DATA WAREHOUSE

WHAT , WHY AND HOW ?

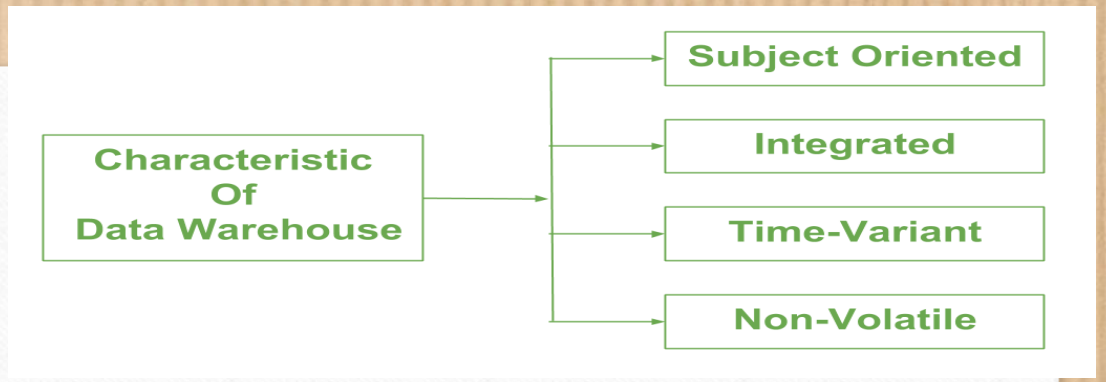
Agenda

- DEFINITION
- DATABASE vs DATAWAREHOUSE
- OLTP vs OLAP
- ETL vs ELT
- FACT vs DIMENSION
- STAR , SNOWFLAKE & FACT CONSTELLATION SCHEMA
- DATA MART

What is Datawarehouse?

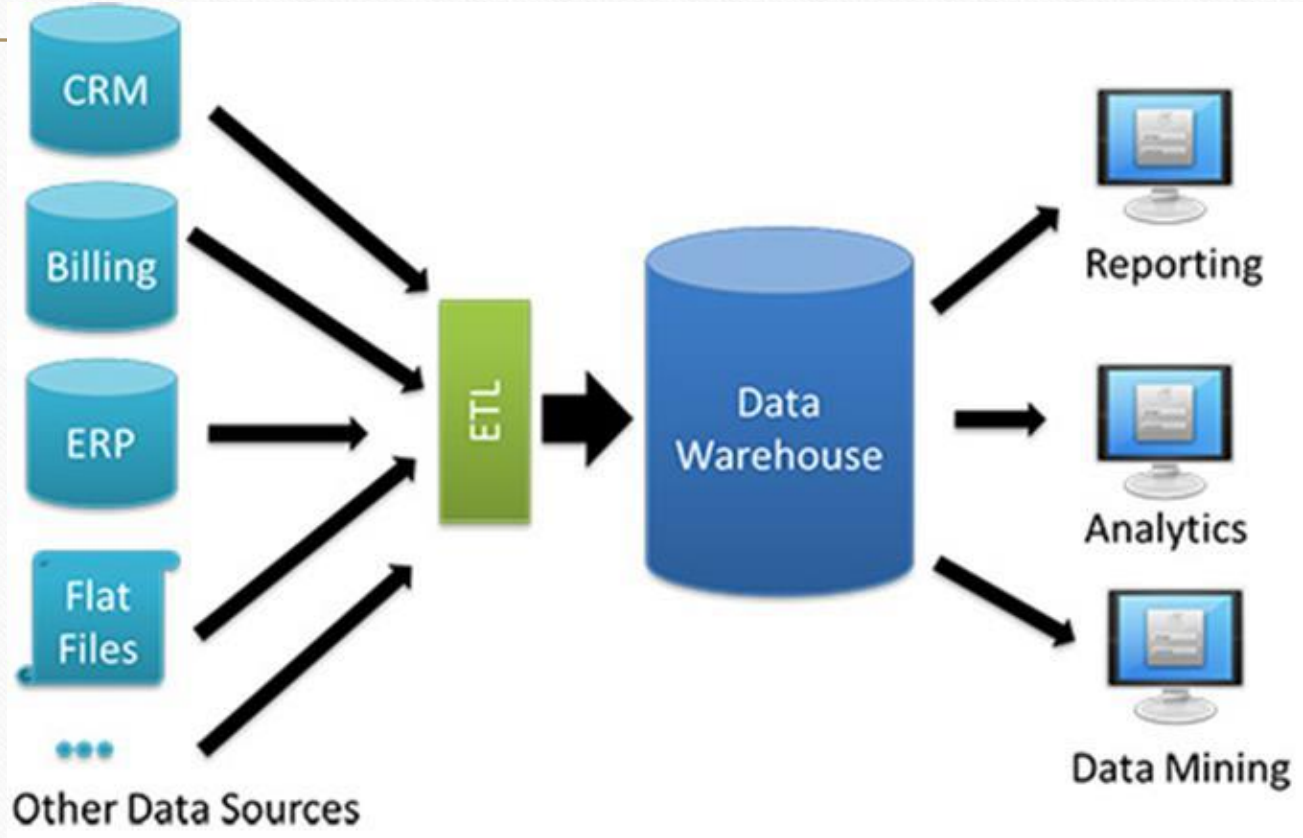
- The term "Data Warehouse" was first coined by Bill Inmon in 1990. According to Inmon, a data warehouse is a **subject-oriented**, **integrated**, **time-variant**, and **non-volatile** collection of data. This data helps analysts to take informed decisions in an organization.
- A Data Warehousing (DW) is process for collecting and managing data from varied sources to provide meaningful business insights. It is optimized for fast querying and analysis, enabling organizations to make informed decisions by providing a single source of truth for data A Data warehouse is typically used to connect and analyze business data from heterogeneous sources. The data warehouse is the core of the BI system which is built for data analysis and reporting

DWH Characteristics



- **Subject Oriented** :A data warehouse focuses on specific subjects or topics, like sales, marketing, or finance. It organizes data based on what people need to analyze. It's like sorting your bookshelf by genre—putting all the mystery books together, all the romance books together, and so on.
- **Time Variant:** In this data is maintained via different intervals of time such as weekly, monthly, or annually etc. Data warehouses keep track of historical information over time. For example, they can show how sales numbers have changed month by month or year by year. Another feature of time-variance is that once data is stored in the data warehouse then it cannot be modified, alter, or updated.
- **Integrated:**This means that a data warehouse combines information from different sources like Databases, ERP,Flat files CRM etc. Integration of data warehouse benefits in effective analysis of data. Reliability in naming conventions, column scaling, encoding structure etc. should be confirmed. Integration of data warehouse handles various subject related warehouse.
- **Non volatile :**As the name defines the data resided in data warehouse is permanent. It also means that data is not erased or deleted when new data is inserted. Data is not updated, once it is stored in the data warehouse, to maintain the historical data.
  In this, data is read-only and refreshed at particular intervals. This is beneficial in analysing historical data and in comprehension the functionality. It does not need transaction process, recapture and concurrency control mechanism. Functionalities such as delete, update, and insert that are done in an operational application are lost in data warehouse environment

Architecture



Types of Datawarehouse

- Enterprise Data Warehouse (EDW)
- Operational Data Store
- Data Mart

- e.g. Teradata, SQL server, DB2 ,Snowflake, Redshift ,Synapse etc.

Database vs Datawarehouse

- A database is a collection of related data that represents some elements of the real world, whereas a Data warehouse is an information system that stores historical and commutative data from single or multiple sources.
- A database is designed to record data, whereas a Data warehouse is designed to analyze data.
- A database is an application-oriented collection of data, whereas Data Warehouse is a subject-oriented collection of data.
- Database uses Online Transactional Processing (OLTP), whereas Data warehouse uses Online Analytical Processing (OLAP).
- Database tables and joins are complicated because they are normalized, whereas Data Warehouse tables and joins are easy because they are denormalized.
- ER modeling techniques are used for designing Databases, whereas Dimension modeling techniques are used for designing Data Warehouse.

ETL vs ELT

- ETL stands for Extract, Transform and Load, while ELT stands for Extract, Load, Transform.
- ETL loads data first into the staging server and then into the target system, whereas ELT loads data directly into the target system.
- ETL model is used for on-premises, relational and structured data, while ELT is used for scalable cloud structured and unstructured data sources.
- Comparing ELT vs. ETL, ETL is mainly used for a small amount of data, whereas ELT is used for large amounts of data.
- When we compare ETL versus ELT, ETL doesn't provide data lake support, while ELT provides data lake support.
- Comparing ELT vs ETL, ETL is easy to implement, whereas ELT requires niche skills to implement and maintain.

ETL VS ELT

CATEGORY	ETL	ELT
SUPPORT DATA WAREHOUSE	Yes, ETL is the traditional process for transforming and integrating structured or relational data on-premises data warehouse. But it does not compatible with Data lake while ELT does.	Yes, ELT is the modern process for transforming and integrating structured/semi or unstructured data into a cloud-based data warehouse
SETUP/ HARDWARE	Traditional, On prem ETL tool require expensive hardware	ELT process is Cloud based; No additional hardware is required.
TRANSFORM	Raw Data is transformed on processing layer, We have Staging area to perform all this.	Raw Data transformed inside the Target system.
SPEED	ETL has additional steps before it loads data into target that slows the system down.	ELT is faster than ETL, It load data directly into destination
COST	Need to hit source again & again so Time consuming & costly to set up	ELT benefits from a robust ecosystem of cloud-based platforms which offer much lower costs
RE-QUERY/AGILITY	In case of any new logic/Transformation We must re query or rerun the entire pipeline from source.	Here we have all data stored in destination so easily can apply many transformations.
IDEAL For	When we are processing smaller, relational data which require complex logics	ELT can handle any type of data and it is ideal for large datasets that require efficiency and speed

Fact vs Dimension

- **Fact**
- Facts are the measurements/metrics or facts from your business process. For a Sales business process, a measurement would be quarterly sales number .

- It will answers, How much, How many, Patterns/Trends & Performance Metrics.

- **Dimension**
- Dimension provides the context surrounding a business process event. In simple terms, they give who, what, where of a fact. In the Sales business process, for the fact quarterly sales number, dimensions would be

- Who – Customer Names
- Where – Location
- What – Product Name
- In other words, a dimension is a window to view information in the facts.

TYPE OF FACT Tables

Transactional Fact Table:

Explanation: Transactional fact tables capture detailed data about individual business transactions or events. They contain granular data at the transaction level, recording every occurrence of a particular event. Transactional fact tables are often used for operational reporting and auditing purposes.

Example: A sales transactional fact table would store detailed information about each sale, including the date, product sold, quantity, price, customer ID, and salesperson ID.

Periodic Snapshot Fact Table:

Explanation: Periodic snapshot fact tables capture data at regular intervals, such as daily, weekly, or monthly snapshots of business activities. They provide a snapshot view of the business at specific points in time, allowing for trend analysis and comparison over time.

Example: A monthly sales snapshot fact table would store aggregated data for each month, including total sales revenue, number of orders, and average order value.

TYPE OF FACT Tables

Accumulating Snapshot Fact Table:

Explanation: Accumulating snapshot fact tables track the progress or status of a process or workflow over time. They capture key milestones or stages in a process and record the duration or status of each stage. Accumulating snapshot fact tables are commonly used for process monitoring and performance analysis.

Example: A customer lifecycle snapshot fact table would track the progression of customers through various stages, such as acquisition, onboarding, engagement, and churn, along with the duration spent in each stage.

Factless Fact Table:

Explanation: Factless fact tables contain no measures or numeric data. Instead, they capture relationships or associations between different entities or dimensions. Factless fact tables are used to represent events or occurrences without numerical values.

Example: A student enrollment factless fact table would store records of student enrollments in courses, including the student ID, course ID, semester, and enrollment status, without any quantitative measures.

Dimension Table & Key points

- A dimension table is a type of table that stores descriptive information about business entities. These entities could include customers, products, time periods, geographic locations, or any other aspect of interest to the organization's operations. Key characteristics of dimension tables include:

1. **Descriptive Attributes:** These tables hold information that describes different parts of a business, like customers, products, time, places, or salespeople. For example, in a sales data set, you might find details about customers' names, product descriptions, or where sales happened.
2. **Primary Key:** Each row in a dimension table has a special ID called a primary key. This key makes sure that each row is unique in the table. It's used to connect the dimension table to the fact table.
3. **No Numeric Data:** Unlike the fact table, dimension tables don't contain numbers that show quantities or amounts. Instead, they store things like names, descriptions, or categories.
4. **Hierarchical Structure:** Some dimension tables have a structure like a family tree, where items are organized into levels. For example, a time dimension table might have levels for years, months, days, and so on.
5. **Low Cardinality:** Dimension tables usually have fewer unique values compared to fact tables. This means there are fewer different options for each item in the table. It helps make searching and analyzing data faster.

TYPEs OF DIMENSION TABLE

Role-Playing Dimension Table:

Explanation: A role-playing dimension table is one that serves multiple roles or purposes within a data model. It represents the same entity but from different perspectives or contexts.

Example: A "Date" dimension table can be role-played to represent different date attributes such as Order Date, Ship Date, and Delivery Date in a sales transaction scenario.

Conformed Dimension Table:

Explanation: A conformed dimension table is one that is shared and consistent across multiple data marts or subject areas within an organization. It ensures uniformity and consistency in reporting and analysis.

Example: A "Customer" dimension table used by both the Sales and Marketing departments, ensuring that customer attributes are consistent and standardized across different reports and analyses.

TYPEs OF DIMENSION TABLE

Junk Dimension Table:

Explanation: A junk dimension table is a small, auxiliary dimension table that stores low-cardinality, non-meaningful attributes that do not fit into other dimension tables. It helps reduce the complexity of the data model.

Example: A "Promotion" dimension table storing flags or codes for various promotional campaigns, such as "Free Shipping" or "Holiday Discount," which are not specific to any other dimension

Degenerate Dimension Table:

Explanation: A degenerate dimension table is a dimension table that consists of one or more attributes that are part of the fact table itself. It eliminates the need for a separate dimension table.

Example: An "Order Number" degenerate dimension in a sales fact table, where the order number serves as a unique identifier for each transaction but does not require additional descriptive attributes.

Slowly changing Dimension

Slowly Changing Dimensions (SCDs) are used in data warehousing to track changes to dimensional attributes over time. In Simple Dimensions which values will be changes e.g. Address, Phone number etc.

- **SCD TYPE-0: NO CHANGE**

- **Explanation:** In a Type 0 SCD, the dimension attributes never change. Once data is loaded into the dimension table, it remains static and does not get updated, regardless of any changes in the source system.
- **Use Case:** This type of SCD is suitable when historical accuracy is essential, and there should be no changes to the dimension data over time.
- **Example:** Consider a "Product" dimension table where the product attributes such as product name, category, and description remain constant and do not change over time.
- **SCD TYPE 1 : OVERWRITE (NO HISTORICAL INFORMATION)**
- **Explanation:** In a Type 1 SCD, changes to dimensional attributes are simply overwritten with new values. There is no preservation of historical data, and the dimension attributes are updated in place.
- **Use Case:** This type of SCD is appropriate when historical data is not important, and only the most recent information is required.
- **Example:** In a "Customer" dimension table, if a customer changes their address, the existing address information is overwritten with the new address without preserving the old address.

Slowly changing Dimension

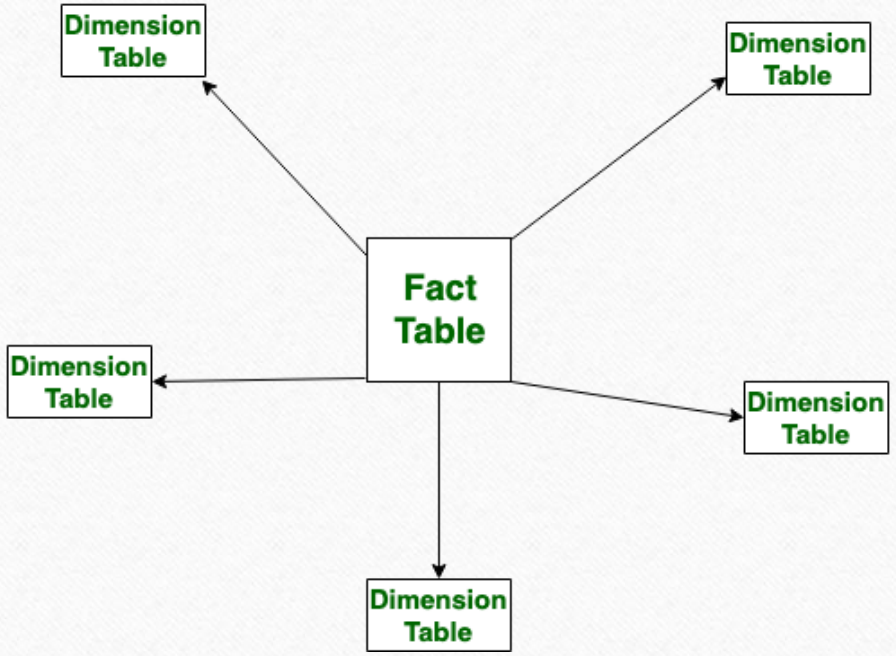
- **SCD TYPE-2: ADD NEW RECORD (with Active flag & Date)**

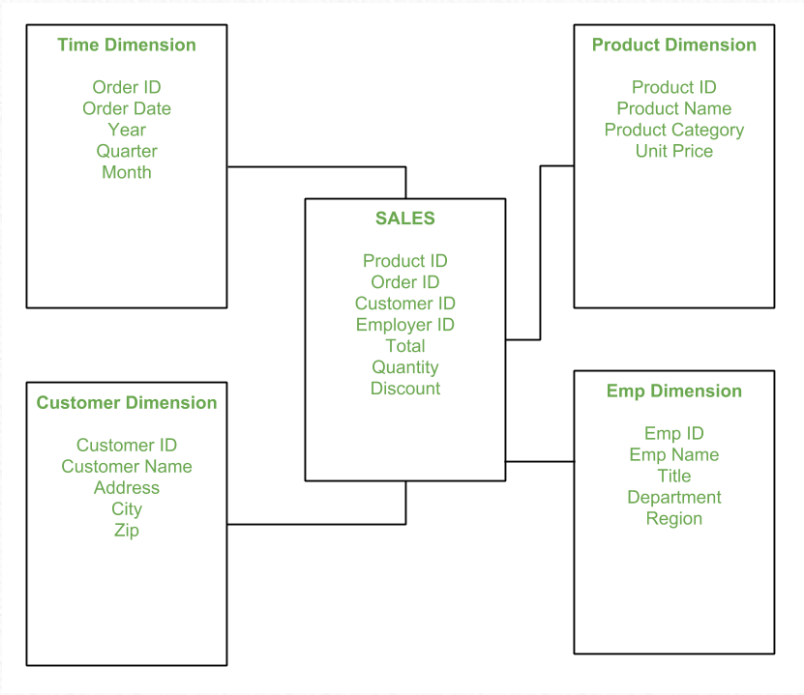
- **Explanation:** In a Type 2 SCD, changes to dimensional attributes are tracked by creating new records for each change. This preserves historical data by maintaining a history of changes over time.
- **Use Case**: This type of SCD is suitable when it's essential to track historical changes and maintain a full history of dimension data.
- **Example:** In an "Employee" dimension table, if an employee's department changes, a new record is created with the updated department information, while retaining the previous record with the old department.
- **SCD TYPE 3 : ADD NEW COLUMN**
- **Explanation:** In a Type 3 SCD, only a limited history of changes is maintained by adding new columns to the dimension table to track specific changes over time. Unlike SCD-2, which creates new records for each change, SCD-3 updates the existing record while retaining historical information in additional columns..
- **Use Case:** SCD-3 is useful when it's necessary to track certain changes to dimensional attributes over time while still maintaining a compact and efficient data model.
- **Example:** Consider a "Product" dimension table where the original price of a product needs to be tracked along with the current price. In SCD-3, a new column "Original_Price" is added to the dimension table to capture the initial price, while the existing "Price" column is updated with the current price

Star Schema vs Snowflake Schema

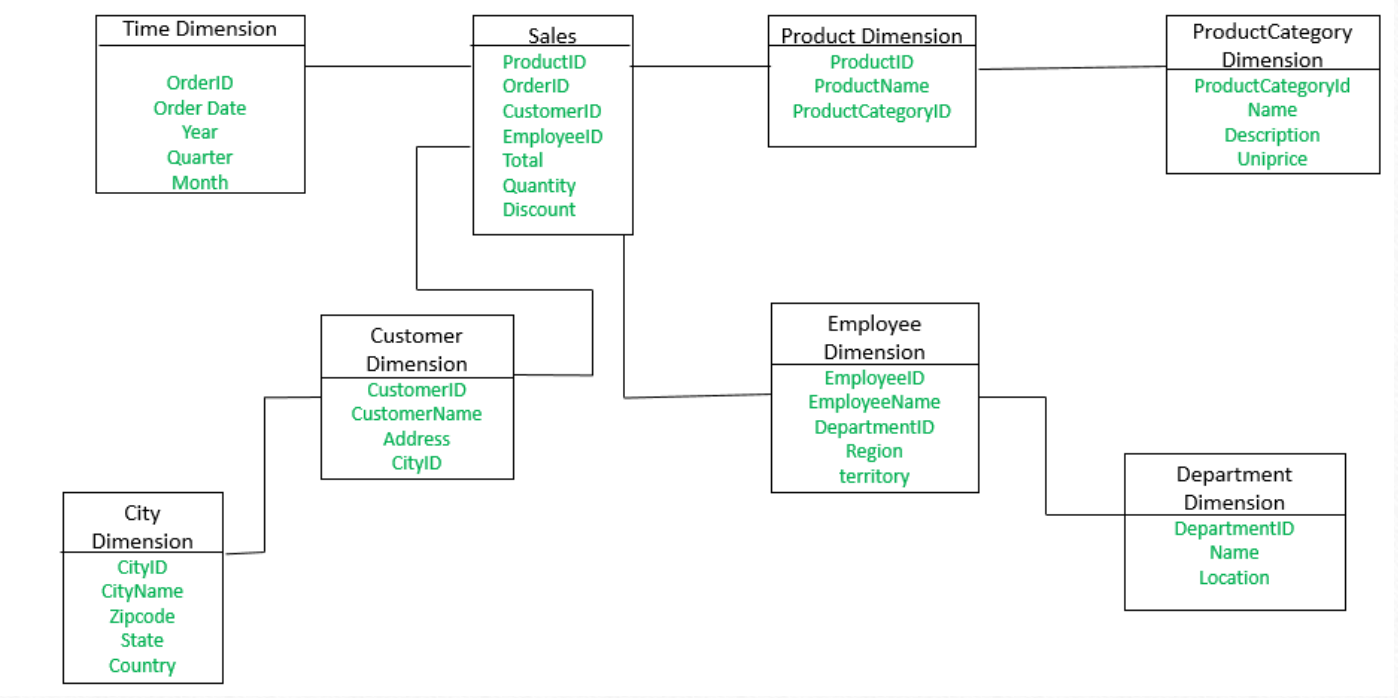
S.N	Star Schema	Snowflake Schema
1	In star schema, The fact tables and the dimension tables are directly connected	While in snowflake schema, The fact tables, dimension tables as well as sub dimension tables are contained and all Dimensions are not connected directly to fact
2	Star schema is a top-down model. (Bill Inmomn method)	While it is a bottom-up model. (Kimball)
3	It takes less time for the execution of queries.	While it takes more time than star schema for the execution of queries.
4	Less Normalized so High Data Redundancy	Highly Normalized so low Data Redundancy
5	The query complexity of star schema is low.	While the query complexity of snowflake schema is higher than star schema.
6	It's understanding is very simple.	While it's understanding is difficult.
7	It has less number of foreign keys.	While it has more number of foreign keys.

Star Schema Design

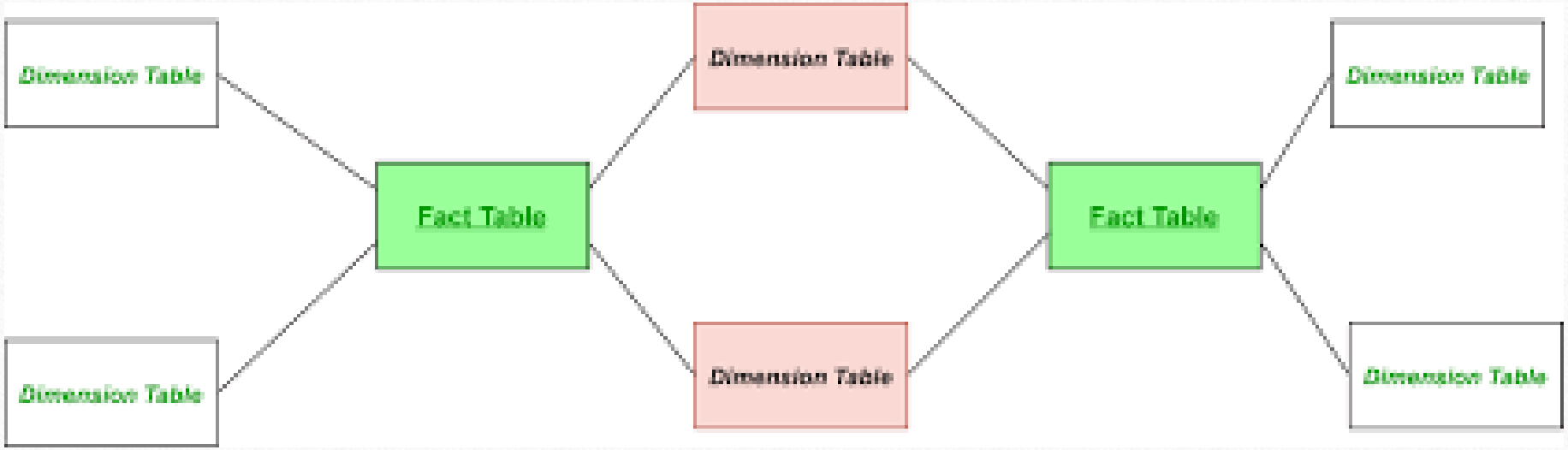




Snowflake Design



Fact Constellation Schema (Galaxy Schema)



Data Mart vs Data Warehouse

- Data Warehouse is a large repository of data collected from different sources, whereas Data Mart is only subtype of a data warehouse.
- Data Warehouse is focused on all departments in an organization, whereas Data Mart focuses on a specific group.
- Data Warehouse designing process is complicated, whereas the Data Mart process is easy to design.
- Data Warehouse takes a long time for data handling, whereas Data Mart takes a short time for data handling.
- Comparing Data Warehouse vs Data Mart, Data Warehouse size range is 100 GB to 1 TB+, whereas Data Mart size is less than 100 GB.

Data warehouse vs. data mart

	Data warehouse	Data mart
SIZE	100s of GBs to TBs and beyond	100s of GBs to potentially TBs
SUBJECT AREA	Global view of the enterprise	More granular view: business unit, subject area, etc.
DATA SOURCES	Applications across the organization	Sources that pertain to a specific business unit or subject area
OWNERSHIP AND CONTROL	Enterprise level	Department level
EASE OF ACCESS FOR BUSINESS UNITS	Access is more tightly controlled and often requires a high level of expertise to analyze and present decision-making data to end users	The system is built to promote ease of access and present users with decision-making data in the shortest time possible
TYPES OF DECISION-MAKING	Strategic and tactical	Strategic, tactical and operational
SPEED OF DECISION-MAKING	Longer	Shorter
STARTUP AND ONGOING SUPPORT COSTS	High startup and ongoing support costs	Lower startup and ongoing support costs
BUILD TIME	Months to years	Months

SOURCE: CHRIS FOOT

DATA LAKE VS DATAWAREHOUSE

	DATA LAKE	DWH
DATA TYPE	Structured, semi-structured, unstructured	Structured
RELATIONS	Relational, non-relational	Relational
SCHEMA	Schema on read	Schema on write
FORMAT	Raw, unfiltered	Processed, vetted
SOURCES	Big data, IoT, social media, streaming data	Application, business, transactional data, batch reporting
SCALABILITY	Easy to scale at a low cost	Difficult and expensive to scale
USERS	Data scientists, data engineers	Data warehouse professionals, business analysts
Use cases	Machine learning, predictive analytics, real-time analytics	Core reporting, BI
COST	Less expensive	Expensive and require more time to manage Data which results extra cost
PROCESSING	ELT is preferred	ETL is Preferred in this case

BY VISHAL KAUSHAL

4 STEP Process

- To create a Dimension Model, there is a 4 step process that should be followed that cover mostly everything
- 1- Select the Business Process
- 2-Declare the Grain
- 3-Identify the Dimensions
- 4- Identify the Facts