Project

Data Warehousing

Topic: Modeling the data for the business.

Worked by: Armand Palla

Table of contents

Abstract

Regardless of what technology will be used, the heart and soul of the data warehouse is the data itself. How the data is organized can have a significant impact on how well the environment will work. A great deal of thought and care must go into the design of the data. This project presents the concepts covering how the data should be organized to support reporting and analysis. The business team will learn the purpose of dimensional models, basic components of a dimensional model and how to contribute to the modeling process. A case study of the dimensional model for call center operations is used to reinforce the concepts presented. This project also includes content that is geared toward the more technical members of the project team, including a technique to document the dimensional model, specifically geared toward the business community, a process for developing the dimensional model with participation from the business community and how to take the model forward. This report is not intended to provide an in-depth guide to dimensional modeling. It does share the basic concepts that can be used to communicate more effectively between the business and technical team members. Before diving into the modeling concepts, the first section reviews the rationale for dimensional modeling and why it is of interest to business people. Dimensional models support the business perspective of the data, and today's technology ensures that they can be effectively implemented. This approach has been applied to data warehousing for nearly thirty years and is supported by a wide variety of database platforms and data access or business intelligence tools. Dimensional data modeling is also best suited for the data warehouse star and snow flake schema. The goal of this paper is to free the data that has been captured and stored by the operational systems and make it available to the business community.

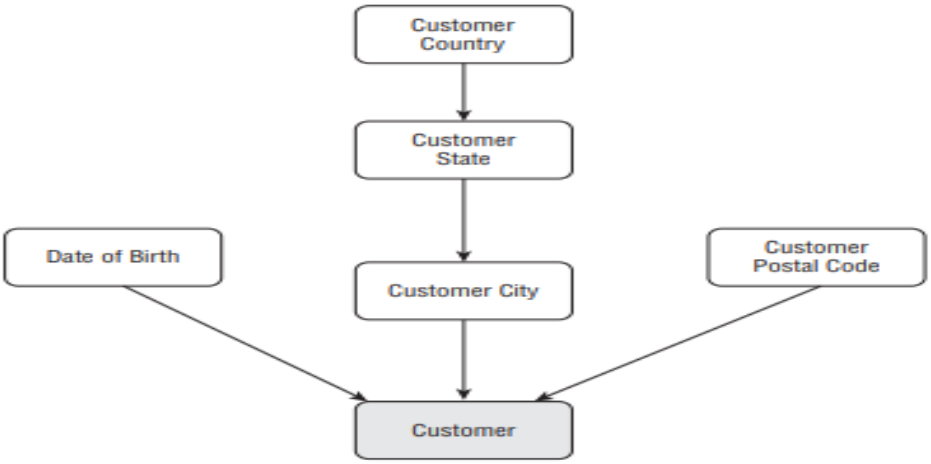
1.The purpose of the Dimensional Models

A specific modeling technique has evolved in order to support the types of queries and analyses that businesses require. This technique is called dimensional modeling. Regardless of how data is structured, business people will ask questions based upon their frame of reference. This perspective is driven by the basic characteristics of the industry and how the company is organized, so why not organize the data to reflect this business perspective? The two primary goals for dimensional modeling are **ease of use** and **query performance**. These are the principles that guide the entire dimensional modeling process. There are other data modeling techniques that play an important part in overall systems development. They help ensure that the data itself and the relationships between different data elements are clearly defined. For operational systems, it is important that the data be organized to facilitate transaction processing. This includes ensuring transaction integrity and speed. The type of modeling used for operational system design is called entity-relationship (E-R) modeling. This may also be referred to as normalized modeling. One specific form of E-R modeling represents the data in third normal form (3NF). There is a complete discipline surrounding this approach to data modeling. This is mentioned to acknowledge the value and purpose of E-R modeling for operational system design. In order to ensure that people will use a data warehouse, the data must be presented in a manner that makes sense to them. If it is too confusing or does not mirror the way the business runs, then people are not likely to use it. Therefore, the dimensional model must cleanly represent the basic components of the business. In addition, the model must be presented in terms that are used by the business. A well-designed dimensional model should be obvious to the business community and be met with a confirmation that it is indeed correct. The second, and equally important, goal of a dimensional model is to ensure good query performance. If requests do not run in a timely manner, the data warehouse will not be used and

will not be helpful to the business. Dimensional modeling takes the need for this query performance into account as part of the inherent design approach.

1.a) Implementing a Dimensional Model.

A dimensional model is a data model organized for the purpose of user understandability and high performance. There are two basic parts of a dimensional model: the dimensions and the facts. These are the building blocks that comprise all dimensional models, simple or complex. Dimensions are groupings of data elements in major business categories. Common dimensions include the following: Customers, Products, Dates, Suppliers, Vendors and Accounts. The individual data elements are called **dimensional attributes**, or reference data. The dimensional attributes are used as row and column headings for reports. They are used to create lists of options to determine what to include or exclude on a report. The relationship between these dimensional attributes creates drill paths or the ability to navigate up and down a hierarchy. The need for dimensional data is often recognized while gathering business requirements. It may not be directly communicated, but realized when someone needs a report by region, by week, and by product category. Each of the terms following the word ‘‘by’’ is a dimensional attribute. These should be included in dimensions to support that type of reporting. An example of a customer dimension is shown in the figure. This is a highly simplified example that only shows the customer’s address and date of birth attributes. Some of these attributes relate to each other in a hierarchy, while others are simply additional characteristics of the customer. **A dimension can include attributes that are descriptive and that relate to each other, creating hierarchies.**



Many different database technologies are available today to store data. Many of these have been developed specifically to support data warehouses. It is useful to have a basic understanding of these in order to be able to put the dimensional model into the appropriate context. A dimensional model is not inherently tied to a specific technology, and it can be implemented in a variety of different ways. Different types of databases include the following:

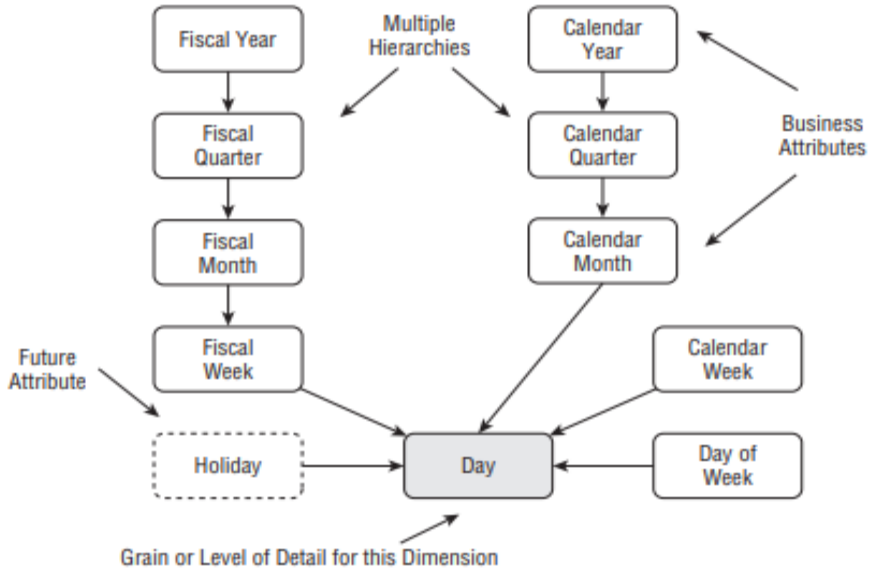
- **Relational databases** are one of the most commonly used databases for both operational and data warehouse systems today. When a dimensional model is stored in a relational database it is called a **star schema**. This is due to the visual appearance of the dimension tables surrounding the fact table.
- **Multi-dimensional databases** are specifically designed to support a dimensional view of the data. The data is stored in proprietary array structures. When a dimensional model is stored in a multi-dimensional database, it is called a **cube** (even though it may actually contain more than three dimensions).
- **Proprietary databases** are also available on the market. Many of these are designed specifically to support reporting and analytical use. There is a wide range of different methods used to physically organize and access the data in these environments. Once the dimensional model is completed, there is often no additional design work required to determine how the data will be structured in these environments. This is all handled by the

proprietary system. When database software and hardware are fine-tuned and bundled together, they are called a **data warehouse** appliance.

There are different ways to document and present dimensional models. One of the most common ways that dimensional models are depicted are as tables to be stored in a relational database. The dimensional model can be documented using the same modeling tool that is used to develop any other data models for the relational database. Each of the dimensional attributes is included and represented using logical names that should be meaningful to the business. This type of table diagram is easily understood by systems professionals, but it is not as clear to business professionals. Another method to document your dimensional model is to present **business diagrams**. The intent is to visually present the model in terms that more closely reflect the interface that will ultimately be presented for access. This is called **business dimensional modeling**, and it can be documented using any visual diagramming tool. The primary and the most important difference is how the model is presented to the business.

With the increasing variety of options for building the data warehouse, it is important to split the business perspective from the technical perspective regarding the data. The Business Dimensional Model (BDM) is a data model that is specifically geared toward working with the business community. It serves as an abstraction layer that insulates the model from technical implementation details. The model also serves as a communication vehicle between the business and systems groups. The model shows diagrams of the dimensions and the facts so that the details can be reviewed and discussed in business terms. This also separates the business discussion from any technical discussions. A dimensional model can reflect a wide range of data from multiple data sources. The focus is on understanding the dimensionality of the business itself and the facts that are needed to measure that part of the business. The business dimensional model will reflect all of the data to be included in the data mart. Each business dimension of the model will be designed and diagrammed separately. The business attributes are included to fully describe the dimension. Each business attribute of the dimension is depicted with a rectangle, as shown below in the sample dimensions. For each dimension, you need to identify the lowest level of detail

that exists. This is also called the **grain** of the dimension. This lowest-level attribute is **shaded**. The relationship between each of the business attributes is noted with an arrow. The direction of the arrow shows the direction to drill down to see more detail. These are also commonly referred to as **hierarchies.**



Sometimes a specific business attribute is requested but is not captured in any current source systems. To facilitate communication, this can be included in the dimension diagram, but the notation is different. The dotted rectangle indicates that the data element is either not captured or is not to be included in the initial implementation. In this example, the Holiday attribute is planned for the future. This ensures that the model reflects true business needs, but it also helps set and maintain expectations that the element is not going to be available at this time. The dimension diagram itself is only part of the documentation that is needed for the dimension. The diagram shows each attribute with a useful business label.

1.b) Facts as the second part of the Model.

Facts are the measurement of business events. These are captured as **specific information** about a business event or transaction. These are measured, monitored, and tracked over time. Facts are typically the amounts and counts that show up as the body of reports. Facts are used as the basis for all calculations. Examples of facts include units ordered, retail price, amount paid, claim payment amount, gross margin, budgeted dollars, revenue forecast, and loan balance, among others. The facts are only interesting within the proper context, and the context comes from the dimensions. For example, the fact that a company had $10,000 in sales is not useful unless you know that it was from red shoes, in the Chicago market, the week before Valentine's Day.

The way that people think about data is often defined by the layout of a spreadsheet: rows and columns with perhaps a separate worksheet to represent another variable. For example, in the figure, a common sales performance report shows the monthly profit results for the current year, reported by product category. Each region of the organization is represented as a separate page. This report is constrained to the current year of data, and uses product, date, and sales organization dimensions. The Product Category attribute is from the Product dimension. The constraint for the current year is on the year attribute of the Date dimension. The months are also from the Date dimension. The Sales Regions (per sheet) are from the Sales Organization dimension. The fact itself is the Sales Profit, listed in thousands of dollars. The dimensions and facts are used together to create basic reports such as this one.

Sales Profit for Northeast Region
For Current Year to Date
(*Profit in thousands of Dollars*)

	Months						
Product Category	**Jan**	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**
Camping Accessories	19	22	37	52	65	84	83
Women's Clothing	63	68	87	65	62	74	69
Men's Clothing	72	81	80	94	87	103	78
Athletic Shoes	201	214	194	183	191	192	199
Fishing Accessories	6	3	8	11	21	17	18
Backpacks	3	4	4	12	16	19	15
Tents	88	83	91	137	139	189	120
Total	452	475	501	554	581	678	582

Modeling the facts is much more than simply creating a list of the business measures that are needed. Each of these facts must be reflected within the proper context. This can be understood by looking at how the data is captured and how the business uses each fact. The dimensions that are relevant to these facts are shown. The grain, or lowest level of detail that applies, is also identified for each applicable dimension. Often, several facts will have the same dimensionality and identical grain. These facts can be put together into a **fact group**. The figure below shows a fact group for retail sales. The name of the fact group is in the hexagon at the center of the diagram. Each dimension that applies to this fact group is diagrammed as a bubble around the hexagon. The specific grain is also included in each bubble. This is important because facts can have similar dimensionality, but be at different levels of detail. If you have two facts that have the same dimensions but one is daily and the other is monthly, these must be split into two separate fact groups. You must design only single-grain fact groups. Mixing the grain of a fact group can cause query results to be incorrect.

Using the diagrams for the dimensions and fact groups, you can see what types of queries can be supported. A more complete **case study** is shown next to better illustrate this capability.

2. A call center case study

Most large organizations have some sort of call center. These centers are sometimes run internally or may be outsourced to third-party organizations. The call center in this example is run internally to handle customer calls. These calls may be to place an order, check the status of an order, request information, file a complaint, or share a compliment. This case study shows the business dimensional model for this organization. While this shows a mostly complete picture of a **dimensional model**, it has been simplified to facilitate an introduction to the modeling approach, rather than to represent a complete model for a real call center. Clearly, there are many nuances and details that are specific to each organization that cannot be addressed here.

I am trying to represent an example of a company to explain in a better way the most important concepts of a dimensional model.

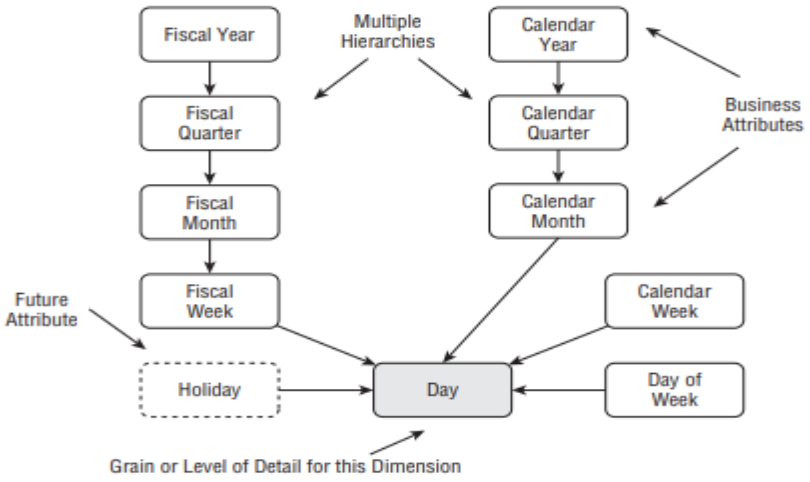
Giant Company : The first organization, which we will call Giant Company, is an extremely large conglomerate. There are thousands of employees in IT alone; and there are multiple divisions, each run as a separate company. Because of the large number of people employed, there are many layers of management and well-structured positions and career paths. As the company grew, many policies and procedures were put in place to assist in the overall management of such a large concern. As employees are brought on board, there is a pre-defined training plan based upon the position. This helps everyone get up to speed as fast as possible. Within each of the divisions, the business is organized into departments with specialized functions. Each department knows what its inputs are, how to get its business processes done, and to whom results should be sent. Much of the business process and decision-making for the daily work has been built into the production application systems. This has enabled the business to expand and ensures consistency. However, knowledge about the rationale and details of the processes has been lost over time. Everyone knows how to keep the systems working, but how they work and why is a mystery. The IT function is centralized, but it is structured to support each of the divisions. To keep it manageable, a clear project methodology is used. This provides consistency from project to project

and helps individuals to understand how to get things done. With such a variety of data warehouse experiences, it is especially critical to compile an inventory of what has been done. With so many layers of management and narrowly defined jobs, it will be challenging to get the right people together to help ensure a successful data warehouse. All employees, businesses, and systems must learn to be more flexible and collaborate. Each person involved with the data warehouse must feel that he or she can challenge the status quo in order to make the solution more useful and valuable.

2.a) Call Center Dimensions

Date Dimension

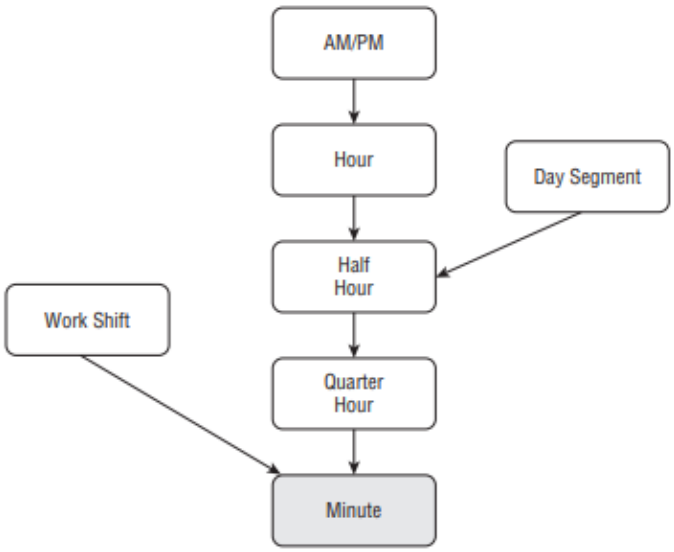
The Date dimension for this call center case study is the same dimension shown previously in Figure. This supports analysis by fiscal and calendar years.



Time Dimension

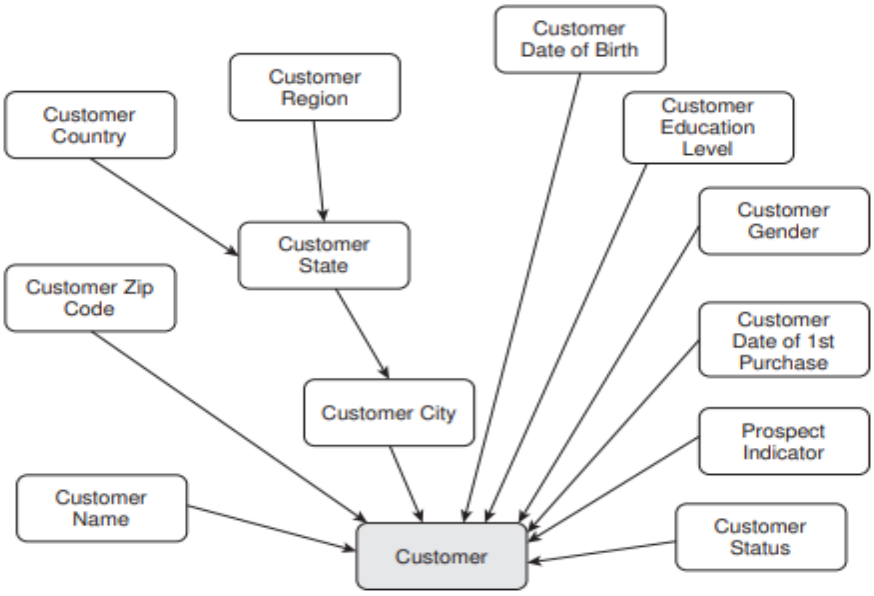
Because call patterns can vary drastically throughout the day, it is important to be able to track the calls at a fine level of detail. Keeping the time of day attributes separate from the date attributes helps simplify the model, reducing the size of the dimension, which provides implementation benefits. The basic hierarchy of the Time dimension shows a straightforward rollup of time. The Time dimension

can also be used to track the work shift of the organization. Although the work shifts currently change on the hour, the company is discussing changing the shifts around. In order to accommodate any possible future definition of the work shift, it is shown to the minute. The Time dimension diagram is shown in the Figure below. Time dimensions are helpful to facilitate the representation of different parts of the day. If the only need for time or timestamp data is to calculate the elapsed time between two events, this would not be needed. If specific attributes are needed for grouping and reporting, these can easily be stored in a Time dimension.



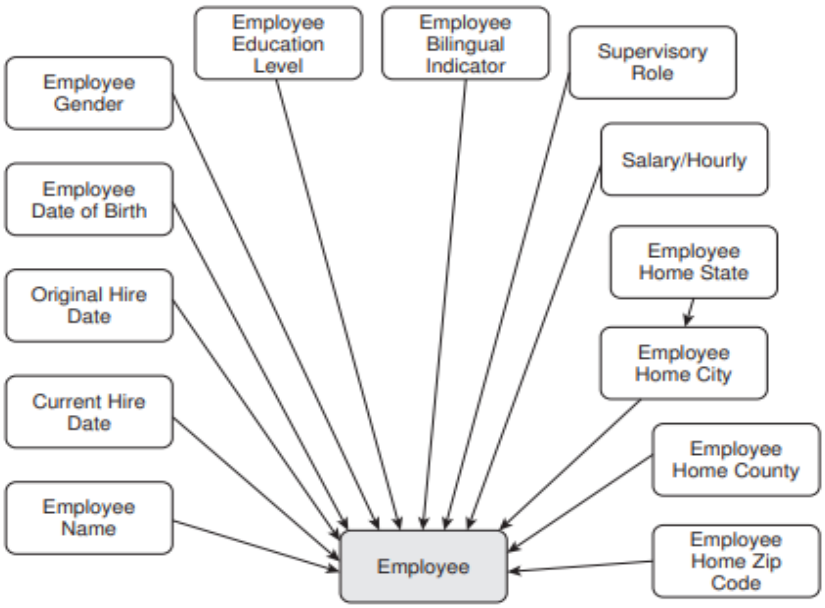
Customer Dimension

The Customer dimension provides information about the individuals who place calls to the company. While some attributes are known through doing business with this customer, the organization may also purchase external demographic data. The Figure that is represented below shows the Call Center Customer dimension.



Employee Dimension

Managing a call center is highly dependent upon the customer service representatives, the people who answer the phones. There is a high turnover rate for call center employees. Productivity of these employees is critical. The Figure shows the Cell Center Employee dimension.



Call Dimension

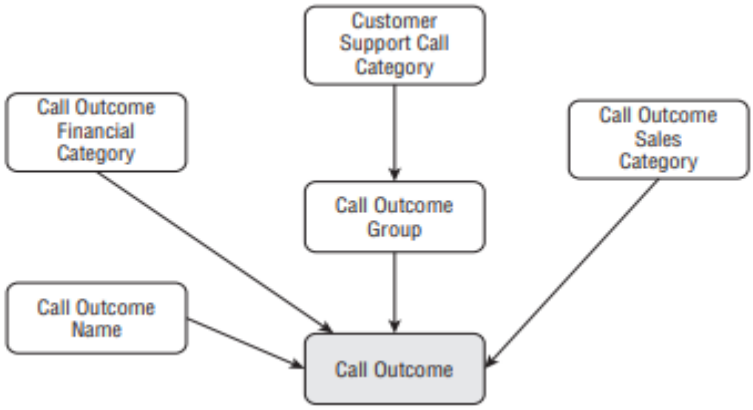
Each specific call must be tracked by the call dimension. There is very little that is known about the call itself that is not already included in other dimensions. This is what is used to link the participation of multiple different employees for a variety of activities. This can happen if a call is transferred. The data warehousing term for this is a degenerate dimension. The Figure below shows the Call dimension.

Call Center Call dimension

Call Transaction. This is the unique identifier that is assigned to a call when it comes into the call center. This is tracked until the call is completed.

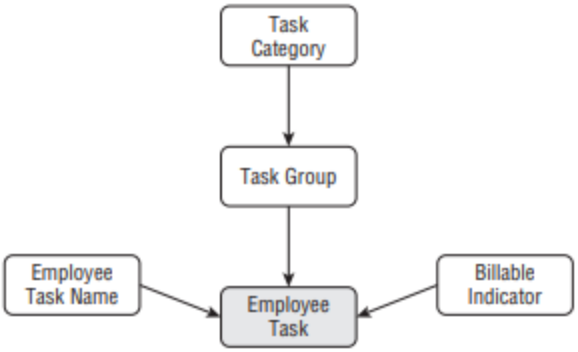
Call Outcome Dimension

When handling incoming calls, the purpose and type of call are tracked. The sales, finance, and call center groups each categorize these calls differently. Each of these groupings is shown in Figure.



Employee Task Dimension

Employee time tracking is based upon whatever activity or task that employee was working on. This is needed to be able to help understand the dynamic between employee activity and call productivity. The Figure below shows the Employee Task dimension.



2. b) Call Center Fact Groups

The second part of the model contains the facts, which is where the business measurements are stored. Modeling the facts is much more than simply creating a list of the business measures that are needed. Each of these facts must be reflected within the proper context. This can be understood by looking at how the data is captured and how the business uses each fact. The dimensions that are relevant to these facts are shown. The grain, or lowest level of detail that applies, is also identified for each applicable dimension. Often, several facts will have the same dimensionality and identical grain. These facts can be put together into a fact group.

Now that each of the dimensions is defined, it is time to look at the different facts that are needed. These are documented next.

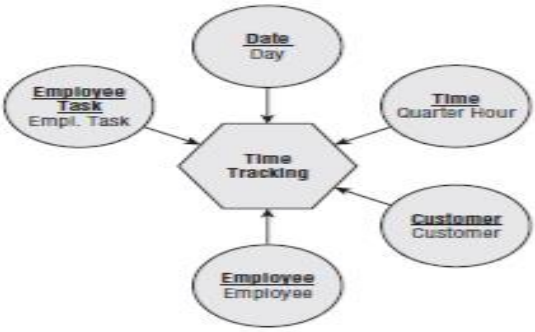
Calls Fact Group

Some of the most critical facts that need to be tracked for a call center are those related to the handling of incoming calls. The Calls fact group is shown in Figure.

Note that not every dimension defined earlier is included in this diagram, only those that are relevant to the calls facts. The minute that the call began for this employee is recorded for the Time dimension.

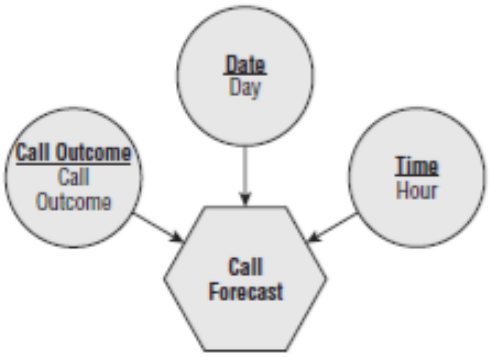
Call Center Time Tracking Fact Group

The ability to understand how employees are spending their time is critical for planning resources. It is also important to be able to track patterns in performance compared to how time is spent. Perhaps more frequent, but shorter breaks make customer service representatives more productive. This is the type of information that needs to be learned by studying the data. To that end, the Figure below shows the Time Tracking fact group. These facts have some of the same dimensions as the Calls fact group, but the Task dimension is new. In addition, it is important to note that the employee time is not tracked at a minute level of detail but in quarter-hour increments.



Call Forecast Fact Group

In order to be able to schedule employees effectively, it is helpful to have an estimate of the expected call volume. The forecast is used to develop employee work schedules. Development of these work schedules is operational in nature. The call volume forecast is useful in the data warehouse to compare with actual call volumes. This enables fine-tuning of the forecasting process. The Figure shows the Call Forecast fact group. Note that the Time dimension is at the hour level of detail. Call volume is not forecast at the minute level of detail. The second Table has the definitions for the Call Forecast facts.



The **definitions** are:

FACT NAME	FACT DEFINITION
Forecast Call Minutes	Expected number of minutes to be spent on the phone with customers
Forecast Number of Calls	Number of calls expected during this hour

3.Star schema and Snowflake schema

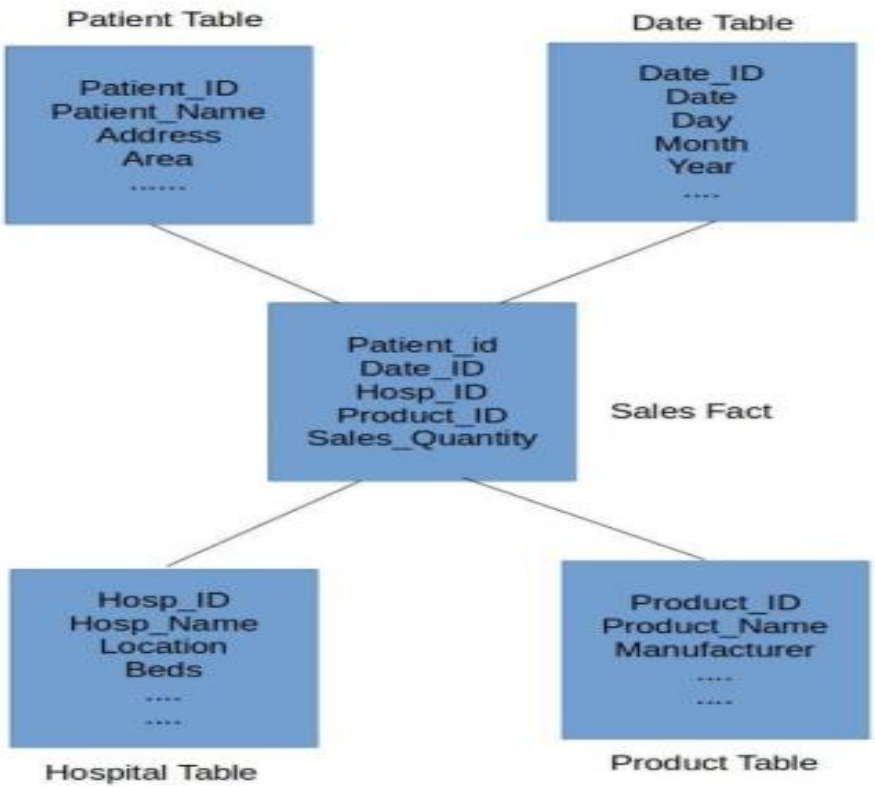
Dimensional data modeling is one of the data modeling techniques used in data warehouse design. The main goal of this modeling is to improve the data retrieval and it is optimized for the SELECT operation. Dimensional data modeling is best suited for the **data warehouse sta**r and **snow flake schema.** I have tried to illustrate two types of schemes with a short example in order to understand the differences.

Star schema

Data warehouse Star schema is a popular data warehouse design and dimensional model, which divides business data into fact and dimensions. In this model, centralized fact table references many dimension tables and primary keys from dimension table flows into fact table as a foreign key. This entity-relationship diagram looks star, hence the name star schema. This model divides the business data into fact which holds the measurable data, and dimension that holds descriptive attributes related to the fact data. For examples, fact data includes price, quantity, weight measurements and related dimension attributes example includes product color, sales person name, sale geographic locations.

To understand this model, we will consider **medicine sales** table that holds the information about the sales quantity for each product, patient, hospital and on certain day. In this example, sales quantity is the measure and primary keys from Product, Date, hospital and Patient dimension flows into fact table. Fact table always holds measures and keys from dimension tables.

This schema is one of largely used data warehouse design methodology and popularly used in data warehouse applications, hence optimized for querying large data sets. This model is popularly used in OLAP, business intelligence and analytics applications.



Characteristics of Star Schema

- Dimensional Tables are not normalized.
- The dimension table should contain the set of attributes.
- Dimension table is joined to only Fact tables. They are not joined to each other.
- Fact table stores keys from dimension tables and measure.
- The Star schema is easy to understand and provides optimal disk usage.

Benefits of Data warehouse Star Schema

Star schema model are intentionally denormalized to speed up the process. Below are the benefits:

- **Simple queries:** Join conditions are simple joins in this schema
- **Simplified business logics:** This model simplifies common reporting business logics

- **Performance**: This model provides performance enhancements for reporting applications.
- **Fast aggregation**
- **Feeding cubes:** This model is generally used by OLAP systems to build cubes. Building cube is a very fast process.

Disadvantages of Data warehouse Star Schema

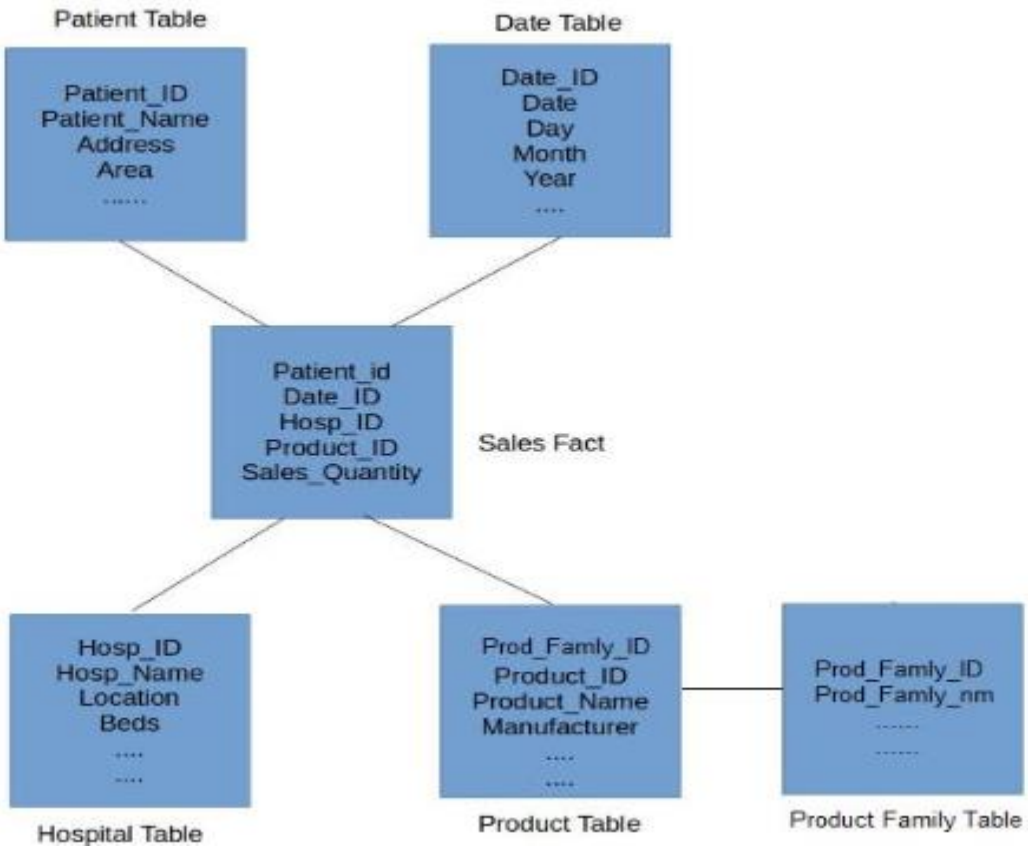
- The main disadvantages is, that **data integrity is not enforced** as in OLTP databases
- This is not flexible inters of analytical application as **normalized databases.**
- This model does not support many to many relationships between business entities. These types of relationships are simplified in star schema.

Snowflake schema

Data warehouse Snowflake schema is extension of star schema data warehouse design methodology, a centralized fact table references to number of dimension tables, however, one or more dimension tables are normalized i.e. dimension tables are connected with other dimension tables. Primary Keys from the dimensions flows into fact table as foreign key. Snowflake schema increases the level of normalization in data, the dimension table is normalized into multiple tables. This schema has a disadvantage in terms of data retrieval, we need to travel through multiple tables to get same information. SQL queries would have more joins in order to fetch the records.

In snowflake schema, dimension table are normalized, where as in star schema these are denormalized. Let us understand this model by using the same example that we discussed in star schema designing. Consider medicine sales table that holds the information about the sales quantity for each product, patient, hospital and on certain day. To implement snowflake schema, let’s create once more

dimension table called **product family**. Primary key from product family table flow into product table as foreign key (product table is normalized). In this example, sales quantity is the measure and primary keys from Product, Date, hospital and Patient dimension flows into fact table. Fact table always holds measures and keys from dimension tables.



You can see the product table refers the product family table. In order to get the product information, you need to join product table with product family table. This type of schema is called snowflake schema.

Benefits of Data Warehouse snowflake schema

- Snowflake model is in same family as the star schema. In fact, it is a special case of star schema. Some of the benefits includes:
- Some OLAP multidimensional model tools are optimized to use snowflake schema.
- Normalizing table saves the storage.

- Improvement in query performance due to minimized disk storage and joining smaller lookup tables.

Disadvantages of Data Warehouse Snowflake schema

- Additional maintenance efforts needed due to the increase number of lookup tables.
- SQL queries would have more joins in order to fetch the records.
- Data loads into the snowflake model must be highly controlled and managed to avoid update and insert anomalies.

4.Summary

Dimensional data modeling in data warehouse is different than the ER modeling where main goal is to normalize the data by reducing redundancy. This model gives us the advantage of storing data in such a way that it is easier to store and retrieve the data once stored in the data warehouse. Dimensional model is the underlying data model used by many of the OLAP systems. Dimensional models are intuitive and identify the data required for business analysis and decision support. The DM is a logical design technique often used for data warehouses. It is the only viable technique for databases that are designed to support end-user queries in a data warehouse. Every dimensional model is composed of one table with a multi-part key, called the fact table, and a set of smaller tables called dimension tables. Each dimension table has a single-part primary key that corresponds exactly to one of the components of the multi-part key in the fact table.

In the first chapter we have described the purposes of a DM and two main purposes in which we have focused more are ease of use and query performance. At the first chapter we have also introduced how to implement a DM and we have

different types of databases as relational database, multi-dimensional databases and proprietary database. All the main elements of a DW are described and also illustrated with schemes in order to understand that in an easy and compact way. Again in the same chapter we have described the Facts as the second component that describes the information more specifically and that forms Facts Group that are several facts with the same dimensionality and identical grain.

In the second chapter we have described the most important concepts of a DM through one example or one case study and we have demonstrated with definitions and with schemes the main elements of a DM in order to understand better the approaches of a business dimensional model. The second part of the chapter contains the facts, which is where the business measurements are stored. Modeling the facts is much more than simply creating a list of the business measures that are needed. Each of these facts must be reflected within the proper context. This can be understood by looking at how the data is captured and how the business uses each fact. This thing we have tried to illustrate above.

In the third chapter we have described Dimensional data modeling as a model suited for the **data warehouse sta**r and **snow flake schema.** I have tried to illustrate two types of schemes with a short example in order to understand the differences and to represent the idea of these schemes of largely used data warehouse design methodology and popularly ,used in data warehouse applications, for querying large data sets. The model that we have described is popularly used in OLAP, business intelligence and analytics applications.

To conclude my opinion I have choose this topics because we have treated them in our lessons and they are very important for business analysis and decision support.

5.Bibliography

[1] A Manager's Guide to Data Warehousing by Laura L. Reeves

[2] https://www.sciencedirect.com/topics/computer-science/dimensional-model

[3] https://dwgeek.com/guide-dimensional-modeling.html/

[4] https://data-warehouses.net/glossary/dimensionalmodel.html

[5] https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/

[6] https://en.wikipedia.org/wiki/Dimensional_modeling

[7] https://www.guru99.com/dimensional-model-data-warehouse.html