The Teradata Enterprise Analytic Data Set

By: Bill Franks,
Partner, Teradata
Advanced Business
Analytics

The Teradata Enterprise Analytic Data Set

Table of Contents

Introduction

It's a well established fact that a large part of any analytical or statistical modeling process is the work surrounding the gathering, cleansing, and manipulation of the data required as input to the final model or analysis. It's often stated that as much as 60% – 80% of the man-effort during a project goes toward these steps, with at least that same percentage of total processing cycles going toward the same effort.

When models are run infrequently or there are only a few models to run, it makes sense to do this work as a part of each specific project. However, once an organization begins incorporating dozens or even hundreds of models and analyses into their business environment on an ongoing basis, the repeated manipulation of large amounts of data becomes inefficient. Teradata Corporation provides best practices that can greatly condense overall processing cycles and vastly reduce the time to create, update, or implement any given model or analysis.

What Is an Enterprise Analytic Data Set?

As models are built over time, certain standard metrics and manipulations become readily apparent. As an example, it's hard to imagine that total customer spending or number of customer transactions would not be of interest in most analysis efforts for a retailer. Similarly, it is hard to imagine that total product sales for recent periods would not be of interest to most product-level analytics. At the same time, any required cleansing or recoding of the detailed data required to facilitate such rollups will become constant once the right analytic procedures are established.

An Enterprise Analytic Data Set (ADS) takes the standard data rollups that are used in a variety of analytic tasks and centralizes their generation. (See Figure 1.) Instead of each analyst or process having to incorporate all of the logic and consume all of the processing time needed to derive data for each analysis, standard metrics are instead created in an automated fashion on a regular schedule and made available to all analysts and processes. In addition to the tables and views that are available from the final Enterprise ADS, there will also be a catalog of supporting queries and processes used to generate them. Any entity that will be the focus of a wide range of analytics is a candidate for an Enterprise ADS. Examples include Customer, Location, Product, Employee, and Vendor.

The advantages of this methodology are clear:

> Consistency is assured in the methodology that various analysts and processes use to generate their analytical data sets. In addition, the chance of an error arising due to the omission or altering of appropriate logic is removed.

> Placing models into production is easy, since the same data structures used for building the models contain all the data needed to deploy them.

> Overall system processing cycles are greatly reduced, since variables requiring repeated processing will be computed just once and then stored and shared, rather than being run time and again.

> Analysts can proceed straight to adding value with their work, rather than focusing repeatedly on the same basic preparation work.

> Once the information is available, new uses for the data will be found that were not previously practical. For example, results can be incorporated into standard reports.

> The Enterprise ADS provides a simplified view of a complex data warehouse environment by providing a condensed, manageable number of analytic tables that represents key information from possibly hundreds of tables of detailed data.

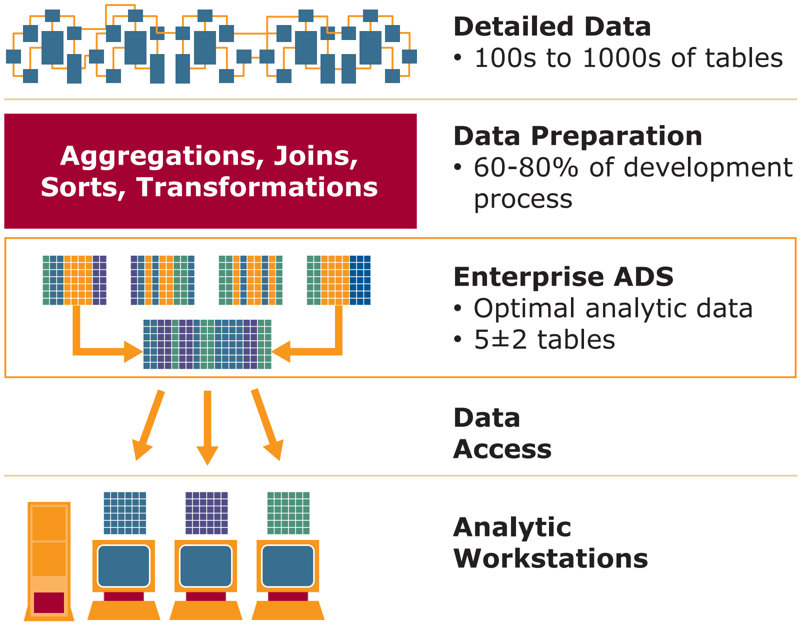

Figure 1. Building and using enterprise analytic data sets.

Making Enterprise Analytic Data Sets Work

Cost and Benefits Overview

There are costs associated with establishing an Enterprise ADS. First, an effort must be undertaken to define and implement the Enterprise ADS. Second, there will be a scheduled process to recreate the data that may involve more processing than any single run. Third, disk space will be required to host results. Finally, data in an Enterprise ADS will often be a snapshot of conditions at a certain point in time, so care will need to be taken to ensure that end users fully understand precisely what is represented in the Enterprise ADS.

However, these costs become minor once a fair amount of analytics are being utilized for the following reasons:

> Analysts are already spending a great deal of time creating data sets. Time spent generating an Enterprise ADS will actually save time in the long run.

> The Enterprise ADS may have more fields than any single process requires, but the benefits of running a single larger process over many smaller processes are realized quickly. Analysts can also experiment with additional data elements that may not have been worth computing for just a single effort.

> Disk space is relatively inexpensive. When the benefits of the analytics are taken into account, the additional storage space should be easily justified.

> Having common metrics available and ready to go within the database will enable any number of other applications or processes to leverage the information and extract value. Many of these other uses would not warrant a special process just by themselves.

> Last, many aggregations within an ADS use data spanning a long period of time. So, not having up-to-the-minute information will have minimal, if any, impact on results. However, if necessary, it's also possible to have the Enterprise ADS generated on demand so it's fully current, although this has additional performance issues to consider.

Figure 2 represents the average time spent in each step of an analysis based on Teradata advanced analytic consultants' implementations. These implementations typically leverage in-database analytics to

Process	Previous Percentage	Traditional in Days	Best Practices	
			Initial	Subsequent
Problem Understanding	5 – 10%	3 days	3 days	3 days
Data Understanding	10 – 15%	5 days	7 days	1 day
Data Preparation	30 – 60%	15 days	30 days	4 days
Modeling	20 – 30%	6 days	6 days	6 days
Evaluation of Results	20 – 30%	6 days	6 days	6 days
Deployment	5 – 10%	3 days	3 days	3 days
Total Time		38 days	55 days	23 days

Figure 2. Average time spent in steps of analysis

analyze large volumes of complex data. These data identify total savings in the development process. While the specific length of time for each step will vary by situation, the overall percentages should remain fairly stable.

These savings were computed in a conservative manner, and still they show nearly a 40% improvement in total time (23 days versus 38). If just the portions of the process that involve the preparation and manipulation of data are included (steps 2 and 3), the savings increases to 75% (5 days versus 20), almost doubling. While these savings are broad averages that can vary from situation to situation, it's not hard to see that there are tremendous gains awaiting the Enterprise ADS effort.

When Do Enterprise Analytic Data Sets Make Sense?

Enterprise Analytic Data Sets make sense for an organization with a strong history of analytics and a solid understanding of their underlying detailed data. When a given data source is first analyzed, it is necessary to identify and account for data anomalies – a process that can take some time. The process of aggregating data may not be stable as new issues are found or better ways to process the data are identified. As time passes and more projects are completed, however, the data become better understood, certain metrics begin to appear repeatedly, and many of the same computations begin to surface again and again.

Once an organization's rules for processing data have stabilized, and the same core set of metrics becomes common, it's time to consider an Enterprise ADS. Clearly, it does not make sense to generate an Enterprise ADS if minimal analysis is expected on an ongoing basis. However, it doesn't take long for the amount of analysis to quickly rise to the point where it is valuable. For example, an organization implementing new CRM initiatives will certainly expect to execute many sophisticated analyses on their customers – and this is a case where an Enterprise ADS makes great sense.

Note that even if an organization feels strongly that only up-to-the-second data should ever be used and that precomputing key metrics is not an option, the concepts of an Enterprise ADS still apply. In these cases, the Enterprise ADS may be a series of views, macros, or stored procedures that are set up to physically create the standardized Enterprise ADS at run time. Most advantages of an Enterprise ADS, such as standardized methodologies and reduced man hours spent on data preparation, will still be realized. The only advantage lost is the compute once, use many aspect.

Defining and Implementing an Enterprise Analytic Data Set

The specific contents, logic, and physical storage aspects of an Enterprise ADS will require a team effort. The business community will play a large role in defining the objectives and focus for the analysts to pursue. Without active business involvement and support, there may be an extra period of trial and error to finalize requirements. Analysts will play a large role in creating and defining the metrics and logic. Database administrators will play a key role in automation, scheduling, and resource allocation for the ongoing processing. Given that these teams are normally already working together to execute analysis without an Enterprise ADS, it's not a big stretch for them to pull together the requirements and design for an Enterprise ADS.

Physical versus Logical Enterprise Analytic Data Set Content

As with database design in general, there will be a logical view of the Enterprise ADS, as well as a physical implementation of that view. Logically, end users can think of an Enterprise ADS as containing one row for each entity and a range of columns containing various metrics for that entity. For example, there is one row per product, which looks like a traditional flat file. Each row contains a variety of columns with metrics related to that product. Another example might be a table with one row per individual customer that contains a wide range of sales, margin, and product purchasing information for each customer.

Physically, the data may or may not be stored in exactly that format. Data will instead be stored in a normalized fashion with one row per entity/time combination, or some other multiple layer structure. Or, there may be several data tables containing only certain columns that are related to each other in some fashion and computed together. For example, one table may contain customer demographics, while another may contain customer behavioral data. Those two tables may be updated on a very different schedule. A view is used to join these two different types of customer information together so that the physical storage difference is transparent to the end user.

The final decision about how to best store the data is best determined by a skilled data modeler and DBA. There are plenty of tricks that can be used to minimize storage requirements. One option is to store information in an atypical format. Another option is to use an *All Other* classification in addition to the Top N individual classifications to minimize the number of metrics produced and remain focused on the most important. An example of this would be having total spend for each of the top 50 product categories in a store for each customer, plus having the spending for all other categories grouped together. If they are small and uncommon, it may not be worth tracking numbers 51+ except when specifically needed.

Regardless of how data are stored, end users should be able to quickly approximate the logical flat file style view of the data outlined above from the physical tables via a series of views or other techniques.

Keeping Processing on Teradata Database

Perhaps the single factor most affecting the ability of an organization to maintain an Enterprise ADS is Teradata® Database's ability to process massive amounts of data in a scalable and timely fashion. Many systems simply aren't capable of handling the degree of data manipulation, complex joins, and full table scan processing that are required for the generation and updating of an Enterprise ADS. For this reason, many organizations pull data off of their systems on a regular basis and process it with an external tool, such as SAS. This common approach introduces a wide range of challenges including, but not limited to:

> Developing procedures for pulling data off of the host system.

> Having the required network bandwidth to execute the transfer.

> Creating a duplicate copy of core data that can quickly become obsolete, thus developing its own data quality and integrity issues as it is used and manipulated.

> Frequently relying on samples – as opposed to using all the data – due to the inability of the architecture to scale.

> Needing to replicate any results found on the extracts in the original environment eventually if the organization is to get full benefit. Many projects have died because something outstanding was found on a sample, but there was no feasible way to apply the findings back to the entire database.

When leveraging the Teradata Database for your analytics, these challenges are eliminated. Using Teradata Warehouse Miner, or even by hand coding Teradata SQL, it's possible to explore your data and generate all the logic required for your Enterprise ADS. Teradata Warehouse Miner eliminates the need to extract the final Enterprise ADS from the system for the majority of common analysis and modeling techniques. Teradata Warehouse Miner supports a wide range of analytical models, including decision trees, regression, factor analysis, and clustering. These common techniques can be executed directly against the Enterprise ADS in the Teradata system with no need to ever move any data anywhere.

In those cases where it's necessary to use an external tool that has a specific required algorithm, however, that tool can now download only the data it requires from a ready-to-go source. Instead of downloading detailed data and aggregating them, the offline tool can simply access the Enterprise ADS. This will minimize the impact of data transfer and also ease the translation of any analysis into the production environment, since only the final model was run outside of Teradata Database. The scoring routine for the model is more easily translated into the production environment since the offline tool took data directly from the same Enterprise ADS that the production scoring routine needs to use.

With the newly emerging PMML, Teradata Warehouse Miner can even accept model results directly from other major modeling tools. These tools can now access an ADS on a Teradata system and directly generate an SQL scoring routine with just a few points and clicks. Thus, no hand coding is required to get models placed into production on Teradata Database, even though they were developed elsewhere.

> **Predictive Model Markup Language (PMML) is an XML-based language that enables the definition and sharing of predictive models between applications. A predictive model is a statistical model that is designed to predict the likelihood of target occurrences given established variables or factors. Increasingly, predictive models are being used to forecast business-related phenomena, such as customer behavior. The PMML specifications establish a vendor-independent means of defining these models, so that problems with proprietary applications and compatibility issues can be circumvented.**

General Guidelines

Obviously, it's not possible to compute every possible variable in your analytic data set. The goal of an Enterprise ADS is to standardize the most common and widely used variables. In many cases, the standard Enterprise ADS will be all that is required for a new analysis to be completed from start to finish. In some cases, it will be necessary to enhance the Enterprise ADS with additional variables. The key is that if certain additional variables become quite common, they should be added to the Enterprise ADS.

Note that some of the same variable types can be computed across a number of dimensions to make the data more robust. For example, a current metric or model score might be stored for a customer along with scores or metrics from three, six, nine, and 12 months ago so that trends can be identified. Or, metrics might be computed individually for certain products or categories, certain day parts, or individual channels. Another example would be storing results for each customer for each quarter, and providing a view that combines the quarterly data into an annual view for the end user. Teradata Database's ability to handle detailed data enables users to add variables and dimensions as needed without restructuring the underlying database – a step often required by other platforms. The sky is the limit.

However, a sense of practicality needs to come into play. Obscure metrics or metrics computed across unimportant or uncommon dimensions should not be in the Enterprise ADS. Instead, they can be computed as needed, unless the system has ample capacity to compute and store this information – in which case, you may as well compute everything that you can think of.

The ultimate decision about what to include in your particular Enterprise ADS will come down some very basic trade-offs. Each additional variable in the Enterprise ADS equates to more processing time, more complex scripts, and more storage space. Note that when generating an Enterprise ADS in real time through views, space is still a consideration alongside processing time. Practicality should determine which variables make the final cut. It is also possible to have different parts of the Enterprise ADS updated on different schedules, as dictated by their usage. For example, a customer cluster classification may only be updated monthly or even quarterly, but basic customer RFM metrics might require weekly refreshment.

Another concept to keep in mind is that many metrics can be computed directly from other metrics, and so do not need to be explicitly stored. For example, given total spend and total number of transactions, you can compute average spend per transaction. Additional processing that can be done directly against the entity-level Enterprise ADS does not need to be physically stored. Rather, views can be utilized to give access to the information. In this case, a view would compute spend per transaction from the two base metrics.

In the end, the unique circumstances surrounding each individual organization will determine what analysis is planned, what system resources are available, and how to define the refresh methodology best.

Conclusions

With the way that relational databases have evolved, it just doesn't make sense to extract data and execute processing outside of the data warehouse environment. This is even truer for Teradata Database than for competing data warehousing platforms. Instead of having many individuals running their own separate project-specific processes to compute the same metrics again and again, organizations should standardize the generation of analytic data sets through the concept of an Enterprise Analytic Data Set. While this isn't a magic bullet that will solve every problem, it can be a tremendously valuable addition to the warehouse environment.

Few analysts will complain about having to spend less time preparing data and more time doing value-added analysis. Few BI specialists will complain about having another source of robust metrics available for inclusion in their reports.

Few data warehousing teams will complain about having their platform become more of a standard data source than it is today and minimizing the number of tools and applications to be supported. And, of course, few executives will complain about a major investment, such as a Teradata solution, being leveraged to the extent possible to generate additional return on investment.

If your organization hasn't yet developed an Enterprise ADS architecture, you should consider doing so. Market leaders in a wide range of industries have begun implementing these architectures. And as they begin executing more analysis as a result, they will further distance themselves from their competition. Why not become one of the organizations leading the pack instead of playing catch-up?

About the Author

Bill Franks oversees the Teradata Advanced Business Analytics Center of Expertise, assisting Teradata retail customers with applying analytical and data mining techniques to their businesses. He also works with Teradata's CRM, Demand Chain Management, and Business Intelligence areas, determining the value data mining can add to these existing applications.

Prior to joining Teradata, Franks was senior vice president and chief analytics officer for SmartDM, where he spearheaded the implementation of a full suite of Internet-based marketing and campaign management tools, including a sophisticated online reporting system.

Franks earned a Bachelor's of Science degree in Applied Statistics from Virginia Tech and a Master's degree in Applied Statistics from North Carolina State University.

Enterprise Analytic Data Sets in Action

- > One major cellular company has created a 450-variable customer ADS in Teradata Database. By leveraging the standard data source, development of new models was cut from weeks to days (See Figures 3 and 4).
- > One leading financial services company currently utilizes a Teradata ADS with 1,400 variables. This enabled them to complete an emergency analysis on exposure to Hurricane Katrina within hours, while their competition took weeks.
- > One well-known retailer has a Teradata ADS with 1,200 variables. The ADS was implemented as one component of an initiative that shortened model development from many weeks to days in most cases.
- > One major internet player maintains a large Enterprise ADS in their Teradata solution that allows them to access the data needed for new models within hours. They are now able to develop response models for new campaigns within days of execution.

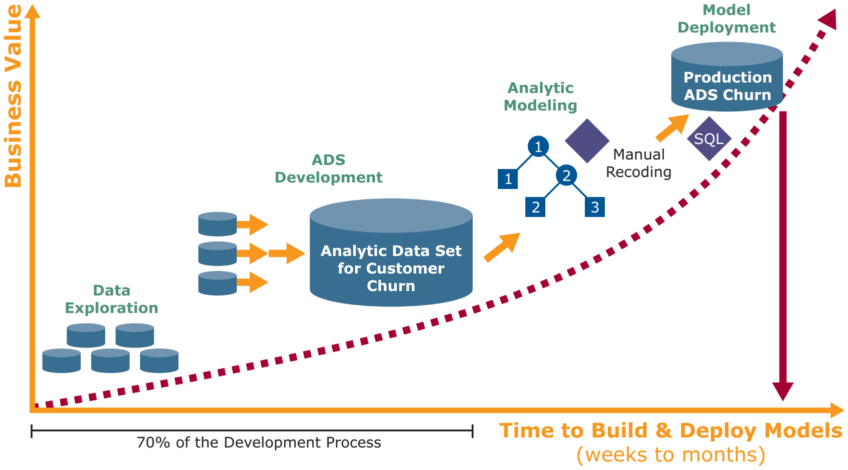

Figure 3. Time to build and deploy initial models can take from weeks to months.

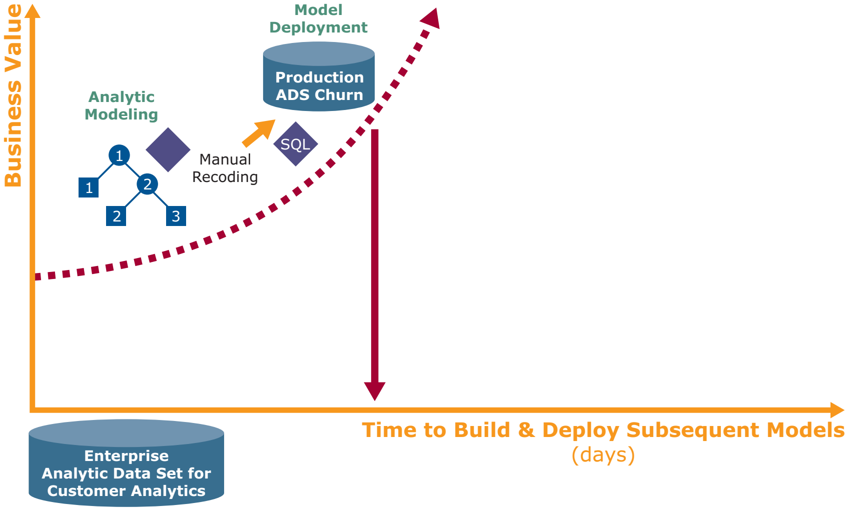

Figure 4. Deploying additional models from the enterprise analytic data set takes much less time.

Appendix: Enterprise Analytic Data Set Example

For practical reasons, details will be further outlined only for what may be the most common Enterprise ADS candidate, as well as having applications to virtually any industry. The area of focus will be Customer. As mentioned previously, the same concepts outlined below can apply to any number of important business entities, including employee, product, and location. Practicality and space dictate what can be addressed directly within the scope of this document. It is assumed that the reader can extrapolate the concepts to other subject areas from the example below.

Customer

Analysis of customers can include a large number of variables. Additionally, there are often millions of customers to track. So, a customer Enterprise ADS is potentially one of the largest and most resource-intensive members of the Enterprise ADS family. There is a wide range of data points that make sense for each customer. First, some metrics will be discussed, and then some dimensions across which those metrics might be computed will be given.

For this example, we'll focus on a customer Enterprise ADS for the Retail industry. Many of the metrics below will overlap with other industries and many will not. Even different retailers will have varying requirements. The key isn't to focus on the specific metrics, but to think about how these types of metrics might apply in your business and what other metrics you would find important.

Described below are ten types of data elements that could be included in the Enterprise ADS. The list is not exhaustive by any means, but touches on a number of key areas:

> Basic RFM metrics – Total spend, number of transactions, and time since last transaction

> Average transaction size, revenue, and profitability (can be computed from above)

> Average distinct products, categories, or departments per transaction

> Total distinct products, categories, or departments purchased over time

> Total purchasing spent towards specific products, categories, or departments

> Discount, markdown, coupon, and other similar information

> Lifestyle metrics that can be identified from the transaction data, such as Low-Carb purchasing, allergy sensitive, or baby present

> Demographic, lifestyle, mail responsiveness, or other data purchased from third parties or acquired directly from customers

> Survey data from customer survey efforts

> Scores from any number of statistical models or deep dive analyses

A number of dimensions for which the above metrics can be computed are described below:

> Time: Metrics might be stored for the current period, plus several past periods.

> Store or Location: Monitor customer behavior by store, store format, or region.

> Channel: See how customer behavior varies based on what channel is being used.

> Product: Compare patterns across products, categories, or departments.

The Customer ADS might have several physical tables at different levels of aggregation. For example, demographics will be stored just at the customer level. However, spend related metrics might be at the customer/store/time period level. A series of views can provide access to different combinations of data. In some cases, it might be desirable to combine demographics with quarterly spend information. In other cases, a year of information might be desired with the demographics. With the appropriate combination of tables and views, the end user can be given whatever is required.