP): Star Schema Based**

The[ROLAP](https://www.geeksforgeeks.org/difference-between-rolap-and-molap/)is based on the premise that data need not be stored multi-dimensionally to be viewed multi-dimensionally, and that it is possible to exploit the well-proven relational database technology to handle the 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 the [SQL](https://www.geeksforgeeks.org/what-is-sql/) statement. ROLAP can handle large amounts of data. ROLAP can leverage functionalities inherent in the relational database.

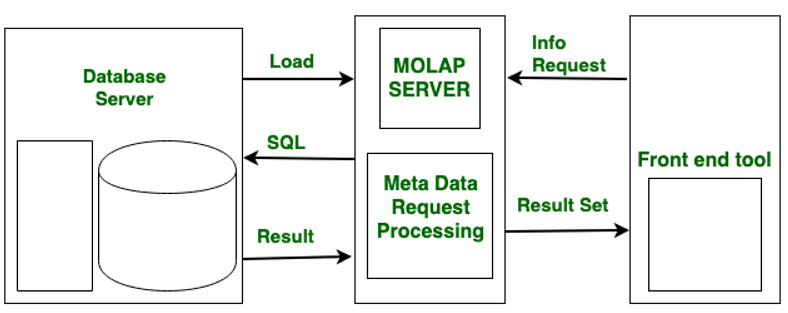


*ROLAP*

**Multidimensional OLAP (MOLAP): Cube-Based**

[MOLAP](https://www.geeksforgeeks.org/difference-between-rolap-molap-and-holap/) 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 elements 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.

But unlike ROLAP, where only records with non-zero facts are stored, all array elements are defined in MOLAP and as a result, the arrays generally tend to be sparse, with empty elements occupying a greater part of it. Since both storage and retrieval costs are important while assessing online performance efficiency, MOLAP systems typically include provisions such as advanced indexing and hashing to locate data while performing queries for handling sparse arrays. MOLAP cubes are fast data retrieval, optimal for slicing and dicing, and can perform complex calculations. All calculations are pre-generated when the cube is created.



*MOALP*

### **Hybrid OLAP (HOLAP)**

HOLAP is a combination of ROLAP and MOLAP.[HOLAP](https://www.geeksforgeeks.org/difference-between-rolap-molap-and-holap/)servers allow for storing large data volumes of detailed data. On the one hand, HOLAP leverages the greater scalability of ROLAP. On the other hand, HOLAP leverages cube technology for faster performance and summary-type information. Cubes are smaller than MOLAP since detailed data is kept in the relational database. The database is used to store data in the most functional way possible.

### Transparent OLAP (TOLAP)

TOLAP systems are designed to work transparently with existing RDBMS systems, allowing users to access OLAP features without needing to transfer data to a separate OLAP system. This allows for more seamless integration between OLAP and traditional [RDBMS](https://www.geeksforgeeks.org/rdbms-full-form/)systems.

## Advantages of OLAP System

* **Fast query response:** OLAP systems are designed to provide fast query response times, even for complex queries involving large amounts of data.
* **Multidimensional analysis:**OLAP systems allow users to analyze data from multiple dimensions, such as time, location, product, and customer, providing a deeper understanding of the data.
* **Flexible and customizable:**OLAP systems are highly customizable, allowing users to define their dimensions, hierarchies, and calculations.
* **Improved decision-making:**OLAP systems provide users with the ability to analyze data from different angles, leading to better insights and more informed decision-making.

## Disadvantages of the OLAP System

* **Complexity:**OLAP systems can be complex to implement and maintain, requiring specialized skills and knowledge.
* **Data storage requirements:** OLAP systems require a large amount of storage space to store multidimensional data, which can be expensive and difficult to manage.
* **Limited transactional processing:**OLAP systems are optimized for analytical processing, but they are not suitable for transactional processing, which can lead to performance issues.
* **Performance degradation with large datasets:**As the size of the dataset increases, the performance of OLAP systems may degrade, requiring additional hardware resources to maintain performance.

1. **Explain different data storing techniques available in distributed DBMS.**

**Ans.**

1. **Explain Fragmentation and Replication in detail.**

**Ans.**

1. **Explain Parallel database architecture.**

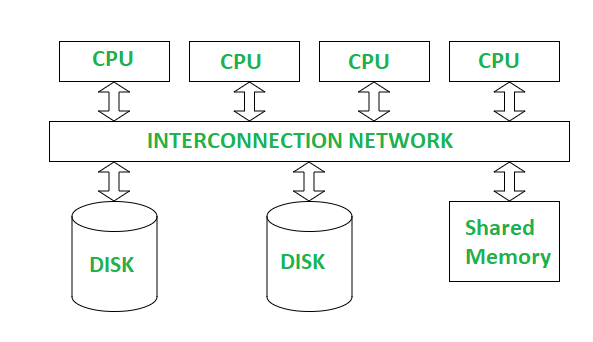
**Ans.** A parallel DBMS is a DBMS that runs across multiple processors or CPUs and is mainly designed to execute query operations in parallel, wherever possible. The parallel DBMS link a number of smaller machines to achieve the same throughput as expected from a single large machine.

In Parallel Databases, mainly there are three architectural designs for parallel DBMS. They are as follows:

1. **Shared Memory Architecture**
2. **Shared Disk Architecture**
3. **Shared Nothing Architecture**

Let’s discuss them one by one:

**1. Shared Memory Architecture-**In Shared Memory Architecture, there are multiple CPUs that are attached to an interconnection network. They are able to share a single or global main memory and common disk arrays. It is to be noted that, In this architecture, a single copy of a multi-threaded operating system and multithreaded DBMS can support these multiple CPUs. Also, the shared memory is a solid coupled architecture in which multiple CPUs share their memory. It is also known as **Symmetric multiprocessing (SMP)**. This architecture has a very wide range which starts from personal workstations that support a few microprocessors in parallel via RISC.



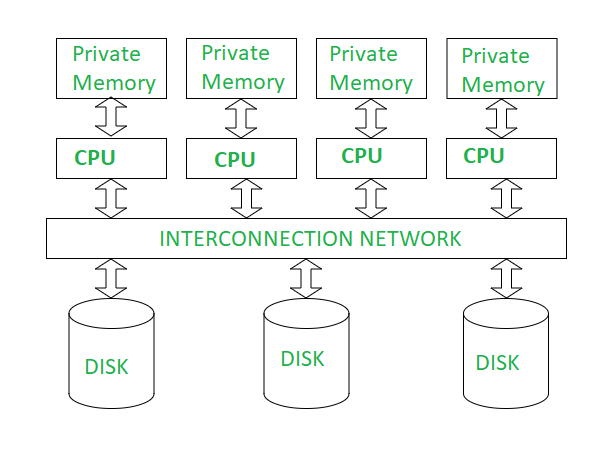
*Shared Memory Architecture*

**Advantages :**

1. It has high-speed data access for a limited number of processors.
2. The communication is efficient.

**Disadvantages :**

1. It cannot use beyond 80 or 100 CPUs in parallel.
2. The bus or the interconnection network gets block due to the increment of the large number of CPUs.
3. **Shared Disk Architectures :**  
   In Shared Disk Architecture, various CPUs are attached to an interconnection network. In this, each CPU has its own memory and all of them have access to the same disk. Also, note that here the memory is not shared among CPUs therefore each node has its own copy of the operating system and DBMS. Shared disk architecture is a loosely coupled architecture optimized for applications that are inherently centralized. They are also known as **clusters**.



*Shared Disk Architecture*

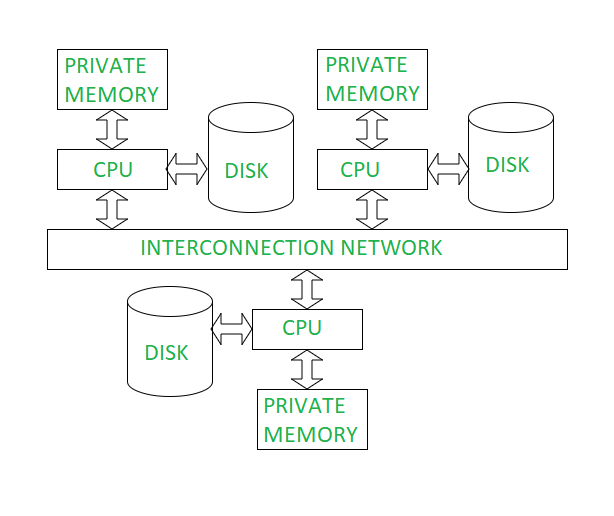
**Advantages :**

1. The interconnection network is no longer a bottleneck each CPU has its own memory.
2. Load-balancing is easier in shared disk architecture.
3. There is better fault tolerance.

**Disadvantages :**

1. If the number of CPUs increases, the problems of interference and memory contentions also increase.
2. There’s also exists a scalability problem.

**3, Shared Nothing Architecture :**  
Shared Nothing Architecture is multiple processor architecture in which each processor has its own memory and disk storage. In this, multiple CPUs are attached to an interconnection network through a node. Also, note that no two CPUs can access the same disk area. In this architecture, no sharing of memory or disk resources is done. It is also known as **Massively parallel processing (MPP).**



*Shared Nothing Architecture*

**Advantages :**

1. It has better scalability as no sharing of resources is done
2. Multiple CPUs can be added

**Disadvantages:**

1. The cost of communications is higher as it involves sending of data and software interaction at both ends
2. The cost of non-local disk access is higher than the cost of shared disk architectures.

Note that this technology is typically used for very large databases that have the size of 1012 bytes or TB or for the system that has the process of thousands of transactions per second.

**4, Hierarchical Architecture :**

This architecture is a combination of shared disk, shared memory and shared nothing architectures. This architecture is scalable due to availability of more memory and many processor. But is costly to other architecture.