What is Data warehouse?

1. It is a database that is designed for querying and analysis rather than for transaction processing.
2. It separates analysis workload from transaction system.
3. This helps in:
    i. Maintaining historical records
    ii. Analyzing the data to gain a better understanding of the business and to improve the business.

What is Data warehouse?

4. Data warehouse is a **subject-oriented, integrated, time varying, non-volatile** collection of data in support of the management's decision-making process.

i. Subject Oriented: This is used to analyze particular subject area.

ii. Integrated: This shows that integrates data from different sources.

iii. Time variant: Historical data is usually maintained in a Data warehouse, i.e. retrieval can be for any period. In transactional system only the most recent/current data is maintained. But in the Data warehouse recent/current and the previous/historical data is maintained.

iv. Non-Volatile: Once the data is placed in the data warehouse, it cannot be changed, which means we will never be able to change the data.

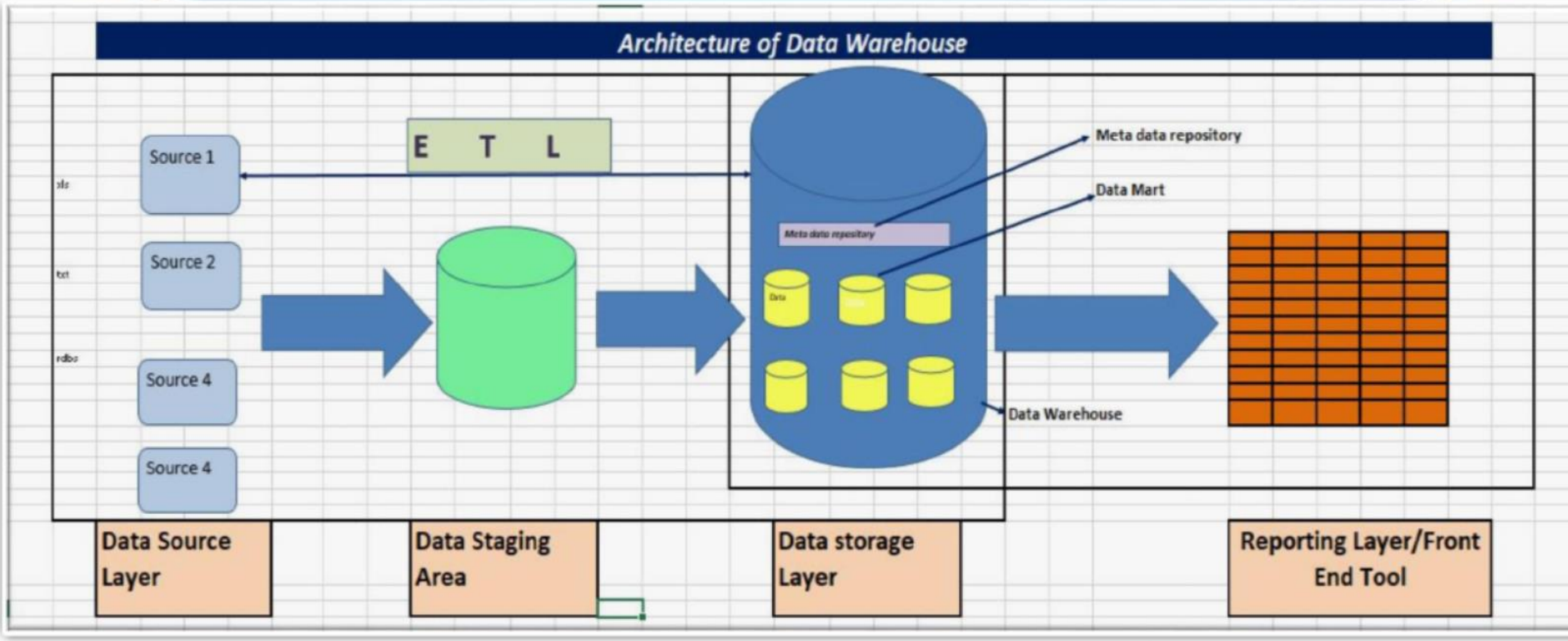
What is ETL ?

1. ETL stands for **Extract-Transform-Load**.

- **Extract** is the process of *reading data* from a source database/ transactional system.
- **Transform** is the process of *converting the extracted data* to required from.
- **Load** is the process of *writing the data* into the target database/ analytical system.

2. It is a process which defines how data is loaded from the source system to the target system (data warehouse).

3.Data Warehouse Architecture



Data Warehouse Architecture

There are four layers in DWH architecture:

- Data Source Layer
- Data Staging Area
- Data Storage Layer
- Reporting Layer

Data Warehouse Architecture

- First layer is the **Data Source layer**, which refers to various data stores in multiple formats like relational database, Flat Files, Excel files, Xml Files etc.
- These data stores business data like Sales, Customer, Finance, Product etc.
- After that the next step is **Extract**, where the required data from **data source layer** is extracted and put into the **data staging area**.
- **Data Staging area** is intermediate layer between Data Source Layer and Data Storage Layer used for processing data during the ETL process.

Data Warehouse Architecture

- Basically needs staging area to hold the data and to perform data transformations, before loading the data into warehouse.
- Actual **transformation transactional data into analytical data** is done in data staging area.
- And finally, we have the **Data Storage layer** i.e. data warehouse, the place where the successfully cleaned, integrated, transformed and ordered data is stored in a multi-dimensional environment. Now, the data is available for analysis and query purposes.
- **In reporting layer**, data in *data storage layer* is used to create various type of management reports from where user can take business decisions for planning, designing, forecasting etc.

Meta Data Repository

- Meta data is nothing but the data about data.
- Meta data repository is use to store meta data of data which is actually present in data warehouse i.e. Data storage layer

Data Mart

- **Data mart can be defined as the subset of data warehouse.**
- A data mart is focused on a single functional area e.g. product, customers, employees, sales etc.
- It is a subject-oriented database and is also known as High Performance Query Structures (HPQS).

➢ OLTP (Online Transaction Processing System):

1. OLTP is nothing but a database which actually stores the daily transactions which are created from one and more applications.
2. Data in OLTP is called as the **current data.**
3. Mostly normalized data is used in OLTP system.

➢ OLAP (Online Analytical Processing System) :

1. OLAP is use to store analytical data
2. It deals with analyzing the data for decision making and planning, designing etc.
3. Data in OLAP is called as the **Historical data.**
4. Mostly Denormalized data is used in OLAP system.

	OLTP	OLAP
Use	It is used for Transaction Processing.	It is used for Query Processing.
Data	· It holds current data.	· It holds current /historical data.
	· It stores all data.	· It stores only relevant data.
	· It has a small database.	· It has a large database.
	· It contains volatile data.(Create,Read, Update,Delete)	· It contains non-volatile data. (Read)
Source	It is the original source of data.	The data comes from various OLTP systems.
Purpose	To control and run fundamental business tasks.	To help with planning, problem solving, and decision support.
Queries	They are standardized and simple sql queries.	They are often complex sql queries.
Database design	It is highly normalized with many tables. (3NF)	It is de-normalized with fewer tables.
Users	It has many users.	It has few users.

What is Normalization ?

- Normalization is the process of efficiently organizing the data in the database.
- Normalization is used to minimize the redundancy. It is also used to eliminate the undesirable characteristics like Insertion, Updation and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.

Example

Candidate_Dts

Candidate_ID	Name	Qualification	City	State	Country
101	John	B.Sc, M.Sc	Houston	Texas	US
102	Ben	B.Sc	New York	Newyork	US
103	Ajay	B.Sc, M.Sc, PhD	Pune	Maharashtra	IND

1st Normal Form (1NF)

- It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.

Candidate_Dts

ID	Candidate_ID	Name	Qualification
1	101	John	B.Sc.
2	101	John	M.Sc
3	102	Ben	B.Sc
4	103	Ajay	B.Sc.
5	103	Ajay	M.Sc
6	103	Ajay	PhD

Second Normal Form (2NF)

For a table to be in the Second Normal Form,

- It should be in the First Normal form.
- And, it should not have Partial Dependency.

Second Normal Form (2NF)

Candidate_Dts

Candidate_ID	Name
101	John
102	Ben
103	Ajay

Qualification

Qualification_Id	Qualification
1	B.Sc
2	M.Sc
3	PhD

Candidate_Qualification_Dts

ID	Candidate_ID	Qualification_Id
1	101	1
2	101	2
3	102	3
4	103	1
5	103	2
6	103	3

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

- It is in the Second Normal form.
- And, it doesn't have Transitive Dependency

Third Normal Form (3NF)

Candidate_Dts

Candidate_ID	Name	City_ID
101	John	1
102	Ben	2
103	Ajay	3

City

City_ID	City_Name	State_Id
1	Texas	Texas
2	Newyork	Newyork
3	Maharashtra	Maharashtra

State

State_Id	State_Name	Country_Id
1	Texas	1
2	Newyork	1
3	Maharastra	3

Country

Country_Id	Country
1	US
2	IND

Normalization

Candidate_Dts

Candidate_ID	Name	City_ID
101	John	1
102	Ben	2
103	Ajay	3

City

City_ID	City_Name	State_Id
1	Texas	Texas
2	Newyork	Newyork
3	Maharastra	Maharastra

State

State_Id	State_Name	Country_Id
1	Texas	1
2	Newyork	1
3	Maharastra	3

Country

Country_Id	Country
1	US
2	IND

Qualification_Dts

Qualification_Id	Qualification
1	B.Sc
2	M.Sc
3	PhD

Candidate_Qualification_Dts

ID	Candidate_ID	Qualification_Id
1	101	1
2	101	2
3	102	3
4	103	1
5	103	2
6	103	3

Data models

- Data model tells how the logical structure of a database is modeled/designed.
- Data models define how data is connected to each other and how it will be processed and stored inside the system.
- Types of Data Models:
    i. Conceptual Data Model
    ii. Logical Data Model
    iii. Physical Data Model

Conceptual Data Model

- A conceptual data model is high level design of database.
  - Features of conceptual data model include:
    1. Displays the important entities and the relationships among them.
    2. No attribute is specified.
    3. No primary key is specified.

Logical Data Model

- Logical Data Model defines the data as much as possible, to show how they can be physically implemented in the database.

i. Includes all entities and relationships among them.

ii. All attributes/columns for each entity/table are specified.

iii. The primary key for each entity is specified.

iv. Foreign keys (keys identifying the relationship between different entities) are specified.

v. Constraints are defined. (Unique, Not null, Check, default etc..)

Physical Data Model

- Actual implementation of logical model into Database is called Physical Data Model.

What is Fact (Measures) ?

- It is counted or measured event.

What is Dimension?

- It contains referential information about fact.

What is Fact Table ?

- Fact table consist of measurements or facts of a business process.
- It is central table in dimension model surrounded by dimension tables.
- A fact table typically has two types of columns:
  i. Those that contain facts.
  ii. Those that are a foreign key to dimension tables.

What is Dimension Table?

- Dimension tables are used to describe dimensions.

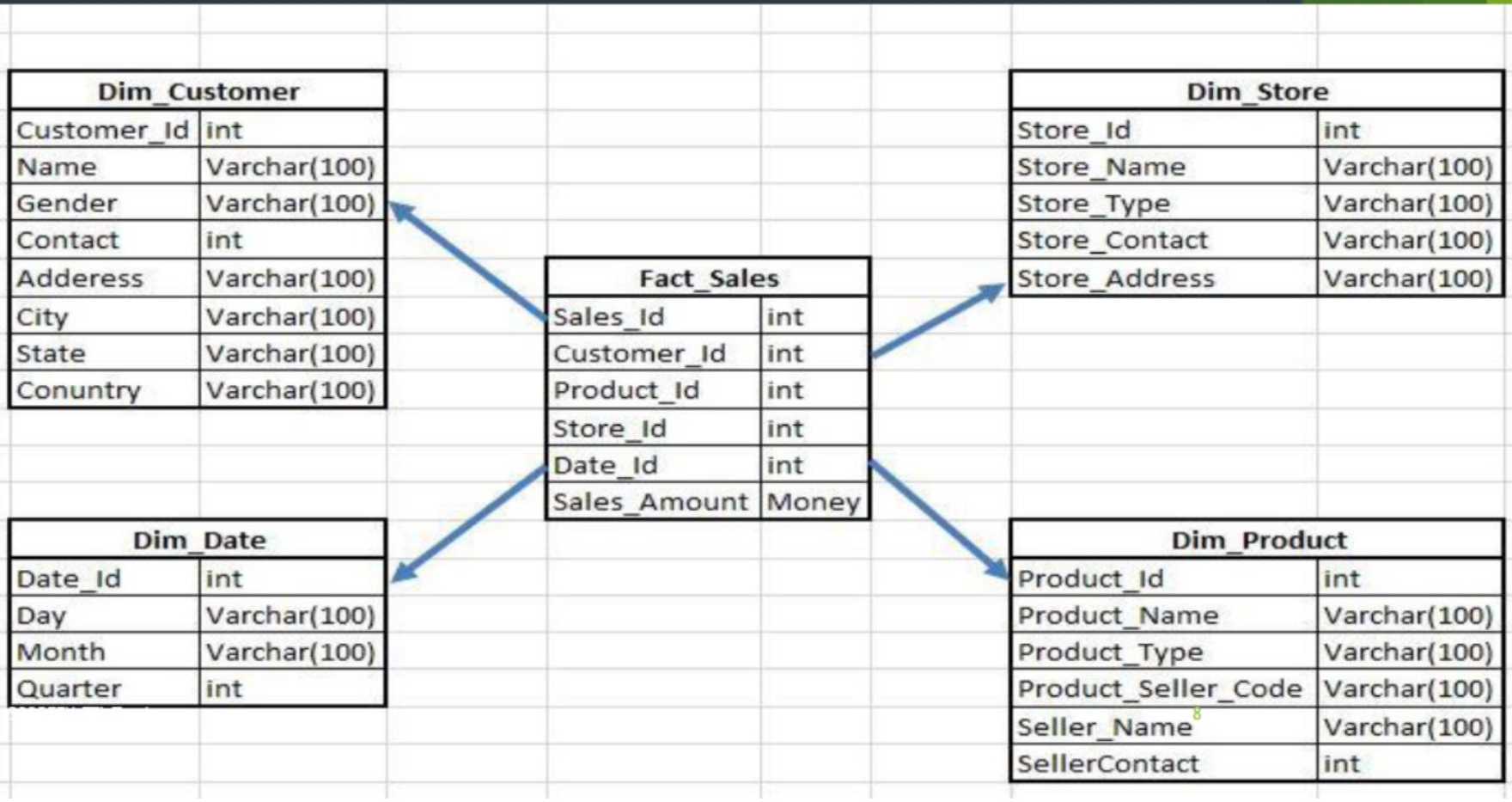
Type of Dimension model

1) Star Schema
2) Snowflake Schema
3) Galaxy or fact Constellation schema

1) Star Schema

1) It is simplest from of dimensional model
2) In Star schema design, central table is called fact table and radially connect other tables are called as dimension tables.
3) It is known as star schema because the entity-relationship diagram of this schemas look like a star.
4) Dimension tables in star schema are in *De-Normalized* form.
5) Star Schema is good for data marts with simple relationships.

Star Schema



Dim_Customer	
Customer_Id	int
Name	Varchar(100)
Gender	Varchar(100)
Contact	int
Adderess	Varchar(100)
City	Varchar(100)
State	Varchar(100)
Conuntry	Varchar(100)

Fact_Sales	
Sales_Id	int
Customer_Id	int
Product_Id	int
Store_Id	int
Date_Id	int
Sales_Amount	Money

Dim_Store	
Store_Id	int
Store_Name	Varchar(100)
Store_Type	Varchar(100)
Store_Contact	Varchar(100)
Store_Address	Varchar(100)

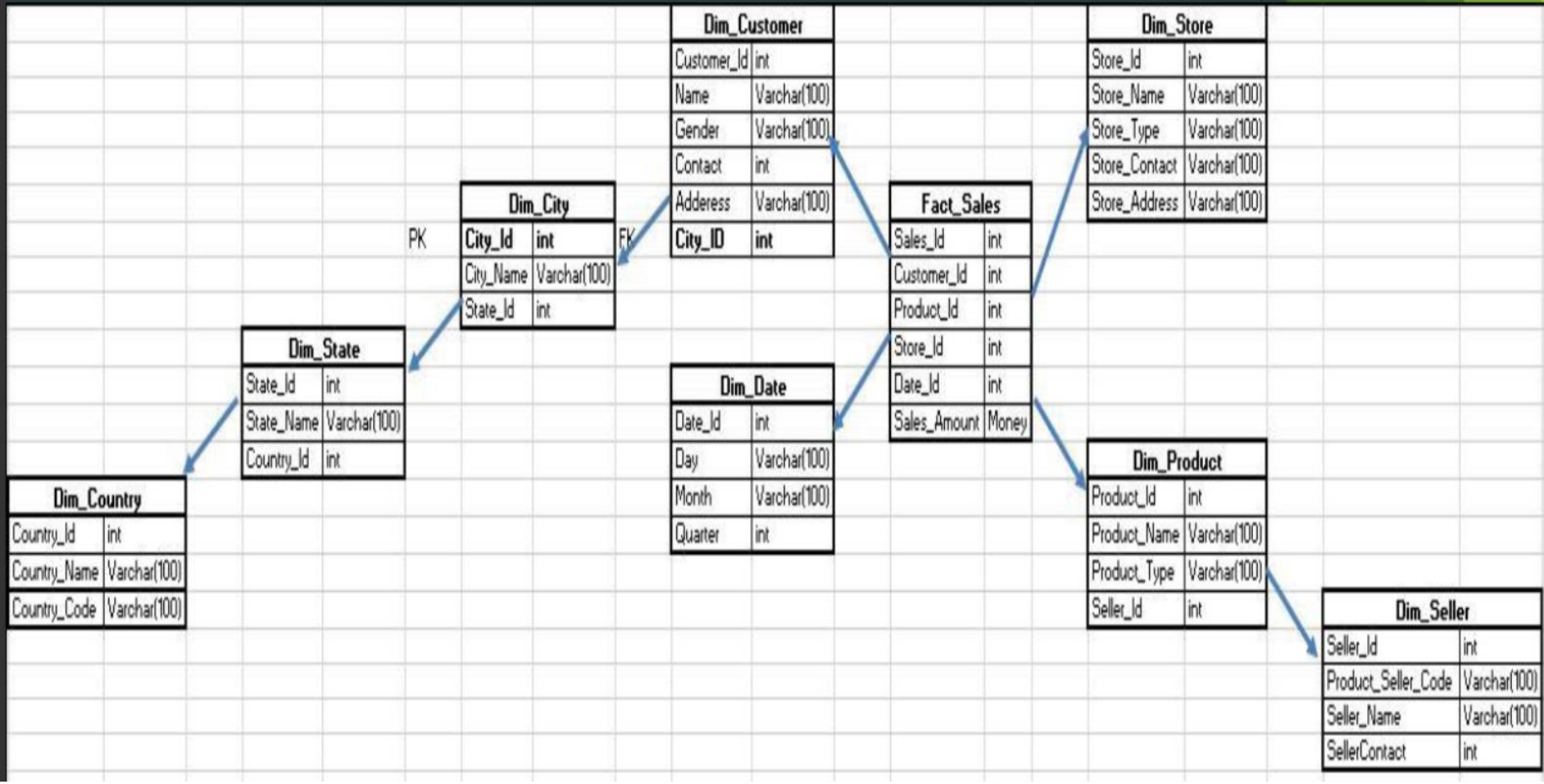
Dim_Date	
Date_Id	int
Day	Varchar(100)
Month	Varchar(100)
Quarter	int

Dim_Product	
Product_Id	int
Product_Name	Varchar(100)
Product_Type	Varchar(100)
Product_Seller_Code	Varchar(100)
Seller_Name	Varchar(100)
SellerContact	int

2) Snowflake Schema

1) The process of normalizing dimension tables is called snow flaking.
2) In Snowflake schema, Dimension Tables are in Normalized form.
3) Snowflake schema is a extension of star schema.
4) It's ER diagram look like a snowflake shape that's why is called as snowflake schema.

Snowflake Schema



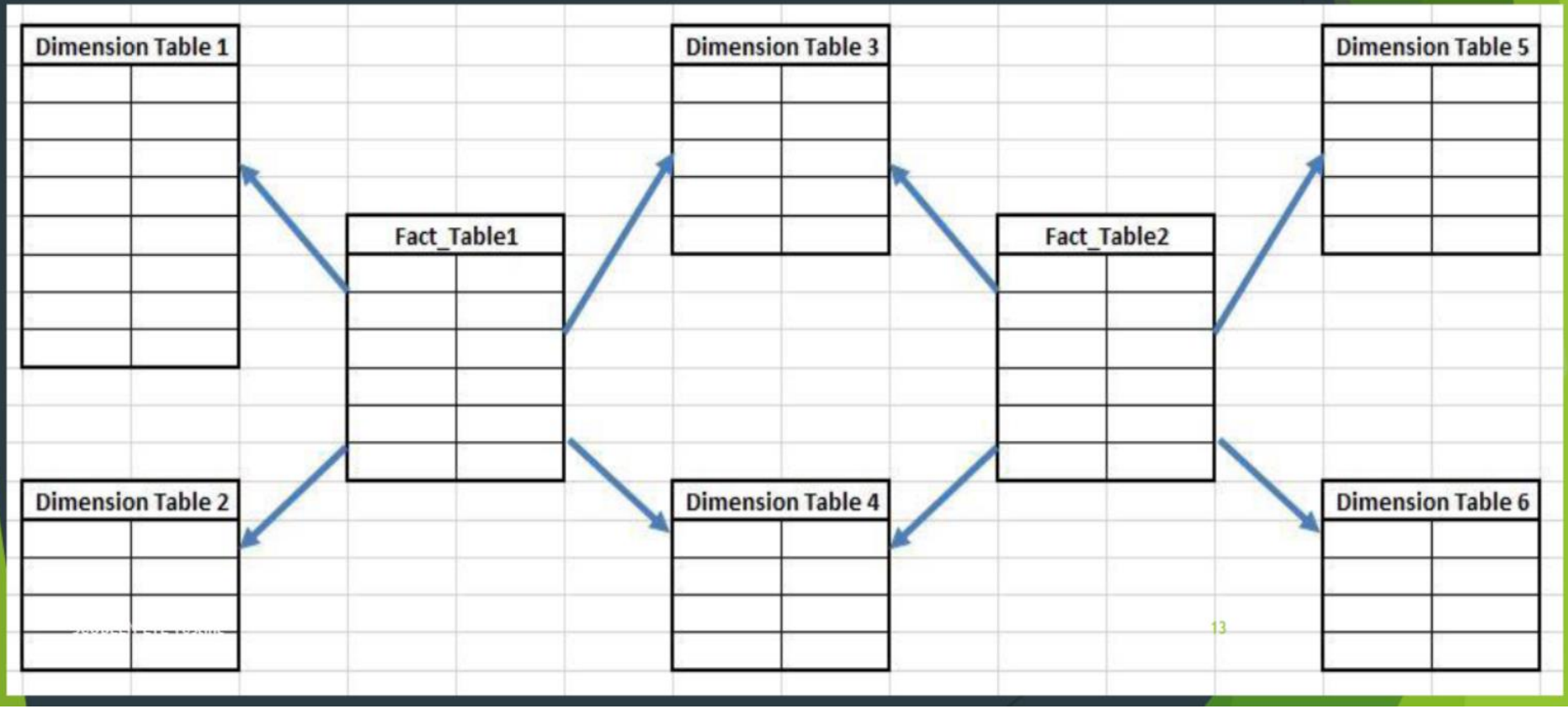
Star Schema vs. Snowflake Schema

	Star Schema	Snowflake Schema
When To Use	When dimension table contains less number of rows, we can choose Star schema.	When dimension table is relatively big in size, snowflaking is better as it reduces space.
	Good for data marts with simple relationships.	Good to use for data warehouse core to simplify complex relationships
Design	Simple DB Design.	Complex DB Design.
Ease of maintenance/change	less easy to maintain/change	easier to maintain and change.
Dimension table	A star schema contains **only single dimension table for each dimension.**	A snowflake **schema may have more than one dimension table for each dimension.**
Normalization/ De-Normalization	Both Dimension and Fact Tables are in De-Normalized form	Dimension Tables are in Normalized form but Fact Table is in De-Normalized form

3) Galaxy Schema

1) Galaxy Schema contains two and more fact tables that share same dimension tables between them.
2) It is also called Fact Constellation Schema.
3) The schema is viewed as a collection of stars hence the name Galaxy Schema.

3) Galaxy Schema



Type Of Facts

1. Additive Fact
2. Non-Additive Fact
3. Semi-Additive Fact

Addictive Fact

- Additive facts are facts that can be summed up through all of the dimensions in the fact table.

Sales_ID	OrderDateKey	ProductKey	CustomerKey	Qty	SalesAmount
101	20200517	24	112	1	1900
102	20200517	87	132	4	2500
103	20200516	65	112	3	3400
104	20200514	87	134	2	1250

Non-Addictive Fact

- Non-additive facts are facts that cannot be summed up for any of the dimensions present in the fact table.

ID	ProductKey	PurchaseAmount	SalesAmount	Profit_Loss
1	24	1100	1600	45.45
2	87	2500	2040	-18.4
3	65	1300	3400	161.54
4	87	1250	1250	0

Semi-Addictive Fact

- Semi-additive facts are facts that can be summed up for some of the dimensions in the fact table, but not with all.

Sales_ID	DateKey	ProductKey	Units_In	Units_Out	ClosingStock
101	20200517	24	112	1	1900
102	20200517	87	132	4	2500
103	20200516	65	112	3	3400
104	20200514	87	134	2	1250

Types Of Dimensions

1. **Slowly Changing Dimensions**
2. Conformed Dimensions
3. Degenerated Dimensions
4. Junk Dimensions

Slowly Changing Dimensions

Slowly Changing Dimensions :

- Dimensions that changes slowly over a period of time, rather than changing on regular schedule.
- A Slowly Changing Dimension (SCD) is a dimension that stores and manages both current and historical data over time in a data warehouse
- It is considered and implemented as one of the most critical ETL tasks in tracking the history of dimension records.

1. Slowly Changing Dimensions

There are many approaches how to deal with SCD. The most popular are:

- **Type 0** - The passive method
- **Type 1** - Overwriting the old value
- **Type 2** - Creating a new additional record
- **Type 3** - Adding a new column
- **Type 4** - Using historical table
- **Type 6** - Combine approaches of types 1,2,3 (1+2+3=6)

1. Slowly Changing Dimensions – Type 0

Type 0 - The passive method.

- In type 0, no special action is performed upon dimensional changes.
- Dimension data that remains same as it was first time inserted.

1. Slowly Changing Dimensions – Type 1

Type 1 - Overwriting the old value.

- In type 1, old value is simply overwritten by new value.
- Only new value is maintain.
- History of dimension changes is not kept in the database.
- This type is easy to maintain and is often use for data which changes are caused by processing corrections. (e.g. removal special characters, correcting spelling errors).

1. Slowly Changing Dimensions – Type 2

Type 2 - New row is created for new data.

- Old value and new value is present is same table.
- New row is created for new value in dimension table.
- In this method all history of dimension changes is kept in the database.

1. Slowly Changing Dimensions – Type 3

Type 3 - Adding a new column.

- In type 3, old and new value is kept in same table and same row.
- The new value is loaded into 'new' column and the old one into 'previous' column.
- History is limited to the number of columns which are created for storing historical data.
- This is the least commonly needed technique.

1. Slowly Changing Dimensions – Type 4

Type 4 -Using historical table.

- In Type 4, separate table are there for old value and new value
- Separate historical table is used to track all historical changes for each of the dimension.
- The 'main' table keeps only the New data (current data) .

  e.g. customer and customer history tables.

1. Slowly Changing Dimensions – Type 6

Type 6 - Combine approaches of types 1,2,3 (1+2+3=6). In this type we have additional columns in dimension table such as

- Current_Address, Current_Year : for keeping current value of the attribute.
- Previous_Address, Previous_Year : for keeping historical value of the attribute.
- Current_Flag : for keeping information about the most recent record.

Why database table needs Primary Key

1) For good practice in database designing, we have to maintain primary key in every table.
2) Primary key is use to uniquely identifies each row/record in a table.

Natural key

1. Primary key is made up of real data in table, that primary key is called as natural primary key.
2. For example,

   In HR database, Employees table having 'Employee_id' column which is unique and not null and we can call this real data because every employee should be identified by its' Employee_id.
3. Here we can make Employee_id as primary key in Employees table and this primary key is called as natural primary key.

Surrogate key

1. Some times in database table we cannot make primary key from real data.
2. In this situation, we have to add one artificial column in table which is unique and not null, and make this column as primary in table.
3. This primary which is generated from artificial column is called as Surrogate key.

Surrogate key

- Example:
  If we have to maintained history of employee table then Employee_id should be not the primary key column in table, because there are chances of duplicity in Employee_id column.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DESIGNATION	CREATED_ON
101	Daniel	Green	Jr. Software Engineer	03-01-18
102	Neena	Kochhar	Software Test Engineer	03-01-19
103	Bruce	Austin	Database Admin	06-05-20
101	Daniel	Green	Software Engineer	19-01-20
102	Neena	Kochhar	Sr. Software Test Engineer	03-02-20

Surrogate key

- To resolve this problem of primary key in employee table, we have to add column EMPLOYEE_KEY column which should be unique, not null .

- And we can make EMPLOYEE_KEY as primary key in employee table,
  as this primary is also called as surrogate key.

EMPLOYEE_KEY	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DESIGNATION	CREATED_ON
1	101	Daniel	Green	Jr. Software Engineer	03-01-18
2	102	Neena	Kochhar	Software Test Engineer	03-01-19
3	103	Bruce	Austin	Database Admin	06-05-20
4	101	Daniel	Green	Software Engineer	19-01-20
5	102	Neena	Kochhar	Sr. Software Test Engineer	03-02-20

Data Mapping Document

1. Data mapping document defines relationship between source data fields to their related target data fields which are involved in ETL process.

2. In simple terms, data mapping document is nothing but the map between source data and target data in ETL process.