Teradata Value Analyzer

01.15 EB6120 PROFITABILITY ANALYTICS

TERADATA.

Table of Contents

Executive Overview

Teradata Value Analyzer (TVA) is Teradata's tool for calculating the profit contribution for all of a company's accounts, customers, relationships or other entities. This gives the management team the valuable information needed to understand and affect the profit dynamics of the business. TVA is not industry-specific; for example, it can be used by financial institutions, telecommunication companies, airlines or shipping companies to help manage their business. This document is a technical overview of how Teradata Value Analyzer interacts with its users and their data to produce results. Its contents reflect TVA functionality as of the TVA 4.2.4 release.

Customized Modeling Environment

The TVA modeling environment is customized from the beginning to reflect a company's business environment. Once the environment has been configured, the business practices of the company are entered into TVA as a set of user-defined rules. These rules represent how the company operates with its customers or other entities. The rules may reference any attribute or characteristic of the customers that affects how the company calculates components that contribute to its profit. In addition to using actual customer attributes, TVA also integrates all of a business's activities into the profitability calculations. The activities may include items such as bill processing, customer support calls, account reviews or any other type of activity related to the customer that may affect the income or expense associated with the customer.

Relationship and Activity Based Rules

TVA calculates profitability by loading detailed data about the company's accounts, customers, relationships and activities into the TVA database. The user-defined rules are matched to the appropriate data sets and detailed calculations are performed. The result is the profit contribution for each of the items and relationships loaded into the database, along with a supporting audit trail. The results may then be aggregated, sorted and examined to any level of detail for additional analysis.

Periodicity and History

TVA is a "snapshot" model of profitability. It is typically processed on a monthly basis, but this can be changed to any periodicity desired such as a weekly, quarterly, semi-annual or annual basis. The system can be configured to retain as much history as is needed by the users.

Integration with Data Warehouses

TVA may be implemented in a stand-alone database or as a fully integrated application within an enterprise data warehouse. This flexibility results from the unique design of the TVA application and database. Several options of integrating TVA with a data warehouse are described here, but others may be developed as needed.

TVA Purpose and Process

Teradata Value Analyzer's main function is to calculate the profitability of the entities that contribute to a company's profitability (or loss). How the entities are defined varies from company to company, as discussed in more detail below. However they are defined, understanding what is driving the entities' profit contributions is critical for understanding what is driving the profitability of the business, as well as managing the business. TVA is not industry specific, that is, TVA can perform these calculations for example on a financial institution's customer base, a telecommunications company's customer base, an airline's customer base, or a shipping company's customer base.

TVA does these profitability calculations by using very detailed data for each entity, and integrating this data with a series of user-defined rules that represent how the company runs the business and recognizes income and expenses. By applying these rules to the detailed data, TVA is able to accurately calculate the profitability of each entity defined in the TVA database. This document will describe the interaction between the various types and sources of data used, and the components of TVA during the rule definition process and the calculation process.

This document is intended as a technical overview of the data processing steps of TVA rather than a business overview. TVA is essentially a business solution, so business terms are used here to describe the process and functionality, but these terms will not be fully defined here. Other technical and business documents are available for more detailed information.

Client Data Sources

In the previous section, the term "entity" was used to describe those objects for which the profitability is to be quantified. The relevant objects will vary from company to company. They may represent households, specific customers, small businesses, large corporations, products or financial commitments. In order to represent these objects in TVA some generic terms are used:

Base Profit Object

The lowest level of detail at which TVA will calculate profitability. This may be a bank customer's checking account, airline's passenger name record or a shipping company's package. A base profit object may or may not be related to an associate profit object (defined below).

Base Profit Object Event

A transaction, activity or any type of event that occurs for a base profit object. Each base profit object event must be related to a base profit object. This might be a deposit into a checking account, a traveler's inquiry to a customer call center, or checkpoint along a package's shipping path.

Associate Profit Object

A higher level of detail than the base profit object at which the company also wishes to measure profitability. This may be a customer, small business, household or large corporation. Using cost and revenues quantified at this level, a final profit or loss value is calculated for each associate profit object. An associate profit object may or may not be related to a base profit object or to other associate profit objects.

Associate Profit Object Event

A transaction, activity or any type of event that occurs for an associate profit object. Each associate profit object event must be related to an associate profit object. This might be a review of business' credit lines, an airline customer's frequent flyer mailing, or time invested in a new customer.

All of the company's profit objects and events are loaded into the TVA database and used to calculate the profit values for each profit object. TVA is designed and operates with an understanding of the explicit relationships between these data sources. These relationships are graphically represented in Figure 1.

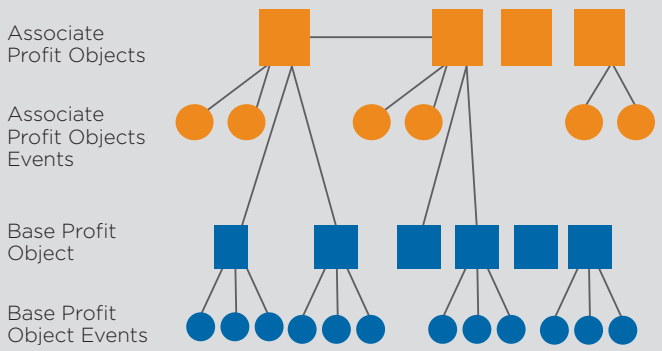

Figure 1. TVA Profit Object and Event Relationship.

General TVA Components

There are three major components of the TVA technical architecture: the Graphical User Interface (GUI); the Selector Engine; and the Calculation Engine. By having these distinct components, TVA is able to maximize flexibility, user-friendliness and processing efficiencies. The components and their interactions are summarized graphically in Figure 2.

TVA Graphical User Interfaces (GUI)

The GUI components of TVA are client-based PC applications written in C# (C Sharp). Although these applications reside on the client, all data is stored in the TVA database. The TVA GUIs allow the user to access data that needs to be updated manually by the user. This includes rule definitions, metadata, user-defined system data, and some processing information. The user actions are grouped into categories described in following sections.

Administration

Part of the TVA GUI is the Administration menu. This functionality allows administrators to perform several key tasks:

- Define Profit Object Types and Profit Object Relationships
- Define Domain code values and descriptions
- Define and identify Attributes to be available to the TVA system
- Define Cross-Charge Relationships
- Identify Driver Rate Tables to be available to the TVA system
- Identify Run Groups

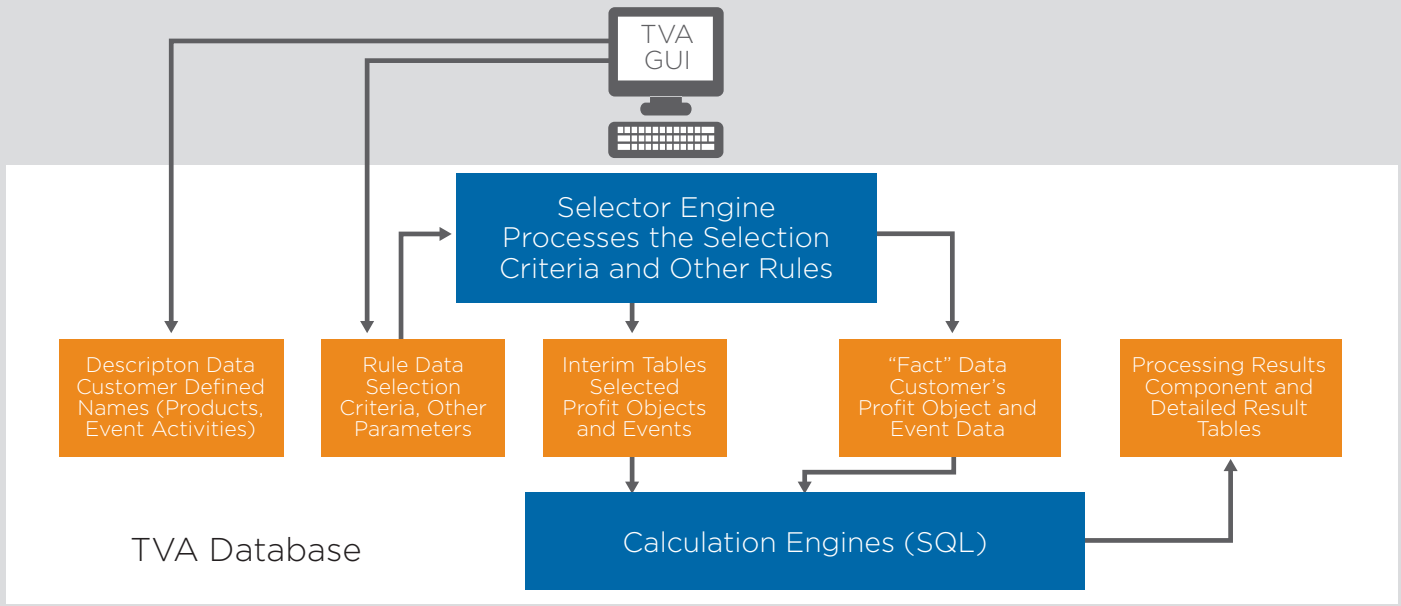

Figure 2. TVA Components and Data Overview.

Generally, these tasks are not performed by the business user, but are the responsibility of the DBA or technical staff supporting the TVA implementation.

Definitions

The TVA GUI is used to input and review many of the user-definable lists within TVA. This includes:

- Balance Types
- Product Types
- Product Groups
- Event Groups
- Rule Groups
- Independent Selection Criteria (shared by multiple Rules)
- Scenarios

These lists and groups define the client's business environment and are used throughout the rule definition process to segment the data.

Rules

To calculate the profitability of any profit object, the associated cash flows need to be classified as income or expense. Rules are defined by the users to represent the cash flows, thus there are several types of rules such as Operating Revenue (OR) rules and Direct Expense (DE) rules. Different types of rules require different input by the users. Often the rules are dependent on activities or "events" related to the profit object. The TVA GUI consists of a series of screens that allow the business user to specify what type of rule is being defined, as well as any specific parameters that are needed for that type of rule. The following is a list of the rules that may be defined using the GUI:

- Net Interest Revenue (NIR), interest revenue earned from borrowing and lending practices used primarily by financial institutions
- Other Revenue (OR)
- Direct Expense (DE)
- Indirect Expenses (IE)
- Risk Provision (RP)
- Allocated Balances (AB)
- Allocated Capital (AC)
- Earnings on Allocated Capital (EAC)
- SVA Hurdle Rate (SVA)

It is important to note here that rules consist primarily of two types of information:

1. **Calculation Information**
   These are parameters that may be needed for any of the calculation formulas. The calculation type determines the set of calculation parameters needed. This might be the unit cost for each local phone call by a customer, or the expense associated with a branch deposit at a financial institution.
2. **Selection Criteria**
   This is information that identifies the profit objects to which the rule is applied. This is a series of qualifiers that allow the user to select a specific set of profit objects. For example, if the number of phone lines a customer has is stored as one of the customer's attributes in database, then we can create selection criteria based on that attribute to identify all customers with two or more phone lines. Or, selection criteria can be created to identify all of a financial institution's regular checking accounts. Selection Criteria contained within the rule may be defined by any number of attributes:

- Base Profit Object Attributes
- Base Profit Object Related Events (or activities)
- Base Profit Object Balance Types and Amounts
- Base Profit Object Products Types and Groups
- Associate Profit Objects Attributes
- Associate Profit Object Related Events (or activities)
- Associate Profit Object Balance Types and Amounts

All Balance Types, Products and Product Groups, Profit Object Attributes and Profit Object Events are user definable. Since these attributes are all available to be used in rule definitions and selection criteria, this gives the user a great deal of flexibility in defining selection criteria and allows for very specific database segmentation to which rules may be applied.

TVA Selector Engine

The Selector Engine is another major component of the TVA architecture. It is used to match up rule definitions to the actual Profit Objects in the database. Recall that each rule has a selection criteria component. This matching is done in a two-step process:

1. **Generating SQL**
   Using the selection criteria of each rule, the Selector Engine generates SQL code that represents that Selection Criteria defined within the rule.
2. **Generating Selector Lists**
   Once the SQL code has been generated, it is executed to produce lists of Base Profit Object/Rule Number combinations and lists of Associate Profit Object/Rule Number combinations. These lists are stored in tables within the database. Where possible, Selector also extracts rule amounts used by the Calculation Engine.

By matching previously defined rules to the profit object numbers, and storing the results in tables, efficiency is greatly increased and audit trails are established.

TVA Calculation Engines

The TVA Calculation Engines are the final major component of the TVA architecture. They perform the actual calculations as defined by the rules and store the results in selected tables within the TVA database. The Calculation Engines go through a series of processes:

1. Identify which rules are to be processed as determined by the scenarios that are being run
2. Obtain the rule calculation parameters needed to perform the calculation (such as the unit cost or amount to be apportioned)
3. Use the Selector Lists to identify those profit objects to which the rule should be applied
4. Gather any other information that may be needed to perform the calculation (e.g., number of profit object events or specific profit object attributes)
5. Perform the calculation as defined by the rule
6. Store the results in predefined result tables
7. Aggregate the results as needed

The TVA Calculation Engine is really a collection of calculation engines with common functionality for most rule types (e.g., Operating Revenue, Direct Expense, Indirect Expense, etc.) and specialized functionality for rules that require non-standard parameters as input (e.g., Net Interest Revenue treatment rates). Each TVA Calculation Engine is further segmented by calculation type. For example, one Operating Revenue Rule may be an apportionment of a dollar amount across many profit objects (apportionment rule) while a different Operating Revenue Rule may calculate a revenue amount directly based on the profit object's activities (direct calculation rule).

Because Rule Type and Calculation Type segment the Calculation Engine, significant processing efficiencies can be realized when running a set of rules. Rules of similar calculation types are automatically run in parallel. The architecture of the TVA Calculation Engine is designed for maximum processing efficiency while maintaining a detailed audit trail.

Summary of Data Sources and Uses

The TVA Database consists of many tables, the majority of which are maintained and managed by TVA, so the users need not actively manage these. However, it is useful to have a basic understanding of the different types of tables within the database and how they are used. All the tables within the TVA database can be classified into one of six categories. Below are descriptions of these categories and the type of data they hold.

Metadata

To make the TVA database highly flexible and customizable, there is a set of tables containing information about the entities in the profitability model and runtime parameters. For example, these tables contain information about:

- Data Types
- Database Names
- View Names
- Profit Object Types
- Profit Object Relationships
- Attributes (Profit Object view columns)
- Attribute domain codes and description

These tables may be populated as needed to customize the TVA database for the client. This is usually done during implementation and is stable after that.

The GUI, Selector Engine and Calculation Engine use these metadata tables to identify the data that is available for use in the rule definition and calculation processes. There is no difference in functionality between a demo TVA database configuration and a customized configuration. Many of these tables can be managed and maintained by the TVA GUI, which allows users to add tables and attributes to the database and makes them available to the TVA application.

Descriptive Data

There are sets of tables used to describe the client's business environment. The Descriptive Data tables allow the user to customize TVA so that the TVA modeling environment reflects the customer's business environment. These descriptive lists include:

- Products and Product Groups
- Balance Types
- Event Groups
- Independent Selection Criteria
- Scenarios and Run Groups

This descriptive data essentially defines the client's organization to TVA in terms the client can recognize. Much of this descriptive data is used in the GUI for rule definitions and reports. Most of these tables are usually populated via the GUI, however, some of the tables can also be populated via a data load process rather than manually entering long lists of codes and descriptions.

Rule Data

There is a set of tables within TVA containing all of the rule definitions. These tables contain the components of the rule definition such as:

- Rule type information (Operating Revenue vs. Indirect Expense)
- Selection Criteria (Specific products or types of profit objects)
- Calculation Type (Apportionment vs. Direct Calculation)
- Calculation Parameters (amount to apportion or Unit Cost)

All these tables are populated and updated by the GUI when the user is defining the rules. The tables are usually static from cycle to cycle and change only as the rules are changed. Although there are a large number of these tables, they comprise a small percentage of the volume of data in the database.

Factual Data

There is a series of tables containing the client's operational data. This is the actual data from the client's systems of record and it is used to drive the profitability calculations. For example, if the client has 30 million accounts and 15 million customers, these tables maybe populated with the following types of information:

- Account numbers and attributes for each of the 30 million base profit objects (in this case the client's accounts are recognized as base profit objects by TVA)
- Events associated with each of the 30 million base profit objects representing perhaps 10's or 100's of millions of activities
- Customer numbers and attributes for each of the 15 million associate profit objects (in this case the client's customers are recognized as associate profit objects by TVA)
- Events associated with each of the associate profit objects

These tables are updated each TVA processing cycle with new data from the client's systems of record. Although these tables are relatively few, they constitute a large amount of the data in the database.

Selector Engine Results

The TVA process populates a number of interim result tables. These are used to streamline processing, decrease processing time and provide a needed audit trail to review and validate results. The Selector Engine populates two sets of these interim tables:

1. **Base Profit Object Level Selector Results**
   This consists of a set of tables containing profit object number/rule number combinations. There is a separate table for each rule type (e.g., Operating Revenue, Indirect Expense etc.). Also, these tables may contain

some profit object level information used for rule calculations. This alleviates the need for the calculation engine to retrieve this data later, thereby increasing efficiency.

2. **Associate Profit Object Level Selector Results**
   This consists of a set of tables containing profit object number/rule Number combinations. There is a separate table for each rule type (e.g., Operating Revenue, Indirect Expense etc.). Also, these tables may contain some profit object level information used for rule calculations. This alleviates the need for the calculation engine to retrieve this data later, thereby increasing efficiency.

These tables are repopulated each time the Selector Engine is run and are fairly large depending on the number of Profit Objects, as well as the number of rules being processed. TVA manages these tables internally and they require no maintenance by the users.

Calculation Engine Results

Engine result data is stored in a series of tables. Individual rule results are stored at a very granular level and then aggregated to higher levels of detail to finally arrive at a profitability component value for each Profit Object.

By using a granular approach the Calculation Engine provides the final results the user needs while maintaining a very detailed audit trail. The level of result aggregation follows the general path:

1. **Initial Results:**
   Target level (Base or Associate Profit Object)
   Rule Type (Operating Revenue, Indirect Expense)
   Rule Identifier
2. **Secondary Results:**
   Target Level (Base or Associate Profit Object)
   Rule Type (Operating Revenue, Indirect Expense)
   Rule Identifier
   Scenario Identifier
3. **Final Profitability Numbers:**
   Target Level (Base or Associate Profit Object)
   Rule Type (Operating Revenue, Indirect Expense)
   Scenario Identifier

Tables are populated for each scenario processed when the Calculation Engine is run. All result tables are managed by TVA and do not require any user intervention.

Processing Relationship Between TVA Components and Data

Now that all the data sources and the basic components of TVA have been reviewed, a more detailed relationship between the data sources and the three main components of TVA can be presented. Below is a brief discussion of how data sources interact with the components of TVA.

Metadata, Descriptive Data, Rule Data and the GUI

Via the TVA GUI the user may log onto the TVA database to manage some of the metadata within TVA. Although the GUI application resides on the client, all data is contained within the TVA database.

Not all metadata are updated via the GUI. This process is summarized in Figure 3.

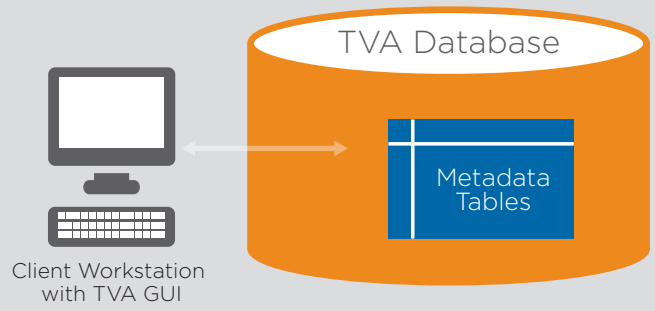

Figure 3. Metadata Update Process.

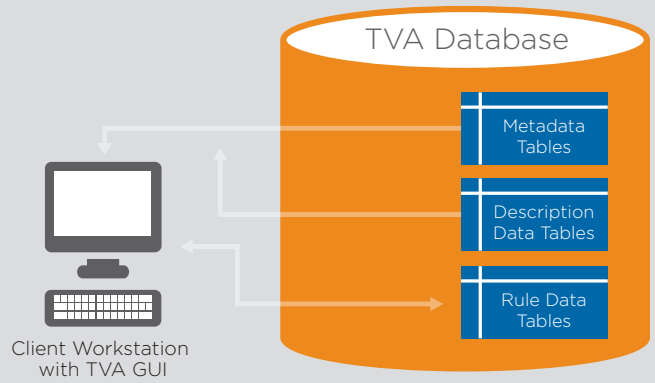

Figure 4. Rule Definition Update Process.

The TVA GUI is also used to create and modify business rules. The business rules are defined using the metadata and the descriptive data defined previously.

Factual Data, Rule Data and Interim Results

The client's factual data is matched up with the rule data using the Selector Engine. This results in a set of tables at the profit object and rule level. The selection criteria portion of each rule determines which Profit Objects will be handled by the rule. The corresponding temporary tables are managed by TVA; they can be quite large, depending on the number of Profit Objects and rules. This process is summarized in the following flow chart.

All Selector processing takes place on a database server and all data, SQL and results reside within the database.

Factual Data, Interim Results and Engine Calculations

After the rules have been defined and the Selector Engine has been run, the TVA Calculation Engine can be run. The Calculation Engines use the results of the Selector Engine, Rule data and the client's operational data to calculate the results of each rule for each applicable Profit Object. After calculating the results of the individual rules, the Calculation Engine goes through a process of aggregating the results by Profit Object and by scenario to arrive at a final profitability component value for each Profit Object in the database.

All of these tables are managed by TVA and are reset as needed for each run of the Calculation Engine. This process is summarized in Figure 6.

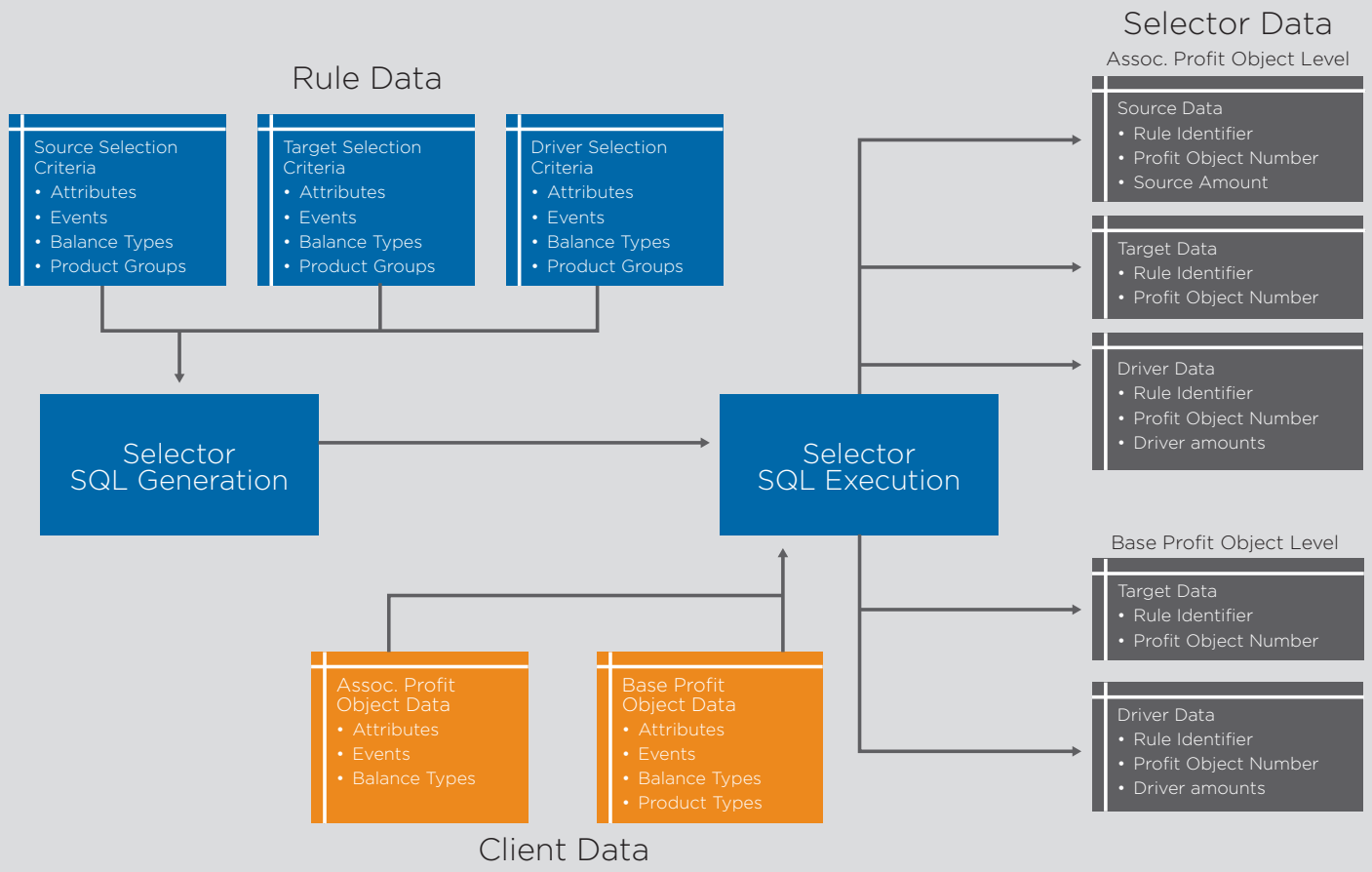

Figure 5. Selector Engine Processing Summary.

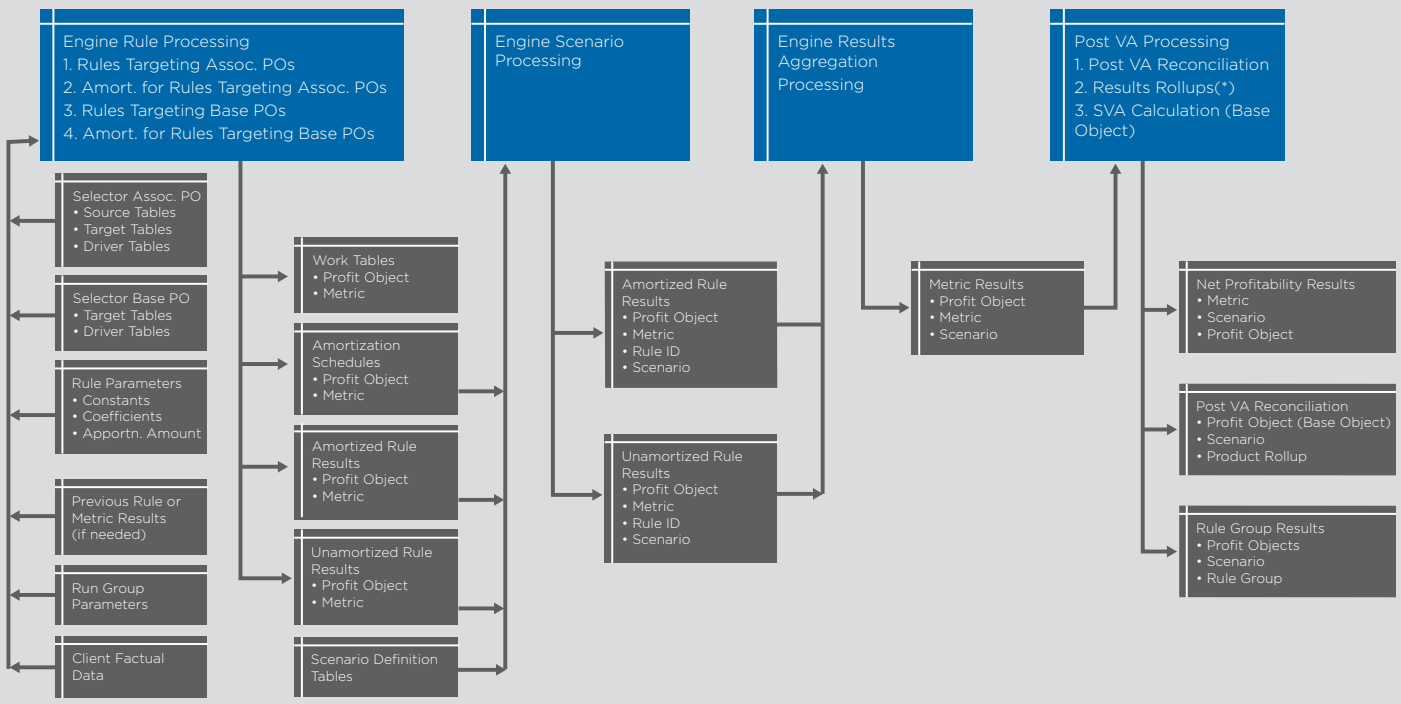

Figure 6. Calculation Engine Processing Overview.

History and Amounts of Data

TVA is a current period profitability model, which means that the Selector Engine and Calculation Engine operate on a single cycle of data, usually a monthly cycle. The current cycle number and current "as-of-date" are contained in one of the metadata tables. This is how the Engine knows what date value or cycle value to use in some of the calculations.

However, multiple months of data may be stored in the TVA database. This is done by telling TVA the data to persist and which data to delete at the end of a cycle.

Persistence of Data

Some of the data within TVA is automatically persisted at the end of a processing cycle, while other tables are automatically deleted. The user can customize this process to persist any of the tables as desired. Table 1 is a summary of which tables are persisted at the end of a cycle and those that are deleted.

These are the default settings for the cycle-end or wrap process. However, as noted before, the administrator may have TVA persist any of the tables desired.

Table Type	Persisted Status
Metadata Tables	Persisted
Definition Tables	Persisted
Rule Tables	Persisted
Factual Tables	Populated each cycle
Selector Engine Table	Deleted (Repopulated each time Selector is run)
Calculation Engine Table	Deleted (Repopulated or regenerated each time Calculation Engine is run)

Table 1.

Cycle-End/Wrap Process

After all the calculations have been done and the users have validated the results, the cycle-end process can be run. The user may edit the list of tables referenced during the wrap process, to indicate to TVA the tables to persist. The Factual tables, Selector tables or Calculation Result tables that are persisted are renamed using a naming convention within TVA.

The month-end cycle process typically includes several steps, such as those in this list:

1. Run any Customized Routines Required
2. Reporting Processes
3. Data or Results Exporting Processes
4. Aggregation Processes
5. Backup the database
6. Run the TVA Wrap stored procedure or a site-specific data persistence script
7. Update the as-of-date
8. Load a new cycle of factual data

Database Sizing

The size of the TVA database is dependent on: the number of rules, complexity of the rules (multiple selection criteria), number of profit objects, number of different kinds of activities, number of products and the kinds and amount of data that the company chooses to persist in the warehouse. All these factors impact the incremental sizing of the database and the additional processing overhead needed. However, the bulk of the data consists of factual data, interim result data and final result data. This means the size of the database fluctuates significantly during the processing cycle and grows over the life of the profitability system. This size of the database can be represented by Figure 7.

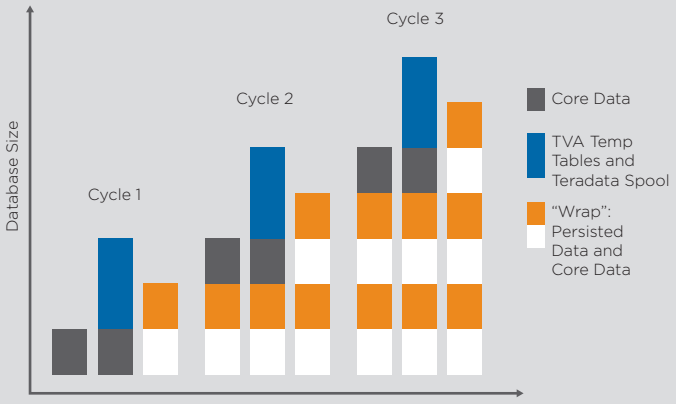

Figure 7. Database Sizing Curve.

Customizable Database Features

Although TVA does have a standard configuration, each of the factual data tables can be customized and new tables are added for use by the TVA application.

The preceding discussion referred to the data "tables" used by TVA. While the actual data are stored in tables, the TVA application operates against views of the database, rather than the actual tables. This allows for a great deal of flexibility in defining how the TVA system accesses business data.

TVA Within the Teradata Warehouse

TVA is designed to be an integral part of a Teradata warehouse. Teradata provides companies the ability to create a single, large scale, multi-functional data warehouse. Using the customization features of TVA discussed above, the user can reduce redundant data storage, minimize production cycle processes by minimizing data flow, and improve data quality and consistency.

Using views, TVA can be implemented to reference existing tables within the warehouse or TVA can be implemented as a set of tables that are an extension of a current data warehouse. TVA can also be implemented as a stand-alone database being fed from a data warehouse or the clients system of record.

The flexibility of the TVA database allows for many different implementation configurations. As an example three possible implementation configurations are depicted In Figures 8, 9 and 10.

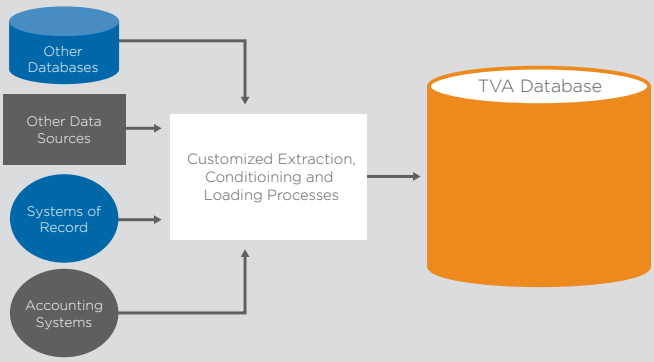

Figure 8. TVA Implemented as a Stand-Alone Database.

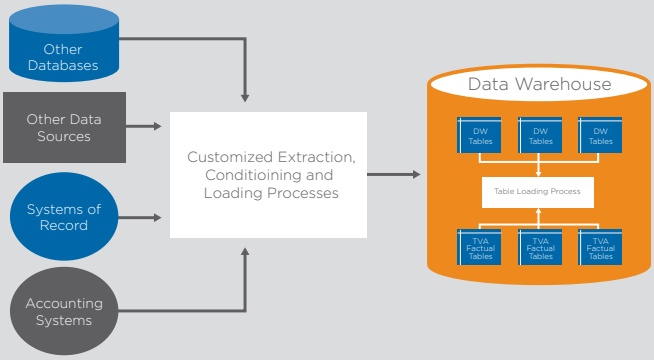

Figure 9. TVA Implemented as an Extension of a Data Warehouse.

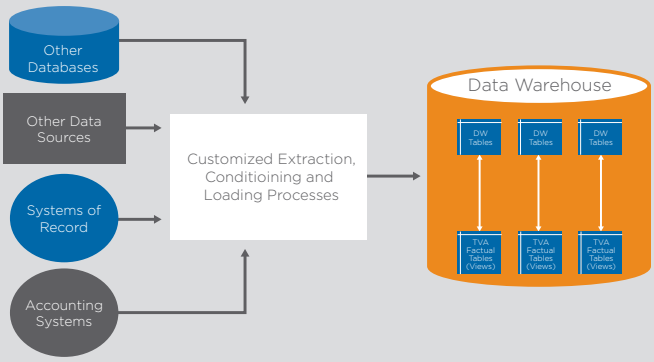

Figure 10. TVA Integrated with a Data Warehouse.

Summary

Teradata Value Analyzer is a highly flexible and powerful decision support tool used to quantify the profit contribution of all of the entities having a relationship with the company. This is done by customizing the TVA modeling environment, defining rules representing the company's business practices, and then generating very detailed, verifiable results.

TVA is able to integrate large amounts of very detailed, event-based data with the business practices of the company to produce these results. TVA can be structured to hold a large amount of history for period-to-period comparison.

TVA's technical architecture is designed to be highly scalable and to leverage the Teradata database's potential for large-scale processing. The architecture also allows for processing and data storage efficiencies. TVA can be implemented as a stand-alone application, or it can be integrated with other databases and data warehouses to varying degrees.

10000 Innovation Drive, Dayton, OH 45342 **Teradata.com**

01.15 EB6120