Data Warehousing Battle of the Giants:

Comparing the Basics of the Kimball and Inmon Models

Mary Breslin

Mary Breslin has worked in both user and IT roles and she is currently exploring Capella University's data warehouse from the user side. marybreslin@earthlink.net

Many organizations today need to create data warehouses—massive data stores of time-series data used for decision support. These organizations face a range of choices, both in terms software tools and development approaches. Making good choices requires an understanding of the two main data warehousing models—Inmon's and Kimball's.

Bill Inmon advocates a top-down development approach that adapts traditional relational database tools to the development needs of an enterprisewide data warehouse. From this enterprisewide data store, individual departmental databases are developed to serve most decision support needs.

Ralph Kimball, on the other hand, suggests a bottom-up approach that uses dimensional modeling, a data modeling approach unique to data warehousing. Rather than building a single enterprisewide database, Kimball suggests creating one database (or data mart) per major business process. Enterprisewide cohesion is accomplished by using another Kimball innovation, a data bus standard.

Understanding how these two models are similar and how they differ gives the reader a foundational knowledge of the most basic data warehouse concepts. We will also explore which organizational characteristics are best suited to each approach.

Introduction and Context

We begin our discussion by defining the data warehouse. We will introduce the Inmon-Kimball debate, and provide a brief history of the evolution of the two models. We also provide a brief explanation of the nature of the data warehouse, and conclude with a discussion of the scope of the article.

Context of the Inmon-Kimball Debate

A data warehouse contains massive amounts of highly detailed, time-series data used for decision support. Data warehouses often contain terabytes of data that can be readily queried by end users. The sources of most of the data in a data warehouse are internal transaction processing systems (also known as operational systems). Specialized software extracts data from operational databases, then summarizes, reconciles, and manipulates it. Then the data is ready to be stored in carefully designed relational database tables in the data warehouse.

An organization must choose a set of data warehouse design and maintenance tools from among scores of software tools commercially available. Not all tools are compatible with each other, and not all tools are appropriate for all development methodologies. Despite the array of choices, the industry's tools and methodologies are generally based on only two models: Inmon's and Kimball's.

Choosing between Inmon's, Kimball's, and a hybrid model is, at the most basic level, a choice of both architecture and methodology (Wells, 2003a). Understanding the basics of the architecture and methodology of both models provides a good foundational knowledge of data warehousing. Upon this foundation, readers can build situation-specific knowledge that is appropriate to their organization's needs.

History of the Data Warehouse

How did Inmon and Kimball come to be giants in this field? Each is a creator of a unique school of thought and practice within data warehousing.

In 1990, Bill Inmon earned the moniker "Father of Data Warehousing" by coining the term in his seminal work *Building the Data Warehouse*. The industry soon began to implement Inmon's vision, with varying degrees of success. In the third edition of this work (2002), Inmon describes a logical architecture that extracts detailed, time-stamped data from disparate operational databases. The data is then transformed and stored in a single database (the data warehouse). Data extracts from this monolithic data warehouse create smaller, departmental databases. Decision support users query and create reports from the departmental databases. To create both the data warehouse and the departmental databases, Inmon proposes a top-down variation of the spiral system development methodology.

After the publication of Inmon's book, other database experts began creating data warehouses. The experience of one scholar-practitioner, Ralph Kimball, led to the development of a model that competes with Inmon's. In 1996, Kimball first published his model in his seminal work, *The Data Warehouse Toolkit*. After several years of experimentation, he published a second edition in 2002. In the latest version, he recommends an architecture of multiple databases, called data marts, organized by business process. The sum of the data marts comprises the data warehouse. He recommends a development methodology that is unique to data warehousing. It involves a bottom-up approach that must adhere to an enterprisewide standard "data bus." (See "The Data Bus and Conformed Dimensions" later in this article for a discussion of the data bus).

Nature of the Data Warehouse

The data warehouse exists to facilitate decision support in the organization. Decision support systems help users with ad hoc analyses and strategic decision making. Generally, decision support systems require historical data, both summarized and at a transaction level of detail. Users need to be able to query these massive amounts of data easily. Often, they may not really know what relationships between data elements they are searching for. One data warehousing anecdote tells how a retail chain learned that new fathers often shopped for diapers and beer in the same trip. Sales of both products soared when the diapers and beer were placed next to one another. Data warehousing technology is credited with the discovery (Albert, 2000).

This example neatly illustrates the nature of data warehousing. What does it take to find a statistically significant purchasing relationship between two such unlikely products? One obvious requirement is that the data you are analyzing must be sufficiently detailed to contain the date of the transaction as well as descriptions of the products purchased. This illustrates why data warehouses tend to contain very large quantities of time-stamped data.

A less obvious requirement of finding the beer-diaper connection is being able to "browse" through the warehouse without really knowing what you are looking for. In data warehousing, you typically submit many queries before you get results worth analyzing. This means data warehouses must make it reasonably easy for end users to make queries. This, in turn, implies user-friendly access tools and reasonable response times. When you consider user-friendly access of massive amounts of detailed data with reasonable response times, you can appreciate the challenges of providing an effective data warehouse solution.

Scope of This Article

This article compares and contrasts the Inmon and Kimball approaches to meeting the challenges of creating a data warehouse. While it discusses the most basic aspects of both approaches, there are many topics it does not address. For example, the article does not address physical design considerations, such as distributed data warehouse processing. It does not discuss special applications of the data warehouse, such as support of executive information needs, or considerations in creating Web-based data warehouses.

This article does not address some concepts that scholar-practitioners in the industry consider fairly basic, such as metadata, snowflaking, or data mining. These topics have been excluded from the article in order to give more thorough attention to the most basic aspects of each model.

The Inmon Model

Inmon's architected environment consists of all information systems and their databases throughout a given organization. He calls this behemoth the Corporate Information Factory, or CIF (Inmon and Imhoff, 2002). Even a cursory discussion of the CIF is beyond the scope of this article, and therefore the following discussion is limited to those components of Inmon's architected environment most essential to the data warehouse.

Inmon divides the overall database environment of the organization into four levels:

- Operational
- Atomic data warehouse
- Departmental
- Individual

The last three levels comprise the data warehouse. The first level contains data from legacy and other transaction processing systems. This level supports the day-to-day operation of the organization; in other words, the first level supports all transaction processing. From the operational systems, data is extensively manipulated and then moved to the atomic data warehouse (Inmon, 2002). (See "Extract, Transform, and Load" later in this article for an overview of the data manipulation performed between the operational and atomic data warehouse levels.)

Inmon uses an example to illustrate the difference between operational data and data stored in the atomic data warehouse. In the example, the entity is a customer, and the attribute of most interest is the customer's credit rating. The operational system's database contains the customer's current credit rating and related information of interest (such as loan balances, address, etc.) in a single record. The atomic data warehouse, by contrast, contains the credit history for this customer, summarized by year, with one record per year (Inmon, 2002).

Inmon does not thoroughly pursue the customer credit example in its transformation from the atomic to the department level. His example is extended here based on a synthesis of various discussions throughout his book. The data contained in the departmental level is lightly to heavily summarized, depending on a given department's information requirements. The credit department might lightly summarize the data by dropping customer address information as irrelevant, but keeping a "flag" to indicate a change of address. In contrast, the marketing department might more heavily summarize the data by dropping all customer-identifying data except zip code. Each department's database can hold data summarized according to its needs. At the same time, Inmon's architecture ensures that all data is consistent because all departmental data comes from the atomic data warehouse.

Individual users create the fourth and final level of the architected environment when they create heuristic, ad hoc data sets as part of decision support analyses. This fourth level tends to be temporary and housed on the individual user's personal computer (Inmon, 2002). For example, a user working in the credit department might ask to see records for all accounts that have been delinquent at least once in the last three years.

If the department's database has not retained the data at the level of detail needed, it is possible to query the atomic data warehouse. Queries against the atomic data warehouse generally go through the IT department. Inmon argues that the atomic data warehouse is worth the initial effort to construct because it allows the creation of any number of departmental databases without risking creating incompatible data between them (Inmon, 2002). This is done using a three-level data model.

The Three-Level Data Model

Inmon proposes three levels of data modeling. The first is ERD (entity relationship diagrams). Just as in the development of operational databases, ERDs are used to explore and refine entities, their attributes, and the relationships between entities. The development team creates one set of ERDs for each department that is expected to use the data warehouse. The corporate ERD is the sum of all department ERDs (Inmon, 2002).

The second (mid-level) data model, establishes the DIS (data item set) for each department. Again, the sum of the departmental DISs comprise the corporate DIS. The mid-level data model includes four constructs:

- A primary data grouping
- A secondary data grouping
- A connector, signifying the relationships of data between major subject area
- "Type of" data

A critical aspect of the mid-level data model is that the primary grouping exists only once for each major subject area. This means that an ERD created in the first-level data model is the basis for a DIS in the second-level data model. Figure 1, taken from Inmon's book, illustrates the ERD-DIS relationship for a given user view. It also shows

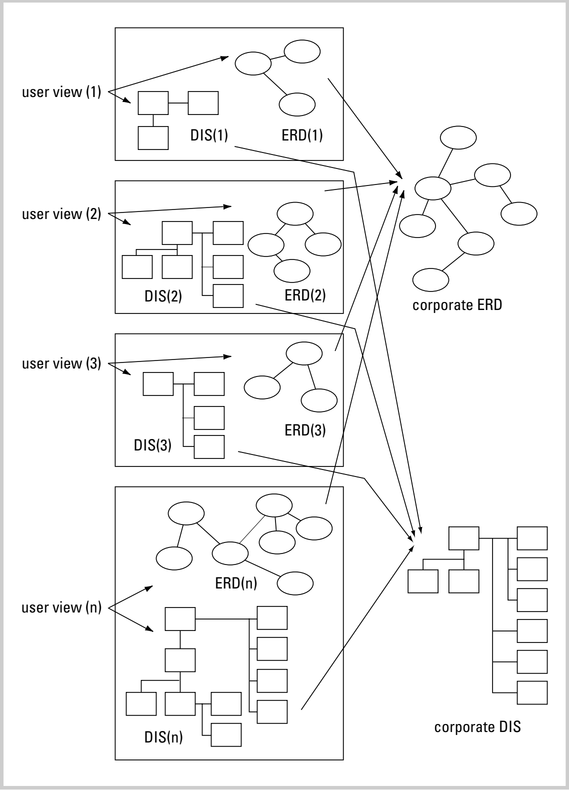


Figure 1. *Relationship between Levels One and Two of Inmon's Data Model (Inmon, 2002)*

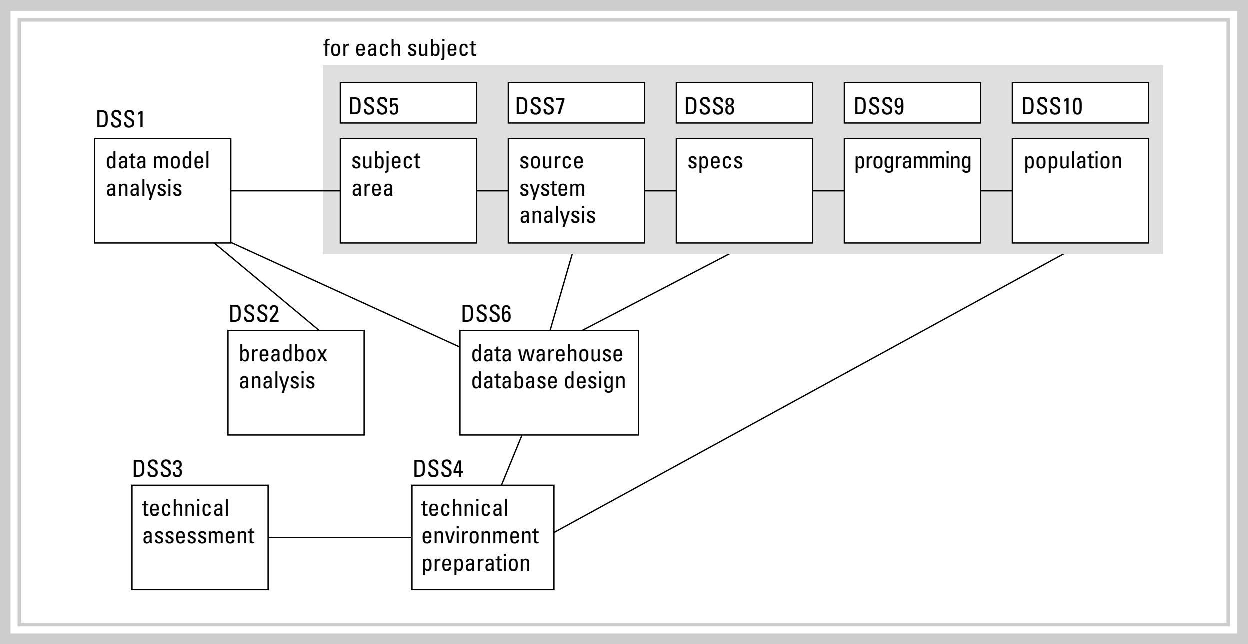


Figure 2. *Inmon's Meth2 (Inmon, 2002)*

how the various user views are combined into a corporate ERD and DIS. Within a DIS, each rectangle represents a logical table in either a departmental or the corporate DIS. The connections between these tables are the same as those that connect entities in the ERDs. Rectangles to the right in a given DIS represent the secondary grouping of data (Inmon, 2002). ("Type of" data does not appear in Figure 1. It would be represented by another column of rectangles branching to the right from the secondary grouping rectangles.)

Inmon's banking example helps make this clear. In banking, the entity "customer" generates a primary grouping of data such as account (primary grouping). "Account" may have several manifestations, such as loan, savings, or trust (secondary grouping). Connectors show that one customer may have several different accounts. Finally, each account may have data generated by similar activities, such as ATM deposits, ATM withdrawals, teller deposits, or teller withdrawals; these are examples of "type of" data (Inmon, 2002).

Creating the departmental and corporate ERDs and DISs shown in Figure 1 requires very high levels of data modeling expertise. It also requires breadth and depth of knowledge of the organization's business processes. Inmon suggests using enterprisewide data models if possible to save development time; they already exist for many industries (Inmon, 2000).

The final level of Inmon's data model is the physical. In his words: "The physical model is created from the mid-level data model merely by extending the mid-level data model to include keys and physical characteristics of the model" (Inmon, 2002). Inmon explains various techniques for optimizing the performance of the data warehouse at both the atomic and departmental levels. Although the techniques may not be familiar (creating arrays of data, preformatting, rejoining tables), the purpose—optimizing I/O performance—is the same as for operational database systems. Most of these techniques involve denormalization of tables.

There are several reasons to denormalize tables at the physical level. For example, records in the atomic data warehouse are rarely updated because the data is historical. This makes it possible to physically place data in ways that would not work for operational data because it is frequently updated (Inmon, 2002).

Once the three-level data model is complete, the data warehouse development has begun.

A Spiral Development Approach (Meth2)

A completed three-level data model is the only prerequisite to using Inmon's special adaptation of the spiral development methodology, which he calls Meth2. (Meth1 is for developing operational systems; Meth3 is for tuning an existing data warehouse). Inmon calls the modeling step DSS1 (for Decision Support 1). He outlines nine more steps, shown in Figure 2.

Using the completed three-level data model as the first input to the process, the next step is to conduct a size/granularity analysis (DSS2 in Figure 2). Granularity is a measure of the detail of the data. For example, transactional data has the lowest level of granularity because it has the most detail. Inmon calls the size/granularity analysis a breadbox analysis, presumably an allusion to the saying: Is it bigger than a breadbox? If the volume of data is massive, then the team needs to consider multiple levels of granularity for the data (Inmon, 2002). This might involve storing some data at a transaction level and other data in summarized forms (such as a daily total).

Once granularity issues are resolved, the first subject area is selected (DSS5). This will become the first departmental database. The team analyzes the source systems of the first subject (DSS7), writes specs (DSS8), code programs (DSS9) and populates the database (DSS10). The atomic data warehouse database design begins concurrently (DSS6). When there is enough information to do so, the team conducts a technical assessment (DSS3). This assessment ensures that the data in the warehouse will be accessible and well managed (Inmon, 2002).

As the team develops each successive departmental database, they impact the atomic data warehouse. Figure 2 shows this iterative aspect of the model by showing lines connecting various steps. Lines connect both the source systems analysis step (DSS7) and the specs step (DSS8) with the atomic data warehouse design (DSS6). This means that the atomic data warehouse design will be revisited each time a new departmental database is developed. The line connecting the population of a departmental database (DSS10) with the preparation of the technical environment (DSS4) also shows the iterative nature of Meth2. By preparation of the technical environment (DSS4), Inmon means making sure that the data warehouses's network, storage hardware, OS, and all interface and access software are ready to receive data (Inmon, 2002).

Being data driven is an essential aspect of Inmon's spiral development methodology. "One of the salient aspects of a data-driven methodology is that it builds on previous efforts—utilizing both code and processes that have already been developed." (Inmon, 2002) His three-level data model helps support a spiral methodology, in that all user views are consistent with the corporate model. The team derives subsequent departmental databases using the code and processes they created when they developed earlier departmental databases. This means the time it takes to produce the second departmental database should be considerably less than the time it took to go through DSS1 through DSS10 for the first departmental database.

Inmon's Philosophy: Evolutionary, Not Revolutionary

Inmon sees the data warehouse as an integral part of the Corporate Information Factory (CIF). This means, among other things, that the data warehouse and operational databases are all part of a larger whole. This perception helps explain why Inmon's data warehouse must adhere to most of the same standards as operational systems. From this premise, it is easy to see how Inmon's evolutionary approach grows out of operational relational database technology and development methods. Each aspect discussed in this article—the architected environment, the three-level data model, and the spiral approach—is consistent with established practices in operational DBMS design and deployment. It is built upon principles and practices that have been in use in the operational database world at least a decade longer than even the earliest data warehouse efforts. Viewed in this context, Inmon's model is much more evolutionary than revolutionary.

A by-product of this evolutionary approach is that Inmon's primary audience is IT professionals, as it takes an IT professional's level of understanding to actively use his tools or development methodology. Inmon's tools and methodology ensure that end users will have mostly passive roles in the development of the data warehouse, reviewing the IT professionals' output.

Kimball's Model

Kimball's model differs in several important respects from a traditional relational database approach. One significant difference is that data warehouses built with the Kimball model use a data modeling method unique to the data warehouse. This is discussed in the next section: "Dimensional Data Modeling."

Another significant difference is that the overall architecture features multiple databases that are expected to be highly interoperable. The data bus is the main design feature that makes this possible (further discussion of the data bus is included in the section "The Data Bus and Conformed Dimensions").

Dimensional Data Modeling

Dimensional modeling may seem strange to IT professionals familiar with traditional relational modeling. Dimensional modeling begins with tables rather than entity-attribute data models such as ERDs. The tables are either fact tables or dimension tables. Fact tables contain metrics, while dimension tables contain attributes of the metrics in the fact tables. Dimension tables routinely contain repeating groups; this violates normalization rules. However, dimensional modeling violates normalization rules in order to achieve a high level of performance in the data warehouse, while keeping it end-user accessible.

An example best illustrates how dimensional modeling meets the dual objectives of ease of use and performance. The first example in Kimball's book is a retailing data warehouse (Kimball, 2002). One fact table from this example is the Daily Product Sales table. This table contains five columns: the product key, store key, date key, quantity sold, and dollar sales amount. The dimension tables in this example include the Date Dimension, Store Dimension, and Product Dimension tables.

Fact tables contain many rows and relatively few columns; this is essential to ease of use and query performance. The number of rows in Daily Product Sales table can be estimated using formulae. While explaining the formulae is beyond the scope of this article, they basically involve assumptions regarding the number of different products sold in each store each day. Kimball estimates that the Daily Product Sales fact table is likely to contain millions of rows, and be about 10 GB or more (Kimball, 2002). Although the table has only five columns, adding just one additional column would increase the file size by 2 GB! This example makes it easy to grasp the importance of keeping the number of columns in fact tables as small as possible.

In contrast, the dimension tables are likely to have only hundreds or thousands of rows (rather than millions), and be only megabytes in total size (Kimball, 2002). Unlike fact tables, dimension tables may have a hundred columns or more. This is because they contain all the attributes of the data in the fact table in highly denormalized forms. Following along with the retailing example, the primary key of the Product Dimension table is the product key. The rest of the dimension table's columns are attributes of product. These include product description, brand description, package type description, department description, package size, weight, shelf life, shelf width, shelf height, and many more. The Date Dimension and Store Dimension tables also have large numbers of columns, but relatively few rows.

It is easy for end users to query the database because virtually all the ways of summarizing the data is already in the dimension tables. This goes a long way toward meeting the ease of use goal. In terms of meeting the performance goal, Kimball says:

> A database engine can make very strong assumptions about first constraining the heavily indexed dimension tables, and then attacking the fact table all at once with the Cartesian product of the dimension table keys satisfying the user's constraints.

Dimensional modeling is a data modeling approach that capitalizes on the unique requirements of the data warehouse. Keeping fact tables to a small number of rows and allowing dimension tables to be highly denormalized are both essential. The resulting data mart is highly accessible to the end user and provides reasonable query response times.

The Data Bus and Conformed Dimensions

In Kimball's architecture, data is copied from operational source systems to a staging area. In the staging area, the data is scrubbed, that is, made consistent and suitable for end-user queries. (The scrubbing process is discussed in "Extract, Transform, and Load" later in this article.) From the staging area, data is loaded into data marts. The data marts are the source of data for user queries.

Each data mart is based on a single business process. Some examples of business processes are point of sale (retail sales), inventory (from receiving dock to point of sale), procurement, and order management. More than one department may be interested in a given business process; therefore, no one department is perceived as the sole owner of a given data mart (Kimball, 2002).

The data warehouse bus is the part of Kimball's architecture that allows the sum of the data marts to truly be an integrated whole—a data warehouse. The bus architecture is another way of saying that all data marts must use standardized conformed dimensions. The basic requirements of conformed dimensions are that keys, column names, attribute definitions, and attribute values are consistent across business processes. Put another way, two dimensions are conformed "when they are exactly the same, or one is a perfect subset of the other. Most important, the row headers produced in answer sets from two different conformed dimensions must be able to be matched perfectly" (Kimball, 2002). This may seem an impossible set of requirements, but a knowledge of dimensional data modeling and adherence to the four-step dimensional design process help keep the requirements manageable.

An example of using conformed dimensions across business processes will help make clear how these requirements can be met without superhuman efforts. One data item that spans multiple business processes is the product dimension. The primary key for the product is an artificial key assigned during the ETL process. The first data mart development defines the product key, and all subsequently developed data marts must use the key. This ensures that queries can be made across data marts without conflicting results. For example, product 18874002 is the same to a user interested in patterns of the product's movement through the warehouse as it is to the user interested in the relative success of a promotion for the product. In other words, conformed dimensions help ensure that product data refers to the same product in the retail sales data mart as in the inventory data mart.

The Four-Step Dimensional Design Process

Kimball recommends a development methodology that is unique to data warehousing. It involves a bottom-up approach, which in the case of data warehouses means to build one data mart at a time. The four steps of the dimensional design process are:

- Select the business process
- Declare the grain
- Choose the dimensions
- Identify the facts

Kimball defines business processes quite broadly. Examples include point of sale (POS) retail sales, inventory, ordering, and shipments, all of which cross department lines in most organizations. For example, the ordering process is of interest to sales, marketing, finance, and inventory control personnel. To choose the first business process for the data warehouse project, select the process that has "the most impact—it should answer the most pressing business questions and be readily accessible for data extraction" (Kimball, 2002).

Declaring the grain is the process of deciding what level of detail the data warehouse will contain. The lowest level of granularity is called atomic, meaning that it cannot be further subdivided. Choosing a grain at the atomic level is highly desirable, since users can always aggregate the data

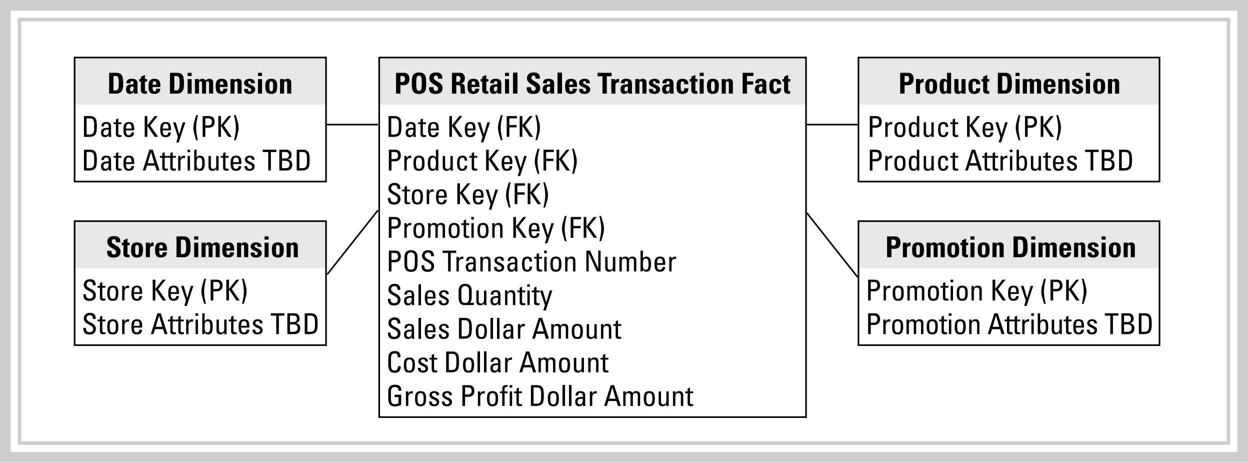


Figure 3. *Sample Kimball Fact and Dimension Tables (Kimball, 2002)*

as desired. Choosing a more summarized level means queries below that level cannot be fulfilled by the data warehouse. In Kimball's words:

> Preferably you should develop dimensional models for the most atomic information captured by a business process… A data warehouse almost always demands data expressed at the lowest possible grain of each dimension. (Kimball, 2002)

In the retail example, the grain declared is an individual line item on a POS transaction. The possibilities for analyzing data with this level of granularity are virtually unlimited. It allows for the discovery of non-obvious relationships in retail sales, such as the beer-and-diaper relationship (as discussed previously). Atomic data granularity provides decision support for virtually every aspect of retail sales. Examples include evaluation of promotions, expansion or contraction of product lines, and cannibalization of the sales of one product due to the promotion of another.

With the grain declared, the next step is to choose dimensions. In the retail example, the dimensions include date, store, product, and promotion. Each of the dimension tables has a large number of attributes. The date dimension table includes many attributes that would make a relational data modeler shudder, including Day Number in Epoch, Week Number in Epoch, Month Number in Epoch, Day Number in Calendar Month, and so on. Kimball justifies this highly denormalized table by pointing out that ten years' worth of such data generates only approximately 3,650 rows and a file measured in kilobytes (Kimball, 2002).

The fourth and final step is to determine which facts to include in the fact tables. In the retail example, Kimball chooses to include some computed values as well as truly atomic values, making queries easy for the end user and providing acceptable data warehouse performance. The values in retail sales fact table are: Date, Product, Store, Promotion, POS transaction number, Sales quantity, Sales dollar amount, Cost dollar amount, and Gross profit dollar amount (Kimball, 2002). Including the gross profit dollar amount is an example of improving performance while violating traditional relational database rules. Users frequently query the data warehouse for gross profit. Therefore including this computed value in the fact table improves query performance.

The result of the four-step process is shown with minimal detail in Figure 3. Sample Kimball Fact and Dimension Tables. The fact table is shown with all the facts, but the dimension tables are shown only with their primary keys. Each of the dimension tables shown in the figure has dozens of dimensions. The wealth of dimensions allow end users to compose virtually unlimited queries.

Basics of Kimball's Data Warehouse Philosophy

Kimball's philosophy shines through every chapter of his book. The business requirements drive both the process and the nature of the data warehouse. In the first chapter, he defines the goals of a data warehouse (Kimball, 2002):

- Make information easily accessible
- Present the organization's information consistently
- Be adaptive and resilient to change
- Protect information
- Serve as the foundation for improved decision making

He ends his list with a warning, masquerading as a goal: "The business community must accept the data warehouse if it is to be deemed successful" (Kimball, 2002). To Kimball, acceptance is measured by how much the data warehouse is used, which is directly related to its user-friendliness. This proactive stance in "designing in" user-friendliness is essential to Kimball's philosophy. Kimball's four-step development methodology is easy enough for the end user to actively participate. The example retail sales dimensional model (Figure 3) shows the user-friendly nature of the final form of the data mart. Both the attribute names and the relationships between the fact table and dimension tables are very familiar to users who need to query retail sales data.

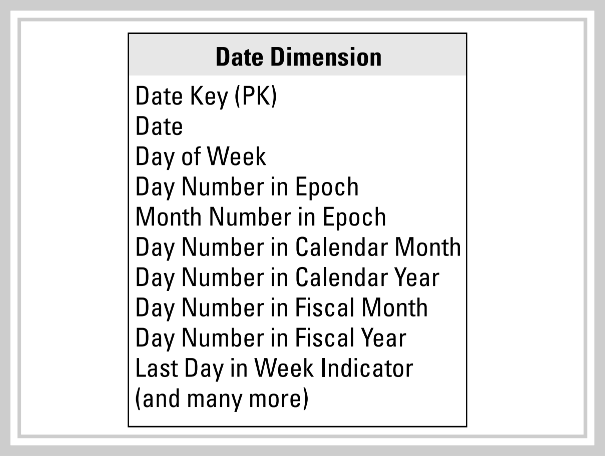
Similarities and Differences: Inmon versus Kimball

Similarities

The most prominent similarities between Inmon's and Kimball's models are the use of time-stamped data, and the extract, transform, and load (ETL) process. Although the execution of these two elements differs between the two models, the data attributes and query results are very similar.

Similar Time-Stamped Data

Operational systems' databases generally carry detailed data for "anywhere from one week to two years" (Inmon, 2002). In contrast, the data warehouse stores data for five or even ten years. The time attribute is arguably the most important defining characteristic of data warehouse data. This is so because it is the time attribute that allows decision support analyses to compare this year's sales of Product X with last year's, or to determine whether more of Product X is sold on weekend than on holidays. So, how the time attribute is captured is critical, because it controls which analyses are possible and which aren't.



Date Dimension
Date Key (PK) Date Day of Week Day Number in Epoch Month Number in Epoch Day Number in Calendar Month Day Number in Calendar Year Day Number in Fiscal Month Day Number in Fiscal Year Last Day in Week Indicator (and many more)

Figure 4. *Kimball's Date Dimension (Kimball, 2002)*

Kimball calls the time attribute the "date dimension;" Inmon calls it the "time element." Figure 4. Kimball's Date Dimension shows a range of possibilities for time attributes for a retail sales data mart. In Kimball's example, the date key is an artificial key that defines a conformed dimension. In an Inmon example, the same attributes would either be contained in several different, more normalized tables or simply be calculated at the time of the user query. The choice of storing versus calculating in Inmon's model would be guided by performance considerations. Whether using an Inmon- or Kimball-based approach, however, end users are able to query the data by day, month, quarter, year, holiday, weekday, weekend, etc.

Similar Extract, Transform, and Load (ETL)

The data warehouse environment begins with (ETL). Data is extracted from operational databases, transformed to meet the data warehouse's standards, and loaded. The data is loaded into either the monolithic data warehouse (á la Inmon) or into a series of smaller databases called data marts (á la Kimball).

The ETL process has spawned a market niche, and there are a number of ETL tools available today. Some are add-ons to DBMSs, such as the ETL tools Oracle provides. Others are DBMS-independent. Even a cursory comparison of ETL tools is beyond this scope of this article. Instead, we will discuss the general process and scope of the ETL activity and its importance to the data warehouse.

Extract: The first part of ETL—extract—involves moving data from operational systems to a persistent staging area. Issues of timing of the extraction can be important in the extract process, in that different systems may make a given data item available at different times. It is also important to know how the operational systems handle exceptions and updates, since once data enters the data warehouse, it is rarely subject to updating.

Transform: Once data is extracted into the staging area, it is ready for the transformation portion of ETL. A simple transform example is renaming a data item, such as when two different operational systems call a single data item by a different name. A more complex example is adjusting the value of a data item, such as when two different operational systems measure a product or process differently and therefore assign a different value to the same data item. In general, the purpose of the transform processes is to ensure data integrity within the data warehouse. There are several methods used to transform data, including field mapping and algorithmic comparisons.

Load: The final step in ETL is loading the data into either the atomic data warehouse (in Inmon's model) or into data marts (in Kimball's model). The load process in either case involves placing the data physically. The main concern in this process is appending the newly extracted and transformed data onto the data already in the data warehouse. Various ETL routines run at this point help ensure data integrity and guard against data redundancy.

ETL is essential to the viability of the data warehouse in that it attempts to ensure data integrity within the data warehouse. Obviously, if two user queries that are essentially the same return two different results, the credibility of the data warehouse is damaged in the eyes of the users. Because operational systems are seldom (if ever) designed to produce results compatible with one another, making the output of these systems consistent is generally a

	Inmon	**Kimball**
Methodology and architecture		
Overall approach	Top-down	Bottom-up
Architectural structure	Enterprisewide (atomic) data warehouse "feeds" departmental databases	Data marts model a single business process; enterprise consistency achieved through data bus and conformed dimensions
Complexity of the method	Quite complex	Fairly simple
Comparison with established development methodologies	Derived from the spiral methodology	Four-step process; a departure from RDBMS methods
Discussion of physical design	Fairly thorough	Fairly light
Data modeling		
Data orientation	Subject- or data-driven	Process oriented
Tools	Traditional (ERDs, DISs)	Dimensional modeling; a departure from relational modeling
End-user accessibility	Low	High
Philosophy		
Primary audience	IT professionals	End users
Place in the organization	Integral part of the Corporate Information Factory (CIF)	Transformer and retainer of operational data
Objective	Deliver a sound technical solution based on proven database methods and technologies	Deliver a solution that makes it easy for end users to directly query the data and still get reasonable response times

Table 1. *Comparison of Essential Features of Inmon's and Kimball's Models*

Herculean effort. Not surprisingly, ETL is frequently considered the most labor-intensive data warehouse activity, surpassing even decision support analysis activities!

Differences

The differences between Inmon's and Kimball's approaches are many and deep. It is interesting to note that the two features that create similarities between the two models—time-stamped data and ETL—are required to make decision support systems viable. In other words, the two models are similar only in the areas in which, arguably, they have to be similar. In all other areas, their differences are profound.

The most essential differences between the two models are in the areas of development methodologies, data modeling, and data warehouse architectures. Table 1 summarizes these differences. Following the table, each major area of difference is discussed in detail.

Differences in Development Methodologies and Architectures

In order to have an atomic data warehouse, as in Inmon's model, some degree of top-down development must be present. The atomic data warehouse must serve the entire enterprise, and all departmental databases obtain their data through the atomic data warehouse. Top-down development efforts have a certain unavoidable degree of complexity, and Inmon's methodology is no exception, although his clear presentation helps it seem less complex.

Overall, Inmon's methodology and architectural orientation is a technical one. His primary interest is ensuring that the technical solution works. Oversimplified, the objective of this technical solution is to optimize I/Os. Inmon's audience is clearly comprised of IT professionals. Few business readers have the background to understand Inmon's development approach because of its emphasis on technical aspects and a lack of understanding of the spiral development approach on which it is based. His emphasis on the technical aspects of the development implies that the IT department members of the data warehousing team will feel the greatest degree of ownership of the data warehouse as they, not the end users, will understand the development methodology.

In contrast, Kimball's four-step development methodology is very accessible to the end user. A user can even understand moderately technical concepts of the data bus and conformed dimensions without extensive study, in contrast to learning to interpret ERDs. By definition, a bottom-up approach involves fewer data elements than a top-down development. Even if users are unfamiliar with the concept of a business process, the smaller scope of the data mart is more accessible to end users. Inmon's Meth2 helps make the enterprisewide scope less daunting, but the data mart scope is still considerably easier for users to grasp.

Differences in Data Modeling

Two obvious ways in which Inmon's and Kimball's data modeling differ are (1) orientation toward the data and (2) modeling rules and techniques.

In his own terms, Inmon takes a subject-oriented or data-driven approach to data modeling. This means that the nature of the data directs the data modeling process. This fits well with Inmon's traditional data modeling tools, such as ERDs and DISs. It also means that the IT members of the data warehouse team will have primary responsibility for data modeling, because the modeling tools and the thought processes they involve require a technical background to use effectively. End users can attend review presentations, but few could review ERDs or DISs unassisted unless they received fairly extensive special training.

In contrast, Kimball takes a process orientation, meaning that data modeling becomes an attempt to define the interaction of data across a business process (such as retail sales or inventory). By their natures, such business processes usually cross departmental lines. This fits well with the new data modeling approach of dimensional data modeling, in which the process determines which metrics (facts) and attributes (dimensions) are important enough to claim a place in the data warehouse. Dimensional modeling tools allow end users to take an active role in the data modeling process.

Philosophical Differences

By now it is clear that Inmon views IT as the primary developer and provider of the data warehouse. Inmon believes that the performance of the completed data ware-

Characteristic	Favors Kimball	Favors Inmon
Nature of the organization's decision support requirements	Tactical	Strategic
Data integration requirements	Individual business areas	Enterprisewide integration
Structure of data	Business metrics, performance measures, and scorecards	Non-metric data and for data that will be applied to meet multiple and varied information needs
Scalability	Need to adapt to highly volatile needs within a limited scope	Growing scope and changing requirements are critical
Persistency of data	Source systems are relatively stable	High rate of change from source systems
Staffing and skills requirements	Small teams of generalists	Larger team(s) of specialists
Time to delivery	Need for the first data warehouse application is urgent	Organization's requirements allow for longer start-up time
Cost to deploy	Lower start-up costs, with each subsequent project costing about the same	Higher start-up costs, with lower subsequent project development costs

Table 2. *Specific Characteristics Favoring Inmon's or Kimball's Model*

house will be maximized by ensuring a technically oriented development process. Meanwhile, Kimball sees end users and IT professionals sharing duties roughly equally. By ensuring the active participation of end users throughout the development process, the likelihood of user acceptance of the completed data warehouse is greatly enhanced.

Of course, both of these experts are well aware that a data warehouse that doesn't involve the users at all points in its lifecycle is just as likely to fail as one that performs poorly for the users. What the two do not agree upon is which of these considerations should be considered the most important.

Choosing the Best Approach

Following are guidelines for determining whether Inmon's or Kimball's approach is best suited to organization's data warehousing needs. Dave Wells addressed this problem in a *TDWI FlashPoint* article in early 2003 (Wells, 2003). He proposes 12 evaluation criteria that focus on the needs, environment, culture, and technical expertise of an organization planning to create a data warehouse. Of the 12, eight can be relatively easily categorized as favoring either Kimball's or Inmon's approach. Whether the remaining four elements (cost to operate, consistency of metadata and business rules, sustainability, and technology requirements) favor Kimball's or Inmon's approach would depend on the implementation of a given data warehouse project.

To at least partially summarize the data in Table 1, an organization is more likely to succeed using Inmon's approach if it has a large team of data warehouse specialists, plans a large project with enterprisewide access needs, stores data that is not primarily business metrics, and can wait to see results over a longer timeframe—from four to nine months (Inmon, 2000). These characteristics and data requirements fit well with Inmon's recommendation to first build a considerable infrastructure on a solid enterprisewide data model.

On the other hand, an organization with different characteristics may be better off with a Kimball-based approach. According to one expert, "A typical requirement is to develop an operational data mart for a specific business area in 90 days, and develop subsequent data marts in 60 to 90 days each" (Mimno, 2002). Kimball's approach is generally recognized as faster than Inmon's, at least for the delivery of the first data mart (versus the first departmental database using Inmon's approach). Kimball's approach is also indicated if the organization is better able to field smaller teams of generalists for data warehouse project development, and expects to store mostly business metrics. An organization with these characteristics and requirements is more likely to succeed with a data mart architecture developed using the dimensional modeling approach.

It is important to realize that choosing an approach to data warehousing is not as simple as the two preceding paragraphs imply. However, as long as the reader understands that these guidelines represent a gross oversimplification of the process, they may be useful as a starting point for discussing the data warehousing needs and characteristics unique to a given organization.

Finally, research shows that having the right set of soft skills is just as important, if not more important, than technical skills and knowledge.

Interestingly, the keys to success are not technical in nature. Projects don't succeed because they use an innovative design or radical new technology. They succeed because of the "soft" stuff—leadership, communication, planning, and interpersonal relationships (Eckerson, 2003).

When building a data warehouse, whether using Inmon's or Kimball's approach, it is critical that the data warehouse team employ soft skills liberally and effectively. This involves ensuring that the organization has a well-articulated vision of the data warehouse's role and usage, and allocates sufficient resources to create and maintain the data warehouse (Eckerson, 2003). These are not typically responsibilities that an IT project development team must shoulder, yet they are critical to the success of a data warehouse project.

Summary

Data warehouses require storage and access of massive amounts of time-stamped data for decision support. Since the building of data warehouses was first attempted in the early 1990s, two models have emerged as dominant: Inmon's and Kimball's.

Inmon's approach stresses top-down development using proven database development methodologies and tools, such as ERDs, DISs, and a modification of the spiral development approach. Inmon's tools and methods are adaptations of traditional tools and methods for operational database development. Inmon sees the data warehouse as a part of a much larger information environment, which he calls the Corporate Information Factory (CIF). To ensure that the data warehouse fits well in this larger environment, he advocates the construction of both an atomic data warehouse and departmental databases.

Inmon's approach is evolutionary rather than revolutionary. His tools and methods can be actively used only by IT professionals. End users have a more passive role in the development process, mostly reviewing the results generated by IT professionals. Inmon's attention to the technical aspects of the data warehouse development process increases the chances of a sound technical solution. For end users, this is likely to mean very good query response times.

Kimball's approach is a departure from traditional database development. His bottom-up approach recommends building one data mart per business process. The sum of all data marts is the organization's data warehouse. The data bus is the aspect of Kimball's architecture that ensures interoperability between various data marts. The data bus requires that all data marts are modeled within consistent data standards called conformed dimensions.

Kimball recommends a four-step development process for each data mart, in which dimensional data modeling plays a central role. Dimensional data modeling involves fact tables that contain metric data, and dimension tables that modify that data. Dimensional modeling tools can be actively used by end users with some special training. This helps ensure that end users are actively involved in the development of the data warehouse. Ease of use and reasonable query response times in the final product (the data mart) are the dual goals of dimensional data modeling.

Inmon's and Kimball's models are similar in some ways, such as the treatment of time-stamped data. Although there are some differences in the ways in which each model handles this challenge, the two models are more similar than not in modeling the time attribute. Likewise, both models address the challenges of massaging operational data similarly. This process, called ETL, is one of the most labor-intensive aspects of the data warehouse.

Noted data warehouse expert Dave Wells suggests characteristics of organizations that favor the adoption of either Inmon's or Kimball's models. Some of these characteristics include the organization's decision support requirements, staffing and skills requirements, time to delivery and cost to deploy. His advice can help organizations begin the process of choosing an approach to developing their data warehouse.

Other research suggests that success in developing a data warehouse relies as much on the soft skills of the data warehouse team as on its technical expertise or business acumen. It is not surprising that a large IT-related project needs "…leadership, communication, planning, and interpersonal relationships" in order to succeed (Eckerson, 2003). What makes the data warehouse more of a challenge than a comparable operational development project is that the data warehouse technology is relatively new. A development team with a sound understanding of Inmon's and Kimball's models is in a much better position to articulate a vision of the data warehouse that matches the organization's characteristics and decision support goals.

REFERENCES

Albert, G. The Importance of Data Warehousing (May 03, 2000), BusinessLine, Internet edition, division of The Hindu Business Line. Retrieved August 12, 2003, from http://www.blonnet.com/businessline/2000/05/03/stories/150339m6.htm.

Eckerson, W. Smart Companies in the 21st Century: The Secrets of Creating Business Intelligence Solutions, (April 2003), TDWI Web site. Retrieved August 11, 2003, from http://www.dw-institute.com/research/.

Inmon, W.H. **Building the Data Warehouse** (Third Edition), New York: John Wiley & Sons, (2002).

Inmon, W.H. and C. Imhoff. Corporate Information Factory Components, (2002), Inmon Associates Inc. Web site. Retrieved September 9, 2003 from http://www.billinmon.com/library/articles/artcifco.asp.

Inmon, W.H. Accelerating the Development of the Enterprise Data Warehouse, (2000), Inmon Associates Inc. Web site. Retrieved September 10, 2003 from http://www.billinmon.com/library/presents/present.asp.

Kimball, R. and M. Ross. **The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling** (Second Edition), New York: John Wiley & Sons, 2000.

Ladley, J. And You Thought It Was Safe to Go Back into the Water?, (2003), TDWI FlashPoint, TDWI Web site. Retrieved September 7, 2003, from http://www.dw-institute.com/research/.

Mimno, P. Project Plan for Bottom-Up Development, (August 28, 2002), TDWI FlashPoint, TDWI Web site. Retrieved August 8, 2003, from http://www.dw-institute.com/research/display.asp?id=6425&t=y.

Mimno, P. et al. Ten Mistakes to Avoid in Bottom-Up Development, (June 18, 2003), TDWI FlashPoint, TDWI Web site. Retrieved September 9, 2003, from http://www.dw-institute.com/research/.

Wells, D., Choosing the Right Data Warehousing Approach, (January 3, 2003),TDWI FlashPoint, TDWI Web site. Retrieved July 24, 2003, from http://www.dw-institute.com/research/.

Wells, D. Making Sense of the Methodology Debate, (August 27, 2003a), TDWI FlashPoint, TDWI Web site. Retrieved September 7, 2003, from http://www.dw-institute.com/research/.

Whiting, R. Startup Netezza Pushes Discount Data Warehouse Products, (September 23, 2002), CommWeb, CMP Media's Web site, division of United Business Media. Retrieved August 11, 2003, from http://www.commweb.com/article/IWK20020920S0010.