Dimensional Modeling and E-R Modeling In The Data Warehouse

By

Joseph M. Firestone, Ph.D.

White Paper No. Eight

June 22, 1998

Introduction

Dimensional Modeling (DM) is a favorite modeling technique in data warehousing. In DM, a model of tables and relations is constituted with the purpose of optimizing decision support query performance in relational databases, relative to a measurement or set of measurements of the outcome(s) of the business process being modeled. In contrast, conventional E-R models are constituted to (a) remove redundancy in the data model, (b) facilitate retrieval of individual records having certain critical identifiers, and (c) therefore, optimize On-line Transaction Processing (OLTP) performance.

Practitioners of DM have approached developing a logical data model by selecting the business process to be modeled and then deciding what each individual low level record in the "fact table" (the grain of the fact table) will mean. The fact table is the focus of dimensional analysis. It is the table *dimensional queries segment* in the process of producing solution sets. The criteria for segmentation are contained in one or more "dimension tables" whose single part primary keys become foreign keys of the related fact table in DM designs. The foreign keys in a related fact table constitute a multi-part primary key for that fact table, which, in turn, expresses a many-to-many relationship. [1]

In a DM further, the grain of the fact table is usually a quantitative measurement of the outcome of the business process being analyzed. While the dimension tables are generally composed of attributes measured on some discrete category scale that describe, qualify, locate, or constrain the fact table quantitative measurements.

Since a dimensional model is visually represented as a fact table surrounded by dimension tables, it is frequently called a star schema. Figure One is an illustration of a DM/star schema using a student academic fact database.

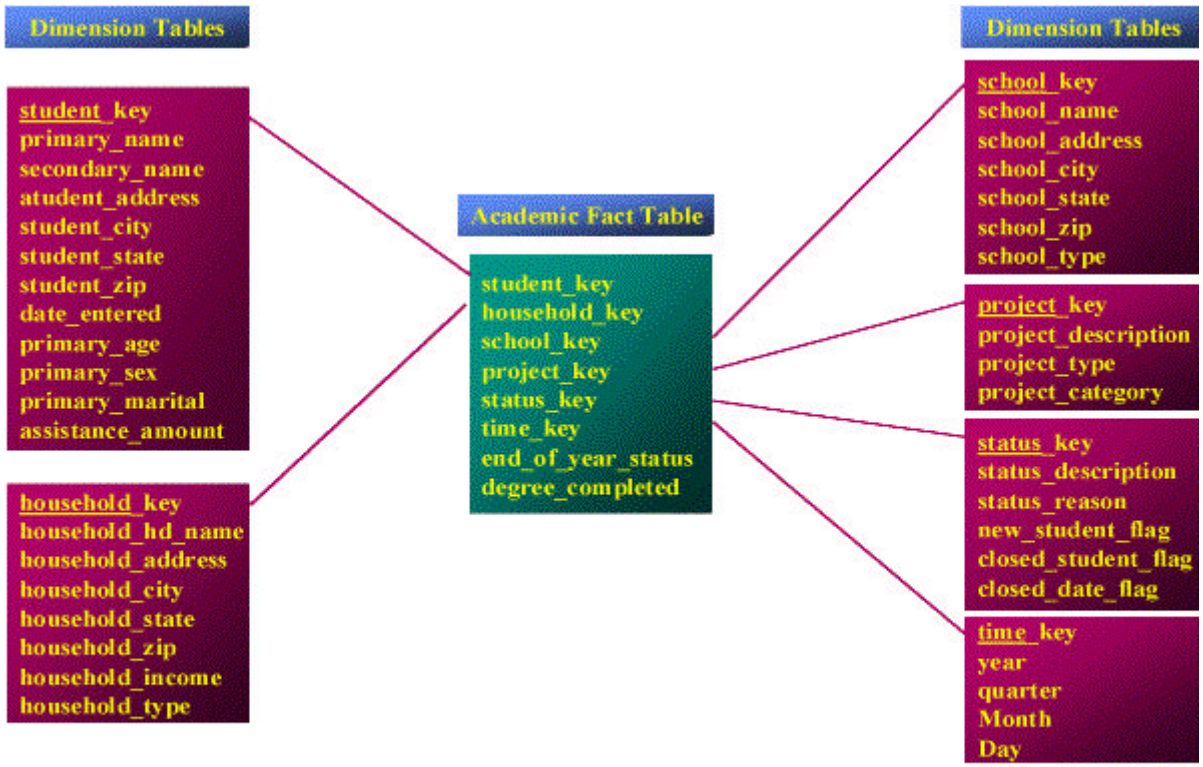

Figure One -- A Dimensional Model Star Schema of A Student Academic Fact Database

While there is consensus in the field of data warehousing on the desirability of using DM/star schemas in developing data marts, there is an on-going controversy over the form of the data model to be used in the data warehouse. The "Inmonites," support a position identified with Bill Inmon, and contend that the data warehouse should be developed using an E-R model. The "Kimballites" believe in Ralph Kimball's view that the data warehouse should always be modeled using a DM/star schema. Indeed Kimball has stated that while DM/star schemas have the advantages of greater understandability and superior performance relative to E-R models, their use involves no loss of information, because any E-R model can be represented as a set of DM/star schema models without loss of information.

In this paper I will comment on two issues related to the controversy. First, the claim that any E-R model can be represented as an equivalent set of DM/star schema models [2], and second, the question of whether an E-R structured data warehouse, absent associative entities, i.e. fact tables, is a viable concept, given recent developments in data warehousing.

Can DM Models Represent E-R Models?

In a narrow technical sense, not every E-R model can be represented as a star schema or

closely related dimensional model. It depends on the relationships in the conceptual model formalized by the logical data model.

As Ralph Kimball has pointed out on numerous occasions, star schemas represent many-to-many relationships. If there are no many-to-many relationships in an underlying conceptual model, there is no opportunity to define a series of dimensional models. That is, the possibility of a dimensional model is associated with the presence of many-many relationships of whatever order. On the other hand, an E-R model can be defined whether or not many-many relationships exist. But without them it would have no fact tables.

Having said the above, it really doesn't directly address the central question of whether an *E-R data warehouse* model can always be represented as a series of dimensional models. But it does shed some light on it. Specifically, the answer to the question depends on whether the underlying conceptual model of a data warehouse must always contain many-to-many relationships. I think the answer to this question is yes, and that it follows that an E-R data warehouse can be expressed as a star schema. Here are my reasons.

(1) Data warehouses must contain "grain" attributes in the sense of the term specified by Ralph Kimball in The Data Warehouse Toolkit. This is a necessary conclusion for anyone who believes either in a queryable data warehouse, or in a data warehouse that will primarily serve as a feeder system for queryable data marts. In either case, the grain attributes must be available as part of the data warehouse, because they provide data on the extent to which any business is meeting its goals or objectives. Without such attributes, business performance can't be evaluated, and a primary DSS-related purpose of the data warehouse architecture can't be fulfilled.

(2) If the grain attributes are present in the data warehouse, what kinds of relationships will be associated with them and what kinds of entities will contain them? In the underlying conceptual model of the data warehouse, there will be attributes that are causally related to the grain attributes, attributes that are effects of the grain attributes, and attributes such as product color, geographic level, and time period that are descriptive of the grain attributes. In the conceptual model, the grain attributes will be associated with many-many relations among these different classes of factors. How can these many-many relations be resolved in a formal model, whether E-R or dimensional?

(3) The various causal, effect, and descriptive factors will be contained in fundamental entities, and perhaps in attributive entities, or sub-type entities as well. In a correct E-R or dimensional model, however, the entities containing the grain attributes can only be associative entities, because the grain attributes will not belong to any one fundamental entity in the model; but will be properties of a many-many relation (an n-ary association) among fundamental entities.

Since fact tables are resolved many-many relations among fundamental entities, it follows that in a correct E-R model, fact tables are a necessary consequence of grain attributes and of

standard E-R modeling rules requiring conceptual correctness and conceptual and syntactic completeness. It goes without saying that fact tables are also the means of resolving many-many relationships in dimensional models.

(4) If fact tables must be present in correct E-R models, it still doesn't follow, however, that the fundamental entities related to them must be de-normalized dimension tables as specified in dimensional models. Here, in my view, is where the major distinction between dimensional and E-R data warehouse models will be found.

In E-R models, normalization through addition of attributive and sub-type entities destroys the clean dimensional structure of star schemas and creates "snowflakes," which, in general, slow browsing performance. But in star schemas, browsing performance is protected by restricting the formal model to associative and fundamental entities, unless certain special conditions (pointed out in "Toolkit," and in Ralph Kimball's various DBMS columns) exist.

So, that's it. In data warehouses, conventional E-R models and Star Schemas are both options, and this is due to the semantics of data warehouses as DSS applications requiring many-to-many relationships containing essential grain attributes. Kimball's position is therefore essentially correct: a data warehouse E-R model can be represented as a series of dimensional models. But this argument has an additional implication I'd like to see widely discussed.

I emphasized earlier that both correct dimensional and E-R models rely on fact tables to resolve the many-many relations encompassing grain attributes that are so essential for the data warehouse. If this is true, then why are fact tables so frequently associated with dimensional data warehouse models and not with correct E-R data warehouse models? I suspect this may be because many E-R data warehouse models may not always explicitly recognize many-many relations and the need to resolve them with associative entities, i.e. fact tables. Instead, these models are being defined with fundamental entities containing some of the characteristics of associative entities but also carrying with them the risks of confusion, contradiction, and redundancy inherent in an incomplete resolution of many-to-many relationships, and ad hoc de-normalization of fundamental entities.

I can't prove that this hunch of mine is valid, and that the problem in E-R data modeling I've inferred is widespread. But there are examples of the problem in the data warehousing literature. One good example is in the recent book by Silverston, Inmon, and Graziano (Wiley, 1997) [3], called "The Data Model Resource Book." Figure 10.2 on P. 266 presents a sample data warehouse data model. This data model contains no fact tables, but three tables come closest:

CUSTOMER_INVOICES,

PURCHASE_INVOICES, and

BUDGET_DETAILS.

Let's focus on CUSTOMER_INVOICES, which is typical of the three. The multi-part primary key is composed of:

INVOICE_ID, and

LINE_ITEM_SEQ.

A number of foreign keys are included as mandatory attributes, but constitute no part of the primary key, and are not determined by it. These are:

CUSTOMER_ID,

SALES_REP_ID, and

PRODUCT_CODE.

Other mandatory attributes are:

INVOICE_DATE, BILL_TO_ADDRESS_ID,

MANAGER_REP_ID, ORGANIZATON_ID,

ORG_ADDRESS_ID, QUANTITY, UNIT_PRICE,

AMOUNT, and

LOAD_DATE.

An optional attribute is PRODUCT_COST.

I believe that this entity diverges as much as it does from a fact table in a dimensional model, not because it is an E-R model-based entity, but because: (a) it fails to adequately model the conceptual distinction between customer invoice and customer sales, (b) doesn't recognize that unit price, amount, and quantity are attributes of a sale, related not only to an invoice but also to Sales Reps, Products, and Customers, and (c) in consequence doesn't correctly resolve the many-many relationship of Sales Reps, Customer Invoices, Products, and Customers. In short, the CUSTOMER_INVOICES entity, as constructed in the example, represents an error in the E-R model. That is why the QUANTITY, UNIT_PRICE, and AMOUNT attributes are not contained in a CUSTOMER_SALES associative entity, a true fact table, with a multi-part key drawn from SALES_REPS, CUSTOMER_INVOICES, PRODUCTS, and CUSTOMERS.

This point is emphasized further by looking at the star schema design for sales analysis provided in Figure 11.1 on P. 271. This design is supposed to provide an example of a departmental specific data warehouse, (or data mart). While this figure includes a CUSTOMER_SALES table that looks a lot like a fact table, it still reflects the conceptual confusion in the underlying model. Specifically, the multi-part key of this "fact table" includes INVOICE_ID, and LINE_ITEM_SEQ, as parts of the primary key. But neither attribute comes from a dimension table, nor are they degenerate dimension attributes since they are part of the primary key.

Instead they originate in the "fact table." And since from the previous CUSTOMER_INVOICES entity we know that INVOICE_ID, and LINE_ITEM_SEQ constitute a unique primary key, it follows that CUSTOMER_SALES is not an associative entity or fact table at all, but instead is another fundamental entity, very similar to CUSTOMER_INVOICES, that again confuses the distinction between CUSTOMER_INVOICES and CUSTOMER_SALES.

In short, Figure 11.1 is not a valid star schema design, as Figure 10.2 is not a valid E-R model. Because neither the CUSTOMER_INVOICES entity in one, nor the CUSTOMER_SALES entity in the other, is an appropriately normalized entity, whose non-key attributes are fully dependent on the primary key. If they were, they would present properly constructed associative entities resolving many-many relations including

CUSTOMER_INVOICES, and CUSTOMER_SALES.

Again, how typical this example is of E-R modeling in data warehousing I can't say. That's the question I'd like to see more widely discussed. Is the widely perceived divergence between E-R and dimensional modeling in data warehousing due to the fact that dimensional modeling necessarily involves fact tables and E-R modeling normally does not, or is the perceived divergence due to the fact that E-R modeling practices in data warehousing are not faithful to E-R modeling principles; and if they were they would involve fact tables to exactly the same extent as dimensional models?

Is An E-R Data Warehouse Model With No Fact Tables A Viable Concept?

DM/Star schemas represent n-ary associations. N-ary associations are embodied in many-to-many relations. These may be resolved within a data model in an entity associating two or more entities. A star schema with one fact table (the associative entity) and two dimension tables represents a binary association. One with one fact table, and three dimension tables represents a ternary association, and so on.

As we have seen E-R models can also represent n-ary associations. They differ from star schemas not in the presence of fact tables, but in the fact that their dimension tables are "snowflaked" to meet the requirements of normalization.

Since star schemas and "snowflaked" E-R models represent n-ary associations, to say that another type of E-R model eliminating fact tables should be used to structure the data in the data warehouse is also to say that n-ary associations should not be used for this purpose. But n-ary associations are essential for analysis in the context of DBMS DSS applications, because analytical DSS queries employ many-to-many relationships and are frequently multi-stage in character. Many-to-many relationships can only be resolved in data models into (1) n-ary associations of various types with associative entities (fact tables), or (2) more atomic data dependency relationships in E-R models without fact tables. I think the second alternative ensures poor query response performance in large databases, and therefore discourages and often prevents execution of a multi-stage analysis process.

It does so because it provides no structure for navigating the logic of the particular n-ary association implied by an analytical DSS query, and therefore requires that the DBMS engine construct the association "on the fly." In contrast, the first alternative provides a navigational structure for such a query, with consequent good query performance, and practical implementing of multistage analysis processes. Among associative models however, a DM/Star design generally provides better navigation and performance than an E-R /Snowflake (in the absence of tools with special capability to handle the more complex snowflake model).

If one accepts this argument (and if it's correct, 95% of it is in some way owed to Ralph Kimball, and if it's wrong, the correct 95% of it is still owed to Ralph Kimball); then the claim that dimensional modeling or "snowflaked" E-R models should not be employed in the data warehouse, largely amounts to the claim that only the limited, constrained analysis supported by data dependency models without associative entities should be employed. That is, the data warehouse becomes no more than a big staging area for data marts, and has no independent analytical function of its own. I can't subscribe to this conclusion.

After all, in recent data warehousing/data mart system architectures, we've added an Operational Data Store (ODS) [4], distinct from the data warehouse, and a non-queryable centralized staging area for storing, extracted, cleansed, and transformed data and for gathering centralized metadata for implementing an Enterprise Data Mart Architecture (EDMA) [5]. Why then do we need yet another non-queryable staging area? Also, if the data warehouse is only a staging area and we can do analysis only in data marts, where do we go for enterprise-wide DSS?

Conclusion

In the context of the "Inmonite"/"Kimballite" dispute over the proper form of data warehouse data models, this paper examined: (1) the claim that any E-R model can be represented as an equivalent set of DM/star schema models; and (2) the question of whether an E-R structured data warehouse, absent associative entities, i.e. fact tables, is a viable concept given recent developments in data warehousing. A number of conclusions are supported by the arguments.

- Not every E-R model can be represented as a set of star schemas containing equivalent information;
- But every properly constructed E-R data warehousing model can be so represented;
- Many E-R data warehouse models are not properly constructed in that they don't explicitly recognize many-many relations and the need to resolve them with associative entities, i.e. fact tables.
- To use data warehousing E-R models specifying atomic data dependency relationships without fact tables is to ensure poor query response performance in large databases, and therefore discourage, and often prevent, execution of a multi-stage analysis process. In effect, it is to make the data warehouse no more than a big staging area for data marts, with no independent analytical function of its own.
- Given the development of ODSs and non-queryable centralized staging areas for storing, extracted, cleansed, and transformed data and for gathering centralized metadata for implementing an Enterprise Data Mart Architecture (EDMA); we don't need another non-queryable staging area called a data warehouse. What we do need, instead, is a dimensionally modeled data warehouse for enterprisewide DSS, prepared to provide the best in query response performance and to support the most advanced OLAP [6] functionality we can devise.

References

[1] Ralph Kimball, The Data Warehouse Toolkit (New York, NY: John Wiley & Sons, Inc., 1996), Pp. 15-16

[2] I thank Ralph Kimball for prodding myself and other participants in the dwlist@datawarehousing.com list server group about the importance of examining this issue.

[3] W. H. Inmon, Claudia Imhoff, and Ryan Sousa, Corporate Information Factory (New York, NY: John Wiley & Sons, Inc., 1998), Pp. 87-100

[4] Len Silverston, W. H. Inmon, and Kent Graziano, The Data Model Resource Book (New York, NY: John Wiley & Sons, Inc., 1997)

[5] Douglas Hackney, Understanding and Implementing Successful Data Marts (Reading, MA: Addison-Wesley, 1997), Pp. 52-54, 183-84, 257, 307-309

[6] "What is OLAP?" The OLAP Report, revised February 19, 1998, @http://www.olapreport.com/fasmi.htm

Biography

Joseph M. Firestone is an independent Information Technology consultant working in the areas of Decision Support (especially Data Marts and Data Mining), Business Process Reengineering and Database Marketing. He formulated and is developing the idea of Market Systems Reengineering (MSR). In addition, he is developing an integrated data mining approach incorporating a fair comparison methodology for evaluating data mining results. Finally, he is formulating the concept of Distributed Knowledge Management Systems (DKMS) as an organizing framework for the next business "killer app." You can e-mail Joe at eisai@home.com.

[Up] [Data Warehouses and Data Marts: New Definitions and New Conceptions]
[Is Data Staging Relational: A Comment]
[DKMA and The Data Warehouse Bus Architecture]
[The Corporate Information Factory or the Corporate Knowledge Factory]
[Architectural Evolution in Data Warehousing]
[Dimensional Modeling and E-R Modeling in the Data Warehouse]
[Dimensional Object Modeling] [Evaluating OLAP Alternatives]
[Data Mining and KDD: A Shifting Mosaic]
[Data Warehouses and Data Marts: A Dynamic View]
[A Systems Approach to Dimensional Modeling in Data Marts]
[Object-Oriented Data Warehousing]