When and Why to Put What Data Where

The usage of ODS and data mart platforms in a well architected data warehouse

By: Rob Armstrong

Teradata Corporation

TERADATA® Raising Intelligence

When and Why to Put What Data Where

Table of Contents

Author's note: This paper is a companion of the "Optimizing the Data Warehouse Layer" white paper. While this paper describes the differences and functions of the ODS and data mart layers, the other paper discusses how to best optimize your core data warehouse repository layer to minimize data movement, minimize data management costs, and increase reusability and end-user accessibility. The companion paper is available upon request.

Executive Summary

Companies are trying to overcome two seemingly conflicting challenges. The first is how to integrate more historical and cross-functional analytic processes into their front-line operations to take the most relevant actions against the most profitable customers. The second challenge is how to integrate more timely interactions and transactions into the back-end analytics so that strategy can be measured and fine tuned at an ever increasing pace. In order to balance these two objectives, companies are looking to push data to the process by incorporating operational data stores and data marts into their data management architectures. Unfortunately, this constant pushing of data results in more data inconsistency, more data latency, and a more fractured understanding of the business as a whole. Understanding the correct usage of the layers helps to alleviate some of the problems. However, the real solution is to integrate the data once and drive the process to the data. Having said that, and understanding the larger challenge of centralization will keep some on the distributed path, there are some guidelines to understanding how and when to use the architecture layers in an efficient, as well as an effective manner.

TERADATA®
Raising Intelligence

When and Why to Put What Data Where

Operational Data Stores

The operational data store is typically used as a staging area for transactional data. This is a good use of this layer. Unfortunately, the ODS often morphs into the reporting and access point for what is deemed operational data. As the front-end processes require more historical or cross-functional data, these structures become the problem rather than the solution.

ODS contains data from transactions that are 'in flight' – Many transactional systems generate data that are very raw in nature. The data may be used throughout the data within the operational systems before the data are considered 'set'. An example would be an order process where the initial order is a very high-level capture of the items and then, once the order is completed, there is validity checking, inventory checking, and data cleansing or matching before the order is considered complete and moves on to fulfillment.

ODS contains data not necessary for cross-functional processes – Once the data are considered set, the next question is, "Who needs to see these data?" If this is simple workflow process, then containing the data in the ODS and sending messages to other organizations is much easier than physically moving the data to yet another staging area. To continue our example above, the order is taken and validated. A message is sent to fulfillment where the processes access the ODS, package the order, and then send the package for distribution. The distribution processes access the ODS for mailing information and so forth. Once the transaction is completely processed the data would be useful in analytics across many of the corporate business units, especially when combined with other corporate data. This is when the transaction needs to be moved from the ODS into a more cross-functional, fully-integrated data warehouse structure.

ODS contains data not available elsewhere – This is the key to keeping data consistent and auditable. Once the data are moved from the ODS, they need to be deleted from the ODS. Any subsequent process needs to access the data from the enterprise data warehouse. The data are now considered cleansed and integrated. The example here is the infamous operational report, which informs management about order processing commitments and metrics. These are after the fact reports and can be accessed easily from the warehouse. While it is called an operational report, it does not need to be generated by an operational data store. This is confusing the process with the platform.

Data Warehouse Core

As data are needed for historical analytics, cross-functional processes, or simple storage for easy access and ongoing management, they belong in what some term the "hub" or "core" data warehouse. Others have called this the enterprise data warehouse or the centralized data warehouse. Whatever the name, it means the same. This is where data are brought together to provide the integration and single version of the truth to the data which drive business analytics and processes.

The core is the integration point for data – While many believe the goal of the data warehouse is consolidation of data, the real effort is to integrate the data. Rather than simply bringing the data into a common platform, the data must be modeled and integrated to preserve data relationships and provide an application agnostic body of data. By having the data integrated and not compromised to any particular application, the environment becomes prepared to handle not only the known, but the unknown, analytics and processes. This will enable the business and operational processes to identify weak points sooner and respond quicker to new requirements.

The core contains the historical relevance and relationships – One of the great challenges, and a driver for pushing data out to data marts, is the need to preserve the *what was, what is* analytics as reference data changes. Questions, such as "How did Region one do last year when compared to this year?", need to have consistency if branch locations have been reassigned. The optimal way to accomplish this is through effective dates being incorporated into the reference tables. Since the historical detailed data reside in the core, and the reference data can be managed in the core reference tables, analytics are allowed to run any point of time reporting. The other benefit to having historical relevance in the core data model is the enabling of Master Data Management across the organization.

The core is where process can come to data – This is the real point of the core data warehouse layer. As the data are brought into the repository at ever increasing frequencies and integrated with ever more subject areas, they become the de facto memory of the corporation. After spending all the time and money to get the data integrated and accessible, why would you want to add yet another extract, transform, and transmit process? Once the data are in the core, companies should use the capabilities and optimizations in the database engine to minimize further movement. All of this would create an environment of minimal data latency and redundancy, maximized data reuse and auditing, and decreased time to develop new analytics or operational processes. Of course, all of this will depend on the database engine being able to manage a volatile, mixed workload against an application-neutral data model. If the engine is not capable of this, then the question you have to ask is whether it is your tools or your requirements that are holding you back?

Data Marts

The data mart is the analytical equivalent of the ODS. These are subset extracts of data for well-defined analytics and perceived higher performance. There are many political reasons for the proliferation of data marts though, in practice, they increase data latency, increase time to deliver on new business needs, and create more confusion due to the multiple versions of the truth coming from the multitude of data marts.

Data Marts are for data considered static in nature – One of the main problems with data marts is the constant updating as the underlying detailed data are maintained. Data marts are best utilized when the data extracted and transformed from the base detail will not change, or change minimally, in the future. Examples of this would be end of month or quarter reporting, householding metrics to combine multiple customers to a single identifier, or corporate reporting for government regulations. Data that are constantly changing would significantly increase the cost of the data marts, as well as decrease the data availability as the data mart is either out of synchronization with the detail or unavailable due to the updating process.

Data Marts are for repetitive, and predictable reporting – The other major use for data marts is when there is a known set of reporting needs across a wide body of users. If the same report will be requested hundreds of times, then it makes sense to generate that report once from the detail and post the result to a data mart. A word of caution on this usage is necessary. There is a difference between users asking for the same report and asking for a similar report. If each buyer is asking for only their products, then it is not the same report. Many companies often build data marts much larger than the detailed data because they must aggregate for every possible scenario. In reality, letting each user ask for their small segment against the detailed data will result in lower CPU and disk utilization as opposed to generating every possible answer set that may or may not be accessed.

Data mart cost must be justified by the ensuing business value – The real question for the data marts is whether or not they are worth the cost. Unfortunately many companies never take this into consideration. Data marts exist to provide the higher performance so, the question is, what actions are being taken because of that better response time? Are the users taking different actions, or taking actions sooner, because they have data more accessible and available? One example would be a process to advise about the next best offer. If the query takes five minutes when going against detailed data, then the offer will not be timely to a customer in the store, at a kiosk, or when calling into the company. However, if the data are preplanned and in a data mart to provide two second response, then that application could be used in a call center and be much more effective in getting customers to accept the offer. The business should be able to quantify the benefit to offset the additional IT cost of the creation and ongoing management of the data mart.

Conclusion

Proper usage of the various layers is critical in the long-term success of complete data warehouse architecture. It is important to understand the goal is to minimize the data movement and increase user accessibility. This will drive lower cost to manage and deliver new capability while increasing the timeliness of data and the relevancy of actions. Here are a few recommendations to help in your total evolution and usage of the data warehouse architecture layers.

Ask Why Not?

Too often, the decision about where to put the data and how to organize them is based on a very narrow view of application and understanding. Many operational application developers have been trained to think the data warehouse repository is not for quick access or recent data. Rather than make arguments as to why the data need to be moved from the core, it is best to ask why the data cannot reside in the core. Is it a latency issue, performance issue, not all data are resident there, or some other explanation? Log the reasons and then, as they are resolved, work to move the process back to the data rather than continuing to push the data to the process.

Solve the Real Problems

While it is often easier to just move the data to resolve an outstanding issue, it is not always the best solution. Performance and latency issues can also be rooted in poor data quality and questionable data modeling choices. When application development has to either reconcile data quality or code around inconsistent data models, performance will suffer. Rather than take the easy fix of moving the data, which will still need to be reconciled and made consistent, go back and fix the underlying problems. Fixing it once, correctly, will increase the time to market for all future processes, fixing it in the application will mean that every new process will need to replicate efforts and most likely lead to even further data inconsistency and delay.

Provide Enterprise-wide, End-to-end Solutions

Many companies have a goal of an enterprise data warehouse with timely, integrated data being accessible directly by the end-user community. However, many of those same companies will isolate crucial technical processes such as ETL, query tools, and performance management, as well as the corporate processes of data quality and funding. In order to drive an enterprise solution, there must be an enterprise mindset. Total end-to-end performance means that when adding indexes, you also incorporate the time it takes to update and manage the index into your costs and latency consideration. Creating new ODS tables means that you not only have to add maintenance time, but you may now have duplicate data that need auditing and reconciliation. A last example is funding. There needs to be a mindset shift away from project funding to program funding. After all, which project funds data quality and integration? These core infrastructure processes and integration points need to be incorporated into new projects to ensure the enterprise – not just the application – succeeds.

About the Author

Rob Armstrong is director of Teradata's Data Warehousing group and has more than 20 years' experience justifying, designing, and implementing data warehouse systems. He received a BA in economics with an emphasis on statistics and relational theory from the University of California, San Diego. Contact him at rob.armstrong@teradata.com.

TERADATA® Raising Intelligence