Data Modeling, Star Schema, Snowflake Schema, Types of Facts, and Dimensions

---

1. Data Modeling

Definition:
Data modeling is the process of creating a visual representation of data and its relationships to facilitate database design and ensure data integrity, performance, and usability. It helps structure data logically and physically for storage and analysis.

Types of Data Models:

1. **Conceptual Data Model**:
    - High-level, abstract model focused on business requirements.
    - Defines entities, relationships, and attributes without technical details.
    - Example: Entities like Customer, Order, Product.
2. **Logical Data Model**:
    - Intermediate model that defines the structure of the data, including entity relationships, attributes, and data types, without focusing on the DBMS.
    - Example: Customer has attributes like CustomerID (Primary Key), Name, Address.
3. **Physical Data Model**:
    - Implementation-focused, includes DBMS-specific details like table structures, indexes, data types, and constraints.
    - Example: MySQL table creation with fields CustomerID INT AUTO_INCREMENT PRIMARY KEY.

---

2. Star Schema

Definition:
The star schema is a data warehouse schema design that consists of a central **fact table** connected to multiple **dimension tables**. It is optimized for analytical queries and decision-making processes.

Components:

1. **Fact Table**:
    - Central table that contains measurable, numeric data (facts).
    - Includes foreign keys referencing dimension tables.
    - Example: Sales_Fact table with fields SaleID, DateID, ProductID, CustomerID, and Revenue.
2. **Dimension Tables**:

- Surround the fact table and store descriptive, textual information about the facts.
- Example: Product_Dimension with fields ProductID, ProductName, Category.

Example Schema:

Fact_Sales	Dimension_Product	Dimension_Customer
SaleID (PK)	ProductID (PK)	CustomerID (PK)
DateID (FK)	ProductName	CustomerName
ProductID (FK)	ProductCategory	Region
CustomerID (FK)		Demographics
Revenue		

Advantages:

- Simplicity: Easy to understand and query.
- Performance: Optimized for read-heavy workloads with fewer joins.

Disadvantages:

- Storage: Denormalized structure leads to redundancy.

---

3. Snowflake Schema

Definition:
The snowflake schema is a normalized version of the star schema. Dimension tables are split into additional tables to reduce redundancy and storage requirements.

Components:

- Central **fact table** connected to normalized **dimension tables**.
- Example: Instead of storing ProductCategory in the Product_Dimension, create a separate Category_Dimension.

Example Schema:

Fact_Sales	Dimension_Product	Dimension_Category
SaleID (PK)	ProductID (PK)	CategoryID (PK)
DateID (FK)	ProductName	CategoryName
ProductID (FK)	CategoryID (FK)	
CustomerID (FK)		

Fact_Sales	Dimension_Product	Dimension_Category
Revenue		

Advantages:

- Reduces redundancy and storage requirements.
- Better suited for slowly changing dimensions.

Disadvantages:

- Increases query complexity due to more joins.

---

4. Types of Facts

Facts are numeric measures that represent business metrics.

Categories of Facts:

1. **Additive Facts**:
    - Can be summed across all dimensions.
    - Example: Sales_Amount can be totaled by date, region, or product.
2. **Semi-Additive Facts**:
    - Can be summed across some dimensions but not others.
    - Example: Account_Balance can be totaled by region but not over time.
3. **Non-Additive Facts**:
    - Cannot be summed across any dimension.
    - Example: Ratios or percentages like Profit_Margin.

---

5. Dimensions

Definition:
Dimensions provide the descriptive context for facts, enabling users to analyze and filter data from various perspectives.

Characteristics:

- Textual and categorical in nature.
- Connected to the fact table via foreign keys.

Examples of Dimensions:

1. **Time Dimension**:
    - Attributes: Date, Week, Month, Year.

2. **Product Dimension**:
    - Attributes: ProductName, Category, Price.
3. **Customer Dimension**:
    - Attributes: CustomerName, Region, Age.

Types of Dimensions:

1. **Conformed Dimensions**:
    - Shared across multiple fact tables or data marts.
    - Example: Time_Dimension used in both Sales_Fact and Inventory_Fact.
2. **Junk Dimensions**:
    - Combines unrelated attributes into a single dimension to reduce clutter.
    - Example: Flag_Dimension for binary indicators like NewCustomer, PromotionalSale.
3. **Degenerate Dimensions**:
    - Dimension data stored in the fact table itself.
    - Example: OrderID in a sales fact table.
4. **Role-Playing Dimensions**:
    - A single dimension table used in different contexts.
    - Example: Time_Dimension used as Order_Date and Ship_Date.
5. **Slowly Changing Dimensions (SCDs)**
   Slowly Changing Dimensions (SCDs) are a methodology for handling changes in dimension data over time in a data warehouse while preserving the history of changes where required.

Types of SCDs

1. SCD Type 0 (Fixed Dimensions)

- **Definition**: The dimension data is static and does not change over time.
- **Use Case**: For attributes like Product Launch Date or Social Security Number that must remain constant.

---

2. SCD Type 1 (Overwrite)

- **Definition**: When a change occurs, the old data is overwritten with the new data, and no history is maintained.
- **Characteristics**:

- Simplest and fastest to implement.
- Suitable for data where historical accuracy is not required.

Example:

CustomerID	CustomerName	Region
101	John Smith	North
101	John Smith	South

---

3. SCD Type 2 (Versioning)

- **Definition**: Maintains full history by creating a new record for each change in dimension data.
- **Characteristics**:
  - Each record is time-stamped or flagged as active/inactive.
  - Ensures complete historical tracking.

Implementation Options:

1. **Row Versioning**: Add a Version column to identify different versions of the same dimension.
2. **Date Range**: Add StartDate and EndDate columns to define the validity period.

Example:

CustomerID	CustomerName	Region	StartDate	EndDate	CurrentFlag
101	John Smith	North	2023-01-01	2023-06-30	0
101	John Smith	South	2023-07-01	NULL	1

---

4. SCD Type 3 (Tracking Limited History)

- **Definition**: Maintains limited history by adding columns to store previous values alongside the current value.
- **Characteristics**:
  - Useful when only a small history is needed (e.g., the last two changes).
  - Adds minimal complexity but sacrifices complete historical tracking.

Example:

CustomerID	CustomerName	CurrentRegion	PreviousRegion
101	John Smith	South	North

---

5. SCD Type 4 (History Table)

- **Definition**: Maintains history in a separate table, while the main dimension table holds only the current data.
- **Characteristics**:
    - Reduces the size of the main dimension table.
    - Separate history table is queried only when historical data is required.

Example:

Dimension Table (Current):

CustomerID	CustomerName	Region
101	John Smith	South

History Table:

CustomerID	CustomerName	Region	StartDate	EndDate
101	John Smith	North	2023-01-01	2023-06-30

---

6. SCD Type 6 (Hybrid SCD - 1+2+3)

- **Definition**: Combines elements of SCD Types 1, 2, and 3 to track both historical and current data while maintaining versioning.
- **Characteristics**:
    - Adds columns for current and previous values (Type 3).
    - Maintains history in rows (Type 2).
    - Overwrites non-essential fields (Type 1).

Example:

CustomerID	CustomerName	CurrentRegion	PreviousRegion	Version	StartDate	EndDate
101	John Smith	South	North	2	2023-07-01	NULL
101	John Smith	North	NULL	1	2023-01-01	2023-06-30

Choosing the Right SCD Type

SCD Type	When to Use
Type 0	When the data is immutable and never changes.
Type 1	When historical data is irrelevant or unnecessary.
Type 2	When complete historical tracking is critical for analysis and reporting.
Type 3	When only a limited history of changes is required.
Type 4	When maintaining a clean, smaller main table while preserving history is needed.
Type 6	When a combination of history tracking and current data is required.