



The Benefits of Data Modeling in Business Intelligence

DECEMBER 2008

Table of Contents

Executive Summary	1
SECTION 1	2
Introduction	2
SECTION 2	2
Why Data Modeling for BI Is Unique	2
SECTION 3	4
Understanding the Meaning of Information	4
SECTION 4	7
Supporting Reporting Needs	7
SECTION 5	8
Conclusion	8

Copyright © 2008 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. This document is for your informational purposes only. To the extent permitted by applicable law, CA provides this document "As Is" without warranty of any kind, including, without limitation, any implied warranties of merchantability or fitness for a particular purpose, or noninfringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, business interruption, goodwill or lost data, even if CA is expressly advised of such damages.

Executive Summary

CHALLENGES

Business intelligence (BI) is critical to many organizations today. Faced with ever-growing amounts of data, the challenge is to make sense of this data and unlock information that is useful and relevant to the business. A data model is a valuable communication tool to ensure that database developers understand and meet the needs of the business in the physical database system. Challenges include:

- Understanding the meaning of key business terms.
- Ensuring that reporting needs are met so that users can create flexible queries using the correct information.

OPPORTUNITIES

If database developers meet the needs of the business in their physical database designs, systems can be developed that unlock the relevant information from the vast quantities of data that their organization holds. Opportunities include:

- Providing accurate reporting of the performance of the organization at every level, from the most detailed information to a high-level overview.
- Producing accurate predictions of future events based on past results, enabling business users to make informed choices about corporate strategy.

BENEFITS

Through data modeling of BI systems, we can meet many of today's data challenges. Through logical and physical modeling of BI systems, we can enable the delivery of the correct business information to business users. Key benefits include:

- Reduced development time of BI systems through a thorough understanding of source systems.
- Increased accuracy of BI results.
- Increased transparency to enable business users and developers to realize the information that is available to them.

SECTION 1

Introduction

It is clear from the name that business intelligence (BI) is intended to deliver intelligence to the business. Businesses use the output from BI systems to develop a strategy for their organizations at the very highest level. This can deliver enormous benefits, but, if the data is misunderstood, there are enormous risks. It is therefore essential to have a thorough understanding of the data to avoid these risks. A data model is a valuable tool that is used to help businesspeople and IT communicate effectively. The databases that are built to support BI must therefore contain the correct information in a format that the business can use.

Business users typically see the output of BI systems as reports or digital dashboards, but underlying these systems are online analytical processing (OLAP) cubes. OLAP cubes pre-aggregate the results of queries, producing results in seconds when they might otherwise take hours.

This white paper discusses the role of data modeling in documenting, explaining, and clarifying the output of the various components of BI. Each component has different structures and requires different approaches for the data modeler. However, with careful use of modeling tools, not only can we avoid the risks of misinformation, we can also extend the benefits of fully understanding the designs of our systems.

Note: All of the models that are used in this white paper are simplified to increase clarity. In reality, there are likely to be many more entities and attributes.

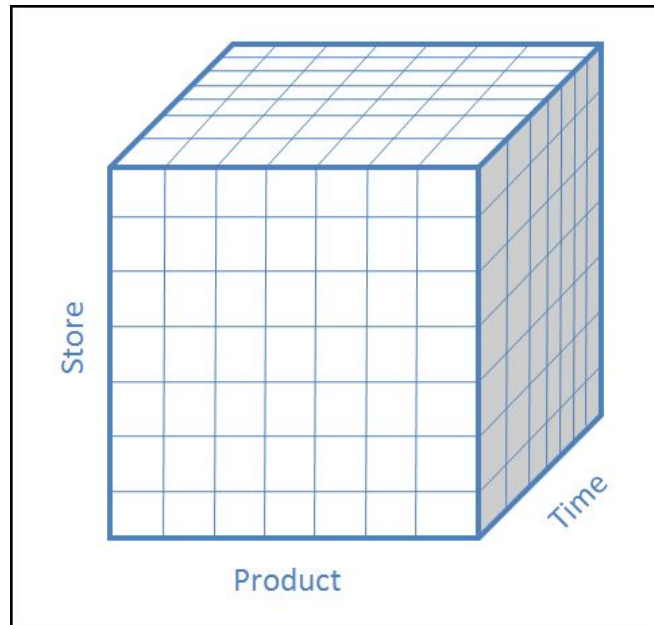
SECTION 2

Why Data Modeling for BI Is Unique

Consider a multinational grocery retailer. Every Monday morning, the trading team uses a pivot table that displays total sales by value and quantity broken down by product group, individual product, region, and store. To create this pivot table, the database needs to perform a query that aggregates millions of individual records. Whenever the information needs to be aggregated in a different way, for example, by week rather than by month, we need to run a highly intensive query again. OLAP seeks to address this problem.

OLAP storage is fundamentally different from a relational system. The simplest way of visualizing OLAP is as a multidimensional structure such as a cube. The axes of the cube are dimensions and the intersections of dimension members are the aggregated results.

Figure 1 shows a graphical representation of a cube with three dimensions:

FIGURE 1: OLAP CUBE**OLAP CUBE WITH PRODUCT, STORE, AND TIME DIMENSIONS**

In reality, there might be many more dimensions, but this becomes difficult to visualize and represent graphically. Also, OLAP structures are always referred to as cubes, regardless of the number of dimensions. Each product represents a column, or slice, of the Product axis; each store represents a slice of the Store axis; and each day represents a slice of the Time axis. If we supply the values for a product, a store, and a date, we have the coordinates for a cell. This cell contains the aggregated value of each measure in our fact table for the supplied dimension members.

Dimensions are not flat structures, but are typically hierarchical. For example, the Time dimension might have Day, Week, Month, Quarter, and Year levels, and aggregation can take place at all of these levels. Furthermore, the Time dimension is not a simple hierarchy because the Week level does not fit neatly into the Month level, necessitating multiple hierarchies for this dimension.

Because of the complexities of OLAP design, it is essential to have a good understanding of the underlying data. This includes having a logical model that presents a more useful structure than the physical model of the data warehouse. Physical models display the exact nature of the database metadata including data types and indexes. Logical models are an abstract layer above this, which present a clarified view to other users.

It is also essential to add details to the entities in the model. Most entities have a single word as a name, but this can lead to questions such as “What is the structure of time in our system?”, “What is a store—does it need to be a physical building or could it be an online web site?”, and “Are products bought or sold?” If we do not fully understand the data, we cannot create useful BI output.

By correctly modeling our systems, we can create OLAP cubes that produce results for our multinational grocery retailer in seconds rather than hours.

SECTION 3

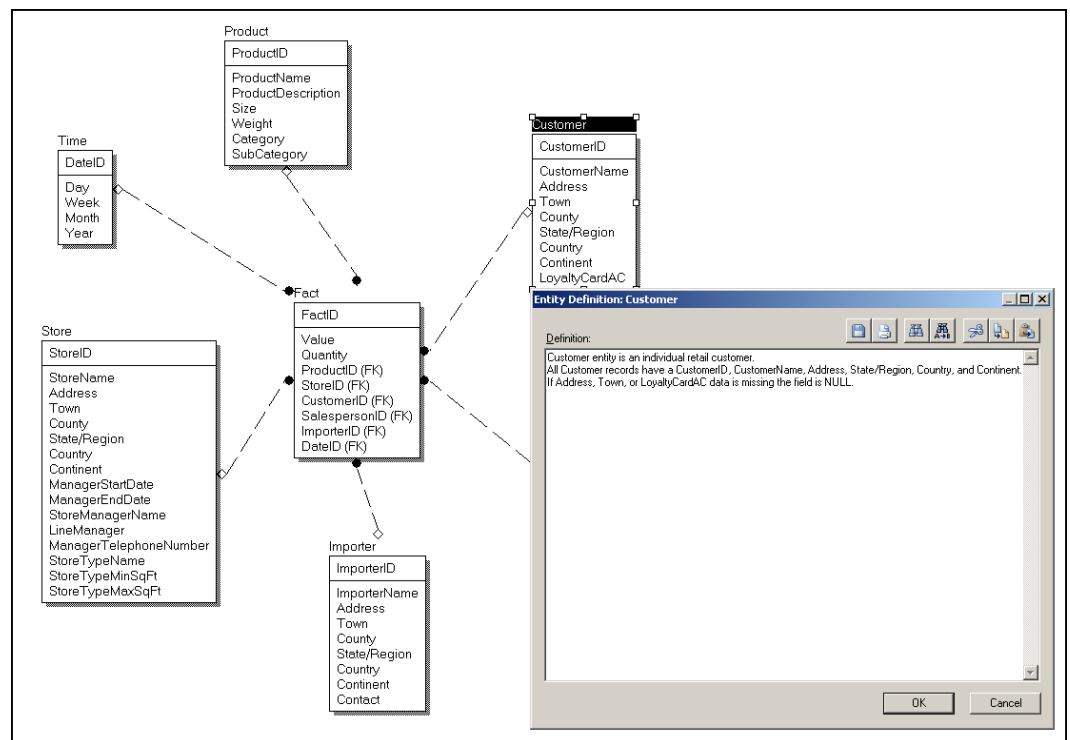
Understanding the Meaning of Information

Consider again the multinational grocery retailer. It has customers that are companies, customers who are individuals who make purchases in stores, and customers who make purchases on the Internet. All of the customer types are described as customers, but in different source systems. This has caused problems in BI systems because incorrect customer data has been fed to the pivot tables of the business users.

To solve this problem, the logical model should be descriptive and there are several considerations to take into account when we design the logical model. For example, what is a customer? Is a customer an individual, or is it a company? If it is a company, do we need to keep track of individuals within that company? Are there relationships between dimensions, for example, between products and suppliers? After we have answered these types of questions, we can create a logical model that describes how all of the elements of the OLAP cube connect, and what the function of each element is. In the example of the multinational grocery retailer, the business users are presented with a pivot table and do not need to understand how that pivot table is created. What they do require is that their definition of a customer, store, product, month, or sales unit is the same as that of the data modeler.

Figure 2 shows that a logical model provides an extra level of detail that is essential when we are creating OLAP cubes.

FIGURE 2: ENTITY DEFINITION

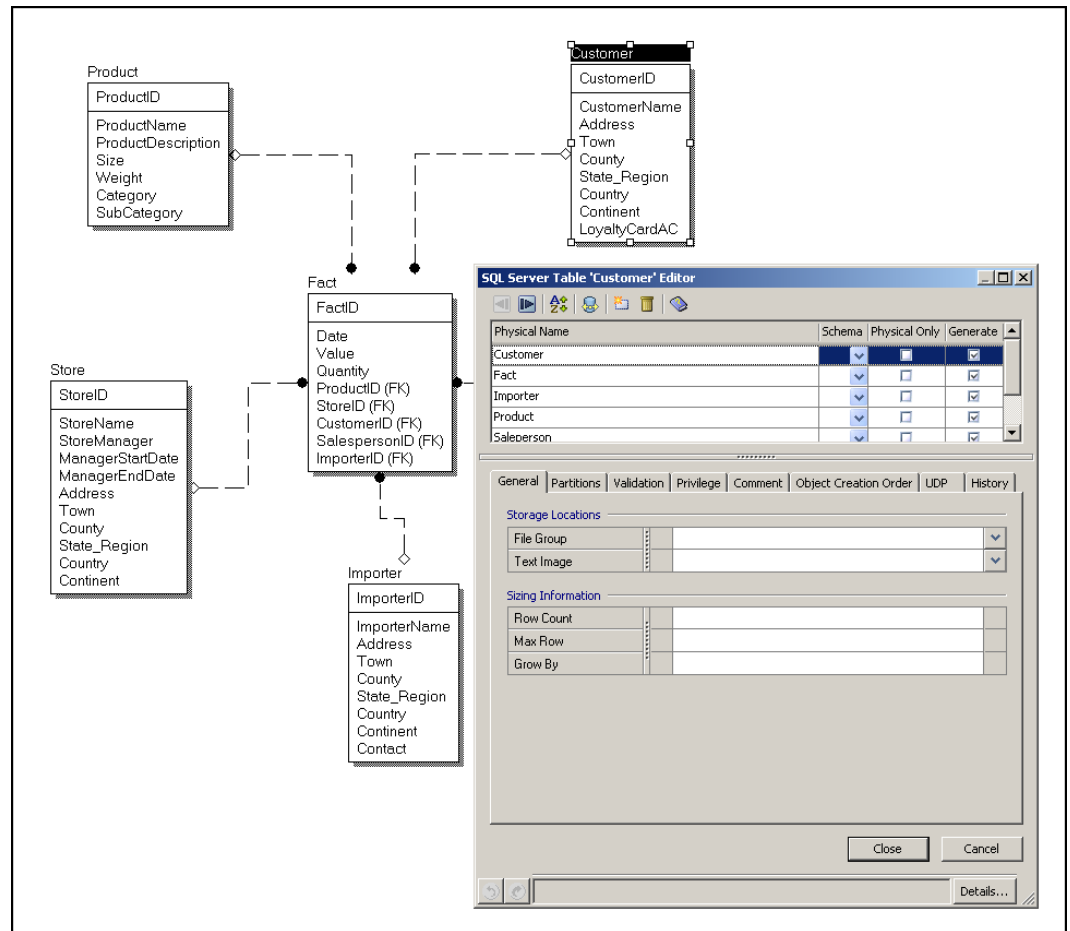


LOGICAL MODEL WITH ENTITY DEFINITION

Furthermore, we should create a model that supports business reporting. Remember that dimensions are not flat structures, but are typically hierarchical. Aggregation can occur at all levels in a dimension, so we cannot treat dimensions as typical transactional tables. Equally, we cannot treat fact tables as typical transactional tables. We need to know which attributes are the measures of the fact table and whether these measures are additive,

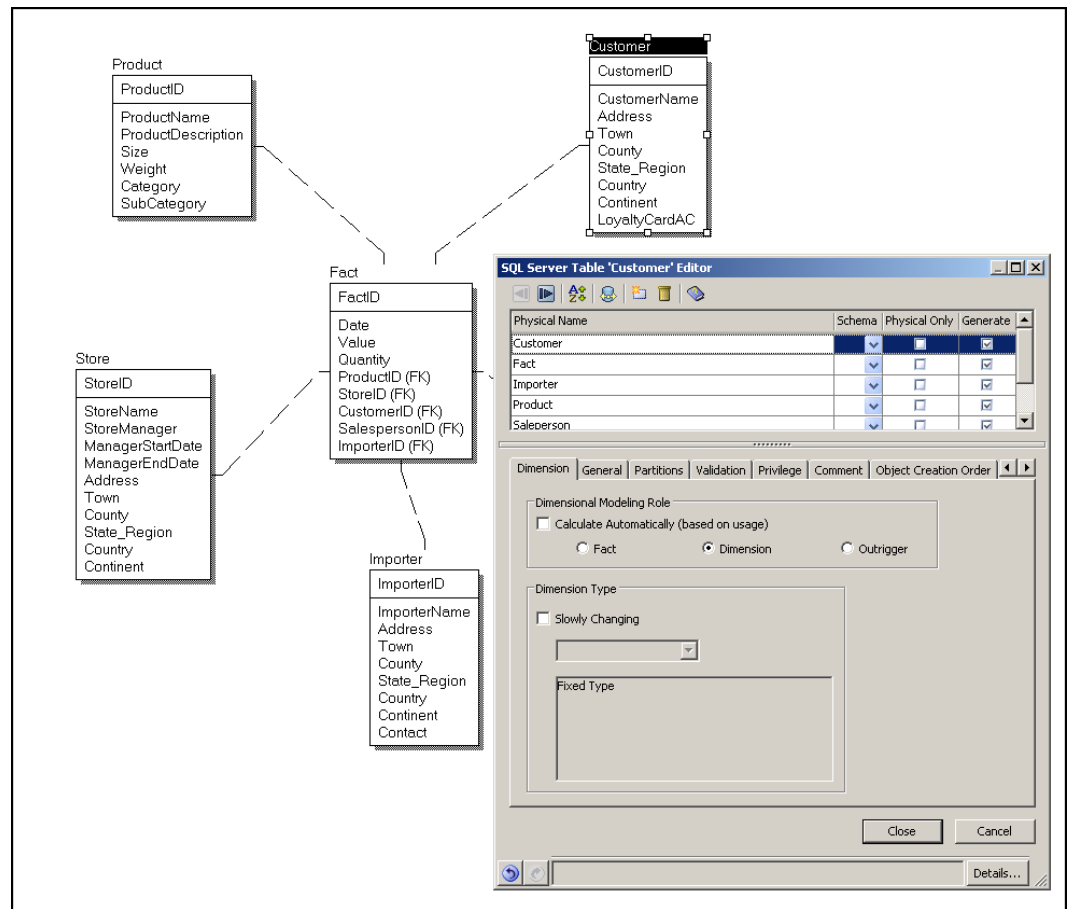
semi-additive, or non-additive. Figure 3 displays the properties of the Customer table in a typical non-dimensional model.

FIGURE 3: NON-DIMENSIONAL MODEL



NON-DIMENSIONAL MODEL DISPLAYING THE ATTRIBUTES OF A FACT TABLE

Figure 4 shows the same tables, but this time we are using dimensional modeling. We can now define the modeling role of the table as a dimension table.

FIGURE 4: DIMENSIONAL MODEL**DIMENSIONAL MODEL DISPLAYING THE ATTRIBUTES OF A FACT TABLE**

The structure of the output of an OLAP system is very different from the source data warehouse, so the physical design of the data warehouse is of little use for developers who are using OLAP data. It is crucial to create logical models that map to the OLAP structure.

A developer who is using the results of the OLAP cube would not understand the physical or logical model of the data warehouse because the structure is very different. We should create a logical model to represent the results of the OLAP cube. For example, the data warehouse contains sales and the dates of these sales, but has no concept of sales value by week. This is the sort of information that is essential to a developer who is using OLAP data. We can see an example of an OLAP output logical model in Figure 5. We have based this on the same system as all of the other diagrams in this white paper.

It is clear that the logical model for the output of the OLAP systems is often substantially different from either the logical model or the physical model of the data warehouse. Furthermore, we can see that logical data modeling is essential to understanding the structure of an OLAP cube.

With correct modeling and documentation, our multinational grocery retailer avoids any misunderstanding and can create accurate and useful information. Initially, the business defined its report requirements and then, after the data-modeling team had received the requirements, each entity was defined and checked with the business users. This removed any potential misunderstandings. The data-modeling team could then create a logical model to match the business requirements and to link to the underlying business structure. This

process would remove ambiguity and confusion, and ensure that the correct information is delivered to the correct people.

SECTION 4

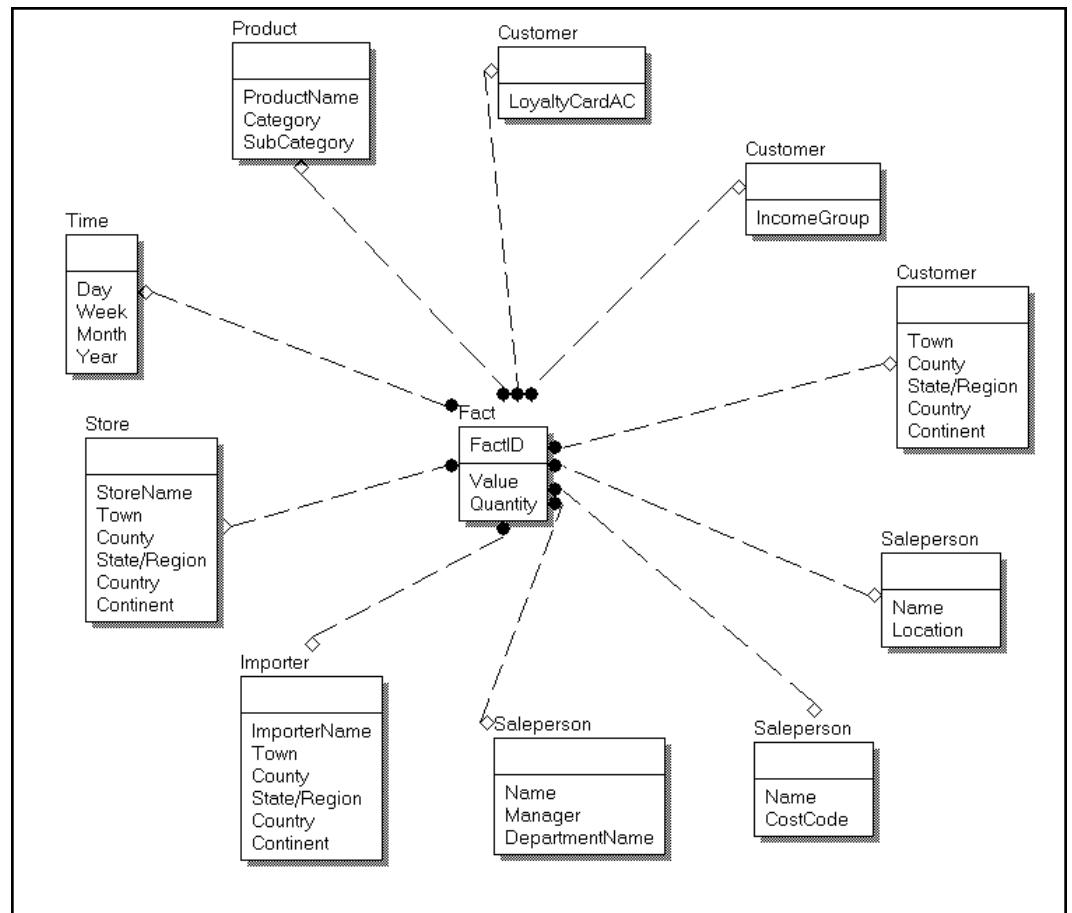
Supporting Reporting Needs

As we have previously discussed, our multinational grocery retailer provides business users with pivot tables to analyze business performance. This has proved successful, but some business users want to know what other information is available. They have been presented with logical models of the enterprise data, but cannot comprehend how this structure relates to the pivot tables and other reports that they use. They have even passed these models to the application developers who create the pivot tables, but again, these models are not fit for this purpose.

In the same way that it is important to create logical models for OLAP, it is also important to create logical data models for both the input and output of reports. Reports examine the output of OLAP and data-mining queries, and the entities that appear do not exactly match the relational storage design. Previously, we have discussed creating logical models for the output of both OLAP systems and data mining. The reporting system is a consumer of this data and, therefore, a user of these models.

For example, monthly sales are not stored anywhere in the data warehouse and the attribute does not occur in the logical or physical model of the data warehouse.

Figure 5 shows an OLAP output logical model. This is based on the same system as all of the other diagrams in this white paper, but the dimensions are split to display each hierarchy. For example, the Customer dimension has a hierarchy for loyalty card account because family members share loyalty card accounts. There is also a hierarchy in the Customer dimension for income group, and another for geographic location. There is no record of the individual customer because this level of detail is seldom required in OLAP cubes. We can also see that we have removed many attributes from this logical model because these attributes are not used in the OLAP cube.

FIGURE 5: OLAP OUTPUT MODEL**OLAP OUTPUT MODEL**

When we are designing reports, it is invaluable to use a logical model that is similar to the model in Figure 5 because it clearly displays the information that is available for the report. The model provides a structure that is useful for application developers to create pivot tables, and is straightforward enough for business users to see what information is available to them.

SECTION 5

Conclusion

Business intelligence can deliver enormous benefits, but also has enormous risks. Business users take the information in their reports as fact, and these facts are used to make business decisions, so poor decisions could be made if information is incorrect. However, if we carefully model our BI systems, these problems can be avoided. Furthermore, the information is more accessible to both developers and business users. We can see that data modeling plays a crucial role in BI development by improving accuracy, reducing development time, and increasing information availability.

