Dimensional Modeling

- **Dimensional modeling** (DM) names a set of techniques and concepts used in data warehouse design.
- Dimensional modeling is one of the methods of data modeling, that help us store the data in such a way that it is relatively easy to retrieve the data from the database.
- Dimensional modeling always uses the concepts of facts (measures), and dimensions (context).

Dimensional Models

- A denormalized relational model
  - Made up of tables with attributes
  - Relationships defined by keys and foreign keys
- Organized for understandability and ease of reporting rather than update
- Queried and maintained by SQL or special purpose management tools.

Benefits of dimensional modeling

Understand ability –

- Compared to the normalized model, the dimensional model is easier to understand and more intuitive.
- In dimensional models, information is grouped into coherent business categories or dimensions, making it easier to read and interpret.
- Simplicity also allows software to navigate databases efficiently.

Benefits of dimensional modeling

Query performance

- Dimensional models are more denormalized and optimized for data querying, while normalized models seek to eliminate data redundancies and are optimized for transaction loading and updating.
- The predictable framework of a dimensional model allows the database to make strong assumptions about the data which may have a positive impact on performance.
- Each dimension is an equivalent entry point into the fact table, and this symmetrical structure allows effective handling of complex queries.
- Query optimization is simple, predictable, and controllable.

Normalized and Denormalized Data in a Data Warehouse

In a data warehouse, **normalization** and **denormalization** refer to how data is structured for storage and access. These terms are based on the concepts of database normalization but are adapted to meet the goals of data warehousing.

1. Normalized Data

Definition: Normalized data is structured to minimize redundancy and ensure data integrity by dividing the data into multiple related tables. Each table contains data about a single entity, and relationships between tables are managed through foreign keys.

Characteristics:

- Data is broken down into smaller, logically grouped tables.
- Designed to reduce redundancy and improve consistency.
- Querying data often requires multiple joins across tables.
- Often used in **operational databases** (OLTP systems) for efficiency in data updates.

Normalized and Denormalized Data in a Data Warehouse

Example of Normalized Data: Suppose we want to store customer orders.

Customers Table:

Customer_ID	Customer_Name	Email
1	John Smith	john@example.com
2	Jane Doe	jane@example.com

Products Table:

Product_ID	Product_Name	Price
P1	Laptop	1000
P2	Mouse	25

Orders Table:

Order_ID	Customer_ID	Order_Date
101	1	2025-01-20
102	2	2025-01-21

Order Details Table:

Order_ID	Product_ID	Quantity
101	P1	1
101	P2	2

To retrieve a complete order summary, you'd need to join these tables, which might be time-consuming for complex queries.

Normalized and Denormalized Data in a Data Warehouse

2. Denormalized Data

Definition: Denormalized data combines related data into fewer tables, often resulting in some redundancy. It is optimized for faster query performance, especially for analytical purposes.

Characteristics:

- Data is consolidated into fewer tables.
- Designed to speed up querying and reporting.
- May introduce some redundancy but simplifies data access.
- Commonly used in **data warehouses** (OLAP systems).

Normalized and Denormalized Data in a Data Warehouse

Example of Denormalized Data: Instead of splitting data into multiple tables, we create a single denormalized table:

•**Orders Summary Table**

Order_ID	Customer_Name	Email	Order_Date	Product_Name	Price	Quantity
101	John Smith	john@example.com	2025-01-20	Laptop	1000	1
101	John Smith	john@example.com	2025-01-20	Mouse	25	2

•With this structure, reporting and analytics are faster because all the necessary data is in a single table, avoiding joins.

Normalized vs. Denormalized Data

Aspect	Normalized Data	Denormalized Data
Use Case	Transactional systems (OLTP).	Analytical systems (OLAP).
Performance Goal	Efficient updates and inserts.	Faster read/query performance.
Data Redundancy	Low (minimal redundancy).	High (some redundancy).
Complexity	Complex queries (requires joins).	Simpler queries (no or fewer joins).

Benefits of dimensional modeling

Extensibility

- Dimensional models are scalable and easily accommodate unexpected new data.
- Existing tables can be changed in place either by simply adding new data rows into the table or executing SQL alter table commands.
- No queries or applications that sit on top of the data warehouse need to be reprogrammed to accommodate changes

Entity-Relationship vs. Dimensional Models

Relational Modeling	Dimensional Modeling
Data is stored in RDBMS	Data is stored in RDBMS or Multidimensional databases
Tables are units of storage	Cubes are units of storage
Data is normalized and used for OLTP. Optimized for OLTP processing	Data is de normalized and used in data warehouse and data mart. Optimized for OLAP
Several tables and chains of relationships among them	Few tables and fact tables are connected to dimensional tables
Volatile (several updates) and time variant	Non volatile and time invariant
Detailed level of transactional data	Summary of bulky transactional data (Aggregates and Measures) used in business decisions
SQL is used to manipulate data	MDX is used to manipulate data
Normal Reports	User friendly, interactive, drag and drop multidimensional OLAP Reports

(Multidimensional Expressions)

MDX (Multidimensional Expressions)

MDX (Multidimensional Expressions) is a query language used in data warehouses, specifically for querying and manipulating multidimensional data stored in **OLAP (Online Analytical Processing)** cubes. It is similar to SQL but designed for multidimensional structures rather than relational tables.

Key Features of MDX:

1.Multidimensional Queries: MDX is used to retrieve and analyze data from OLAP cubes, which organize data into dimensions (e.g., Time, Geography, Product) and measures (e.g., Sales, Profit).

2.Hierarchical Data Access: It supports hierarchical structures like Year → Quarter → Month → Day, allowing for drill-down or roll-up analysis.

3.Calculated Members: MDX enables the creation of calculated measures or members for advanced analytics, such as "Year-to-Date Sales" or "Percentage Growth."

4.Aggregations and Slicing: MDX can aggregate data and slice specific portions of the cube for analysis, such as filtering sales by region or product category.

Time Variant Vs Time Invariant

Aspect	Time-Variant	Time-Invariant
Definition	Data that changes over time and captures historical values.	Data that remains constant and does not change over time.
Timestamp	Includes a timestamp to track when the data was valid.	No timestamp; data is stored as static reference information.
Purpose	Used for trend analysis and tracking changes over time.	Used for referencing static information.
Historical Data	Stores historical data; old values are not overwritten.	Does not store historical data; updates overwrite old values.
Examples	Sales data, salary history, stock prices.	Product names, employee names, geographical data.

MDX (Multidimensional Expressions)

Basic Structure of an MDX Query:

An MDX query has two main clauses:

- **SELECT Clause:** Specifies the dimensions and measures to retrieve.
- **FROM Clause:** Specifies the OLAP cube to query.
- Example MDX Query:

```
SELECT {[Measures].[Sales], [Measures].[Profit]} ON COLUMNS,
{[Time].[2025].[Q1], [Time].[2025].[Q2]} ON ROWS
FROM [SalesCube] WHERE ([Region].[USA])
```

- *COLUMNS: Retrieves "Sales" and "Profit" measures.*
- *ROWS: Retrieves data for Q1 and Q2 of 2025.*
- *FROM: Queries the cube named SalesCube.*
- *WHERE: Filters data to include only the USA region.*

MDX VS SQL

Aspect	MDX	SQL
Purpose	Querying multidimensional data in OLAP cubes.	Querying relational data in tables.
Structure	Hierarchical, multidimensional.	Tabular, relational.
Key Elements	Dimensions, measures, calculated members.	Tables, rows, columns.
Use Case	Analytical querying (e.g., trends, KPIs).	Operational querying (e.g., transactions).

Entity-Relationship vs. Dimensional Models

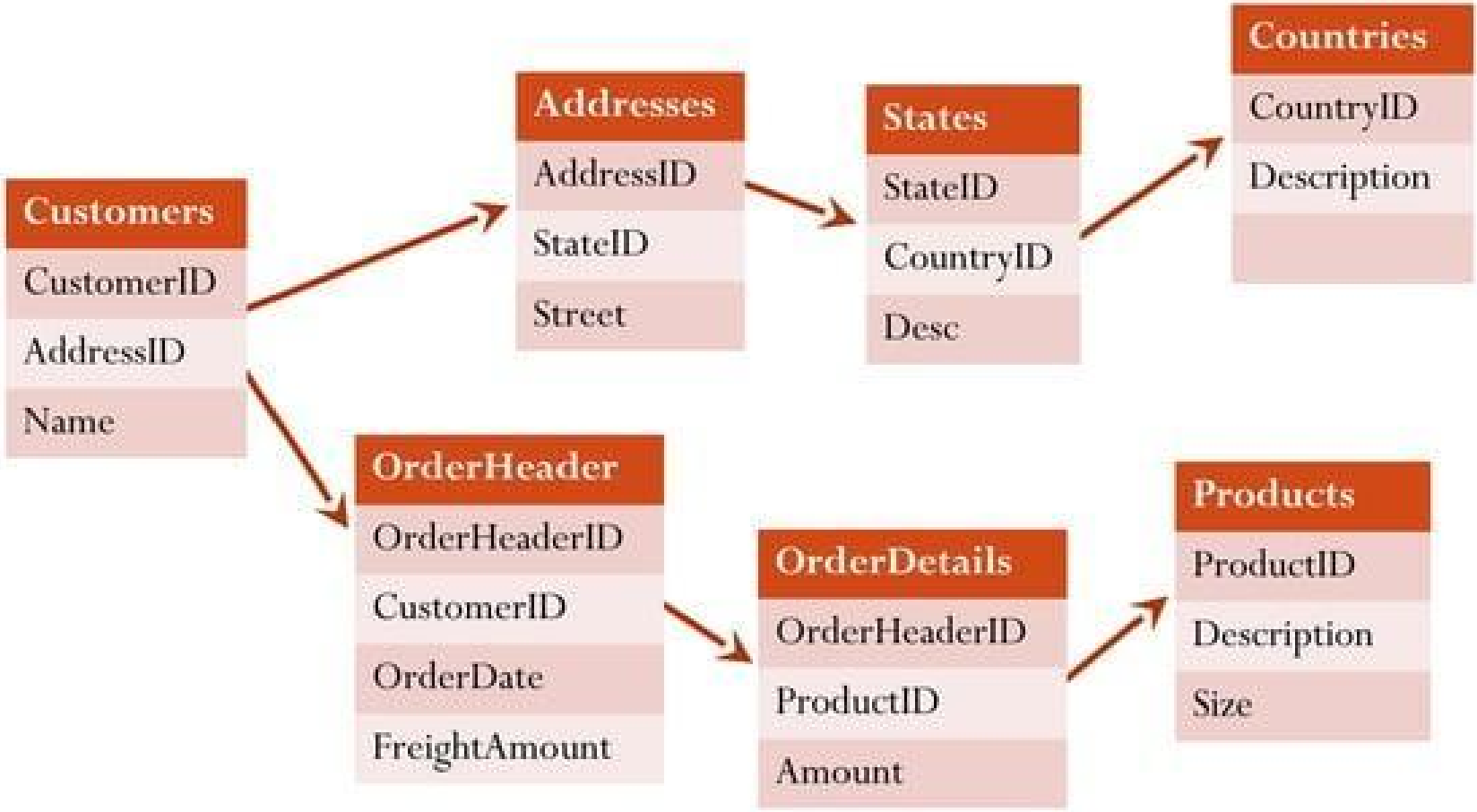
- One table per entity
- Minimize data redundancy
- Optimize update
- The Transaction Processing Model

- One fact table for data organization
- Maximize understandability
- Optimized for retrieval
- The data warehousing model

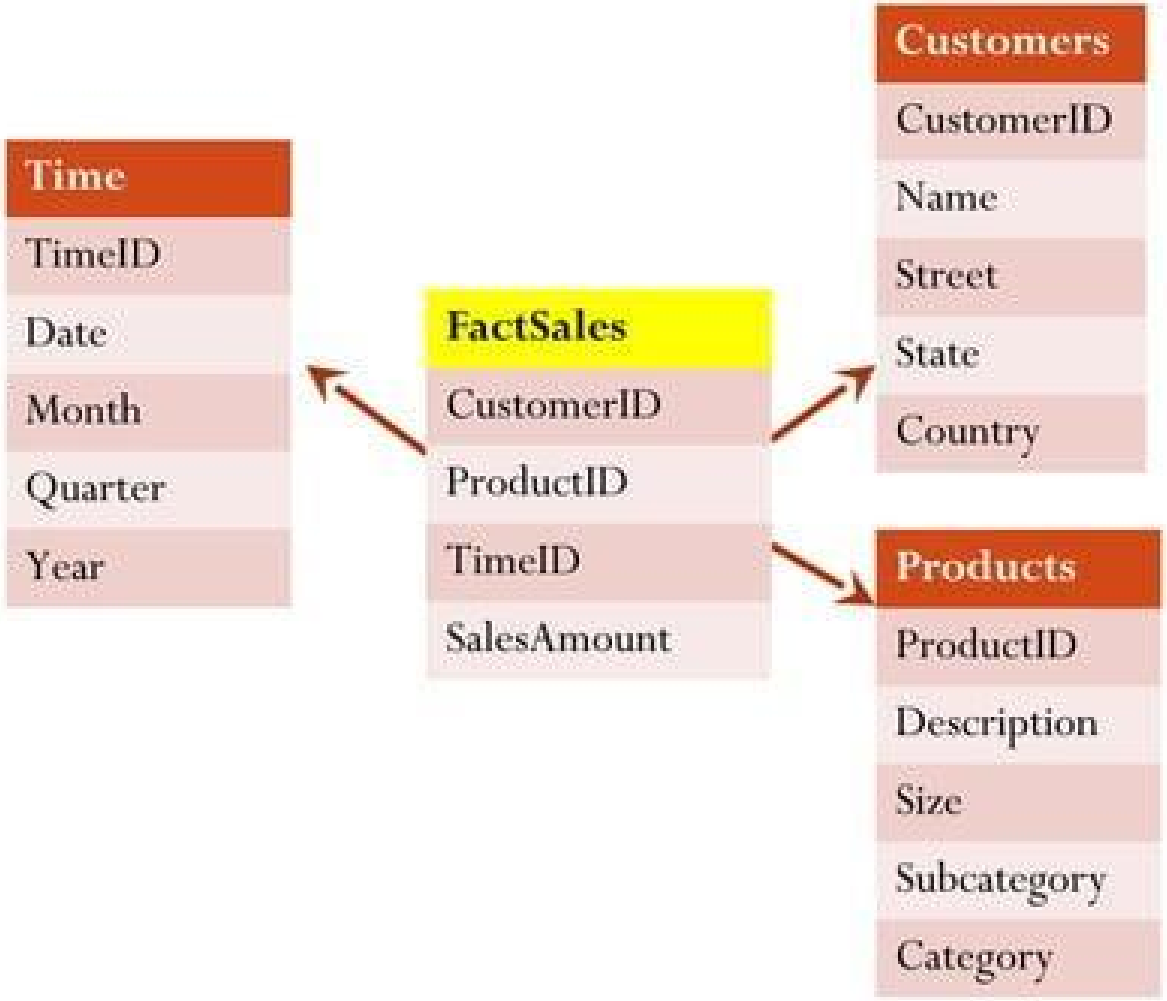
Facts & Dimensions

- There are two main types of objects in a dimensional model
  - **Facts** are quantitative measures that we wish to analyse and report on.
  - **Dimensions** contain textual descriptors of the business. They provide *context* for the facts.

A Transactional Database



A Dimensional Model



Fact Tables

- A fact table stores quantitative information for analysis and is often denormalized.
- Contains two or more foreign keys.
- Tend to have huge numbers of records.
- Useful facts tend to be numeric and additive.
- A fact table holds the data to be analyzed, and a dimension table stores data about the ways in which the data in the fact table can be analyzed. Thus, the fact table consists of two types of columns. The foreign keys column allows joins with dimension tables, and the measures columns contain the data that is being analyzed.

Example :Fact Table

- Suppose that a company sells products to customers. Every sale is a fact that happens, and the fact table is used to record these facts. For example:
-

Time ID	Product ID	Customer ID	Unit Sold
4	17	2	1
8	21	3	2
8	4	1	1
5	20	2	5
3	4	4	7

Dimension Table

- A dimension table stores attributes, or dimensions, that describe the objects in a fact table.
- A data warehouse organizes descriptive attributes as columns in dimension tables.
- For Example: A customer dimension's attributes could include first and last name, birth date, gender, etc., or a website dimension would include site name and URL attributes.
- A dimension table has a primary key column that uniquely identifies each dimension record (row).

Example:Dimension Table

Customer ID	Name	Gender	Income	Education	Region
1	Brian Edge	M	2	3	4
2	Fred Smith	M	3	5	1
3	Sally Jones	F	1	7	3

- The dimension table is associated with a fact table using this PRIMARY key.
- It is not uncommon for a dimension table to have 50 to 100 attributes;
- Dimension tables tend to have fewer rows than fact tables

- Dimension tables are referenced by fact tables using keys.
- When creating a dimension table in a data warehouse, a system-generated key is used to uniquely identify a row in the dimension. This key is also known as a surrogate key.
- The surrogate key is used as the primary key in the dimension table.
- The surrogate key is placed in the fact table and a foreign key is defined between the two tables. When the data is joined, it does so just as any other join within the database.

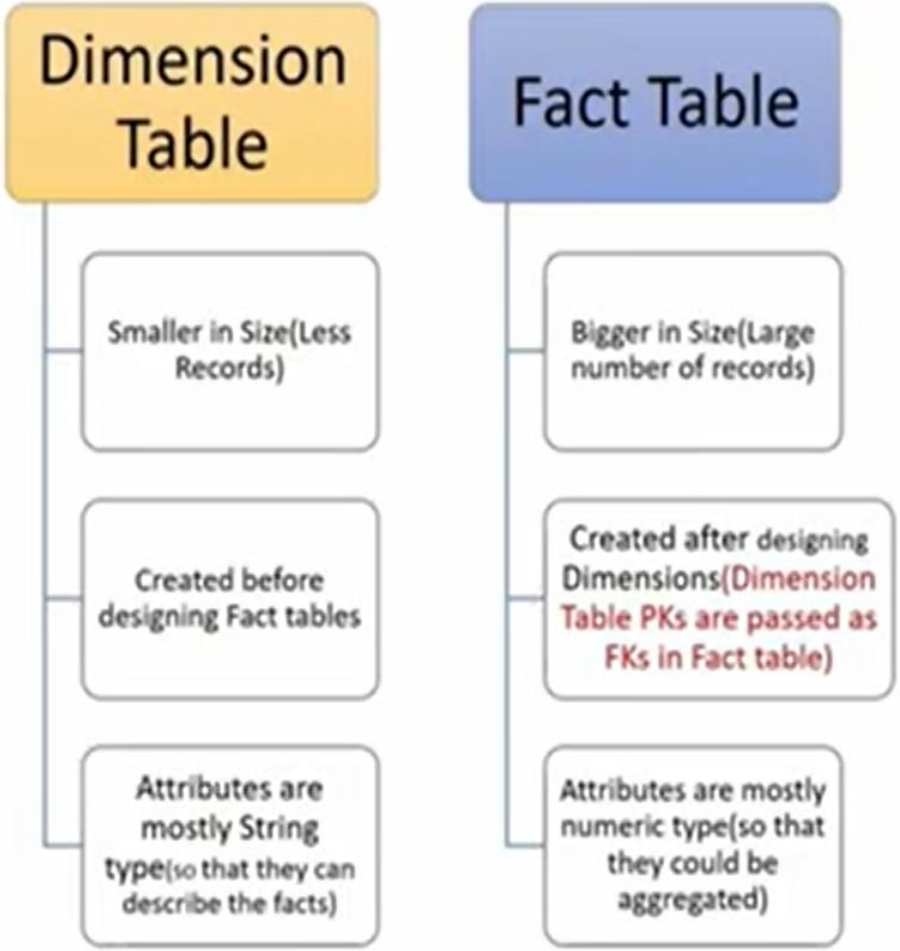
- describe the "who, what, where, when, how, and why" associated with the event.

Product Dimension
Product Key (PK)
SKU Number (Natural Key)
Product Description
Brand Name
Category Name
Department Name
Package Type
Package Size
Abrasive Indicator
Weight
Weight Unit of Measure
Storage Type
Shelf Life Type
Shelf Width
Shelf Height
Shelf Depth
...

Facts and Dimensions

Criteria	Fact Attributes	Dimension Attributes
Purpose	Measurements for reporting or analysis	Constraints or qualifiers for the measurements
Data type	Additive or semi-additive quantitative data	Textual, descriptive
Size	Larger number of records	Smaller number of records
Reporting use	Main report contents	Row or report headers
Examples	Measurements for sales	About time, people, departments, objects, geographic units

Dimensions and Facts in Data Warehouse



Types of Dimension

Dimensions are of various types:

- Role Playing Dimensions
- Slowly Changing Dimensions
- Junk Dimensions

Role Playing Dimensions

Def : When Single Dimension is referenced several times in Fact table and each reference linking to a different role for the dimension.

-> Same Dimension is used several times in Fact table with different meanings every time.

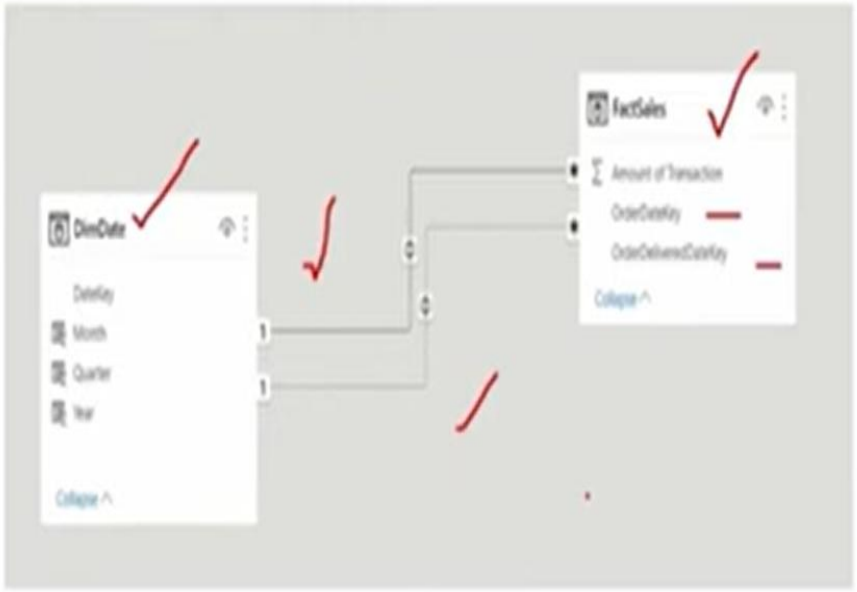
-> Important in Data Modelling

We have two tables Fact Sales and Dim Date Dimension table. Now, we have Order_Date_Key and Ship_Date_Key as date columns which are related to Dim Date through Primary Key , Foreign Key Relationship

Key: Date Dimension is playing the multiple roles.

** It depends on which Date Key columns of Facts Sales table we are going to join then Date Dimension will play the respective role

Role Playing Dimensions (Scenario Explained using Power BI Data Model)



Tables used: Fact Sales & Date Dimension.

Here, DimDate table is related to Fact Sales table using two relationships in which one is active (solid line) and one is In active (dashed line).

Use Case:

Active Case

When we need to showcase count of orders placed on each day we can create report by simply dragging the fields and perform aggregation as count

In Active Case:

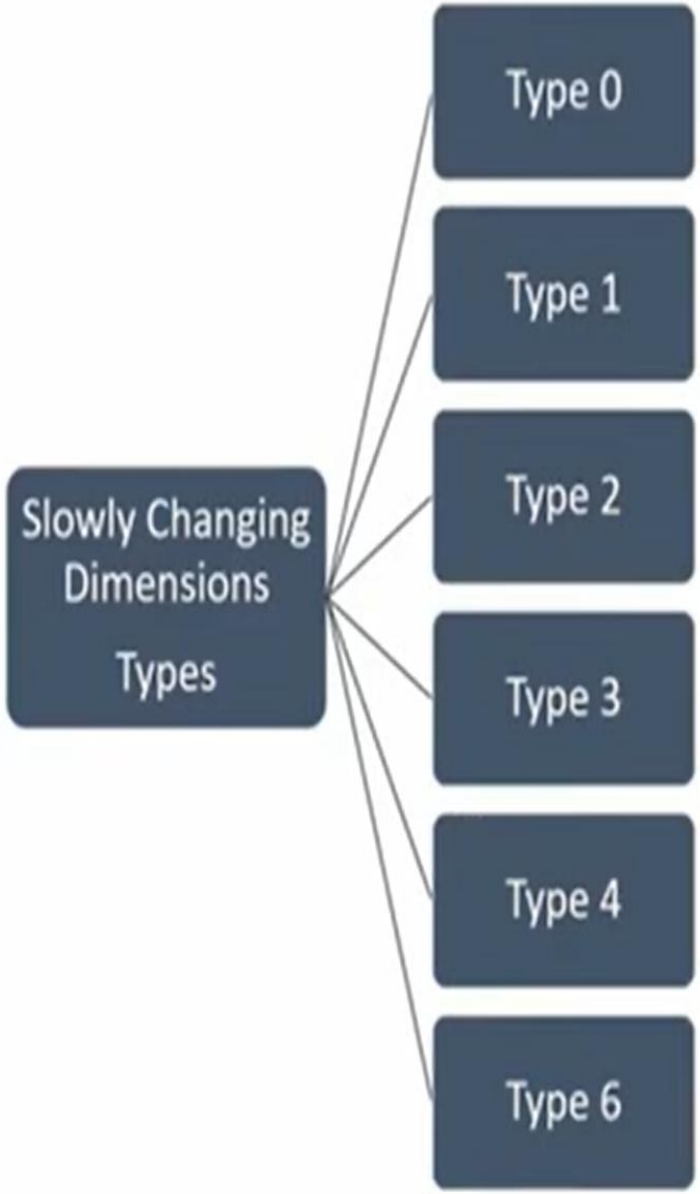
But when we want to showcase count of orders delivered each day we need to use USERELATIONSHIP(DAX function in power BI) to utilize our inactive relationship

Slowly Changing Dimensions

Def: Slowly Changing Dimension is a dimension in which values of attributes changes over time.

Why Slowly Changing Dimension are required:

Data Warehouse is designed to analyze the historical data hence we need some approach to handle changes in Dimensions.



Slowly Changing Dimensions(Type 0)

Type 0(SCD)

- Attributes of dimension which never change their value
- Original Value of attributes remains the same

Example

Employee ID	Employee Name	Employee Location	Employee DOB	Employee Joining Date
100	ABC	India	1980-01-01	2020-01-01

Slowly Changing Dimensions(Type 1)

Type 1(SCD)

- Simply over writes the old attribute values with new values
- Advantage: Easy to Maintain
- Disadvantages:
  - Historical data is not maintained
  - All existing calculations on updated column requires re-calculation

Example

Employee ID	Employee Name	Employee Location
100	ABC	India

After Overwrite:

Employee ID	Employee Name	Employee Location
100	ABC	USA

It cannot maintained the historical data

Slowly Changing Dimensions(Type 2)

Type 2(SCD)

[General Method for tracking the changes]

- When there is update in attribute value, New row is added in dimension table
- Additional columns required: **Surrogate Key, Effective_From_Date, Effective_To_Date, Flag**
- Advantage: Unlimited history is maintained
- Disadvantage: Not a good choice when there are frequent changes in dimension model

Example

Surrogate Key

Employee Key	Employee ID	Employee Name	Employee Location	Effective_From_Date	Effective_To_Date	Flag
1	100	ABC	India	2022-01-01	2022-03-22	0
2	100	ABC	USA	2022-03-23	9999-12-31	1

Flag 0 means in active record
We can maintained the historical data
Drawback: Size of dimension table increases

Slowly Changing Dimensions(Type 3)

Type 3(SCD)

- For tracking the history separate new column is added rather than adding the new row
- Advantage: Useful when we need to track recent historical changes
- Disadvantages: Limited history meaning we are keeping only last version of history

Example

Newly Added Column

Employee ID	Employee Name	Current Location	Previous Location
100	ABC	USA	India

Limited History data is maintained. If gained location changed

Slowly Changing Dimensions(Type 4)

Type 4(SCD)

- In Existing dimension table we updates the record like Type 1 SCD and Separate history tables is maintained to track the history of changes
- Advantage: Separating the historical table keep the original dimension table smaller in size and we can also perform analysis on historical data.

Example

Dimension Table

Employee ID	Employee Name	Location
100	ABC	USA

History Table (Type 4)

Employee Key	Employee ID	Employee Name	Employee Location	Date Created
1	100	ABC	India	2022-01-01
2	100	ABC	USA	2022-03-23

Slowly Changing Dimensions(Type 4)

Type 4(SCD)

- In Existing dimension table we updates the record like Type 1 SCD and Separate history tables is maintained to track the history of changes
- Advantage: Separating the historical table keep the original dimension table smaller in size and we can also perform analysis on historical data.

Example

Dimension Table

Employee ID	Employee Name	Location
100	ABC	USA

History Table (Type 4)

Employee Key	Employee ID	Employee Name	Employee Location	Date Created
1	100	ABC	India	2022-01-01
2	100	ABC	USA	2022-03-23

Slowly Changing Dimensions(Type 6)

Type 6(SCD) Hybrid Approach

- Implementation of all types: Type 1,Type 2 and Type 3 techniques(1+2+3)
- Advantages: We can perform complex analysis easily using this approach.

Current and Previous locations are same

Employee Key	Employee ID	Employee Name	Employee Location	Employee Current Location	Effective_From_Date	Effective_To_Date	Flag
1	100	ABC	India	India	2022-01-01	9999-12-31	1

Type1

1 + 2

Employee Key	Employee ID	Employee Name	Employee Location	Employee Current Location	Effective_From_Date	Effective_To_Date	Flag
1	100	ABC	India	USA	2022-01-01	2022-03-22	0
2	100	ABC	USA	USA	2022-03-23	9999-12-31	1

Type1

Type 2: adding surrogate Key

Type1= Current and Previous Value

Type3: maintaining the Emploee Current and previous value

Junk Dimensions

Like True/False or Yes/No type of columns

Definition: Combination of several low cardinality flags and attributes into a single dimension table rather than modeling them as separate dimensions

Operational or OLTP system contains many flags and transactional codes which are difficult to accommodate in existing conventional dimensions like Customer and Employee also if we keep them in fact table then size of fact table will increase many folds also there is no point of creating separate dimensions for them

Solution

Best solution is to keep all those flags and indicator columns in a single dimension

Advantages:

1) Helps in Reducing the fact table size
2) Making Dimensional modelling easy as we are not creating unnecessary dimensions

Low cardinality means less unique values

Junk Dimensions

Sample Fact Transaction table

Initial Structure of Fact Sales Table

Transaction ID	Date key	Sales Employee ID	Product ID	Quantity	Sales Amount	Shipped	Delivered	Received
T100	2022-01-01	100	P231	2	200	Y	Y	Y
T101	2022-01-02	101	P232	1	100	Y	N	N

We have 2 possible values for Shipped, Delivered and Received columns hence Junk Dimension contains 2*2*2= 8 rows

We have replaced 3 columns with single column

Junk Dimension

Txn_Status_Key	Shipped	Delivered	Received
1	Y	Y	Y
2	Y	Y	N
3	Y	N	Y
4	Y	N	N
5	N	Y	Y
6	N	Y	N
7	N	N	Y
8	N	N	N

1...*

Final Structure of Fact Sales Table

Transaction ID	Date key	Sales Employee ID	Product ID	Quantity	Sales Amount	Txn_Status_Key
T100	2022-01-01	100	P231	2	200	1
T101	2022-01-02	101	P232	1	100	4

Degenerate Dimensions

Definition: Degenerate Dimensions is a modelling technique in which high cardinality attributes are placed as a part of fact table.

Because of High Cardinality

1) There would be no storage saving if we create separate dimension on the contrary storage is required for surrogate keys

2) Also, Join between this dimension and fact table would also be expensive

If we will store these attributes as part of fact table significant amount of space is saved and query cost could also be reduced

In our Data Model, suppose *there is a dimension growing at a pace of fact table* then that dimension could be a good candidate for degenerate dimension.

Degenerate Dimensions

Transaction Table(OLTP)

Order Number	Invoice Number	Product Description	Customer Name	Order Quantity	Order Amount
O100001	IN0000001	Mobile	ABC	1	20000
O100002	IN0000001	Television	PQR	1	40000

These are high cardinality columns and there is no point of storing them as a separate dimension

Hence, these columns are stored as part of fact tables only.

Conformed Dimensions

Definition: Conformed dimension has same meaning with all the related fact tables
Conformed dimensions act as consistent interface in combining data from different fact tables

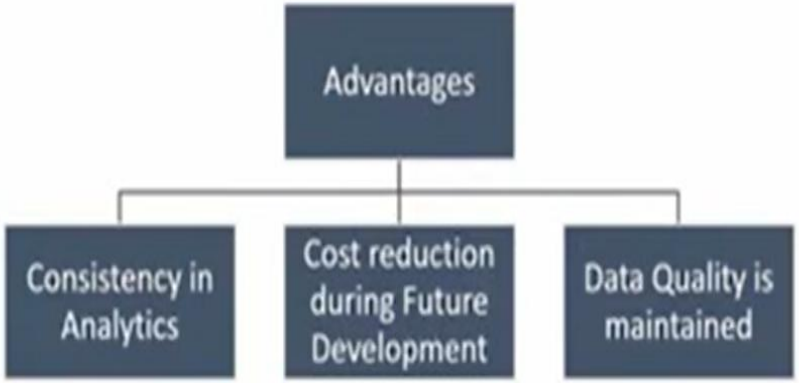
Master Table

Key Pointers

- Data Warehouse will not function as an integrated unit in the absence of conformed dimensions

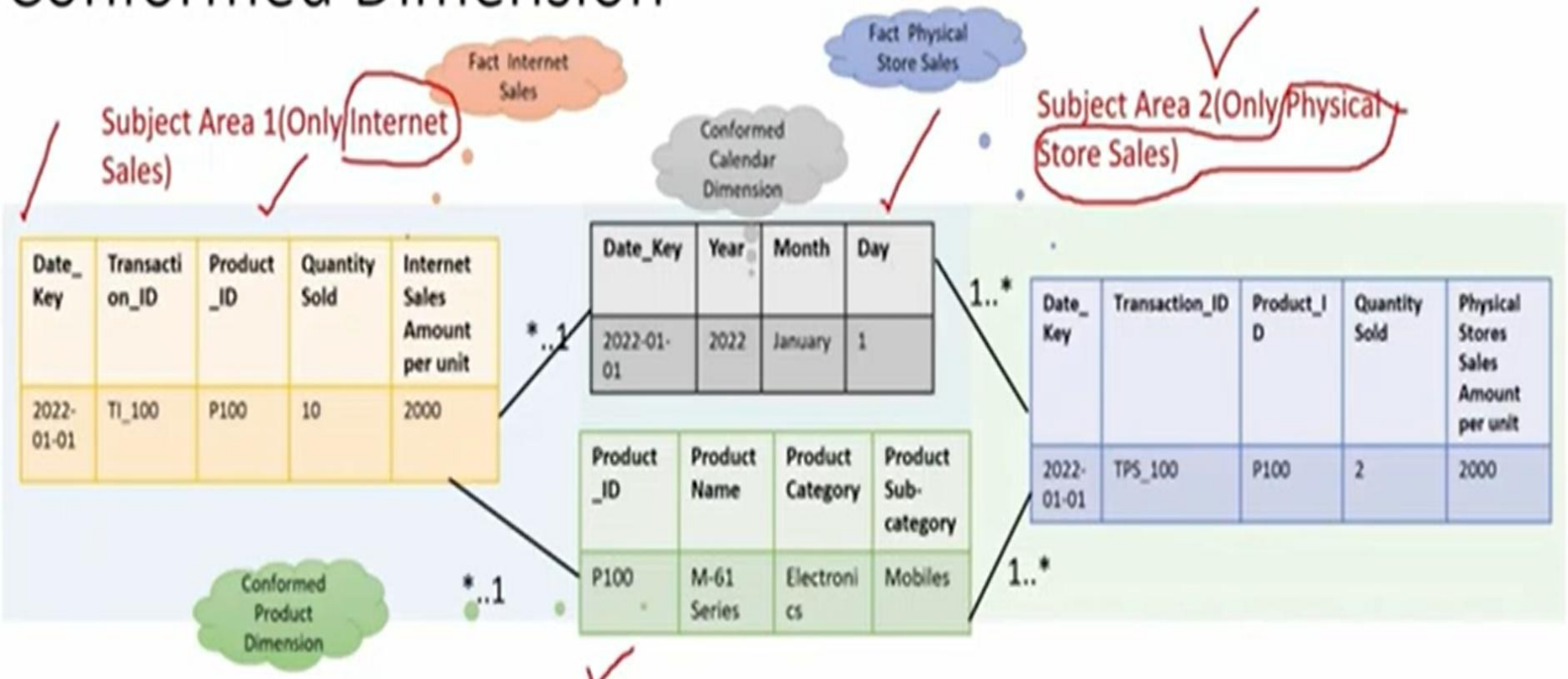
Important Practices:

- Conformed Product Dimension must contain master list of products with appropriate level of detail
- Conformed Calendar dimension should contain all the relevant columns required in analysis of fact tables of different subject areas



Note: Creating the conformed dimension is one of the important step during designing of Data Warehouse.
[Primary responsibility of central Data Warehouse team and provide it as resource for rest of the enterprise]

Conformed Dimension

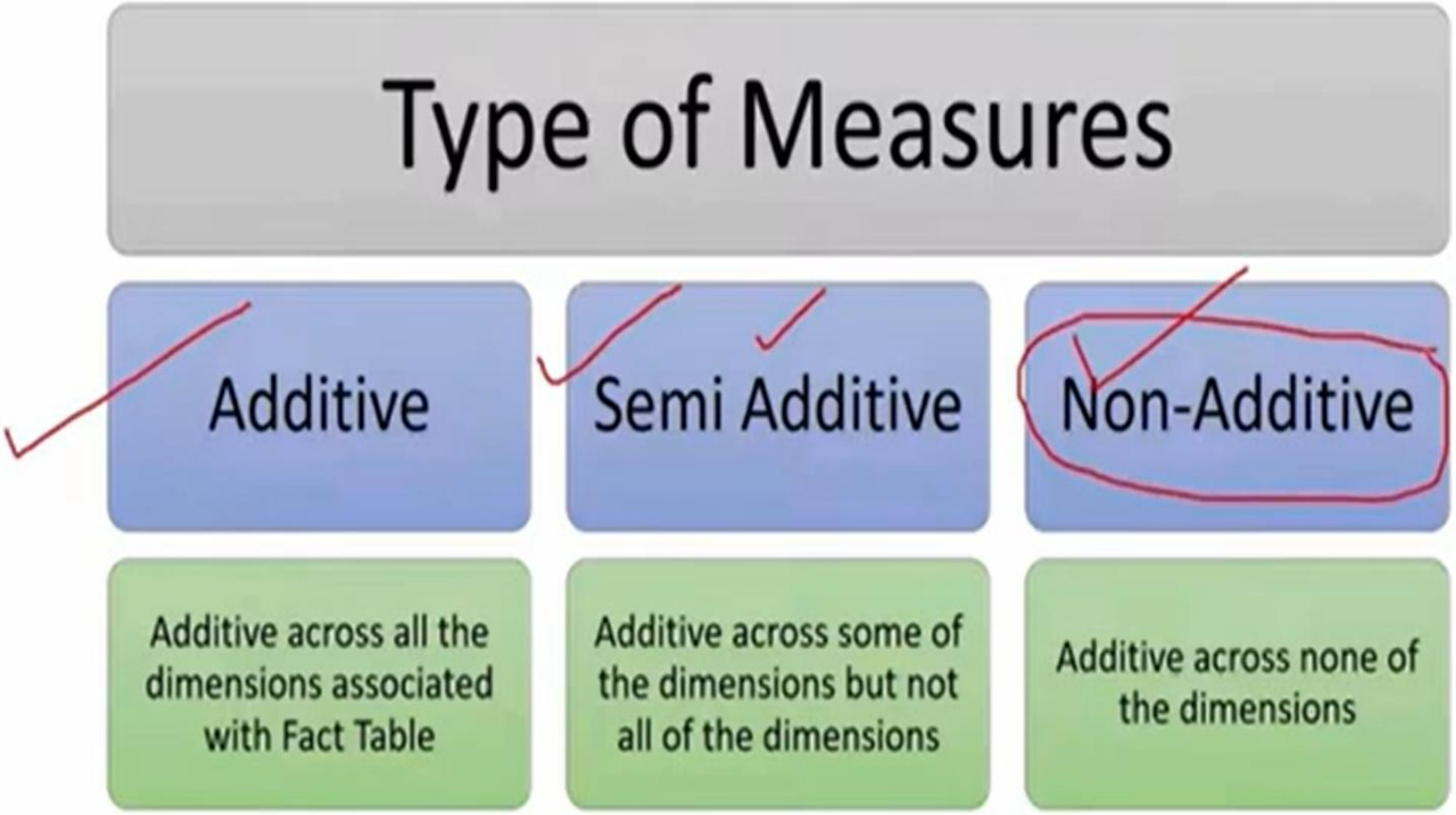


Now, we can easily drill across these fact tables and generate a report like total amount of transactions completed per day through online(e-commerce) channel and total amount of transactions completed via physical stores for a particular product.

Date_Key	Product Name	Internet Sales Amount	Physical Stores Sales Amount
2022-01-01	M-61 Series	20000	4000

Types of Measures in Fact Tables

Numeric Measures in Fact Tables are categorized in 3 primary ways



Types of Measures in Fact Tables

Example:
Sales Amount measure is Additive across all the dimensions associated with Fact Table

Date_Key	Sales Person ID	Region	Sales Amount($)
2022-01-01	10	Europe	2000
2022-01-02	10	Europe	3000
2022-01-03	20	Asia	10000
2022-01-04	30	America	2300

Types of Measures in Fact Tables

Example:

Account Balance is semi-additive measure as it is not additive against Date dimension

Date_Key	Account ID	Account Balance($)
2022-01-03	0001	100
2022-01-04	0001	150
2022-01-05	0001	150
2022-01-06	0001	130
2022-01-07	0001	130

Deposit of 50$

Withdrawal of 20$

Need to identify account balance at the end of the week

Now, here we cannot say account balance for account id **0001** at the end of the week is **660$**, it is actually **130$** hence Account Balance column is not additive against Date Dimension

Types of Measures in Fact Tables

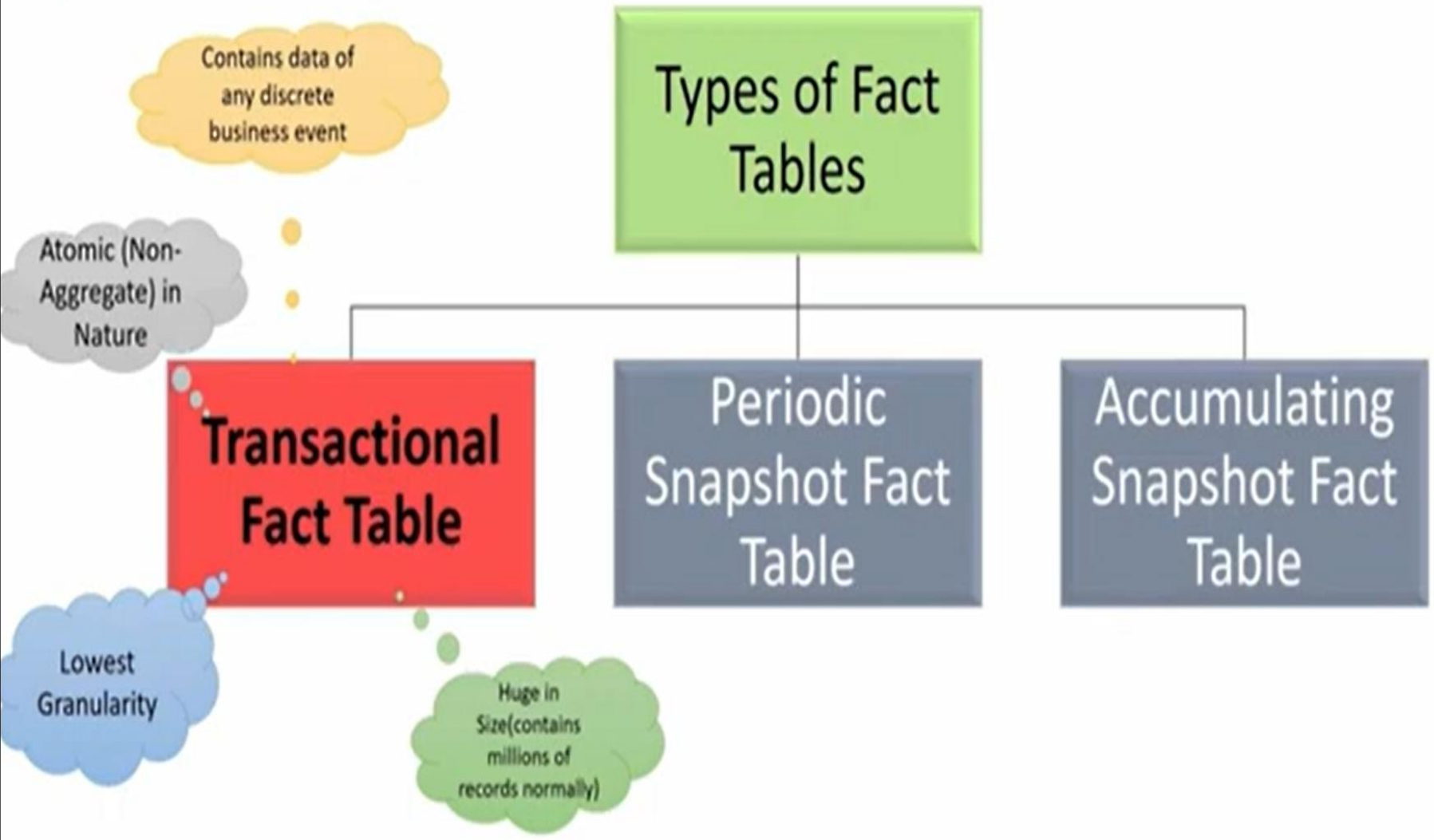
Ratio and % columns

Example:
Gross Profit Margin is Non-Additive measure as it is not additive against any of the dimensions associated with Fact Table

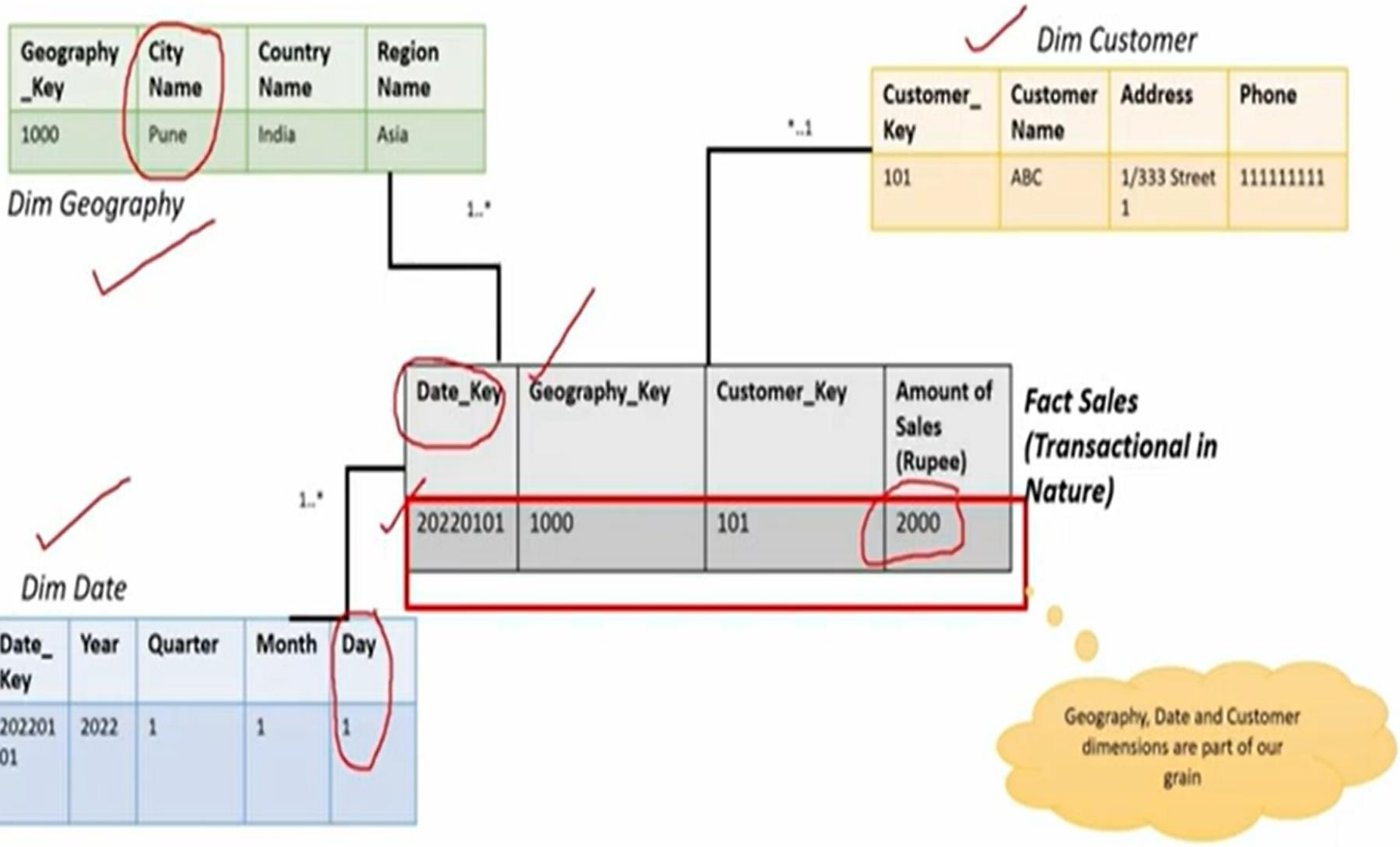
Quarter	Organization ID	Gross Profit Margin (Revenue - COGS/Revenue) * 100
Q12021	O100	20%
Q12021	O101	18%
Q22021	O100	30%
Q22021	O101	25%
Q32021	O100	17%
Q32021	O101	19%
Q42021	O100	33%
Q42021	O101	15%

We cannot perform summation over Gross Profit Margin across time or organization dimension

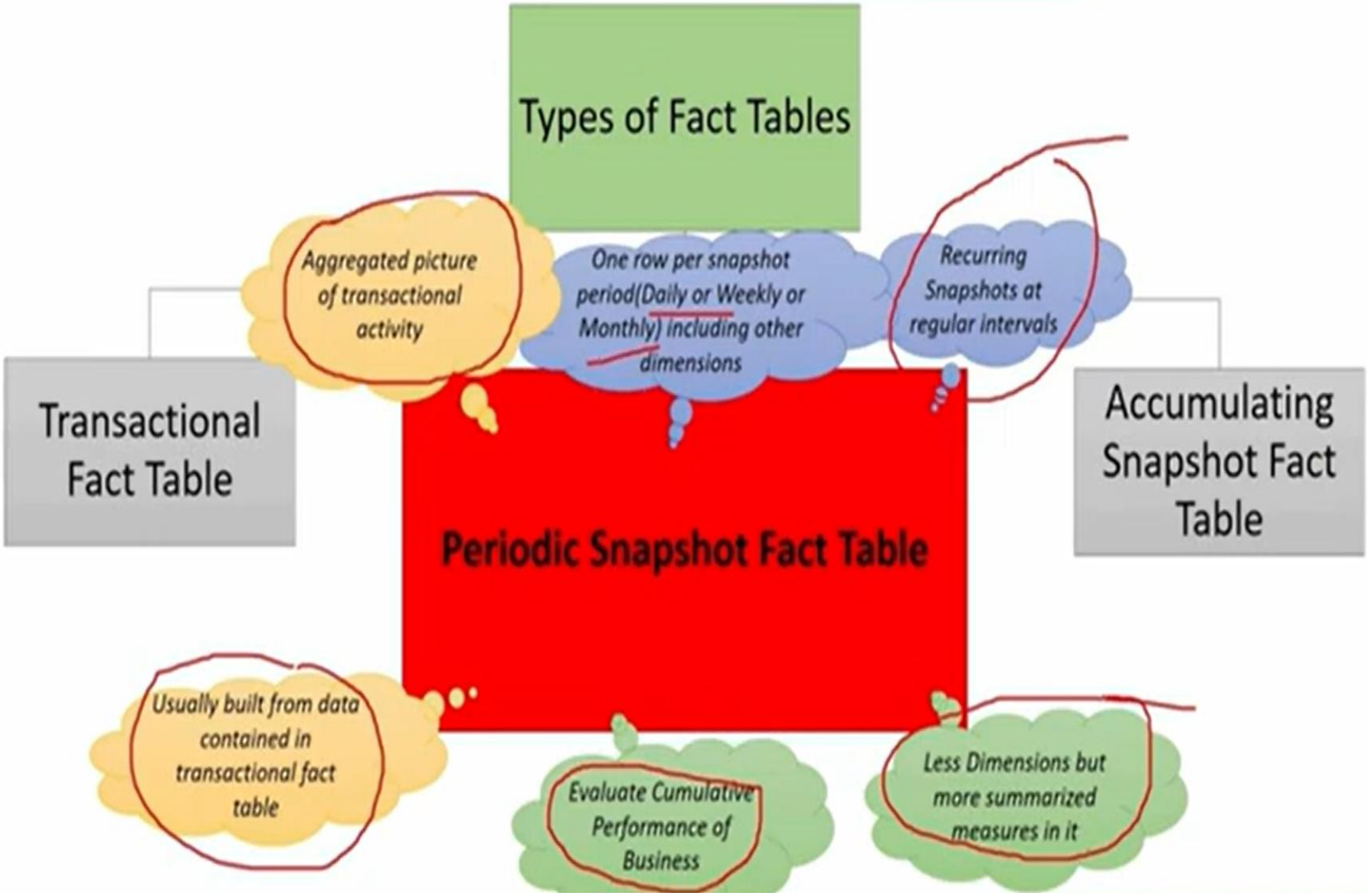
Types of Fact Tables



Types of Fact Tables : Transactional Fact Table



Types of Fact Tables : Periodic Snapshot Fact Table



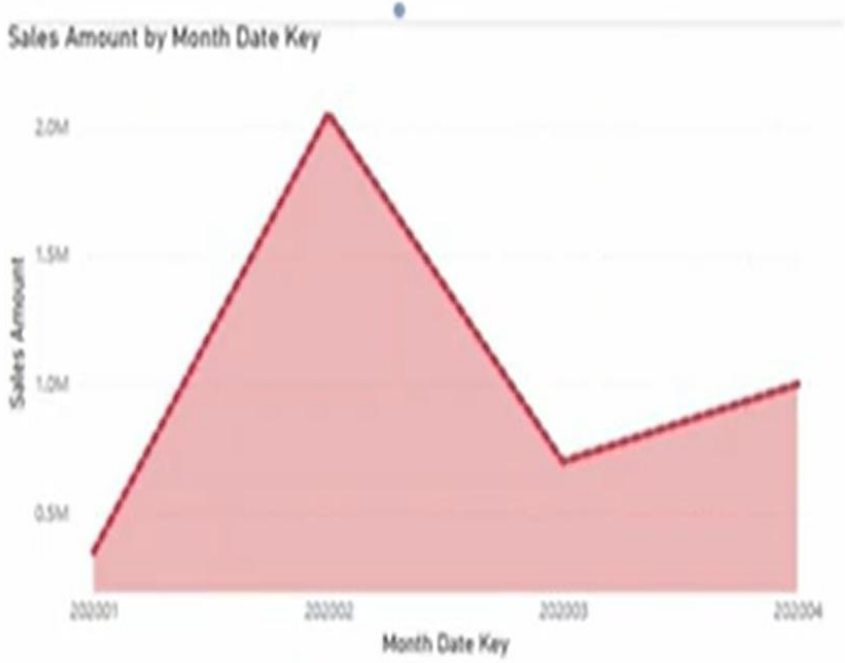
Types of Fact Tables : Periodic Snapshot Fact Table

Monthly Snapshot of Store Sales

Month Date Key	Product Key	Quantity	Sales Amount ($)
202001	1001	2000	200000
202001	1002	3000	150000
202002	1003	6000	1800000
202002	1002	5000	250000
202003	1004	3000	600000
202003	1001	1000	100000
202004	1004	5000	1000000

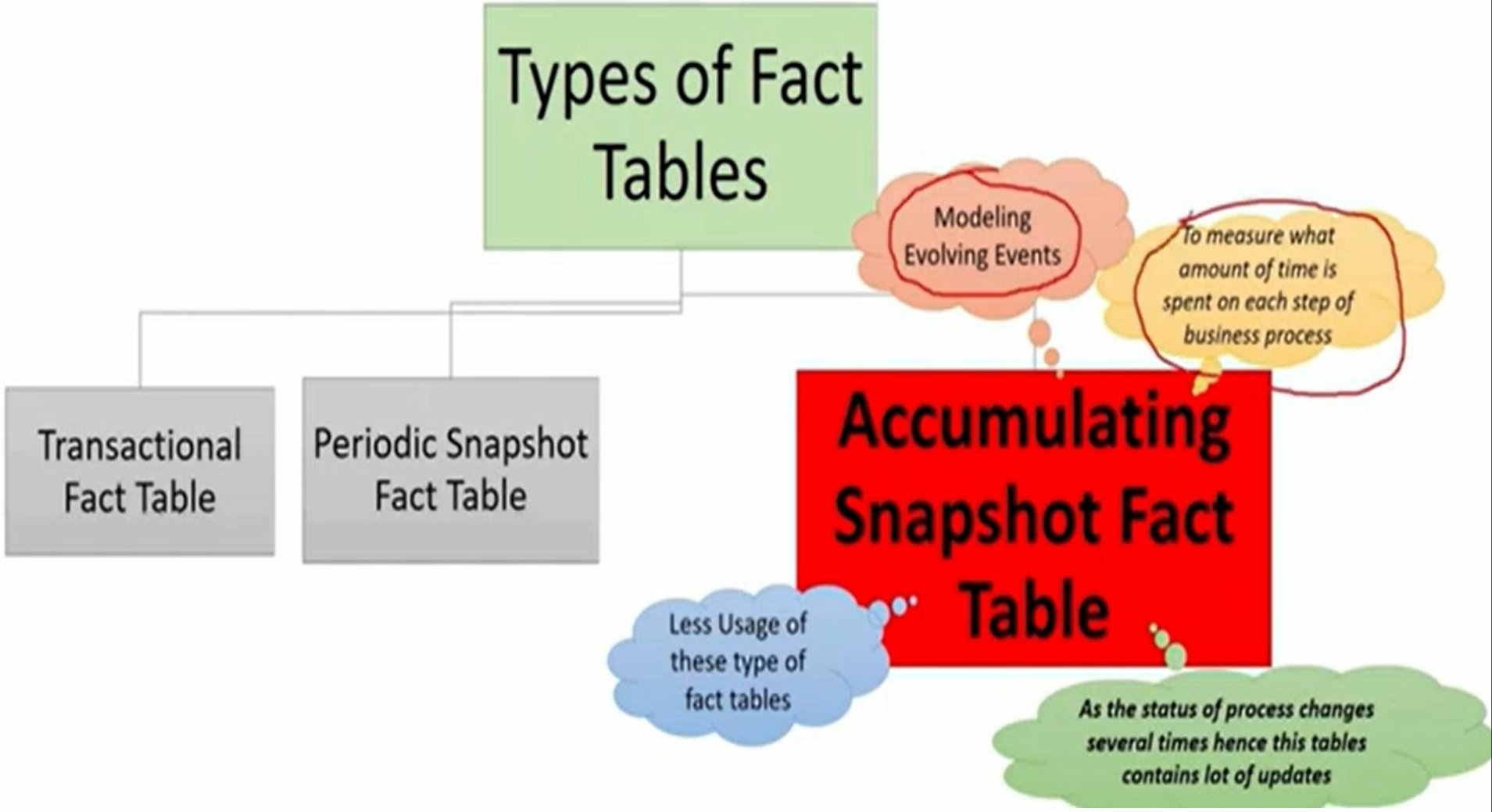
Grain of Fact table is Month

Trend Analysis using Snapshot Fact Table

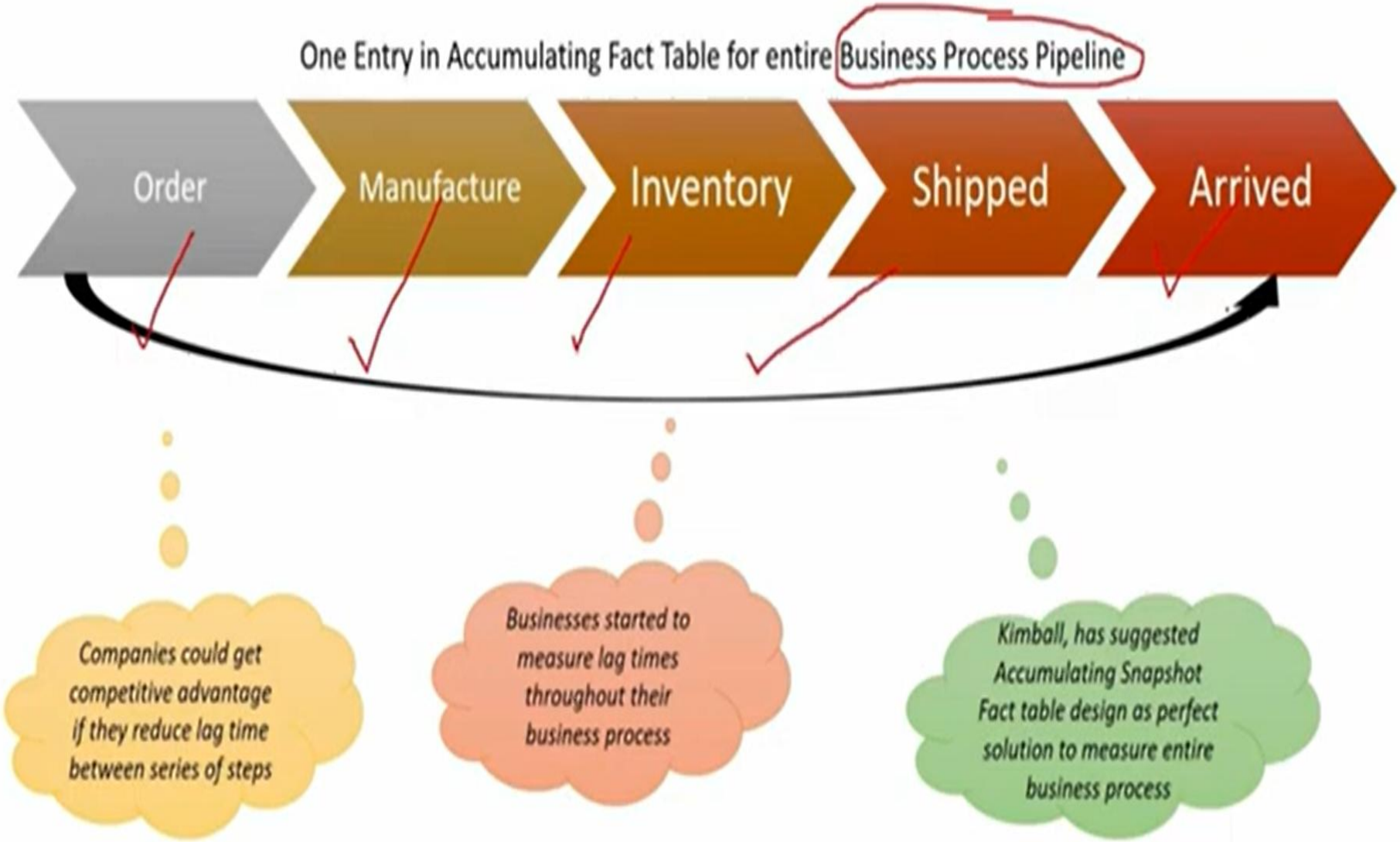


Managers will evaluate overall performance of store on MOM basis

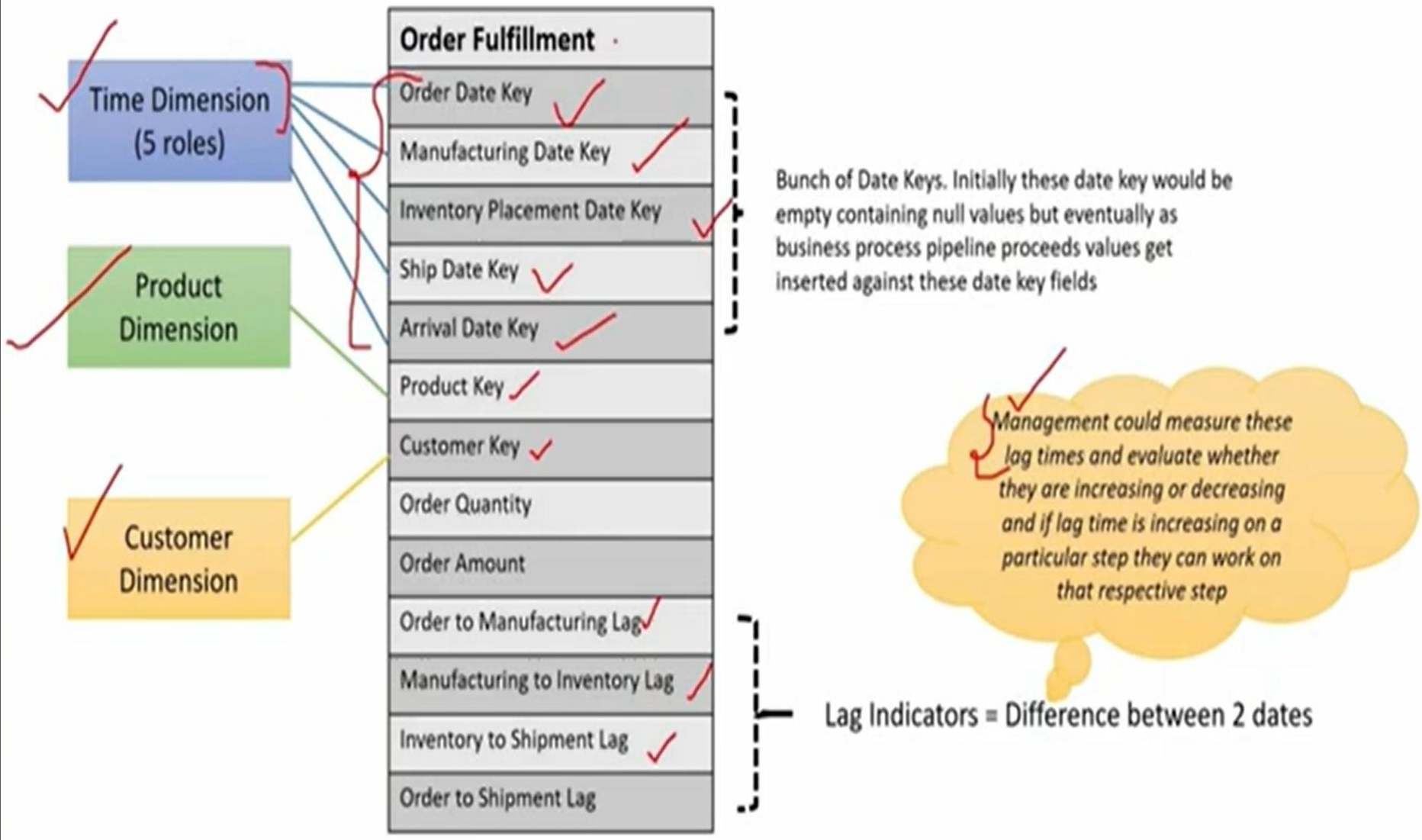
Types of Fact Tables: Accumulating Snapshot Fact Table



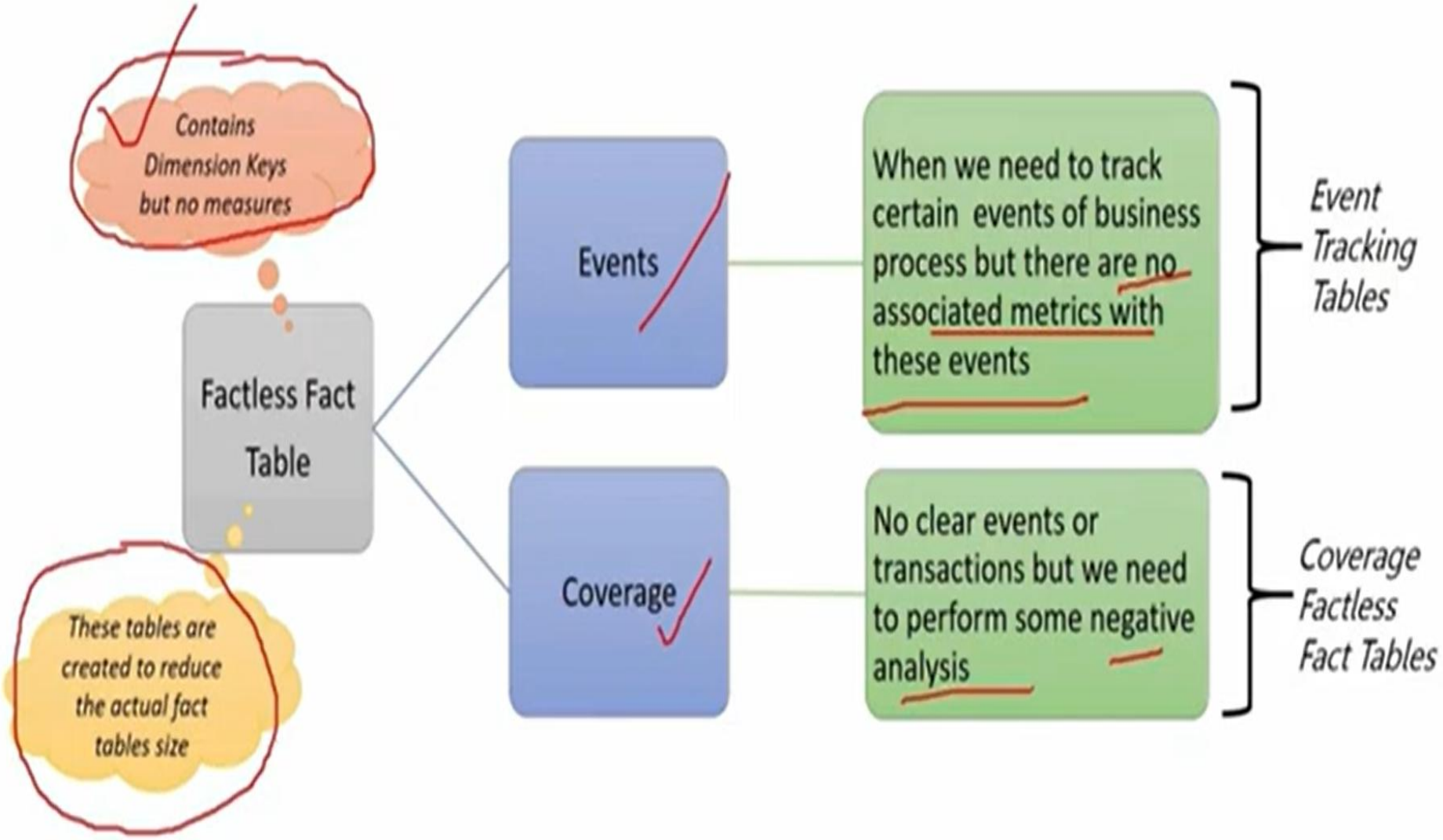
Types of Fact Tables: Accumulating Snapshot Fact Table



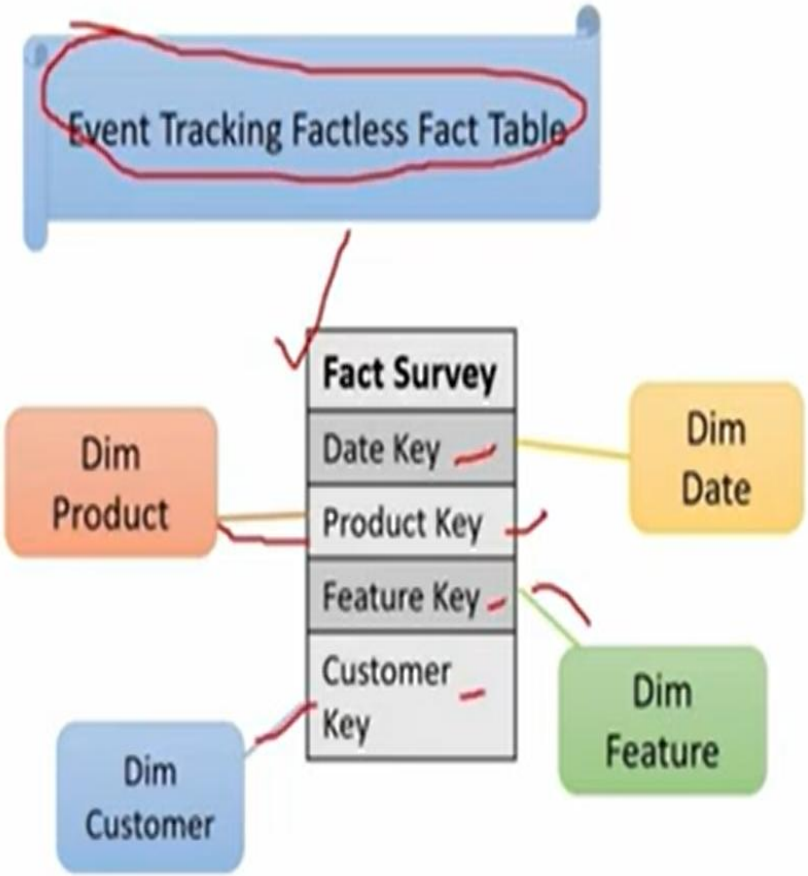
Types of Fact Tables: Accumulating Snapshot Fact Table



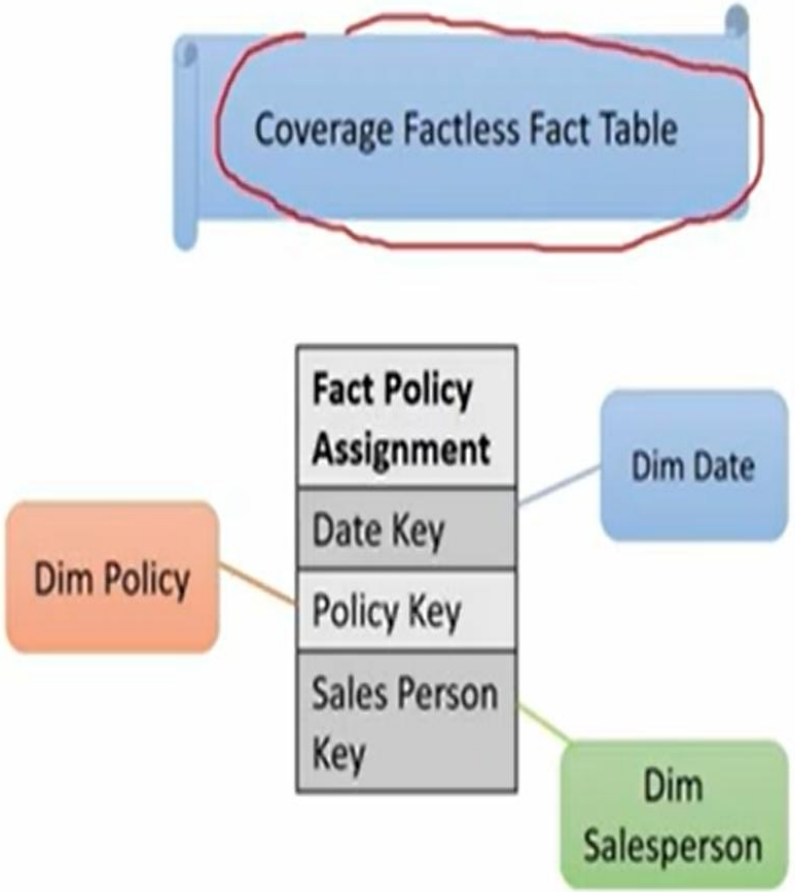
Factless Fact Tables



Factless Fact Tables



This model shows details of survey conducted by a company for its different products and their respective features



Fact Policy Assignment coverage table represents a specific time period in which Sales Person is trying to sell policy to bank existing customers. Now, we can compare this table with actual Fact Policy table to identify which sales persons are not able to sell the policies.

Strengths of the Dimensional Model

- Predictable, standard framework
- Respond well to changes in user reporting needs
- Relatively easy to add data without reloading tables
- Standard design approaches have been developed
- There exist a number of products supporting the dimensional model

Conceptual Modeling of Data Warehouses

- Star schema
- Snowflake schema
- Fact constellations

Schemas

Stars, Snowflakes and Fact Constellations: Schemas for Multidimensional Data models

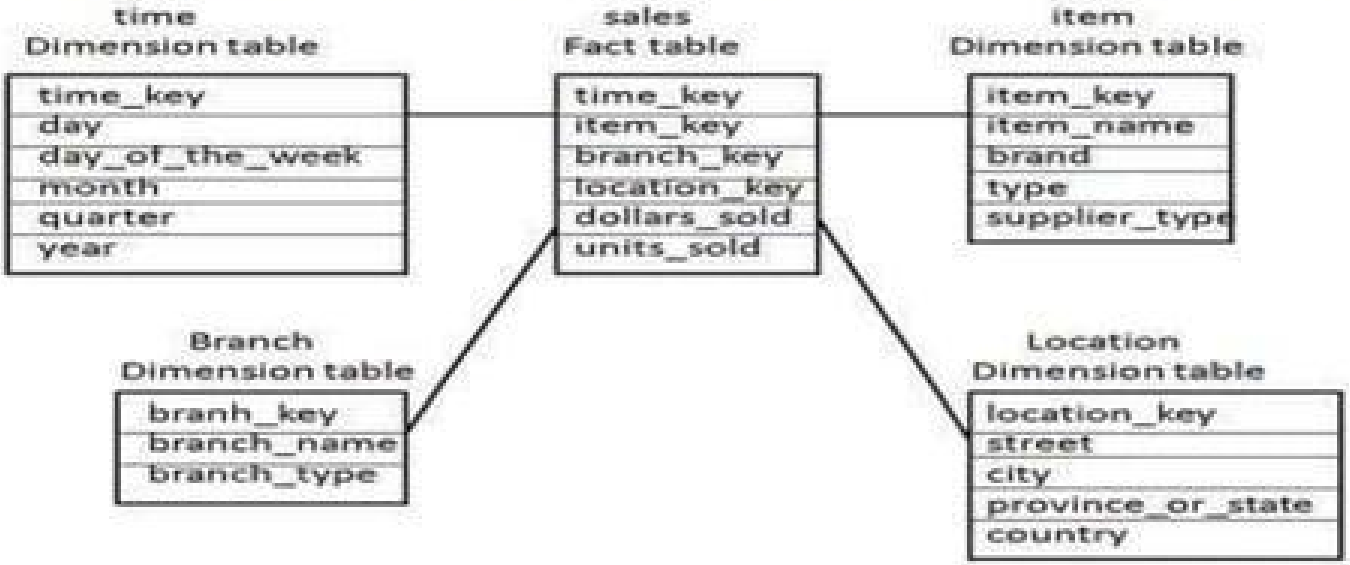
- ER data model is commonly used in design of relational databases, where ***database schema*** consists of a set of entities and the relationships between them.
- The most popular data model for a data warehouse is ***multidimensional data*** that consists of Star schema, Snowflake schema and Fact constellation schema.

a) Star Schema:

- It establishes the relationships between the ***fact table*** and any one of the dimension tables.
- It has single fact table connected to dimension tables ***like a star***.
- It has one fact table and is ***associated with numerous dimensions*** table and depicts a star.

Schemas

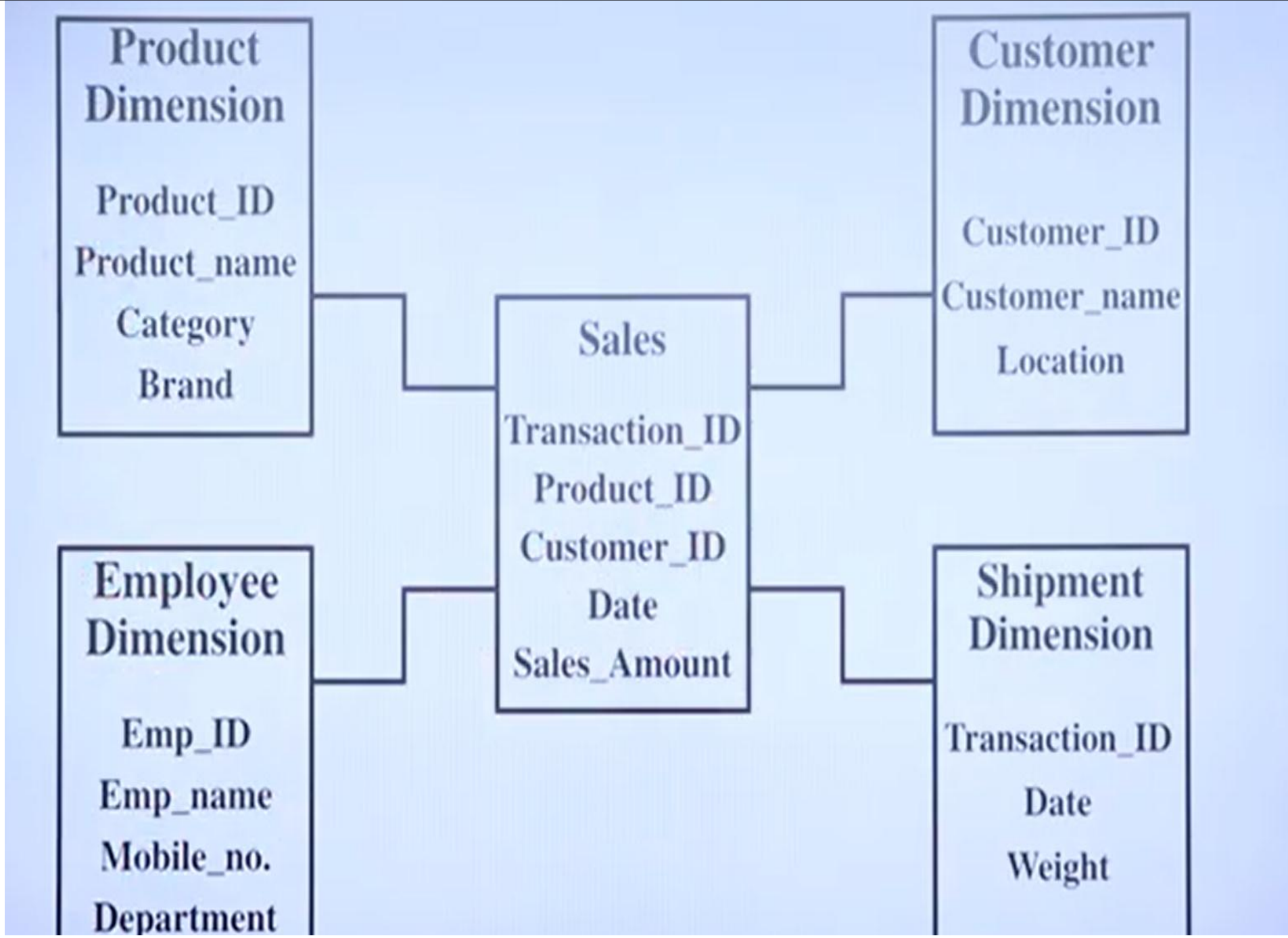
(E.g.) Star Schema diagram for sales data of a company with respect to four dimensions: time, item, branch & location.



- Each dimension represented with only ***one dimension table***.
- Dimension table contains ***set of attributes***.
- Fact table at centre & it contains key to each of ***four dimensions.***
- Fact table contains attributes like, ***dollars sold*** & units sold.

Star Schema

- A star schema is a type of data modeling technique used in data warehousing to represent data in a structured way.
- The fact table in a star schema contains the measures or metrics.
- The dimension tables in a star schema contain the descriptive attributes of the measures in the fact table.


Product Dimension
Product_ID
Product_name
Category
Brand
Customer Dimension
Customer_ID
Customer_name
Location
Sales
Transaction_ID
Product_ID
Customer_ID
Date
Sales_Amount
Employee Dimension
Emp_ID
Emp_name
Mobile_no.
Department
Shipment Dimension
Transaction_ID
Date
Weight

Transaction ID	Date	Product ID	Customer ID	Sales Amount	Quantity Sold	Profit
101	2023-07-21	P1	C1	1000	3	700
102	2023-07-21	P2	C2	1500	2	600
103	2023-07-22	P3	C1	2000	5	1500
...	...	...	...	...	...	...

Product ID	Product Name	Category	Brand
P1	Laptop	Electronics	ABC Tech
P2	Smart Watch	Electronics	XYZ Tech
...	...	...	...

Customer ID	Customer Name	Location
C1	Varun	Chandigarh
C2	Ravi	Delhi

Advantages of star Schema

The **Star Schema** is one of the most commonly used database structures in **data warehousing** due to its simplicity and efficiency. Here are some key advantages:

1. Query Performance

- Star schema enables fast **query execution** because **denormalized tables** reduce the number of joins.
- Optimized for **OLAP (Online Analytical Processing)** queries.

2. Simplicity and Ease of Use

- The **intuitive structure** (fact table at the center, dimension tables surrounding it) makes it easy to understand.
- Queries (especially in **SQL**) are easier to write and optimize.

3. Improved Read Performance

- Since the schema is **denormalized**, there is **less need for joins**, leading to **faster read operations**.
- Best suited for **analytical queries** and **reporting**.

Advantages of star Schema

4. Efficient Aggregation and Reporting

- Star schema is optimized for **aggregations** (SUM, COUNT, AVG, etc.), making it ideal for **BI (Business Intelligence)** tools.
- Suitable for **dashboarding** and **data visualization tools**.

5. Better Indexing

- Fact tables can have **bitmap indexing**, which enhances query performance for **large datasets**.

6. Scalability

- Handles **large volumes** of data efficiently.
- New **dimensions** and **facts** can be added with minimal changes.

7

Advantages of star Schema

7. **Compatibility with OLAP Systems**

- Works well with **ROLAP (Relational OLAP)** and **MOLAP (Multidimensional OLAP)** systems.
- Commonly used in **data marts** for analytical reporting.

8. **Supports Historical Data Analysis**

- Since the fact table stores transactional data, it is ideal for **trend analysis** over time.

Schemas

b) Snowflake Schema:

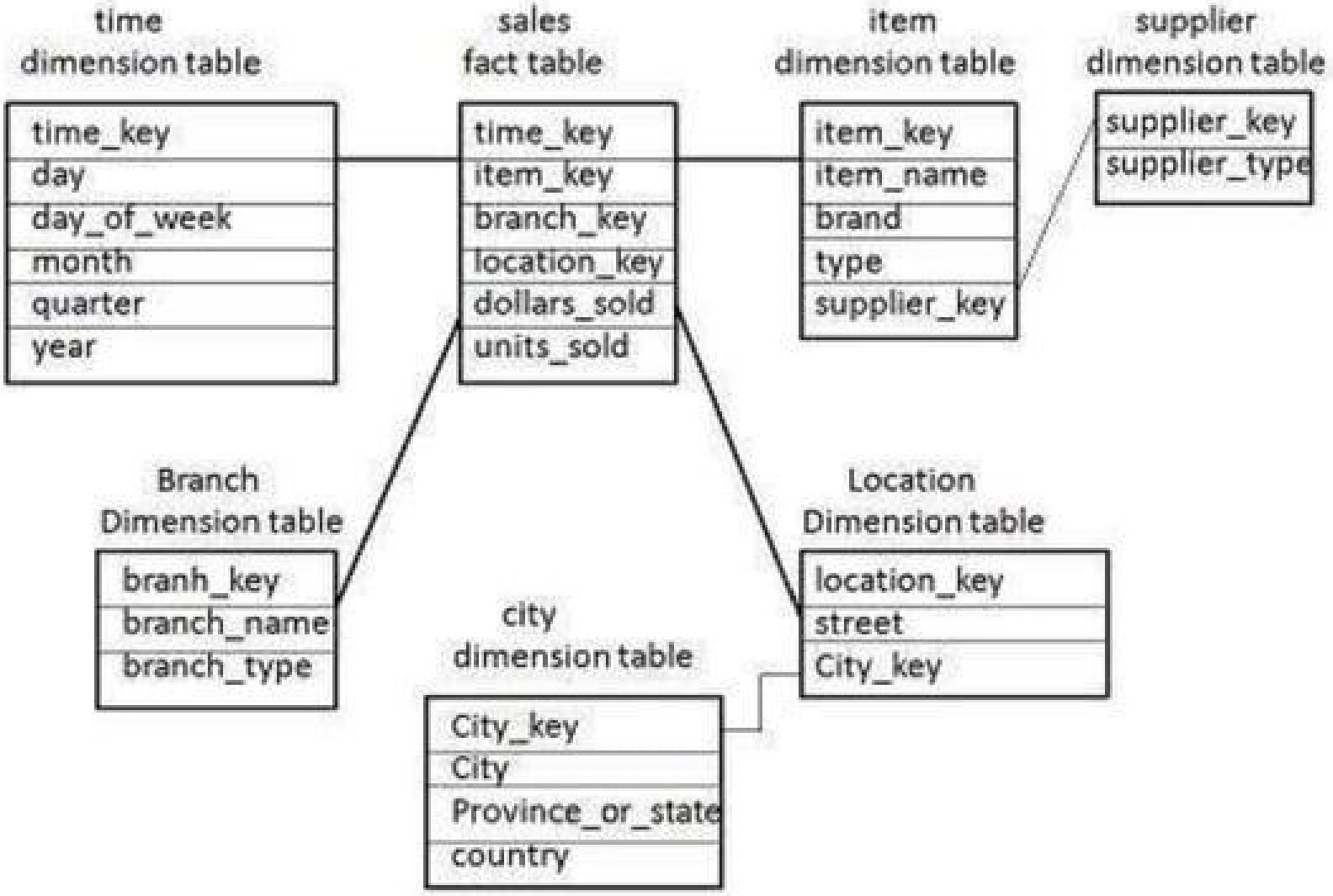
- It is the ***extension of star schema***.
- Large dimension tables are normalized into ***multiple tables.***
- The normalization split up the data into ***additional tables***.
- There is relationship between the dimensions tables & it has to do joins to fetch the data.

In diagram, item dimension table is normalized and split into two dimension tables namely, item and supplier table.

- Item dimension table contains attributes item_key, item_name, type, brand and supplier_key.
- Supplier dimension table contains the attributes supplier_key and supplier_table.

Schemas

(E.g.) Snowflake Schema diagram for sales data of a company:



Schemas

c) Fact constellation:

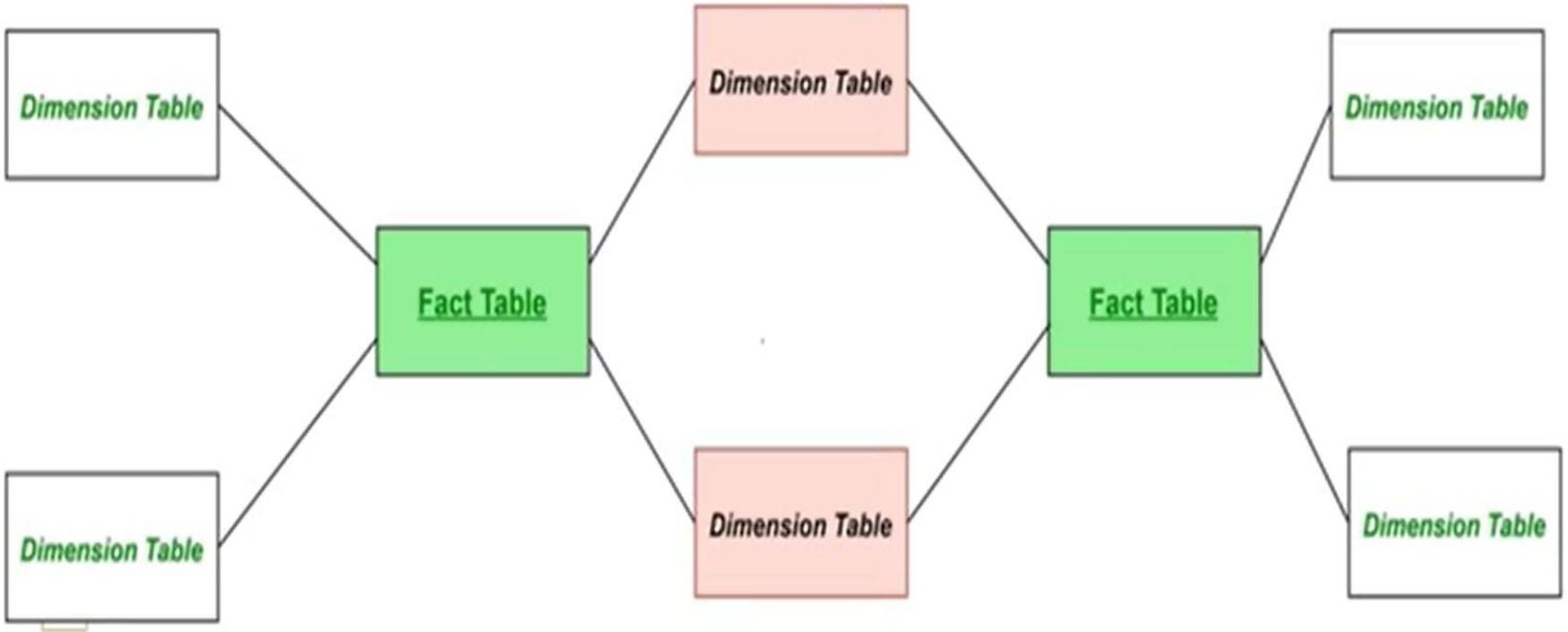
- Fact constellation is a measure of ***online analytical processing***, which is collection multiple fact tables sharing dimension tables, viewed as a collection of stars.
- It consists of ***multiple fact tables*** & this schema is also known as galaxy schema.

In following diagram, it consists of ***two fact tables***, sales and shipping.

- Sale fact table is same as star schema.
- Shipping fact table has five dimensions namely, item_key, time_key, shipper_key, location.
- It contains two measures namely dollars sold and units sold.
- It is possible for dimension table to share between fact tables.

WHAT IS FACT CONSTELLATION

- Definition: Data warehouse schema with multiple fact tables sharing dimensions.
- Also Known As: Galaxy schema.
- Structure: Multiple fact tables sharing one or more dimensions.
- Purpose: Allows for complex relationships between data.
- Challenges: More difficult to implement and maintain due to complexity.



Schemas

(E.g.) Fact Constellation Schema diagram for sales data of a company:

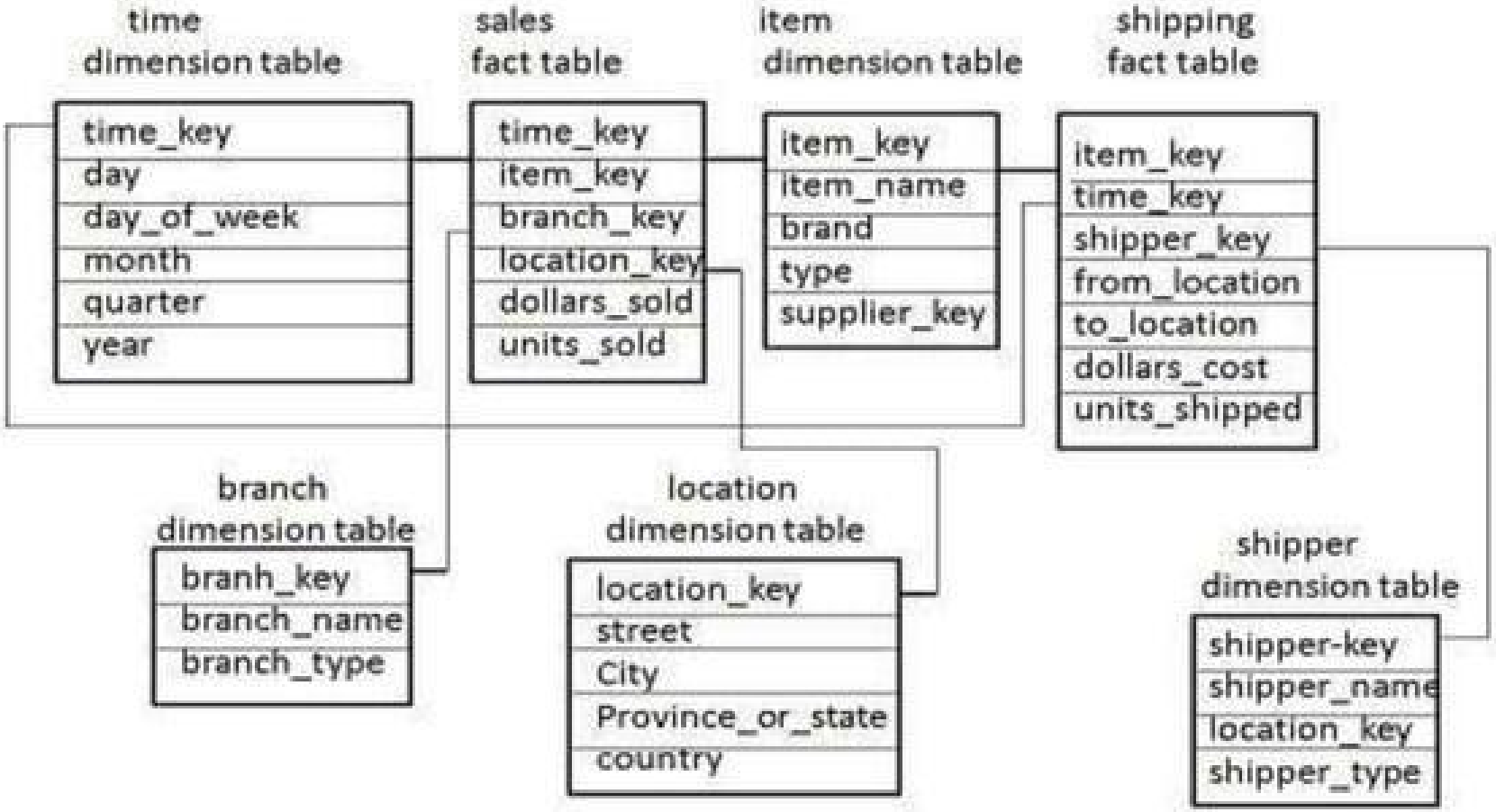


Table 13.1 Comparison among Star, Snowflake and Fact Constellation Schema

Parameter	*Star*	*Snowflake*	*Fact constellation*
Query Joins	Require simple joins	Requires complicated joins	Requires complicated joins
Data structure	De-normalized data structure	Normalized data structure.	Normalized data structure.
Number of Fact Tables	Single fact table	Single fact table	Multiple fact tables
Query Performance	It gives faster query results due to fewer join operations.	It is slow in query processing due to larger join operations.	It is slow in query processing due to larger join operations.
Dimension	Dimension table does not split into pieces.	Dimension tables are split into many pieces.	Dimension tables are split into many pieces.
Data Redundancy	Data is redundant due to de-normalization.	Data is not redundant as dimensions are normalized.	Data is not redundant as dimensions are normalized.
Data Integrity	Tough to enforce data integrity due to redundancy of data.	Easy to enforce data integrity due to no redundancy of data.	Easy to enforce data integrity due to no redundancy of data.

Difference between Star, Snowflake & Fact Constellation Schema

	Star Schema	Snowflake Schema	Fact Schema
Elements	Single Fact Table connected to multiple dimension tables with no sub-dimension tables	Single Fact Table connects to multiple dimension tables that connects to multiple sub-dimension tables	Multiple Fact Tables connects to multiple dimension tables that connects to multiple sub-dimension tables
Normalization	Denormalized	Normalized	Normalized
Number of Dimensions	Multiple dimension tables map to a single Fact Table	Multiple dimension tables map to multiple dimension tables	Multiple dimension tables map to multiple Fact Tables
Data Redundancy	High	Low	Low

Difference between Star, Snowflake & Fact Schema

	Star Schema	Snowflake Schema	Fact Schema
Performance	Fewer foreign keys resulting in increased performance	Decreased performance compared to Star Schema from higher number of foreign keys	Decreased performance compared to Star and Snowflake. Used for complex data aggregation.
Complexity	Simple, designed to be easy to understand	More complicated compared to Star Schema – can be more challenging to understand	Most complicated to understand. Reserved for highly complex data structures

Difference between Star, Snowflake & Fact Schema

	Star Schema	Snowflake Schema	Fact Schema
Storage Usage	Higher disk space due to data redundancy	Lower disk space due to limited data redundancy	Low disk space usage compared to the level of sophistication due to the limited data redundancy
Design Limitations	One Fact Table only, no sub-dimensions	One Fact Table only, multiple sub-dimensions are permitted	Multiple Fact Tables permitted, only first level dimensions are permitted

Operational Data base systems and Data warehouse

Operational Data base systems:

- Operational system assist a company or organization in its ***day-to-day business.***
- Its applications and data are highly structure and provide immediate focus on ***business functions*** with the help of OLTP.
- It required to support a ***large number of transaction*** on a daily basis.
- Operational data stores ***small, focusing the database*** on specific business area and eliminating database overhead in areas such as indexes.

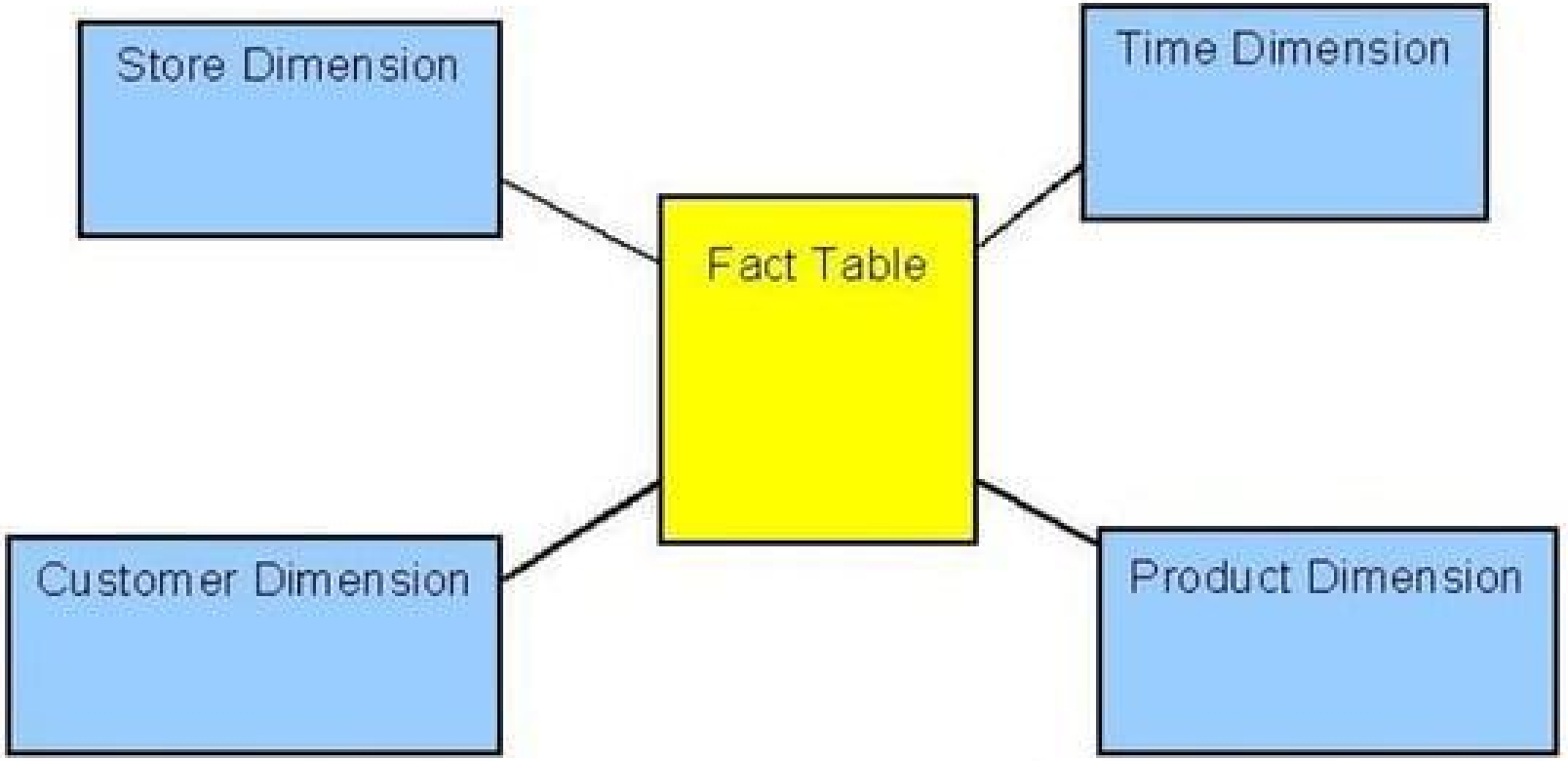
Data Warehouse system:

- This system is organized ***around the trends*** or patterns in those events set by operational systems.
- Data warehouse ***focus on business needs*** and requirements.
- It develop ideas for ***changing the business rules*** to make these events more effective.

Star schema

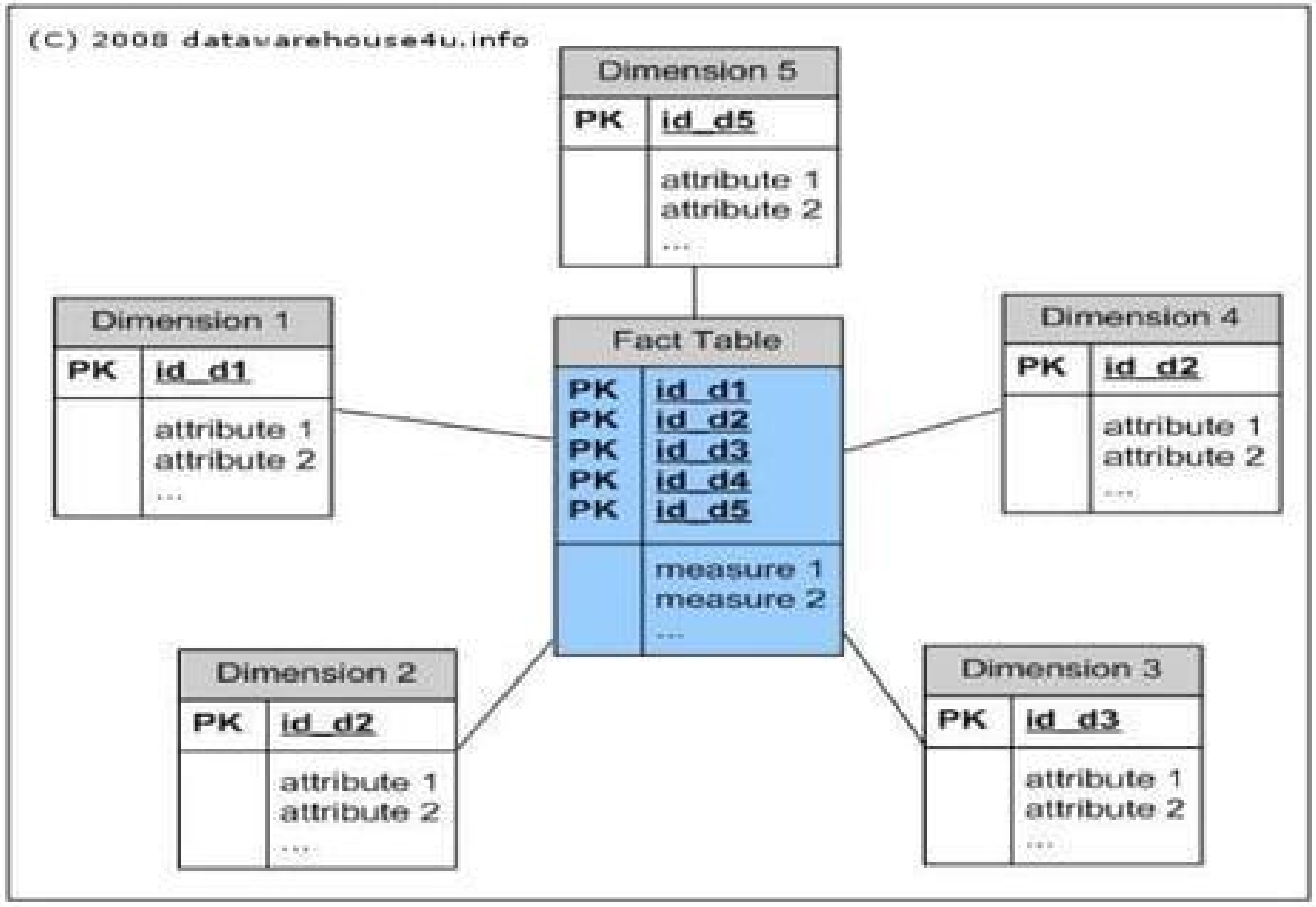
- The star schema architecture is the simplest data warehouse schema.
- It is called a star schema because the diagram resembles a star, with points radiating from a center.
- The center of the star consists of fact table and the points of the star are the dimension tables.
- Usually the fact tables in a star schema are in third normal form(3NF) whereas dimensional tables are de-normalized.

Star schema

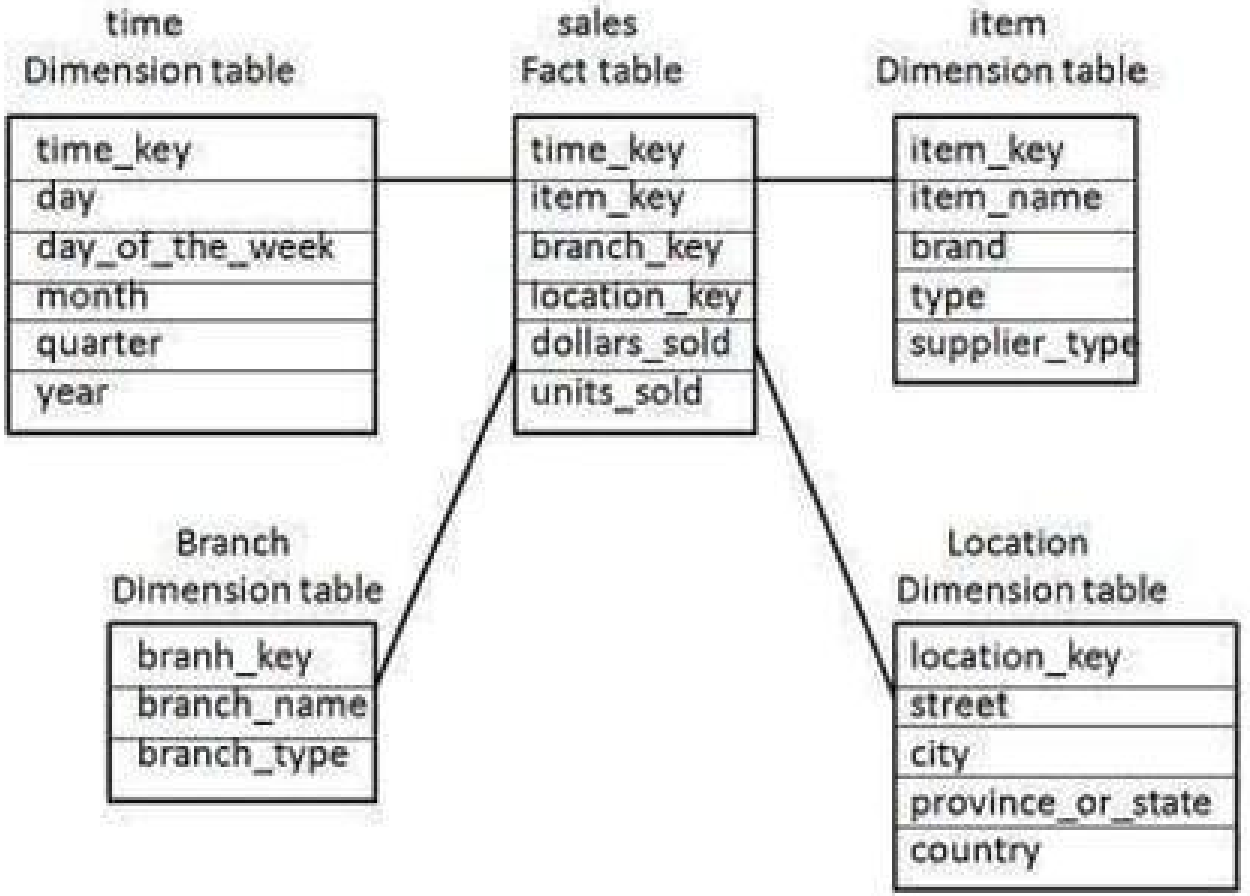


http://www.1keydata.com/datawarehousing/star-schema.html

Star schema

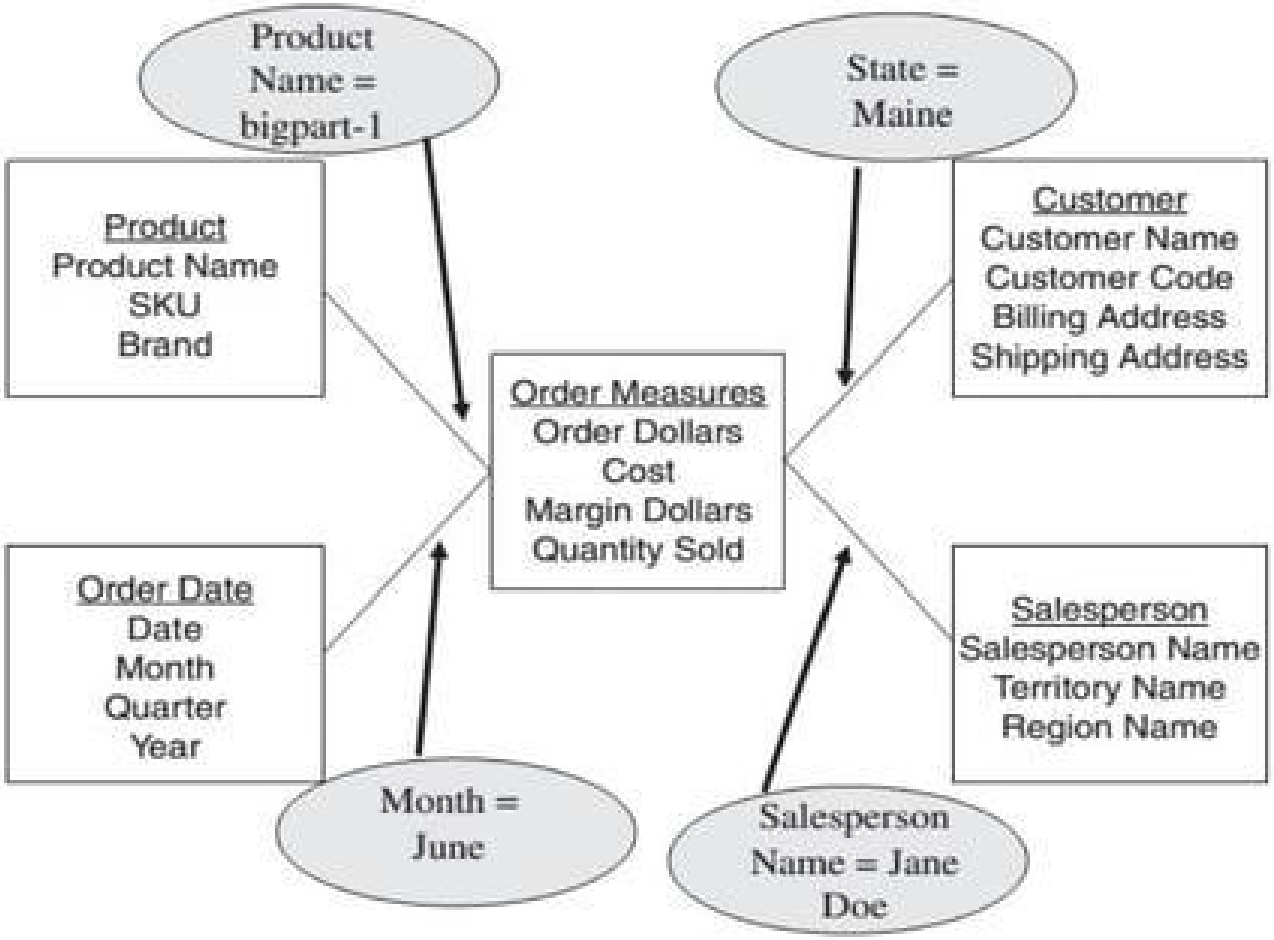

(C) 2008 datawarehouse4u.info
Dimension 5
PK
id_d5
attribute 1
attribute 2
...
Dimension 1
PK
id_d1
attribute 1
attribute 2
...
Fact Table
PK
id_d1
PK
id_d2
PK
id_d3
PK
id_d4
PK
id_d5
measure 1
measure 2
...
Dimension 4
PK
id_d2
attribute 1
attribute 2
...
Dimension 2
PK
id_d2
attribute 1
attribute 2
...
Dimension 3
PK
id_d3
attribute 1
attribute 2
...

Star schema


time
Dimension table
time_key
day
day_of_the_week
month
quarter
year
sales
Fact table
time_key
item_key
branch_key
location_key
dollars_sold
units_sold
item
Dimension table
item_key
item_name
brand
type
supplier_type
Branch
Dimension table
branh_key
branch_name
branch_type
Location
Dimension table
location_key
street
city
province_or_state
country

Querying Star Schema

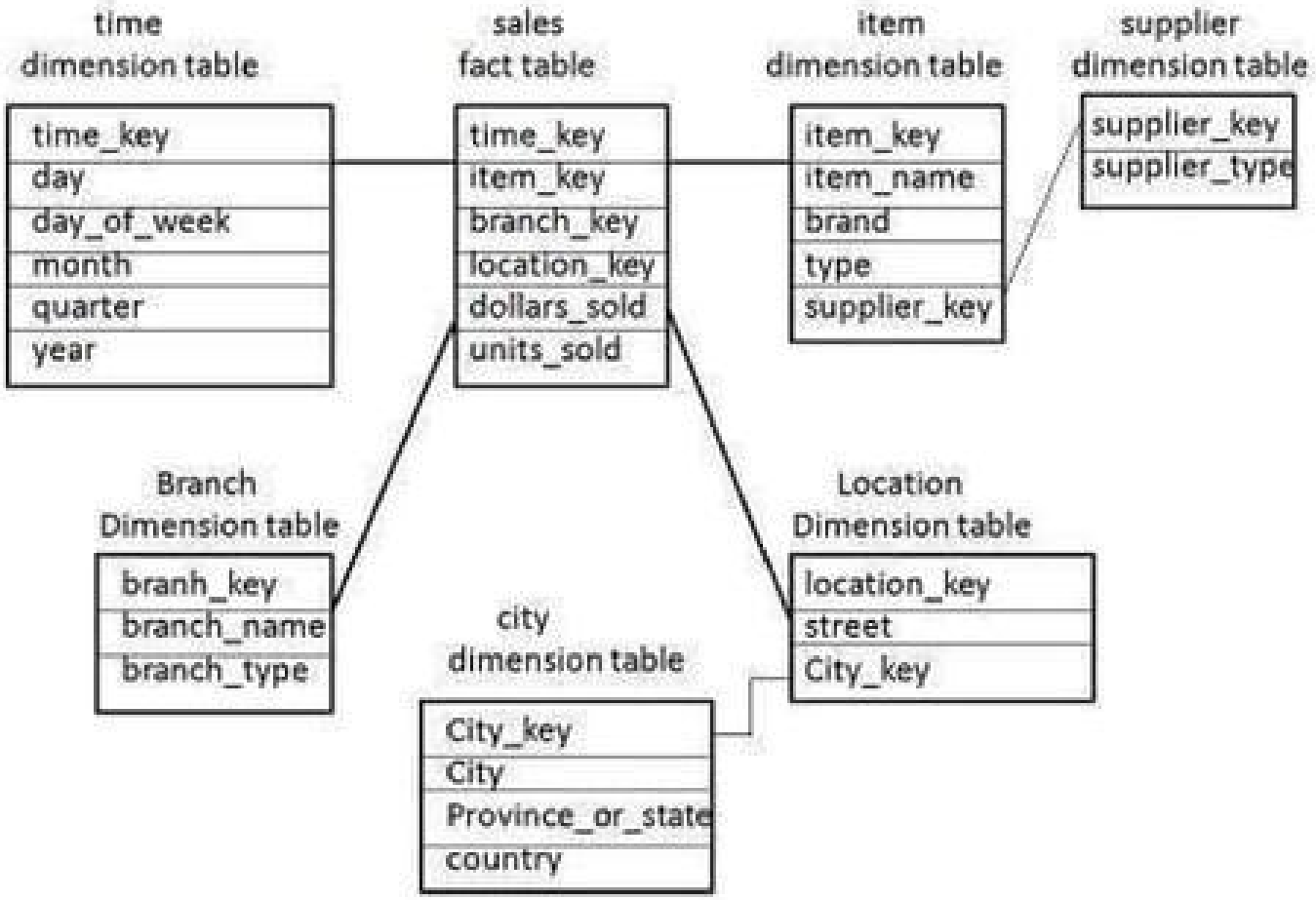
If marketing department wants the quantity sold and order dollars for productbigpart-1, relating to customers in the state ofMaine, obtained by salesperson Jane Doe, during the month of June.



Snowflake Schema

- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.

Snowflake Schema



Fact Constellation Schema

- A fact constellation has multiple fact tables. It is also known as galaxy schema.
- The following diagram shows two fact tables, namely sales and shipping.
- It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.

Handling Data Granularity in Data Warehouse

What is Data Granularity?

Data **granularity** refers to the **level of detail** in stored data. A **higher granularity** means data is stored in a more **detailed** format (e.g., per second, per transaction), whereas **lower granularity** means data is more **aggregated** (e.g., daily, monthly summaries).

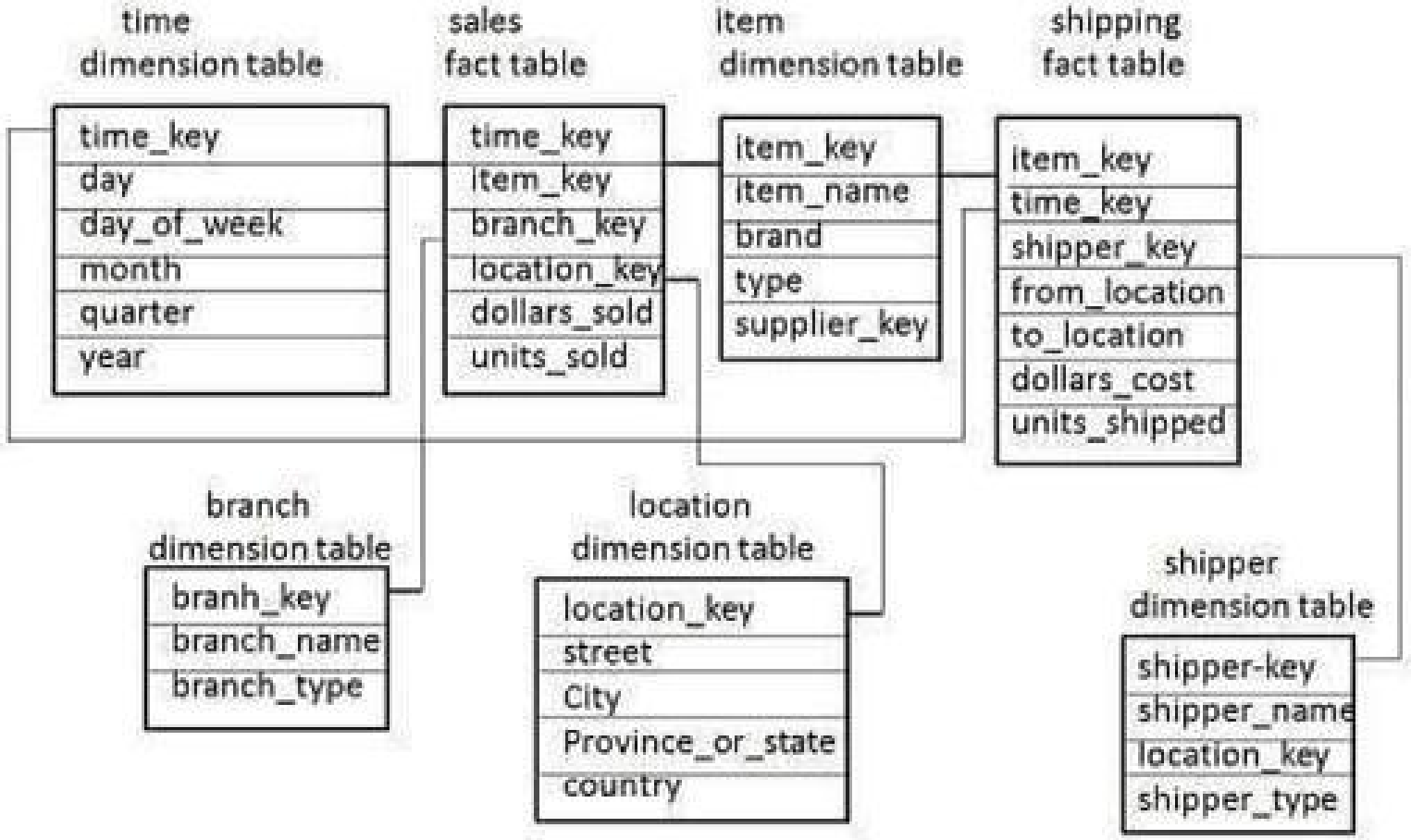
How to Handle Granularity in a Data Warehouse?

1. Choosing the Right Level of Granularity

•**Fine-grained (high granularity):** Stores data at the most detailed level, such as **individual transactions, timestamps, and specific product details**.

•**Coarse-grained (low granularity):** Stores summarized or aggregated data, such as **daily sales totals, monthly revenue, or yearly profit**.

✓ **Best Practice:** Maintain fine-grained data in the fact table while also creating aggregated summaries for better performance.


time
dimension table
time_key
day
day_of_week
month
quarter
year
sales
fact table
time_key
item_key
branch_key
location_key
dollars_sold
units_sold
item
dimension table
item_key
item_name
brand
type
supplier_key
shipping
fact table
item_key
time_key
shipper_key
from_location
to_location
dollars_cost
units_shipped
branch
dimension table
branh_key
branch_name
branch_type
location
dimension table
location_key
street
City
Province_or_state
country
shipper
dimension table
shipper-key
shipper_name
location_key
shipper_type

Business Model

As always in life, there are some disadvantages to 3NF:

- Performance can be truly awful. Most of the work that is performed on denormalizing a data model is an attempt to reach performance objectives.
- The structure can be overwhelmingly complex. We may wind up creating many small relations which the user might think of as a single relation or group of data.

The 4 Step Design Process

- Choose the Data Mart
- Declare the Grain
- Choose the Dimensions
- Choose the Facts

Building a Data Warehouse from a Normalized Database

The steps

- Develop a normalized entity-relationship business model of the data warehouse.
- Translate this into a dimensional model. This step reflects the information and analytical characteristics of the data warehouse.
- Translate this into the physical model. This reflects the changes necessary to reach the stated performance objectives.

Structural Dimensions

- The first step is the development of the structural dimensions. This step corresponds very closely to what we normally do in a relational database.
- The star architecture that we will develop here depends upon taking the central intersection entities as the fact tables and building the foreign key => primary key relations as dimensions.

Steps in dimensional modeling

- Select an associative entity for a fact table
- Determine granularity
- Replace operational keys with surrogate keys
- Promote the keys from all hierarchies to the fact table
- Add date dimension
- Split all compound attributes
- Add necessary categorical dimensions
- Fact (varies with time) / Attribute (constant)

Converting an E-R Diagram

- Determine the purpose of the mart
- Identify an association table as the central fact table
- Determine facts to be included
- Replace all keys with surrogate keys
- Promote foreign keys in related tables to the fact table
- Add time dimension
- Refine the dimension tables

Choosing the Mart

- A set of related fact and dimension tables
- Single source or multiple source
- Conformed dimensions
- Typically have a fact table for each process

Fact Tables

Represent a process or reporting environment that is of value to the organization

- It is important to determine the identity of the fact table and specify exactly what it represents.
- Typically correspond to an associative entity in the E-R model

Grain (unit of analysis)

The grain determines what each fact record represents: the level of detail.

- For example
  - Individual transactions
  - Snapshots (points in time)
  - Line items on a document
- Generally better to focus on the smallest grain

Facts

Measurements associated with fact table records at fact table granularity

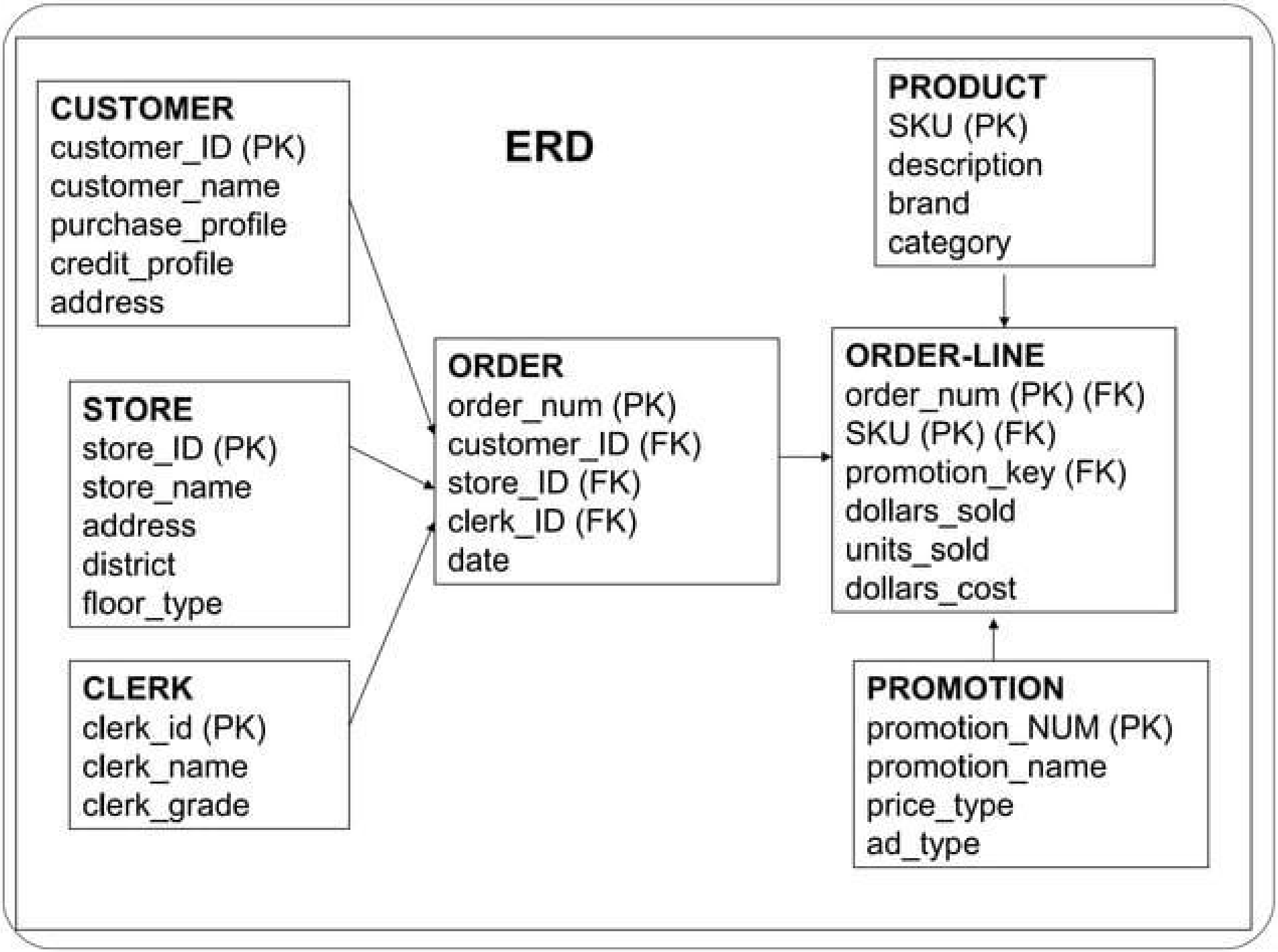
- Normally numeric and additive
- Non-key attributes in the fact table

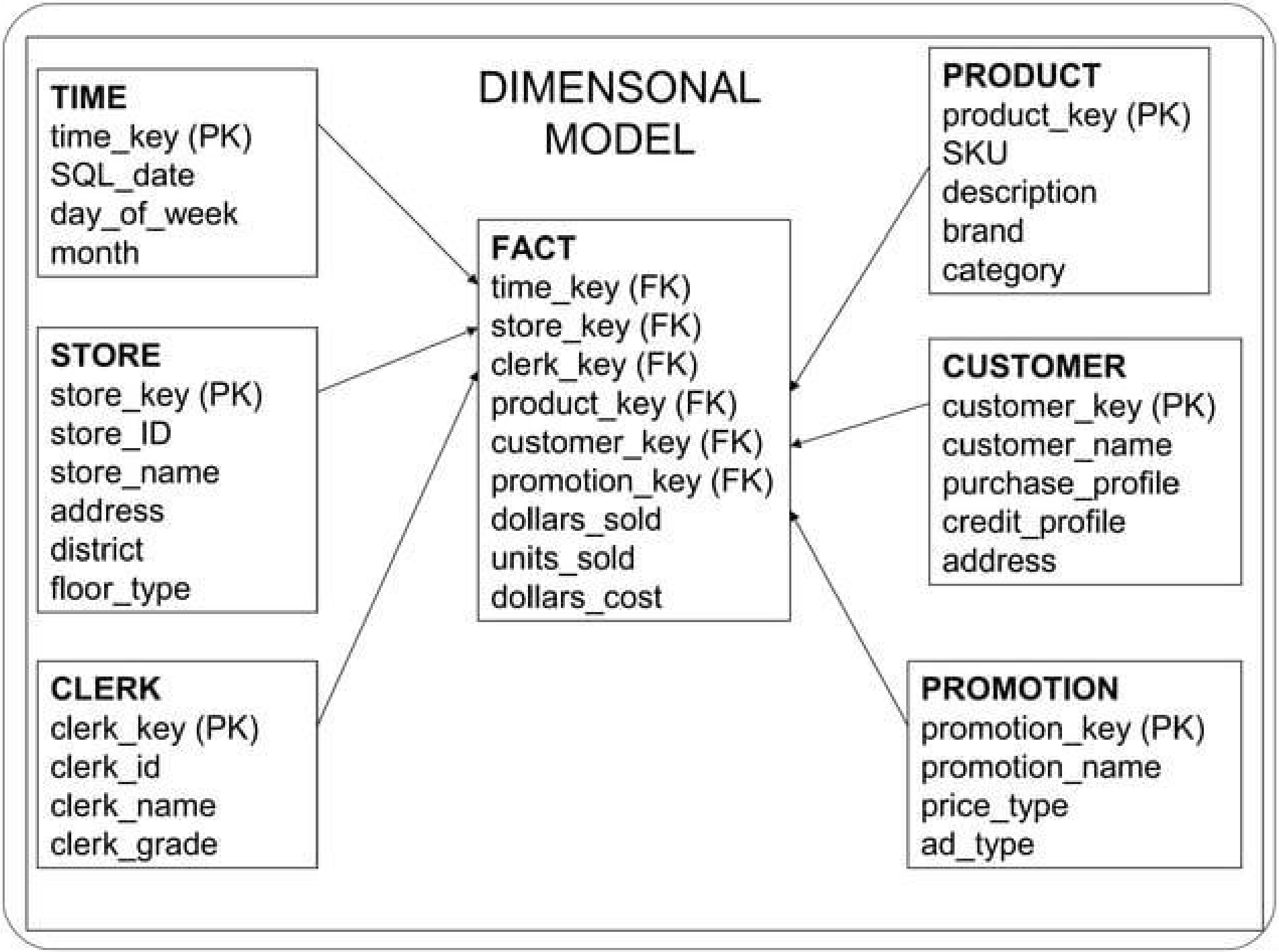
  Attributes in dimension tables are constants. Facts vary with the granularity of the fact table

Dimensions

A table (or hierarchy of tables) connected with the fact table with keys and foreign keys

- Preferably single valued for each fact record (1:m)
- Connected with surrogate (generated) keys, not operational keys
- Dimension tables contain text or numeric attributes

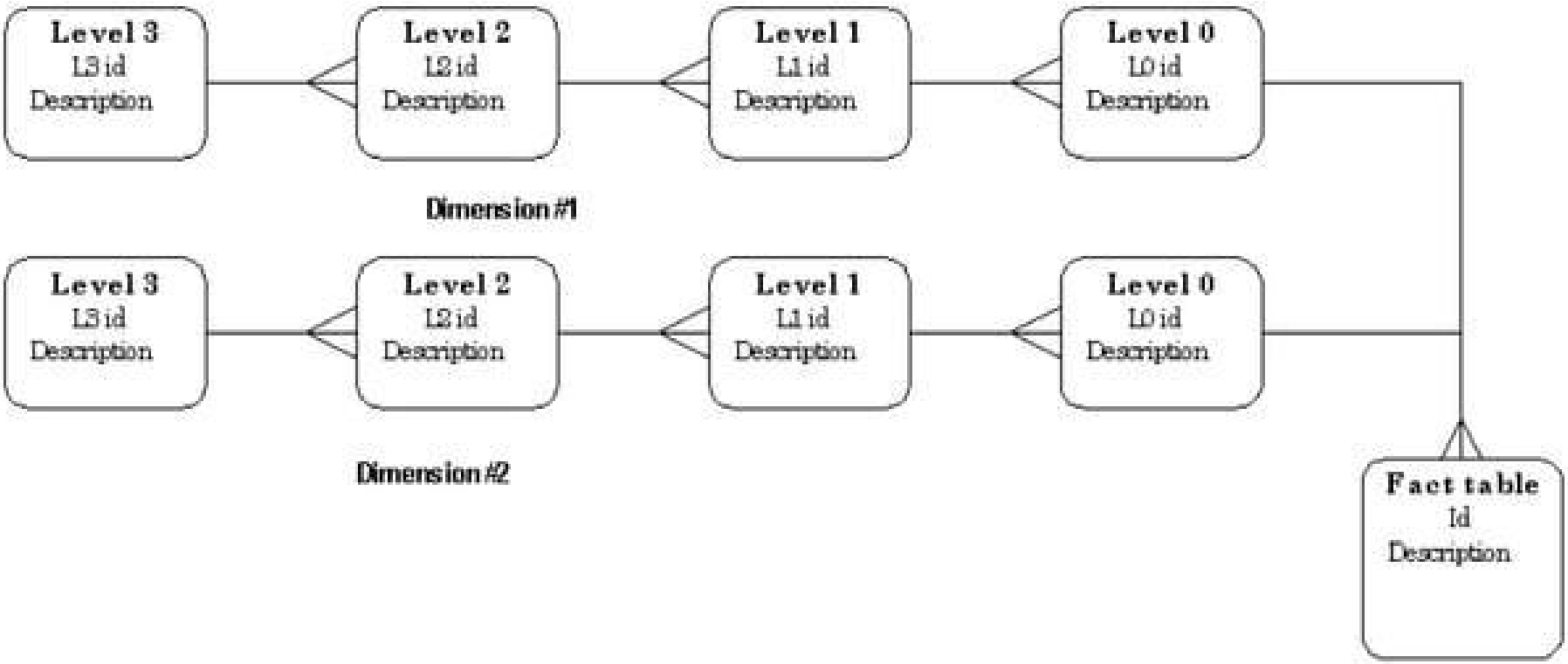

ERD
CUSTOMER
customer_ID (PK)
customer_name
purchase_profile
credit_profile
address
PRODUCT
SKU (PK)
description
brand
category
STORE
store_ID (PK)
store_name
address
district
floor_type
ORDER
order_num (PK)
customer_ID (FK)
store_ID (FK)
clerk_ID (FK)
date
ORDER-LINE
order_num (PK) (FK)
SKU (PK) (FK)
promotion_key (FK)
dollars_sold
units_sold
dollars_cost
CLERK
clerk_id (PK)
clerk_name
clerk_grade
PROMOTION
promotion_NUM (PK)
promotion_name
price_type
ad_type


DIMENSONAL MODEL
TIME
time_key (PK)
SQL_date
day_of_week
month
STORE
store_key (PK)
store_ID
store_name
address
district
floor_type
CLERK
clerk_key (PK)
clerk_id
clerk_name
clerk_grade
FACT
time_key (FK)
store_key (FK)
clerk_key (FK)
product_key (FK)
customer_key (FK)
promotion_key (FK)
dollars_sold
units_sold
dollars_cost
PRODUCT
product_key (PK)
SKU
description
brand
category
CUSTOMER
customer_key (PK)
customer_name
purchase_profile
credit_profile
address
PROMOTION
promotion_key (PK)
promotion_name
price_type
ad_type

Snowflaking & Hierarchies

- Efficiency vs Space
- Understandability
- M:N relationships

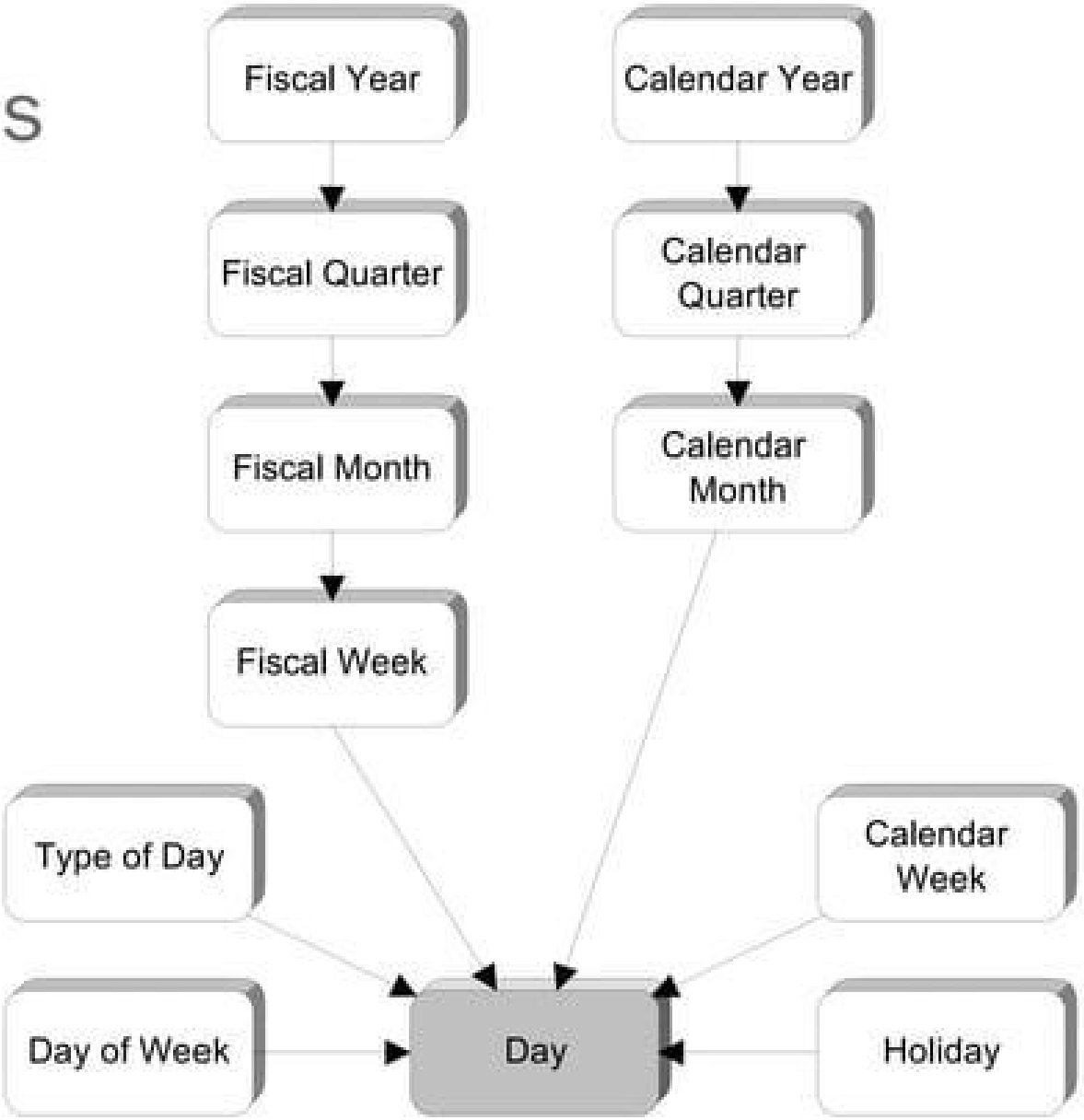
Simple DW hierarchy pattern.



Good Attributes

- Verbose
- Descriptive
- Complete
- Quality assured
- Indexed (b-tree vs bitmap)
- Equally available
- Documented

Date Dimensions



Attribute Name	Attribute Description	Sample Values
Day	The specific day that an activity took place.	06/04/1998; 06/05/1998
Day of Week	The specific name of the day.	Monday; Tuesday
Holiday	Identifies that this day is a holiday.	Easter; Thanksgiving
Type of Day	Indicates whether or not this day is a weekday or a weekend day.	Weekend; Weekday
Calendar Week	The week ending date, always a Saturday. Note that WE denotes	WE 06/06/1998; WE 06/13/1998
Calendar Month	The calendar month.	January,1998; February, 1998
Calendar Quarter	The calendar quarter.	1998Q1; 1998Q4
Calendar Year	The calendar year.	1998
Fiscal Week	The week that represents the corporate calendar. Note that the F	F Week 1 1998; F Week 46 1998
Fiscal Month	The fiscal period comprised of 4 or 5 weeks. Note that the F in the data	F January, 1998; F February, 1998
Fiscal Quarter	The grouping of 3 fiscal months.	F 1998Q1; F1998Q2
Fiscal Year	The grouping of 52 fiscal weeks / 12 fiscal months that comprise the financial year.	F 1998; F 1999

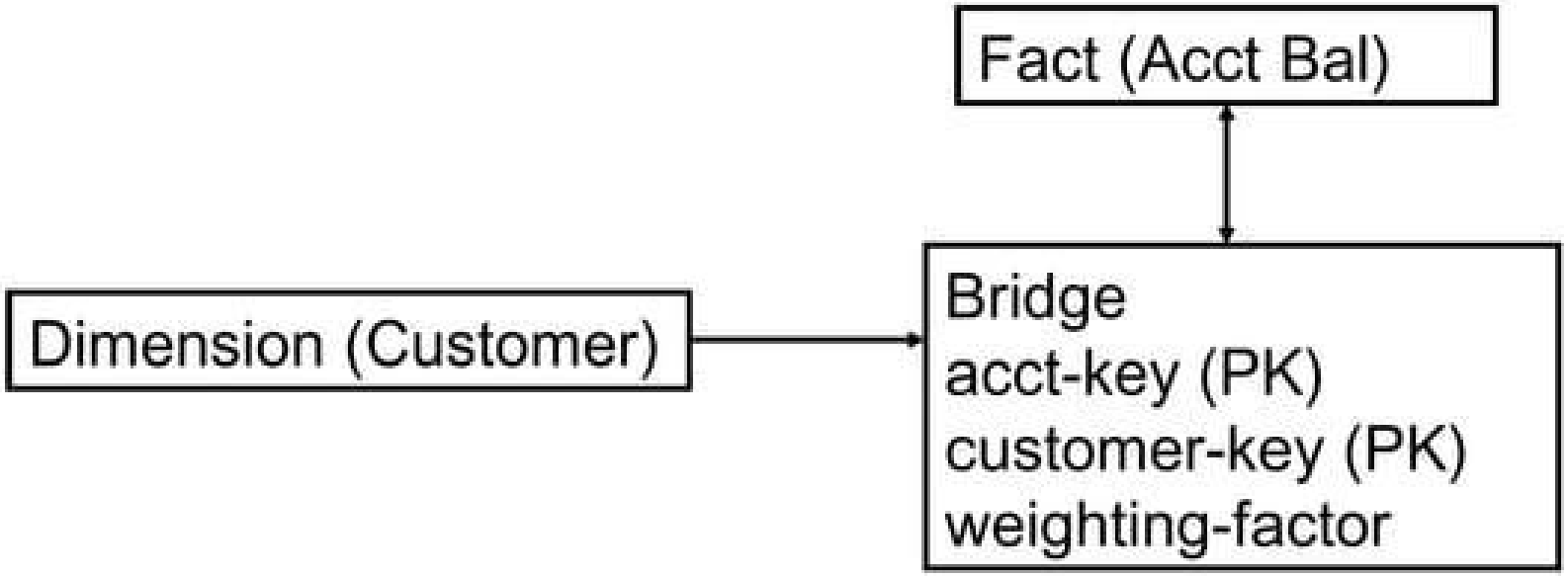
Slowly Changing Dimensions

(Addresses, Managers, etc.)

- Type 1: Store only the current value
- Type 2: Create a dimension record for each value (with or without date stamps)
- Type 3: Create an attribute in the dimension record for previous value

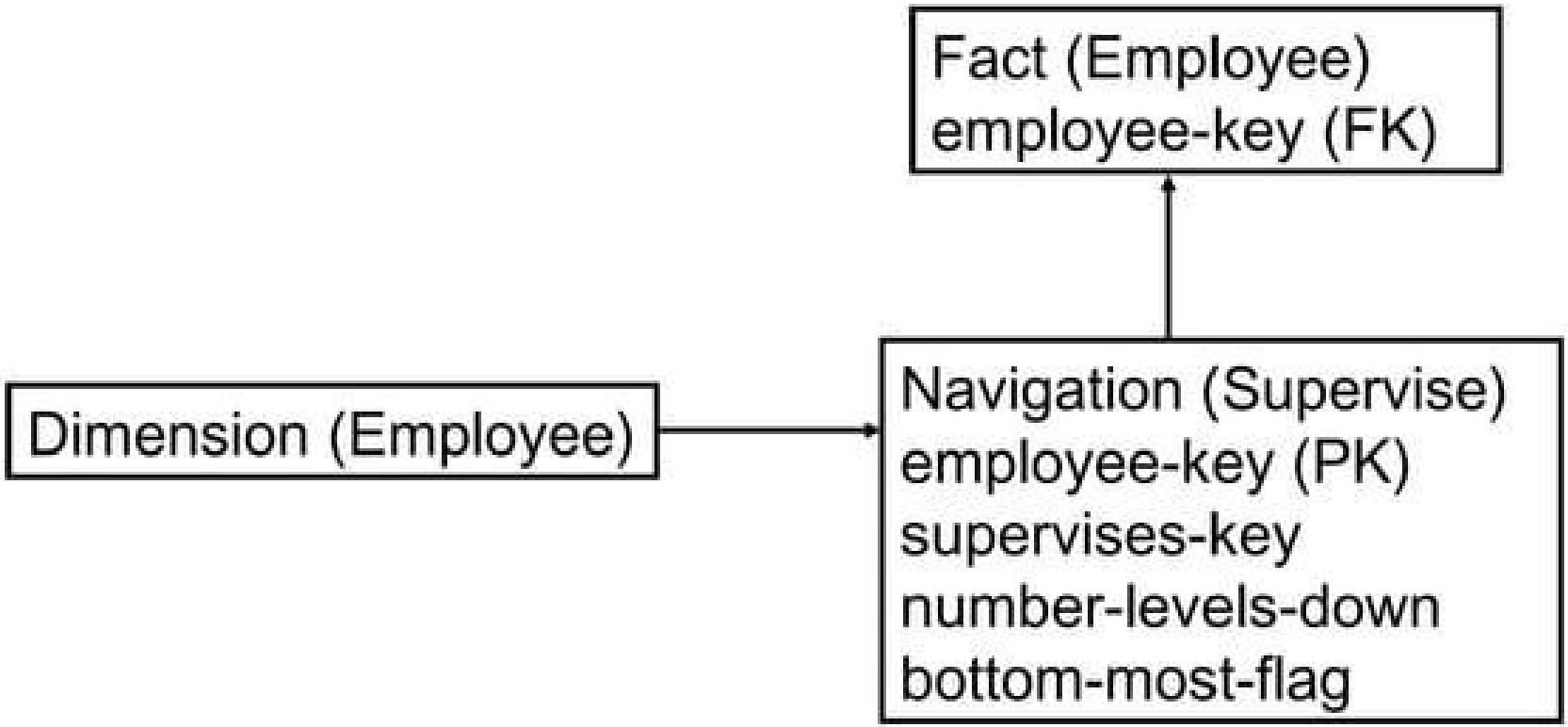
Many to many

- Use a Bridge Table
- Add a weighting factor to correct fact addition



Recursive

- Use a Bridge Table
- Add a level count and bottom flag



GRANULARITY

- The Lowest level to which the data is maintained in called Granularity
- The detailed data such as seconds, single product, one specific attribute
- The higher the granularity of a fact table, the more data in Fact.

It is related to Fact table

Understanding Granularity

- What Is Meant By The Term Granularity In A DW?
  - The level of detail available
- What Determines Granularity?
  - The level of data loaded into the fact table
    - Per order numbers
    - Daily numbers
    - Weekly numbers
  - The number and detail level of dimensions
    - Quarter, Year, etc.

Understanding Granularity

- Granularity Should Be Determined During Database Design
  - Changes can be challenging later on
  - Changes involve a few steps
    - Change structure of fact table
    - Possible changes in dimension tables
    - Changes in data loading
    - Changes in data queries

CARDINALITY

- This is all about how many distinct values are in a column
- It's more common to call it by "high" and "low" cardinality
- E.g Product can have high Cardinality for big companies and for small restaurants it is low Cardinality
- High cardinality: email ids
- Low cardinality: Isclosed

Mostly Refers to Dimension Business Keys

Multi Dimensional Data Model

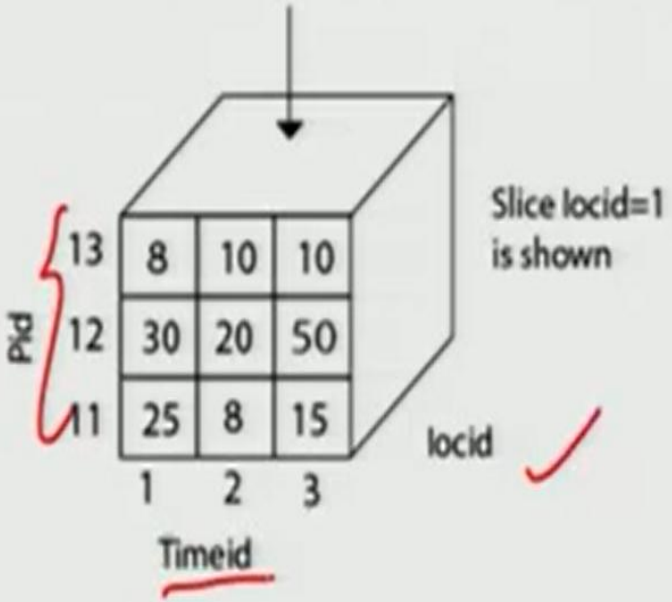
- A multidimensional model views data in the form of a data-cube. It is defined by dimensions and facts. It is a method which is used for ordering data in the database along with good arrangement and assembling of the contents in the database. This can be defined as a method for arranging the data in the database, with better structuring and organization of the contents in the database. It is typically used in the organizations for drawing out Analytical results and generation of reports, which can be used as the main source for imperative decision-making processes.

Diagrammatic Representation

Tabular representation

Pid	Timeid	locid	Sales
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35

Multidimensional representation



Disadvantages

- The multi-dimensional Data Model is slightly complicated in nature and it requires professionals to recognize and examine the data in the database.
- During the work of a Multi-Dimensional Data Model, when the system caches, there is a great effect on the working of the system.
- It is complicated in nature due to which the databases are generally dynamic in design.
- The path to achieving the end product is complicated most of the time.
- As the Multi Dimensional Data Model has complicated systems, databases have a large number of databases due to which the system is very insecure when there is a security break.

Data Cubes

- Grouping of data in a multidimensional matrix is called data cubes, data is grouped or combined in multidimensional. The data cube method has a few alternative names or a few variants, such as "Multidimensional databases," "materialized views," and "OLAP (On-Line Analytical Processing)."
- The general idea of this approach is to materialize certain expensive computations that are frequently inquired.. In Data warehousing, we generally deal with various multidimensional data models as data will be represented by multiple dimensions and multiple attributes. This multidimensional data is represented in the data cube as the cube represents a high-dimensional space. The Data cube pictorially shows how different attributes of data are arranged in the data mode

Sr. No.	Category	OLAP (Online analytical processing)	OLTP (Online transaction processing)
5.	Normalized	In an OLAP database, tables are not normalized.	In an OLTP database, tables are normalized (3NF).
6.	Usage of data	The data is used in planning, problem-solving, and decision-making.	The data is used to perform day-to-day fundamental operations.
4.	Application	It is subject-oriented. Used for Data Mining, Analytics, Decisions making, etc.	It is application-oriented. Used for business tasks.

OLAP Operations

1.Roll-Up:

- The roll-up operation (**also known as drill-up or aggregation operation**) performs aggregation on a data cube, by climbing down concept hierarchies, i.e., dimension reduction. Roll-up is like **zooming-out** on the data cubes.
- Figure shows the result of roll-up operations performed on the dimension location. The hierarchy for the location is defined as the Order Street, city, province, or state, country. The roll-up operation aggregates the data by ascending the location hierarchy from the level of the city to the level of the country.

- When a roll-up is performed by dimensions reduction, one or more dimensions are removed from the cube. For example, consider a sales data cube having two dimensions, location and time. Roll-up may be performed by removing, the time dimensions, appearing in an aggregation of the total sales by location, relatively than by location and by time.

Example

Consider the following cubes illustrating temperature of certain days recorded weekly:

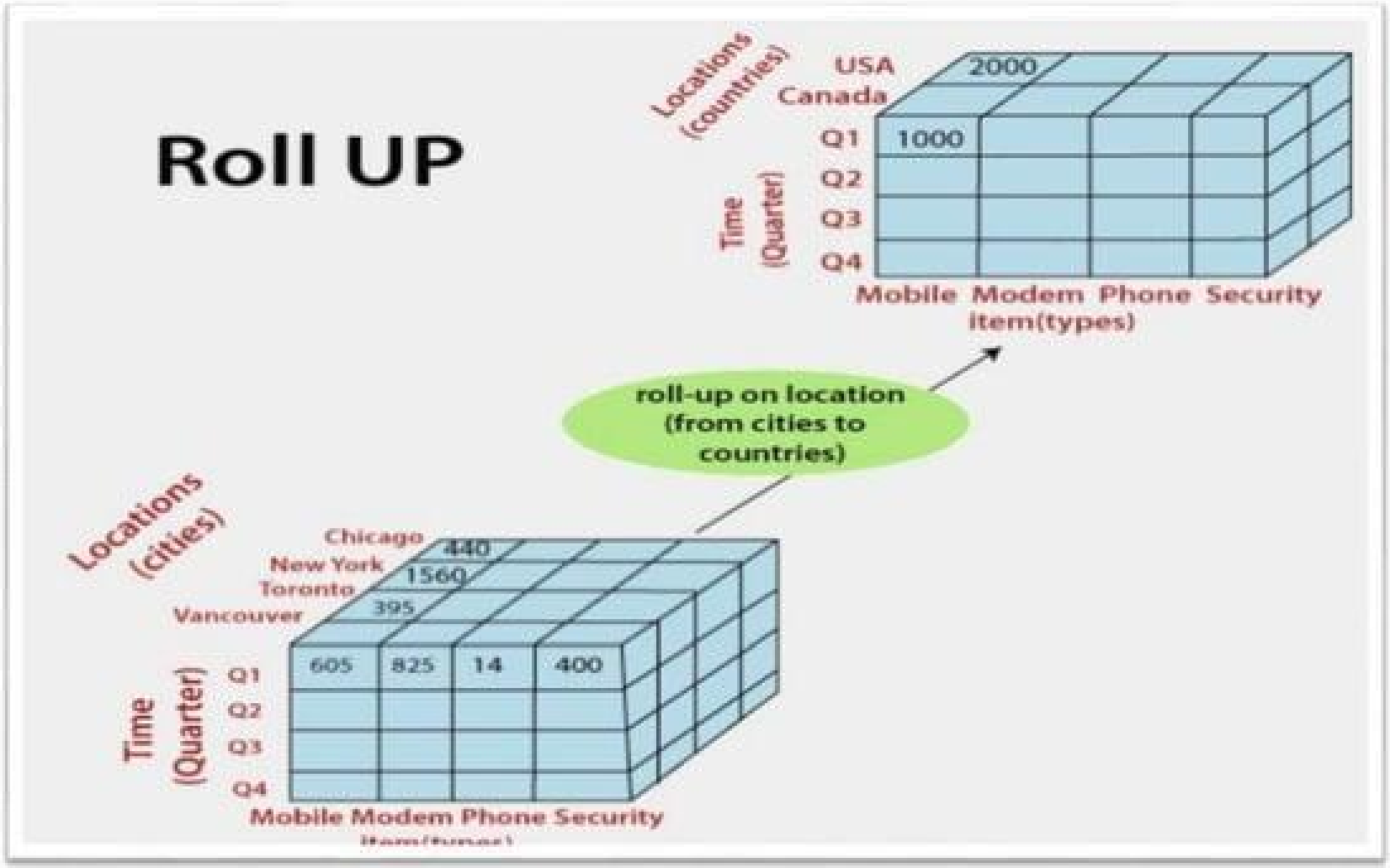
Temperature	64	65	68	69	70	71	72	75	80	81	83	85
Week1	1	0	1	0	1	0	0	0	0	0	1	0
Week2	0	0	0	1	0	0	1	2	0	1	0	0

Consider that we want to set up levels (hot (80-85), mild (70-75), cool (64-69)) in temperature from the above cubes.
To do this, we have to group column and add up the value according to the concept hierarchies. This operation is known as a roll-up.

By doing this, we contain the following cube:

Temperature	cool	mild	hot
Week1	2	1	1
Week2	2	1	1

The roll-up operation groups the information by levels of temperature.
The following diagram illustrates how roll-up works.


Roll UP
Locations (countries)
USA
Canada
2000
Q1
1000
Q2
Q3
Q4
Time (Quarter)
Mobile Modem Phone Security
item(types)
roll-up on location (from cities to countries)
Locations (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
605
825
14
400
Q1
Q2
Q3
Q4
Time (Quarter)
Mobile Modem Phone Security

2.Drill-Down

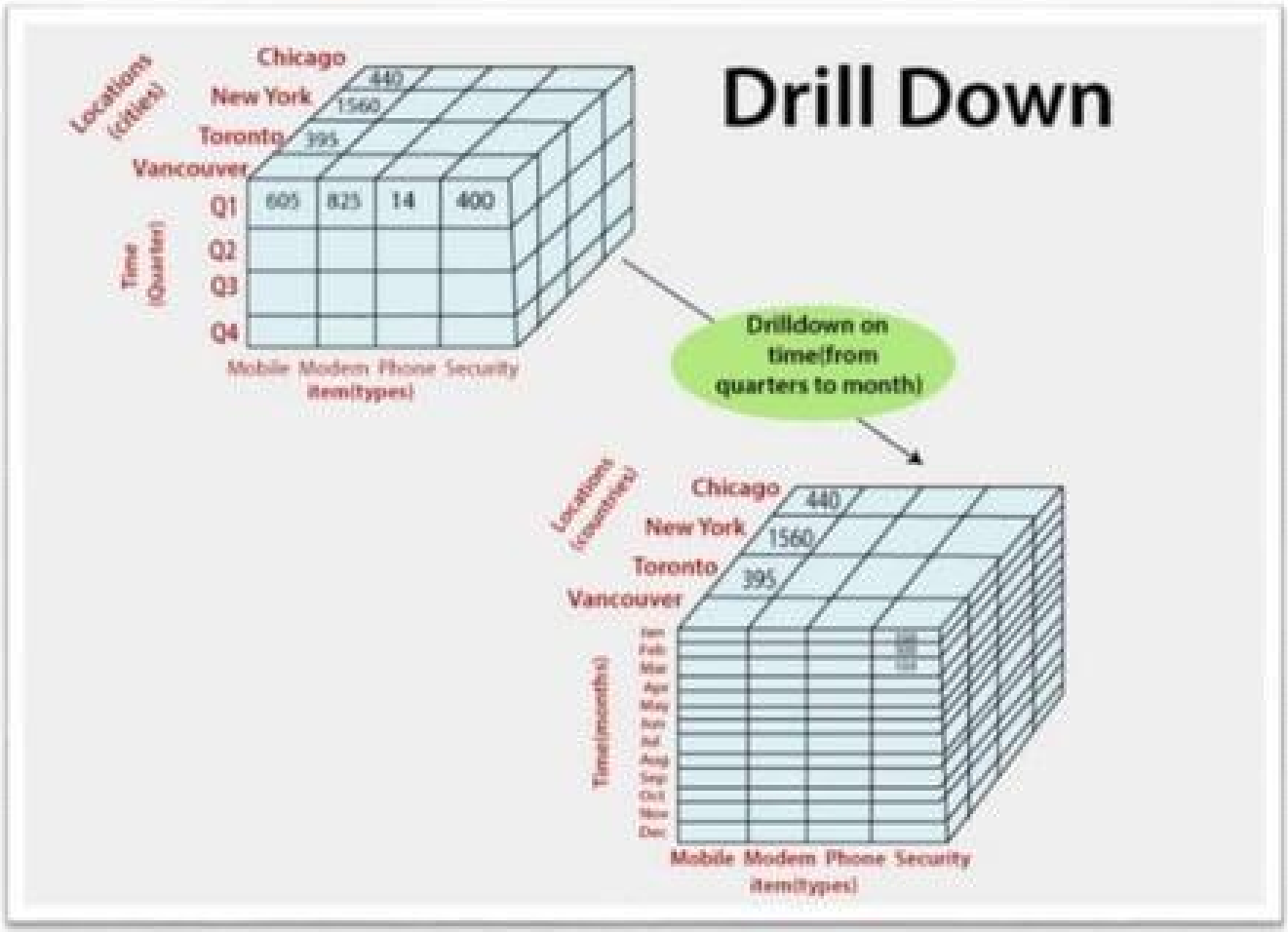
- The drill-down operation (**also called roll-down**) is the reverse operation of **roll-up**. Drill-down is like **zooming-in** on the data cube. It navigates from less detailed record to more detailed data. Drill-down can be performed by either **stepping down** a concept hierarchy for a dimension or adding additional dimensions.
- Figure shows a drill-down operation performed on the dimension time by stepping down a concept hierarchy which is defined as day, month, quarter, and year. Drill-down appears by descending the time hierarchy from the level of the quarter to a more detailed level of the month.

- Because a drill-down adds more details to the given data, it can also be performed by adding a new dimension to a cube. For example, a drill-down on the central cubes of the figure can occur by introducing an additional dimension, such as a customer group.

Example

Drill-down adds more details to the given data

Temperature	cool	mild	hot
Day 1	0	0	0
Day 2	0	0	0
Day 3	0	0	1
Day 4	0	1	0
Day 5	1	0	0
Day 6	0	0	0
Day 7	1	0	0
Day 8	0	0	0
Day 9	1	0	0
Day 10	0	1	0
Day 11	0	1	0
Day 12	0	1	0
Day 13	0	0	1
Day 14	0	0	0

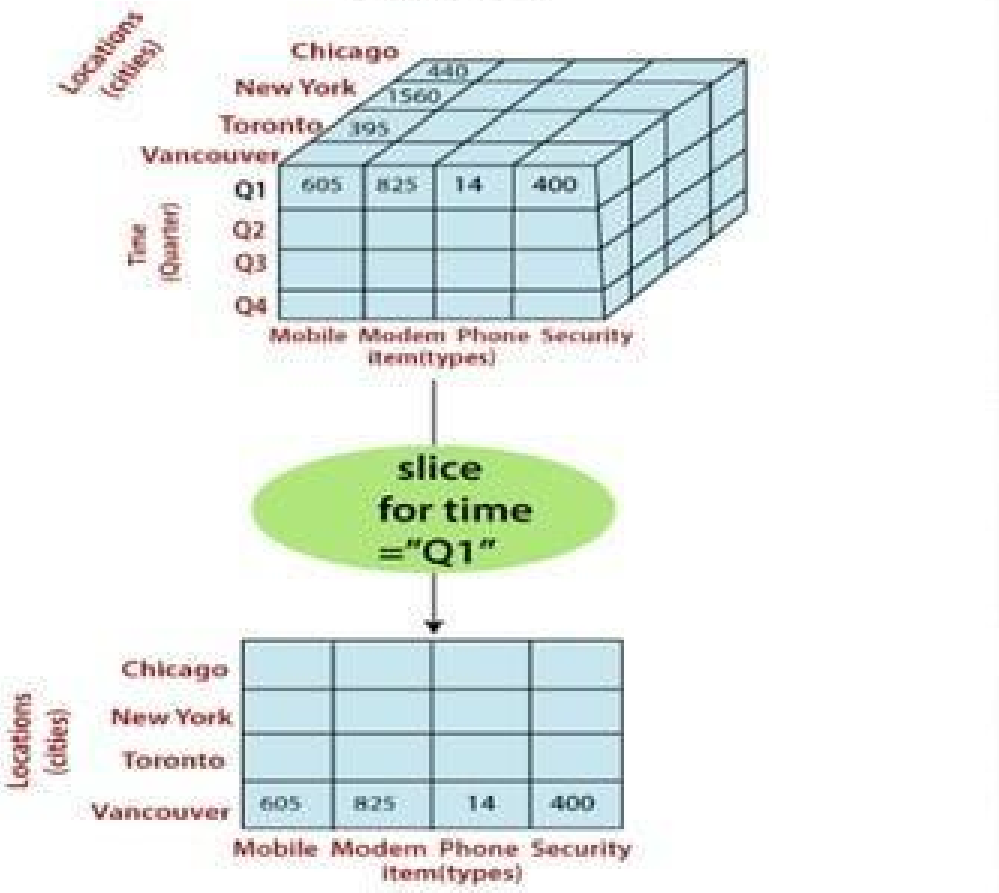

Drill Down
Locations (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
Q1
Q2
Q3
Q4
605
825
14
400
Time (Quarter)
Mobile Modem Phone Security
item(types)
Drilldown on time(from quarters to month)
Locations (countries)
Chicago
New York
Toronto
Vancouver
440
1560
395
Time(months)
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Mobile Modem Phone Security
item(types)

Slice

- A **slice** is a subset of the cubes corresponding to a single value for one or more members of the dimension. For example, a slice operation is executed when the customer wants a selection on one dimension of a three-dimensional cube resulting in a two-dimensional site. So, the Slice operations perform a selection on one dimension of the given cube, thus resulting in a subcube.
- For example, if we make the selection, temperature=cool we will obtain the following cube:

Temperature	cool
Day 1	0
Day 2	0
Day 3	0
Day 4	0
Day 5	1
Day 6	1
Day 7	1
Day 8	1
Day 9	1
Day 11	0
Day 12	0
Day 13	0
Day 14	0

Slice



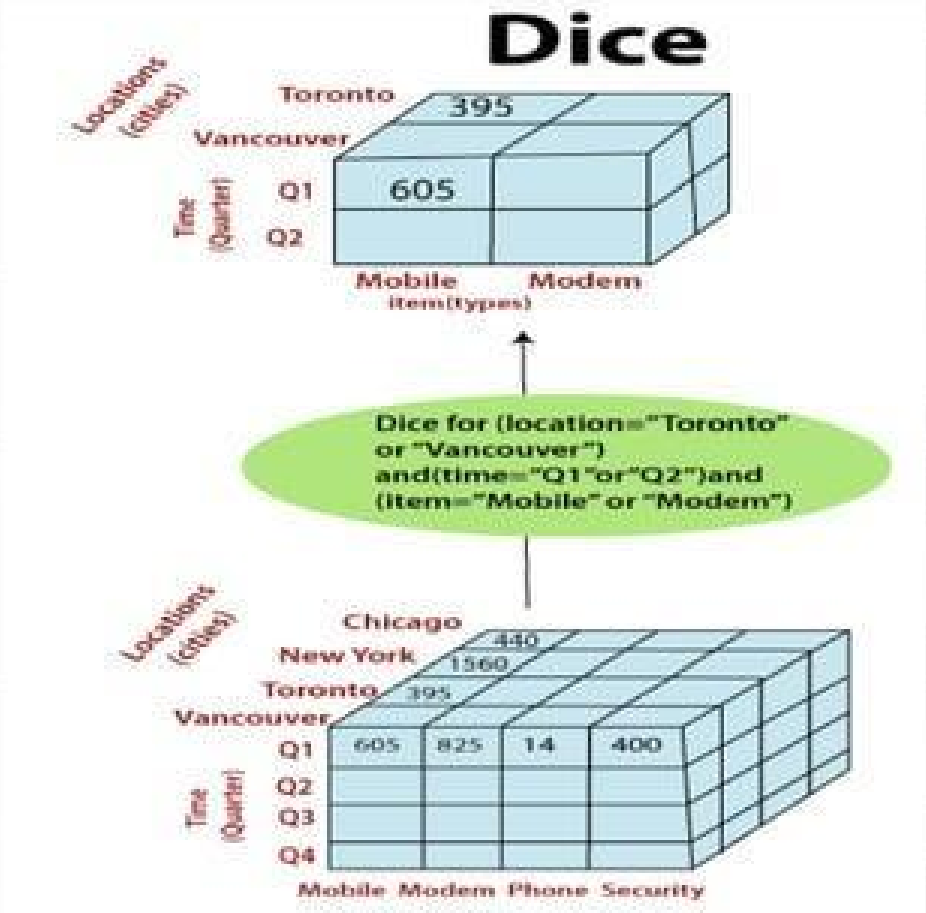
Here Slice is functioning for the dimensions "time" using the criterion time = "Q1".
It will form a new sub-cubes by selecting one or more dimensions.

Dice

- The dice operation describes a subcube by operating a selection on two or more dimension.
- **For example**, Implement the selection (time = day 3 OR time = day 4) AND (temperature = cool OR temperature = hot) to the original cubes we get the following subcube (still two-dimensional)

Temperature	cool	hot
Day 3	0	1
Day 4	0	0

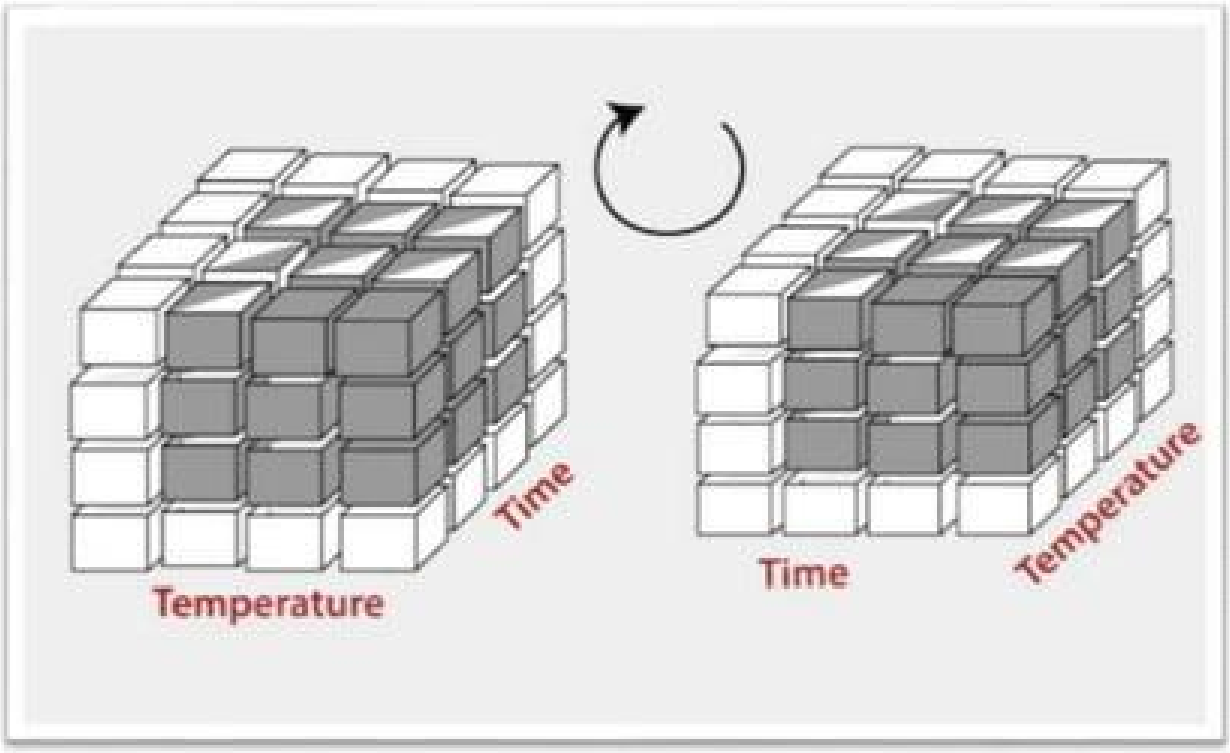
Consider the following diagram, which shows the dice operations.



- The dice operation on the cubes based on the following selection criteria involves three dimensions.
- (location = "Toronto" or "Vancouver")
- (time = "Q1" or "Q2")
- (item =" Mobile" or "Modem")

Pivot

- The pivot operation is also called a rotation. Pivot is a visualization operations which rotates the data axes in view to provide an alternative presentation of the data. It may contain swapping the rows and columns or moving one of the row-dimensions into the column dimensions.


Time
Temperature
Time
Temperature

- Consider the following diagram, which shows the pivot operation.