Facts and Fables About Dimensional Modeling

A Kimball Group White Paper

KIMBALL GROUP
Consulting | Kimball University

In this white paper, we tackle misunderstandings about dimensional modeling that frequently appear in DW/BI industry publications, training, and marketing materials. Unfortunately these issues, rooted in incomplete or misleading information, often lead to erroneous design decisions.

Fable: Dimensional data warehouses are appropriate for summary level data only.

This fable is an echo from data warehousing in the early 1990s, when 10 gigabytes was a big number. In those days everyone talked about summarizing data before loading it into a data warehouse. This hasn't been best practice–for a dimensional data warehouse or any other kind–for more than a decade. Fine grained atomic data has proven to be the most robust data possible. By definition, such data can withstand every possible "ad hoc attack" from users seeking to constrain their queries using highly specific attributes. You should build every data warehouse at the finest grain possible. Even server-based OLAP systems have become robust enough to hold this detail. The beauty of exposing all the levels of data in a dimensional format is that you can start the drill-down process at an aggregated level, but then smoothly descend all the way to granular detail using the same BI tool.

Fable: Dimensional models presuppose the business question and are therefore inflexible.

Dimensional models with atomic data are independent of the business question; they are the most flexible and symmetrical framework for presenting business data.

Fable: Dimensional models are departmental, labeled with the vocabulary most familiar to each department.

Atomic dimensional models should be structured based on business processes (such as orders, shipments and payments), not organizational business departments. Each core business process subject area captures/generates unique performance metrics with unique granularity. Common dimensions are reused across the process-centric dimensional models. If schemas are created on a departmental basis for Finance, Marketing and Sales, then the same atomic metrics are replicated repeatedly for each department. What's the likelihood that the metrics are consistently defined, labeled and populated in the departmental data stores? A departmental approach is highly vulnerable to inconsistent, non-integrated point solutions. Process-centric dimensional models deliver a single version of the truth.

Fable: Since dimensional models are built with a singular focus on a specific group of users or requirements, new dimensional models must be built to accommodate new or additional business requirements.

This misconception stems from erroneously combining two other fables: dimensional models are built by business department and dimensional models contain only summarized data. Building dimensional models to support business processes with atomic detail result in an environment that can respond to a wide variety of requirements. While you must create new fact tables when incorporating new business processes into the data warehouse environment, it is not necessary to build new schema to present the same data to different users or reporting requirements.

Fable: Bringing a new data source into a dimensional data warehouse breaks the existing schemas and requires creating new fact tables.

If the new data source presents data at the same grain (level of detail) as an existing fact table, then the new data source can be gracefully added to that fact table without altering any existing applications. This is one of the great strengths of dimensional modeling since there are a set of documented and well defined "graceful" modifications that have this characteristic. If the new data source is at a different grain, then a new fact table must be created, but this has nothing to do with the modeling approach. All data representations must create a new entity when a new table with different keys is introduced.

Fable: A good way to narrow the scope and control the risk of data warehouse development is to focus on delivering the single report most requested by business users.

Starting with a specific report is a terrible way to build a data warehouse. Data warehouse development risk is concentrated almost entirely in sourcing and transforming the needed data. A high profile report, such as customer profitability or customer satisfaction, may require a dozen different data sources. The user's expectations for this report are likely unrealistic; they assume the data warehouse will erase all their existing problems. It is far better to roll out a succession of dimensional models, each based on individual business process sources of data, and use the enterprise data warehouse bus architecture to gradually provide the components of desired high profile reports over time.

Fable: Dimensional models are fully denormalized.

Dimensional models combine normalized and denormalized table structures. The dimension tables of descriptive information are highly denormalized with detailed and hierarchical roll-up attributes in the same table. Meanwhile, the fact tables with performance metrics are typically normalized. While we advise against a fully normalized dimension with snowflaked dimension attributes in separate tables, a single denormalized "big wide table" containing both metrics and descriptions in the same table is also ill-advised.

Fable: Normalizing data is a prerequisite for data integration.

Normalization does not deliver integration; at best, it forces the data analyst to confront the inconsistencies. Integration requires organizational agreement on matching rules, domain values, and standard labels. Reaching agreement or conformance is the tough part of data integration; normalization is merely a structure for storing the agreed upon results.

Fable: Dimensional data is organized differently from relational data.

This is like saying "a Ford and a car are different." The fable results from confusion between the terms normalized and relational. Normalization is a modeling approach to support high volume transactions in a relational database environment. It removes redundancy to process transactions quickly. A dimensional model is designed to support analytical queries and user access. These queries typically involve selecting and aggregating arbitrary subsets of data; rarely do they involve inserts or updates. Normalized and dimensional models are simply different design approaches to solve different problems, but both can be implemented in a relational database.

Fable: Instead of deploying Kimball's conformed dimensions which require people to use the same, consistent names for data, a metadata repository can used to equate data elements having different names in different sources.

When faced with data integration challenges, some designers believe that a simple intermediate data structure is all that's needed to "perform translation on the fly." Unfortunately, true data integration supporting integrated "drill across" reports can only succeed if the textual descriptors (fields) in each separate source are physically altered so they have the same label (column name) and content (data domain values).

Fable: Dimensional data warehouses are passé. You can cost-effectively substitute the power of a database platform for the design and transformation work required to build dimensional data models.

Any data warehouse worthy of the name has clean data that fully tracks history. We recommend dimensional models for two reasons: they are easy for business users to understand and navigate, and they are efficient to query. If you have enough computing horsepower, why not use views or some other logical layer to present a dimensional view to users, but keep data in a normalized format that's similar to the source systems and hence easy to maintain? Feasible: probably. Cost-effective: probably not. First, why waste server cycles at query time? It's more efficient to perform the restructuring work once when the data are loaded. Second, you're not simplifying the overall system. Instead, you're shifting the workload from ETL system developers to the BI front end developers, who need to figure out how to make a normalized schema look like a dimensional model that users can understand.

Fable: Attributes such as employee age or gender should be treated as degenerate dimensions in the fact table rather than as employee dimension attributes.

Don't allow textual attributes to clutter fact tables under the guise of degenerate dimensions. Degenerate dimensions are typically reserved for operational control numbers such as invoice, purchase order, or check payment numbers. You'll encounter them in transactional fact tables as dimensional keys that don't join to actual dimension tables.

Fable: A changing descriptive attribute (as in slowly changing dimensions) is only a problem for dimensional designs.

Time variance is a fundamental issue that must be dealt with in any data warehouse. When the description of a fundamental entity like customer or product changes, the data warehouse must have a systematic approach for recording the change.

Dimensional modeling deals with time variance with the standard design technique known as slowly changing dimensions (SCDs). When normalized models step up to the issue of time variance, they typically add time stamps to the entities in various configurations. These time stamps serve to capture every entity change (just like a type 2 SCD does), but without using a surrogate key for each new row, the query interface must issue a double-barreled join that constrains both the natural key and the time stamp between every pair of tables that must be joined. Not very business user friendly, is it?

Fable: Data mining and statistical methods cannot be utilized if data is structured in dimensional models.

Data mining is most effective when presented with a rich set of transactional data. Virtually all data mining technologies expect to read data into their environment from a flat observation table. They make no assumptions regarding how the data was structured before loading into the tool. An atomic, transaction-grained dimensional model is an excellent source of data for data mining and statistical analysis.

Fable: The primary key of a fact table consists of all the referenced dimension foreign keys.

A fact table often has 10 or more foreign keys joining to the dimension tables' primary keys. However, only a subset of the fact table's foreign key references is typically needed for row uniqueness. Most fact tables have a primary key that consists of a concatenated/composite subset of the foreign keys.

Fable: Data should be "application neutral" in the data warehouse, meaning the data model should not be built for a specific BI application, such as product profitability.

Providing atomic data in a dimensional data warehouse delivers maximum flexibility because it doesn't pre-suppose the business question. Product profitability, however, is in a different league. You can't expect business analysts to transform raw atomic data into profitability metrics by performing cost allocations on the fly at query time. For complex, cross-process applications like profitability analysis, the data should be structured in a procedural subsystem of the ETL back room, allowing the business users to execute simpler queries with consistent results.

Fable: In a bus architecture, there would be no persistent intermediate database; all ETL processing would be handled by the ETL programs without "parking" the data between source and target.

The data warehouse bus architecture is a specific framework for integrating data from a variety of sources by using conformed dimensions. Data warehouses based on the bus architecture routinely stage the data after extraction, cleaning, and final transformation.

Fable: Ralph Kimball invented the fact and dimension terminology.

While Ralph played a critical role in establishing these terms as industry standards, he didn't "invent" the concepts. As best as we can determine, the terms facts and dimensions originated from a joint research projected conducted by General Mills and Dartmouth University in the 1960s. By the 1970s, both AC Nielsen and IRI used these terms consistently when describing their syndicated data offerings. Ralph first heard about "dimensions," "facts," and "conformed dimensions" from AC Nielsen in 1983 as they explained their dimensional structures for simplifying the presentation of analytic information.