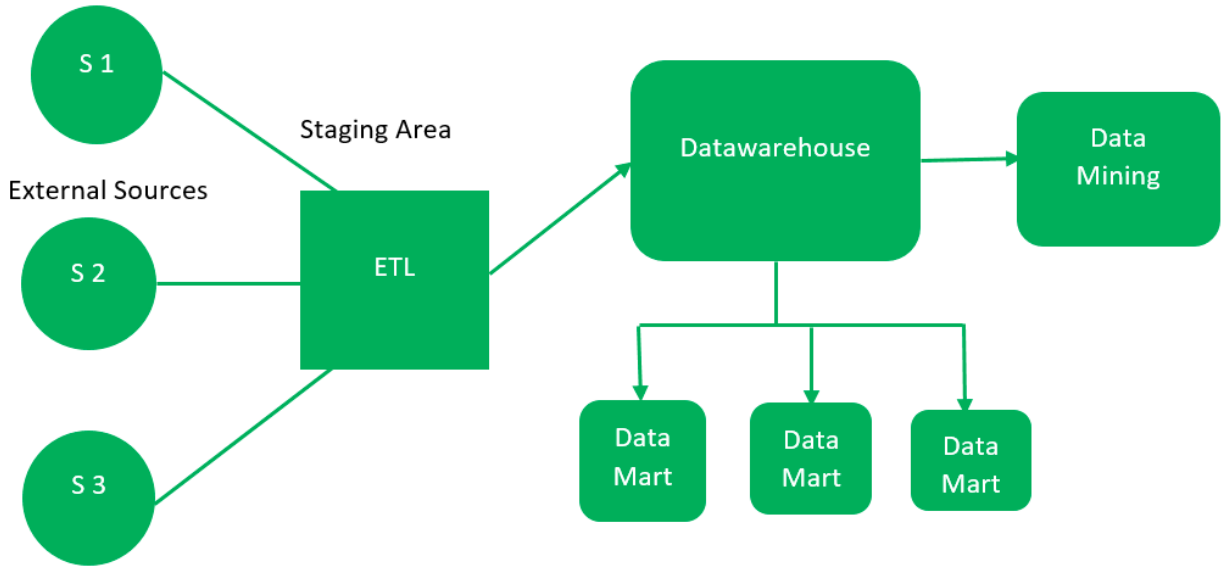
ADBMS_UNIT-4

Data Warehouse Architecture

A Data Warehouse therefore can be described as a system that consolidates and manages data from different sources to assist an organization in making proper decisions. This makes the work of handling data to report easier. Two main construction approaches are used: Two of the most common models that have been developed are the Top-Down approach and the Bottom-Up approach and each of them possesses its strengths and weaknesses.

What is Top-Down Approach?

The initial approach developed by Bill Inmon known as the top-down approach starts with building a single source data warehouse for the whole company. Merges and processes external data through the ETL (**Extract, Transform, Load**) process and subsequently stores them in the data warehouse.



The essential components are discussed below:

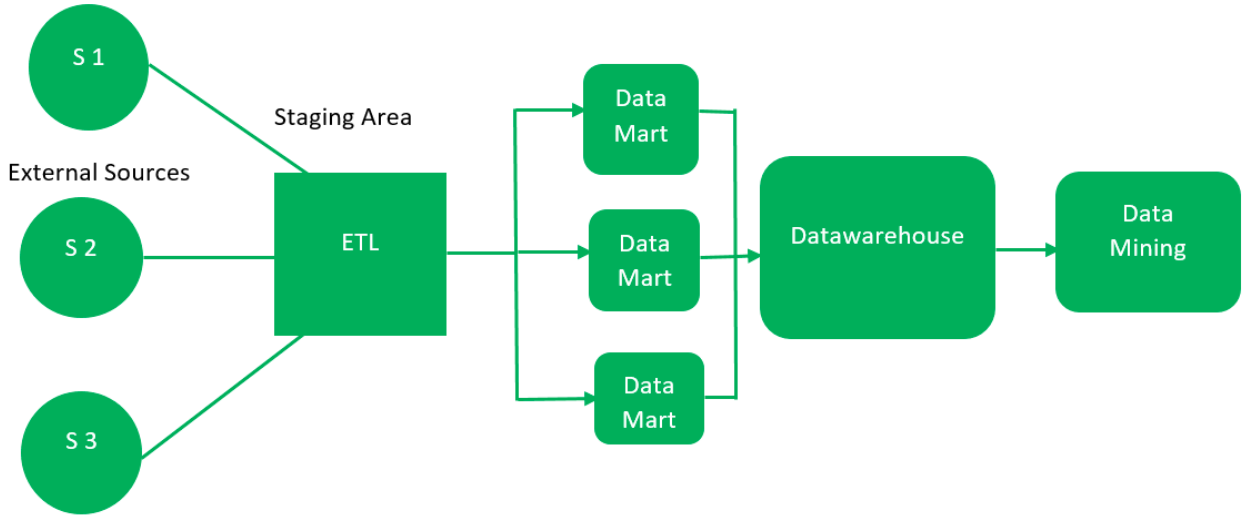
1. **External Sources:** External source is a source from where data is collected irrespective of the type of data. Data can be structured, semi structured and unstructured as well.

2. **Stage Area:** Since the data, extracted from the external sources does not follow a particular format, so there is a need to validate this data to load into data warehouse. For this purpose, it is recommended to use **ETL** tool.
    - **E(Extracted):** Data is extracted from External data source.
    - **T(Transform):** Data is transformed into the standard format.
    - **L(Load):** Data is loaded into data warehouse after transforming it into the standard format.
3. **Data-warehouse:** After cleansing of data, it is stored in the data warehouse as central repository. It actually stores the meta data and the actual data gets stored in the data marts. **Note** that data warehouse stores the data in its purest form in this top-down approach.
4. **Data Marts:** Data mart is also a part of storage component. It stores the information of a particular function of an organization which is handled by single authority. There can be as many number of data marts in an organization depending upon the functions. We can also say that data mart contains subset of the data stored in data warehouse.
5. **Data Mining:** The practice of analyzing the big data present in data warehouse is **data mining**. It is used to find the hidden patterns that are present in the database or in data warehouse with the help of algorithm of data mining.

    This approach is defined by **Inmon** as – data warehouse as a central repository for the complete organization and data marts are created from it after the complete data warehouse has been created.

What is Bottom-Up Approach?

Bottom up Approach is the Ralph Kimball's approach of the construction of individual data marts that lie at the center of specific business goals or functions such as marketing or sales. These data marts are extracted transformed & loaded first to provide organizations' ability to generate reports instantly. In turn, these data marts are affiliated to the more centralized and broad data warehouse system.



1. First, the data is extracted from external sources (same as happens in top-down approach).
2. Then, the data go through the staging area (as explained above) and loaded into data marts instead of data warehouse. The data marts are created first and provide reporting capability. It addresses a single business area.
3. These data marts are then integrated into data warehouse.

Top-Down Approach (Enterprise-wide Data Warehouse → Data Marts)

Advantages:

1. **Integrated and Consistent Data**: Provides a unified view of enterprise data, ensuring consistency across all departments.
2. **Strategic Focus**: Aligns with long-term business goals and ensures scalability for future needs.
3. **Eliminates Redundancy**: Reduces duplicate data as all data originates from a single, central warehouse.

Disadvantages:

1. **High Initial Cost and Effort**: Requires significant time, money, and resources to design and implement.

2. **Delayed Benefits**: The system may take a long time to deliver usable results.
3. **Complex Implementation**: Requires deep planning and coordination across the entire organization.

Bottom-Up Approach (Departmental Data Marts → Enterprise Data Warehouse)

Advantages:

1. **Faster Implementation**: Data marts can be quickly developed for immediate needs.
2. **Cost-Effective**: Requires less initial investment as it starts with smaller-scale projects.
3. **Flexible and Adaptable**: Can adapt to specific departmental needs without waiting for the entire enterprise structure.

Disadvantages:

1. **Lack of Integration**: Risk of fragmented or inconsistent data across departments.
2. **Scalability Challenges**: Integrating multiple data marts into a central warehouse later can be complex.
3. **Duplicate Efforts**: May lead to redundant data storage and processing across data marts.

Characteristics of a Data Warehouse

1. **Subject-Oriented**: Organized around major business areas like sales, finance, or marketing, rather than day-to-day operations.
2. **Integrated**: Combines data from different sources into a consistent and unified format.
3. **Time-Variant**: Stores historical data for trend analysis, enabling time-based insights.
4. **Non-Volatile**: Data is stable and not frequently updated or deleted; it allows read-only access for analysis.

5. **Optimized for Query and Analysis**: Designed for complex queries and reporting, not for transaction processing.

Limitations of a Data Warehouse

1. **High Initial Cost**: Development and maintenance require significant investment in infrastructure, tools, and expertise.
2. **Complex Data Integration**: Combining data from multiple sources can be technically challenging.
3. **Scalability Issues**: As data grows, maintaining performance and storage efficiency can become difficult.
4. **Limited Real-Time Analysis**: Often designed for batch processing, making it less suitable for real-time data.
5. **Time-Consuming Implementation**: Building a fully functional data warehouse can take months or years.

Star Schema in Data Warehousing

A **Star Schema** is a data modeling technique used in data warehousing to structure and organize data for efficient querying and analysis. It consists of a central **fact table** surrounded by related **dimension tables**, resembling a star.

Key Components

1. **Fact Table**
   - Contains the quantitative data (measures) being analyzed, such as revenue, sales quantity, or profit.
   - Includes foreign keys referencing dimension tables.
   - Example: **Fact Table: SALES**
     - Columns: Product ID, Order ID, Customer ID, Employee ID, Total Sales, Quantity, Discount
2. **Dimension Tables**

- Store descriptive attributes (context) about the measures in the fact table.
- Connected to the fact table through foreign keys.
- Examples:
  - **Employee Dimension Table**: Employee ID, Employee Name, Department, Region
  - **Product Dimension Table**: Product ID, Product Name, Product Category, Unit Price
  - **Customer Dimension Table**: Customer ID, Customer Name, Address, City, Zip
  - **Time Dimension Table**: Order ID, Order Date, Month, Year, Quarter

---

Features

1. **Central Fact Table**: Core table containing numerical data.
2. **Dimension Tables**: Provide context and descriptive details.
3. **Denormalized Structure**: Redundancy is allowed for faster query performance.
4. **Simple Queries**: Easy and fast to join the fact table with dimension tables.
5. **Aggregated Data**: Supports multiple levels of data granularity (e.g., daily, monthly).
6. **Fast Performance**: Optimized for OLAP (Online Analytical Processing).
7. **Ease of Understanding**: Intuitive design, even for non-technical users.

Advantages

1. **Simpler Queries**: Easy to write and execute.
2. **Faster Query Performance**: Optimized for OLAP systems.
3. **Scalability**: Easily extended by adding new dimensions or measures.

Disadvantages

1. **Data Integrity Issues**: Redundancy can lead to inconsistencies.
2. **Not Flexible**: Limited analytical capabilities for complex relationships.
3. **No Support for Many-to-Many Relationships**: At least not directly.

Snowflake Schema in Data Warehousing

The **snowflake schema** is a variation of the star schema used in data warehousing, where the dimension tables are normalized into multiple related tables, forming a hierarchical structure. This design helps to reduce redundancy and improve data integrity, making it efficient for handling large datasets with detailed hierarchical relationships. However, the normalization of data also adds complexity to queries and can impact query performance due to the increased number of joins.

Key Features of Snowflake Schema:

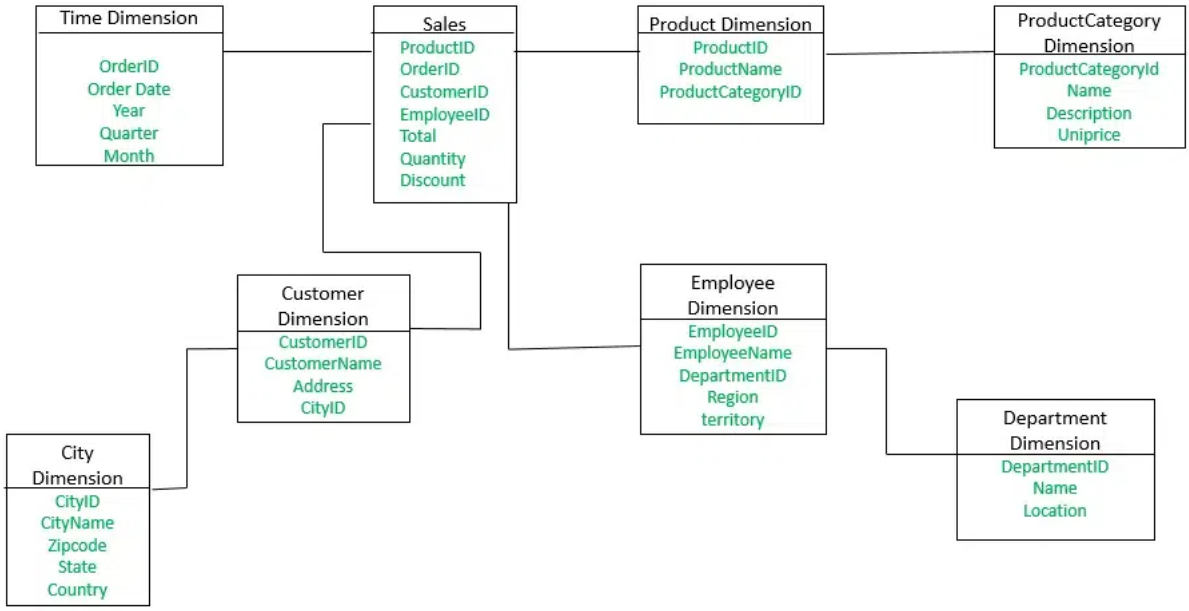
1. **Normalization**: Reduces data redundancy by organizing data into related tables.
2. **Hierarchical Structure**: Allows detailed and granular analysis through multiple levels of related tables.
3. **Joins**: Requires more complex queries involving multiple table joins.
4. **Scalability**: Handles large datasets effectively, though its complexity can make it harder to manage.

Advantages:

1. **Improved Data Integrity**: Reduces redundancy and ensures consistency.
2. **Storage Efficiency**: Saves disk space by normalizing dimension tables.
3. **Detailed Data Analysis**: Allows for deep analysis through hierarchical data organization.

Disadvantages:

1. **Slower Query Performance**: More joins are needed, impacting query speed.
2. **Increased Complexity**: The schema is harder to understand, manage, and maintain.
3. **Minimal Space Savings**: The reduction in disk space usage may not be significant compared to the overall size of the data warehouse.



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

Comparison: Star Schema vs. Snowflake Schema

Feature	**Star Schema**	**Snowflake Schema**
Normalization	Denormalized	Normalized

Data Redundancy	High	Low
Query Performance	Fast (fewer joins)	Slower (more joins)
Ease of Use	Simple to understand	Complex
Storage Requirement	Higher	Lower
Scalability	Moderate	High
Data Integrity	Moderate	High

OLAP

Online Analytical Processing(OLAP) refers to a set of software tools used for data analysis in order to make business decisions. OLAP provides a platform for gaining insights from databases retrieved from multiple database systems at the same time. It is based on a multidimensional data model, which enables users to extract and view data from various perspectives. A multidimensional database is used to store OLAP data. Many Business Intelligence (BI) applications rely on OLAP technology.

Type of OLAP servers:

The three major types of OLAP servers are as follows:

- **ROLAP**
- **MOLAP**
- **HOLAP**

Relational OLAP (ROLAP):

Relational On-Line Analytical Processing (ROLAP) is primarily used for data stored in a relational database, where both the base data and dimension tables are stored as relational tables. ROLAP servers are used to bridge the gap between the relational back-end server and the client's front-end tools. ROLAP servers store and manage warehouse data using RDBMS, and OLAP middleware fills in the gaps.

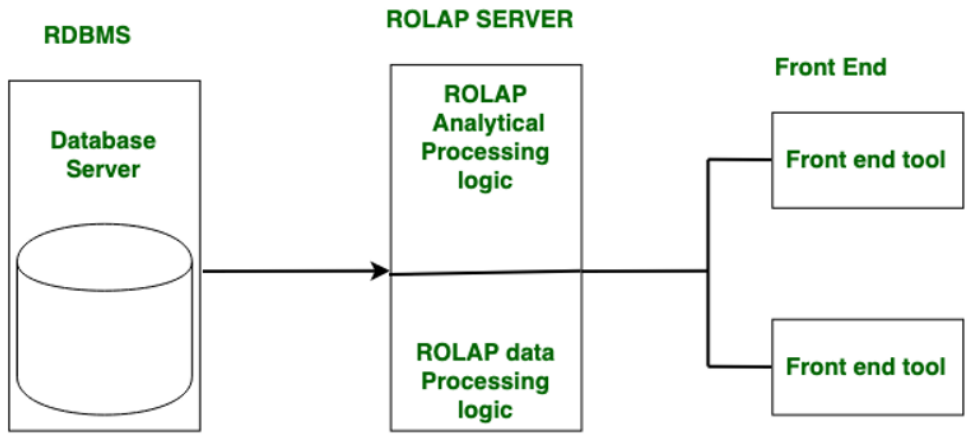
Benefits:

- It is compatible with data warehouses and OLTP systems.

- The data size limitation of ROLAP technology is determined by the underlying RDBMS. As a result, ROLAP does not limit the amount of data that can be stored.

Limitations:

- SQL functionality is constrained.
- It's difficult to keep aggregate tables up to date.



Multidimensional OLAP (MOLAP):

Through array-based multidimensional storage engines, Multidimensional On-Line Analytical Processing (MOLAP) supports multidimensional views of data. Storage utilization in multidimensional data stores may be low if the data set is sparse.

MOLAP stores data on discs in the form of a specialized multidimensional array structure. It is used for OLAP, which is based on the arrays' random access capability. Dimension instances determine array elements, and the data or measured value associated with each cell is typically stored in the corresponding array element. The multidimensional array is typically stored in MOLAP in a linear allocation based on nested traversal of the axes in some predetermined order.

However, unlike ROLAP, which stores only records with non-zero facts, all array elements are defined in MOLAP, and as a result, the arrays tend to be sparse, with empty elements occupying a larger portion of them. MOLAP systems typically include provisions such as advanced indexing and hashing to locate data while performing queries for handling sparse arrays, because both storage and retrieval

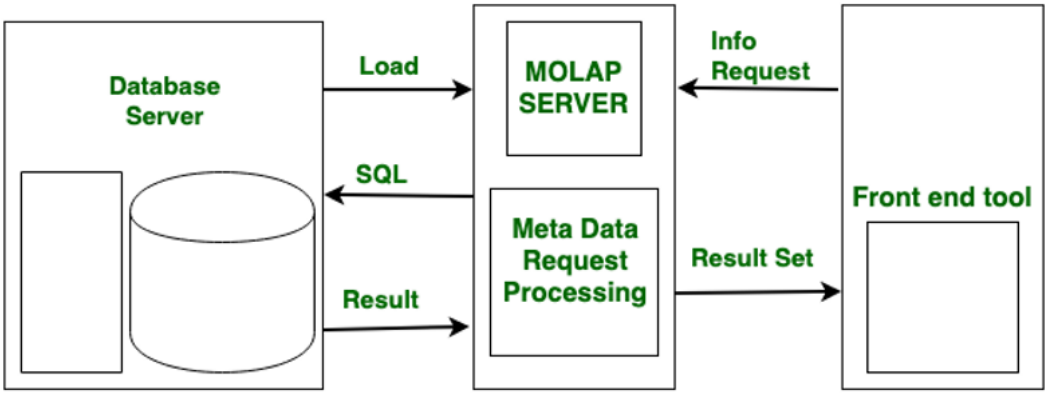
costs are important when evaluating online performance. MOLAP cubes are ideal for slicing and dicing data and can perform complex calculations. When the cube is created, all calculations are pre-generated.

Benefits:

- Suitable for slicing and dicing operations.
- Outperforms ROLAP when data is dense.
- Capable of performing complex calculations.

Limitations:

- It is difficult to change the dimensions without re-aggregating.
- Since all calculations are performed when the cube is built, a large amount of data cannot be stored in the cube itself.



Hybrid OLAP (HOLAP):

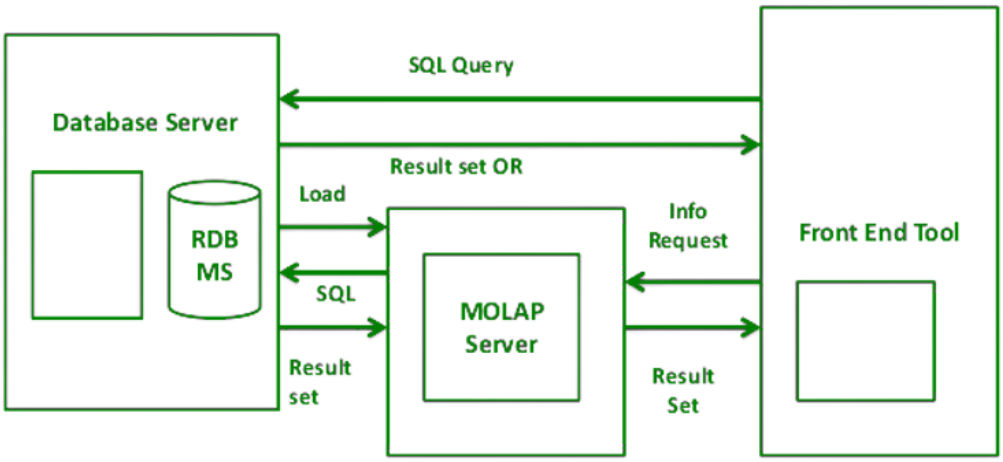
ROLAP and MOLAP are combined in Hybrid On-Line Analytical Processing (HOLAP). HOLAP offers greater scalability than ROLAP and faster computation than MOLAP.HOLAP is a hybrid of ROLAP and MOLAP. HOLAP servers are capable of storing large amounts of detailed data. On the one hand, HOLAP benefits from ROLAP's greater scalability. HOLAP, on the other hand, makes use of cube technology for faster performance and summary-type information. Because detailed data is stored in a relational database, cubes are smaller than MOLAP.

Benefits:

- HOLAP combines the benefits of MOLAP and ROLAP.
- Provide quick access at all aggregation levels.

Limitations

- Because it supports both MOLAP and ROLAP servers, HOLAP architecture is extremely complex.
- There is a greater likelihood of overlap, particularly in their functionalities.



Other types of OLAP include:

- **Web OLAP (WOLAP):** WOLAP refers to an OLAP application that can be accessed through a web browser. WOLAP, in contrast to traditional client/server OLAP applications, is thought to have a three-tiered architecture consisting of three components: a client, middleware, and a database server.
- **Desktop OLAP (DOLAP):** DOLAP is an abbreviation for desktop analytical processing. In that case, the user can download the data from the source and work with it on their desktop or laptop. In comparison to other OLAP applications, functionality is limited. It is less expensive.
- **Mobile OLAP (MOLAP):** Wireless functionality or mobile devices are examples of MOLAP. The user is working and accessing data via mobile devices.

- **Spatial OLAP (SOLAP):** SOLAP egress combines the capabilities of Geographic Information Systems (GIS) and OLAP into a single user interface. SOLAP is created because the data can be alphanumeric, image, or vector. This allows for the quick and easy exploration of data stored in a spatial database.

Decision Support System (DSS)

A Decision Support System (DSS) is a tool that helps individuals and organizations make better decisions by combining data, models, and software. It enables real-time decision-making by providing the ability to analyze semi-structured and unstructured data, offering insights that would be difficult to generate manually. DSS improves decision quality, speed, and efficiency, supporting decisions based on comprehensive, timely, and accurate information. Components of a Decision Support System (DSS)

A Decision Support System (DSS) consists of three primary components that work together to help users make informed decisions. These components are: Database (Knowledge Base), Model, and User Interface.

1. Database (Knowledge Base)
   The Database or Knowledge Base is the foundation of a DSS, containing both internal and external data necessary for decision-making.
   Purpose: It stores and integrates data from various sources, including transactional databases, external data feeds, and historical records. This data is used to inform decision-making processes and support analyses.
   Data Storage and Integration: The database can store large amounts of structured and unstructured data, such as past sales records, customer information, market trends, and competitor data.
   Example: In a retail DSS, the database might contain past sales details, current stock levels, customer demographics, and macroeconomic data, all of which are needed to analyze pricing, inventory, and sales trends.
   Data Consistency: Ensures that information is accurate and up-to-date, allowing for reliable decision-making.

2. Model (Decision Context and User Criteria)
   The Model component consists of decision models, algorithms, and analytical tools that help process and analyze the data stored in the database. It enables decision-makers to perform what-if analysis, optimization, and forecasting based on different variables and scenarios.

   Purpose: The model analyzes data, generates scenarios, and makes predictions to support decision-making. It can help in optimizing choices, evaluating risks, and forecasting outcomes.
   Types of Models:
   Statistical Models: Used for predicting future trends based on historical data.
   Optimization Models: Used to determine the best decision by minimizing or maximizing certain factors, such as cost or profit.
   Simulation Models: Simulate different scenarios to predict outcomes based on variable changes.
   Example: A financial DSS might use forecasting models to predict future cash flow, risk evaluation models for assessing investment portfolios, and optimization models to allocate resources efficiently.

3. User Interface (UI)
   The User Interface (UI) is the part of the DSS that allows users to interact with the system. It enables users to input data, define parameters, view analysis results, and generate reports in a way that is easy to understand.
   Purpose: The UI provides a way for users to interact with the system, making the data and decision-making process accessible and comprehensible. It typically features tools for data input, customization, and visualization.
   Visualization and Reporting: The UI often includes dashboards, graphs, and charts to present data in an easy-to-understand format, helping users make informed decisions quickly.
   Interactivity: Allows users to manipulate data, run simulations, and explore different scenarios in real-time.
   Example: In a healthcare DSS, the UI might include visual elements like a clickable map of a patient's body, where clinicians can input symptoms and receive treatment suggestions based on the system's analysis of previous cases.

Key Purpose of DSS (Top 4-5):
Improves Decision Quality: By providing comprehensive, accurate, and timely data, DSS enhances the quality of decisions.
Handles Complex Problems: DSS supports decision-making in complex, semi-structured, or unstructured situations where traditional approaches may fall short.
Facilitates Rapid Decision Making: Automates data collection and analysis, enabling quicker response to dynamic business environments.
Supports Strategic Planning: DSS provides tools for long-term forecasting, scenario modeling, and simulations, helping organizations plan for the future.

Advantages of DSS (Top 4-5):
Improved Decision Quality: DSS helps decision-makers make more informed and effective choices based on comprehensive analysis and data modeling.
Efficiency and Speed: Automates many parts of the decision-making process, cutting down the time needed for analysis and decision execution.
Enhanced Productivity: By streamlining data processing, DSS allows decision-makers to focus on other important tasks.
Better Data Management: Integrates and organizes data from various sources, ensuring accuracy and consistency.
Supports Complex Analysis: DSS enables organizations to handle intricate decision-making tasks that are difficult to manage with traditional methods.

Disadvantages of DSS (Top 4-5):
High Cost: Developing and maintaining a DSS can be expensive, especially for large organizations.
Complexity: Designing and implementing a DSS can be challenging, requiring specialized knowledge and expertise.
Data Quality Issues: The effectiveness of a DSS is heavily dependent on the quality and consistency of the data it uses.
User Resistance: Employees may resist adopting DSS due to unfamiliarity or fear of technology.
Over-Dependency on Technology: Relying too heavily on the system's outputs can lead to issues if the system malfunctions or if users don’t critically evaluate the results.

Views of Decision Support Systems

The **views** of a Decision Support System refer to the different perspectives from which the system is utilized by decision-makers, data analysts, and IT professionals. These views emphasize the key functions and goals of DSS in supporting decision-making:

1. **Data-Driven View**:
    - Focuses on the access, transformation, and reporting of data. DSS in this view primarily serves as a tool for collecting, storing, and analyzing historical and real-time data to assist in decision-making.
    - **Example**: A sales analysis system that helps executives analyze past sales data to forecast future trends.
2. **Model-Driven View**:
    - Emphasizes the use of mathematical models, simulations, and algorithms to optimize decision-making. DSS in this view supports decision-makers in evaluating different scenarios based on various decision variables.
    - **Example**: Financial forecasting models that help businesses optimize investment portfolios or predict cash flows.
3. **Communication-Driven View**:
    - Focuses on enhancing communication and collaboration among decision-makers. DSS in this view supports group decision-making, brainstorming, and the sharing of ideas and information.
    - **Example**: Group Decision Support Systems (GDSS) used in board meetings or collaborative planning sessions to make joint decisions.
4. **Knowledge-Driven View**:
    - DSS in this view is used to provide specialized knowledge, rules, or heuristics for decision-making. It often integrates expert systems to assist in decision-making in specific domains, such as healthcare, law, or engineering.

- **Example**: A medical DSS that helps healthcare professionals diagnose diseases based on patient symptoms and medical history.

Decision Support

Decision support refers to the process by which a DSS helps individuals or organizations make better decisions. This support is provided through various techniques and tools, which are tailored to different types of decision-making:

1. **Structured Decisions**:
    - These decisions are routine and can be addressed using standard procedures and algorithms. A DSS helps automate data gathering and reporting for these decisions.
    - **Example**: Inventory management decisions, where a DSS can track stock levels and automatically reorder products when they fall below a certain threshold.
2. **Semi-Structured Decisions**:
    - These decisions are partially structured and require the application of both quantitative data and qualitative factors. DSS helps provide insights by analyzing data and presenting different options.
    - **Example**: Budget planning, where a DSS may provide budget forecasts based on past trends and provide what-if analysis to adjust for new policies.
3. **Unstructured Decisions**:
    - These decisions are complex, with no standard solution, often involving a high degree of uncertainty or requiring subjective judgment. DSS aids decision-makers by offering detailed analysis, predictions, and scenario modeling.
    - **Example**: Strategic business decisions, such as market entry or product launch, where the decision-maker needs to assess various external factors and internal constraints.