Reprinted from DM REVIEW

Take a Trip and Never Leave the Farm: Virtual Data Marts in the CIF

The mission of the data warehouse in the Corporate Information Factory is clear. It serves as the repository of integrated, cleansed, historical snapshots of detailed data for use in strategic decision-making applications. To fulfill its mission, the design of the warehouse should meet these requirements:

- Non-redundancy of data: Because data redundancy slows the load process, the data warehouse design should minimize the occurrences of any given attribute.

- Stability: The data warehouse design is as process-free as we can make it. The last thing we want to do is to design a data warehouse based on business processes. Why? Because they can change with frightening regularity, requiring that the warehouse be unloaded, redesigned and reloaded for each change.

- Business rules: Data warehouse design must uphold universally agreed-upon business rules of the enterprise. Parochial or department-specific interpretations are not stored here.

- Lowest level of detail: The data warehouse must support the "least common denominator" of data that is ultimately used by the business community. If one application requires monthly snapshots and another uses weekly summaries, the warehouse must be designed to store daily data to accommodate both, even though neither uses daily data.

Similar to the Saturday Night Live cone-heads, data warehouses "consume mass quantities" of data. In response, data warehouse designers need to optimize/shorten the process of getting data in. They accomplish this via the data model.

Many data warehouse designers use the normalization techniques developed by Chris Date and Ted Codd in the '80s for their data models. Third normal form is the level of normalization most common for warehouses. That is, "every attribute is dependent upon the key, the whole key and nothing but the key" (so help me Codd). This level of normalization goes a long way in satisfying the four design requirements; however, until recently, it wasn't practical for supporting data analysis.

In particular, multifaceted queries generated from the business community frequently translate to multidimensional analysis, which requires the combination or joining of many entities. In a third normal form database, that kind of navigation can be complicated for the business community to understand and slow from a performance point of view.

Enter the data mart – the data warehouse's "mini-me." Smaller, focused, modeled according to the query types submitted, and usually physically separate, data marts are great at giving the business community access to data in exactly the desired format. Data mart designers are less concerned with getting data into the environment. Instead, they focus on getting the right information out into the hands of the business community, at the right time, in the right format, and with the appropriate "spin" – that is, delivering the "view" of the information that meets each user's needs.

So, it's fair to say that the differences in functional requirements between data warehouses and data marts have historically translated into differences in technological requirements, which usually lead to the physical separation of the two.

Virtual Data Marts

What if the differences in functional requirements could be adequately supported via a single technological platform? What if we could do more than just data mart prototyping inside the data warehouse? What if we could implement fully functioning, virtual data marts inside the data warehouse?

Similar to data mart prototyping in the data warehouse, virtual data marts are logical views of data warehouse data. That is, the data mart is not physically manifested anywhere. Aggregation and summarization logic is contained within the views, and business users access their data mart directly through these views.

Believe it or not, virtual data marts have been around for years. The truth is that any relational database management system (DBMS) that will support data warehousing will also support virtual marts to some degree. However, the prosperity of standalone data mart applications is proof that something was missing.

Until recently, what was missing was a combination of database engines that were designed with virtual marts in mind and hardware horsepower. It's worth noting that Teradata's platform has been capable of sustaining virtual data marts for years. Although they probably have the most experience at it, others such as Microsoft, IBM and Oracle are now in the game. As far as hardware horsepower goes, Moore's law didn't let us down.

In deciding whether a virtual data mart will work for you, keep the following in mind. Your data warehouse environment must support:

- **Views** supporting the variety of data modeling usually associated with data marts. This means the data warehouse must support more

than just star schemas; it must support flat files, floating point files, statistical subsets, etc.

- **Queries** usually associated with data marts. Trends, pattern analyses, drill up/down/around/through, segregation and scoring and market-basket analyses are just a few of the types of queries that can be routinely used.

- **Access software** usually associated with data marts including OLAP, statistical analysis, data mining or sophisticated analytics, exploration and reporting technologies.

In addition, here are some pros and cons of virtual data mart implementation.

As far as pros go, virtual data marts:

- Reduce disk storage.

- Reduce network traffic as there is no need to transfer data from the data warehouse to the data mart(s).

- Reduce backup and recovery tasks.

- Are already commonly used for prototyping; therefore, moving from prototype to production requires fewer resources.

- Provide immediate visibility to implementers when a data mart attribute is not available in the data warehouse.

- Offer one-stop shopping. All of the strategic data is in one location and everyone knows where it is. When additional attributes are added, it's easier to change a view than it is to modify multiple steps in the process of getting the new attributes from the data warehouse to the data mart(s).

On the con side, virtual data marts:

- Require modern, specialized, high-performance DBMSs running on high-performance hardware in order to avoid potential performance issues. Obviously, turning a great number of users loose on the virtual data marts with myriad types of queries and analyses could bring even the highest capacity environment to its knees.

- Centralize network traffic. Virtual marts can not be distributed; their creation requires all users to have access to the platform in which the virtual marts are located.

- May be difficult to create with views if they are extremely complex and thus can't be represented by SQL statements. Sophisticated market analytics, for example, may not be ideal candidates for the virtual world for these reasons.

- Require that data warehouse DBAs be able to translate complex data models (star schemas, snowflakes, etc.) into SQL statements.

- Go against traditional wisdom that was previously validated by repeated experience with hardware and software limitations. Physical data marts are "the devil we know." Anything that introduces change causes a certain level of discomfort. This may be the biggest hurdle.

More than a decade ago, early relational database technologies such as IBM's DB2 struggled for performance reasons. Although the ideas were good at the time, the hardware and software just hadn't evolved enough to sufficiently support them. That situation didn't last long. Likewise, virtual data marts are a good idea that has struggled for performance reasons – and that situation won't last long either.

Claudia Imhoff is the president of Intelligent Solutions, Inc. Imhoff is a popular speaker and internationally recognized expert on the Corporate Information Factory, business intelligence and CRM. She has coauthored four books and more than 40 articles on these topics. She may be reached at cimhoff@intelsols.com.