**Data Modeling**

Data modeling is the process of creating a visual representation of an information system to illustrate the types of data used, their relationships, and how they can be grouped and organized.

**Types of Data Models**

1. **Conceptual Data Models**: These offer a high-level view of the system, focusing on the main entities and their relationships. They are used to gather initial project requirements and define business rules.
2. **Logical Data Models**: These provide more detail about the data structures, including attributes and relationships, without specifying technical system requirements. They are useful for data-oriented projects like data warehouse design.
3. **Physical Data Models**: These describe how the data will be physically stored in a database, including tables, columns, keys, and constraints. They are the least abstract and are used to implement the database design.

**Data Modeling Process**

1. Identify Entities: Determine the things, events, or concepts represented in the data set.
2. Identify Key Properties: Define the unique attributes of each entity.
3. Identify Relationships: Specify the nature of the relationships between entities.
4. Map Attributes to Entities: Ensure the model reflects how the business will use the data.
5. Assign Keys and Normalize: Use keys to represent relationships and reduce redundancy.
6. Finalize and Validate: Iterate and refine the model as business needs change.

**Types of Data Modeling**

* Hierarchical Models: Represent one-to-many relationships in a tree-like format.
* Relational Models: Use tables to represent data and their relationships, often employing SQL for data management.
* Entity-Relationship (ER) Models: Use formal diagrams to represent relationships between entities.
* Object-Oriented Models: Represent data as objects, similar to object-oriented programming.
* Dimensional Models: Optimize data retrieval speeds for analytic purposes, often used in data warehouses.

**Benefits of Data Modeling**

Data modeling offers several benefits:

* Reduces Errors: Minimizes errors in software and database development.
* Increases Consistency: Ensures consistency in documentation and system design.
* Improves Performance: Enhances application and database performance.
* Facilitates Communication: Improves communication between developers and business intelligence teams.
* Speeds Design Process: Eases and speeds the process of database design at various levels.

**Additive Measures**

* Definition: Measures that can be summed across all dimensions in a data model.
* Examples: Sales revenue, total units sold, and total cost.
* Usage: These measures can be aggregated across any dimension, such as time, geography, or product categories.

**Semi-Additive Measures**

* Definition: Measures that can be summed across some dimensions but not all.
* Examples: Inventory levels, account balances, and headcount.
* Usage: These measures can be aggregated across some dimensions (e.g., geography or product categories) but not across others (e.g., time). For instance, summing inventory levels over time doesn't make sense because it would result in an inflated number.

**Non-Additive Measures**

* Definition: Measures that cannot be summed across any dimension.
* Examples: Ratios, percentages, and averages.
* Usage: These measures require specific aggregation methods, such as averaging or calculating ratios, rather than simple summation.

**Dimensional Modeling**

**Dimensional Modeling (DM)** is a data structure technique optimized for data storage in a Data warehouse. The purpose of dimensional modeling is to optimize the database for faster retrieval of data.

A dimensional model in data warehouse is designed to read, summarize, analyze numeric information like values, balances, counts, weights, etc. in a data warehouse. In contrast, relation models are optimized for addition, updating and deletion of data in a real-time Online Transaction System.

**Elements of Dimensional Data Model**

**Fact**

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

**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.

**Attributes**

The Attributes are the various characteristics of the dimension in dimensional data modeling.

In the Location dimension, the attributes can be

* State
* Country
* Zipcode etc.

Attributes are used to search, filter, or classify facts. Dimension Tables contain Attributes

**Fact Table**

A fact table is a primary table in dimension modelling.

A Fact Table contains

1. Measurements/facts
2. Foreign key to dimension table

**Dimension Table**

* Dimensions offers descriptive characteristics of the facts with the help of their attributes
* They are joined to fact table via a foreign key.
* Dimension tables are de-normalized tables.
* No set limit set for given for number of dimensions
* The dimension can also contain one or more hierarchical relationships

**Steps of Dimensional Modelling**

The accuracy in creating your Dimensional modeling determines the success of your data warehouse implementation. Here are the steps to create Dimension Model

1. Identify Business Process
2. Identify Grain (level of detail)
3. Identify Dimensions
4. Identify Facts

The model should describe the Why, How much, When/Where/Who and What of your business process

**Step 1) Identify the Business Process**

Identifying the actual business process a data warehouse should cover. This could be Marketing, Sales, HR, etc. as per the [data analysis](https://www.guru99.com/what-is-data-analysis.html) needs of the organization. The selection of the Business process also depends on the quality of data available for that process. It is the most important step of the Data Modelling process, and a failure here would have cascading and irreparable defects.

**Step 2) Identify the Grain**

The Grain describes the level of detail for the business problem/solution. It is the process of identifying the lowest level of information for any table in your data warehouse. If a table contains sales data for every day, then it should be daily granularity. If a table contains total sales data for each month, then it has monthly granularity.

**Step 3) Identify the Dimensions**

Dimensions are nouns like date, store, inventory, etc. These dimensions are where all the data should be stored. For example, the date dimension may contain data like a year, month and weekday.

Dimensions: Product, Location and Time

Attributes: For Product: Product key (Foreign Key), Name, Type, Specifications

Hierarchies: For Location: Country, State, City, Street Address, Name

**Step 4) Identify the Fact**

This step is co-associated with the business users of the system because this is where they get access to data stored in the data warehouse. Most of the fact table rows are numerical values like price or cost per unit, etc.

**Step 5) Build Schema**

In this step, you implement the Dimension Model. A schema is nothing but the database structure (arrangement of tables). There are two popular schemas

1. **Star Schema**

The star schema architecture is easy to design. It is called a star schema because diagram resembles a star, with points radiating from a center. The center of the star consists of the fact table, and the point of the star is dimension tables.

The fact tables in a star schema which is third normal form whereas dimensional tables are de-normalized.

1. **Snowflake Schema**

The snowflake schema is an extension of the star schema. In a snowflake schema, each dimension are normalized and connected to more dimension tables.

**Rules for Dimensional Modelling**

* Load atomic data into dimensional structures.
* Build dimensional models around business processes.
* Need to ensure that every fact table has an associated date dimension table.
* Ensure that all facts in a single fact table are at the same grain or level of detail.
* It’s essential to store report labels and filter domain values in dimension tables
* Need to ensure that dimension tables use a surrogate key
* Continuously balance requirements and realities to deliver business solution to support their decision-making.

**Conformed Dimension**

Conformed dimensions are a cornerstone of seamless data integration and consistent reporting, providing the foundation for reliable and unified insights in [data warehousing](https://www.acceldata.io/blog/data-warehouse-concepts-benefits-and-emerging-trends).

Conformed dimension is a dimension table shared across multiple fact tables in a database, ensuring that attributes such as "Customer ID," "Product Category," or "Region" are universally consistent. This consistency allows businesses to compare data across various processes without discrepancies.

**Primary characteristics of conformed dimensions**

1. **Standardized attributes**: Every shared attribute has a consistent definition, format, and meaning across datasets.
2. **Universal accessibility**: These are accessible to all fact tables in a data warehouse, ensuring uniformity in analysis.
3. **Support for multiple schemas**: Conformed dimensions are essential in data models such as the star schema, where a central fact table interacts with multiple dimensions.

**Key Benefits of Conformed Dimensions**

**1. Consistency in reporting**

Conformed dimensions ensure that key data points, such as customer demographics or product categories, are defined uniformly across all fact tables. This eliminates discrepancies, enabling accurate comparisons and insights.

**2. Simplified data integration**

Businesses can integrate data from disparate sources into a single, cohesive data model by using shared dimensions.

**3. Improved scalability and maintenance**

Conformed dimensions support the evolution of data models as organizations grow.

**4. Enhanced decision-making**

Uniformity in dimensions translates to reliable analytics and business intelligence. Decision-makers can confidently rely on accurate, aggregated data to guide strategic initiatives

**Absolute URLs**

An absolute URL provides the complete address of a webpage or file on the internet, including the protocol "**http:// or https://**" domain, and path to the resource.

eg. https://www.example.com/images/logo.png

**Relative URLs**

A relative URL specifies the path to a resource about the current document's path or the base URL of the website, without the domain name and [protocol](https://www.geeksforgeeks.org/types-of-network-protocols-and-their-uses/).

Example: If the current URL is "**https://www.example.com/about/team.html"**, a relative URL to another page in the same directory like **contact.html** would simply be **contact.html**.