Tanmay Maheshwari — Published On February 28, 2023

Advanced Data Engineering Data Warehouse Database Technique

Introduction

One of the most important assets of any organization is the data it produces on a daily basis. This data is used by an organization to find valuable insights which help in improving an organization's growth and strategies and give them an upper hand over its competitors. This article explains to you the idea of Dimensional Modeling as part of Data Warehousing with different steps involved.

Learning Objective

- Understand Data Warehousing
- Understand the difference between Normalised and Denormalised form
- Learn Dimensional Modeling and its implementation using real-world application

But this data, both current and historical, can only be utilized to its full potential when it is easily accessible and available. So now the question arises of how we can store this data in such a way that it fulfills the above requirements.

Here comes the concept of Data Warehousing.

Table of Contents

Data Warehousing

Data warehousing is a technique in which information is stored in a central repository that can be used by business analysts, data engineers, data scientists, and decision-makers through business intelligence (BI) tools to make decisive decisions.

It is used to handle large amounts of data to understand relationships and trends across the data.

Advantages of Data Warehousing –

1. It makes information easily accessible.
2. It presents information consistently.

Difference Between Normalized and Denormalized Form

In normalized form, data is stored in multiple tables, reducing data redundancy and inconsistency, thus achieving data integrity. In the denormalized form, data is stored in a limited number of tables (maybe a single table) to reduce querying time.

Both of them contain joined tables, but the key difference between them is the degree of normalization. As the degree of normalization increases, the complexity of the model increases, and as the complexity of the model increases, the time to retrieve data also increases.

Dimensional Modeling

The data model used to store data in the denormalized form is called Dimensional Modeling. It is the technique of storing data in a Data Warehouse in such a way that enables fast query performance and easy access to its business users. It involves creating a set of dimensional tables that are designed to support business intelligence and reporting needs.

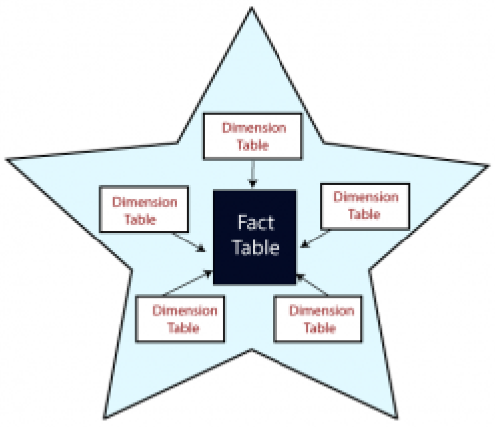
The goal of dimensional modeling is to provide a simple and intuitive way to access and analyze data, making it easy for business users to understand and use it. It aims at making simple data models. When the data models are as simple as possible, they can be understood easily, allowing the software to navigate and deliver results quickly and efficiently.

The core concept of dimensional modeling is the creation of a star schema. It is called so as the tables are arranged in the form of a star. Dimensional modeling includes facts and dimensions. Let's have a basic idea of what Facts and Dimensions are.

Fact tables contain measures or numerical data associated with a business process, like the number of products sold. In contrast, dimensional tables store the description or textual information related to the business process, like who bought the products. We will discuss facts and dimensions in detail later in this article.

A dimensional model represents the different business processes of an organization. A fact table with its dimension table is a single business process.

Each dimensional model consists of many fact tables, with each fact table joined with corresponding dimension tables. A fact table is connected to another fact table via a common dimensional table between them; this common dimensional table is called a bridge table. We can even connect a fact table with a fact table directly, but it is not a wise option as it makes the model complex and difficult to understand.



Star Schema

A dimensional model contains the same information as a normalized model. Still, the data in a dimensional model is packed in such a way that delivers user understandability, query performance, and resilience to change.

Implementation of Dimensional Modeling in Real World

Dimensional Modeling can be best understood by implementing it in real-world cases. We will pick up the E-Commerce industry like Myntra, Flipkart, Amazon, etc., as it is familiar to everyone. We will be creating a dimensional model depicting different business processes that take place in an E-Commerce industry.

Four Steps of Designing a Dimensional Model

Step 1: Select the Business Process

The first step involves selecting the business process, and it should be an action resulting in output.

Business Process #1:The e-Commere industry is widely known for selling and buying goods over the internet, so our first business process will be the products bought by the customers.

Business Process #2: Delivery status is also one of the most important business processes in this industry. It tells us where the product is currently from. It's dispatched from the warehouse to the customer's given address.

Business Process #3: Maintaining the inventory in order to ensure that items don't run out of stock, how sales are going on etc.

Step 2: Decide the Grain of each Business Process

A **grain** is a business process at a specified level. It tells us what exactly a row, in fact, a table, represents. All the rows in a fact table should result from the same grain. Each fact table is the result of a different grain selected in a business process. The grain should be as granular (at the lowest level) as possible.

Grains for the above business processes are

Grain 1: We can have the grain as the products purchased by the customer, i.e., each row of the fact table will represent all the products checked out by the customer from the cart but suppose a customer ordered 100 products, so this will be represented as a single row. Imagine how complex it will become to query such data, so we must choose a grain as granular as possible. Therefore, our grain will be an individual product ordered by a customer, i.e., one product per row. This will make the data simple and easy to query.

Similarly, we will select the most granular grains for the remaining processes.

Grain 2: Here also, the grain will be the status of an individual product shipped from the warehouse to the delivery location.

Grain 3: Here, each row will represent the daily inventory for each product in each store., it will tell the stock of that product left in the inventory and how many products have already been sold.

Step 3: Identify the Dimensions for the Dimensional Table

Before identifying the dimensions we will understand what a dimensional table is.

Dimensional Tables

These are the tables that are joined to fact tables. It describes the "who, what, where, when, how, and why" associated with the business event. It contains the descriptive attributes used for grouping and filtering the facts.

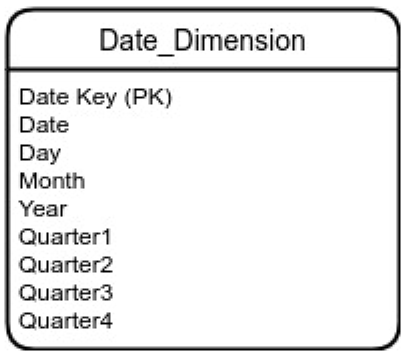
Some important points regarding Dimension Tables:

4. Each dimension table has a primary key that is joined to its given fact table.

5. Dimension attributes are the primary source of query constraints, grouping, and filtering.

Dimensions describe the measurements of the fact table. For example, customer id is a measurement, but we can describe its attributes further, more as what is the name of the customer, the address of the customer, gender, etc.
Our dimensional model will have the following dimensions:

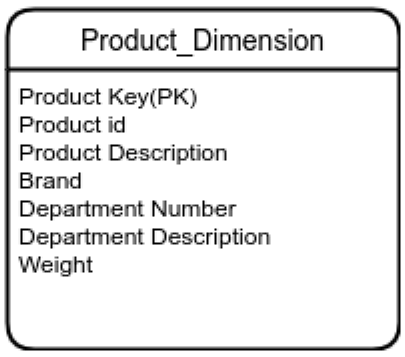
Date Dimension: This dimension table is used in almost every dimensional model as it helps monitor the business's performance with time.



Date_Dimension

Date Key (PK)
Date
Day
Month
Year
Quarter1
Quarter2
Quarter3
Quarter4

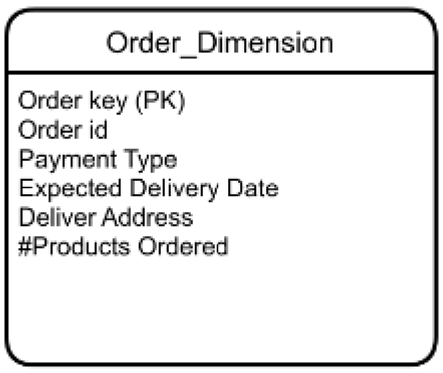
Product Dimension: This table will contain information regarding the product ordered.



Product_Dimension

Product Key(PK)
Product id
Product Description
Brand
Department Number
Department Description
Weight

Order Dimension: This detail will contain information regarding the order.

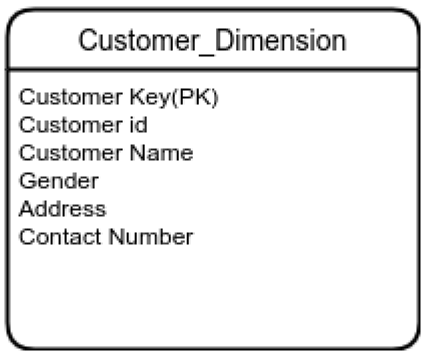


Order_Dimension

Order key (PK)
Order id
Payment Type
Expected Delivery Date
Deliver Address
#Products Ordered

Customer Dimension: This dimension table will contain the customer's information

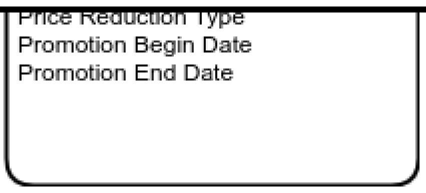
.



Customer_Dimension

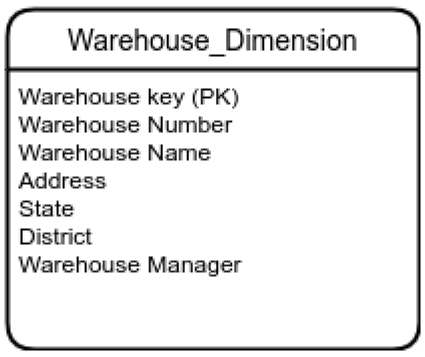
Customer Key(PK)
Customer id
Customer Name
Gender
Address
Contact Number

Promotion Dimension: This table covers the promotion condition under which the product was sold. The promotion conditions include temporary sales, reduction in price, discounts, etc.



Price Reduction Type
Promotion Begin Date
Promotion End Date

Warehouse Dimension: This table will contain information about the different warehouses located across the country.



Warehouse_Dimension

Warehouse key (PK)
Warehouse Number
Warehouse Name
Address
State
District
Warehouse Manager

Step 4: Identify the Facts for the Dimensional Table

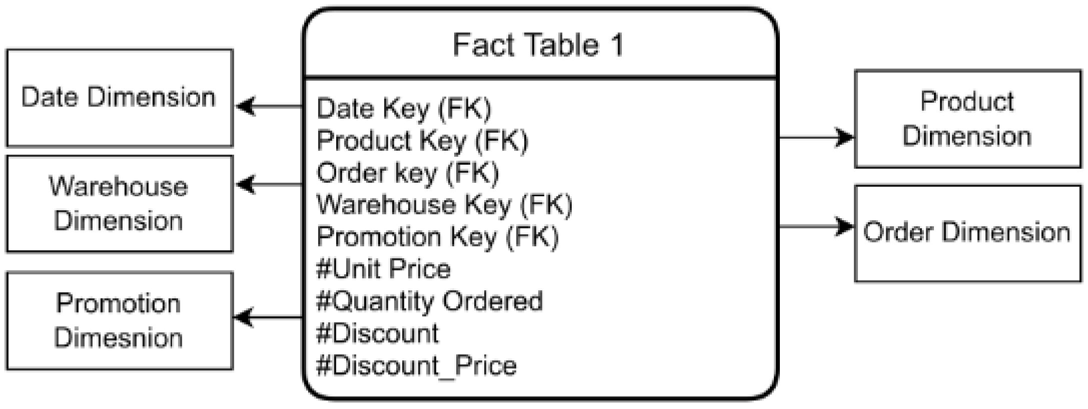
This is the final step in which we have to decide which facts (measurements) must be included in the fact table, but before that, let's discuss what a fact table is.

Fact Table

The term fact represents a business measure; therefore, a fact table in dimensional modeling stores the performance measurements resulting from a business process. These performance measurements measure the business, i.e., these are the metrics through which we can infer whether our business is in profit or loss. Different business measurements can be unit price, number of goods sold, etc. Each row in a fact table is a business event that results in measurements, and each fact table represents a business process in the organization. Now the event depends upon the grain we select. The selection of grain plays a vital role in the success of our dimensional model as it helps in selecting the measurements in the fact table to which further dimension tables are joined.
Since we have chosen three business processes, we will have three fact tables, but sometimes we get confused about whether an attribute should be added to the fact table or dimension table. To avoid that confusion, we will be using the following points to identify whether an attribute is a fact or dimension:

1. Textual data is generally stored in dimension tables where, whereas numeric data is generally stored in the fact table.
2. Continuous valued numeric values are stored in the fact tables, whereas discrete numeric values are stored in the dimension table.
3. The values that constantly change are kept in the fact table, whereas values that remain static or change very less with time are kept in the dimensional table.

Fact Table 1:



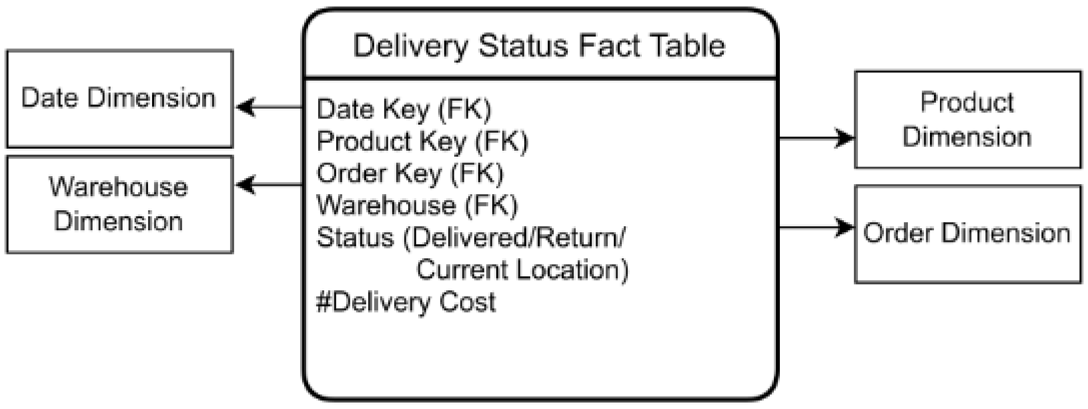
Fact Table 1

Date Key (FK)
Product Key (FK)
Order key (FK)
Warehouse Key (FK)
Promotion Key (FK)
#Unit Price
#Quantity Ordered
#Discount
#Discount_Price

So we will select the measurements corresponding to this grain. When we check out, the measurements that come are unit price, quantity, ordered, discount, etc., so we have added these measurements.

In this way, we add our measurements to a fact table. Weight also could have been added as a measurement, but its value remains constant therefore, we will keep it in the dimension table.

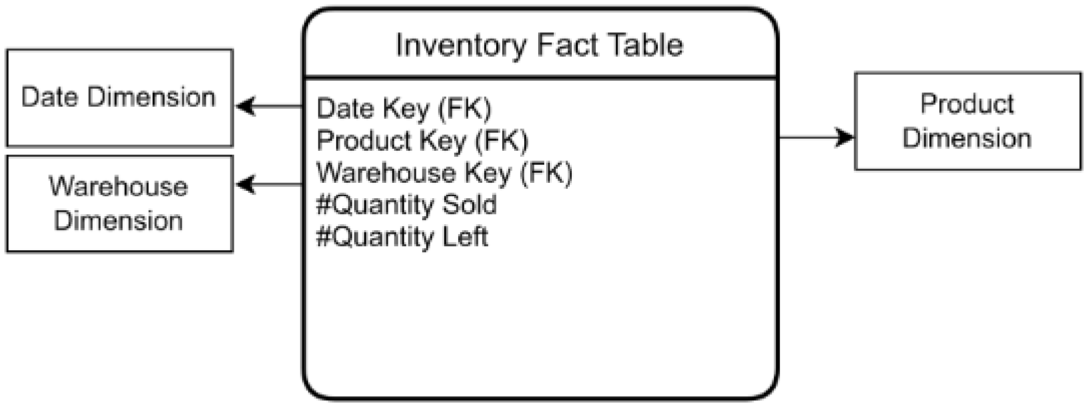
Fact Table 2:



Grain: Delivery Status of individual products in the order

This fact table will tell us the delivery status of the product, i.e., the location of the product it got delivered, etc.

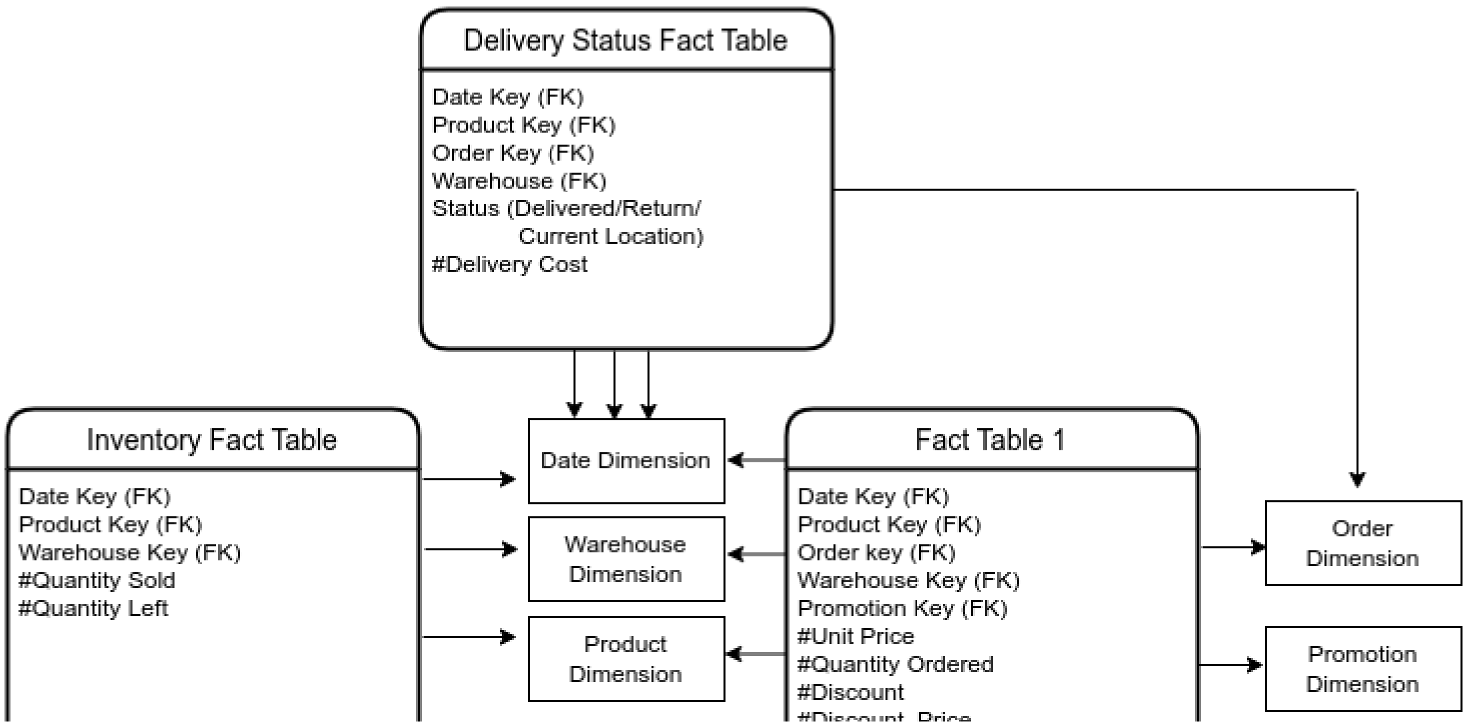
Fact Table 3:



Grain: Daily inventory for each product in each store.

So through this fact table, we will track the stock of the different products. This is how we create our dimensional model by following the above steps.

Our final model will look like this:



customers and companies are depicted.

2. **Periodic Fact Tables:** These record a row in the table for a definite period of time. That period of time can be daily, monthly, etc. Here the grain is the period of time we select. Example – The inventory fact table that we have taken is an example of a periodic fact table in which we will be calculating the stock of the products on a daily basis.
3. **Accumulating Fact Tables:** It stores predictable steps between the process's beginning and end. Whenever a predictable step is recorded, the table is revisited and updated. Updation of a row is unique to this type of fact table only as compared to the other types of fact tables. Example: The order's delivery status is an example of the accumulating fact table in which we have to update the product's location whenever it reaches the desired location until it is delivered.
4. **Factless Fact Tables:** It records a row in the table with no numeric measurement, but that grain is important to store. This table is also used to record data that didn't happen. Example: A table showing the products on sale which were not sold.

Some important points regarding Fact Tables:

1. A fact table generally contains numeric and additive facts. E.g., the number of products sold, the cost of each product, etc
2. Facts can also be semi-additive or non-additive.
3. Generally, textual data is not stored in a fact table; if required, it can be stored.
4. Continuous numeric values are stored in the fact table. E.g., The cost of products can take value in a broad range.
5. We should not store redundant textual information in fact tables unless the text is unique for every row in the fact table.
6. We should avoid using 0 for no activity in the fact table.
7. Fact tables have more rows and less number of columns.
8. Fact tables have two or more foreign keys that connect to the dimension table's primary keys.
9. It answers the question, "What is the business process measuring."

Conclusion

I hope through this article; you got a basic understanding of Dimensional modeling. This article shows how fact and dimension tables are created and the steps we follow to implement a dimensional model successfully.

Key Takeaways

- Data Warehouse is a central repository used to store data that data analysts and data engineers can use for analysis purposes.
- Dimensional modeling is a technique used to store normalized forms of data.
- Dimensional modeling consists of creating Fact and Dimension tables.

data warehouse   Database normalization   Dimensional modeling   fact table