A database geek's answers

(answers from a Microsoft MVP specializing in SQL server integration services)

Question #2: Explain the main advantage of the Kimball method of data warehousing, using a simple example to illustrate.

Answer #1:

The primary advantage of the Kimball method of data warehousing over an Inmon method is to minimize the amount of IO required to answer analytical (read-only) queries. Standard database design – maintaining or applying normalization principles to stored data – results in a "snowflake" database schema as opposed to a "star" schema. A snowflake schema requires more JOIN operations to stitch together a result, resulting in increased requirements for non-sequential disk I/O, and memory to process the query.

For example, a normalized set of tables representing sales could be defined with a fact table containing three columns: dollar amount of sales, date/time of the transaction, and product code sold. A product table would contain three columns: product code, product description and product category. The product category table would contain two columns: product category and product category description. A query to request total sales for a product category requires three tables to be read and JOINed. A Kimball dimensional model would have two tables, the fact table would contain the same three columns, but there would only be one product table containing product code, product description, product category, and product category description.

Although the non-fact tables in Kimball dimensional models are larger and contain repetitive data compared to a normalized schema, they are still relatively small and optimized for read operations. Since normalized databases only store any one piece of information once, they are optimized for quick update operations, but cause locking and I/O issues for large sequential read operations.

Answer #2

The main advantage of a "bottom-up" database, rather than a "top-down" model, is the ease with which which we can integrate data marts to create a comprehensive data warehouse optimized for business intelligence. Dimensions that are shared (in a specific way) between facts in two or more data marts are redundant and less optimal for other uses (such as updates), but minimize the operations required to access the needed data when compared with a big and often complex centralized data array.

As an example, we might construct a sales-data mart that also contains production data. In a "top-down" configuration, these data would reside in separate tables, requiring many individual integration "points" between data marts to conform the dimensions as data is accessed. In a Kimball data warehouse, the integration of the data in the data warehouse is centered on the conformed dimensions (residing in a "bus") where data is grouped along the keys of the (shared) conformed dimensions of each fact. That is, a Kimball data warehouse will join tables, integrating information that in a 'top-down' configuration is stored in separate tables, even if this requires duplication of production information because (for example) two separate products are made in the same factory and share identical production dates. This is called a 'normalized' table, where a join on the keys of these grouped (summarized) facts has already been performed before any reads are made, optimizing read efficiency. The disadvantage is that this normalization results in duplicate data, so an important concern is making sure dimensions among data marts are consistent.