Unit 4: Data Warehouse and OLAP

Comprehensive Overview of Data Warehousing and Analytical Processing

Sunil Regmi, Lecturer

June 4, 2025

Outline

1. Data Warehouse Architecture
2. ETL vs ELT
3. Understanding Data Mart and Data Lake
4. Data Warehouse vs Data Mart vs Data Lake
5. OLTP vs OLAP
6. OLAP Operations
7. Relational OLAP (ROLAP)
8. Multidimensional OLAP (MOLAP)
9. Hybrid OLAP (HOLAP)
10. Relational OLAP: Schema Types
11. Multidimensional Data Model and MDX

Data Warehouse

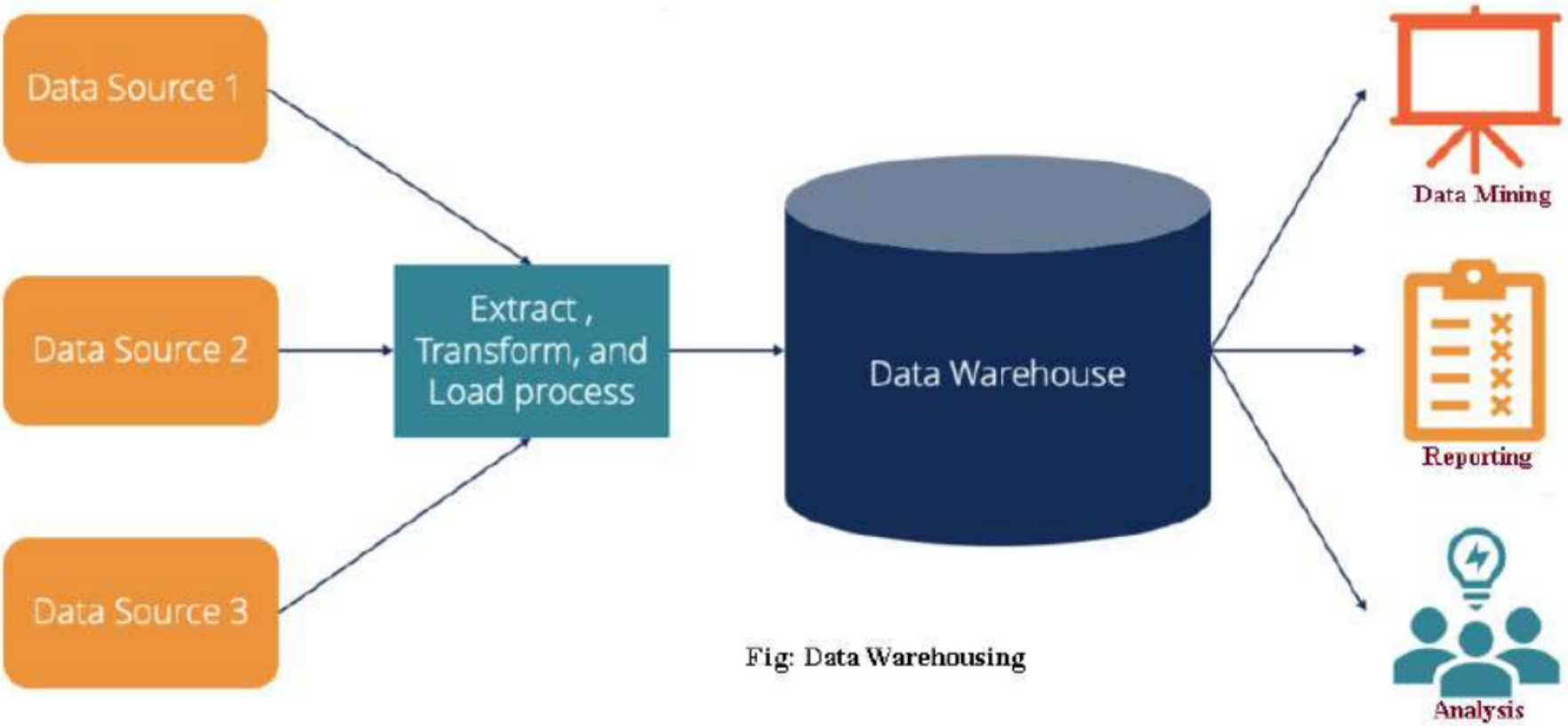


Fig: Data Warehousing

Data Warehouse Architecture: Overview

- A **data warehouse** is a centralized repository for storing large volumes of data from multiple sources, optimized for analysis and reporting.
- Purpose: Support **business intelligence (BI)** activities like querying, reporting, and data analysis.
- Key characteristics:
  - **Subject-oriented**: Focused on specific business areas (e.g., sales, inventory).
  - **Integrated**: Combines data from diverse sources into a consistent format.
  - **Time-variant**: Stores historical data for trend analysis.
  - **Non-volatile**: Data is stable, not updated in real-time.
- **Example**: A retail company analyzing sales trends across regions over years.

Functions of Data Warehousing

- ***Data Extraction*:** Involves gathering data from multiple heterogeneous sources.
- ***Data Cleaning:*** Involves finding and correcting the errors in data.
- ***Data Transformation:*** Involves converting the data from legacy format to warehouse format.
- ***Data Loading:*** Involves sorting, summarizing, consolidating, checking integrity, and building indices and partitions.
- ***Refreshing:*** Involves updating from data sources to warehouse.

Operational Database System (OLTP System)	*Data Warehouse (OLAP System)*
Operational system are generally designed to support high-volume transaction processing.	Data warehousing systems are generally designed to support high volume analytical processing. (i.e. OLAP)
It is used for day-to-day operations.	It is used for long-term informational requirements and decision support.
Operational data are the original sources of the data.	Data comes from various OLTP Databases.
In operational system data is stored with a functional or process orientation.	In data warehousing systems data is stored with a subject orientation.
It provides detailed and flat relational view of data.	It provides summarized and multidimensional view of data.
It focuses on "Data In".	It focuses on Information out.
The tables and joins are complex since they are normalized (for RDMS). This is done to reduce redundant data and to save storage space.	The tables and joins are simple since they are de-normalized. This is done to reduce the response time for analytical queries.

Figure: ODA and Data warehouse

Entity-Relationship modelling techniques are used for RDMS database design.	Data-Modeling techniques are used for the Data Warehouse design.
Performance is low for analysis queries.	High performance for analytical queries.
Data within operational systems are generally updated regularly.	Data within a data warehouse is non-volatile, meaning when new data is added old data is not erased so rarely updates.
Data volumes are less and historical data is generally not maintained.	It involves large data volumes and historical data.
Simple queries are capable of fetching the data.	Complex queries are required to fetch data.
Processing speed is fast.	Processing speed is slow because of large size.
The common users are clerk, DBA, database professional.	The common users are knowledge worker (e.g. manager, executive, analyst)

Figure: ODA and Data warehouse

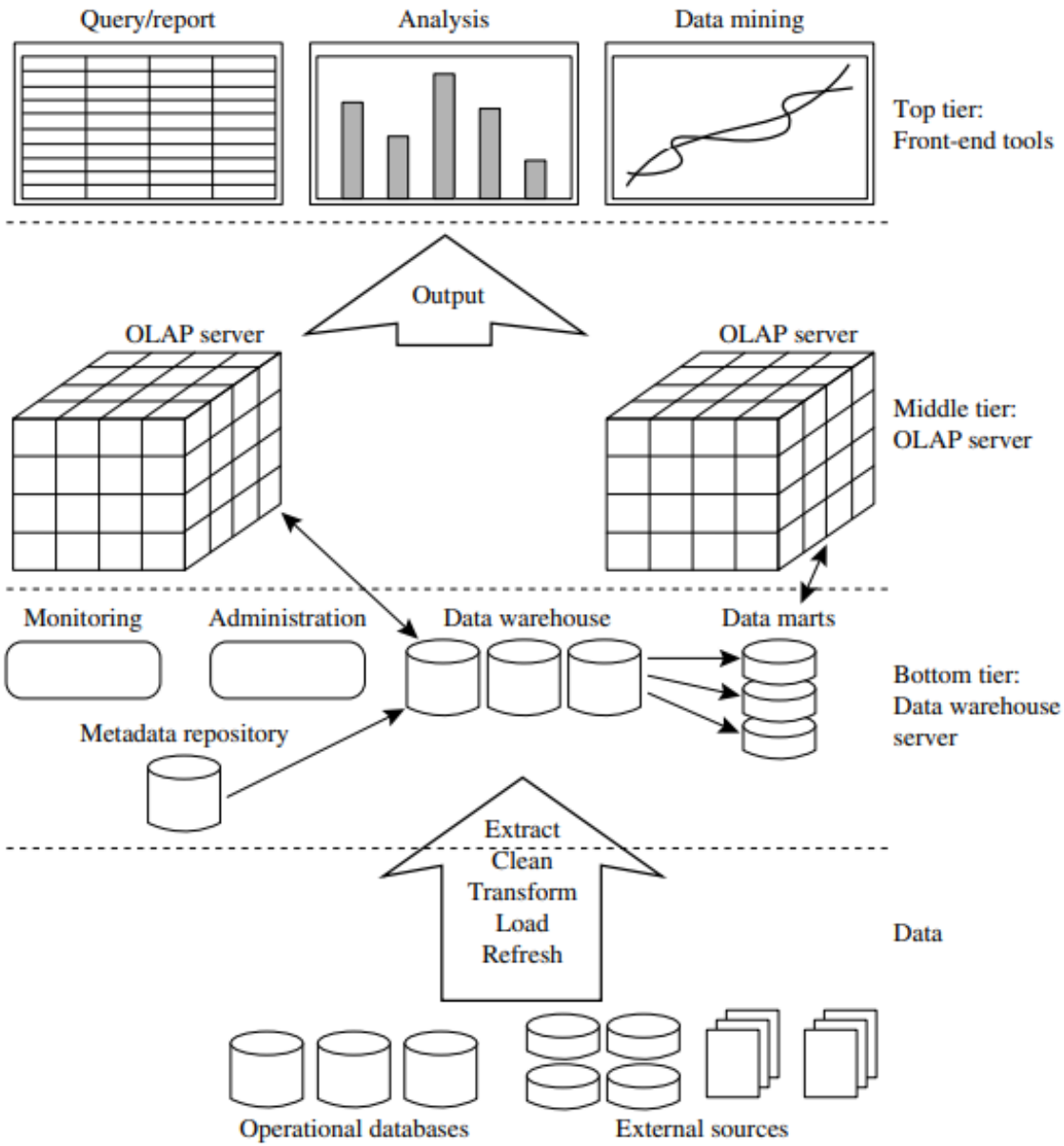
Components of Data Warehouse Architecture

- **Data Sources**: Operational databases (e.g., CRM, ERP), flat files, external data (e.g., APIs, IoT).
- **ETL/ELT Process**: Extracts data, transforms it (cleaning, aggregating), and loads it into the warehouse.
- **Data Storage**: Centralized repository with historical and aggregated data.
- **Metadata Repository**: Stores data about data (e.g., schema, source, transformations).
- **Query and Analysis Tools**: BI tools (e.g., Tableau, Power BI), reporting tools, dashboards.
- **End Users**: Analysts, data scientists, business managers.

Types of Data Warehouse Architectures

- **Single-Tier Architecture**:
  - Single layer for data storage and access.
  - **Example**: Small business storing sales data in a single database.
  - Limitation: Limited scalability for large datasets.
- **Two-Tier Architecture**:
  - Separates warehouse (storage) and client tools (querying).
  - **Example**: Medium-sized firm with a warehouse and BI tools.
  - Limitation: Limited flexibility for complex analytical queries.
- **Three-Tier Architecture**:
  - Bottom Tier: Data warehouse database server.
  - Middle Tier: OLAP server for analytical processing.
  - Top Tier: Front-end tools for querying/reporting.
  - **Example**: Enterprise system for global sales analysis.
  - Most common for large-scale systems.

Data Warehouse Three Architecture Diagram



Another way of representation

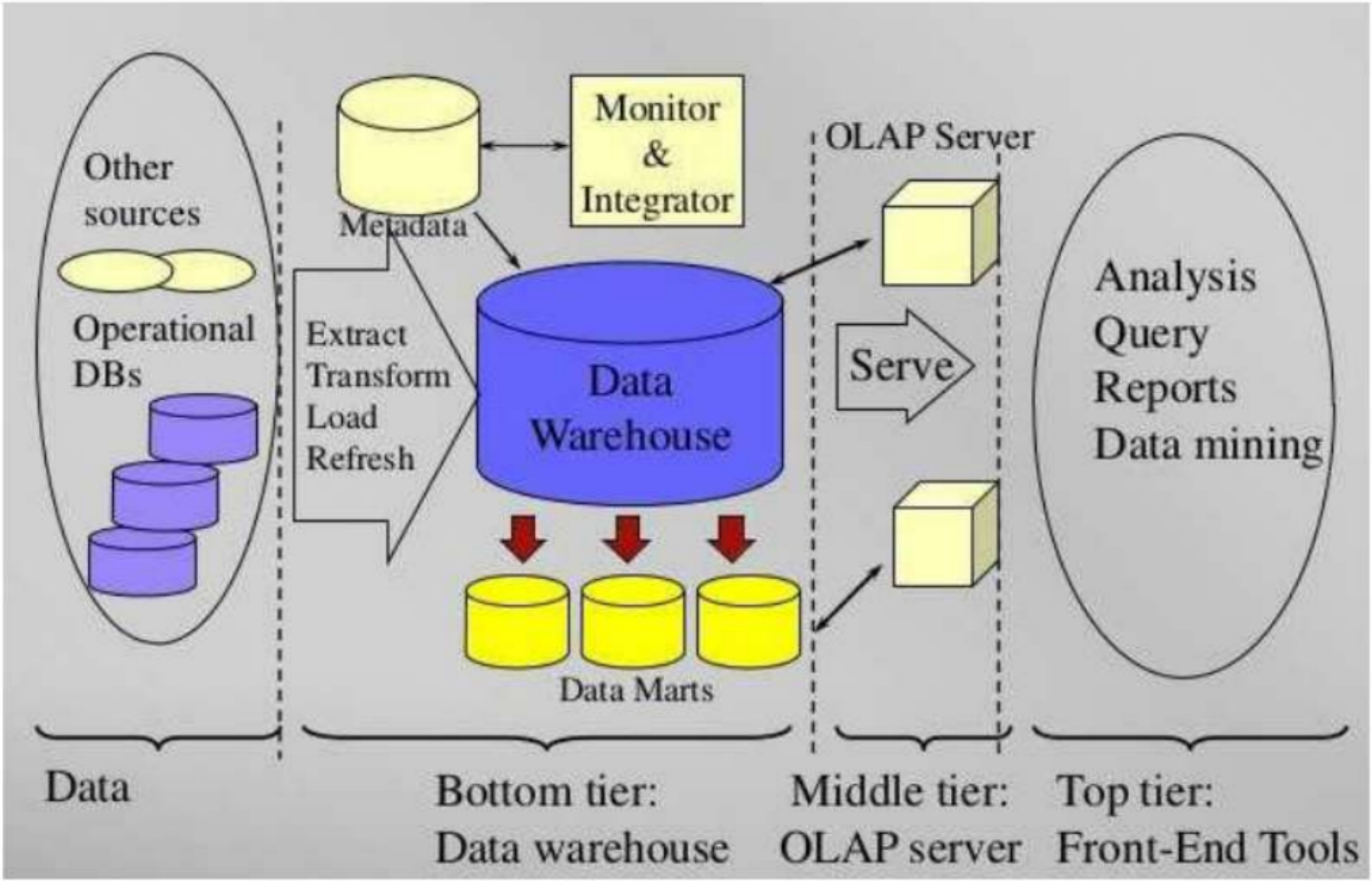


Fig: A three-tier data warehousing architecture.

ETL: Extract, Transform, Load

- **Extract**: Retrieve raw data from sources (e.g., SQL databases, CSV files, APIs).
- **Transform**: Clean, normalize, aggregate, enrich data to fit warehouse schema.
- **Load**: Store transformed data into the data warehouse.
- **Example**: Extract sales data from a CRM, transform dates to a standard format (e.g., YYYY-MM-DD), load into a warehouse.

Key Features

- Transformation occurs *before* loading.
- Ideal for structured data and traditional warehouses.
- Ensures data quality and consistency before storage.

ETL Process Diagram

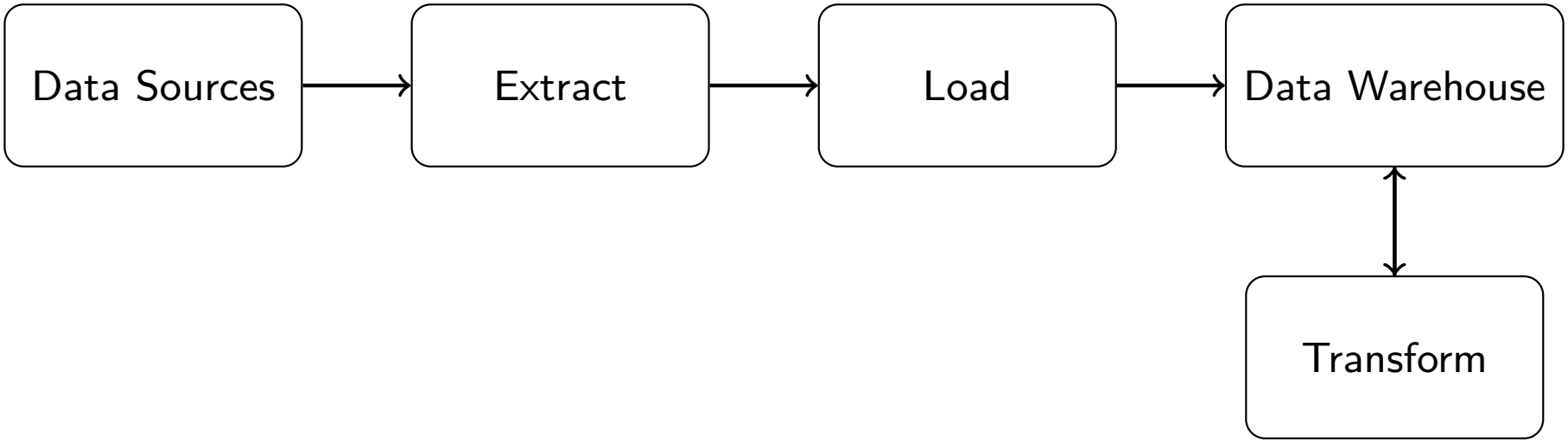
ELT: Extract, Load, Transform

- **Extract**: Retrieve raw data from sources.
- **Load**: Store raw data directly into the warehouse or data lake.
- **Transform**: Perform transformations within the warehouse using its compute power.
- **Example**: Load raw JSON data from IoT devices into Snowflake, then transform using SQL to aggregate daily metrics.

Key Features

- Transformation occurs *after* loading.
- Leverages cloud-based warehouse engines (e.g., Snowflake, BigQuery).
- Ideal for big data and unstructured/semi-structured data.

ELT Process Diagram



ETL vs ELT: Comparison

Criteria	ETL	ELT
Process Order	Extract, Transform, Load	Extract, Load, Transform
Transformation Location	External ETL server	Inside data warehouse
Data Volume	Smaller, structured data	Large, diverse data
Speed	Slower (pre-transformation)	Faster (direct loading)
Flexibility	Less flexible for raw data	Highly flexible for raw data
Tools	Informatica, Talend	Snowflake, Databricks
Cost	Higher for complex transformations	Lower with cloud scalability

When to Use ETL vs ELT

Use ETL When:

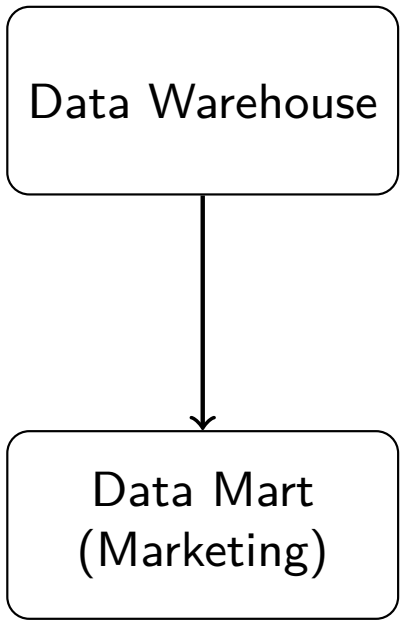
- Strict governance and data cleaning required before loading.
- On-premises data warehouses with limited compute power.
- Smaller datasets with well-defined schemas.
- **Example**: Financial institution cleaning sensitive customer data before loading.

Use ELT When:

- Handling large-scale, unstructured, or semi-structured data.
- Using cloud-based warehouses with high compute power.
- Need faster data availability for analysis.
- **Example**: E-commerce platform analyzing raw clickstream data in real-time.

Data Mart: Focused Analytical Storage

- **Definition**: A subset of a data warehouse, tailored for a specific business unit or department.
- **Characteristics**:
  - Contains structured, processed data.
  - Focused scope (e.g., marketing, finance).
  - Faster access for specific analytics.
- **Example**: A marketing data mart storing campaign performance data for analysis.



Types of Data Marts

Depending on the source of data, data marts can be categorized as dependent, independent, and hybrid.

- **Dependent data marts** draw data from central data warehouse that has already been created.
- **Independent data marts** are standalone systems built by drawing data directly from operational or external sources of data or both.
- **Hybrid data marts** can draw data from operational systems or data warehouses.

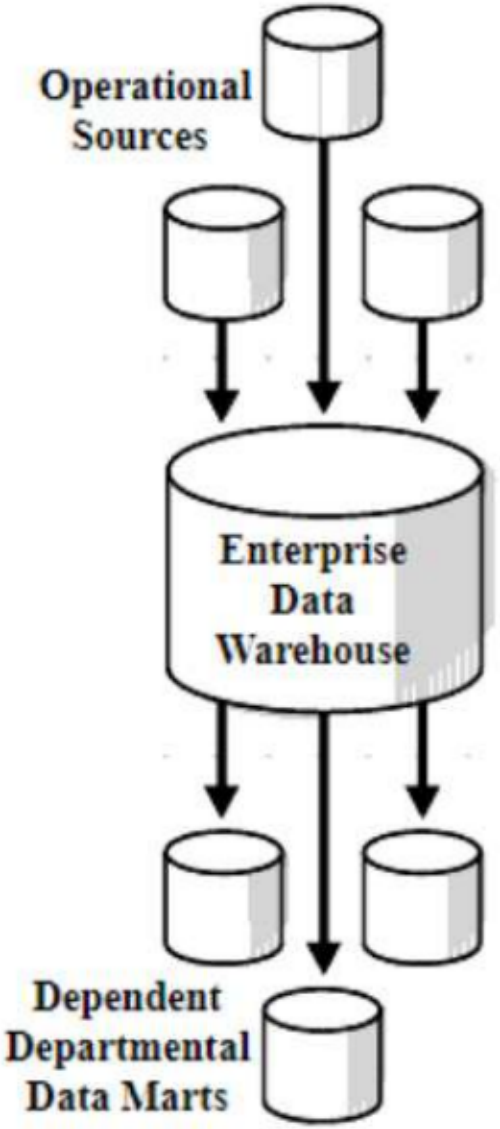


Fig: Dependent Data Mart

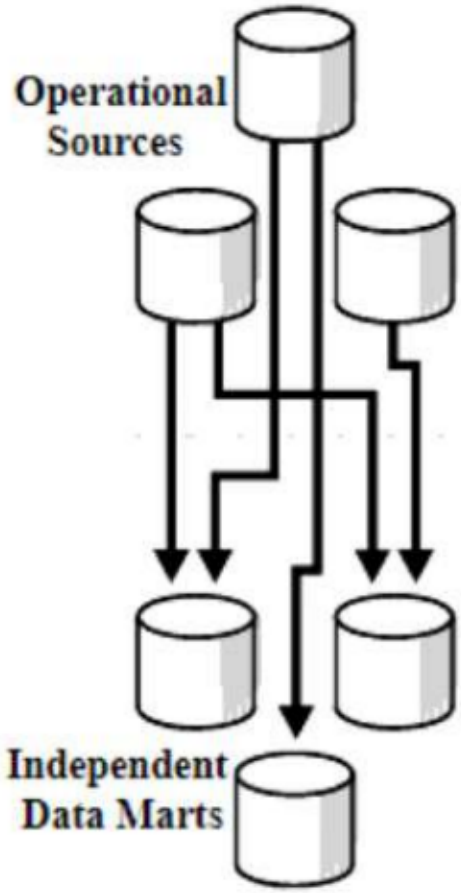


Fig: Independent Data Mart

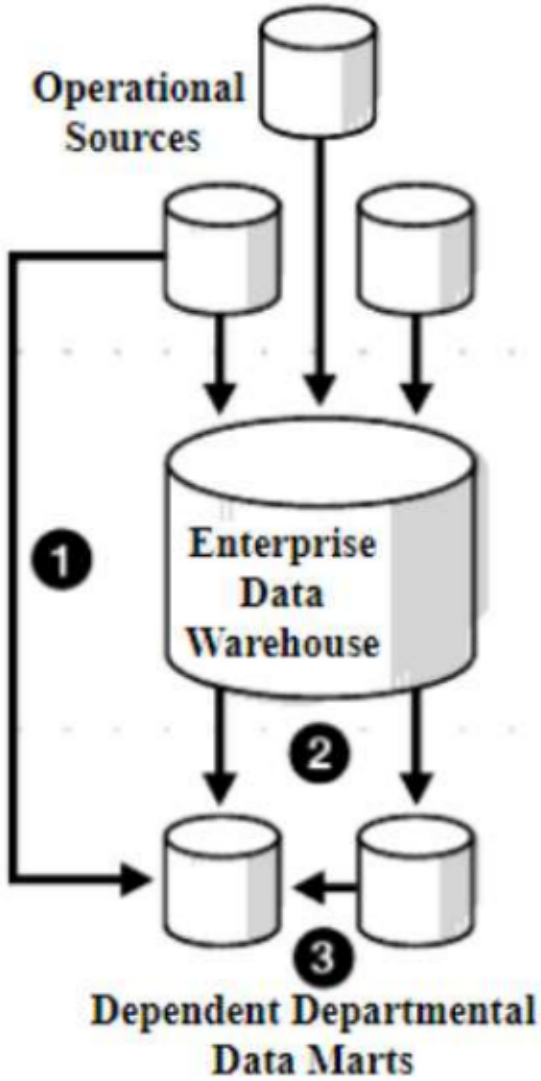
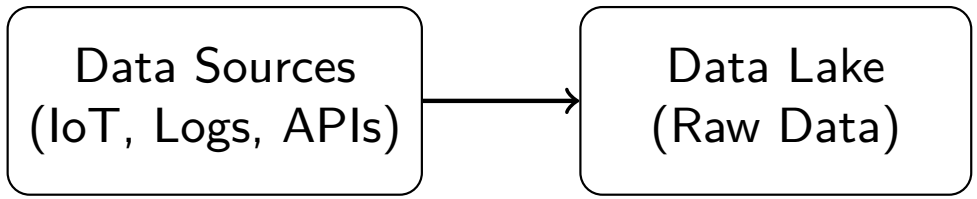


Fig: Hybrid Data Mart

Data Lake: Raw Data Repository

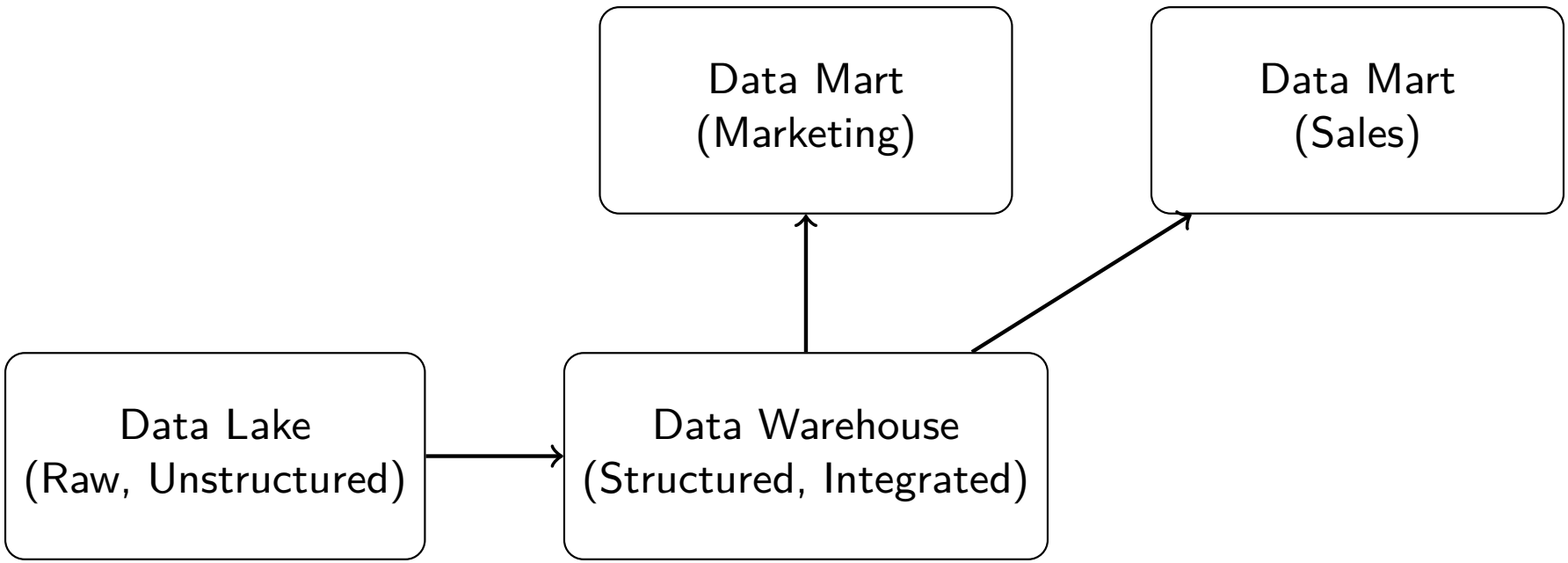
- **Definition**: A centralized repository for storing raw, structured, and unstructured data at scale.
- **Characteristics**:
  - Stores raw, unprocessed data.
  - Supports big data, AI, and machine learning.
  - Flexible schema-on-read approach.
- **Example**: Storing raw IoT sensor data for predictive maintenance models.



Data Warehouse vs Data Mart vs Data Lake

Criteria	Data Warehouse	Data Mart	Data Lake
Definition	Centralized repository for integrated, historical data	Subset of a data warehouse for a specific business unit	Repository for raw, structured, and unstructured data
Data Type	Structured, processed	Structured, processed	Raw, structured, and unstructured
Scope	Enterprise-wide	Department-specific	Enterprise-wide
Use Case	Business intelligence, reporting, and analytics	Focused analytics, such as marketing or finance	Big data, AI, and machine learning applications
Example	Company-wide sales analysis	Marketing teams campaign perfor-	Raw IoT sensor data for machine

Diagram: Data Warehouse, Data Mart, Data Lake



OLTP vs OLAP: Overview

- **OLTP (Online Transaction Processing)**:
  - Manages day-to-day transactional data.
  - Optimized for insert, update, delete operations.
  - **Example**: Processing customer orders in an e-commerce system (e.g., updating inventory after a purchase).
- **OLAP (Online Analytical Processing)**:
  - OLAP is based on multidimensional data model.
  - It allows managers and analysts to get an isnsight of the information through fast, consistent, and interactive access to information.
  - Supports complex queries and analysis on historical data.
  - Optimized for read-heavy operations.
  - **Example**: Analyzing sales trends across regions over multiple years.

OLTP vs OLAP: Comparison

Criteria	OLTP	OLAP
Purpose	Transaction processing	Analytical processing
Data Type	Current, detailed	Historical, aggregated
Query Type	Simple, frequent updates	Complex, read-heavy
Users	Operational staff	Analysts, managers
Example	ATM transactions	Sales forecasting

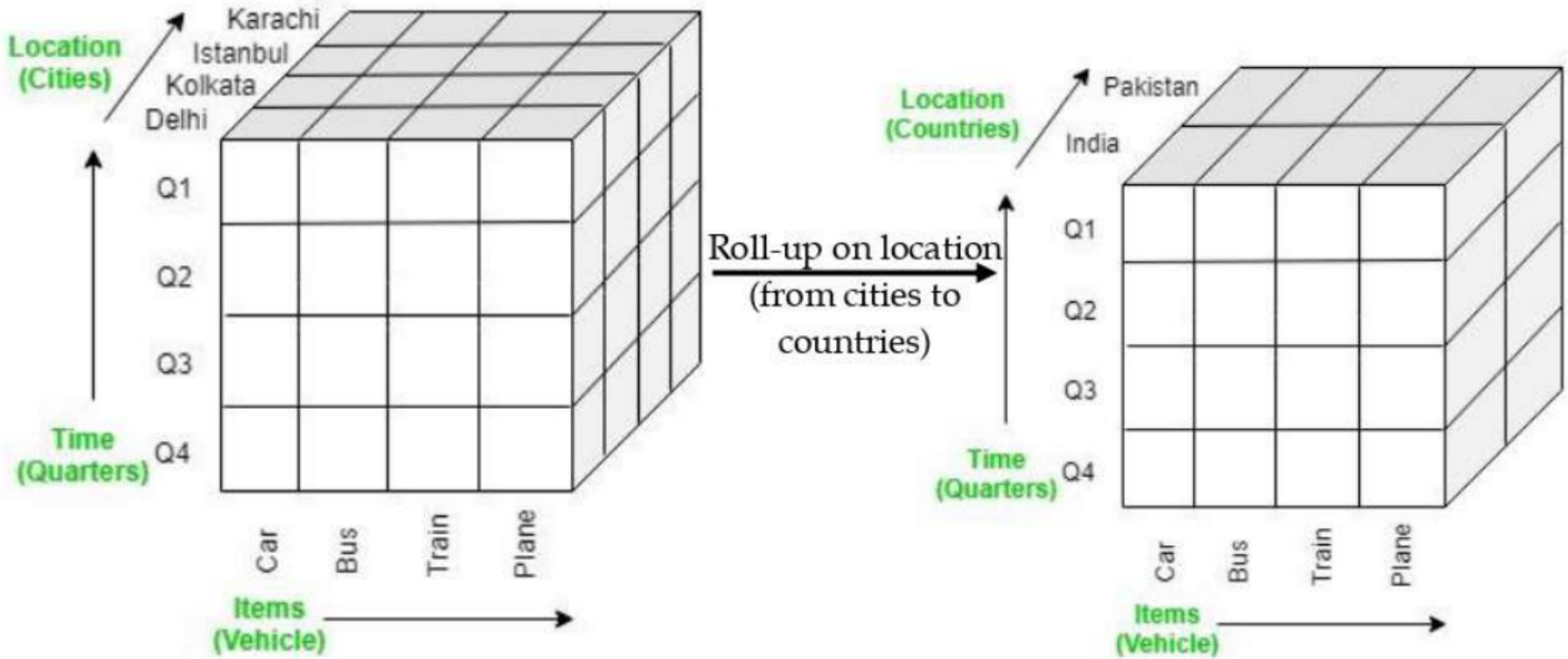
OLAP Operations: Overview

- OLAP enables multidimensional analysis of data using a data cube.
- Common operations:
  - **Roll-up**: Aggregate data to a higher level (e.g., sales by month to year).
  - **Drill-down**: Break down data to a lower level (e.g., sales by year to month).
  - **Pivoting**: Rotate data axes for different perspectives (e.g., sales by region vs product).
  - **Slice**: Select a specific dimension value (e.g., sales for a single region).
  - **Dice**: Select a subset of data across multiple dimensions.
  - **Select**: Filter data based on criteria.
- **Example**: Analyze sales data by region, product, and time.

5 basic analytical operations that can be performed on an OLAP cube.

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

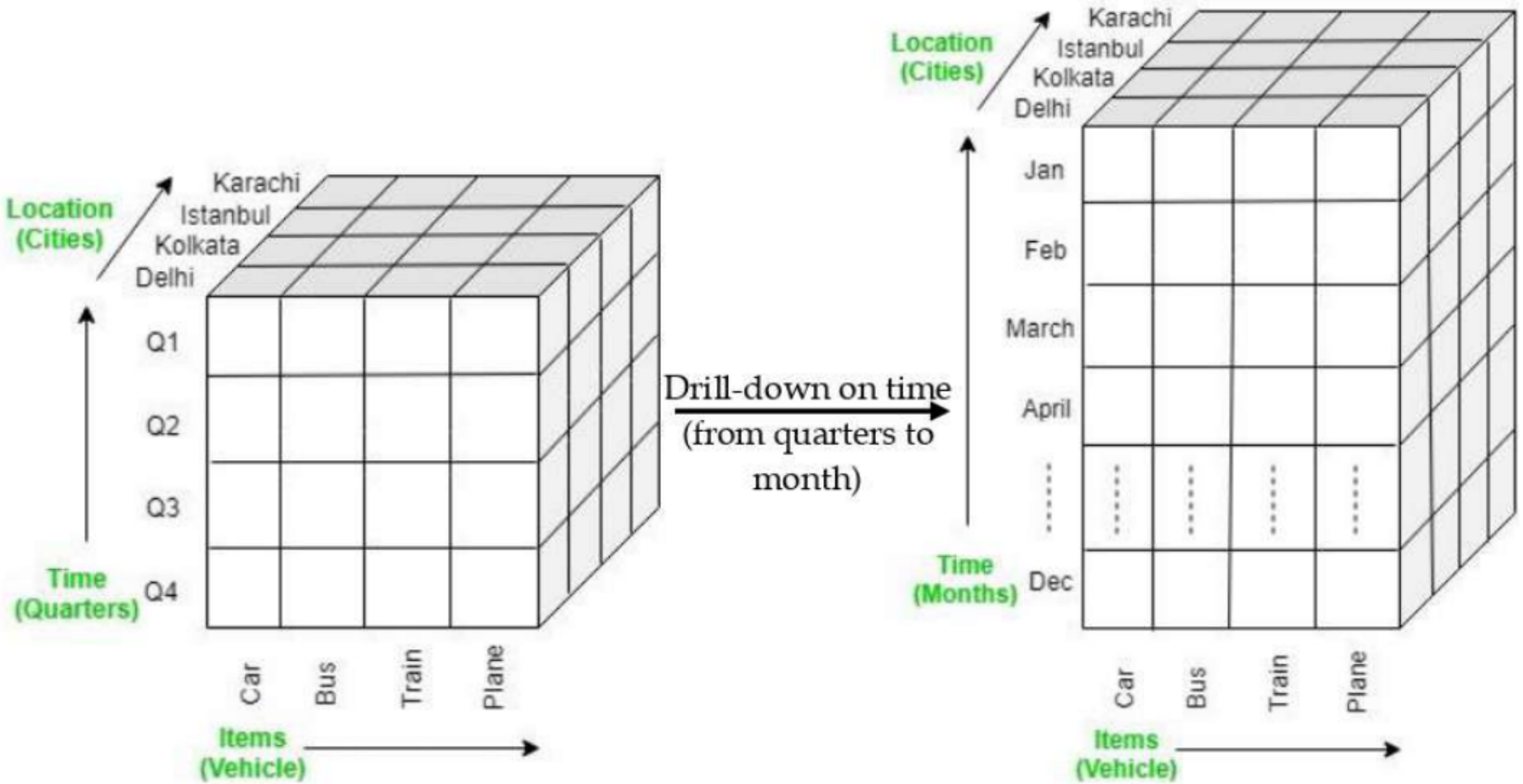
   Example:



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

2. ***Drill-down (Roll-down):*** Drill-down is the reverse of roll-up. In drill-down operation, the less detailed data is converted into highly detailed data. It can be done by either stepping down a concept hierarchy for a dimension or introducing additional dimensions. When drill-down is performed, one or more dimensions from the data cube are added.

Example:

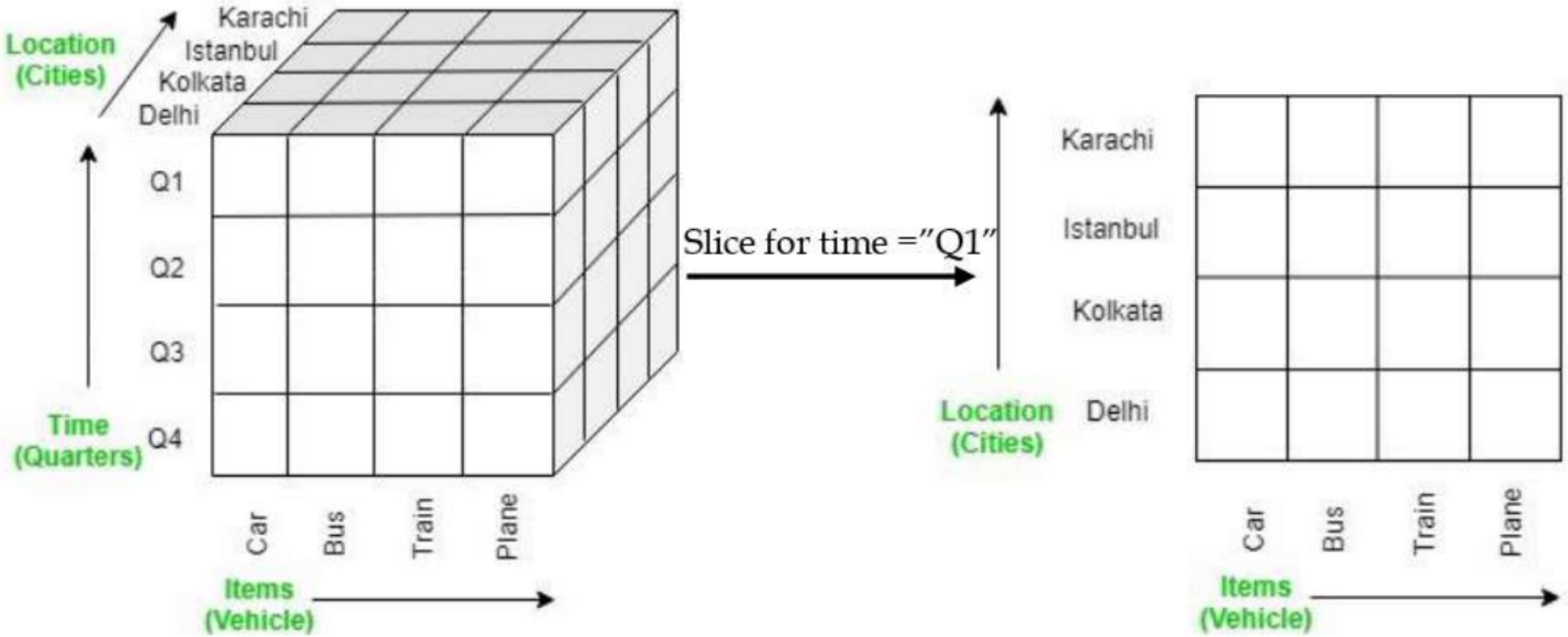


In this example, the drill down operation is performed by moving down in the concept hierarchy of *Time* dimension (Quarter -> Month).

5 basic analytical operations that can be performed on an OLAP cube.

3. ***Slice:*** The slice operation selects one particular dimension from a given cube and provides a new sub-cube. It reduces the dimensionality of the cubes.

Example:

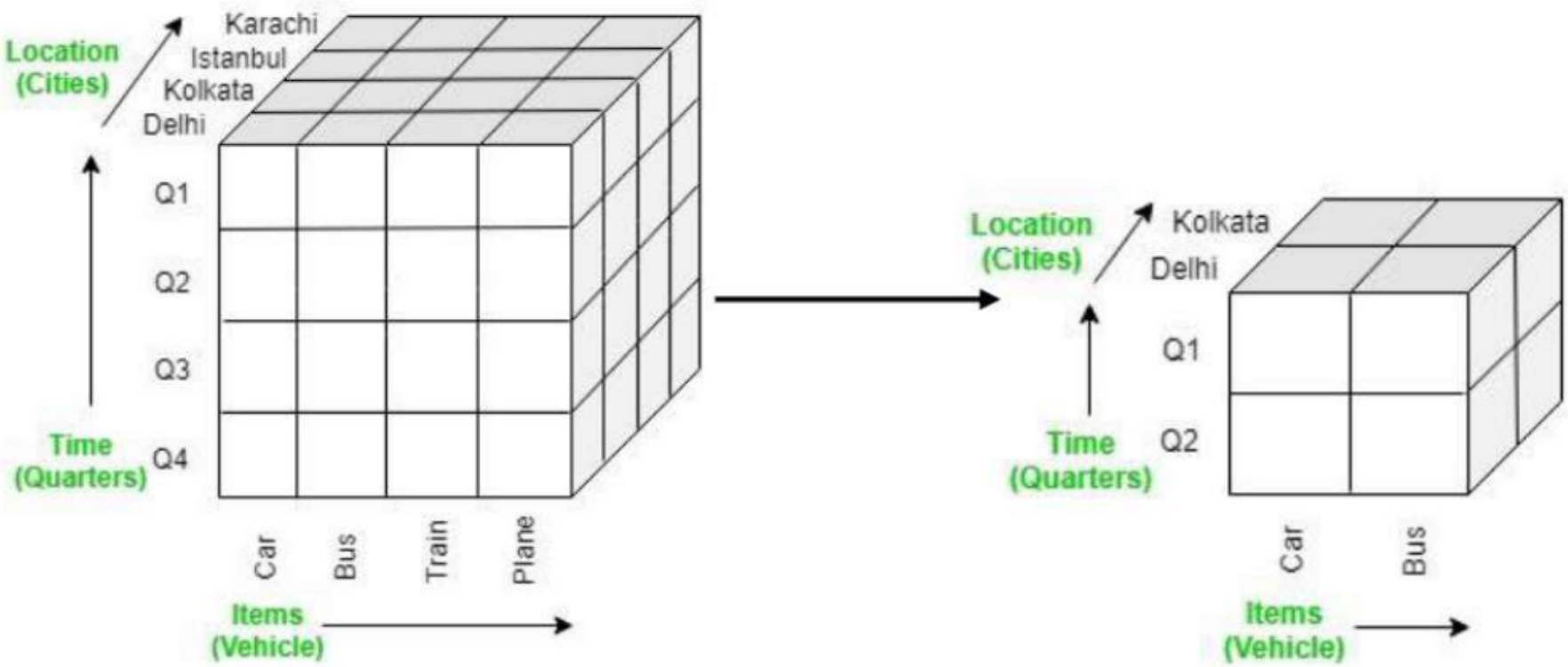


In this example, slice is performed for the dimension "time" using the criterion time = "Q1".

5 basic analytical operations that can be performed on an OLAP cube.

4. ***Dice:*** Dice selects two or more dimensions from a given cube and provides a new sub-cube.

   Example:

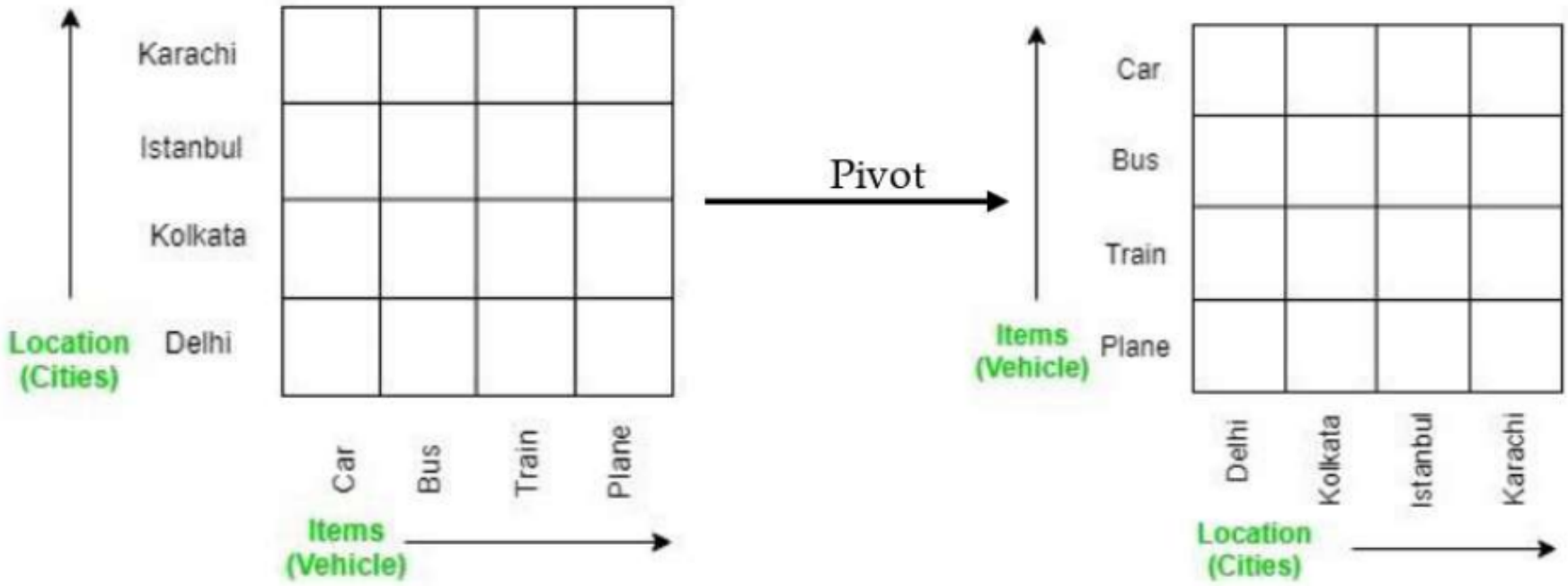


In this example, a sub-cube is selected by selecting following dimensions with criteria:

Location = "Delhi" or "Kolkata"
Time = "Q1" or "Q2"
Item = "Car" or "Bus"

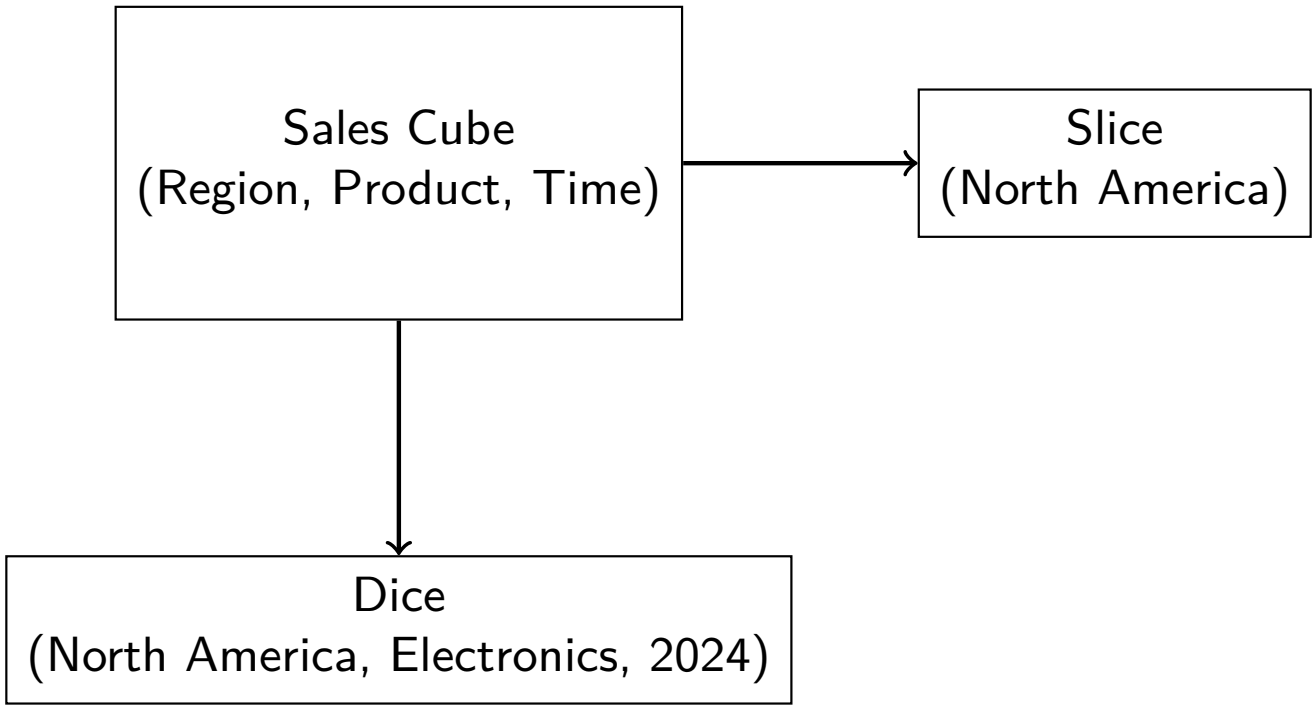
5. Pivot: The pivot operations is also known as rotation. It rotates the data axis to view the data from different prespectives.

Example: In the sub-cube otained after the slice operation, performing pivot operation gives a new view of it.



OLAP Operations: Example

- **Dataset**: Sales data with dimensions (Region, Product, Time).
- **Roll-up**: Total sales by year instead of month (e.g., 2024 total sales).
- **Drill-down**: Sales by month for 2024.
- **Pivoting**: View sales by product instead of region.
- **Slice**: Select sales for "North America" only.
- **Dice**: Select sales for "North America" and "Electronics" in 2024.



OLAP servers

Three main types:

- Relational OLAP
- Multidimensional OLAP
- Hybrid OLAP

Relational OLAP (ROLAP)

- Uses relational databases to store and manage warehouse data.
- Queries are translated to SQL and executed in RDBMS.
- Suitable for large volumes of data and complex schemas.
- Supports dynamic multidimensional analysis via SQL joins.
- Example: Metacube by Stanford Technology Group

Advantages

- 2-D relational tables can be viewed in multidimensional forms.
- Scalable with existing RDBMS infrastructure.
- Handles large datasets effectively.
- database security through authorization

Disadvantages

- difficul to perform complex calculations
- Slower query performance due to real-time joins.
- No pre-aggregated data.

ROLAP Architecture Diagram

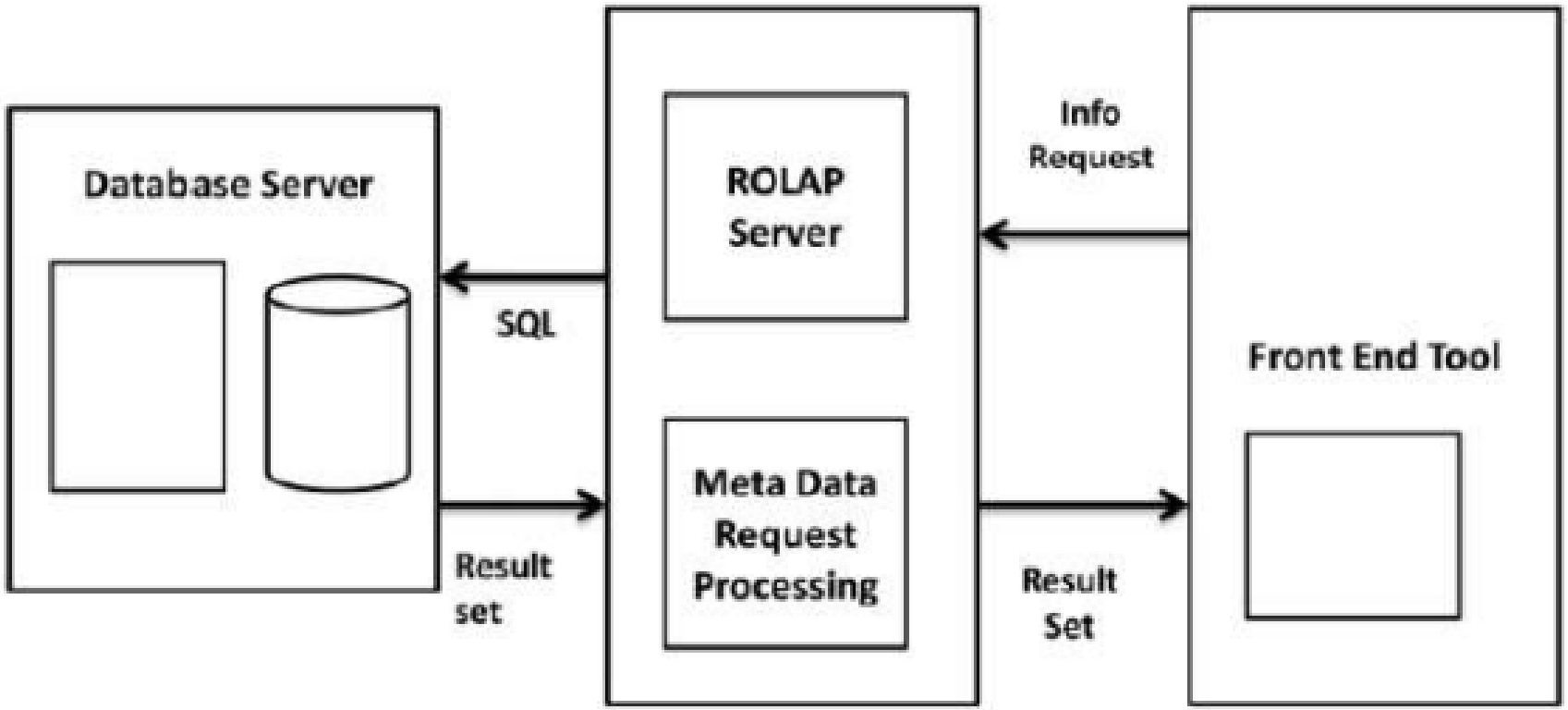


Fig: ROLAP Server

Multidimensional OLAP (MOLAP)

- Stores data in a multidimensional cube format.
- Data is pre-aggregated and indexed for performance.
- Ideal for fast query execution on pre-computed data.
- Example: Oracle's Express server

Advantages

- Very fast query performance.
- Efficient for repetitive analysis.

Disadvantages

- High storage cost.
- Not suitable for large-scale dynamic data.

MOLAP Architecture Diagram

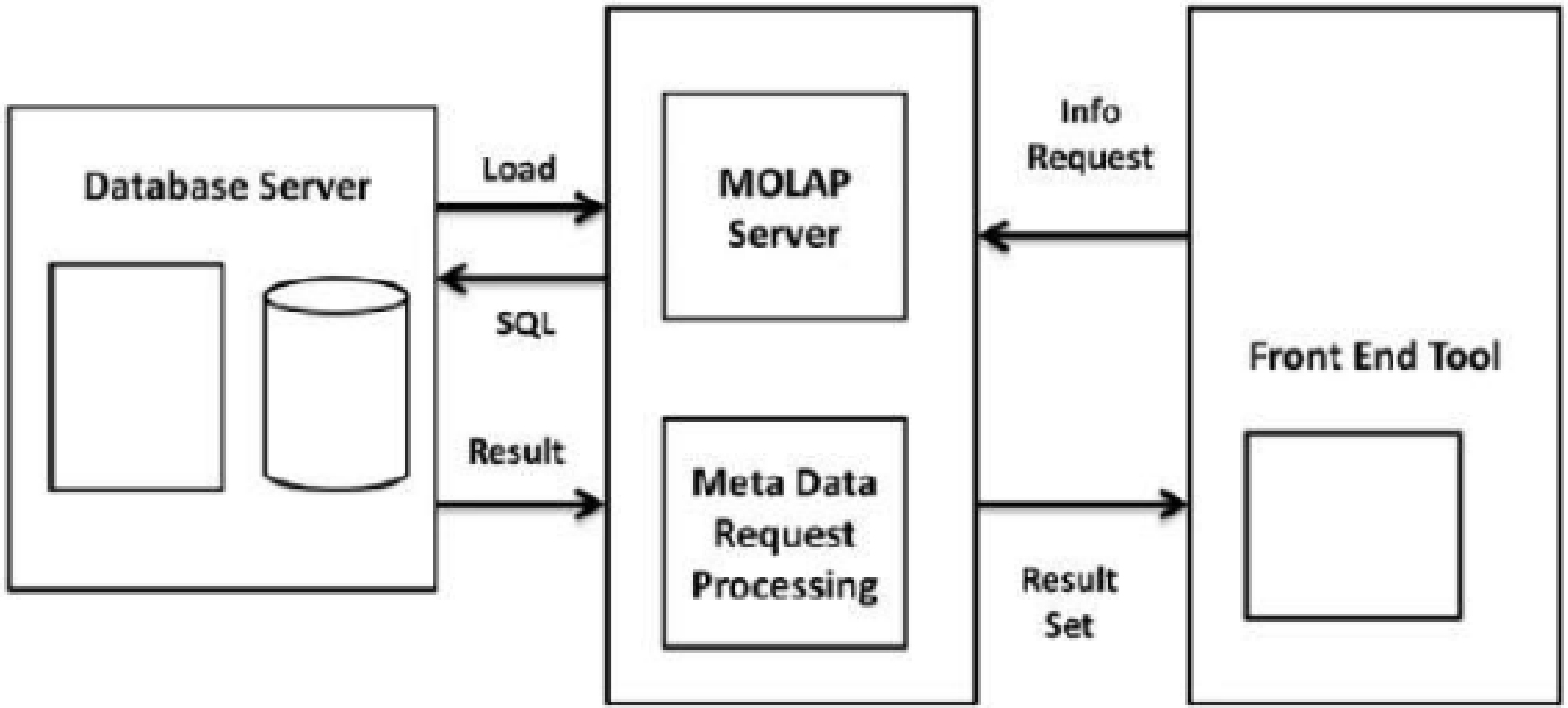


Fig: MOLAP Server

Hybrid OLAP (HOLAP)

- Combines features of both ROLAP and MOLAP.
- Summary data in cubes (MOLAP), detailed data in relational DB (ROLAP).
- Offers flexibility and performance for mixed workloads.

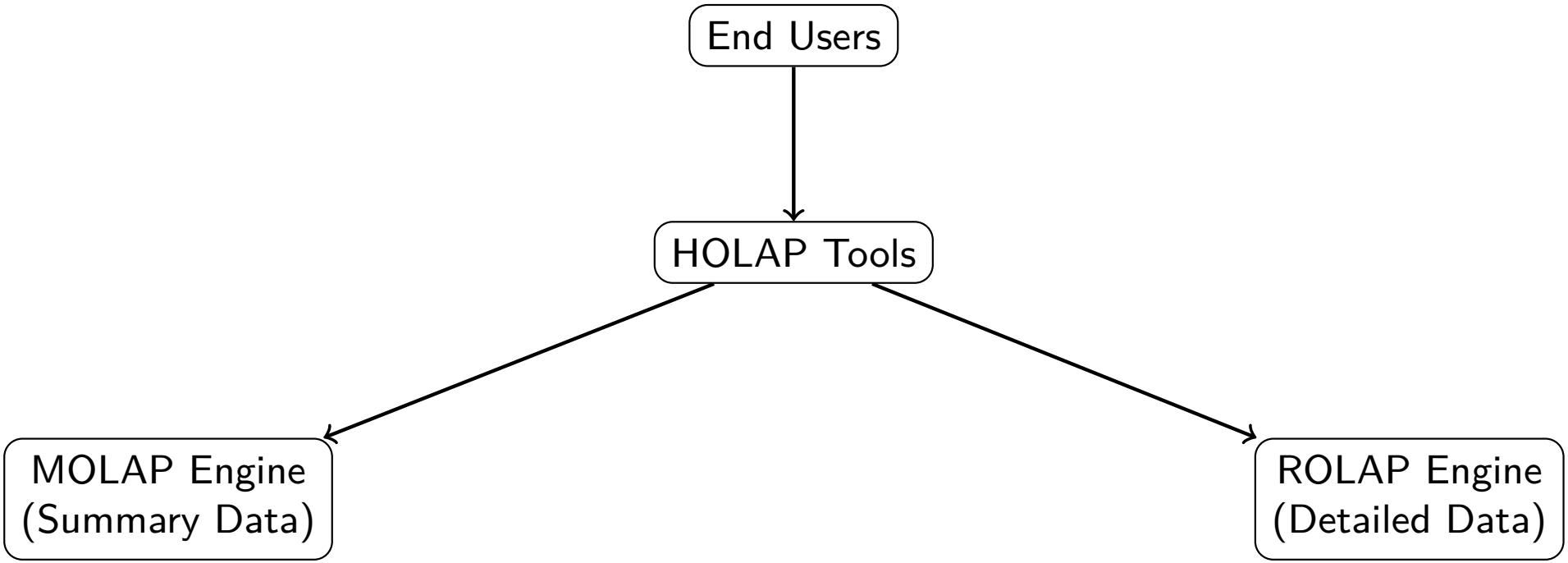
Advantages

- Balanced approach: speed + scalability.
- Handles large and complex data efficiently.

Disadvantages

- Complex to implement and maintain.
- Requires dual infrastructure.

HOLAP Architecture Diagram



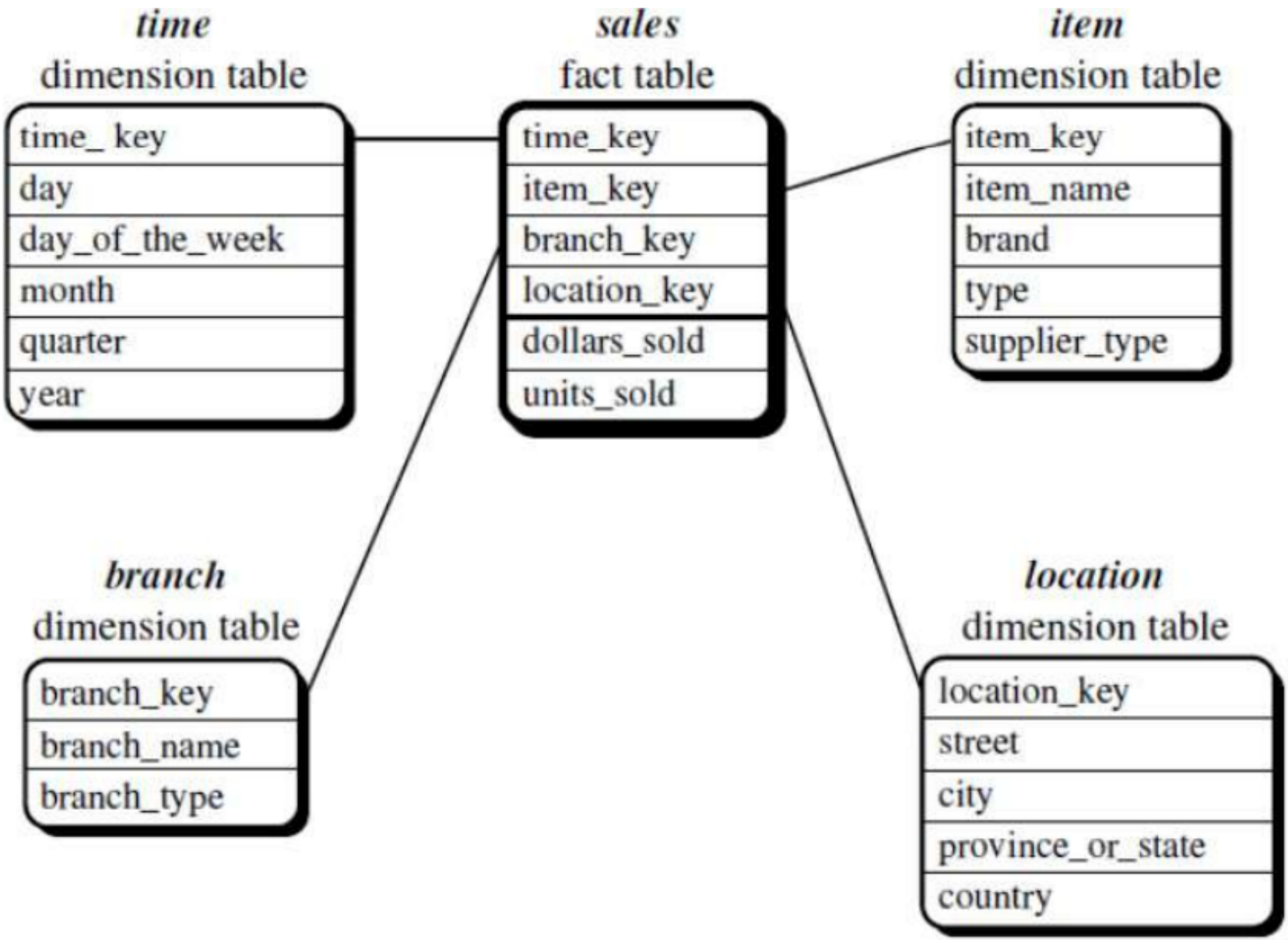
Conceptual Modeling of Data Warehouse

- A conceptual data model recognizes the highest-level relationships between the different entities.
- goal is to develop a schema for logical representation of data stored in datawarehouse.
- **Schema** is a logical description of the entire database.
- It includes the name and description of records of all recoed types including all associated data-items and aggregates.
  - Star schema
  - snowflake schema
  - Fact constellation schema (sometimes Galaxy schema)

Relational OLAP: Database Schemas

- **Star Schema**:
  - Central fact table connected to dimension tables.
  - Simple, denormalized, fast queries.
  - **Example**: Sales fact table with dimensions (Product, Time, Region).
- **Snowflake Schema**:
  - Normalized dimension tables, reducing redundancy.
  - More complex, slower queries but saves storage.
  - **Example**: Time dimension split into Year, Quarter, Month tables.
- **Star Constellation Schema**:
  - Multiple fact tables sharing dimension tables.
  - Used for complex data warehouses.
  - **Example**: Sales and Inventory fact tables sharing Product and Time dimensions.

Star Schema Diagram



Star Schema

- Data warehouse schema that contains two types of tables:
  - **Fact Table**
    - Contains the detailed summary data.
    - Primary key has one key per dimension.
    - Each tuple consists of a foreign key pointing to each dimension table.
    - Stores numeric values.
  - **Dimension Tables**
    - Consists of columns that correspond to the attributes of the dimensions.
    - Primary key of a dimension table is a foreign key in the fact table.

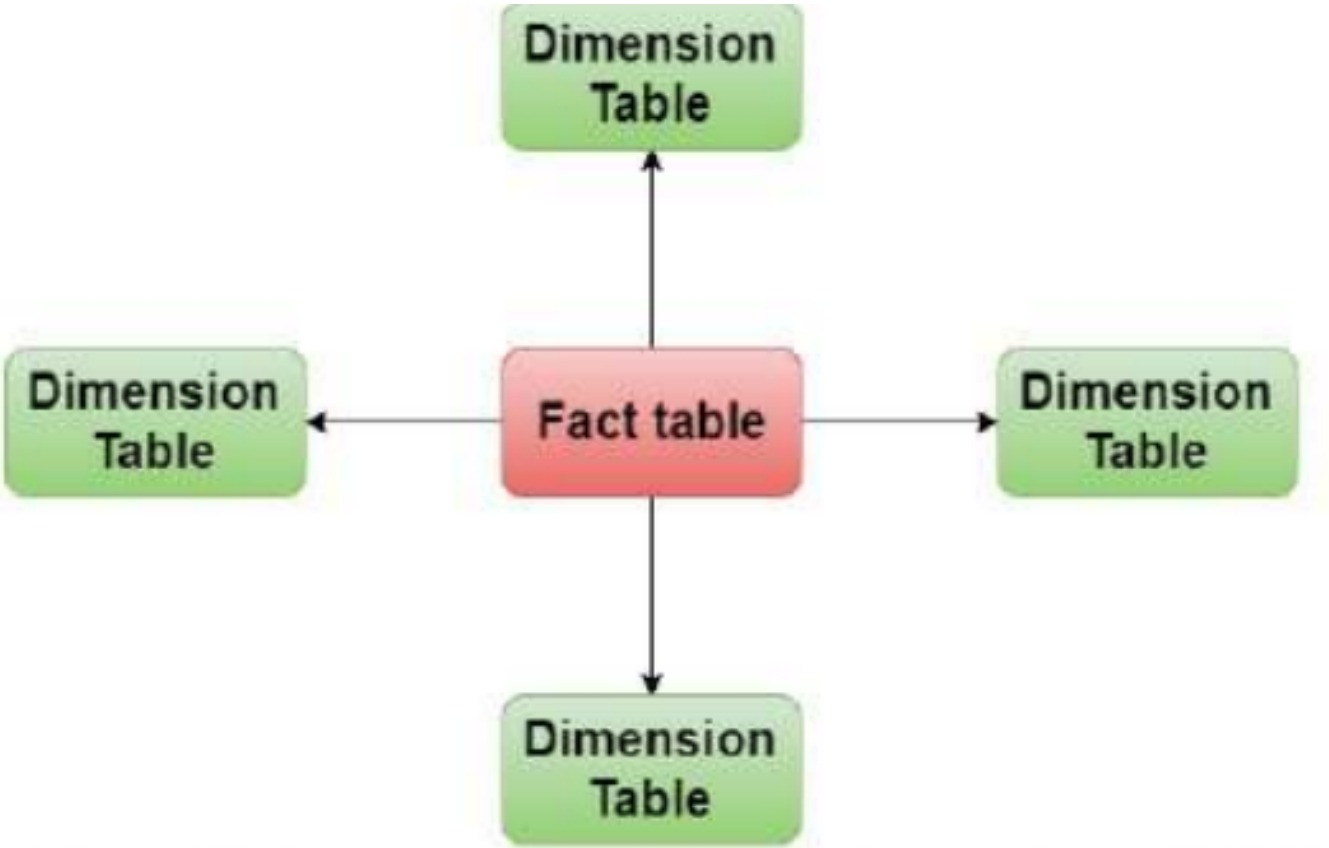


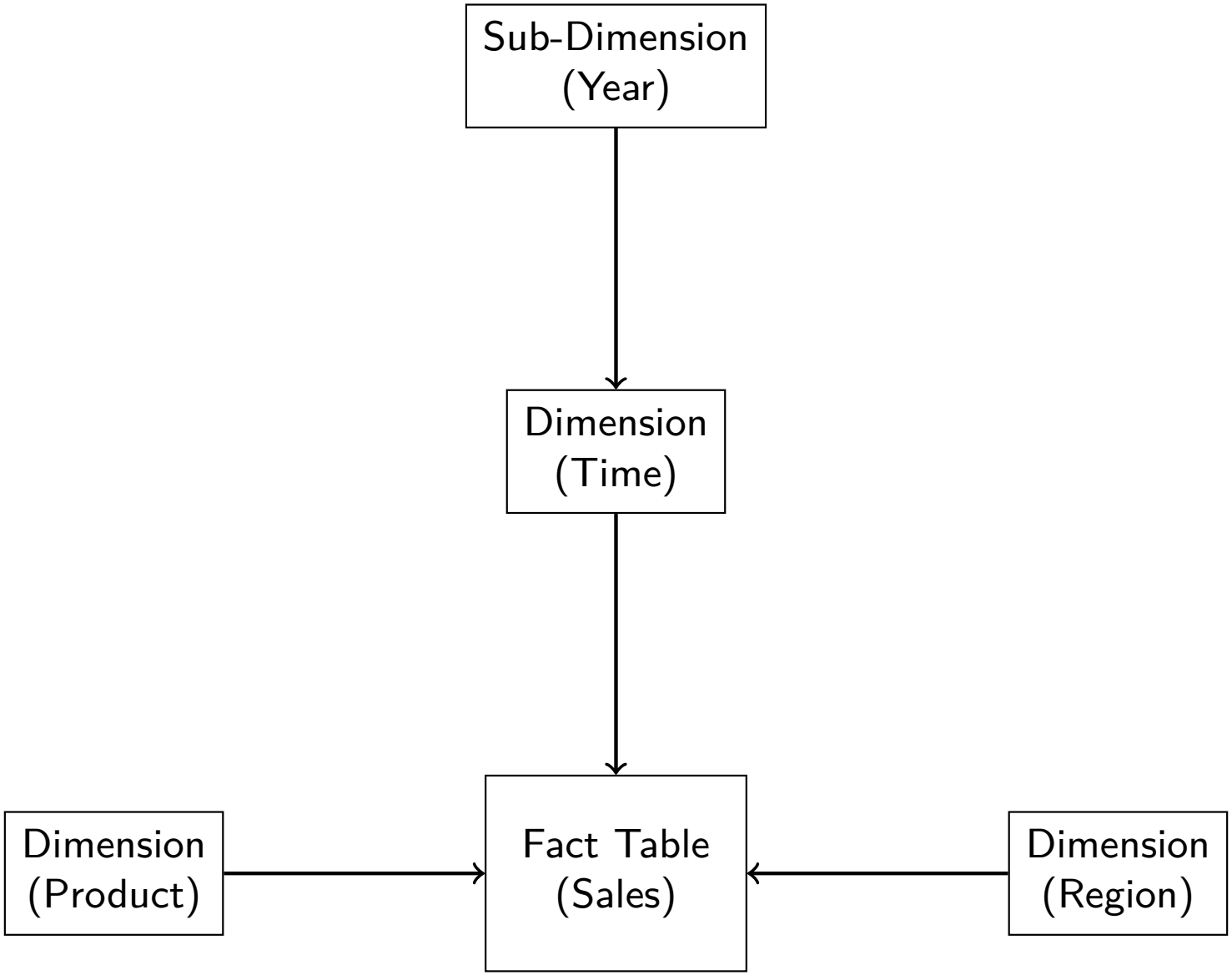
Fig: Star Schemas for Multidimensional Modal

Star Schema: Advantages and Disadvantages

- **Advantages of Star Schema**
  - It is easy to understand and small number of tables can join.
  - Since star schema contains de-normalized dimension tables, it leads to simpler queries due to lesser number of join operations and it also leads to better system performance.

- **Disadvantages of Star Schema**
  - It is difficult to maintain integrity of data in star schema due to de-normalized tables.
  - Redundancy of the data hence occupies additional space.

Snowflake Schema Diagram



- The snowflake schema is a variant of the star schema model, where some dimension tables are normalized which splits data into additional tables.
- The snowflake schema is represented by centralized fact table which is connected to multiple dimension table and this dimension table can be normalized into additional dimension tables.
- Snowflake Schema eliminates the redundancies and hence saves the storage space.
- It increases the number of dimension tables and requires more foreign key joins.
- The result is more complex queries and reduced query performance.

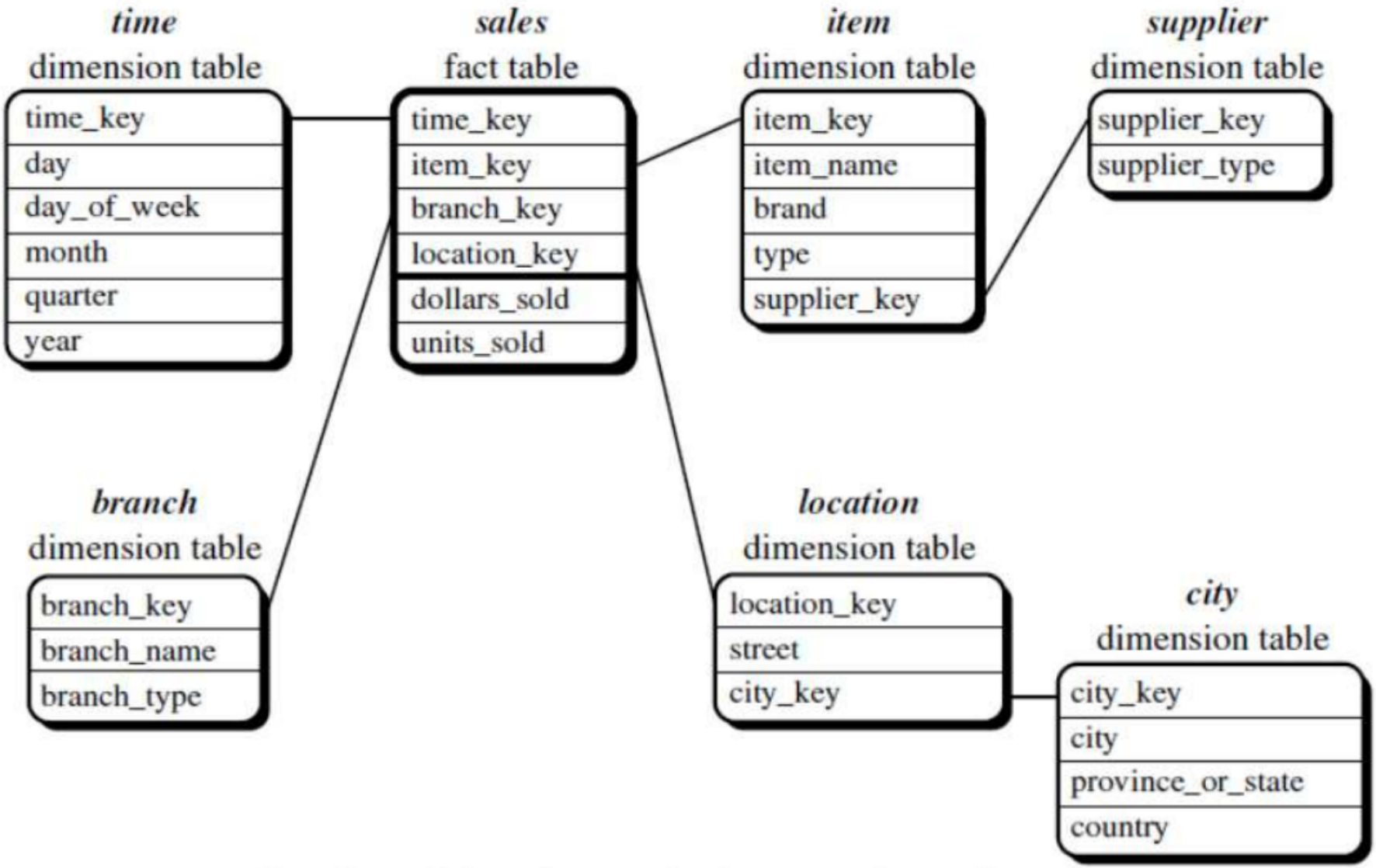


Fig: Snowflake schema of a data warehouse for sales.

Normalization in Snowflake Schema

- The item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.
- The item dimension table now contains the attributes item key, item_name, brand, type, and supplier_key where supplier_key is linked to the supplier dimension table containing supplier_key and supplier_type information.
- Similarly, the location dimension table is normalized into two tables, location and city table for location_key and street, city_key information.
- The city key in the new location table links to the city dimension table.

Fact Constellation Schema

- A fact constellation schema is a type of schema which consists of more than one fact table that share many dimension tables.
- This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation.
- The main disadvantage of fact constellation schemas is its more complicated design.

Fact constellation Schemas for Multidimensional Model

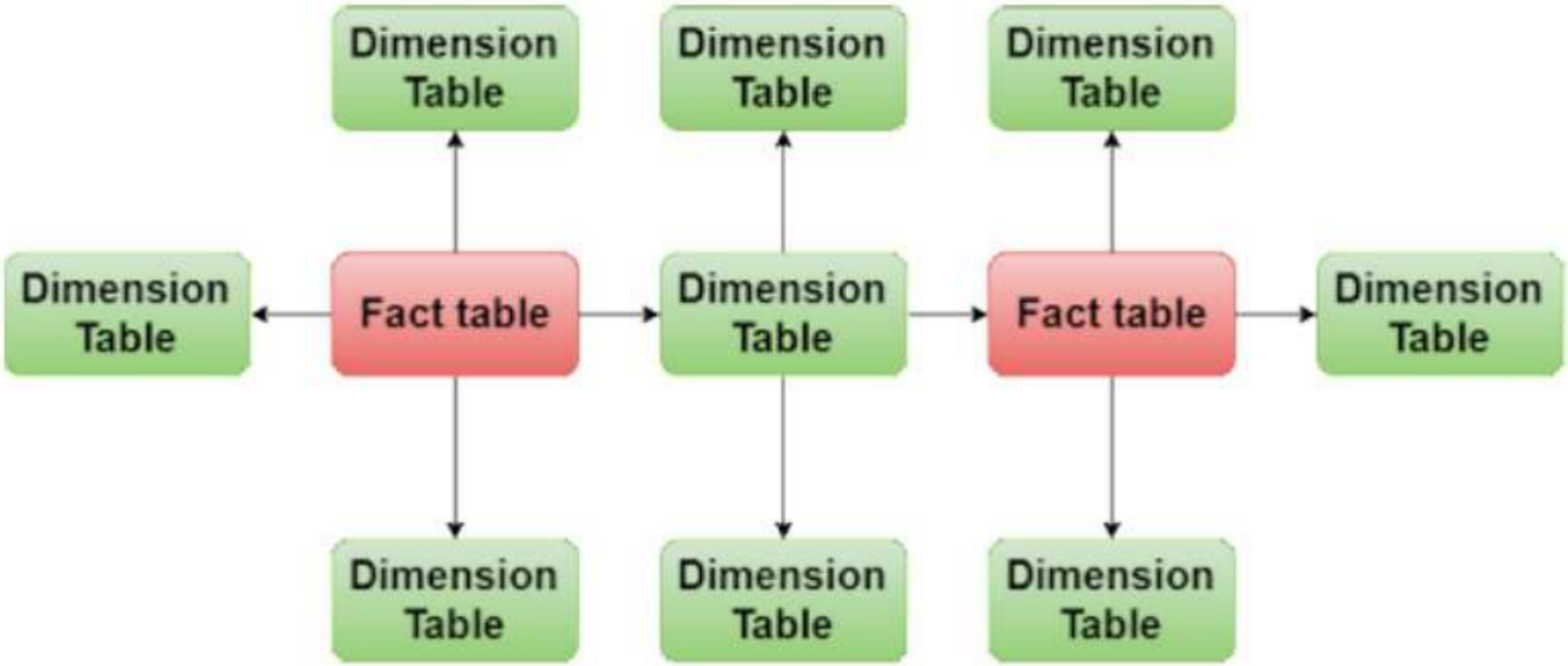


Fig: Fact constellation Schemas for Multidimensional Modal

Fact constellation schema of a data warehouse for sales and shipping.

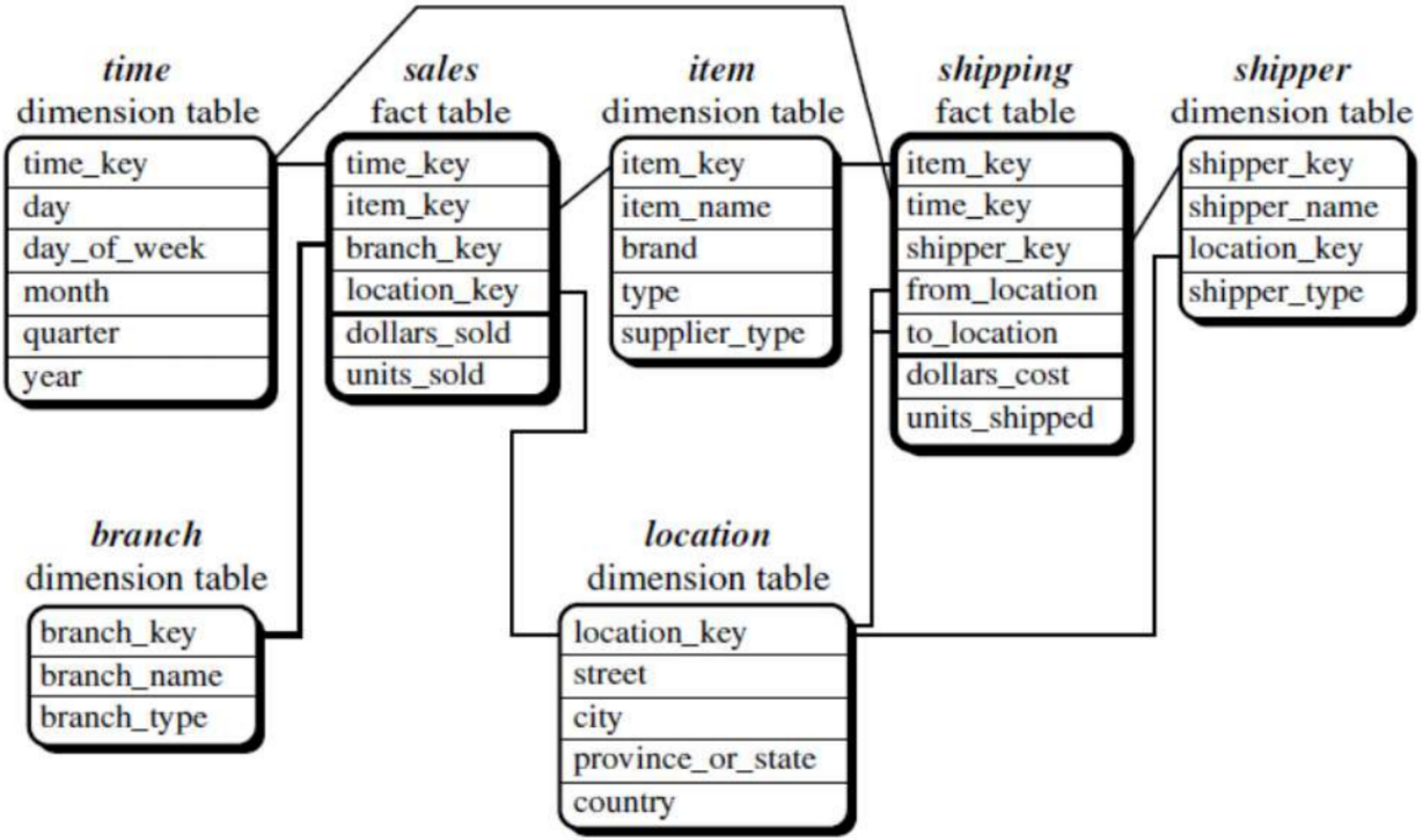


Fig: Fact constellation schema of a data warehouse for sales and shipping.

- The schema specifies two fact tables, sales and shipping.
- The sales table contains identical dimension tables to that of the star schema.
- The shipping table has five dimensions, or keys, namely time key, shipper key, from location, and to location, and two measures: dollars cost and units shipped.
- A fact constellation schema allows dimension tables to be shared between fact tables.
- For example, the dimensions tables for time, item, and location are shared between both the sales and shipping fact tables.

Multidimensional Data Model

- Represents data as a **data cube** with multiple dimensions (e.g., Time, Product, Region).
- Enables multidimensional analysis for OLAP.
- **Components**:
  - **Dimensions**: Descriptive attributes (e.g., Time, Region).
  - **Measures**: Quantitative data (e.g., Sales Amount).
  - **Hierarchies**: Levels within dimensions (e.g., Year $\rightarrow$ Quarter $\rightarrow$ Month).
- **Example**: Sales cube with dimensions (Time, Product, Region) and measure (Sales Amount).

MDX Queries: Overview

- **MDX (MultiDimensional eXpressions)**: Query language for OLAP databases.
- Used to retrieve and manipulate multidimensional data.
- Syntax similar to SQL but designed for cubes.
- **Example**: Retrieve total sales for "Electronics" in "North America" for 2024.

Summary

- **Data Warehouse Architecture**: Centralized repository with layers for data sources, ETL/ELT, storage, metadata, and BI tools.
- **ETL vs ELT**: ETL for structured data with pre-loading transformation; ELT for big data with in-warehouse transformation.
- **Data Warehouse vs Data Mart vs Data Lake**: Differ in scope, data type, and use case.
- **OLTP vs OLAP**: OLTP for transactions, OLAP for analytics.
- **OLAP Operations**: Roll-up, drill-down, pivoting, slice, dice, select.
- **Relational OLAP Schemas**: Star, Snowflake, Star Constellation for organizing data.
- **Multidimensional Model and MDX**: Enables complex analytical queries on data cubes.