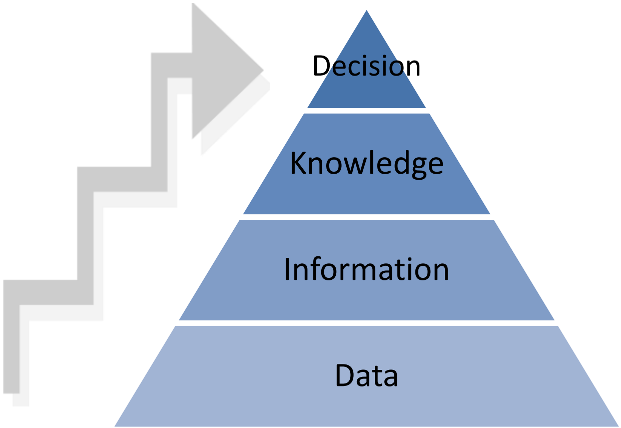
Data Warehouse

By: Abdelwahed Ashraf

@linkedin @gmail

Data –> Information –> Knowledge –> Decision



A Brief History of BI

- The term BI was coined by the Gartner Group in the mid-1990s
- However, the concept is much older
  - 1970s — MIS reporting — static/periodic reports
  - 1980s — Executive Information Systems (EIS)
  - 1990s — OLAP, dynamic, multidimensional, ad-hoc reporting -> coining of the term “BI”
  - 2005+ — Inclusion of AI and Data/Text Mining capabilities; Web-based Portals/Dashboards
  - 2010s — BI Term is evolved to include BA & BPM

Introduction to DWH

Motivation to the Data Warehouse (DWH)

- Data could be a product for some companies.
- It could be decision support for other products or businesses.
- It could be reporting the results after passing the data life-cycle from storage (Database). Some challenges are facing the people who work on data management backend:
  - Performance,
  - Integration,
  - and Applying analytical functions.
- Vendors who are working on solving the above challenges are creating their product of DWH. Their ultimate goal is to optimize the above points.

Definition of a Data Warehouse (DWH)

A DWH is a technique for collecting and managing data from varied sources to **provide meaningful business insights**.

The information is subject orientated, recorded over time and may be stored at various degrees of summarization

Motivation to the Data Warehouse (DWH)

- The DWH is not a product but an environment.
- It is a process of transforming data into information and make it available to users in a **timely manner** to make a difference.
- It is an architectural construct of an information system that provides users with current and historical decision support information which is difficult to access or present in the traditional operational data store.
- The DWH is the core of the BI system built for data analysis and reporting.

Motivation to the Data Warehouse

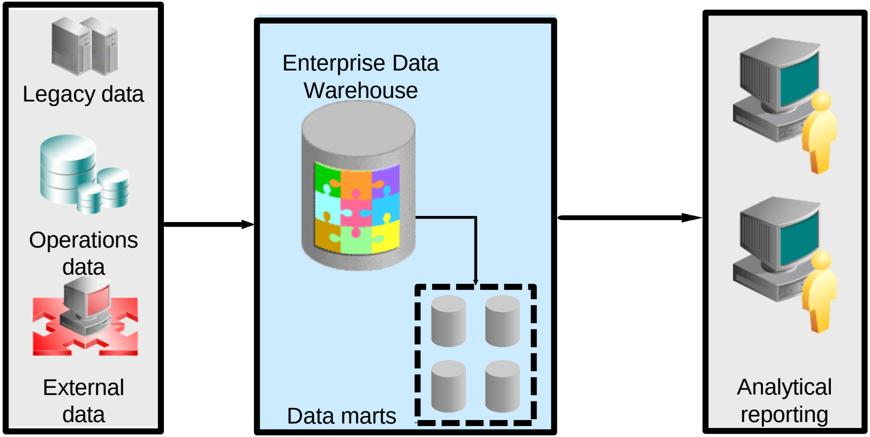
Other names for the Data warehouse system:

- Decision Support System (DSS).
- Business Intelligence Solution.
- Executive Information System.
- Management Information System.
- Analytic Application.
- Data Warehouse.

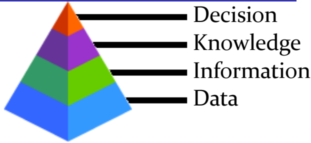
Differences Between DWH and Operational DB

Transactions DB (OLTP)	DWH
Works with small Pieces of Information	Works with Enterprise wide Information
Support Insert, Update, Delete or Select	Read Only
Normalized	Not required (De-normalized in many use cases)
Small To Large Database	Large to Very Large Database
Volatile Data	Non Volatile
Applications that **Run** the business	Applications that **analyze** the business

Data Warehousing and Business Intelligence

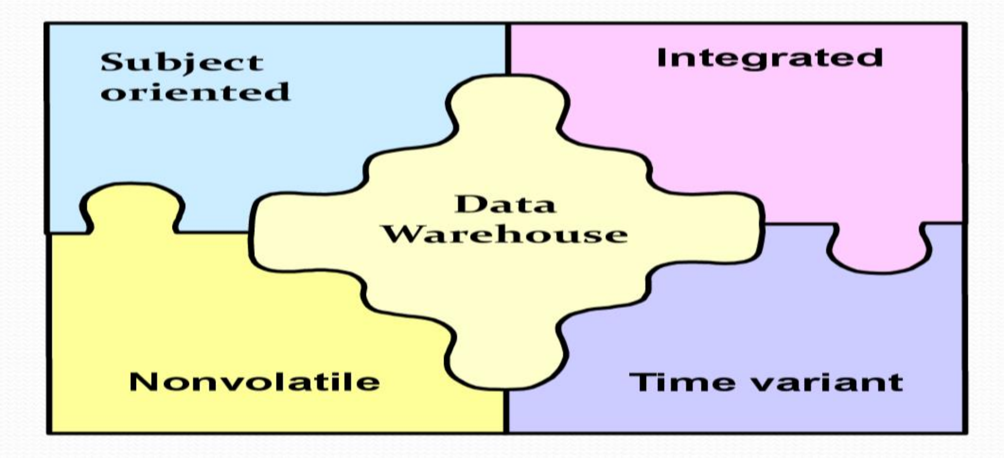


Definition of BI



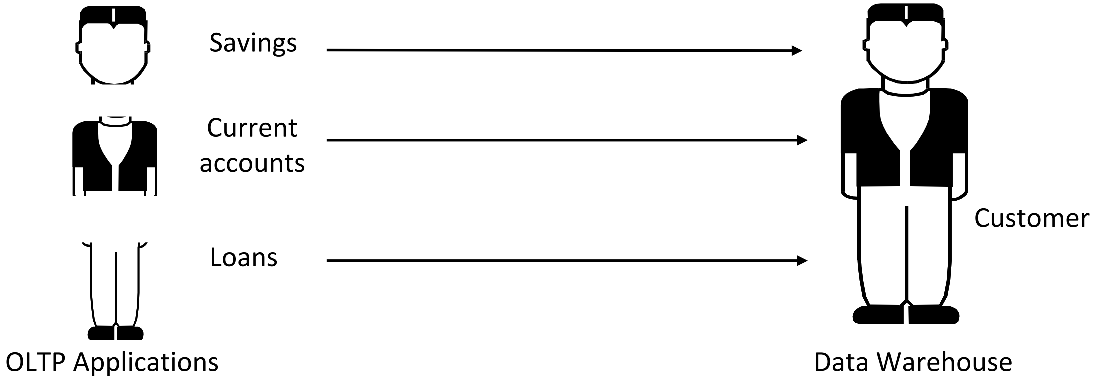
- BI is an umbrella term that combines architectures, tools, databases, analytical tools, applications, and methodologies.
- BI a content-free expression, so it means different things to different people.
- BI helps transform data, to information (and knowledge), to decisions and finally to action.

DWH Characteristics



The characteristics of DWH:

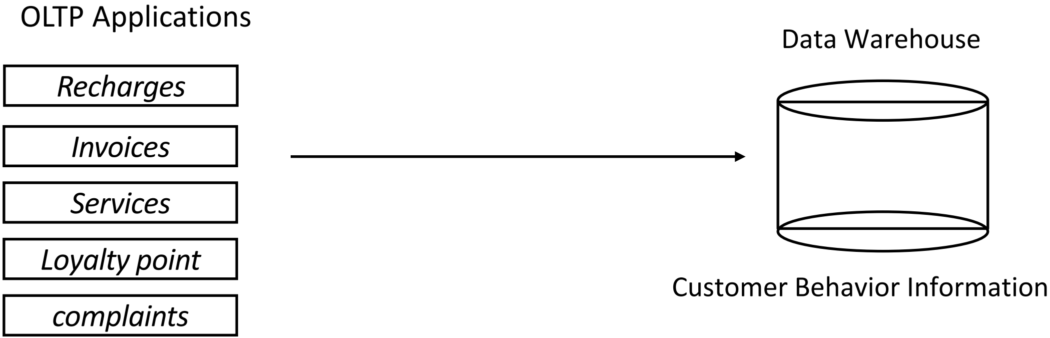
- **Integrated:**
  - *DWH is an integrated environment which allows us to integrate different source systems. Data are modeled (organized) in a unified manner.*



The characteristics of DWH:

Subject-oriented:

- Data is categorized and stored by business subject rather than by application.



OLTP Applications

Recharges

Invoices

Services

Loyalty point

complaints

Data Warehouse

Customer Behavior Information

The characteristics of DWH:

Time-Variant:

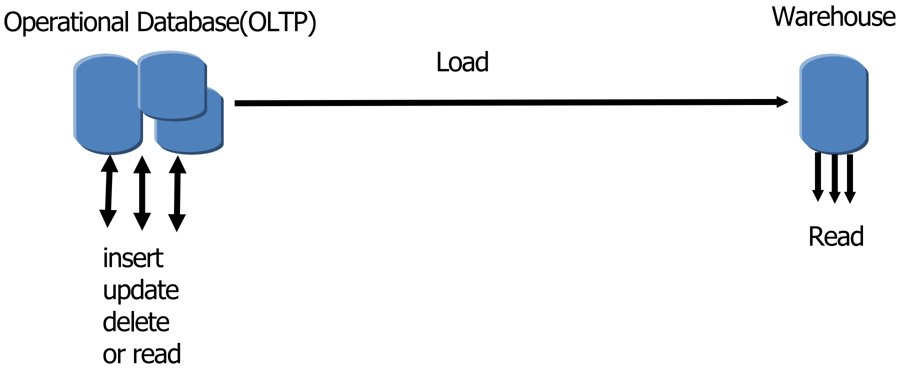
- Data modeled (organized) based on periods (hourly, daily, weekly, monthly, quarterly, yearly).

Time	Data
Jan-97	January
Feb-97	February
Mar-97	March

The characteristics of DWH:

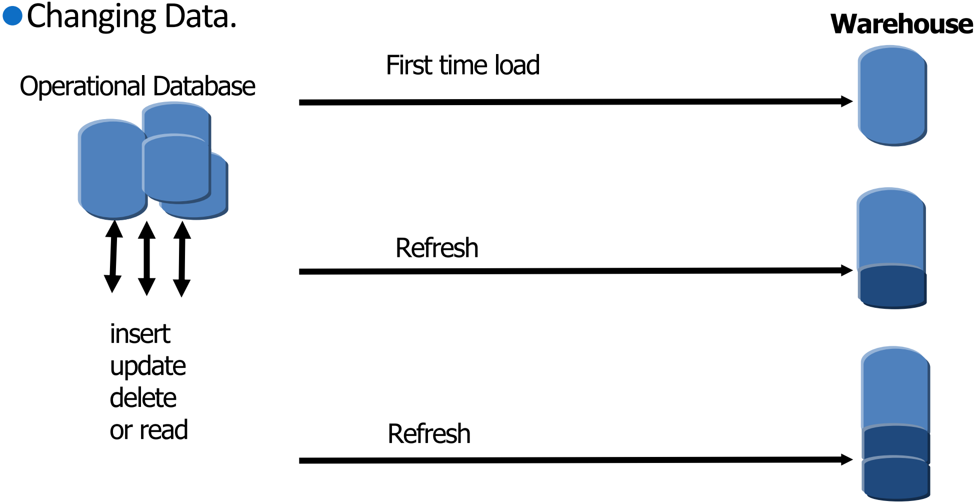
● Non-Volatile:

- Typically, data in the data warehouse is not updated or deleted.



The characteristics of DWH:

- **Non-Volatile:**
  - Changing Data.



Types of DWH

Motivation to Data Warehouse

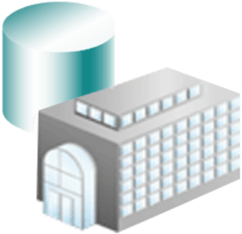
Types of Data Warehouse

Enterprise Data Warehouse (E-DWH): It provides decision support service across the enterprise. It offers a unified approach for organizing and representing data (DWH Model). It offers data classifications according to the subject with privileges policy.

Operational Data Store (ODS): is a central database that provides an up-to-date (real-time) data from multiple transnational systems for operational reporting into a single DWH.

Data Mart: A departmental data warehouse that stores only relevant data, It specially designed for a particular line of business, such as sales or finance.

Enterprise Data Warehouse (E-DWH)



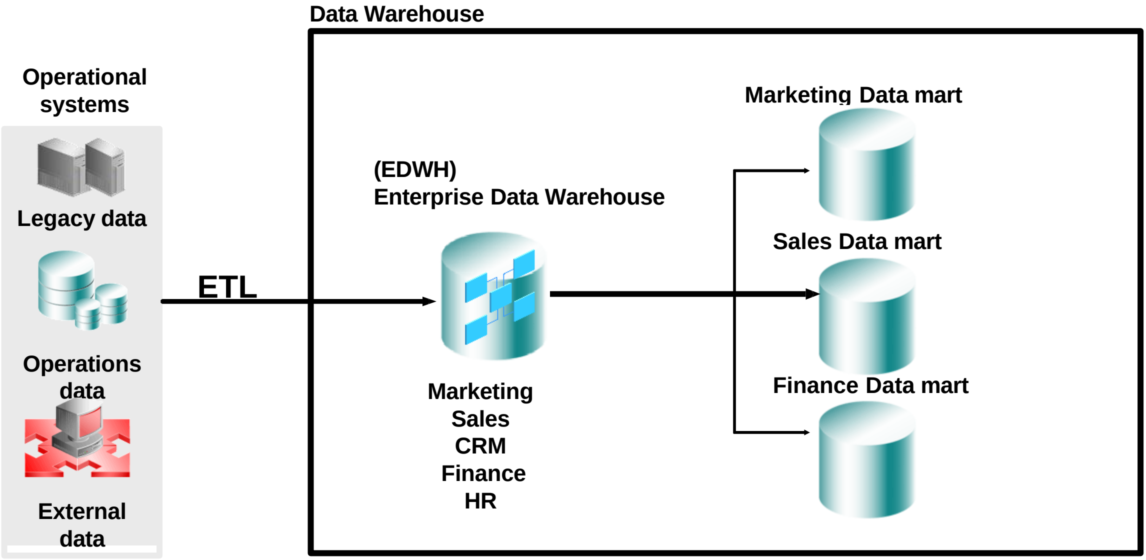
- Supports large-scale implementation
- Scopes the entire business
- Contains data from all subject areas
- Is developed incrementally
- Is a single source of enterprise-wide data
- Is the single distribution point to dependent data marts

Data Mart

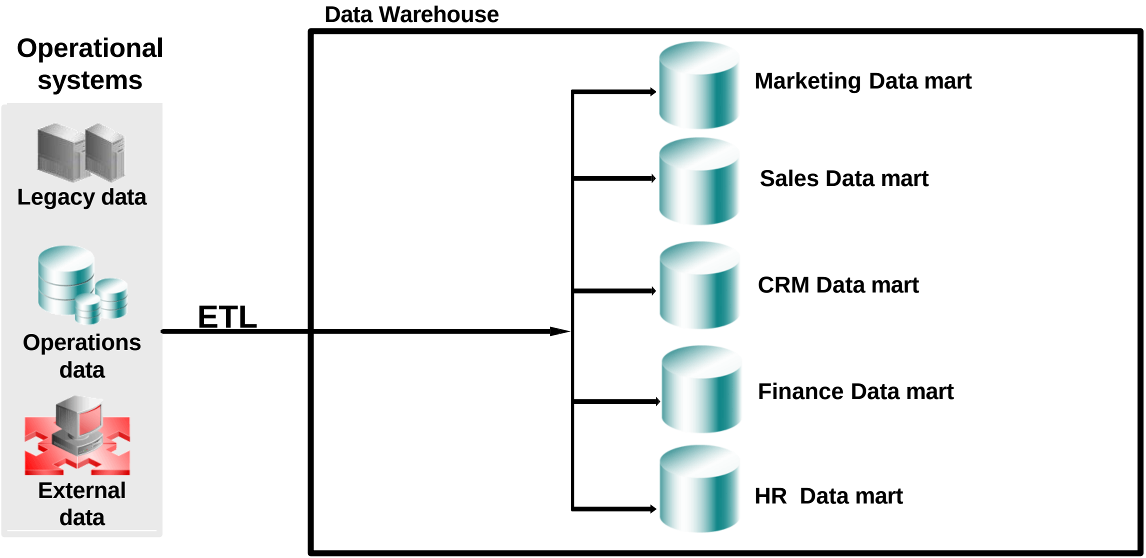
Data Mart: A departmental data warehouse that stores only relevant data

- Dependent data mart
  A subset that is created directly from a E-data warehouse
- Independent data mart
  A small data warehouse designed for a strategic business unit or a department

Dependent Data Mart



Independent Data Mart



DWH vs ODS vs Data Mart

Metric	E-DWH	ODS	Data Mart
Latency	Day -1 (Batch)	Real-time(Stream)	Day -1
Data level	Transnational	Transnational	Summary
Historical	Long-term	Snapshot	Aggregated Long-Term
Size	TB/PB	GB	GB/TB
Orientation	Multi sources	Multi sources	Product
Business Units	Multi organizational units	Product team	Business team

Use Cases of Operational DB vs DWH

Use case (Operational DB)

A Retail Sales company named XSales.

- They have lots of systems.
- One of this systems is a CRM system as example of operational DB.
- The CRM system handles the customer activities with the company including (sales, Return, inquiries and other activities).
- This system has a backend database (OLTP).
- CRM team can report their sales and customer activities from their database.
- Product owner can take a decision based on their system backend reports.

Use case (DWH)

What is the need for DWH?

- This company has other systems Marketing, Stock, Call center
- They need to report information related to the CRM, Stock , Call center source systems in one report.
  - So, they need to ingest (transfer) the data from the source systems to one single database.
  - The decision from the DHW is a **global and strategical decision.**
  - If the company needs to build a machine learning model which needs data from different sources. They need to load the data from a centralized database rather than read each source alone.

Use case (ODS)

- Why do we need the ODS?
- How does it fit in our system?

Use case (ODS)

XSales has a call center system which handles the customer inquiries.

This system requires the some data related to order status, Return, customer information, billing details, shipping to be calculated and accumulated in **real-time** to be able to give the customer the right answer for his inquires.

Use case (ODS)

So, What is the challenge for this system?

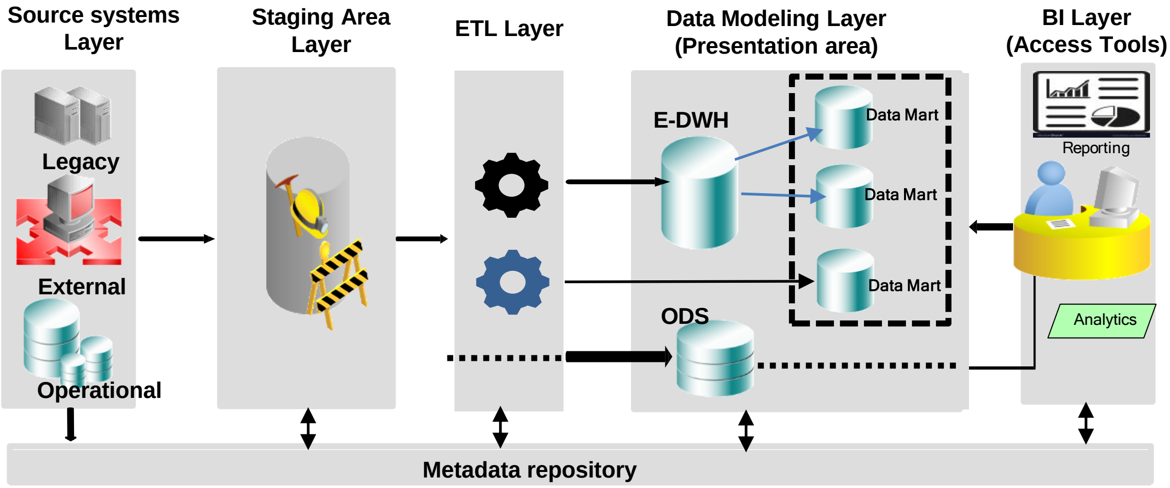
- It needs specific information from different source systems.
- It requires to track the source system database changes or update **in real-time.**
- It’s functionality is based on the aggregate data not the transactions

Use case (ODS)

- ODS is based on change data capture (CDC).
- This approach used to determine the data change and apply action based on this change.
- ODS uses the real-time aggregations to support the online systems from different source systems.

DWH Architecture

Data Warehouse Components



Data Warehousing Architectures

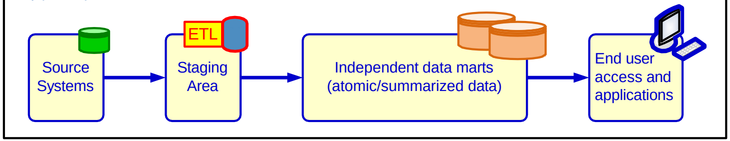
- Issues to consider when deciding which architecture to use:
  - Which database management system (DBMS) should be used?
  - Will parallel processing and/or partitioning be used?
  - Will data migration tools(ETL) be used to load the data warehouse?
  - What tools will be used to support data retrieval and analysis?

Hosted Data Warehouses

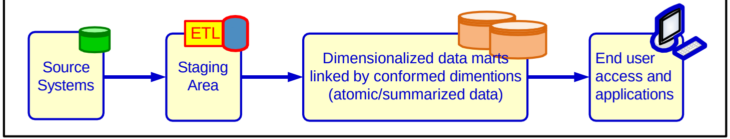
- Benefits:
  - Requires minimal investment in infrastructure
  - Frees up capacity on in-house systems
  - Frees up cash flow
  - Makes powerful solutions affordable
  - Enables powerful solutions that provide for growth
  - Offers better quality equipment and software
  - Enables users to access data remotely
  - Allows a company to focus on core business
  - Meets storage needs for large volumes of data

Five Main DW Architectures

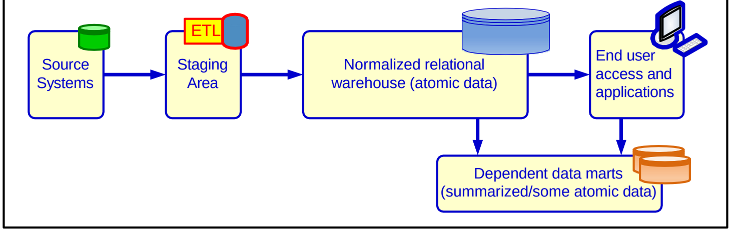
(a) Independent Data Marts Architecture


Source Systems
ETL
Staging Area
Independent data marts (atomic/summarized data)
End user access and applications

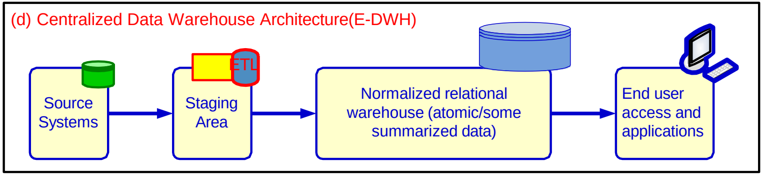
(b) Data Mart Bus Architecture with Linked Dimensional Datamarts


Source Systems
ETL
Staging Area
Dimensionalized data marts linked by conformed dimentions (atomic/summarized data)
End user access and applications

(c) Hub and Spoke Architecture (Corporate Information Factory)

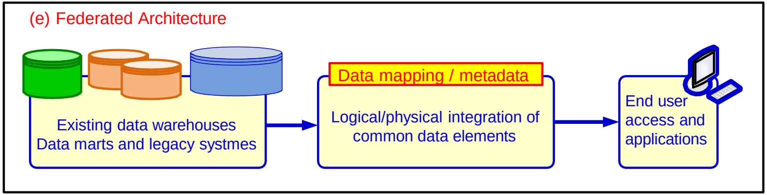

Source Systems
ETL
Staging Area
Normalized relational warehouse (atomic data)
End user access and applications
Dependent data marts (summarized/some atomic data)

Five Main DW Architectures



(d) Centralized Data Warehouse Architecture(E-DWH)

Source Systems
ETL
Staging Area
Normalized relational warehouse (atomic/some summarized data)
End user access and applications



(e) Federated Architecture

Existing data warehouses Data marts and legacy systmes
Data mapping / metadata
Logical/physical integration of common data elements
End user access and applications

Five Main DW Architectures

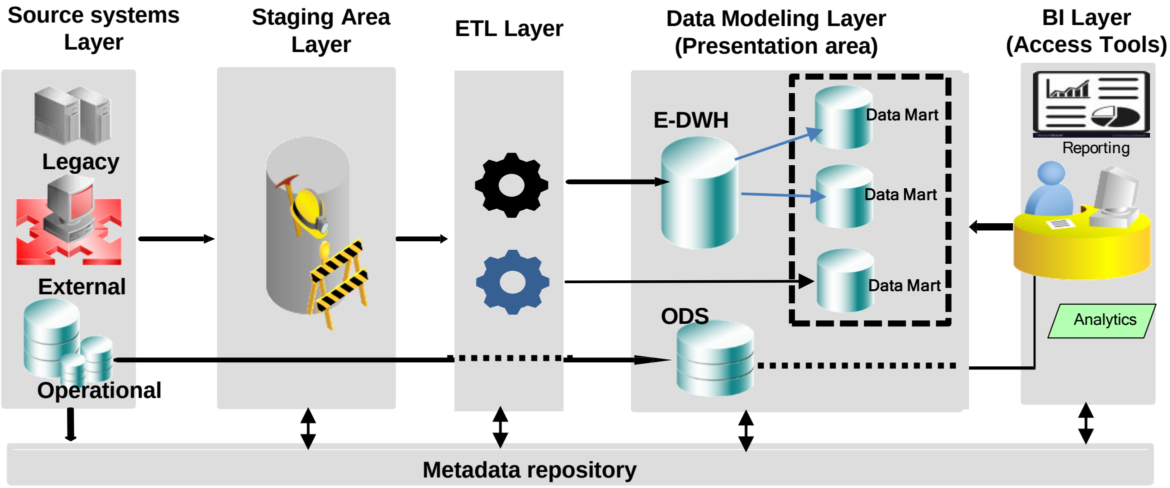
1. Independent Data Marts
2. Data Mart Bus Architecture
3. Hub-and-Spoke Architecture
4. Centralized Data Warehouse
5. Federated Data Warehouse

- Each has pros and cons!

Data Warehouse Development

- Data warehouse development approaches
  - Inmon Model: EDW approach (top-down)
  - Kimball Model: Data mart approach (bottom-up)
  - Which model is best?
    - There is no one-size-fits-all strategy to DW
  - One alternative is the hosted warehouse
- Data warehouse structure:
  - The Star Schema vs. Relational

Data Warehouse Components



DWH Architecture Layers

- DWH architecture contains the following layers:
  - Source system layer.
  - Extraction layer.
  - Staging Area.
  - Data Modeling.
  - ETL layer.
  - Storage layer.
  - Reporting (UI) layer.
  - Metadata layer.
  - System operations layer.

Source System Integration Process

Source System Integration Process

- In some companies, they hire or dedicate a team for this part (business analyst, system analyst, data analyst, or demand team).
- Before we start, we need to document all the communications into any format.
  - Word, Excel sheet or any other tool.
  - Make the discussion and put comments to make the history available always.
  - We need to clarify all the tasks and what is the expected output,

    (analysis means to document data structure, format, column names, etc.).

Source System Integration Process

- Requirements gathering.
- Identify the stakeholders (Data owner(s)).
- Data Analysis includes but not only (format, latency, and column definitions).

- Check the source system access and perform connectivity assessment.
- Initiate the technical discussion about the best way to ingest the data.
- Data Ingestion method and format.
- Sign or confirmation for every point between the stakeholders.
- This layer deliver a data analysis (Source system interface) document.

Extraction Layer

Extraction Layer

- In In some companies, they hire or dedicate a team for this part (extraction or ingestion team), but in other companies, it is part of the data engineering team.
- This layer takes the output analysis and decisions from the previous layer (source system analysis) and implement the extraction (quality from the previous team output highly affect this team).
- There is a lot of consideration this team needs to take care of or deal with, but we can summarize it in the following:
  - Data latency analysis as it affects the tool and the methodology (stream or batch).
  - Data extraction method (push or pull).
  - Data size and format compared with the available resources for this project.
- This layer output is a minimal data cleansing (no transformation) into the staging/landing layer.

Staging Layer

Staging Layer

- The main purpose of the Staging Layer is to load source data into the DWH environment for further processing (the process from source-to-staging).
- In other words, the Staging Layer is responsible for the physical movement of data from the source platform onto the DWH platform.
- all required data must be available before data can be integrated into the Data Warehouse.
- All the ETL layers are working on top of this layer.
- The decision of the storage type is based on the use case and the data.

Data Modeling Layer

What is data model?

- The data model
  - is An abstract model that organizes elements of data.
  - It describes the objects, entities, and data structure properties, semantic, and constraint.
  - It formalizes the relationship between entities.
  - It describes the conceptual design of a business or an application with its flow, logic, semantic information (rules), and how things are done.
  - It refers to a set of concepts used in defining such as entities, attributes, relations, or tables.

What is data model?

Data model is not

- a science.
- a static design for each organization.
- Nature of end-user tasks
- a new invention which needs to be done for each project.

Data model is

- an engineering design practices.
- a general concept that leads to build full architecture.
- different based on the use case and the database type.
- customizable, and we can utilize some of the ready built architecture.
- affecting information reporting performance.

What is data model?

- The data model
  - The first part before starting integration with any new source system.
  - The connection layer between business requirements and technical design.
  - It is also the translation between logical and physical layer.
  - It is unified across all systems and has the same patterns and practices.
  - It engaged with any source systems integration from the early stages.
  - This stage output is a data model design document or mapping sheet.

Why does the data model are important?

- Data models are currently affecting software design.
- It decides how engineers think about the problem they are solving.

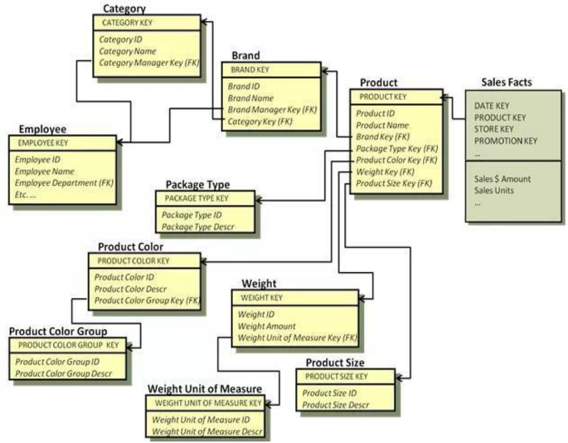
Dimensional Modeling

- **Dimensional Modeling :**

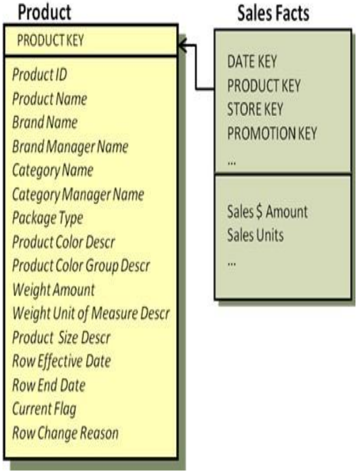
a technique for designing data warehouses that organizes data into facts and dimensions, a retrieval-based system that supports high-volume query access.

- **Schema:** a schema is a logical structure that defines how the facts and dimensions are related and stored in a database
- **Three common types of schemas in dimensional modeling:**
- star schema, snowflake schema, Fact Constellation Schema(Galaxy Schema).

Relational Vs Dimensional Models



Relational Model



Dimensional Model

Dimensional Modeling

- **Dimensional Modeling :**
  - **Fact tables**
    - Dimensions(FK)
    - Facts(Numeric /Measure) (Agg)
  - **Dimension tables**
    - Attributes

Elements of Dimensional Modeling

- **Facts:** are numerical measures of business events.

- **Dimensions:** are descriptive attributes that provide context for the facts, such as product, customer, or date.

- **Attributes :** are the various characteristics of the dimension. In the previous examples, the attributes can be customer details (from customer_id get the gender, age, nationality, etc.).

(Fact table) Elements of Dimensional Modeling

- **Fact table :** is a primary table in a dimensional model.
- A Fact Table contains (Measurements/facts and Foreign key to dimension table).
- It located at the center of schema and surrounded by dimensions.
- Most data in data warehouse is in fact tables, which can be extremely large
- Read-only data that will not change over time
- Most useful fact tables contain one or more numerical measures, or ‘facts’ that occur for each record.

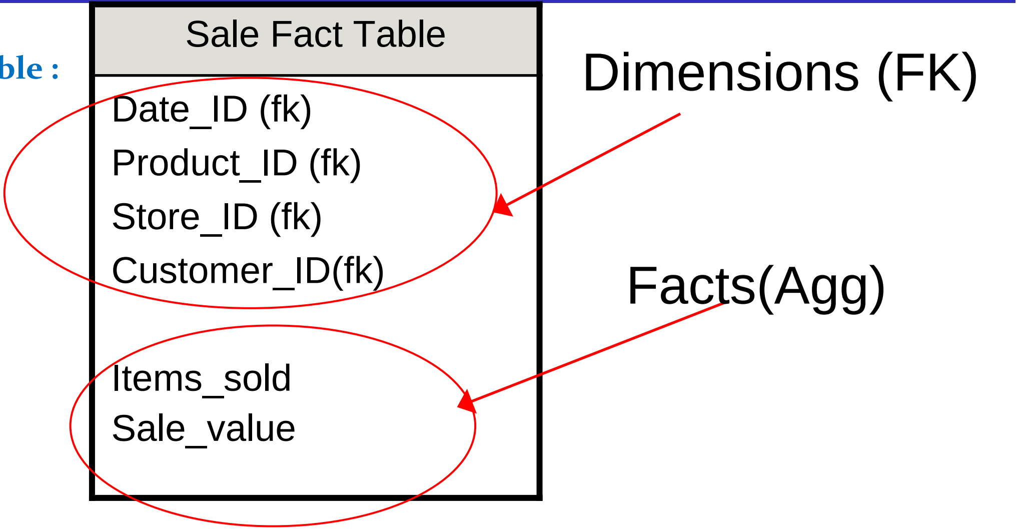
(Fact table) Elements of Dimensional Modeling

- List of dimensions defines the grain of the fact table
    - The dimensions are foreign keys (FK) that connects to primary keys of Dimension Tables

- Primary key of the fact table is combination of the foreign keys in the fact table
    - composite key

Fact table

- **Fact table :**



Sale Fact Table
Date_ID (fk) Product_ID (fk) Store_ID (fk) Customer_ID(fk) Items_sold Sale_value

Dimensions (FK)

Facts(Agg)

(Dimension table) Elements of Dimensional Modeling

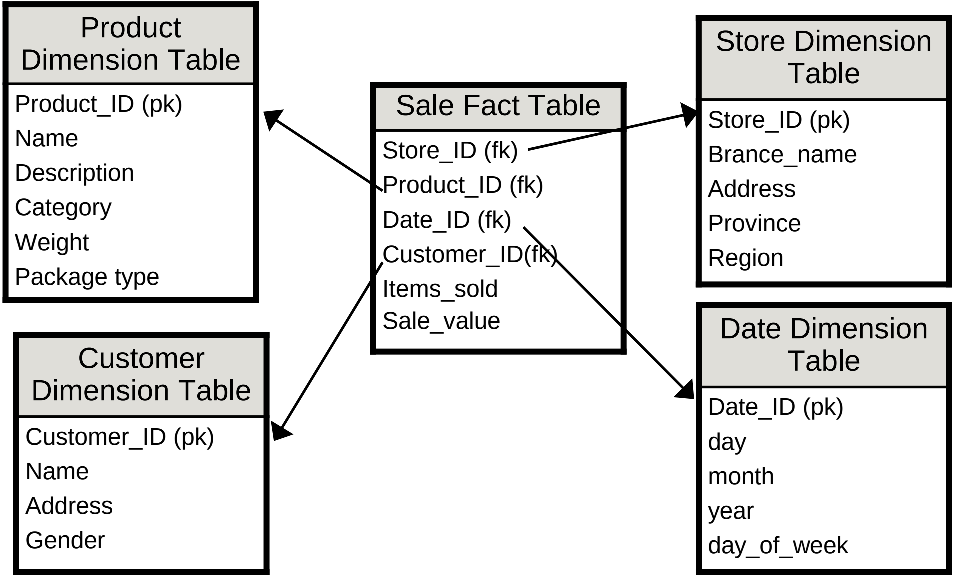
- **Dimension Tables:** Contain the textual context associated with a business process measurement event.
- Dimension tables usually contain descriptive textual information
- Dimension attributes are used as conditions in data warehouse queries
- In star schema, dimension table is denormalized to improve query performance

(Dimension table) Elements of Dimensional Modeling

Example: Dimension Tables

Product Dimension Table
Product_ID (pk)
Name
Description
Category
Weight
Package type

Connecting Fact and Dimension Tables(Star Schema)



Product Dimension Table

Product_ID (pk)
Name
Description
Category
Weight
Package type

Customer Dimension Table

Customer_ID (pk)
Name
Address
Gender

Sale Fact Table

Store_ID (fk)
Product_ID (fk)
Date_ID (fk)
Customer_ID(fk)
Items_sold
Sale_value

Store Dimension Table

Store_ID (pk)
Brance_name
Address
Province
Region

Date Dimension Table

Date_ID (pk)
day
month
year
day_of_week

Benefits of Dimension Model

- Simplicity
  - Data is easier to understand and navigate for business users
- Performance
  - Query processed more efficiently with fewer joins
- Easy for creating reports
  - Fact tables provide numeric values
  - Dimension attributes provide report labels.

Dimensional model life cycle:

- Steps to create Dimension Model:
    - The Gathering Requirements (Source Driven, Business/User Driven).
    - Identify granularity of the facts(Level of details)
    - Identify the dimensions
    - Identify the facts

Fact Table

Fact Table Recap:

- What is the fact table?
  - It is the foundation of the data warehouse.
  - It consists of facts and measurements of a particular business side and processes ex: daily revenue for a product.
  - It is the target of queries in most of DWH analysis and reports.
  - It contains measurements/facts and foreign keys to dimensions table.
  - It located at the center of the schema and surrounded by dimension tables.

How to design a fact table?

How to design a fact table?

- Choose the business process.
- Identify the grain.
- Identify the dimensions.
- Identify the facts.

Fact Granularity

- The grain is the definition of what a single row in the fact table will represent or contains.
- The level of detail or depth of the information recorded in a fact table is referred to as the table's grain.
- The grain describes the physical event which needs to be measured.
- Grain controls the dimensions which are available in fact.
- Grain represents the level of information we need to represent. It is not always time; it could be the physical business measurement level.
- A successful fact table must be designed at the lowest level.
- Design from the lowest possible grain.

Fact Granularity

- **Granularity**
- refers to the level of detail of the data stored fact tables in a data warehouse.
- Low granularity refers to detailed data that is at or near the transaction level (atomic level).
- Higher granularity refers to data that is summarized or aggregated, usually from the atomic level data.
- ***Atomic grain*** **refers to the lowest level of grain.**

Fact Table Types

Fact Tables Types

- There are three types of fact tables:
  - Transaction.
  - Periodic snapshot.
  - Accumulated Snapshot.

Transaction Fact Table

- Fact grain set at a single transaction (OLTP).
- It has one row per transaction.
- For each transaction, we add a new single record.
- The transaction fact table is known to grow very fast as the number of transactions increases.

Fact Table Types: Transaction Example

Transaction_Fact_Table							
Transaction_id	store_FK	Customer_FK	Transaction_Date_FK	Product_FK	QTY	price	total_sales
101	1	1234	2000-01-03	34	3	20	60
102	1	1234	2000-01-03	70	5	31	155
103	1	1234	2000-01-03	3	2	22	44
104	1	2323	2000-01-03	5	2	17	34
105	1	2323	2000-01-03	3	1	22	22
106	1	2323	2000-01-03	70	4	30	120

Periodic Fact Table

- A periodic fact table contains one row for a group of transactions over a period OLTP.
- It must be from lower granularity to higher granularity hourly, daily, monthly, and quarterly, then yearly.

Accumulated Snapshot Fact Table

- An accumulating fact table stores one row for the entire process.
- It does not accumulate time it accumulates business process.
- A row in an accumulating snapshot fact table summarizes the measurement events occurring at predictable steps between the beginning and the end of a process
- Accumulating Fact tables are used to show the activity of progress through a well-defined process and are most often used to research the time between milestones.
- These fact tables are updated as the business process unfolds, and each milestone is completed.

Accumulated Snapshot Fact Table

- Accumulated Snapshot Use Cases.
- It also uses to measure the process performance life-cycle.
  - Order life-cycle
  - Hiring process.

Fact tables types comparison.

Feature	**Transaction**	**Periodic**	**Accumulating**
Grain	1 row/transaction	1 row/time-period	1 row/entire event stages
Date Dimension	Lowest granularity	End-of-period granularity	Multiple date
Facts	Transaction activities	Periodic activities	Defined lifetime activities
Size	Largest	Medium	Smallest
Update	No	No	Yes, after stage finished

Fact types

Fact types

- Each fact table includes facts and it has different types:
  - Additive facts.
  - Semi-additive facts.
  - Non-additive facts.
  - Derived facts.
  - Textual facts.
  - Factless fact.

Additive facts

- It is the most flexible and useful facts.
- Its measures can be summed across any of the dimensions associated with the fact table.

SalesFact

```
Date_FK
Store_FK
Product_FK
Sales_Amount(agg)
```

Semi-additive facts

- It can be added across some dimensions but not all also known as (partially-additive).

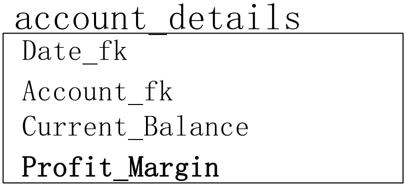
account_details

Date_fk
Account_fk
Current_Balance
Profit_Margin

- what's the total current balance for all accounts in the bank?
- What's the current balances for a given account for each day of the month does not give us any useful information?

Non-additive facts

- It can’t be added for any of the dimensions.
- Non-additive facts are usually the result of ratios (percentage) or other mathematical calculations.
- **Profit_Margin** is an example non-additive.



account_details

Date_fk

Account_fk

Current_Balance

Profit_Margin

Derived facts

- Derived facts are created by performing a mathematical calculation on a number of other facts, and are sometimes referred to as calculated facts.
- Derived facts may or may not be stored inside the fact table.
- **Total_sales = Qty_Sold * (Unit_price - Discount)**

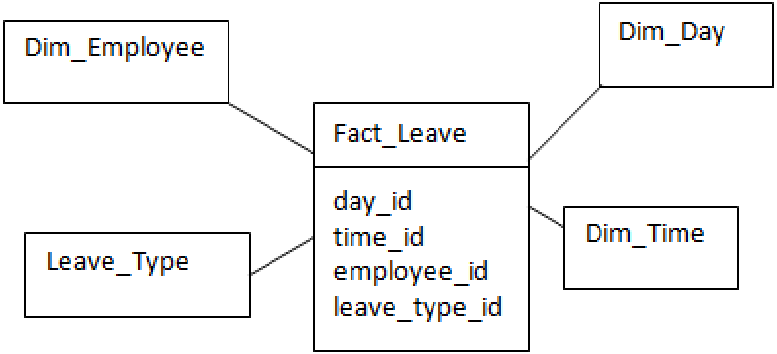
Order_Details
Order_id
Item_id
Order_date
Qty_Sold
Unit_price
Discount
Total_sales

Textual facts

- A textual fact consists of one or more characters such as flags and indicators.
- **It should be avoided in the fact table.**

Factless fact

- **A fact table with only foreign keys** and no facts is called a factless fact table.



Dimension Tables Types

Dimension Tables Types

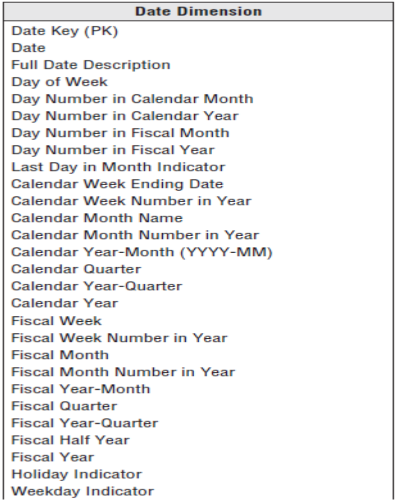
- **There are three types of Dimension tables:**

  - Date Dimension table.
  - Conformed Dimension table.
  - Slowly Changing Dimension table(SCD).

Dimension Table Types: Date Dimension Table

Dimension Table Details (Date Dimension Table)

- The date dimension is a special dimension because it is the one dimension nearly guaranteed to be in every dimensional model since virtually every business process captures a time series of performance metrics.
- Dimensional models always need an explicit date dimension table. There are many date attributes not supported by the SQL date function, including week numbers, fiscal periods, seasons, holidays, and weekends.
- Time is included in a Dimension or Fact ?



Date Dimension
Date Key (PK)
Date
Full Date Description
Day of Week
Day Number in Calendar Month
Day Number in Calendar Year
Day Number in Fiscal Month
Day Number in Fiscal Year
Last Day in Month Indicator
Calendar Week Ending Date
Calendar Week Number in Year
Calendar Month Name
Calendar Month Number in Year
Calendar Year-Month (YYYY-MM)
Calendar Quarter
Calendar Year-Quarter
Calendar Year
Fiscal Week
Fiscal Week Number in Year
Fiscal Month
Fiscal Month Number in Year
Fiscal Year-Month
Fiscal Quarter
Fiscal Year-Quarter
Fiscal Half Year
Fiscal Year
Holiday Indicator
Weekday Indicator

Dimensions Types: Conformed Dimension

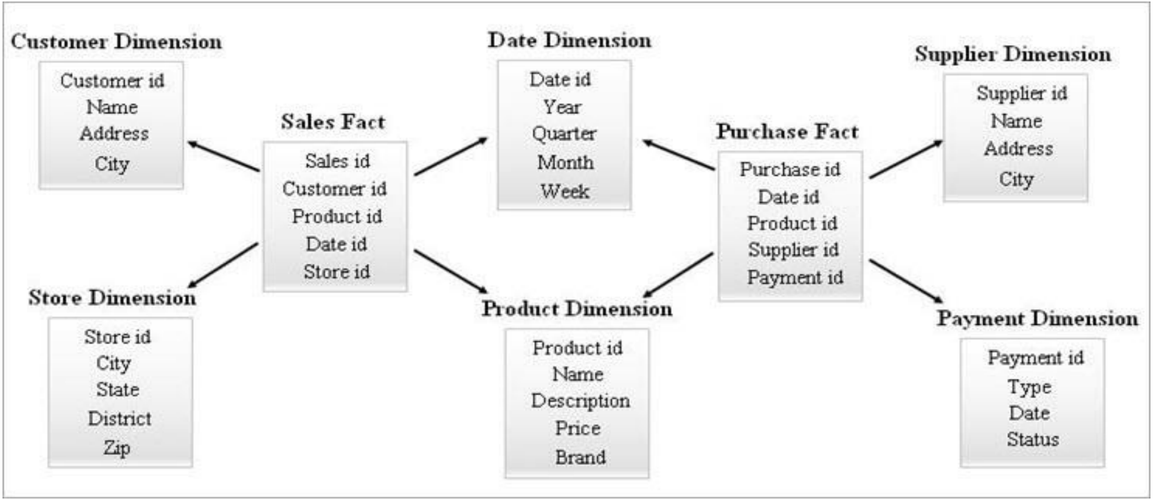
Conformed Dimensions

- Conformed Dimensions: the dimension which is ***identical*** and has the *same meaning* across many fact tables which it relates and used in different areas of the warehouse.

Example:

(Date as a Key): if we have a date column across many facts, we could use the date as key in all tables. So, it should be a unified format.

Conformed Dimensions



Dimensions Types: Slowly Changing Dimensions

Slowly Changing Dimensions

- Although data in fact table are normally static, data in dimension tables may be changed.
  - A customer may change address during 5 year-period
- Problem known as “Slowly Changing Dimensions” (Kimball, 2002)
- Common changing types:
  0: No change
  1: Overwrites value
  2: Adds new row
  3: Adds new column (update rows)

Slowly Changing Dimensions

Type-1 Change

- Replace old value in the dimension table with the new value
- Simple
- Cannot query old value

Product ID	Name	Brand	Serial No.
12345	TV 20”	National	ABC00-X

update

12345	TV 20”	Panasonic	ABC00-X

Slowly Changing Dimensions

Type-2 Change

- Add new row with the new information
- Surrogate keys required
- Allows tracking history (versioning)
- Used most often in DW

Product ID	Name	Brand	Serial No.
12345	TV 20”	National	ABC00-X
25984	TV 20”	Panasonic	ABC00-X

Inserted new row

Slowly Changing Dimensions

Type-3 Change

- Add new column that stores previous value
- Allow user to view new and previous values at the same time

Inserted new column

Product ID	Name	Brand	Previous Brand	Serial No.
12345	TV20”	Panasonic	National	ABC00-X

Schema Types

Schema Types

- Star Schema.
- Snowflake Schema.
- Fact constellation(Galaxy Schema)

Schema Types: Star Schema

Star Schema Example

- **Star Schema:** the most commonly used and the simplest style of dimensional modeling

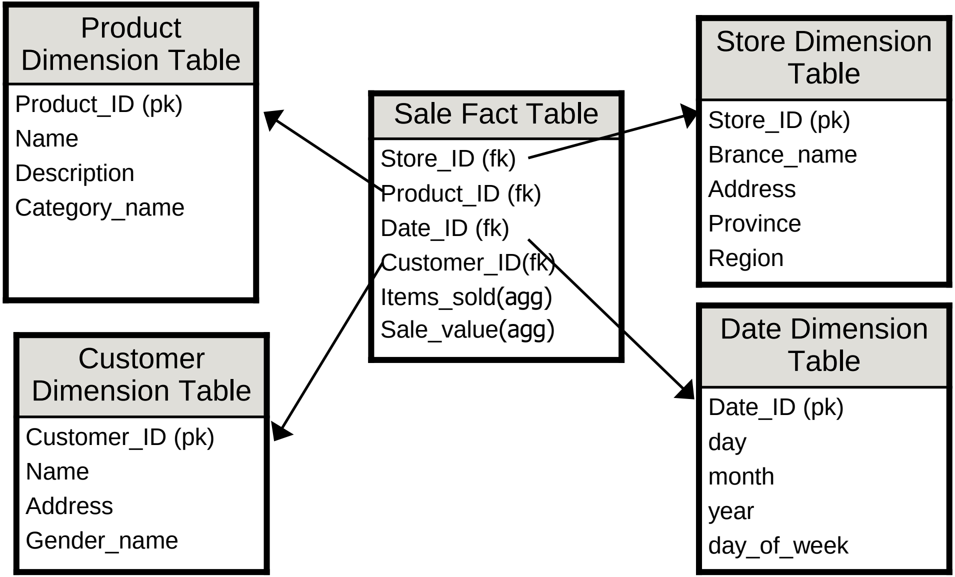
Star Schema Characteristics

- **Simplicity:** It is the simplest type of DWH schemas.
- **Query effectiveness:** Because of simplicity, It needs less join to query the data (It is optimized to query large dataset).
- **Data Redundancy and Large Table Size:** Due to
- de-normalization, it has a data redundancy, and the table size is huge.
- **Most** used and **widely** supported.

Star Schema Characteristics

- Dimensions represented by one one-dimension table.
- The dimension table are not joined to each other
- The fact table would contain key and measure.
- Data integrity is not enforced due to the de-normalized structure.

Star Schema Example



Product Dimension Table

Product_ID (pk)
Name
Description
Category_name

Sale Fact Table

Store_ID (fk)
Product_ID (fk)
Date_ID (fk)
Customer_ID(fk)
Items_sold(agg)
Sale_value(agg)

Store Dimension Table

Store_ID (pk)
Brance_name
Address
Province
Region

Customer Dimension Table

Customer_ID (pk)
Name
Address
Gender_name

Date Dimension Table

Date_ID (pk)
day
month
year
day_of_week

Schema Types: Snowflakes Schema

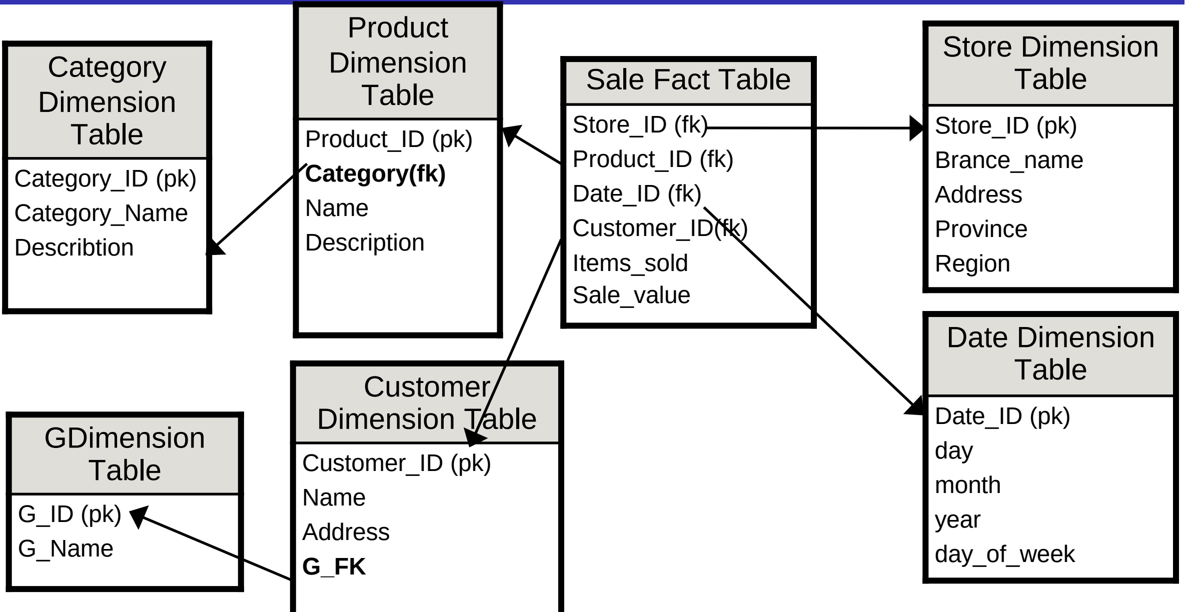
Snowflakes Schema

- **Snowflakes schema :** an extension of star schema where the diagram resembles a snowflake in shape

Snowflake Schema Characteristics

- **Extension:** Snowflake is an extension of the Star Schema.
- **Normalized:** Dimension tables are normalized; this means every dimension may expand into additional tables.
- **Disk Space Efficiency:** Due to its normalization methodology, it uses less desk space, which enhances the query as we scan less data size.
- **Complicated:** Due to the normalization query needs to join more table in some cases to get the data which reduces the performance.

Snowflakes Example



Category Dimension Table

Category_ID (pk)
Category_Name
Describtion

Product Dimension Table

Product_ID (pk)
Category(fk)
Name
Description

Sale Fact Table

Store_ID (fk)
Product_ID (fk)
Date_ID (fk)
Customer_ID(fk)
Items_sold
Sale_value

Store Dimension Table

Store_ID (pk)
Brance_name
Address
Province
Region

Date Dimension Table

Date_ID (pk)
day
month
year
day_of_week

GDimension Table

G_ID (pk)
G_Name

Customer Dimension Table

Customer_ID (pk)
Name
Address
G_FK

Star Vs. Snowflake Schema

Star Schema	**Snowflake Schema**
Dimension represented by one-table	Dimension tables are expanded into multi-tables
Fact table surrounded by dimension tables	Fact table surrounded by Hierarchy of dimension tables
Less join	Requires many joins
Simple Design	Very Complex Design
De-normalized Data structure	Normalized Data Structure
High level of Data redundancy	Very low-level data redundancy
Maintenance is difficult	Maintenance is easier
Cube processing is faster.	Cube processing might be slow because of the complex join.

Surrogate vs Natural Key

Entity Identification

- Each entity need to an identifier column.
- Identifier attribute for each instance (row) help to distinguish between the row or instance of the same entity.

Customer Table
Customer_PK(System)(SK)
Cus_Name
Cus_Address
Cus_Gender
Cus_National id (BK)
Cus_Phone (BK)

Surrogate vs Natural Key

Metrics	**Surrogate Key**	**Natural Key(BK)**
Uniqueness	Yes Guaranteed to be unique	Yes (In most Cases) Could change over time
Name	System generated	Business Key
Business Meaning	Doesn’t have a business meaning	Has a business meaning
Conceptual Relation	Doesn’t relate	Part of conceptual model
Creation	System(Database)	Set of column(s) from the data
Space	Extra Column added	No extra space
Maintenance	Easy for Maintenance	Difficult for Maintenance

Natural or Surrogate ???

It depend on several factors.

- The natural of the data.
- Database (DWH) platform.
- The group who uses this data.

Do we need to remove the natural kay to use surrogate key ?

- NO, we will keep both in the table and treat the surrogate key as primary key.