Yashoda Shikshan Prasarak Mandal's
Yashoda Technical Campus
Approved by AICTE Delhi/ Govt. of Maharashtra
NH-4, Wadhe, Satara 415011
Email : principalpoly_ytc@yes.edu.in Call: 02162-271238/39 Mob. 9172220775
Faculty of Polytechnic
DEPARTMENT OF INFORMATION TECHNOLOGY

DWM NOTES

Chapter 2: Unit - II Data Warehouse Modelling – Data Cube and Online Analytical Processing (OLAP)				
Teaching Hours: 10	Marks Distribution			
	Remember =04 M	Understanding=08 M	Applying =04 M	Total =16 M

Topics and subtopics:

2.1 Data Cube: A Multidimensional Data Model

2.2 Stars, Snowflakes, and Fact Constellations: Schemas for Multidimensional Data Models

2.3 Dimensions: The Role of Concept Hierarchies

2.4 Measures: Categorization and Computation

2.5 OLAP Operations - Roll-up, Drill-down, Slice and Dice

Data warehouse modelling helps in organizing data so that users can perform fast, meaningful analysis. Two important concepts in this modelling are:

1. **Data Cube**
2. **Online Analytical Processing (OLAP)**

2.1 Data Cube: A Multidimensional Data Model

A data cube is a core component of the multidimensional data model, which organizes and represents business data in a way that allows for fast, flexible, and interactive analysis across various perspectives or dimensions. It is a metaphor for an n-dimensional data structure, not limited to a literal three-dimensional cube.

A **Data Cube** is a key concept in data warehouse modelling.
It represents data in **multiple dimensions** to support fast and flexible analysis.

What is a Data Cube?

A **Data Cube** is a **multidimensional structure** used to store and analyze data from different perspectives (dimensions).

It allows users to:

- View data at different levels of detail
- Perform quick summarizations (totals, averages)
- Analyze trends and patterns

A data cube is widely used in **OLAP (Online Analytical Processing)**.

Understanding Dimensions and Measures

1. Dimensions

These are perspectives or categories for analysis.
Examples:

- **Time** (Year, Quarter, Month)
- **Product** (Category, Brand)
- **Location** (City, Region)

- **Customer**

2. Measures

These are numeric values that are analyzed.
Examples:

- Sales
- Quantity
- Profit
- Attendance
- Marks

A Data Cube = **Dimensions + Measures**

Why is it Called a Cube?

Because data is visualized like a **3D cube**, where each axis represents a dimension.

Example of 3 dimensions:

- X-axis → Time
- Y-axis → Product
- Z-axis → Location

This structure makes it easy to ask questions like:

- "Total sales of Mobiles in Pune in 2024?"

- "Which product sold most in Q3?"

Features of a Multidimensional Data Model

1. Multidimensional View

Represents data as a cube with multiple dimensions.

2. Fast Query Performance

Summaries and aggregations are precomputed.

3. Support for OLAP Operations

- Roll-up
- Drill-down
- Slice
- Dice
- Pivot

4. Hierarchical Dimensions

Each dimension can have levels.
Example (Time dimension):
Day → Month → Quarter → Year

5. Handles Large Data

Designed for analytical processing, not routine transactions.

Example of a Data Cube

Consider a **Sales Cube** with:

- **Dimensions:** Time, Product, Region
- **Measure:** Sales Amount

This allows queries like:

- Total yearly sales (Roll-up)
- Monthly sales of Laptops (Slice)
- Comparison of sales in Pune vs Mumbai (Dice)

2.2 Stars, Snowflakes, and Fact Constellations: Schemas for Multidimensional Data Models

In data warehousing, data is organized in a **multidimensional format** so that it can be easily analyzed.

Multidimensional databases (used in Data Warehousing and OLAP) organize data using **schemas**. A schema is a way of arranging tables so that analysis becomes fast and simple.
To design this structure, three popular schemas are used:

1. **Star Schema**
2. **Snowflake Schema**
3. **Fact Constellation Schema (Galaxy Schema)**

These schemas define how **fact tables** and **dimension tables** are connected.

1. Star Schema

Definition

A Star Schema is the **simplest** and most commonly used schema in data warehousing.

Has **one main Fact Table** in the center.

Surrounding it are **Dimension Tables** (like Product, Time, Customer).

Dimension tables are **not normalized** (data is repeated).

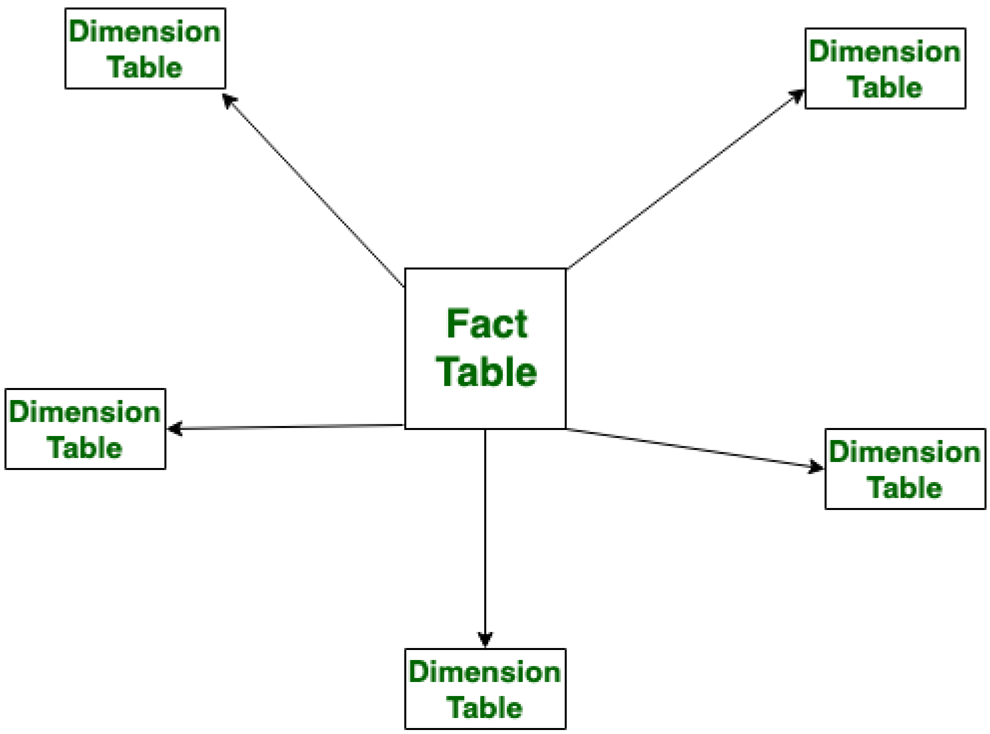
Looks like a **star shape**.

Very **easy to understand** and **fast for queries** because fewer joins are needed.

It consists of:

- **One central Fact Table**
- **Multiple Dimension Tables** around it
  → arranged like a "star".

Structure



Example

Fact Table: Sales

Contains numerical values (measures) such as:

- Quantity
- Sales Amount
- Profit

Dimension Tables:

- Customer

- Product
- Time
- Store

Characteristics

✔ Simple design
✔ Easy to understand
✔ Faster query performance
✔ Dimension tables are **denormalized**

Components of Star Schema

1. Fact Table (Center of the Star)

Contains **numeric, measurable values** used for analysis.

Examples of Measures:

- Sales Amount
- Quantity Sold
- Profit
- Marks
- Attendance

Also contains:

- Foreign keys referencing dimension tables

---

2. Dimension Tables (Surround the Fact Table)

These contain **descriptive attributes** used to analyze the facts.

Examples:

🔹 **Product Dimension**

- Product_ID
- Product_Name
- Category
- Brand

🔹 **Customer Dimension**

- Customer_ID
- Name
- Age
- City

- **Time Dimension**
  - Time_ID
  - Day
  - Month
  - Quarter
  - Year
- **Store Dimension**
  - Store_ID
  - Location
  - Manager

Dimension tables are **denormalized**, meaning all attributes are stored in one table.

Features of Star Schema

- Easy to understand and implement
- Fast query performance
- Ideal for OLAP operations

- Supports slicing, dicing, roll-up, drill-down
- Denormalized structure improves speed

Advantages of Star Schema

- Simple design
- Better query performance
- Easy to maintain and expand
- Suitable for Data Marts
- Reduced number of joins

Disadvantages of Star Schema

- Data redundancy (same data repeated in dimensions)
- May take more storage
- Not suitable for very large or complex dimension hierarchies

Real-Life Example

📘 **Star Schema for College Attendance System**

Fact Table: FACT_ATTENDANCE

| Student_ID | Course_ID | Date_ID | Present_Count |

Dimension Tables:

- DIM_STUDENT (Name, Department, Semester)
- DIM_COURSE (Course Title, Credits)
- DIM_DATE (Day, Month, Year)

---

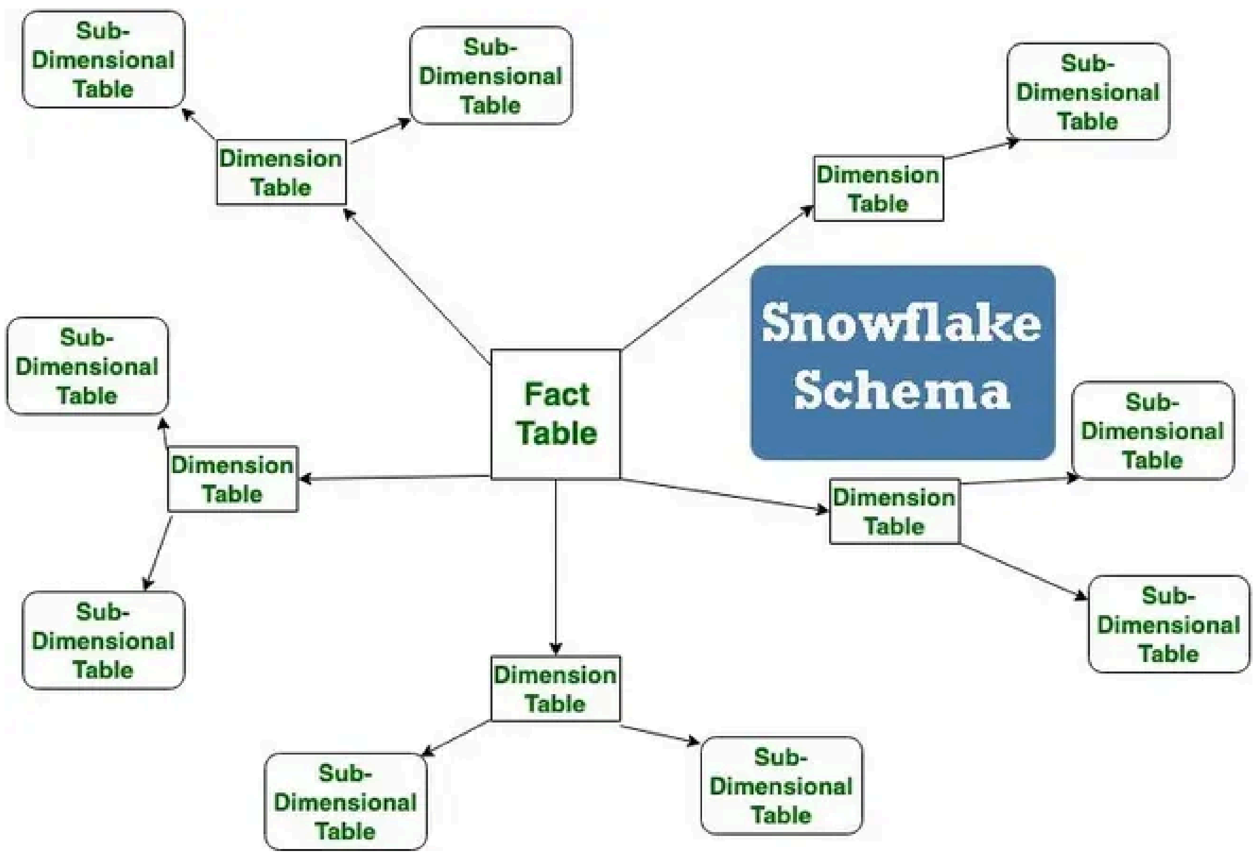
2. Snowflake Schema

A **Snowflake Schema** is an extension of the Star Schema where **dimension tables are normalized (split into sub-tables)**.
This creates a shape similar to a **snowflake**.

A snowflake schema is a logical data model used in data warehousing where a central fact table is connected to multiple, normalized dimension tables, which are further broken down into sub-dimensions. This structure resembles a snowflake in diagrams because the normalized dimensions branch out into a hierarchy, similar to a star schema but with greater normalization to reduce data redundancy and storage costs, though it can increase query complexityA snowflake schema is a type of data model where the fact table links to normalized dimension tables split into multiple related tables. It's a more detailed version of the star schema and is used to handle complex data structures. The snowflake effect applies only to dimension tables, not the fact table.

The Snowflake Schema is an extension of the Star Schema used in data warehousing.

It gets its name because its structure looks like a snowflake—a central fact table connected to multiple normalized dimension tables that branch out into more sub-tables.



- The dimension tables are normalized into multiple related tables, creating a hierarchical or "snowflake" structure.
- The fact table is still located at the center of the schema, surrounded by the dimension tables. However, each dimension table is further broken down into multiple related tables, creating a hierarchical structure that resembles a snowflake.

Key Features of Snowflake Schema

1. Normalized Dimension Tables

- Dimension tables are split into multiple related tables.
- Reduces data redundancy and improves data integrity.
- Example: A Product dimension may split into Product, Category, and Sub-Category tables.

2. More Complex Structure

- Because dimensions are normalized, the schema has more levels.
- Requires more joins compared to a Star Schema.

3. Better Data Integrity

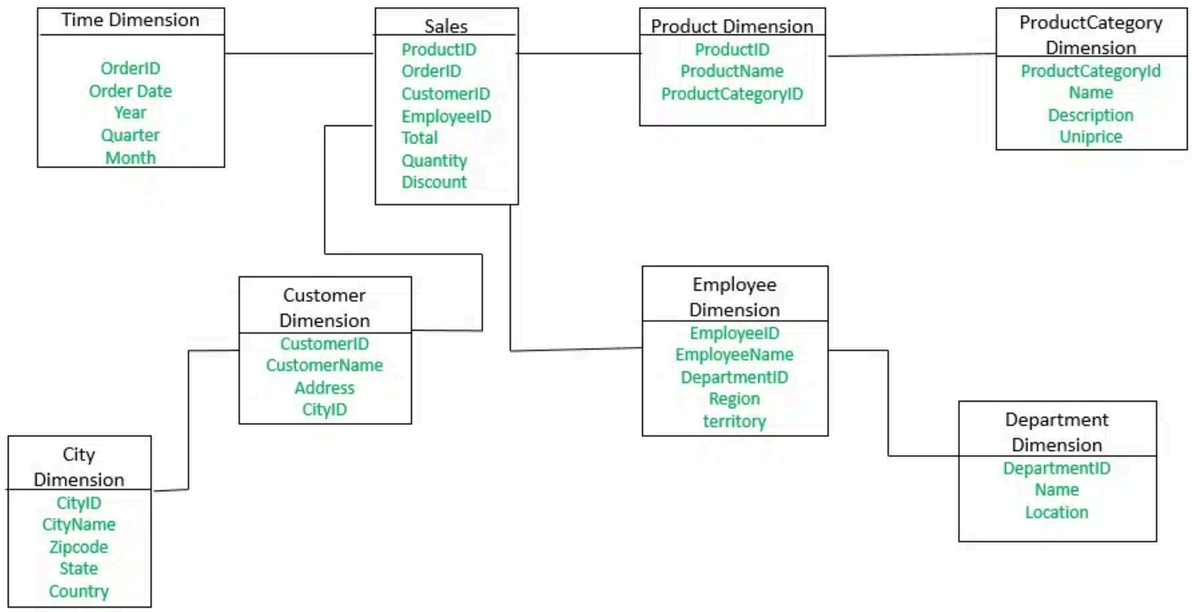
- Update anomalies are reduced because data is stored in smaller, related tables.

4. Slower Query Performance

- Queries become slower as they need to join multiple tables.
- Best suited for environments where data consistency is more important than speed.

Example of Snowflake Schema
The Employee dimension includes attributes like EmployeeID, Name, DepartmentID, Region, and Territory. DepartmentID links to the Department table, which holds department details like Name and Location.

The Customer dimension includes CustomerID, Name, Address, and CityID. CityID links to the City table, which stores City Name, Zipcode, State, and Country.



Characteristics of Snowflake Schema

- The snowflake schema uses small disk space.
- It is easy to implement the dimension that is added to the schema.
- There are multiple tables, so performance is reduced.
- The dimension table consists of two or more sets of attributes that define information at different grains.
- The sets of attributes of the same dimension table are populated by different source systems.

Features of the Snowflake Schema

- Normalization: Snowflake schema uses normalized tables to reduce redundancy and improve consistency.
- Hierarchical Structure: Built around a central fact table with connected dimension tables.
- Multiple Levels: Dimensions can be split into multiple levels, allowing detailed drill-down analysis.
- Joins: Requires more joins, which can slow performance on large datasets.
- Scalability: Scales well for large data, but its complexity makes it harder to manage.

Advantages of Snowflake Schema

- It provides structured data which reduces the problem of data integrity.
- It uses small disk space because data are highly structured.

Disadvantages of Snowflake Schema

- Snowflaking reduces space consumed by dimension tables but compared with the entire data warehouse the saving is usually insignificant.
- Avoid snowflaking or normalization of a dimension table, unless required and appropriate.
- Do not snowflake hierarchies of dimension table into separate tables. Hierarchies should belong to the dimension table only and should never be snowflakes.

- Multiple hierarchies that can belong to the same dimension have been designed at the lowest possible detai

---

3. Fact Constellation Schema (Galaxy Schema)

The Fact Constellation Schema, also known as the Galaxy Schema, is a complex data warehouse schema that contains multiple fact tables sharing common dimension tables.

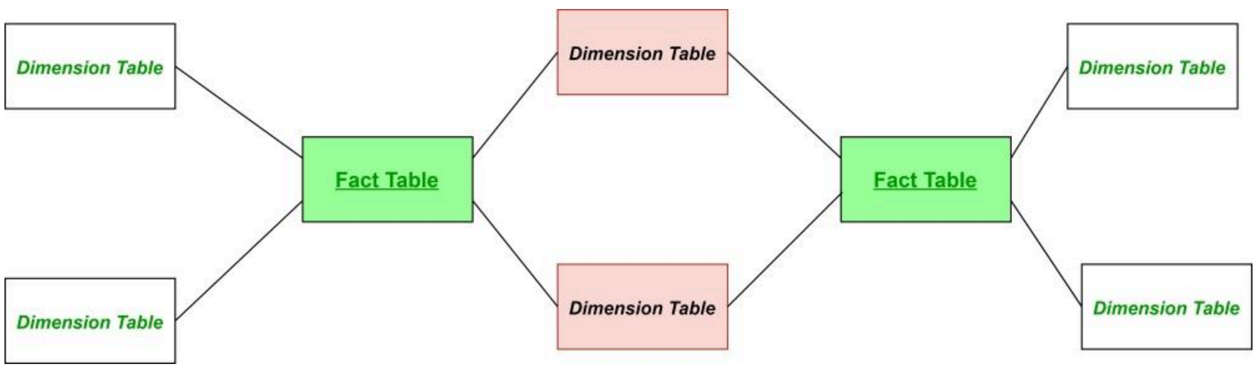
It is used when an organization needs to analyze multiple business processes together.

Fact Constellation in Data Warehouse modeling is a schema design that integrates multiple fact tables sharing common dimensions, often referred to as a "Galaxy schema." This approach allows businesses to conduct multi-dimensional analysis across complex datasets.

- Fact Constellation Schema, also known as the Galaxy Schema, is an advanced data modeling technique used in designing data warehouses. Unlike simpler models like the Star Schema and Snowflake Schema, the Fact Constellation Schema consists of multiple fact tables that share common dimensional tables.
- This model is ideal for handling complex systems and large-scale analytical queries, offering flexibility for business intelligence and data mining.

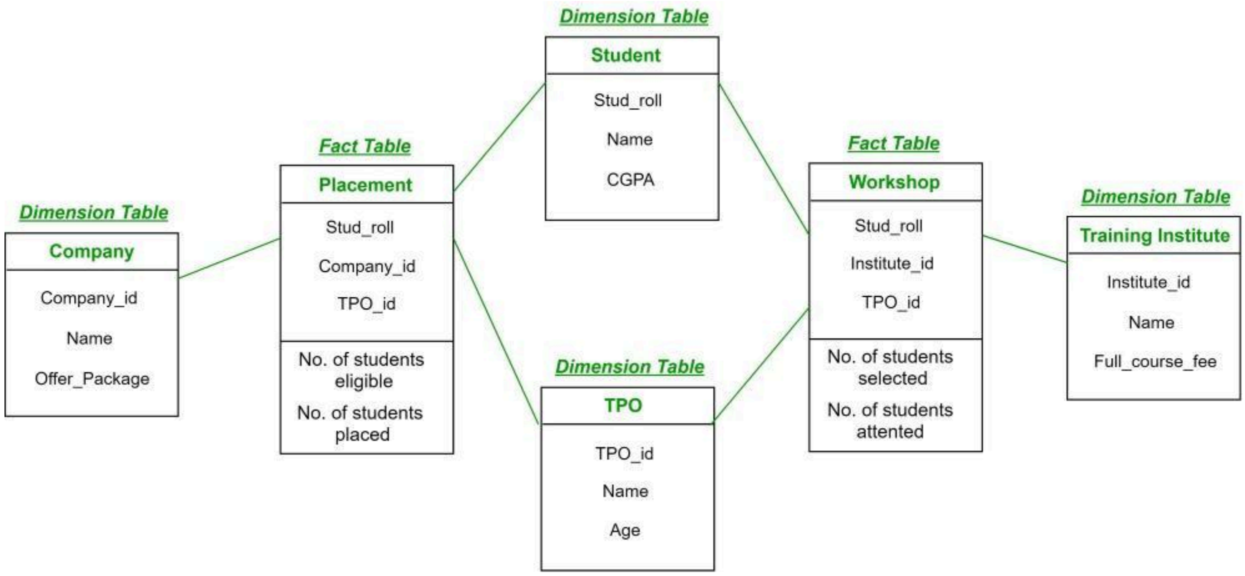
- The core components of the Fact Constellation Schema include Fact Tables and Dimension Tables. Fact tables store measurable, quantitative data, such as sales or revenue, while dimension tables store descriptive attributes like time, location, or product. These tables are interconnected, with multiple fact tables sharing the same dimension tables.

Here, the pink colored Dimension tables are the common ones among both the star schemas. Green colored fact tables are the fact tables of their respective star schemas.



Example

In the diagram below, we have two fact tables: Placement and Workshop, each associated with their own star schemas. These star schemas share common dimension tables, forming the Fact Constellation schema, also referred to as the Galaxy Schema.



2.3 Dimensions: The Role of Concept Hierarchies

In a **data warehouse**, information is organized using **dimensions**—such as **Time, Location, Product, Customer,** etc. Each dimension contains attributes that help analyze data from different perspectives.

What Are Concept Hierarchies?

A **concept hierarchy** is a structured way of organizing data from **low-level detailed data** to **high-level summarized data**.
It shows how data can be **rolled up (summarized)** or **drilled down (detailed)** during analysis.

Why Concept Hierarchies Are Important?

Concept hierarchies help users perform OLAP operations like:

- **Roll-up:** Moving from detailed data → summarized data
- **Drill-down:** Moving from summarized data → detailed data
- **Slice/Dice:** Selecting specific levels to filter data

1 Roll-up (Data Summarization)

Meaning:

Roll-up means **moving from lower-level detailed data to higher-level summarized data** in a concept hierarchy.

How it works:

You choose a **higher level** in the hierarchy and combine (aggregate) the data.

Example (Time Hierarchy):

Day → Month → Quarter → Year

If you roll-up from **Month to Year**, you get:

- Total sales for the whole year (instead of per month)
- Less detailed and more summarized information

Where used?

✔ For management reports
✔ When analyzing trends
✔ When decision makers want a big-picture view

---

2 Drill-down (Going into Details)

Meaning:

Drill-down means **moving from summarized data to more detailed data**.

How it works:

You choose a **lower level** in the hierarchy to see finer details.

Example (Location Hierarchy):

Country → State → City

If a report shows:

- **Sales in India (Country level)**
  Drill-down can show:

- Sales in Maharashtra (State level)

- Then in Satara (City level)

Where used?

✔ For identifying problems
✔ For deep analysis
✔ For understanding detailed behavior

---

3 Slice (Selecting a Single Level of Data)

Meaning:

Slice means **selecting a single value** from one dimension to get a smaller portion of the cube.

How it works:

You choose one selection like:

- A single year
- A single product
- A single region

Example:

If you select **Year = 2024**, you get all data only for 2024.
This removes that dimension's other values.

Where used?

✔ For focused analysis of one category
✔ When analyzing one specific time period or product

4️⃣ Dice (Selecting Multiple Conditions)

Meaning:

Dice means selecting **multiple values** or applying **multiple filters** across dimensions.

How it works:

You apply conditions on more than one dimension.

Example:

Select:

- Year = 2024 and 2025
- Product Category = Electronics
- City = Mumbai, Pune

This gives a smaller cube with selected categories only.

Where used?

✔ For custom analysis
✔ For comparing specific groups
✔ For analyzing combined conditions

How Concept Hierarchies make these operations possible

Operation	Uses Hierarchy for	Purpose
Roll-up	Going upward	Summaries
Drill-down	Going downward	Details
Slice	Selecting one level or value	Focus view
Dice	Selecting multiple values	Filtered multi-view