MUST

Wisdom & Virtue

MIRPUR UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF SOFTWARE ENGINEERING

Data Warehousing (DWH)

Saman Fatima
(Lecturer)

Date: 26-5-2023

Data Warehousing (DWH)

A system used for reporting and data analysis, typically used for creating decision support systems.

Key Points:

Centralized data repository

Integrates data from multiple sources

Supports analytical processing and business intelligence (BI)

OLTP vs OLAP

Feature	OLTP	OLAP
Full Form	Online Transaction Processing	Online Analytical Processing
Purpose	Run day-to-day operations	Analyze data for decision making
Data Type	Current, detailed	Historical, summarized
Query Type	Simple, short queries	Complex queries
Database Design	Normalized	Denormalized
Speed	Fast for write operations	Fast for read and analysis
Users	Clerks, DBAs	Analysts, Managers

Example:

?

- OLTP: ATM transactions
- OLAP: Monthly sales trends

Dimensional Modeling - Kimball Approach

Dimensional Modeling is a design concept used for data warehouses.
Proposed by Ralph Kimball, it focuses on ease of querying and performance.

Key Concepts:

Fact Tables: Contain measurable data (e.g., sales amount)

Dimension Tables: Contain descriptive data (e.g., customer, product)

Relationships between facts and dimensions

Star Schema Structure:

A star schema is a type of data modeling technique used in data warehousing to represent data in a structured.

In a star schema, data is organized into a central fact table that contains the measures of interest, surrounded by dimension tables that describe the attributes of the measures.

fact table

star schema contains the measures or metrics that are of interest to the user or organization. For example, in a sales data warehouse, the fact table might contain sales revenue, units sold, and profit margins. Each record in the fact table represents a specific event or transaction, such as a sale or order.

Star Schema Structure:

The **dimension tables** in a star schema contain the descriptive attributes of the measures in the fact table. These attributes are used to slice and dice the data in the fact table, allowing users to analyze the data from different perspectives. For example, in a sales data warehouse, the dimension tables might include product, customer, time, and location.

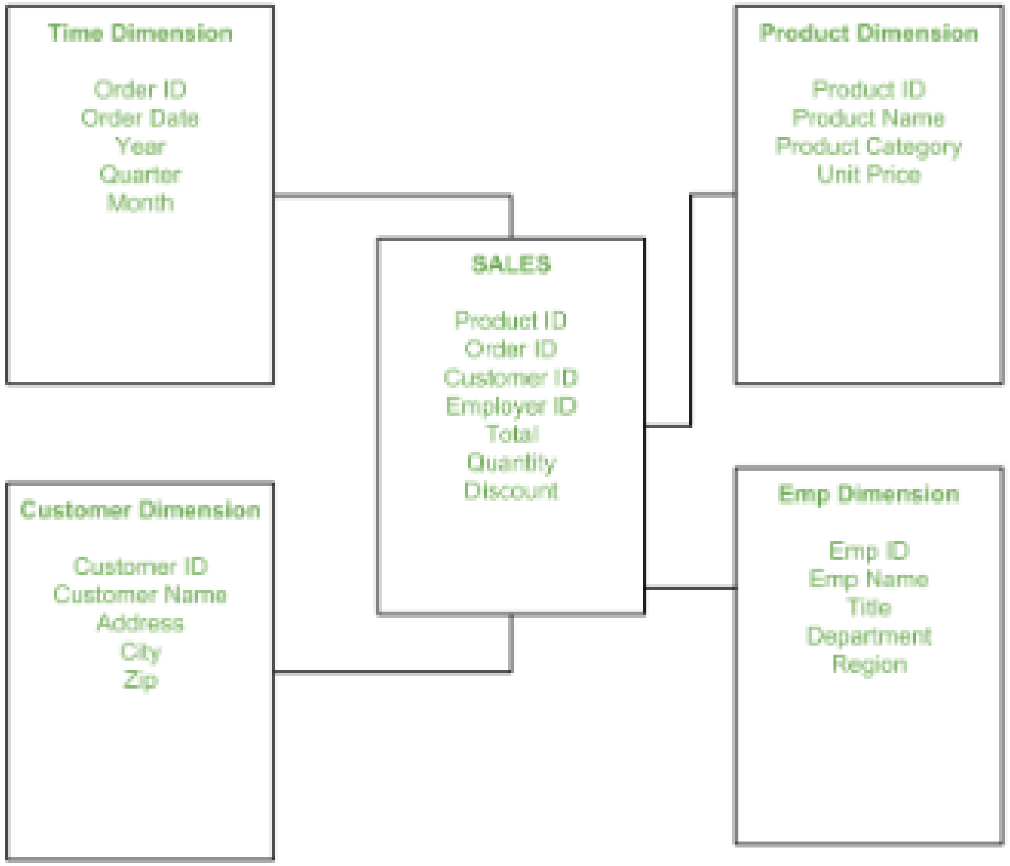
Each dimension table is joined to the fact table through a foreign key relationship. This allows users to query the data in the fact table using attributes from the dimension tables. For example, a user might want to see sales revenue by product category, or by region and time period.

Example of Star Schema

Example of Star Schema

In the given demonstration, SALES is a fact table having attributes i.e. (Product ID, Order ID, Customer ID, Employer ID, Total, Quantity, Discount) which references to the dimension tables. **Employee dimension table** contains the attributes: Emp ID, Emp Name, Title, Department and Region. *Product dimension table* contains the attributes: Product ID, Product Name, Product Category, Unit Price. *Customer dimension table* contains the attributes: Customer ID, Customer Name, Address, City, Zip. *Time dimension table* contains the attributes: Order ID, Order Date, Year, Quarter, Month.

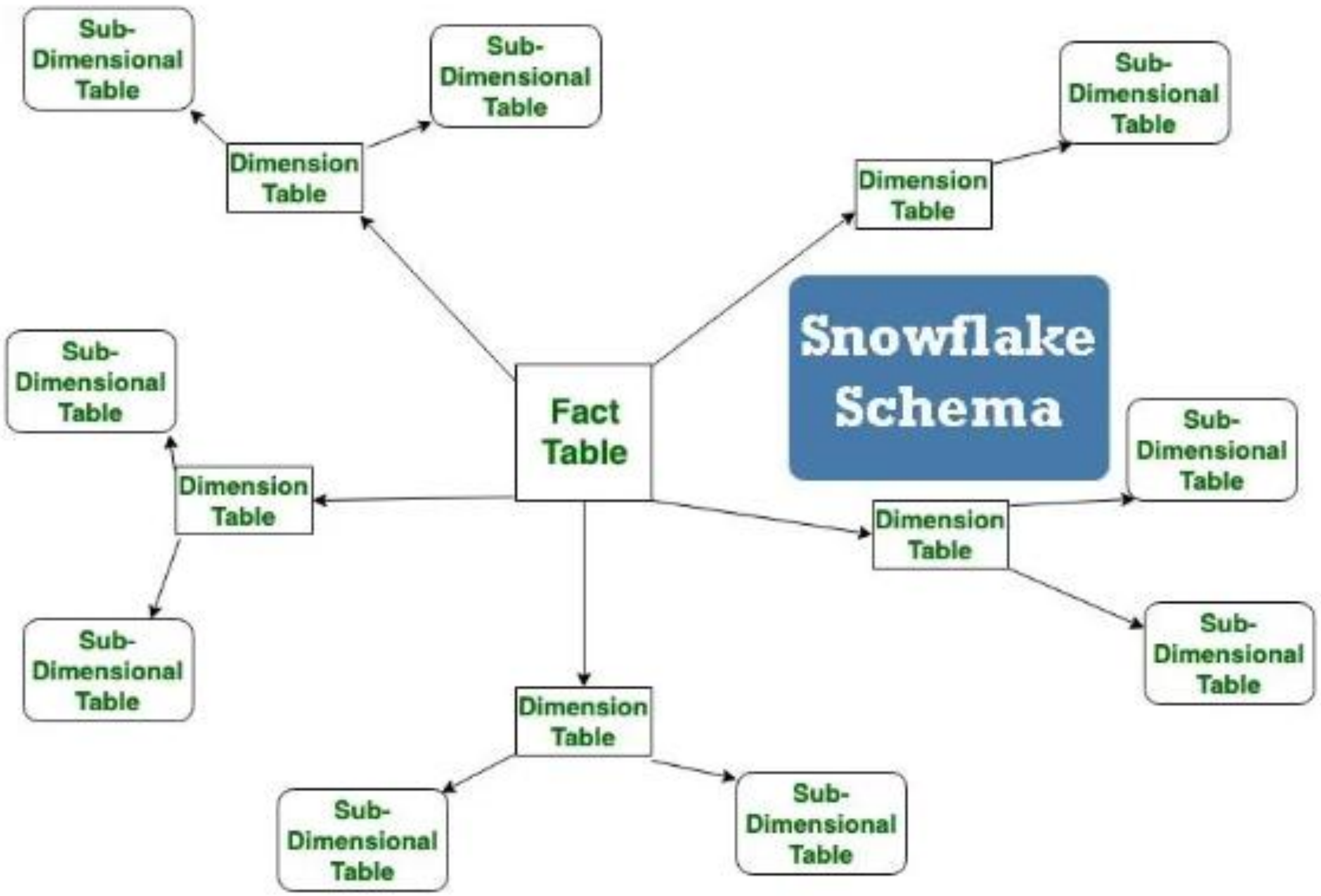
Example of Star Schema



Snowflake Schema in Data Warehouse Model

- A snowflake schema is a type of data modeling technique used in data warehousing to represent data in a structured way that is optimized for querying large amounts of data efficiently.
- It is a variant of the star schema. Here, the centralized fact table is connected to multiple dimensions.
- In the snowflake schema, dimensions are present in a normalized form in multiple related tables. The snowflake structure materialized when the dimensions of a star schema are detailed and highly structured having several levels of relationship and the child tables have multiple parent tables.
- The snowflake effect affects only the dimension tables and does not affect the fact tables.

Snowflake Schema in Data Warehouse Model



Snowflake Schema

Snowflake Schema in Data Warehouse Model

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.

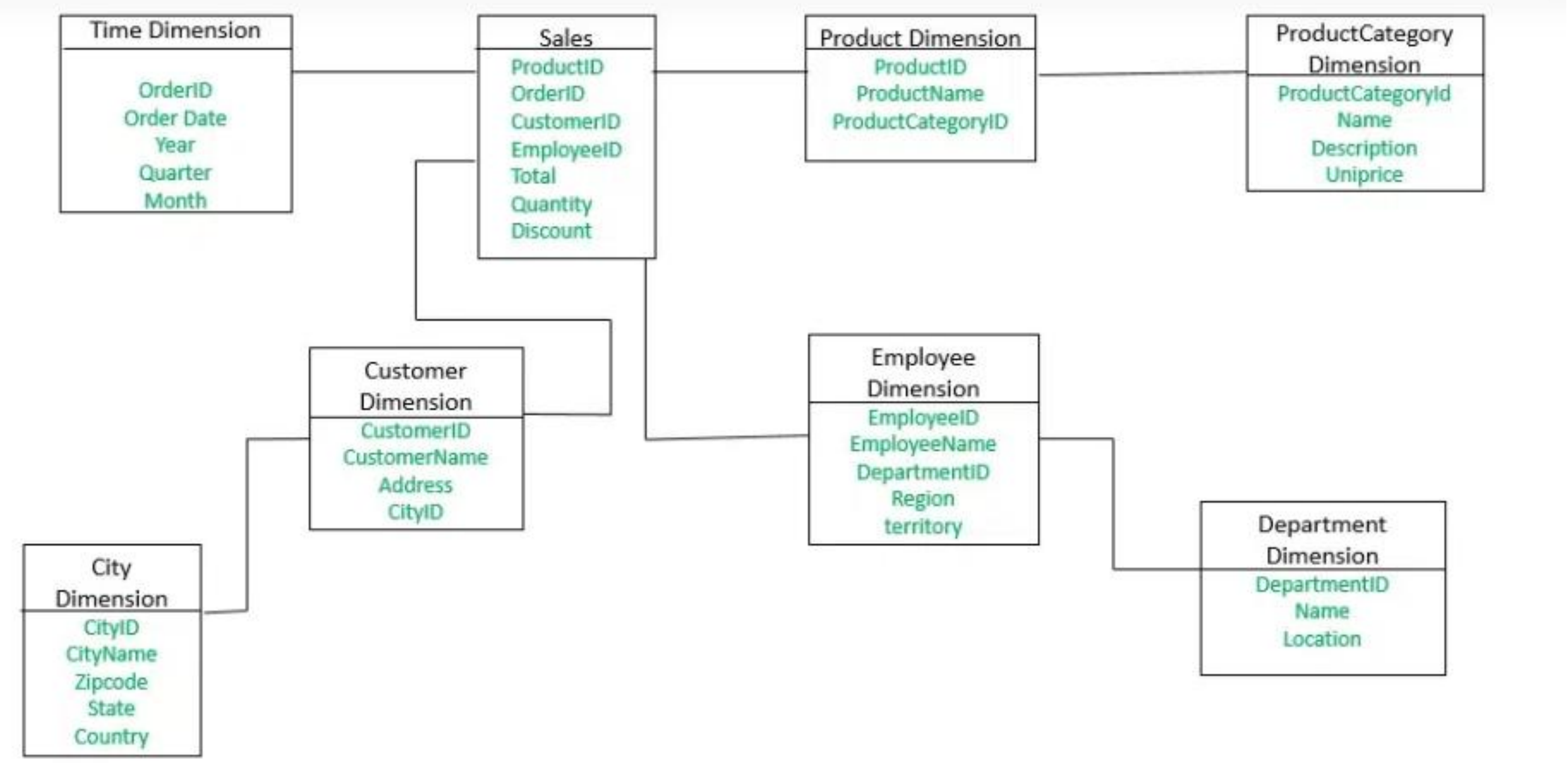
Example of Snowflake Schema

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.

Example of Snowflake Schema

?

Example of Snowflake Schema



Snowflake Schema

Difference Between Star and Snowflake Schema

Feature	Star Schema	Snowflake Schema
Structure	Central fact table connected to dimension tables	Fact table connected to normalized dimension tables
Data Normalization	Denormalized dimension tables	Normalized dimension tables
Performance	Faster query execution due to fewer joins	Slower query performance due to multiple joins
Design Complexity	Simple and easy to understand	Complex design with multiple levels of relationships
Space Usage	Uses more storage due to denormalization	Uses less storage due to normalization
Data Redundancy	Higher data redundancy	Lower data redundancy
Foreign Keys	Fewer foreign keys	More foreign keys
Use Cases	Best for large datasets and quick ad-hoc queries	Best for structured, predictable queries

Difference Between Star and Snowflake Schema

Query Complexity	Low query complexity	High query complexity due to multiple joins
Maintainability	Easier to maintain due to simple design	More difficult to maintain due to complexity
Scalability	Scalable but may encounter performance issues with large data volumes	More scalable for very large data sets due to normalization
Suitability for BI Tools	Ideal for BI tools and quick reporting	Better for systems that require detailed reporting and data analysis
Data Integrity	Lower data integrity due to redundancy	Higher data integrity due to normalization
Updates and Modifications	More difficult to update due to denormalization	Easier to update as data is normalized

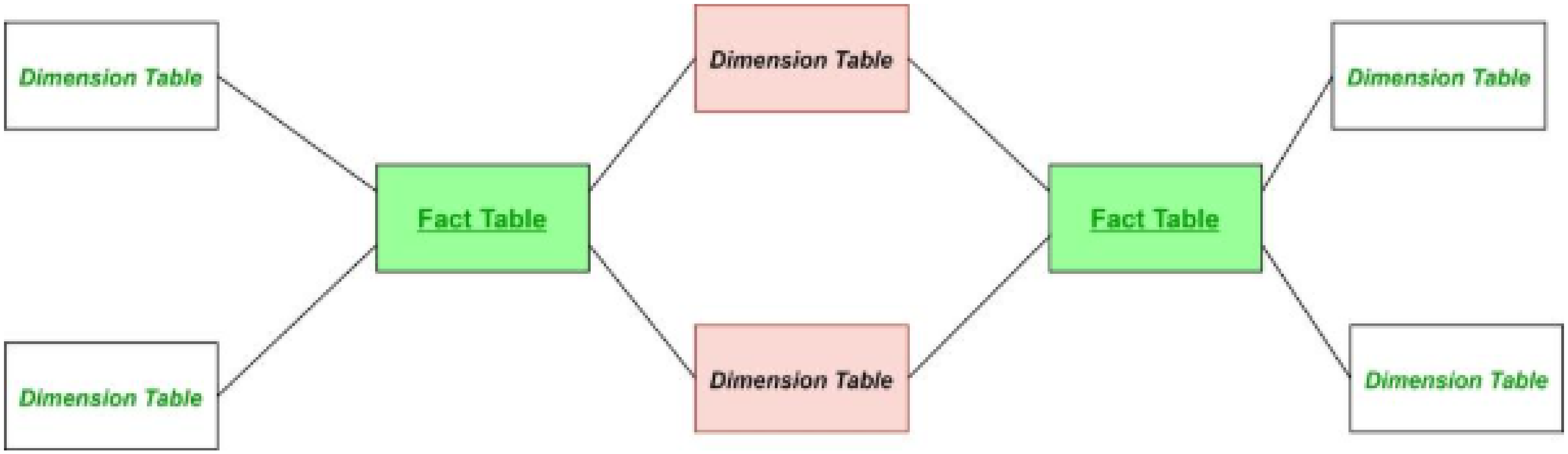
Fact Constellation in Data Warehouse modelling

- 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.

Fact Constellation in Data Warehouse modelling

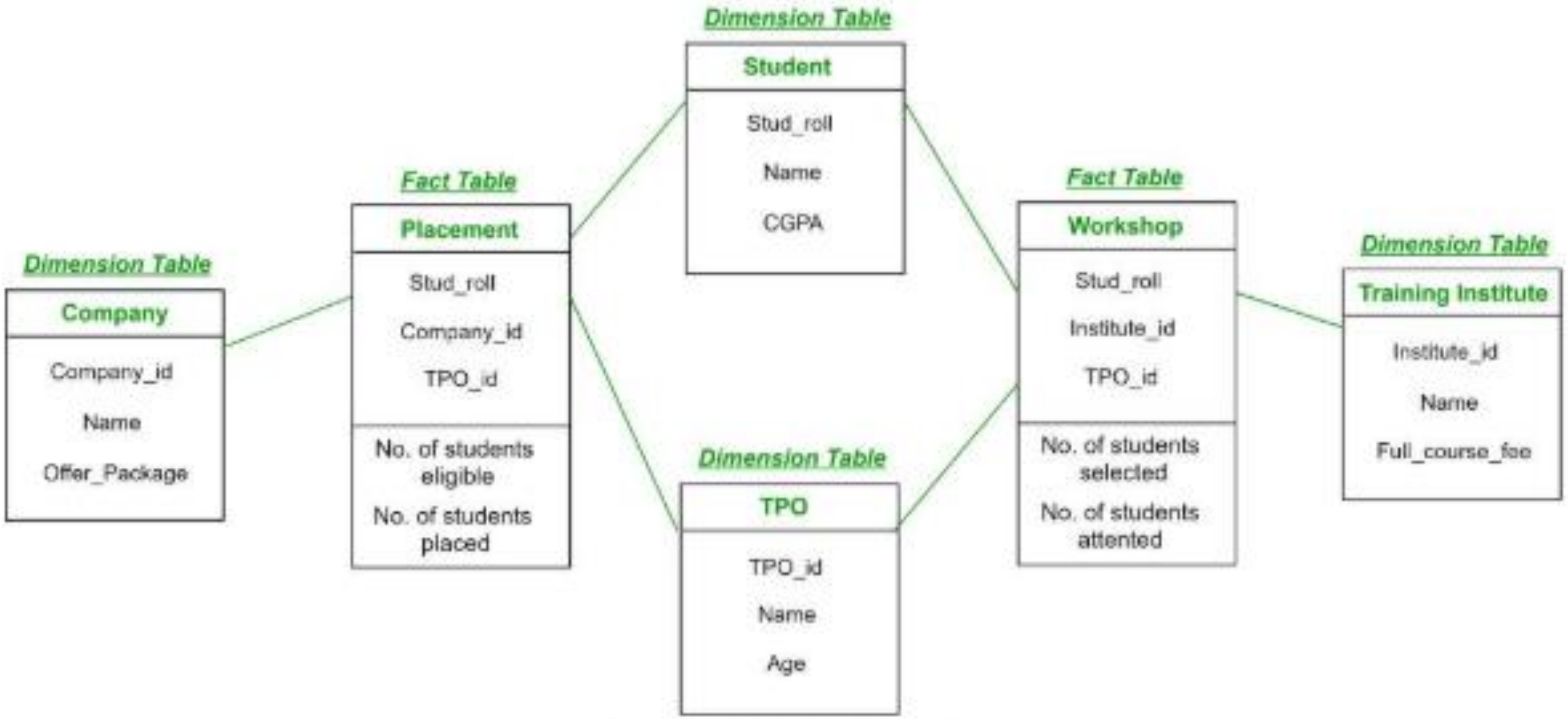
- Unlike simpler models like the Star Schema and Snowflake Schema, the Fact Constellation Schema consists of multiple fact tables that share common dimensional tables.
- This model is ideal for handling complex systems and large-scale analytical queries, offering flexibility for business intelligence and data mining.

Fact Constellation Schema



Fact Constellation Schema

Example Fast Constellation



Fact Constellation Example

Rolling-Up (Aggregation)

- Rolling-up refers to **data aggregation** — moving from a lower level of detail to a higher level in a hierarchy. It summarizes or groups data.
- **How it works**:
- You **reduce the level of detail** by grouping values.
- You go **up in the hierarchy** — e.g., from **days** → **months** → **quarters** → **years**.
- **Example (Time Hierarchy)**:
- You have sales data **daily**.
- You roll-up to view it by **month**, then **quarter**, then **year**.

Rolling-Up (Aggregation)

Date	Product	Sales Amount
01-Jan-24	A	100
02-Jan-24	A	120
→ Rolling-Up to Month		
Jan-24	A	220

Use Cases:
•Viewing total **monthly revenue** instead of daily.
•Generating **summary reports** for executives.

Rolling-Down (Drill-Down or Decomposition)

Rolling-down (also called **drill-down**) is the opposite of rolling-up. It involves breaking data down to a **lower level of granularity**.

How it works:

You **increase the level of detail** by viewing more granular data.

You go **down in the hierarchy** — e.g., from **years** → **quarters** → **months** → **days**.

Example (Time Hierarchy):

You see a **yearly** sales total.

You drill-down to see the **monthly**, then **daily** sales.

Rolling-Down (Drill-Down or Decomposition)

Year	Product	Sales Amount
2024	A	1500
→ Rolling-Down to Quarters		
Q1-24	A	400
Q2-24	A	500

Use Cases:

•Investigating **sales dips** by drilling down into specific months or days.

•Root-cause analysis in **performance reports**.

Rolling-Down (Drill-Down or Decomposition)

Time Hierarchy:
Year
 └── Quarter
   └── Month
     └── Day

Rolling-up: Day → Month → Quarter → Year
Rolling-down: Year → Quarter → Month → Day

Rolling-UP and Down

Feature	Rolling-Up	Rolling-Down
Direction	From detail → summary	From summary → detail
Purpose	Aggregation	Drill-down analysis
Hierarchy Flow	Bottom → Top	Top → Bottom
Example	Daily sales → Monthly sales	Yearly sales → Monthly → Daily

Class Task

Date	Product	Region	Sales
01-Jan-24	A	East	120
02-Jan-24	A	East	150
15-Feb-24	B	West	300
21-Feb-24	A	East	200
03-Mar-24	C	South	400
15-Mar-24	B	North	350

Rolling-Up (Aggregation)

Month	Region	Sales
2024-01	East	270
2024-02	East	200
2024-02	West	300
2024-03	North	350
2024-03	South	400

Rolling-Down (Drill-Down)

Quarter	Sales
2024Q1	1520

Month	Sales
2024-01	270
2024-02	500
2024-03	750

Rolling-Down (Drill-Down)

Day	Sales
2024-01-01	120
2024-01-02	150
2024-02-15	300
2024-02-21	200
2024-03-03	**400** ← Highest
2024-03-15	350

Highest Sales Day:
📅 **March 3, 2024** — 💰 **400**

THANKS