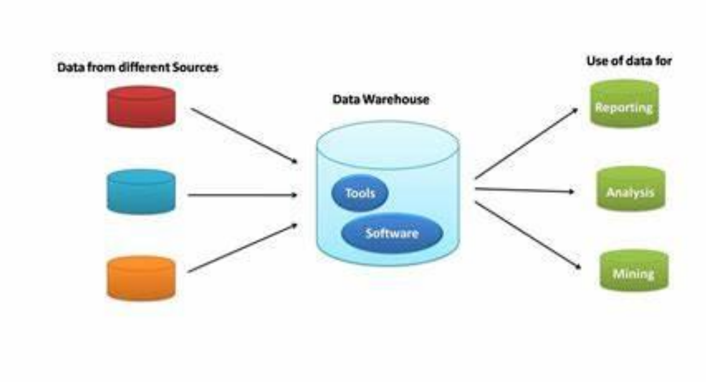
DATA WAREHOUSE & MINING MODULE 1

What is Data Warehouse

- A data warehouse is a centralized repository that stores large amounts of structured data from various sources.
- It's designed to facilitate complex queries and analysis, making it easier for organizations to generate reports, perform data mining, and make data-driven decisions.
- The data in a warehouse is typically cleaned, transformed, and organized for efficient querying and analysis.
- Datawarehouse is not loaded every time data is generated



NEED OF DATAWAREHOUSE:

 Centralized Data Storage: It consolidates data from multiple sources into a single, integrated repository, making it easier to manage and analyze.

 Improved Query Performance: Data warehousing optimizes data for query performance, enabling faster retrieval and analysis of large volumes of data.

 Historical Analysis: It allows for the storage of historical data, facilitating trend analysis and longitudinal studies over time.

 Data Consistency: It ensures consistency and accuracy of data by standardizing and cleaning data before it is loaded into the warehouse.

 Enhanced Decision-Making: By providing a comprehensive view of data, it supports better decision-making through advanced analytics and reporting tools.

 Support for Business Intelligence: It integrates with business intelligence tools to help in generating insights, forecasts, and strategic planning.

In summary datawarehouse is important to gain a edge over competitors by taking smart decisions

FEATURES OF DATA WAREHOUSE

1] SUBJECT ORIENTED DATA

Subject-oriented data in a data warehouse refers to how data is organized and categorized based on key business subjects or domains rather than specific transactions or operational processes. This organization allows for more meaningful and insightful analysis. Here's a closer look:

1. **Focus on Business Areas:** Data is grouped around major business subjects such as sales, finance, marketing, and customer service. This helps users to analyze and understand trends and patterns related to specific business functions.
2. **Enhanced Reporting:** By organizing data according to subjects, it simplifies the creation of reports and dashboards focused on particular areas of interest, such as sales performance or financial summaries.

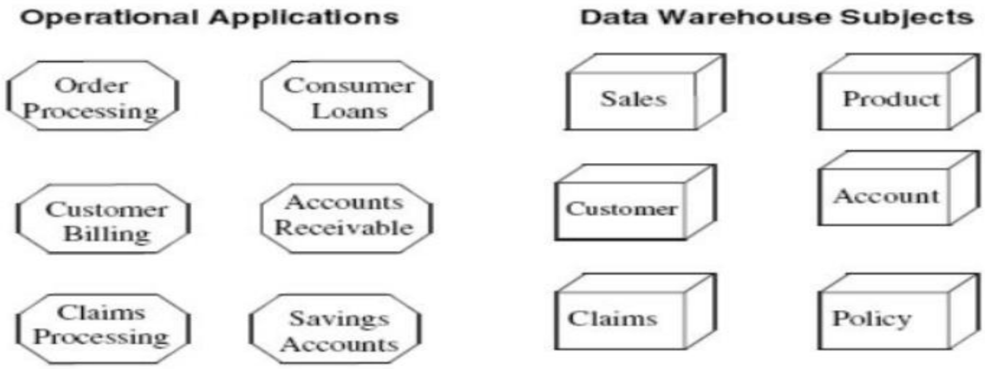


Figure 2-1 The data warehouse is subject oriented.

2] INTEGRATED DATA

Integrated data in a data warehouse refers to the process of combining data from multiple sources into a unified, consistent format. This integration ensures that data from different systems, departments, or applications can be analyzed together seamlessly. Here are key aspects of integrated data:

1. **Unified Data View:** Consolidates data from disparate sources, such as CRM systems, ERP systems, and external databases, into a single repository. This unified view allows for comprehensive analysis across various business functions.
2. **Consistency:** Standardizes data formats, units of measurement, and naming conventions, ensuring consistency across the organization. This reduces discrepancies and makes data more reliable for analysis.

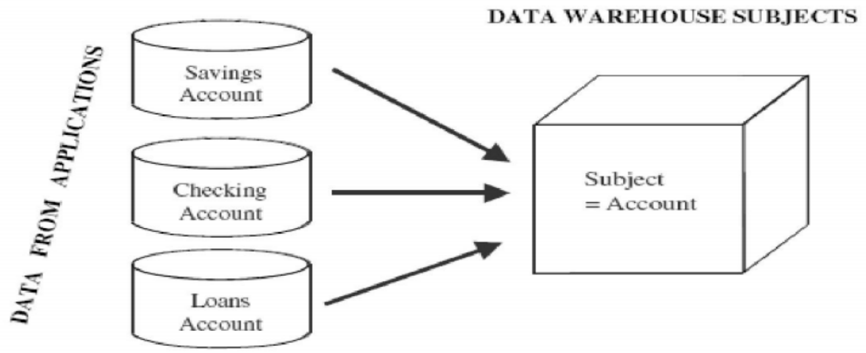


Figure 2-2 The data warehouse is integrated.

Before the data from various disparate sources can be usefully stored in a data warehouse, you have to:

▪ remove the inconsistencies;

▪ standardize the various data elements;

▪ make sure of the meanings of data names in each source application

3] Time Variant:

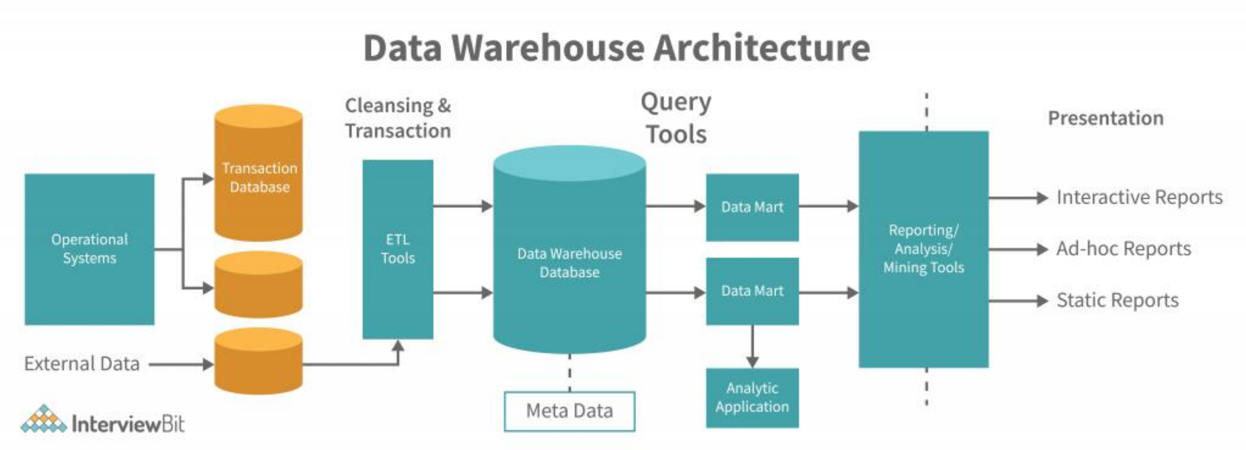
- **Time-Variant Data** is a fundamental feature of data warehousing that refers to the ability of a data warehouse to store and manage historical data.
- Unlike operational databases that typically handle current transactional data, a data warehouse is designed to track changes over time, enabling users to perform time-based analyses.
- In a data warehouse, data is stored with time stamps that indicate when it was recorded or last updated.
- This historical perspective allows users to view data as it existed at various points in time, supporting trend analysis, historical comparisons, and longitudinal studies.
- For instance, organizations can analyze sales performance over several years to identify trends and patterns, assess changes in customer behavior, or evaluate the impact of business decisions over time.
- Time-variant data management enables complex queries that involve data from different periods, such as comparing current performance with past performance or forecasting future trends based on historical data.
- By maintaining a historical record, data warehouses provide a comprehensive view of business evolution and support strategic planning and decision-making

4] Non Volatile Data:

Non-Volatile Data is a critical feature of data warehousing that refers to the characteristic of data being stable and immutable once it is entered into the data warehouse. Unlike operational databases, where data is frequently updated and modified, data in a data warehouse remains unchanged after it has been loaded.

This stability ensures that historical data is preserved in its original state, providing a reliable basis for analysis and reporting. Non-volatile data allows organizations to perform accurate trend analysis, historical comparisons, and time-based queries without concerns that the underlying data might change unexpectedly. For example, a retail company can analyze sales data over multiple years to identify patterns and forecast future performance, knowing that past data remains consistent and unaltered.

DATA WAREHOUSE ARCHITECTURE



1. Data Sources

- **Operational Databases:** These include transactional systems, CRM systems, ERP systems, and other operational databases where data is initially generated and stored.
- **External Data Sources:** External sources such as social media, market data, and other external feeds may also be integrated.

2. ETL (Extract, Transform, Load) Layer

- **Extract:** Data is extracted from various source systems. This step involves retrieving data from operational databases, files, or other sources.
- **Transform:** Extracted data is transformed into a format suitable for analysis. This process includes data cleansing, normalization, aggregation, and enrichment to ensure consistency and quality.

- **Load:** The transformed data is loaded into the data warehouse. This step involves inserting data into the data warehouse's storage structures, such as fact and dimension tables.

3. Data Warehouse Database

- **Data Storage:** The core of the data warehouse, where data is stored in a structured format. It typically uses a relational database management system (RDBMS) or other storage solutions optimized for analytical queries.
- **Schema Design:** Data is organized into schemas such as star schema or snowflake schema. These schemas define how data is structured and related, using fact tables (measuring business processes) and dimension tables (contextual data).

4. Data Access and Analysis

- **Business Intelligence (BI) Tools:** These tools provide the interface for users to interact with the data warehouse. They include reporting tools, dashboards, and data visualization tools that help in generating insights and making data-driven decisions.
- **Ad-Hoc Querying:** Users can run custom queries to explore specific aspects of the data as needed.

5. Metadata

- **Data Dictionary:** Contains information about the data's origin, format, and structure. It helps users understand the data and its context.
- **Metadata Repository:** Stores metadata related to data warehousing processes, including ETL operations, data lineage, and schema definitions.

7. Data Marts

- **Specialized Data Marts:** Data warehouses often contain data marts that focus on specific business areas or departments (e.g., sales, finance). Data marts are subsets of the data warehouse optimized for particular types of analysis.

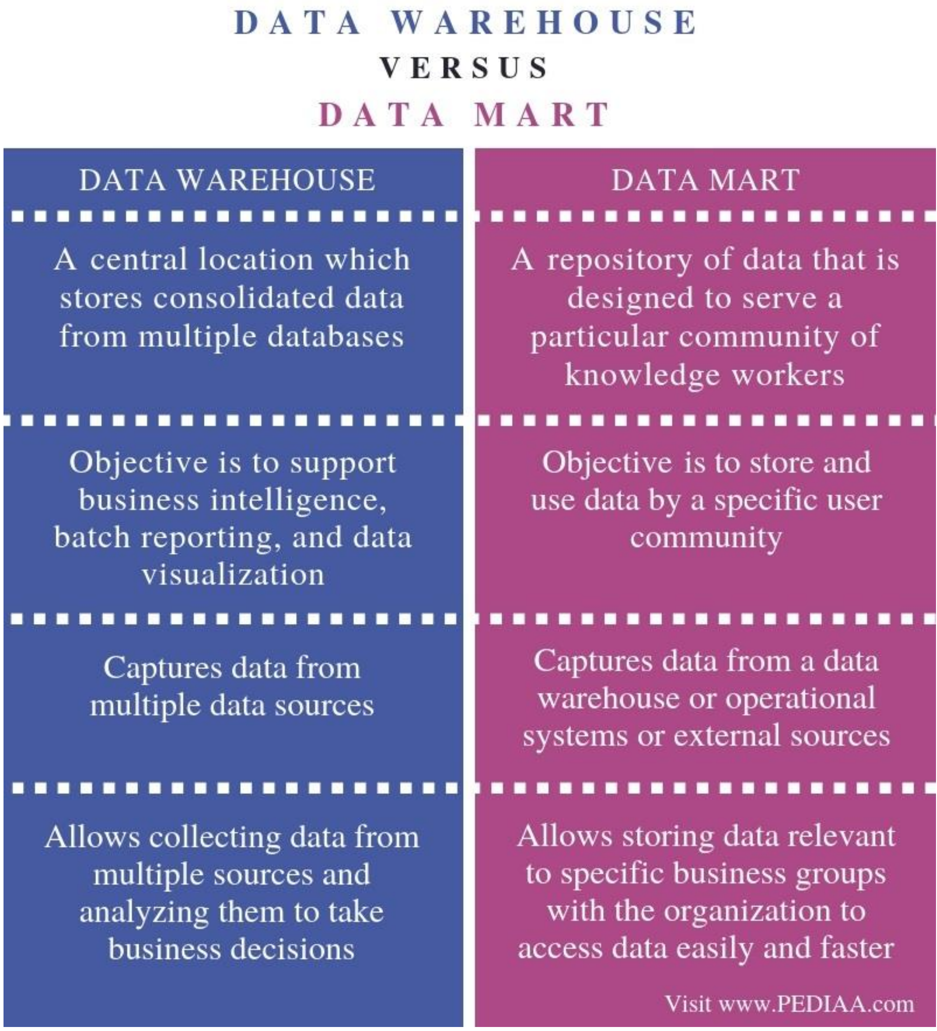
DATA MARTS:

Data Marts are specialized subsets of a data warehouse that focus on specific business areas or departments within an organization. Unlike a data warehouse, which serves as a central repository for all organizational data, data marts are designed to cater to the needs of particular user groups or functions, such as sales, finance, or marketing.

Data marts contain data relevant to a specific domain, enabling more focused and efficient querying and reporting. They simplify access to data for users by providing a streamlined view of information pertinent to their particular business area. For instance, a sales data mart might include sales transactions, customer information, and product details, tailored to support sales analysis and reporting.

Data marts can be either dependent or independent. A **dependent data mart** derives its data from a central data warehouse, ensuring consistency with the enterprise-wide data. An **independent data mart** is created directly from operational systems or external sources and does not rely on the central data warehouse. **Hybrid Data Mart**: Data fed from

both OLTP source and DWH



DATA WAREHOUSE
VERSUS
DATA MART

DATA WAREHOUSE	DATA MART
A central location which stores consolidated data from multiple databases	A repository of data that is designed to serve a particular community of knowledge workers
Objective is to support business intelligence, batch reporting, and data visualization	Objective is to store and use data by a specific user community
Captures data from multiple data sources	Captures data from a data warehouse or operational systems or external sources
Allows collecting data from multiple sources and analyzing them to take business decisions	Allows storing data relevant to specific business groups with the organization to access data easily and faster

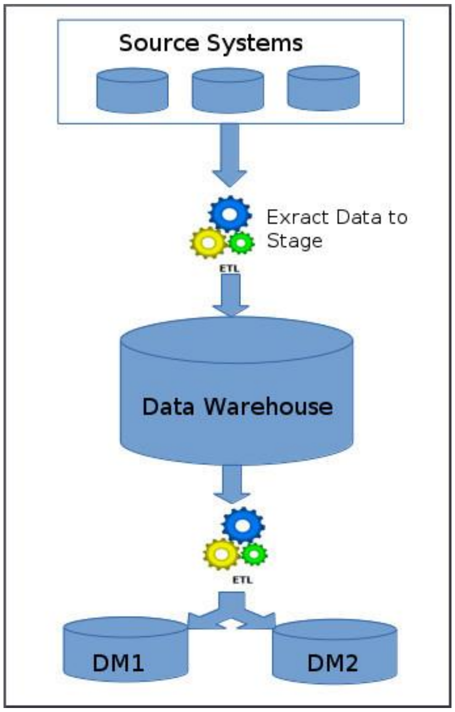
Visit www.PEDIAA.com

c

TOP-DOWN APPROACH IN DWM:

In the top-down approach, the data warehouse is designed first and then data mart are built

- **Data Extraction and ETL:**
    - Data is extracted from various source systems, such as operational databases and external data feeds, using ETL (Extract, Transform, Load) tools. This data is then validated to ensure its accuracy and consistency before being loaded into the central data warehouse.
- **Data Aggregation and Summarization:**
    - Once in the data warehouse, the data undergoes aggregation and summarization. Techniques are applied to transform the raw data into summarized formats that support complex querying and analysis. This step involves creating aggregates and summary tables that optimize performance and facilitate business insights.
- **Creation of Data Marts:**
    - After aggregation and summarization are completed, data marts are created. These are specialized subsets of the central data warehouse, tailored to specific business areas or departments. Data marts extract the necessary data from the central warehouse and apply additional transformations to meet the specific needs of their users. This may involve further data restructuring to align with the particular requirements of each data mart.

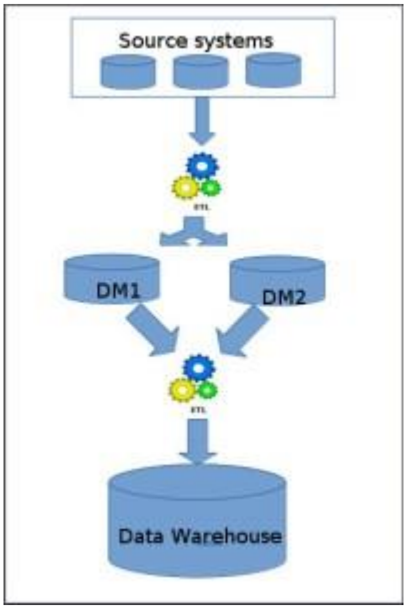


BOTTTOM-UP APPROACH IN DWM:

The Bottom-Up Approach begins by creating data marts that cater to specific business areas or departments. Over time, these individual data marts are integrated into a comprehensive data warehouse that provides a unified view of the organization’s data.

Process

1. **Development of Data Marts:**
    - **Identify Requirements:** Start by identifying the analytical needs of specific business units or departments. Develop data marts that are tailored to these needs.
    - **ETL Processes:** Extract, transform, and load (ETL) data from operational systems into the data marts. This includes cleansing, aggregating, and structuring data to meet the requirements of the specific business area.
2. **Integration into Central Data Warehouse:**
    - **Central Data Warehouse Design:** As individual data marts are developed, plan and build a central data warehouse that integrates these marts. This involves consolidating data from various marts to create a comprehensive and consistent repository.
3. **Enterprise-Wide Data Analysis:**
    - **Unified Data Access:** Once the central data warehouse is established, it provides a unified view of the organization’s data. This allows for enterprise-wide analysis and reporting, drawing insights from across different business areas.



What is Dimensional Model?

● A dimensional model is a data structure technique optimized for Data warehousing tools.

● The concept of Dimensional Modelling was developed by Ralph Kimball and is comprised of "fact" and "dimension" tables.

● A Dimensional model is designed to read, summarize, analyze numeric information like values, balances, counts, weights, etc. in a data warehouse.

● In contrast, relational models are optimized for addition, updating and deletion of data in a real-time Online Transaction System.

● ER modeling is for reducing redundancy of data, where as dimensional model arranges data in such a way that it is easier to retrieve information and generate reports

● These dimensional and relational models have their unique way of data storage that has specific advantages.

● Dimensional models are used in data warehouse systems and not a good fit for relational systems

Dimensional Data Model

The **Dimensional Data Model** is designed to organize data in a way that facilitates efficient querying and analysis. The core elements of this model are:

1. Fact Tables

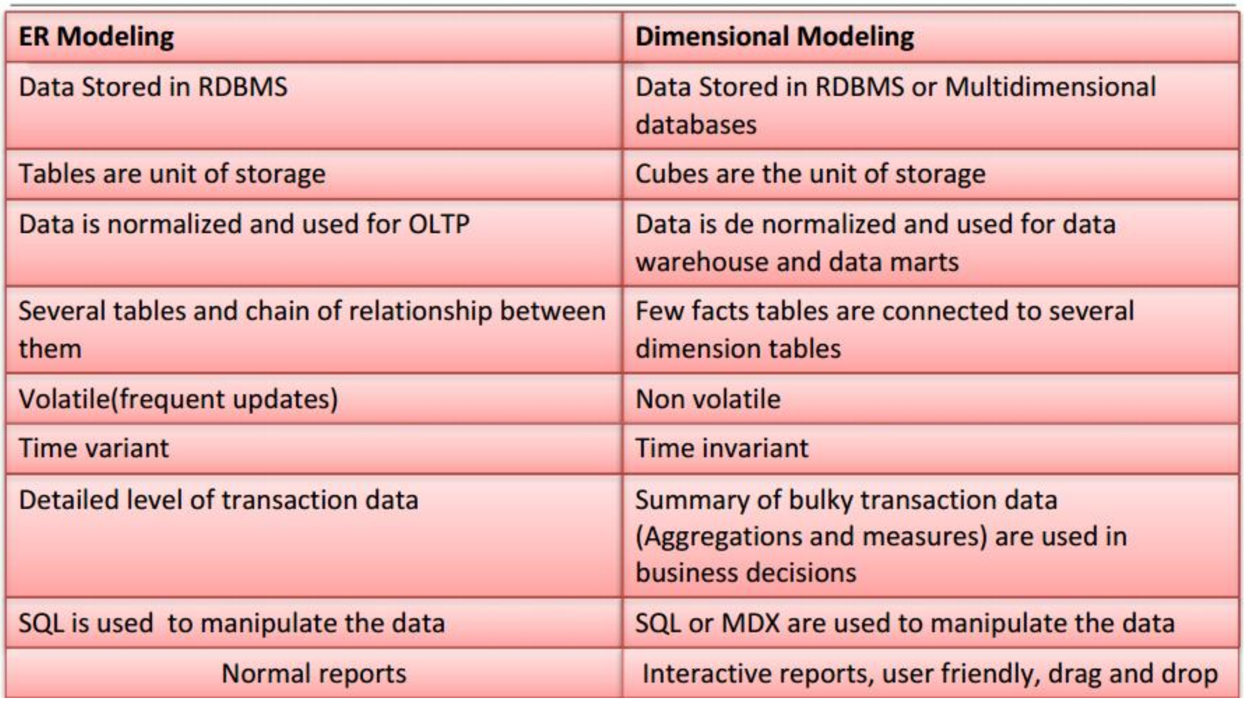
- **Definition:** Central tables in the dimensional model that store quantitative data or metrics related to business processes.
    - **Foreign Keys:** References to dimension tables to provide context for the measures (e.g., product_id, date_id).

2. Dimension Tables

- **Definition:** Tables that provide descriptive context to the measures stored in fact tables. They contain attributes related to the dimensions.
- **Contents:**
    - **Attributes:** Descriptive data that provide detail and context, such as product_name, category, date, customer_name, and region.

Attributes

The Attributes are the various characteristics of the dimension in dimensional data modeling.



ER Modeling	Dimensional Modeling
Data Stored in RDBMS	Data Stored in RDBMS or Multidimensional databases
Tables are unit of storage	Cubes are the unit of storage
Data is normalized and used for OLTP	Data is de normalized and used for data warehouse and data marts
Several tables and chain of relationship between them	Few facts tables are connected to several dimension tables
Volatile(frequent updates)	Non volatile
Time variant	Time invariant
Detailed level of transaction data	Summary of bulky transaction data (Aggregations and measures) are used in business decisions
SQL is used to manipulate the data	SQL or MDX are used to manipulate the data
Normal reports	Interactive reports, user friendly, drag and drop

STAR SCHEMA

A **Star Schema** is a type of database schema that is used in data warehousing to simplify and optimize the organization of data for querying and reporting. It is called a "star" schema because of its structure: a central fact table is surrounded by multiple dimension tables, and when visualized, it resembles a star.

The central fact table in a star schema contains the core quantitative data or metrics related to business processes, such as sales figures, transaction amounts, or other measurable data. This fact table typically includes foreign keys that link to dimension tables, which provide descriptive context for the data stored in the fact table.

The dimension tables in a star schema hold attributes related to the business entities that are being measured. For example, a sales star schema might have dimension tables for Product, Customer, Time, and Location, each containing attributes like product name, customer demographics, date of sale, and store location. These dimension tables are usually denormalized, meaning they store redundant data to simplify the structure and make querying faster and easier.

The simplicity and straightforward design of the star schema make it easy for end-users to understand and use

SNOWFLAKE SCHEMA

A **Snowflake Schema** is a more complex variation of the star schema used in data warehousing, designed to further normalize the data structure. Unlike the star schema, where dimension tables are typically denormalized (with redundant data), the snowflake schema breaks down dimension tables into multiple related tables, creating a more intricate, snowflake-like structure.

In a snowflake schema, the central fact table remains the core component, containing quantitative data or metrics related to business processes, such as sales totals, transaction counts, or other key performance indicators. However, the dimension tables that provide descriptive context to the data in the fact table are normalized into multiple related tables.

For example, in a snowflake schema for sales data, instead of having a single Product dimension table with attributes like product name, category, and brand, these attributes might be split into separate tables. You might have a Product table with just a product ID and product name, a Category table linked to the Product table that lists product categories, and a Brand table linked to the Product table that lists brands. Each of these tables can be linked back to the fact table through a series of foreign key relationships.

GALAXY SCHEMA

A **Galaxy Schema**, also known as a **Fact Constellation Schema**, is a complex data warehousing schema that consists of multiple fact tables sharing common dimension tables. This schema is essentially a combination of multiple star schemas and is used to model sophisticated and interrelated business processes within an organization.

Structure:

- **Fact Tables:** The galaxy schema includes two or more fact tables that represent different business processes or subjects. Each fact table stores quantitative data or metrics, such as sales amounts, inventory levels, or customer transactions.
- **Shared Dimension Tables:** The fact tables in a galaxy schema are linked to shared dimension tables. These dimension tables contain descriptive attributes that provide context to the facts, such as time, product, customer, or location data. By sharing dimensions, the schema allows for the integration of related facts, enabling complex analysis across different business processes.
- **Example:**
    - In a retail environment, one fact table might track sales (Sales Fact), while another tracks inventory levels (Inventory Fact). Both fact tables might share dimension tables for Product, Time, and Store. The Sales Fact table could have measures like

sales_amount and units_sold, while the Inventory Fact table might track inventory_count and reorder_level.

Advantages:

- **Data Reusability:** Since dimension tables are shared among multiple fact tables, the schema reduces data redundancy and ensures consistency across different analyses.
- **Scalability:** The galaxy schema can scale well as it allows the addition of new fact tables and dimensions without disrupting the existing schema.

FACTLESS FACT TABLE

A **Factless Fact Table** is a type of fact table in a data warehouse that does not contain any measurable, quantitative data (or "facts"). Instead, it is used to capture the occurrence of events or the relationships between dimensions. Despite the absence of numeric data, factless fact tables play a crucial role in certain types of analyses.

Types of Factless Fact Tables

1. **Event Tracking:**
   - **Purpose:** Captures the occurrence of events without any associated measures. It records that an event happened, rather than how much or how many.
   - **Example:** A factless fact table might be used to track student attendance. The table could include foreign keys linking to Student, Class, and Date dimension tables, simply indicating that a particular student attended a particular class on a specific date, without any numeric data.
2. **Coverage or Eligibility:**
   - **Purpose:** Indicates a potential event or the possibility of something happening, such as the availability or coverage of a particular option.
   - **Example:** In a retail setting, a factless fact table might indicate which products are available in which stores. The table would contain foreign keys linking to Product and Store dimensions, showing the possible combinations without recording any sales data.

Structure:

- **Foreign Keys:** The table is composed primarily of foreign keys that link to dimension tables. These keys represent the different entities involved in the event or relationship.
- **No Measures:** Unlike regular fact tables, there are no numerical measures or metrics in a factless fact table.

Online Analytical Processing (OLAP)

Online Analytical Processing (OLAP) is a category of data processing that enables users to interactively analyze multidimensional data from multiple perspectives. Unlike Online Transaction Processing (OLTP), which focuses on managing and processing daily transactional data, OLAP is designed to facilitate complex queries, data analysis, and reporting.

In OLAP, data is organized into multidimensional structures called **cubes**, where each dimension represents a different aspect of the data, such as time, geography, or product categories. This multidimensional approach allows users to perform sophisticated analyses, such as slicing and dicing the data, drilling down to more detailed levels, and rolling up to summarize information. For example, a sales OLAP cube might allow users to view sales data by product, region, and time period, and to explore trends or patterns across these dimensions.

OLAP supports

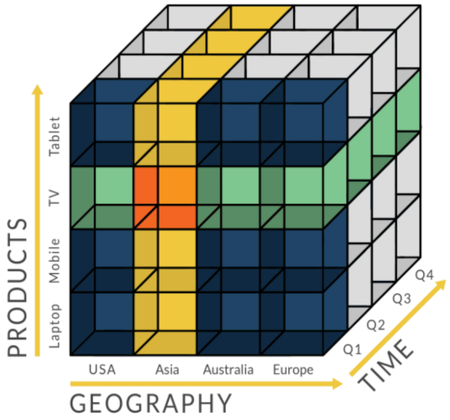
– Business analysis queries

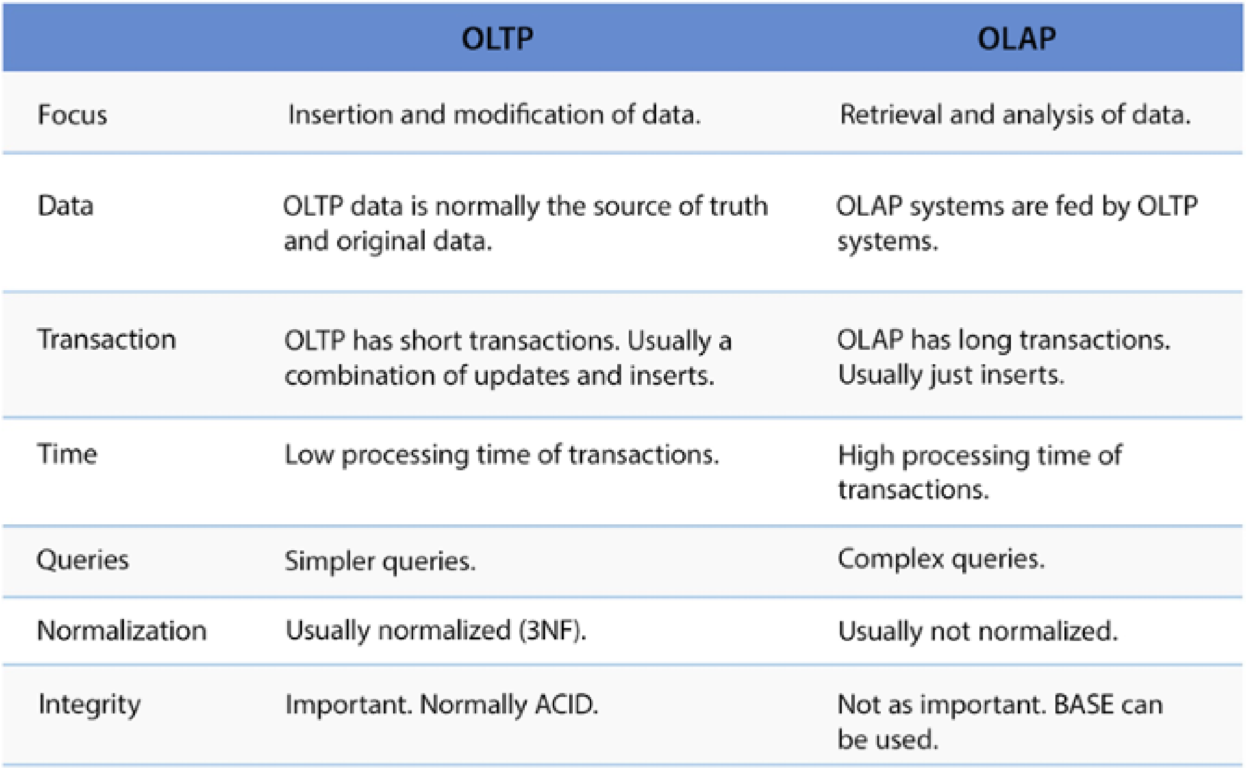
– Data visualization

– Trend analysis

– Scenario analysis

– User defined queries





	OLTP	OLAP
Focus	Insertion and modification of data.	Retrieval and analysis of data.
Data	OLTP data is normally the source of truth and original data.	OLAP systems are fed by OLTP systems.
Transaction	OLTP has short transactions. Usually a combination of updates and inserts.	OLAP has long transactions. Usually just inserts.
Time	Low processing time of transactions.	High processing time of transactions.
Queries	Simpler queries.	Complex queries.
Normalization	Usually normalized (3NF).	Usually not normalized.
Integrity	Important. Normally ACID.	Not as important. BASE can be used.

OLAP (Online Analytical Processing) Operations

In OLAP (Online Analytical Processing), several key operations enable users to interactively analyze multidimensional data and derive insights. These operations are designed to facilitate complex queries and analytical tasks by leveraging the multidimensional structure of data. Here’s a brief overview of the primary OLAP operations:

1. Slicing

- **Definition:** The process of selecting a single dimension from a multidimensional cube to create a new, lower-dimensional view of the data.
- **Example:** In a sales data cube, slicing might involve selecting data for a specific year, which results in a two-dimensional view of sales by product and region for that year only.

2. Dicing

- **Definition:** The process of selecting a subset of dimensions and filtering data within those dimensions to create a sub-cube.
- **Example:** Dicing might involve selecting data for a specific combination of product category and region, providing a focused view of sales within those criteria.

3. Drilling Down

- **Definition:** The process of navigating from more summarized data to more detailed data within the same dimension. This operation increases the granularity of the data.
- **Example:** Drilling down from annual sales figures to monthly sales figures, allowing users to examine data at a finer level of detail.

4. Rolling Up

- **Definition:** The process of aggregating data from a detailed level to a more summarized level within the same dimension. This operation decreases the granularity of the data.
- **Example:** Rolling up from monthly sales figures to annual sales figures, summarizing data to provide a broader view of performance over a longer period.

5. Pivoting (or Rotation)

- **Definition:** The process of rotating the data axes to view the data from different perspectives. This operation allows users to reorganize and reorient the data for better insight.
- **Example:** Pivoting might involve changing the view from sales by product and time to sales by region and product, allowing users to analyze the data from different angles.

6. Aggregation

- **Definition:** The process of summarizing data at various levels of granularity. Aggregation involves calculating total values, averages, or other metrics based on the data in the cube.
- **Example:** Aggregating daily sales data to calculate monthly or quarterly sales totals.

Different OLAP (Online Analytical Processing) Operations

1. MOLAP (Multidimensional OLAP)

- **Definition:** MOLAP uses a multidimensional data model and stores data in pre-aggregated and summarized formats within OLAP cubes. This approach allows for fast query performance and efficient data retrieval.
- **Features:**
  - **Data Storage:** Data is stored in a multidimensional cube format.
  - **Performance:** High performance due to pre-computed aggregations and optimizations.

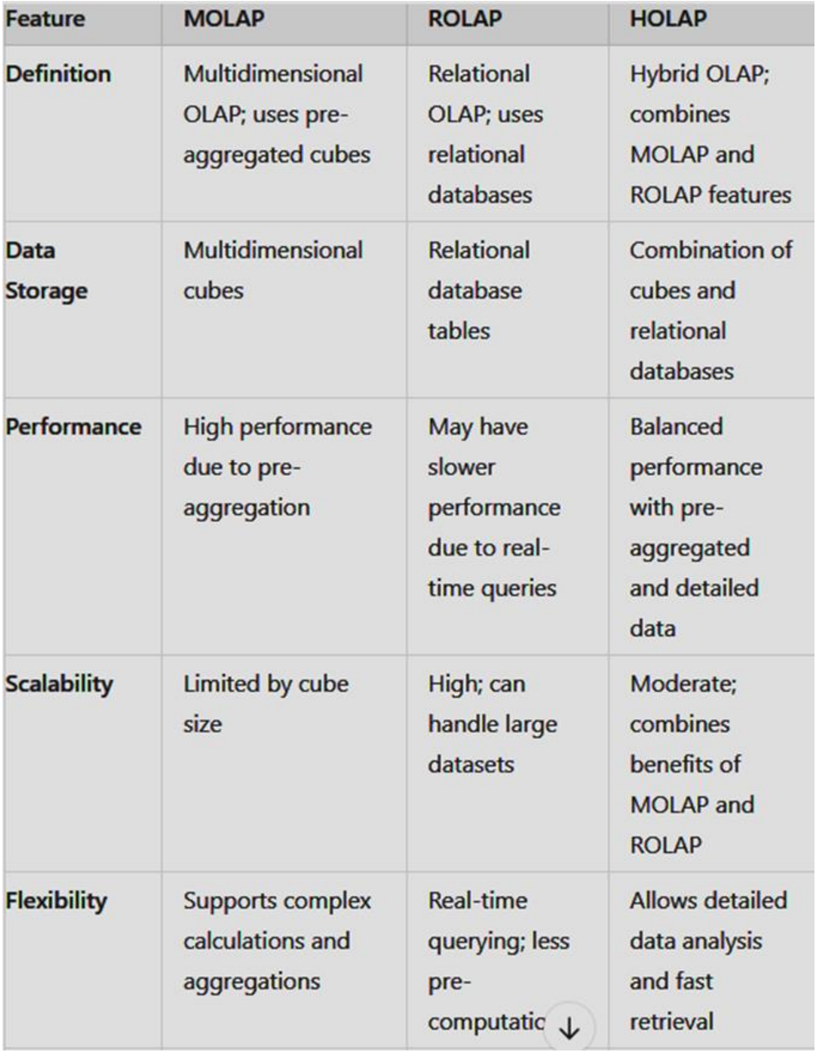
    - **Flexibility:** Supports complex calculations and aggregations.
- **Examples:** Microsoft Analysis Services (SSAS) and IBM Cognos TM1.

2. ROLAP (Relational OLAP)

- **Definition:** ROLAP uses relational database systems to store data and perform OLAP operations on the data in real-time. It does not pre-aggregate data but generates results dynamically by querying the underlying relational databases.
- **Features:**
    - **Data Storage:** Data is stored in relational database tables.
    - **Performance:** May have slower query performance compared to MOLAP due to dynamic aggregation and real-time queries.
    - **Scalability:** Can handle large volumes of data without the need for extensive pre-aggregation.
- **Examples:** Oracle OLAP, IBM Db2 Warehouse.

3. HOLAP (Hybrid OLAP)

- **Definition:** HOLAP combines aspects of both MOLAP and ROLAP. It stores detailed data in relational databases and aggregated data in multidimensional cubes, aiming to leverage the strengths of both models.
- **Features:**
    - **Data Storage:** Uses relational databases for detailed data and multidimensional cubes for aggregated data.
    - **Performance:** Offers a balance between high performance for pre-aggregated data and flexibility for detailed data analysis.
    - **Flexibility:** Allows for complex analysis while maintaining efficient data retrieval.
- **Examples:** Microsoft SQL Server Analysis Services (SSAS) in HOLAP mode.



Feature	MOLAP	ROLAP	HOLAP
Definition	Multidimensional OLAP; uses pre-aggregated cubes	Relational OLAP; uses relational databases	Hybrid OLAP; combines MOLAP and ROLAP features
Data Storage	Multidimensional cubes	Relational database tables	Combination of cubes and relational databases
Performance	High performance due to pre-aggregation	May have slower performance due to real-time queries	Balanced performance with pre-aggregated and detailed data
Scalability	Limited by cube size	High; can handle large datasets	Moderate; combines benefits of MOLAP and ROLAP
Flexibility	Supports complex calculations and aggregations	Real-time querying; less pre-computatic	Allows detailed data analysis and fast retrieval

ETL (Extract, Transform, Load) is a critical process in data warehousing and integration that involves three main stages: extraction, transformation, and loading. This process is designed to consolidate data from various sources into a centralized repository, such as a data warehouse, to facilitate comprehensive analysis and reporting.

1. **Extract:** In the extraction phase, data is gathered from diverse source systems, which may include databases, flat files, APIs, or external data sources. The goal is to collect raw data from these disparate sources efficiently and accurately. Extraction methods are often designed to handle data in various formats and structures, ensuring that the necessary data is retrieved without altering its original form.

2. **Transform:** Once the data is extracted, it undergoes the transformation phase. This stage involves cleaning, standardizing, and converting the data into a format suitable for analysis. Transformations may include data cleansing (e.g., removing duplicates or correcting errors), data enrichment (e.g., adding missing information), data aggregation

(e.g., summarizing data), and applying business rules to ensure consistency and accuracy. The transformed data is then structured in a way that aligns with the requirements of the target system, such as a data warehouse.

3. **Load:** The final phase is loading, where the transformed data is inserted into the target data repository, such as a data warehouse or data mart. The loading process involves transferring the data into the appropriate tables and structures within the target system. This phase ensures that the data is available for querying, reporting, and analysis by end-users.

The ETL process is essential for integrating and preparing data from various sources to support business intelligence, reporting, and decision-making. It helps in creating a unified view of data, ensuring that it is accurate, consistent, and readily accessible for analytical purposes.

Primary Keys

- **Definition:** A primary key is a unique identifier for a record in a database table. Each table in a relational database must have a primary key to ensure that each row in the table can be uniquely identified.
- **Characteristics:**
  - **Uniqueness:** The primary key must contain unique values; no two rows can have the same primary key value.
  - **Non-Null:** The primary key cannot have null values, ensuring that every record has a valid identifier.
  - **Stability:** Ideally, the primary key should be stable and not change over time.
- **Example:** In a Customer table, the CustomerID column might serve as the primary key, uniquely identifying each customer.

Surrogate Keys

- **Definition:** A surrogate key is an artificial or synthetic key created to uniquely identify a record in a table. It is not derived from the business data but is instead generated by the database system or application.
- **Characteristics:**
  - **Generated:** Surrogate keys are usually sequential numbers or unique identifiers generated by the system (e.g., auto-increment integers or UUIDs).

    - **No Business Meaning:** Unlike natural keys, surrogate keys do not have any intrinsic meaning related to the business data.
    - **Ease of Use:** They simplify the process of managing and linking records, especially in data warehousing and integration scenarios.
- **Example:** A Product table might use a surrogate key like ProductID that is an auto-incrementing integer, regardless of the product's actual attributes.

Foreign Keys

- **Definition:** A foreign key is a column or set of columns in a table that establishes a link between that table and another table. It refers to the primary key of another table, creating a relationship between the two tables.
- **Characteristics:**
    - **Referential Integrity:** Foreign keys ensure referential integrity by enforcing that the value in the foreign key column matches a valid primary key value in the referenced table.
    - **Relationships:** They are used to define relationships between tables, such as one-to-many or many-to-many relationships.
    - **Constraints:** Foreign keys can enforce constraints that prevent invalid data from being entered into the table.
- **Example:** In an Order table, the CustomerID column might be a foreign key referencing the CustomerID primary key in the Customer table. This relationship links each order to a specific customer.