Getting Started with SAS® Profitability Management 1.3
## Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Introduction to the Tutorial</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Technical Support</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Additional Training and Documentation</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Business Requirements for Profitability Management</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Reporting Solutions that SAS Profitability Management Provides</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>The Baby Bank Model</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Company Background</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Baby Bank Sample Model</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Profitability Management Goals</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Source Behaviors Model Structure</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Transaction Data Collection</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Basic Steps for Building the Model</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Populate the Input Directory</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Retrieve the Tutorial Data</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Create and Populate the Input Directory</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Create an Output Directory</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>Set Up the Environment</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Add Users</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Identify Input and Output Directories to Profitability Management</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Import Tables into the Input Directory</td>
<td>17</td>
</tr>
<tr>
<td>5</td>
<td>Create a New Profitability Model</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Open the Profitability Management Client Application</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Open the Model Wizard</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Name the Model and Select the Time Dimension</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Select the Output Libraries</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Verify the Data Locations for the Model</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Select the Behavior Table</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Identify Dimension Tables</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Add Report Tables</td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>Define Transaction Table Groups</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Define Transaction Table Groups</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Define the ABMCost Group</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Define the CallCenter Group</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>Define the Revenue Group</td>
<td>42</td>
</tr>
<tr>
<td>7</td>
<td>Change Analysis Settings</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>Overview</td>
<td>47</td>
</tr>
</tbody>
</table>
Select Periods for the Cube 47
Select Formats for Numeric Measures 48

Chapter 8 Define Rules 51
Introduction 51
Define the First Rule 51
Define Another Rule 55
Import the Remaining Rules 58

Chapter 9 Associate Behaviors With Rules 61
Associate Behaviors with Rules 61
Import the Remaining Associations 62

Chapter 10 Prepare Reports 65
Define a Summary Report 65
Define a Detail Report 68

Chapter 11 Calculate the Model 73
Calculate the Model 73
Calculation – a Conceptual View 76

Chapter 12 View the Reports 83
View the Summary Report 83
View the Detail Report 101

Chapter 13 Summary of Model Elements 105
Data Requirements for the Model 105
Behavior Table 106
Period Dimension 108
Custom Dimensions 109
Rule Definition Table 114
Rule/Behavior Associations 119
Report Hierarchy 121
Report Layout 124
Transaction Tables 127

Chapter 14 Finishing Up 133
Baby Bank Conclusions 133
Additional Features 133
What to Do Next: Useful Links 134
Introduction to the Tutorial

This tutorial is intended to familiarize you with the basic business profitability modeling concepts that are used in SAS Profitability Management software. To complete the profitability modeling process, move through this tutorial from beginning to end exactly as it is presented.

Even though you may be familiar with the concepts of SAS Profitability Management and customer detailed profitability reporting, working through this tutorial will make you familiar with the SAS Profitability Management software – the concepts, terminology, commands, dialog boxes, and Web reporting tools.

The key to computing segment profitability is the ability to accurately associate costs with business segments. The heart of the problem is the difference in how revenue and costs are managed and tracked in accounting systems. Revenue is generated by the customer. It is usually automatically associated with business segments by sales order, invoicing, or funds transfer systems. This makes it relatively easy to perform business segment analysis using revenue alone. In contrast, costs are not as easily associated with business segments. IT, operations, support, distribution, and administration functions generally support many business segments simultaneously.

These shared and indirect costs should ideally be tracked based on logical cause-and-effect relationships to products, services, channels and customers. Traditional cost systems violate this process by using arbitrary cost allocations with broad averages (such as the number of customers).

SAS Profitability Management is a highly flexible analysis tool that provides the ability to associate a cost and revenue with individual business transactions. Using the software, you can calculate profit and loss based on individual transactions. SAS Profitability Management provides the level of reporting detail that allows business managers to actively manage profit as a performance metric. The product enables business managers to track the profit performance of customer groups or individual customers, product groups or individual Stock-keeping units (SKU), channels or specific branches or combinations of these dimensions, or others as defined by the customer.
Technical Support

If you encounter problems that you cannot solve by reading the online help or this tutorial, refer to the SAS technical support home page at: http://support.sas.com/techsup/intro.html.

Our support goal is to provide you with the resources you need to answer any questions or solve any problems you encounter when using SAS software. We provide a variety of tools to help you solve problems on your own and a variety of ways to contact our technical support staff when you need help. Free technical support is available to all sites licensing SAS software. This includes unlimited telephone support for customers in North America. Customers outside of North America should contact their local SAS office.

Additional Training and Documentation

Additional training and tutorials can be found at www.sas.com and www.bettermanagement.com. Bettermanagement.com offers in-depth domain content about selected management concepts that are aimed at improving an organization’s performance. The Web site is a comprehensive source for performance management information and resource including Web casts, white papers, training, and tutorial materials. Topics that are covered on the Web include value-based management, profitability analysis, strategic enterprise management, activity-based costing and management, business intelligence, analytic analysis, scorecarding, and performance measurement.

The documentation for SAS Profitability Management can be found at http://support.sas.com/documentation/onlinedoc/pm/.

Business Requirements for Profitability Management

Profitability management is the most significant issue for any corporation. Profitability is derived from both analyzing the revenue performance for a given business dimension (customer, product, region, channel, customer segment), and analyzing the costs directly associated with serving those customers and providing those products. The critical challenge for business is to appropriately correlate revenue and costs into a meaningful profit and loss statement at the level of detail.

In the growing level of corporate complexity and detailed transactional information tracing corporate and customer interactions, detailed data analysis can be overwhelming. Business managers need a clear tool to deal with millions of detailed transactions and to produce an actionable profit and loss statement at a customer detailed level. Businesses with millions of customer transactions have the most to gain from implementing SAS Profitability Management. This solution is most crucial in the telephone and banking industries, where customer differentiation can be most decisive to overall corporate profitability.
With SAS Profitability Management, decision makers can define the segmentation reports that they need on the fly. SAS enables business managers to drill-down into revenue and cost categories so they can manage profit as a performance metric.

**Reporting Solutions that SAS Profitability Management Provides**

SAS Profitability Management matches cost and revenue behaviors to detailed transactions. The association of the behaviors to the transactions is based upon a wizard-driven rules engine. The resulting calculated detailed transaction tables are then used as source content for a profit and loss statement.

The web-deployed profit and loss statement that SAS Profitability Management provides:

- Is based upon a custom-defined report layout and can support complex calculation logic to present your company's reporting needs
- Is drillable for increasing level of details (revenue breakdowns or contributing costs details)
- Is drillable based upon dimensional hierarchies
- A summary cube report that can be:
  - Created to include only specific dimensions
  - Summarized by depths noted in any dimension
- A detailed cube report that can be:
  - Defined with filter logic for a single dimension member
  - Run on the fly.
Reporting Solutions that SAS Profitability Management Provides

Chapter 1
Company Background

The Baby Bank is a small sample model focused on the banking industry. The company has branches and also services customers through call centers. It performs both retail and corporate banking. They are trying to produce customer profitability so that the banking managers can view the details of a customer’s behavior and profitability value to Baby Bank. By having this detailed profit and loss information at the managers’ fingertips, they can make better management decisions on how to service the existing customers and what specific types of customers to focus on.

Baby Bank Sample Model

The Baby Bank model consists of the following:

- Five dimensions
  1. Channel (3 members: ATM, branch, and call center)
  2. Customer (101 members: 50 individuals and 51 businesses)
  3. Customer type (4 members: corporate banking, private banking, retail consumer banking, and small business banking)
  4. Product (14 members: credit products, credit-unsecured, credit-secured, deposit products, term, savings, recurring, checking, fee-based products, other products, revolving credit products, overdrafts, credit cards, and third-party products)
  5. Regions (204 members: by area, country, state, and city)
- Two periods (three levels each: year, quarter, and scenario)
  1. 2006_q4_actual
  2. 2006_q4_budget
- Six transaction tables with a total of 4,180 records
  - ABMCost:
    1. Load_Trans_q4a
    2. Load_Trans_q4b
  - CallCenter:
    3. CallCenter_q4a
    4. CallCenter_q4b
  - Revenue:
    5. Revenue_q4a
    6. Revenue_q4b
Profitability Management Goals

Baby Bank is constantly growing its customer base. They have been very effective with their new marketing campaign. But it has not been growing in profitability, so the previous goals for increasing revenue as the primary goal for Baby Bank are being re-evaluated. The new focus for Baby Bank is on controlled profitable growth. So the profitability of all customers will be evaluated on a detailed basis. Once Baby Bank can determine what are the characteristics of profitable customers, internal controls can be implemented to decrease the costs associated with high cost activities. The ultimate goal is to move existing customers to profitability through behavior changes or increased fees. Recruiting more profitable new customers will be keyed on marketing efforts to target new customers with behaviors that will be profitable to Baby Bank.

Source Behaviors Model Structure

In the Baby Bank example model, the source content for the behaviors came from an activity-based costing model. This ABC model started with general ledger expenses and assigned the costs to activities based upon staff efforts required and capital employed. Then the activities were traced to cost objects by channel (ATM, branch, call center), by product supported (savings, checking, mortgage), and by transaction type (open account, check balance, make deposit). These cost objects from ABM by channel, product, and transaction type are the source behaviors feeding into the SAS Profitability Management model.

In implementing SAS Profitability Management, behavior costs can come from any source. ABC is not a required source for SAS Profitability Management. As long as the appropriate costs are traced for each behavior at an appropriate level of granularity and an appropriate cost basis unit or total has been reliably calculated, it will be a good source feed for behaviors into SAS Profitability Management.

Transaction Data Collection

Operational transaction data tracing customer interactions with the company are critical to the effectiveness of the SAS Profitability Management model. This transaction detail can come from a multitude of source systems within your corporation. For the Baby Bank model, there are three sources for the transactional costs. The first is the transaction register for all interactions on an account basis (ATM and branch transactions). The second is a register for all call center interactions on a customer and product basis. The third source system is a register for all customer revenue tracing fees for credit cards and interest payments.

Basic Steps for Building the Model

Using SAS Profitability Management involves the following tasks:

1. Populate the input directory
2. Set up the environment
3. Create a new profitability model
4. Define transaction table groups
5. Define rules and associate them with behaviors
6. Calculate the model
7. Prepare reports
8. View the reports
CHAPTER 3

Populate the Input Directory

Retrieve the Tutorial Data
Create and Populate the Input Directory
Create an Output Directory

Retrieve the Tutorial Data

The sample SAS tables for this tutorial are on the CD as part of the middle-tier install files. Follow the instructions in the SAS Profitability Management 1.3 Installation Instructions to install the tutorial data on a machine that your workspace has access to.

The following files are provided on the installation CD for use with this tutorial:

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>association.sas7bdat</td>
<td>Holds the relationships between the behaviors (where costs reside) and the rules (which define the assignment logic to transactions).</td>
</tr>
<tr>
<td>behavior.sas7bdat</td>
<td>Holds all of the transactions and costs (total or unit)</td>
</tr>
<tr>
<td>callcenter_q4a.sas7bdat</td>
<td>Call center transaction details for the 4&lt;sup&gt;th&lt;/sup&gt; quarter actual</td>
</tr>
<tr>
<td>callcenter_q4b.sas7bdat</td>
<td>Call center transaction details for the 4&lt;sup&gt;th&lt;/sup&gt; quarter budget</td>
</tr>
<tr>
<td>dim_channel.sas7bdat</td>
<td>Hierarchy definition for the channel dimension</td>
</tr>
<tr>
<td>dim__customer.sas7bdat</td>
<td>Hierarchy definition for the customer dimension</td>
</tr>
<tr>
<td>dim__custtype.sas7bdat</td>
<td>Hierarchy definition for the customer type dimension</td>
</tr>
<tr>
<td>dim__period.sas7bdat</td>
<td>Hierarchy definition for the time period dimension</td>
</tr>
<tr>
<td>dim__product.sas7bdat</td>
<td>Hierarchy definition for the product dimension</td>
</tr>
<tr>
<td>dim__region.sas7bdat</td>
<td>Hierarchy definition for the region dimension</td>
</tr>
<tr>
<td>load_trans_q4a.sas7bdat</td>
<td>ABMCost transaction details 4&lt;sup&gt;th&lt;/sup&gt; quarter actual</td>
</tr>
<tr>
<td>load_trans_q4b.sas7bdat</td>
<td>ABMCost transaction details 4&lt;sup&gt;th&lt;/sup&gt; quarter budget</td>
</tr>
<tr>
<td>reportHierarchy.sas7bdat</td>
<td>Report hierarchy defining the drill-down detail</td>
</tr>
<tr>
<td>reportLayout.sas7bdat</td>
<td>Report layout defining the profit and loss calculations</td>
</tr>
<tr>
<td>revenue_q4a.sas7bdat</td>
<td>Revenue transaction details 4&lt;sup&gt;th&lt;/sup&gt; quarter actual</td>
</tr>
<tr>
<td>revenue_q4b.sas7bdat</td>
<td>Revenue transaction details 4&lt;sup&gt;th&lt;/sup&gt; quarter budget</td>
</tr>
<tr>
<td>rules.sas7bdat</td>
<td>Definition of the assignment rules logic</td>
</tr>
</tbody>
</table>

For additional details describing the source tables required for SAS Profitability Management, please refer to “Summary of Model Files” on page 105.
Create and Populate the Input Directory

You must create a directory on your SAS Profitability Management server to hold the source files for the tutorial. These are the files that you just extracted from tutorial.zip.

1 Create a directory on the SAS Profitability Management server.

   You can place the directory where you like and name it what you like. For purposes of illustration, we assume that you create and name it as follows:

   C:\SAS\ProfitabilityManagement\Bank_In

2 Give yourself (or whoever is doing the tutorial) read, modify, and write access to the directory. If you have an access problem in doing the tutorial, that permission has probably not been granted.

3 Copy the files from your local machine to the directory that you just created on the SAS Profitability Management server.

Create an Output Directory

You must also create a directory on your SAS Profitability Management server to hold the calculated transaction tables. This is where SAS OLAP cube generation gets its source content for generating cubes. Do not put anything into the output directory.

1 Create a directory on the SAS Profitability Management server.

   You can place the directory where you like and name it what you like. For purposes of illustration, we assume that you create and name it as follows:

   C:\SAS\ProfitabilityManagement\Bank_Out

2 Give yourself (or whoever is doing the tutorial) read, modify, and write access to the directory. If you have an access problem in doing the tutorial, this permission has probably not been granted.
Add Users

Before anyone can begin using SAS Profitability Management, you must use the SAS Management Console as an administrator to add a Profitability Management user account.

1. Log on to the SAS Management Console as an administrator.
2. Select the **Foundation** repository.
3. Select **Environment Management**.
4. Right-click **User Manager**, and select **New > User**.

5. Name the new user, and enter other user information on the **General** tab.
6. Click the **Groups** tab, and add **PM Users** to the list of groups that the user is a member of.
7 Click the **Logins** tab, and then click **New**.

- Enter the new user’s **User ID**, including domain, on the network.
- Leave the **Password** field blank (it comes from the Profitability Management logon).
- Leave the **Confirm Password** field blank (it comes from the Profitability Management logon).
- Select **DefaultAuth** for the authentication domain.

8 Leave the **Authorization** tab blank.
9 Click OK.

The user is created. You can log off the SAS Management Console as administrator.

---

**Identify Input and Output Directories to Profitability Management**

Having created an input and output directory on the server, you must let SAS Profitability Management know where they are located. To do so, you use the SAS Management Console:

1 Log on to the SAS Management Console with the user account that you just created.
2 Select the Foundation repository.
3 Expand Data Library Manager.
4 Right-click SAS Libraries.
5 Click New Library.

6 Select SAS Base Engine Library as the type of library to be created, and then click Next.

7 Name the library Bank_In, and then click Next.

You can give the library any name you want (eight characters maximum). For convenience, we give it the same name as the directory.
8 Type **Bank_In** as the Libref name. For convenience, again, we use the name of the directory.

9 Specify **BASE** as the engine type.

10 Specify the library directory path, and then click Next.

   **Note:** If the path does not exist to be selected in the list-box of available paths, then click **New** to create the path.

The directory can be anywhere on the server, but we have assumed that you created it at: C:\SAS\ProfitabilityManagement\Bank_In.
11 Click **Next**, and then click **Finish**.

12 Click **View > Refresh** from the menu bar to see the library listed.

Repeat this process for the output library. Be sure to give it a different name and libref, such as **Bank_Out**, and specify a different directory path – where your output directory is located.

**Note:** You can choose to save the output tables in a database rather than in SAS data sets. For information, see the following section.

---

**Saving Output Tables in a Database**

If you want to save your output tables in a database library, then do the following to create your output library:

1 Log on to the SAS Management Console with the user account that you just created.

2 Select the **Foundation** repository.

3 Expand **Data Library Manager**.

4 Right-click **SAS Libraries**.

5 Click **New Library**.

6 Select among the Database Libraries for the type of library to be created, and then click **Next**.
The following table shows the appropriate choices of Database Library for each of the possible types of database.

<table>
<thead>
<tr>
<th>Database</th>
<th>Appropriate Database Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>DB2 Library for Unix and PC Hosts</td>
</tr>
<tr>
<td></td>
<td>DB2 Library for z/OS Hosts</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>ODBC Library (Windows)</td>
</tr>
<tr>
<td></td>
<td>OLE DB Library (Windows)</td>
</tr>
<tr>
<td></td>
<td>Microsoft SQL Server Library for Unix Hosts</td>
</tr>
<tr>
<td>MySQL</td>
<td>MySQL Library</td>
</tr>
<tr>
<td>Oracle</td>
<td>Oracle Library</td>
</tr>
<tr>
<td>MS Access</td>
<td>OLE DB Library (Windows)</td>
</tr>
</tbody>
</table>

7 Name the library, and then click **Next**.

You can give the library any name you want (eight characters maximum).

8 Type a **Libref** name, and then click **Next**.

9 Specify the **Database Server** and Login information, and then click **Next**.

10 Optionally, select the SAS server where this library is to be assigned, and then click **Next**.

11 Review the library information, and then click **Finish**.

12 Click **View > Refresh** from the menu bar to see the library listed.

13 Right-click the newly created library and select **Properties**.

The Properties window opens.

14 Select the **Extended Attributes** tab.

15 Click **New**.

16 In the **Field Name** column, type **DBMSType**.
17 In the **Value** and **Description** columns, type one of the following pairs depending upon your database (the Description is optional).

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQL</td>
<td>Microsoft SQL Server</td>
</tr>
<tr>
<td>Oracle</td>
<td>Oracle</td>
</tr>
<tr>
<td>DB2</td>
<td>DB2</td>
</tr>
<tr>
<td>MySql</td>
<td>MySql</td>
</tr>
<tr>
<td>Jet</td>
<td>MS Access</td>
</tr>
</tbody>
</table>

For example, as shown in the following graphic:

---

**Import Tables into the Input Directory**

Even though you used the operating system to copy source files into the input directory, SAS does not yet know about them. Use the SAS Management Console to store metadata regarding the files by importing them (only structural metadata is imported, and not the actual content of the files).

**Note:** If you modify the input tables, you must **reimport** them so that the metadata that is maintained by SAS Management Console is updated (column names and data types).

1. Log on to the SAS Management Console with the user account that you just created.

2. Select the **Foundation** repository.

3. Expand **Data Library Manager**.

4. Expand **SAS Libraries**.

5. Select the input library, **Bank_In**.
6 Select **Actions > Import Tables**.

The Connect to SAS window opens.

7 Select **SASMain** as the SAS server, and click **Next** to log on to SASMain.

8 Verify that the input library is correct, and then click **Next**.
9 Select the tables to be imported, and then click **Next**.

10 View the summary of which tables are to be imported, and then click **Finish**.

After the tables have been imported, the SAS library should look like the following:
Introduction

Creating a new profitability model involves the following steps:

1. Naming the model and identifying its time dimension
2. Specifying libraries for model output
3. Verifying the location for the model definition
4. Identifying the behavior table for the model
5. Identifying custom dimension tables for the model
6. Identifying the report hierarchy and report layout tables

Open the Profitability Management Client Application

Open the SAS Profitability Management rich client application.
1 Specify the user ID and password that you created in the SAS Management Console.

2 Specify the SAS Profitability Management server. This is dependent on your installation.

3 Specify the port. 8561 is the default for the SAS Metadata Server.

4 Click Log On.

The SAS Profitability Management rich client application opens.
Open the Model Wizard

1. Select **File > Organize Models** (or click the Organize Models icon).
2. Click **New Model**.

   ![Image of Model Wizard]

   The Model wizard opens.

   **Note:** If this is the first time that the SAS Profitability Management client has ever been invoked, the Model wizard opens automatically.

Name the Model and Select the Time Dimension

1. Name the model, for example **Baby Bank**.
2. Click **Select** to select the time dimension table.
3. Select **DIM_PERIOD** as the time dimension table.

   We are assuming that the library in which you placed the input files is named **Bank_In**.
4  Click OK.

The time dimension table defines the time periods in the model. The number of periods in a model varies with the reporting needs of a business. The sample time dimension table, DIM_PERIOD, is shown in the following graphic (you can click the Preview button in the Model wizard to see the table).

<table>
<thead>
<tr>
<th>ID</th>
<th>L1_Scenario</th>
<th>L2_Year</th>
<th>L3_Quarter</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006_Q4_Actual</td>
<td>Actual</td>
<td>2006</td>
<td>2006_Q4</td>
</tr>
<tr>
<td>2006_Q4_Budget</td>
<td>Budget</td>
<td>2006</td>
<td>2006_Q4</td>
</tr>
</tbody>
</table>

5  Click Next.

Select the Output Libraries

1  Specify the Analysis view name.

The analysis view name is used as the name of the database view that is created to join the transaction output tables into a single virtual fact table that the OLAP cube is built from. By default, the analysis view name is the same as the model name.

2  Specify the Analysis view library.

For this tutorial, we have created a single output directory, Bank_Out. You can select this directory to store the analysis view in.

Note: The Analysis view library must be a SAS Base Engine Library.

3  Select the Output table library.
While it can be helpful to have separate directories to hold the analysis view and calculated transaction tables, for this tutorial we have created a single output directory, Bank_Out.

**Note:** The output table library can be either a SAS Base Engine Library or a Database Library. If it is a Database Library, then it must be separate from the Analysis View Library, which is required to be a SAS Base Engine Library. For information on using a Database Library, see “Saving Output Tables in a Database” on page 15.

Assuming, for this tutorial, that you choose the same directory to hold all your tutorial output, the page appears as follows:

4 Click Next.

### Verify the Data Locations for the Model

1 Verify the **Server folder** where data associated with the model is stored.

2 Verify the **Metadata folder** where model metadata is stored.

**Note:** Both storage locations were established during installation. Do not change them now. At this point, the dialog box is informational only.

3 Click Next.
Select the Behavior Table

1. Click **Select** to select the behavior table for the model.
2. Select **BEHAVIOR** from the available tables.

3. Click **OK**.
4. Identify fields in the behavior table.

A behavior table must contain a field for each of the following: ID, Name, Unit Value, Total Value, and Period. You must identify which field is which in the table selected.

By default, SAS Profitability Management assumes that these fields are named as follows: “ID”, “Name”, “UnitValue”, “TotalValue”, and “Time”. If fields with those names exist in the behavior table, then they are automatically mapped. If the fields are named differently, then you must match them manually.

The sample behavior table, BEHAVIOR, contains these fields, so they are mapped automatically.
5 Click Next.

What is a Behavior Table?

Behaviors, typically, are things that your customers do. For example, the customers of a bank can check their balance, make deposits, transfer funds, and make withdrawals.

The following graphic shows part of the sample behavior table, BEHAVIOR. Notice the following:

- Each behavior (each row of the behavior table) has either a non-zero UnitValue, or a non-zero TotalValue, but not both.
- The table contains an extra column named “AssignmentRule”. All of your source tables can have extra columns. This particular field is an extra column containing a character string that is used in filtering the table to select a subset of behaviors.
Identify Dimension Tables

1. Click **Add** to select the custom dimension tables that make up the cube.

2. Select the following dimension tables, and click **Add**:
   
   - DIM_CHANNEL
   - DIM_CUSTOMER
   - DIM_CUSTYPE
   - DIM_PRODUCT
   - DIM_REGION

   **Note:** You can select multiple dimensions by using the Shift or Ctrl key.

3. Click **OK**.
The dimension tables are added to the model as shown in the following graphic.

4 Click Next.

**Add Report Tables**

1 Click Select for the Report hierarchy, and select REPORTHIERARCHY.
2 Click Select for the Report layout, and select REPORTLAYOUT.
3 Click **Finish**, and verify that the new model, **Baby Bank**, is added to the list of models.
Define Transaction Table Groups

Transaction tables that share the same column structure are organized into table groups. A single Profitability Management model is likely to have multiple table groups. Multiple rules likely use the same source table group.

**Note:** There is one transaction table for each period in a model.

For the Baby Bank model, you define three transaction table groups:

- **ABMCost** holds the transaction costs that relate to the ATM activities and the detailed branch activities.
- **CallCenter** holds the details of the activities performed at the call center.
- **Revenue** holds the detailed revenue for all of the customers noted.

Enter these table group names exactly because they must match the names in a predefined rules definition table.

**Define the ABMCost Group**

Defining transaction table groups is a two-part process:

- Define the transaction table group.
- Add transaction tables to the group (and associate each transaction table with a time period).
Define the Transaction Table Group

1. Select the **Transaction** workspace.
   Make sure that the **Baby Bank** model is selected.

2. Click **Add table group** (or the Add table group icon).
   The Add Table Group window opens.

3. Name the group **ABMCost**, and then click **Next**.

4. Select **LOAD_TRANS_Q4A** as the table whose schema serves as the schema for the table group.
Define Transaction Table Groups

Define the ABMCost Group

Note: All the tables in a table group share the same schema, and each table is associated with a different time period.

5 Click Next.

6 Specify whether each column in the table is
   - Text
   - Numeric
   - Dimension member (if a dimension member, specify which one).

The specifications for LOAD_TRANS_Q4A should look like those in the following graphic:

7 Click Finish.

The table group, ABMCost, is added to the list.
Add Tables to the Group

1. Click **Add transaction tables** (making sure that **ABMCost** is selected).

Or click the icon.

The Add Transaction Tables window opens.

2. Select **LOAD_TRANS_Q4A** and **LOAD_TRANS_Q4B**.

3. Click **Add**.

The tables are added.
4 Click **Next**.

The Period window opens.

5 Associate each of the tables with a period as follows:

<table>
<thead>
<tr>
<th>Table</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOAD_TRANS_Q4A</td>
<td>Actual &gt; 2006 &gt; 2006_Q4</td>
</tr>
<tr>
<td>LOAD_TRANS_Q4B</td>
<td>Budget &gt; 2006 &gt; 2006_Q4</td>
</tr>
</tbody>
</table>

6 Click **Finish**.
The tables are added to the group.

Transaction Table LOAD_TRANS_Q4A

The following graphic shows the transaction table, LOAD_TRANS_Q4A, and how each of its columns is identified as either text, numeric, or a dimension member. You can see that each column that is identified as a dimension member contains values that correspond to values in the ID field of the corresponding dimension table.
Define Transaction Table Groups

Define the CallCenter Group

Repeat the same general process to define a transaction table group named CallCenter:

- Define the transaction table group
- Add transaction tables to the group (and associate each transaction table with a time period).

Define the Transaction Table Group

1. Select the Transaction workspace.
2. Click Add table group.
3. Name the group CallCenter, and then click Next.
4 Select CALLCENTER_Q4A as the table whose schema serves as the schema for the table group.

5 Click Next.

6 Specify whether each column in the table is text, numeric, or a dimension member as shown in the following graphic, and then click Finish.
The table group, CallCenter, is added to the list.

Add Tables to the Group

1. Click Add transaction tables (making sure that CallCenter is selected).
2. Add CALLCENTER_Q4A and CALLCENTER_Q4B to the list, and then click Next.
3. Associate each of the tables with a period as shown:

4. Click **Finish**.

   The tables are added to the group.
Define Transaction Table Groups

Transaction Table CALLCENTER_Q4A

The following graphic shows a portion of the contents of the transaction table CALLCENTER_Q4A.
Define the Revenue Group

Repeat the same general process to define a transaction table group named Revenue.

Define the Transaction Table Group

1. Click Add table group.
2. Name the group Revenue, and then click Next.
3. Select REVENUE_Q4A as the table whose schema serves as the schema for the table group.
4. Click Next.
5. Specify whether each column in the table is text, numeric, or a dimension member as shown in the following graphic, and then click Finish.
Define Transaction Table Groups

Define the Revenue Group

The table group, Revenue, is added to the list.

Add Tables to the Group

1. Click Add transaction tables (making sure that Revenue is selected).
2. Add REVENUE_Q4A and REVENUE_Q4B to the list, and then click Next.
3 Associate each of the tables with a period as shown:

4 Click **Finish**.

The tables are added to the group.
Define Transaction Table Groups

Define the Revenue Group

Transaction Table REVENUE_Q4A

The following graphic shows a portion of the contents of the transaction table REVENUE_Q4A.
### REVENUE_Q4A

<table>
<thead>
<tr>
<th>Channel</th>
<th>CustID</th>
<th>Product</th>
<th>CustType</th>
<th>Region</th>
<th>AMT</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM</td>
<td>00001</td>
<td>OVD</td>
<td>PRB</td>
<td>Reg. 67</td>
<td>353.3</td>
<td>10002</td>
</tr>
<tr>
<td>ATM</td>
<td>00001</td>
<td>OVD</td>
<td>PRB</td>
<td>Reg. 67</td>
<td>8.16</td>
<td>12002</td>
</tr>
<tr>
<td>ATM</td>
<td>00001</td>
<td>OVD</td>
<td>PRB</td>
<td>Reg. 67</td>
<td>20.47</td>
<td>13001</td>
</tr>
<tr>
<td>ATM</td>
<td>00001</td>
<td>OVD</td>
<td>PRB</td>
<td>Reg. 67</td>
<td>25.59</td>
<td>13002</td>
</tr>
<tr>
<td>ATM</td>
<td>00001</td>
<td>OVD</td>
<td>PRB</td>
<td>Reg. 67</td>
<td>5.12</td>
<td>14001</td>
</tr>
<tr>
<td>ATM</td>
<td>00002</td>
<td>CRC</td>
<td>PRB</td>
<td>Reg. 188</td>
<td>4353.35</td>
<td>10001</td>
</tr>
<tr>
<td>ATM</td>
<td>00002</td>
<td>CRC</td>
<td>PRB</td>
<td>Reg. 188</td>
<td>1744.98</td>
<td>12001</td>
</tr>
<tr>
<td>ATM</td>
<td>00002</td>
<td>CRC</td>
<td>PRB</td>
<td>Reg. 188</td>
<td>134.53</td>
<td>13001</td>
</tr>
<tr>
<td>ATM</td>
<td>00002</td>
<td>CRC</td>
<td>PRB</td>
<td>Reg. 188</td>
<td>221.31</td>
<td>13002</td>
</tr>
<tr>
<td>CCT</td>
<td>05845</td>
<td>FBP</td>
<td>CRB</td>
<td>Reg. 705</td>
<td>276.64</td>
<td>13002</td>
</tr>
<tr>
<td>CCT</td>
<td>05845</td>
<td>OTP</td>
<td>CRB</td>
<td>Reg. 705</td>
<td>55.33</td>
<td>14001</td>
</tr>
<tr>
<td>CCT</td>
<td>05845</td>
<td>OTP</td>
<td>CRB</td>
<td>Reg. 705</td>
<td>23.38</td>
<td>13001</td>
</tr>
<tr>
<td>CCT</td>
<td>05845</td>
<td>OTP</td>
<td>CRB</td>
<td>Reg. 705</td>
<td>23.22</td>
<td>13002</td>
</tr>
<tr>
<td>CCT</td>
<td>05845</td>
<td>OTP</td>
<td>CRB</td>
<td>Reg. 705</td>
<td>5.04</td>
<td>14001</td>
</tr>
<tr>
<td>CCT</td>
<td>05846</td>
<td>UCR</td>
<td>SBB</td>
<td>Reg. 1033</td>
<td>6478.6</td>
<td>10002</td>
</tr>
<tr>
<td>CCT</td>
<td>05846</td>
<td>UCR</td>
<td>SBB</td>
<td>Reg. 1033</td>
<td>332.17</td>
<td>13001</td>
</tr>
<tr>
<td>CCT</td>
<td>05846</td>
<td>UCR</td>
<td>SBB</td>
<td>Reg. 1033</td>
<td>415.22</td>
<td>13002</td>
</tr>
<tr>
<td>CCT</td>
<td>05846</td>
<td>UCR</td>
<td>SBB</td>
<td>Reg. 1033</td>
<td>93.04</td>
<td>14001</td>
</tr>
<tr>
<td>CCT</td>
<td>05847</td>
<td>FBP</td>
<td>SBB</td>
<td>Reg. 269</td>
<td>21.04</td>
<td>13001</td>
</tr>
<tr>
<td>CCT</td>
<td>05847</td>
<td>FBP</td>
<td>SBB</td>
<td>Reg. 269</td>
<td>26.3</td>
<td>13002</td>
</tr>
<tr>
<td>CCT</td>
<td>05847</td>
<td>FBP</td>
<td>SBB</td>
<td>Reg. 269</td>
<td>5.26</td>
<td>14001</td>
</tr>
<tr>
<td>CCT</td>
<td>05847</td>
<td>OVD</td>
<td>CRB</td>
<td>Reg. 448</td>
<td>3208.39</td>
<td>10002</td>
</tr>
<tr>
<td>CCT</td>
<td>05847</td>
<td>OVD</td>
<td>CRB</td>
<td>Reg. 448</td>
<td>164.5</td>
<td>13001</td>
</tr>
<tr>
<td>CCT</td>
<td>05847</td>
<td>OVD</td>
<td>CRB</td>
<td>Reg. 448</td>
<td>298.63</td>
<td>13002</td>
</tr>
<tr>
<td>CCT</td>
<td>05847</td>
<td>OVD</td>
<td>CRB</td>
<td>Reg. 448</td>
<td>41.13</td>
<td>14001</td>
</tr>
</tbody>
</table>
Overview

Analysis settings control the formatting of numbers in cube reports. The value field is particularly critical to be correctly formatted. The value field displays revenue and cost in the profit and loss reporting on the web.

1 Select Edit > Analysis Settings.

The Analysis Settings window opens.

Select Periods for the Cube

1 Check Include transaction tables from all periods.

The view is created with a join across all available calculated transaction tables.
The selected periods are included in the view that is used as the source for the OLAP cubes and, thus, the reporting.

If you want the cubes and reporting produced to include only a selected set of periods, use this dialog box to select the periods and, thus, define the table joins in the view.

Each transaction table represents a single period. These calculated transaction tables are stored in the output table library. The cubes represent results across multiple periods, but the cubes do not need to include every calculated period.

**Note:** The periods that you select for inclusion in a cube must also be chosen for calculation when you calculate the model. Unless a period has been calculated, it cannot be included in a cube.

---

### Select Formats for Numeric Measures

1. Click the **Measures** tab.

   The Measures tab lists all numeric properties available to the model. All numerical properties are available to be included in profit and loss reporting. This includes:
   - all entered numeric properties from the transaction tables
   - all the calculated properties:
     - Quantity – a numeric value based upon the rules driver formula
     - Value – the calculated behavior amounts (revenue or cost)

2. For the **VALUE** measure, specify **DOLLAR5.2**.

   Value is the calculated cost or revenue for a transaction. The method of its calculation depends upon whether the behavior table row that is accessed by an assignment rule contains a unit value or a total value. See “Calculation – a Conceptual View” on page 76.

   Default number formats are all 8.2 (eight characters with two of them after the decimal).
The following SAS formats are available:

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEST(w)</td>
<td>Best available format with width=(w)</td>
</tr>
<tr>
<td>COMMA(w.d)</td>
<td>Comma and decimal points with width=(w) and decimal=(d)</td>
</tr>
<tr>
<td>COMMAX(w.d)</td>
<td>Comma and decimal points with width=(w) and decimal=(d) (switches the role of comma and decimal points)</td>
</tr>
<tr>
<td>DOLLAR(w.d)</td>
<td>Dollar signs, comma, and decimal points with width=(w) and decimal=(d)</td>
</tr>
<tr>
<td>DOLLARX(w.d)</td>
<td>Dollar signs, comma, and decimal points with width=(w) and decimal=(d) (switches the role of comma and decimal points)</td>
</tr>
<tr>
<td>EURO(w.d)</td>
<td>Euro signs, comma, and decimal points with width=(w) and decimal=(d)</td>
</tr>
<tr>
<td>EUROX(w.d)</td>
<td>Euro signs, comma, and decimal points with width=(w) and decimal=(d) (switches the role of comma and decimal points)</td>
</tr>
<tr>
<td>PERCENT(w.d)</td>
<td>Percentage with width=(w) and decimal=(d) and a percentage sign</td>
</tr>
<tr>
<td>(z.w.d)</td>
<td>Prefixed with zero to get width=(w) and decimal=(d)</td>
</tr>
<tr>
<td>(w.d)</td>
<td>width=(w) and decimal=(d)</td>
</tr>
</tbody>
</table>
CHAPTER

8

Define Rules

Introduction

An assignment rule associates rows in a transaction table with a behavior. An assignment rule:

- is associated with one or more rows in a behavior table
- specifies which rows in a transaction table group to include in a calculation
- specifies what quantity to include in the calculation.

An assignment rule is implicitly associated with a single transaction table in a table group — namely, the transaction table whose period is specified in the behavior table row with which the rule is associated.

Define the First Rule

1. Click the Rules workspace.
2. Select ABMCost as the table group to use the rule.
3. Click Add assignment rule.
Define the First Rule  Chapter 8

The Add Rule window opens.

4 Name the rule **ATM_CHK_Check balance**.

5 Click **Define**.

The Selection Criteria window opens.

6 Select **AssignmentRule** from the drop-down list of column names to select.

7 Select “=” from the drop-down list of operators.

8 Click **Select**, and select **ATM_CHK_Check balance** from the dialog box of possible text strings.

The dialog box shows all the possible text strings in the AssignmentRule column of the ABMCost table group. (Remember that you selected ABMCost as the table group to which this rule applies.)
Define Rules

Define the First Rule

After making your selections, the Selection Criteria window should look like the following:

9 Click OK.

You return to the Add Assignment Rule window.

10 Select **AMT** from the drop-down list of values for the driver formula.

The formula uses the value of the AMT column to calculate a value for every row picked out by the selection criterion:

\[
\text{AssignmentRule} = \text{ATM_CHK\_Check balance}
\]
Define the First Rule  Chapter 8

AMT represents the number of times a transaction (such as checking one's balance) occurred.

Count holds a value of 1 for every transaction. If you need to know, for example, how many customers used a specific product, you can filter by product and use the Count property to calculate.

11 Click OK.

The rule is added to the list of rules for the table group ABMCost.
How Rules Work

An assignment rule associates rows in a transaction table with a behavior. An assignment rule:

- is associated with one or more rows in a behavior table
- includes selection criteria that specify which rows in a transaction table group to include in a calculation
- includes a driver formula that specifies what quantity to include in the calculation.

An assignment rule is implicitly associated with a single transaction table in a table group — namely, the transaction table whose period is specified in the behavior table row with which the rule is associated.

Define Another Rule

1 Make sure you are in the Rules workspace.
2 Make sure that ABMCost is selected as the table group to use the rule.
3 Click Add assignment rule.

The Add Rule window opens.
Define Another Rule  Chapter 8

4 Name the rule **ATM_OTP_Manage Transactions**.

5 Click **Define**.

The Selection Criteria window opens.

6 Select the dimension **Channel** from the drop-down list of column names to select.

7 Select **is child of** from the drop-down list of operators.

   The “child of” operator is hierarchically driven. By selecting a member of a dimension at a point in the hierarchy, you are actually including that specific member and all of its children in the dimensional hierarchy.

8 Click **Select**, and select [ATM] from the dialog box of possible values.

After making your selections, the Selection Criteria window should look like the following:

9 Click **Add New Row**.

10 Select the dimension **Product** as the column to filter on.

11 Select **is child of** from the drop-down list of operators.

12 Select **Other Products** from the dialog of possible values.
13 Click OK.

You return to the Add Assignment Rule window.

14 Select Count from the drop-down list of values for the driver formula.

The formula uses the value of the Count column in calculating a value for every row picked out by the selection criteria.

15 Click OK.

The rule is added to the list of rules for the table group ABMCost.
Import the Remaining Rules

Instead of defining rules one by one, you can put the rule definitions in a file and import the file into SAS Profitability Management.

1. Click **Import assignment rules**.

   The Import Rules window opens.

2. Click **Select** and select **RULES** as the table to import.

3. Identify fields in the rules table.

   A rules import table must contain a field for each of the following: Name, Selection Criteria, Driver Formula, and Table Group Name.
Define Rules

By default, SAS Profitability Management assumes that these fields are named as follows: “AssignmentRule”, “SelectionCriteria”, “DriverFormula”, and “TableGroup”. If fields with those names exist in the rules table, then they are automatically mapped. If the fields are named differently, then you must match them up manually.

The sample rules table, RULES, contains these fields, so they are mapped automatically.

4 Click **OK**.

The rules are imported.

**Note:** It is considered an error if a rule already exists. It is not overwritten.
### Import Table for Rules

The following is a portion of the import table – RULES. Notice that it contains rules operating upon all three table groups in the Baby Bank model – ABMCost, CallCenter, and Revenue. It is not necessary to create a separate table of rules for each table group.

<table>
<thead>
<tr>
<th>TableGroup</th>
<th>AssignmentRule</th>
<th>SelectionCriteria</th>
<th>DriverFormula</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABMCost</td>
<td>ATM_CKN_Requests</td>
<td>IsChild(Children, TM_CHANNEL) AND IsChild(Product, TM_PRODUCT)</td>
<td>Count * 0.05 * AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_CKN_Deposits</td>
<td>AssignmentRule = ATM_CKN_Deposits</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_CKN_Fund Transfer</td>
<td>AssignmentRule = ATM_CKN_Fund Transfer</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_CKN_Withdrawals</td>
<td>AssignmentRule = ATM_CKN_Withdrawals</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_HCM_Requests</td>
<td>IsChild(Children, TM_CHANNEL) AND IsChild(Product, TM_PRODUCT)</td>
<td>Count * 0.05 * AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_HCM_Deposits</td>
<td>AssignmentRule = ATM_HCM_Deposits</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_HCM_Fund Transfer</td>
<td>AssignmentRule = ATM_HCM_Fund Transfer</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_HCM_Withdrawals</td>
<td>AssignmentRule = ATM_HCM_Withdrawals</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_HCM_Requests</td>
<td>IsChild(Children, TM_CHANNEL) AND IsChild(Product, TM_PRODUCT)</td>
<td>Count * 0.05 * AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_HCM_Deposits</td>
<td>AssignmentRule = ATM_HCM_Deposits</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_HCM_Fund Transfer</td>
<td>AssignmentRule = ATM_HCM_Fund Transfer</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_HCM_Withdrawals</td>
<td>AssignmentRule = ATM_HCM_Withdrawals</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_TCM_Requests</td>
<td>IsChild(Children, TM_CHANNEL) AND IsChild(Product, TM_PRODUCT)</td>
<td>Count * 0.05 * AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_TCM_Deposits</td>
<td>AssignmentRule = ATM_TCM_Deposits</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_TCM_Fund Transfer</td>
<td>AssignmentRule = ATM_TCM_Fund Transfer</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_TCM_Withdrawals</td>
<td>AssignmentRule = ATM_TCM_Withdrawals</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_TCM_Requests</td>
<td>IsChild(Children, TM_CHANNEL) AND IsChild(Product, TM_PRODUCT)</td>
<td>Count * 0.05 * AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_TCM_Deposits</td>
<td>AssignmentRule = ATM_TCM_Deposits</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_TCM_Fund Transfer</td>
<td>AssignmentRule = ATM_TCM_Fund Transfer</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_TCM_Withdrawals</td>
<td>AssignmentRule = ATM_TCM_Withdrawals</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_TCM_Requests</td>
<td>IsChild(Children, TM_CHANNEL) AND IsChild(Product, TM_PRODUCT)</td>
<td>Count * 0.05 * AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_TCM_Deposits</td>
<td>AssignmentRule = ATM_TCM_Deposits</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_TCM_Fund Transfer</td>
<td>AssignmentRule = ATM_TCM_Fund Transfer</td>
<td>AMT</td>
</tr>
<tr>
<td>ABMCost</td>
<td>ATM_TCM_Withdrawals</td>
<td>AssignmentRule = ATM_TCM_Withdrawals</td>
<td>AMT</td>
</tr>
</tbody>
</table>

### Revenue

<table>
<thead>
<tr>
<th>Revenue</th>
<th>ID</th>
<th>DriverFormula</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM Fee</td>
<td>ID = 1002</td>
<td>AMT</td>
</tr>
<tr>
<td>Certificate of Deposit Pay</td>
<td>ID = 11002</td>
<td>AMT</td>
</tr>
<tr>
<td>Charge for Funds</td>
<td>ID = 11002</td>
<td>AMT</td>
</tr>
<tr>
<td>Checking Account Fees</td>
<td>ID = 12004</td>
<td>AMT</td>
</tr>
<tr>
<td>Credit Card Fees</td>
<td>ID = 10001</td>
<td>AMT</td>
</tr>
<tr>
<td>Credit Card Interest Income</td>
<td>ID = 15001</td>
<td>AMT</td>
</tr>
<tr>
<td>Check for Funds</td>
<td>ID = 13001</td>
<td>AMT</td>
</tr>
<tr>
<td>Investment Account Fees</td>
<td>ID = 12003</td>
<td>AMT</td>
</tr>
<tr>
<td>Investment Bank Fees</td>
<td>ID = 11003</td>
<td>AMT</td>
</tr>
<tr>
<td>Investment Securities Fees</td>
<td>ID = 11003</td>
<td>AMT</td>
</tr>
<tr>
<td>Loan Interest Income</td>
<td>ID = 10002</td>
<td>AMT</td>
</tr>
<tr>
<td>Mortgage Income</td>
<td>ID = 10003</td>
<td>AMT</td>
</tr>
<tr>
<td>Revenue</td>
<td>ID = 14001</td>
<td>AMT</td>
</tr>
<tr>
<td>Savings Interest Payments</td>
<td>ID = 11001</td>
<td>AMT</td>
</tr>
</tbody>
</table>
Each behavior must be associated with only one rule. Multiple behaviors can be associated with the same rule.

1. Click the **Behaviors** workspace.
2. Select the behavior **20003 ATM_CHK_Check balance**.
3. Click **Create a new association**.

The Associate Rules window opens.

4. Select the **ABMCost** table group – the table group to which the rule is applied.
5. Make sure that the behavior **ABM_CHK_Check balance** is selected.
6. Select the rule **ABM_CHK_Check balance**.
7. Click **Associate**.
8. Click **OK**.
The rule is associated with the behavior, as shown in the behaviors workspace.

**Import the Remaining Associations**

1. Click **Import associations**.
   
The Import Associations window opens.

2. Click **Select**.
   
The Select Table dialog opens.

3. Select the **ASSOCIATIONS** table to import.
4 Click OK.

5 Identify fields in the ASSOCIATIONS table.

A rules association table must contain a field for each of the following: Behavior Id, Table Group Name, and Rule Name.

By default, SAS Profitability Management assumes that these fields are named as follows: “BehaviorId”, “TableGroup”, and “AssignmentRule”. If fields with those names exist in the associations table, then they are automatically mapped. If the fields are named differently, then you must match them up manually.

The sample table, ASSOCIATIONS, contains these fields, so they are mapped automatically.

6 Click OK.

The behaviors are associated with rules.

**Note:** If a behavior is already associated with a rule, an error message is issued and the association is not replaced.
Import the Remaining Associations  Chapter 9
Define a Summary Report

A summary report in SAS Profitability Management represents an OLAP cube. In defining a summary report, you define the attributes of a cube.

1. Select the Reports tab.
2. Click New summary report definition.

The New Summary Report Definition window opens.

3. Name the report, for example Baby Bank, and click Next.
Select the dimensions and dimension levels to be included in the report.

- Select **L3_Quarter** for the DIM_PERIOD dimension.
- Select **L5_Profit** for the REPORTHIERARCHY dimension.

For each dimension to be included in the summary cube, specify the level of drill-down allowed.

The default drill-down depth is always the top of the dimension. In this case the top of the time dimension is 2006. If no additional drill-down was selected, the summary report would hold totaled values for 2006, with no additional detail available to drill down.

You can select to include any level of depth defined in your dimensional hierarchy to include in your report.

All decisions on the number of dimensions and the depth of drill-down in any selected dimension have a dramatic impact on the reporting performance. It is wise to limit summary reports to specific areas where business decisions will be made.
It is also wise to produce lots of individual summary reports meeting specific managers’ needs focused upon specific dimensions rather than creating an all-encompassing summary report that has all dimensions and all drill-down levels.

5 Click **Next**.

6 Select **VALUE** as a measure to be included in the report.

The VALUE property is the default. It is the calculated cost for a transaction.

All numeric properties are available for inclusion in a summary report. The following are available:

- Source table transactions numeric properties, by source table group:
  - ABMCost (AMT, Count)
  - CallCenter (communication, complaints, count, crossSell, inquiry, offer, requests)
  - Revenue (AMT)

- Calculated results
  - Quantity – the result of the rules driver formula noted on each row of the transaction tables based on behaviors.
  - Value – the calculated result for the behavior source (revenue and costs) applied to the transaction details based on the rules defined. The method of its calculation depends upon whether the behavior table row that is accessed by an assignment rule contains a unit value or a total value, as we shall see in greater length in “Calculation – a Conceptual View” on page 76.

7 Click **Finish**.

The report definition is added.
Define a Detail Report

Detail reports produce a report based on a single dimension and by filtering a single value in that dimension. Detail reports are run from the SAS Profitability Management Web Client and create a cube on the fly.

1. Click the **Detail Reports** tab.
2. Click **New detail report definition**.

The New Detail Report Definition window opens.

3. Select **CustID** as the column to be featured in the detail report, and then click **Next**.
When defining a detail report, you should choose the dimension that will most uniquely filter the result. A detail report results in a cube created on the fly. Selection logic is processed to filter the starting transactional content to the single customer selected.

The goal for quick reporting response time is to try to filter your result such that the selected answer has fewer than 1,000 records. So, while there are other dimensions in the model, it would be unwise to select any dimension other than customer, because the filtered results would be more than 1,000 records.

4 Click Next.

5 Select the dimensions and dimension levels to be included in the report.
   - Select L3_Quarter for the DIM_PERIOD dimension.
   - Select L5_Profit for the REPORTHIERARCHY dimension.
6 Click Next.

7 Select VALUE as a measure to be included in the report.

The VALUE property is the default. It is the calculated cost for a transaction.

All of the numeric properties are available to be included in a report.

8 Click Finish.

The system processes all of the calculated transaction tables and adds an index for the dimension you have selected for detailed reporting. The SAS log notes that a simple index has been created. A dialog box displays the results.
Prepare Reports

Define a Detail Report

And, the report definition is added to the list of available detail reports. The list of detail reports is the same list that will be available in the Web Report client.
Calculate the Model

1. Select Actions > Calculate Model.
   The Calculate window opens.

2. Select both periods to be calculated for the model, and then click Next.

   ![Calculate Window]
   Each transaction table represents a single period. When you are calculating a model, you do not need to recalculate all of the transaction tables. This is ideal for calculations for incremental periods. You can calculate January, for example, and then calculate February as a separate calculation. You never need to process a single period's transactions more than once as the months proceed through the year.

   The summary reports and detail reports reflect transaction tables for time periods that have been calculated and that have been selected for cube inclusion in the analysis settings (see “Select Periods for the Cube” on page 47). So the generated cubes contain the complete details for calculated transaction tables across periods.

3. Select all three table groups to be calculated for the model, and then click Next.
4 Select the cubes (summary reports) to be generated and then click Next.

The window shows Baby Bank, which is the name you assigned to your summary report.

5 Select the number of concurrent sessions, and then click Finish.

SAS Profitability Management can start concurrent sessions to distribute the workload across multiple CPUs. One transaction is allocated to each session.
When the calculation and cube generation completes, you can view the SAS log.

6 Select **File > Exit** to close the application.

You are now finished working with the SAS Profitability Management rich client.

7 Verify the calculation process by reviewing the files created on the server.

Calculation result tables are stored in the directory where you defined the target destination LIBNAME. If you remember, we assumed that you created a directory on the server for this purpose named

**C:\SAS\ProfitabilityManagement\Bank_Out**.

Log on to the server and view the output files. In addition to creating the calculated transaction tables, the generation process creates **babybank.sas7bview**. The analysis view name is used as the name of the
database view that is created to join the transaction output tables into a single virtual fact table that the OLAP cube is built from.

**Calculation – a Conceptual View**

The following graphic shows a sample transaction table before and after a calculation. The calculation has added three new columns to the transaction table in the output directory:

**Driver Quantity**

is the number of units calculated by the driver formula. The calculation is based on whether the behavior-table row that is accessed by an assignment rule contains a unit value or a total value.

**Calculated Value**

is the calculated cost for a transaction. The method of its calculation depends upon whether the behavior table row that is accessed by an assignment rule contains a unit value or a total value.

**Behavior**

is the ID of the row from the Behavior table that is used in the calculation for a transaction.
When a behavior table row contains a unit value, the driver formula calculates the number of units that are involved in each transaction that is picked out by the selection criteria. The number of units appears in the DriverQty field. Then, the cost for that transaction (Value) is determined by multiplying the number of units (driver quantity) by the unit cost (in the behavior table) of the transaction.

The following graphic shows a conceptual example of a calculation using a unit value from the behavior table.
1. Select the periods and table groups to calculate.

2. In the behavior table, find all the behaviors for this period.

3. Using the behavior ID (Bx in this example), find the rule associated with that ID in the rule association table. Behavior Bx is associated with rule R2.

4. The criteria for rule R2 select all rows in the transaction table where CustID = 3. The formula specifies that DriverQuantity = AMT + 1

5. Because the behavior contains a UnitValue, the Value for the two rows is calculated as follows:
   - UnitValue ($2) times DriverQuantity (by the formula AMT+1 = 3) equals Value ($6)
   - UnitValue ($2) times DriverQuantity (by the formula AMT+1 = 4) equals Value ($8)
Calculation Using a Total Value

When a behavior table row contains a total value, the driver formula is used to calculate the number of units involved in each transaction that is picked out by the selection criteria. Then, the cost for that transaction is determined in the following way:

1. The total number of units for all transactions (picked out by the selection criteria) is calculated by adding the number of units (as determined by the driver formula) for all the transactions picked out by the selection criteria.

2. The cost per unit is calculated by dividing the total value (in the behavior table row) by the total number of units.

3. The cost for each transaction (Value) is calculated by multiplying the cost per unit times the number of units (as determined by the driver formula) for that transaction.

The following graphic shows a conceptual example of a calculation using a total value from the behavior table.
1. Select the periods and table groups to calculate.

2. In the behavior table, find all the behaviors for this period.

3. Using the behavior ID (Bx in this example), find the rule associated with that ID in the rule association table. Behavior Bx is associated with rule R2.

4. The criteria for rule R2 select all rows in the transaction table where Product=CHK. The formula specifies that DriverQuantity = Count / Amt.
Output Tables Grow in Length

Something important to understand about calculation is that the number of rows in the calculated transaction table will generally increase. It will increase when multiple rules access the same rows in the input transaction table. In that case, an additional row is created in the output table each time a new rule accesses the same row in the input table. For example, the following graphic shows the second rule (ATM_CHK_Requests) picking out the same three rows in the input table as the first rule (ATM_CHK_Check balance). The second rule, therefore, adds three more rows to the output table – in addition to the three rows added by the first rule.
View the Summary Report 83
Open the Baby Bank Report 83
See More Columns of Data 85
See More Rows of Data 86
Customize the View 88
Filter Data 89
Change the Appearance 92
Format Numbers 93
Add a Bar Chart to the Report 96
Customize the Bar Chart 97
Save the Report 100
View the Detail Report 101

View the Summary Report

After using the SAS Profitability Management rich client to calculate a model and generate reports, you use the SAS Profitability Management Web Report client to view the reports. Anyone with a browser and access to the Web can view the reports – provided, of course, that they have permission.

Open the Baby Bank Report

1 Log onto the SAS Web Report client.

Access it with the URL: http://<<your web server>>:8080/SASProfitability/.

The user name and password are what you defined earlier using the SAS Management Console – the same user name you have used up to now in accessing the SAS Profitability Management rich client.
The SAS Web Report client opens at the **Summary Report** tab.

2. Select the **Baby Bank** model and click the **Baby Bank** cube.

The Baby Bank summary report opens in the SAS Web OLAP viewer.
The profit and loss statement is displayed in table format. By default, the display shows:

- **Rows**: determined by the combination of the report layout table and the report hierarchy table.
- **Columns**: the time period dimension.
- **Slicer**: the Value property as calculated by the application of rules to the transaction tables.

### See More Columns of Data

See more detail in the time dimension by expanding the columns.

1. Click the plus sign next to **Actual** to expand the column.

   ![Click to Expand]

<table>
<thead>
<tr>
<th>L1 Scenario</th>
<th>Actual</th>
<th>Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Click the plus sign next to **2006** to expand the year.

   ![Click to Expand]

<table>
<thead>
<tr>
<th>L1 Scenario</th>
<th>Actual</th>
<th>Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2 Year</td>
<td>2006</td>
<td></td>
</tr>
</tbody>
</table>

You should see a column for actual and a column for budget, as shown in the following graphic.
Note: You can click the plus sign to expand, or click the down-arrow to drill down.

Drilling down replaces the current header with the next lower-level header.
Expanding shows the next lower-level header, and leaves the current header in place.

After expanding or drilling down, you can navigate in the opposite direction (collapsing or drilling up) by clicking in the appropriate place.

See More Rows of Data

1 Click the arrow at the bottom of the table to scroll to the last row of the table.
2. Click the plus sign next to **Sales and Marketing Effort** to expand the row.

Click the plus sign next to **Cost to Retain** to expand it.

3. Continue clicking the plus sign to fully expand that branch of the **Sales and Marketing Effort**. Your display should look like the following:
The levels available to drill to additional detail in the cube depend upon the number of dimension levels in the dimension hierarchy and the number of dimension levels that you select in the definition of the summary cube. This is explained further in “Report Hierarchy” and “Report Layout” starting on page 121.

**Customize the View**

You can remove blank content from the report with the Preferences dialog box.

1. Click **Preferences** (near the top of the window).

The Preferences dialog box opens.

2. Select **two columns** for the layout.

3. Check **Suppress empty data**.
4 Click **OK**.

**Filter Data**

You can select specific values to display using a filter.

1 Click the **Filter** icon on the tool bar.

2 Select **REPORTHIERARCHY** to filter the report by its rows, and then click **Edit** to pick the rows.

3 Select the rows as shown in the following graphic (you will have to scroll to select them all).
After you click **OK**, the filter expression shows that you have set a filter on the **REPORTHIERARCHYSet**.

4. Click **OK**, and the report is filtered to show only the rows you have selected. Notice that the top of the report shows what filter is applied.
5 Clear the filter by doing the following:
   a. Click the Filter icon.
   b. Select REPORTHIERARCHYSet.
   c. Click Clear.
   d. Click OK.
Change the Appearance

1. Click Properties.

The Properties dialog box opens.

2. Select the following properties, and then click OK.
   - Select All rows to display on the page.
   - Select All columns to display on the page.
   - Select a Column Heading Color.
   - Select Show color gradients in row and column headings to indicate depth of drilling with colors.

The appearance of the report changes accordingly.
The default format for the numbers can be directly defined in the SAS Profitability Management rich client. You have already edited analysis settings to select the format DOLLAR15.2. If you want to create an additional custom format for your numbers in the SAS Web OLAP viewer (maybe to display values in thousands), you need to create a new calculated measure and apply a custom format to that measure.

1. Click the **Calculate New Measure** icon on the tool bar.

2. Name the new measure **Thou**.

3. Assign the following calculation expression for the measure: 
   \[ \langle \langle \text{root.VALUE} \rangle \rangle / 1000 \].

4. Click **Select Format**.
5 Select **Numeric** as the format type.
6 Select **DOLLAR**.

7 Click **OK**.

The format is added to the Calculate New Measure dialog box.
8 Click **Add**, and then click **OK**.

The Thou measure is automatically added to the columns of numbers displayed, as you can see in the following graphic.

If you want only the Thou column to display, then do the following:

9 Select **VALUE** from the Selected items panel.

10 Click **Remove**.

11 Click **Apply**.
Add a Bar Chart to the Report

You can add bar charts and other graphs to your report. To illustrate this, let’s first start over with a fresh view of the summary report.

1. Click Profitability Management at the top of the window to start over.

The list of available summary reports appears.
2 Click Baby Bank.

The Baby Bank cube opens.

3 Click the Bar Chart icon on the tool bar.

A bar chart is added to the bottom of your report.

Customize the Bar Chart

1 Click Properties above the bar chart.
The Properties window opens.

2 On the **Graph** tab, select the following:
   - Select **Ocean** for style.
   - Select **3D** for depth.
   - Select **Vertical** for grid lines.

3 On the **Bar** tab, select the following:
   - Select **Horizontal** for orientation.
4 On the **Axis** tab, select the following:

- Select **Tahoma** for labels.
- Select **Tahoma** for values.

5 On the **Legend** tab, select the following:

- Select **Top** for position.
- Select **Shadow** for border.
Click **OK**.

The bar chart is redrawn to your specifications.

---

**Save the Report**

When you have a report layout that you believe is useful, save it.

1. Select **File > Save As**.
2 Name the view, for example **Baby Bank Bar Chart**, and click **OK**.

The view is saved as a data exploration and can be opened again and shared with others for review. It is saved under your user profile in the BIP tree.

**Foundation\BIP Tree\Users\<<your user ID>>\Data Explorations**

**Note**: If the cube for your summary report is regenerated in the SAS Profitability Management rich client, then your saved data exploration views must be refreshed to display the new content.

---

**View the Detail Report**

1 Click **Profitability Management** at the top of the window to return to the SAS Profitability Management Web workspace.

2 Click the **Detail Reports** tab.
3. Select the **CustID** detail report that you defined previously (for the Baby Bank model).

4. Type **00018** in the **Search For** field to report on that customer, and then click **View**.

The report for customer **00018**, Lynn York, is displayed.

<table>
<thead>
<tr>
<th>L1 Scenario</th>
<th>L2 Actual</th>
<th>L3 Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>L1_Profit</strong></td>
<td><strong>L2_Profit</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Interest Income</strong></td>
<td><strong>Credit Card Interest Income</strong></td>
<td>$546.19</td>
</tr>
<tr>
<td></td>
<td><strong>Loan Interest Income</strong></td>
<td>$1,552.13</td>
</tr>
<tr>
<td></td>
<td><strong>Mortgages Income</strong></td>
<td>.</td>
</tr>
<tr>
<td><strong>Total Interest Income</strong></td>
<td></td>
<td>$2,098.32</td>
</tr>
<tr>
<td><strong>Interest Expense</strong></td>
<td></td>
<td>.</td>
</tr>
<tr>
<td></td>
<td><strong>Savings Interest Payments</strong></td>
<td>.</td>
</tr>
<tr>
<td></td>
<td><strong>Certificates of Deposit Payments</strong></td>
<td>.</td>
</tr>
<tr>
<td></td>
<td><strong>Investment Securities Payments</strong></td>
<td>.</td>
</tr>
<tr>
<td><strong>Total Interest Expense</strong></td>
<td></td>
<td>.</td>
</tr>
<tr>
<td><strong>Net Interest Income</strong></td>
<td></td>
<td>$2,098.32</td>
</tr>
</tbody>
</table>

On review of Lynn York's profit and loss statement, you can see the details of her relationship with Baby Bank.

- You can notice that she has a credit card and a loan and has been paying a significant amount of credit card interest.
- You can notice she has also been paying a lot of unsecured loans interest income, but she does not have a mortgage.
- She might be a prime target for a marketing communication for a second mortgage to consolidate her existing debts.
You can also notice that she does not have any interest expense, which indicates that her savings, CDs, and investments are not with Baby Bank. So she may be a prime target for marketing those products.

When you scroll to the next page and see the bottom line, you notice that Lynn York is a very profitable customer for Baby Bank.

<table>
<thead>
<tr>
<th>Net Interest Income</th>
<th>$2,098.32</th>
<th>$1,977.11</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Funds</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Credit for Funds</td>
<td>$394.78</td>
<td>$371.97</td>
</tr>
<tr>
<td>Charge for Funds</td>
<td>$493.48</td>
<td>$464.96</td>
</tr>
<tr>
<td><strong>Net Funds</strong></td>
<td>$-98.70</td>
<td>$-92.99</td>
</tr>
<tr>
<td><strong>Fees</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Credit Card Fees</td>
<td>$218.63</td>
<td>$206.00</td>
</tr>
<tr>
<td>ATM Fees</td>
<td>$138.24</td>
<td>$130.26</td>
</tr>
<tr>
<td>Investment Account Fees</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>Checking Account Fees</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td><strong>Total Non Interest Income</strong></td>
<td>$356.87</td>
<td>$336.26</td>
</tr>
<tr>
<td>Provision For Losses</td>
<td>$98.69</td>
<td>$92.99</td>
</tr>
<tr>
<td><strong>Direct Product</strong></td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td><strong>Relationship Management</strong></td>
<td>$97.30</td>
<td>$102.06</td>
</tr>
<tr>
<td><strong>Sales and Marketing Effort</strong></td>
<td>$10.46</td>
<td>$9.15</td>
</tr>
<tr>
<td><strong>Servicing Effort</strong></td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td><strong>Total Non Interest Expense</strong></td>
<td>$206.45</td>
<td>$204.20</td>
</tr>
<tr>
<td><strong>Net Contribution</strong></td>
<td>$2,150.04</td>
<td>$2,016.18</td>
</tr>
</tbody>
</table>
Summary of Model Elements

Data Requirements for the Model 105
Behavior Table 106
Period Dimension 108
Custom Dimensions 109
  DIM_CHANNEL 109
  DIM_CUSTOMER 110
  DIM_CUSTYPE 111
  DIM_PRODUCT 112
  DIM_REGION 113
Rule Definition Table 114
  Rule Selection Criteria 115
    Text Values 115
      Numerical Properties 116
      Dimensional Member Values 116
      Matching Fields in the Behavior Table 118
    Rule Driver Formula 119
      Single Numeric Property 119
      Multiple Numeric Properties 119
      Single Entered Value 119
      SAS Numeric Expression 119
  Rule/Behavior Associations 119
  Report Hierarchy 121
    Report Hierarchy to Resulting Report 122
    Behavior Table to Report Hierarchy 124
  Report Layout 124
    Report Layout to Resulting Report 125
    Summary: Behavior to Hierarchy to Layout to Report 127
Transaction Tables 127
  ABMCost Group 128
  CallCenter Group 130
  Revenue Group 130

Data Requirements for the Model

The following graphic shows the development sequence in SAS Profitability Management from the input of various data tables to the output of OLAP cubes. A detailed description of each of the input tables follows in this chapter.
For all data tables, data validation needs to occur before the content is loaded into SAS Profitability Management. Review source tables to eliminate special characters. The following list contains the reserved characters to remove. These characters are not valid in cubes, where profit and loss reporting will be done: . , ; ' ` : ? * & % $ ! - + = ( ) [ ] { } / .

Behavior Table

The behavior source table holds all of the content to provide the cost and the revenue values that will be assigned to the transactional volumes based upon the defined rules.

The behavior table contains the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Maximum Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Char 32</td>
<td>The period for the transaction cost</td>
</tr>
<tr>
<td>ID</td>
<td>Char 32</td>
<td>The ID for the behavior</td>
</tr>
<tr>
<td>Name</td>
<td>Char 32</td>
<td>The name of the behavior</td>
</tr>
<tr>
<td>TotalValue</td>
<td>Num 8</td>
<td>The total source amount to be divided</td>
</tr>
<tr>
<td>UnitValue</td>
<td>Num 8</td>
<td>The unit cost for each transaction with this source</td>
</tr>
</tbody>
</table>

Note:
- Only one of the UnitValue and TotalValue columns can contain a non-zero value for any row of the behavior table.
- The order of the columns is arbitrary (you identify the columns during model definition).
- The name of the columns is arbitrary (but, if you use these names, they are mapped automatically).

The following graphic shows the behavior table, BEHAVIORS, for the Baby Bank model:

<table>
<thead>
<tr>
<th>Time</th>
<th>ID</th>
<th>Name</th>
<th>AssignmentRule</th>
<th>TotalValue</th>
<th>UnitValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006_Q4</td>
<td>Actual</td>
<td>ATM_chk_Check balance</td>
<td>ATM CHK Check balance</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2006_Q4</td>
<td>Actual</td>
<td>ATM_CHK_Deposits</td>
<td>ATM_CHK_Deposits</td>
<td>0.0</td>
<td>2.1885018295073905E 4</td>
</tr>
<tr>
<td>2006_Q4</td>
<td>Actual</td>
<td>ATM_CHK_Fund Transfer</td>
<td>ATM CHK Fund Transfer</td>
<td>0.0</td>
<td>5.328984320898614E 4</td>
</tr>
<tr>
<td>2006_Q4</td>
<td>Actual</td>
<td>ATMCHK_Deposits</td>
<td>ATM CHK Withdrawals</td>
<td>0.0</td>
<td>0.0037864785318</td>
</tr>
<tr>
<td>2006_Q4</td>
<td>Actual</td>
<td>ATM_CHK_Deposits</td>
<td>ATM CHK Withdrawals</td>
<td>0.0</td>
<td>0.0037864785318</td>
</tr>
<tr>
<td>2006_Q4</td>
<td>Actual</td>
<td>ATM CHK_Deposits</td>
<td>ATM_CHK_Deposits</td>
<td>0.0</td>
<td>0.0037864785318</td>
</tr>
<tr>
<td>2006_Q4</td>
<td>Actual</td>
<td>ATM CHK_Deposits</td>
<td>ATM CHK_Deposits</td>
<td>0.0</td>
<td>0.0037864785318</td>
</tr>
<tr>
<td>2006_Q4</td>
<td>Actual</td>
<td>ATM CHK_Deposits</td>
<td>ATM CHK_Deposits</td>
<td>0.0</td>
<td>0.0037864785318</td>
</tr>
<tr>
<td>2006_Q4</td>
<td>Actual</td>
<td>ATM CHK_Deposits</td>
<td>ATM CHK_Deposits</td>
<td>0.0</td>
<td>0.0037864785318</td>
</tr>
<tr>
<td>2006_Q4</td>
<td>Actual</td>
<td>ATM CHK_Deposits</td>
<td>ATM CHK_Deposits</td>
<td>0.0</td>
<td>0.0037864785318</td>
</tr>
</tbody>
</table>

The following graphic shows how a behavior table is related to other files:
The period dimension table defines the time periods used in a model. Depending on your business reporting needs, you may have more or fewer levels in the period hierarchy.

A period dimensions table can contain the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Maximum Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Char 32</td>
<td>The identifying value for the period</td>
</tr>
<tr>
<td>L1_Scenario</td>
<td>Char 32</td>
<td>Top-level dimension member value (for example, Actual or Budget)</td>
</tr>
<tr>
<td>L2_Year</td>
<td>Char 32</td>
<td>Second-level dimension member value (for example, 2008)</td>
</tr>
<tr>
<td>L3_Quarter</td>
<td>Char 32</td>
<td>Third-level dimension member value (for example, 2008_Q4)</td>
</tr>
</tbody>
</table>

**Note:**
- The order of the columns is significant.
- The name of the columns is arbitrary.
- The number of levels is arbitrary.

The following graphic shows the period dimension table, DIM_PERIOD, for the Baby Bank model:
Custom Dimensions

A definition table must be loaded for each dimension used in the SAS Profitability Management model and in your transactional tables. Transaction tables have a column for each custom dimension defining the crossing where transactional values are collected. All dimension names must be a single word.

The Baby Bank model contains the following custom dimensions:
- Channel
- Customer
- CusType
- Product
- Region.

**DIM_CHANNEL**

The following table shows the channel dimension. It is a one-level table that defines the following channels:

<table>
<thead>
<tr>
<th>L1_Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM</td>
</tr>
<tr>
<td>BRH</td>
</tr>
<tr>
<td>Call Center</td>
</tr>
</tbody>
</table>

The following graphic shows the entire table contents:
DIM_CUSTOMER

The following table shows a portion of the contents of the DIM_CUSTOMER dimension. You can see that the table defines two types of customers: personal and business. For personal customers, the table contains both last name and first name. For business customers, the last name contains the name of the business, and the FirstName field is blank.

<table>
<thead>
<tr>
<th>L1_Type</th>
<th>L2_LastName</th>
<th>L3_FirstName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal</td>
<td>Adcock</td>
<td>James</td>
</tr>
<tr>
<td></td>
<td>Farm</td>
<td>Peter T.</td>
</tr>
<tr>
<td></td>
<td>Aden</td>
<td>Shelly</td>
</tr>
<tr>
<td>Business</td>
<td>Logic Consulting</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Metrogas SA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FASHBI Worldwide</td>
<td></td>
</tr>
</tbody>
</table>

The following graphic shows part of the contents of the DIM_CUSTOMER table:
**DIM_CUSTYPE**

The following table represents the contents of DIM_CUSTYPE. It shows the levels in the dimension.

<table>
<thead>
<tr>
<th>L1_CType</th>
<th>L2_CType</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal</td>
<td>Retail Consumer Banking</td>
</tr>
<tr>
<td></td>
<td>Small Business Banking</td>
</tr>
<tr>
<td>Business</td>
<td>Private Banking</td>
</tr>
<tr>
<td></td>
<td>Corporate Banking</td>
</tr>
</tbody>
</table>

The following graphic shows the entire contents of DIM_CUSTYPE.
## DIM_PRODUCT

The follow table represents the contents of DIM_PRODUCT.

<table>
<thead>
<tr>
<th>L1_Product</th>
<th>L2_Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Products</td>
<td>Secured/Mortgages</td>
</tr>
<tr>
<td></td>
<td>Unsecured</td>
</tr>
<tr>
<td>Deposit Products</td>
<td>Checking</td>
</tr>
<tr>
<td></td>
<td>Recurring</td>
</tr>
<tr>
<td></td>
<td>Savings</td>
</tr>
<tr>
<td></td>
<td>Term</td>
</tr>
<tr>
<td>Fee-Based Products</td>
<td></td>
</tr>
<tr>
<td>Other Products</td>
<td></td>
</tr>
<tr>
<td>Revolving Credit Products</td>
<td>Credit Cards</td>
</tr>
<tr>
<td></td>
<td>Overdrafts</td>
</tr>
<tr>
<td>Third-Party Products</td>
<td></td>
</tr>
</tbody>
</table>

The following graphic shows the entire contents of DIM_PRODUCT.

<table>
<thead>
<tr>
<th>ID</th>
<th>L1_Product</th>
<th>L2_Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRP</td>
<td>Credit Products</td>
<td>Secured/Mortgages</td>
</tr>
<tr>
<td>DEP</td>
<td>Deposit Products</td>
<td>Unsecured</td>
</tr>
<tr>
<td>FBP</td>
<td>Fee Based Products</td>
<td></td>
</tr>
<tr>
<td>OTP</td>
<td>Other Products</td>
<td></td>
</tr>
<tr>
<td>RCP</td>
<td>Revolving Credit Products</td>
<td>Credit Cards</td>
</tr>
<tr>
<td>TPP</td>
<td>Third Party Products</td>
<td>Overdrafts</td>
</tr>
<tr>
<td>SCR</td>
<td>Credit Products</td>
<td></td>
</tr>
<tr>
<td>UCR</td>
<td>Credit Products</td>
<td>Un secured</td>
</tr>
<tr>
<td>CHK</td>
<td>Deposit Products</td>
<td>Checking</td>
</tr>
<tr>
<td>REC</td>
<td>Deposit Products</td>
<td>Recurring</td>
</tr>
<tr>
<td>SAV</td>
<td>Deposit Products</td>
<td>Savings</td>
</tr>
<tr>
<td>TRM</td>
<td>Deposit Products</td>
<td>Term</td>
</tr>
<tr>
<td>CRC</td>
<td>Revolving Credit Products</td>
<td>Credit Cards</td>
</tr>
<tr>
<td>OVD</td>
<td>Revolving Credit Products</td>
<td>Overdrafts</td>
</tr>
</tbody>
</table>
### DIM_REGION

The following table represents only a portion of the DIM_REGION dimension. It is included here to help you understand the level structure. The region dimension has four levels.

<table>
<thead>
<tr>
<th>Area</th>
<th>Country</th>
<th>State</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>Canada</td>
<td>Ontario</td>
<td>Don Mills</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Metcalfe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quebec</td>
<td>St. Hubert Quebec</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>St. Laurent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Town of Mont Royal</td>
</tr>
<tr>
<td></td>
<td>Saskatchewan</td>
<td></td>
<td>Sheffield</td>
</tr>
<tr>
<td>Mexico</td>
<td></td>
<td>Coahuila</td>
<td>Saltillo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mexico</td>
<td>Cuauhtemoc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Colonia Granada</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Polanco</td>
</tr>
<tr>
<td>South America</td>
<td>Brazil</td>
<td>Brazil</td>
<td>Uberlandia - Mg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pe Brazil</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sao Paulo</td>
<td>Ribeirao Preto</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sao Paulo</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Costa Rica</td>
<td>Costa Rica</td>
<td>San Jose</td>
</tr>
<tr>
<td>Ecuador</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uruguay</td>
<td>Montevideo</td>
<td>Montevideo</td>
<td></td>
</tr>
<tr>
<td>Venezuela</td>
<td>Caracas</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following graphic shows the corresponding part of the actual table contents.
A rule provides

- Selection criteria that determine which rows are selected in the transaction tables for calculation of values (revenue or cost).
- Driver formula that provides the calculation necessary to assign an amount for the source values (cost or revenue) from the behavior table to selected records in the transaction tables.

Rules can be defined through an interactive method in the SAS Profitability Management client user interface, or rules can be imported from a specific source table, using the appropriate rules definitional syntax to provide the selection logic and the driver quantities.

The rule definition table contains the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Maximum Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TableGroup</td>
<td>Char 64</td>
<td>The name of the transaction table group to which the assignment rule will be applied</td>
</tr>
<tr>
<td>AssignmentRule</td>
<td>Char 64</td>
<td>The name of the assignment rule; this must be unique within the model</td>
</tr>
<tr>
<td>SelectionCriteria</td>
<td>Char 1024</td>
<td>The expression defining the filter logic</td>
</tr>
<tr>
<td>DriverFormula</td>
<td>Char 1024</td>
<td>The formula defining the calculated quantity</td>
</tr>
</tbody>
</table>

**Note:**

- The order of the columns is arbitrary (you identify them during import).
- The name of the columns is arbitrary (but, if you use these names, they are automatically mapped during import).

The following graphic shows the rule import table, RULES, for the Baby Bank model:
Rule Selection Criteria

Rule selection criteria filter the transaction table rows that are calculated. You can filter by:

- text values
- numerical properties
- dimensional member values
- matching fields in the behavior table

Text Values

<table>
<thead>
<tr>
<th>Operators</th>
<th>Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>=, not =</td>
<td>text property = 'filter value'</td>
<td>AssignmentRule = 'ATM_CHK_Check balance'</td>
</tr>
</tbody>
</table>
Numerical Properties

| Operators: | =, not =, <, <=, >, >=
| And, Or |
| Syntax: | column operator value |
| Example: | AMT > 1000 |

Dimensional Member Values

| Operators: | IsChildOf |
| And, Or |
| Syntax: | IsChildOf (source column, [dimension table].[value]) |
| Example: | IsChildOf (channel, '[dim_channel].[ATM]') |
| Example: | IsChildOf (channel, '[dim_channel].[ATM]') And IsChildOf (product, '[dim_product].[deposit products].[checking]') |

The **IsChildOf** operator allows you to select from any point in a dimensional hierarchy tree, and all children of that hierarchy are selected as matching. In the following graphic, these selection criteria pick out Secured Mortgages (SCR) and Unsecured (UCR) because they are both children of Credit Products (and are in the ATM channel):

IsChildOf (Product, '[DIM_PRODUCT].[Credit Products]') AND
IsChildOf (Channel, '[DIM_CHANNEL].[ATM]')

**Note:** This rule does not exist in the Baby Bank tutorial model. It is shown here for the purpose of illustration.
Matching Fields in the Behavior Table

| Operators: | MatchColumns  
| And        |
| Syntax:    | MatchColumns(Transaction(column_name), Behavior(column_name))  
| Example:   | MatchColumns(Transaction(Product), Behavior(Product))  

The **match Behavior** operator enables you to compare fields between the behavior table and a transaction table. The selection criteria are met when the value of every matching column in the transaction table row equals the value of the corresponding matching column in the behavior table row.

For example, “Product” in a transaction table can have a corresponding matching column “Product” in the behavior table. Note that the “Product” field in the behavior table is not a required field. You can add additional fields to the behavior table for matching purposes.

Both text and numeric column types are supported for matching fields. You can combine multiple match behaviors with the AND operator.
**Rule Driver Formula**

The result of a driver formula is the driver quantity. The driver formula can be based on:
- a single numeric property
- multiple numeric properties
- a single entered value
- a SAS numeric expression.

**Single Numeric Property**

Example: AMT

**Multiple Numeric Properties**

Example: COUNT * .05 + AMT * .01

**Single Entered Value**

Example: 15

**SAS Numeric Expression**

The driver formula can consist of any SAS numeric expression. Columns referenced in the formula must be in the input transaction table.

Example: MAX(Requests, Complaints)*.25
Example: EUROCURR(AMT, 'eur', 'frf')
Example: IF CustID = 'C00650' THEN (0.54*Distance)
ELSE IF CustID = 'C00692' THEN (0.74*Distance)
ELSE (0.23*Distance)

**Rule/Behavior Associations**

The association table is used to correlate rules with behaviors. Each rule can be used by multiple behaviors, but each behavior can only have one rule. The association process can be done interactively in the user interface or the associations can be defined based upon an import table to load.

The rule associations table contains the following fields:
### Rule/Behavior Associations

#### Chapter 13

<table>
<thead>
<tr>
<th>Name</th>
<th>Maximum Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BehaviorId</td>
<td>Char 32</td>
<td>The ID of the behavior table row</td>
</tr>
<tr>
<td>BehaviorName</td>
<td>Char 55</td>
<td>Optional field. The name of the behavior</td>
</tr>
<tr>
<td>TableGroup</td>
<td>Char 64</td>
<td>The table group to which the rule is applied</td>
</tr>
<tr>
<td>AssignmentRule</td>
<td>Char 64</td>
<td>The name of the associated assignment rule. This name must be unique within the model</td>
</tr>
</tbody>
</table>

**Note:**
- The order of the columns is arbitrary (you identify them during import).
- The name of the columns is arbitrary (but, if you use these names, they are automatically mapped during import).

The following graphic shows the rule associations table, ASSOCATIONS, for the Baby Bank model:
The report hierarchy table defines the dimension hierarchy for drilling down in profit and loss reporting. The number of levels in the hierarchy is a critical model design decision, and it is subject to the ultimate reporting needs for the business managers to make decisions.

The report hierarchy table contains the following fields:

<table>
<thead>
<tr>
<th>Position</th>
<th>Name</th>
<th>Maximum Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>Char 32</td>
<td>The ID of the behavior table row for this member</td>
</tr>
<tr>
<td>2</td>
<td>L1_Profit</td>
<td>Char 32</td>
<td>1st level dimension members for the hierarchy</td>
</tr>
<tr>
<td>3</td>
<td>L2_Profit</td>
<td>Char 32</td>
<td>2nd level dimension members for the hierarchy</td>
</tr>
<tr>
<td>4</td>
<td>L3_Profit</td>
<td>Char 32</td>
<td>3rd level dimension members for the hierarchy</td>
</tr>
<tr>
<td>5</td>
<td>L4_Profit</td>
<td>Char 32</td>
<td>4th level dimension members for the hierarchy</td>
</tr>
<tr>
<td>5</td>
<td>L5_Profit</td>
<td>Char 32</td>
<td>5th level dimension members for the hierarchy</td>
</tr>
</tbody>
</table>

**Note:**

- The order of the columns is significant.
- The name of the columns is arbitrary.
- The number of levels is arbitrary.

The following graphic shows the report hierarchy table for the Baby Bank model:
The report hierarchy table defines the dimension hierarchy for drilling down into the profit and loss report. The number of levels in the hierarchy is a critical decision in model design and is subject to the ultimate reporting needs of the business managers to make decisions.

### Report Hierarchy to Resulting Report

In the following graphic, you can see how the columns in the report hierarchy relate to the columns in the display table of a summary report.
Behavior Table to Report Hierarchy

Let's take a closer look at the report hierarchy – in particular the relationship between the report hierarchy and the behavior table. In the following graphic you can see that items in the behavior table occur at the lowest level in the report hierarchy table. Because the hierarchy can have different depths at different places, this means that the behavior table items can occur in different columns (different depths) in the report hierarchy table. Also, note that intermediate levels can correspond to custom dimensions (for example, the channel and product dimensions at levels 3 and 4) but they do not have to. For example, interest income and interest expense (in the first six rows of the report hierarchy table) have no existence outside the report hierarchy table itself.

Report Layout

The report layout defines the calculation logic for the profit and loss statement.
A report layout table contains the following fields:

<table>
<thead>
<tr>
<th>Position</th>
<th>Name</th>
<th>Maximum Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>Char 32</td>
<td>The ID of the behavior table row for this member</td>
</tr>
<tr>
<td>2</td>
<td>Name</td>
<td>Char 32</td>
<td>The name of the behavior for display on the report</td>
</tr>
<tr>
<td>3</td>
<td>Formula</td>
<td>Char 255</td>
<td>For calculated rows, the formula for the calculation based on the IDs</td>
</tr>
<tr>
<td>4</td>
<td>RowOrder</td>
<td>Numeric 8</td>
<td>The position of the row in the profit-and-loss report</td>
</tr>
</tbody>
</table>

**Note:**
- The column names must be exactly as shown.
- The order of the columns is arbitrary.

The following graphic shows the report layout table for the Baby Bank model:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Formula</th>
<th>RowOrder</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10001 Credit Card Interest Income</td>
<td>[10001]</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>10002 Loan Interest Income</td>
<td>[10002]</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>10003 Mortgages Income</td>
<td>[10003]</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>TOT INT INC Total Interest Income</td>
<td>[10001]+[10002]+[10003]</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>TOT INT EXP Total Interest Expense</td>
<td>[10001]+[10002]+[10003]</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>NET INT INC Net Interest Income</td>
<td>[TOT INT INC]-[TOT INT EXP]</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>NET FUNDS Net Funds</td>
<td>[13001]+[13002]</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>TOT NON INT INC Total Non-Interest Income</td>
<td>[12001]+[12002]+[12003]+[12004]</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>TOT NON INT EXP Total Non-Interest Expense</td>
<td>[12001]+[12002]+[12003]+[12004]</td>
<td>23</td>
</tr>
<tr>
<td>10</td>
<td>NET CONTRIB Net Contribution</td>
<td>[NET INT INC]+[NET FUNDS]-[TOT NON INT INC]-[TOT NON INT EXP]</td>
<td>24</td>
</tr>
</tbody>
</table>

**Report Layout to Resulting Report**

The report layout controls how the OLAP viewer displays the cube when it is first opened. You can see in the following graphic that the report layout table controls:

- The order in which columns are displayed. REPORTLAYOUT causes income variables display before expense variables.
- Whether fields are initially displayed expanded or collapsed. You can see in the following graphic that the **Servicing Effort** field is displayed collapsed because it
is included in REPORTLAYOUT without any of its children showing. By contrast, the **Interest Income** field is displayed expanded to level 2 because its children (credit card interest income, loan interest income, mortgage income) are included in REPORTLAYOUT but not the parent – Interest Income.

Note: The OLAP viewer always shows the highest level of a dimension. The report layout determines to what level the highest dimension is expanded when the report is initially opened. A user can navigate the dimension, expanding and collapsing it at will.

- **Totals.** For example, line 4 of the report layout displays a total whose formula is: \([10001]+[10002]+[10003]\) – that is line 1 + line 2 + line 3.
Summary: Behavior to Hierarchy to Layout to Report

The following graphic summarizes the relationships among the behavior table, report hierarchy, report layout, and resulting OLAP report.

Transaction Tables

A single SAS Profitability Management model can have multiple transaction tables, depending upon how the data is collected for your corporation. A table group is a set of
tables sharing the same schema (same columns with the same field definitions). In a table group, each table represents one period of the period table.

There must be a separate transaction table for each period in the model. For a single SAS Profitability Management model, multiple table groups will likely be defined. It is also likely that multiple rules will use the same source table group.

Transactional tables layout is affected by the rules definition process (filter logic and driver logic). There are three critical aspects to every transaction table: dimensional signature, filter criteria selection logic, and quantities for calculation.

A transaction table contains the following columns and must conform to the following rules:

- Each column must have the length shown.
- The position of columns is arbitrary.
- The name of columns is arbitrary.
- The number of columns in a transaction table is arbitrary.
- A transaction table can contain other columns not specifically used by SAS Profitability Management.

<table>
<thead>
<tr>
<th>Name</th>
<th>Maximum Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions</td>
<td>Char 32</td>
<td>Each value identifies the row (in a custom dimensions table) for this transaction. The number of dimensions is optional (minimum 1).</td>
</tr>
<tr>
<td>Driver quantities</td>
<td>Numeric 8</td>
<td>A rule's driver formula uses these values to calculate the driver quantity for a transaction. The number of columns is optional (minimum 1).</td>
</tr>
<tr>
<td>Optional columns</td>
<td>optional</td>
<td>You can add any number of optional text or numeric columns of any length to a transaction table. For example, you can use an optional column for matching fields in a behavior table.</td>
</tr>
</tbody>
</table>

**ABMCost Group**

The ABMCost table group consists of the following two transaction tables:

- LOAD_TRANS_Q4A – for the 4th quarter actual period
- LOAD_TRANS_Q4B – for the 4th quarter budget period.

The schema for the ABMCost group consists of the following fields:
Summary of Model Elements

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CustID</td>
<td>ID of customer dimension</td>
</tr>
<tr>
<td>2</td>
<td>Product</td>
<td>ID of product dimension</td>
</tr>
<tr>
<td>3</td>
<td>CustType</td>
<td>ID of customer type dimension</td>
</tr>
<tr>
<td>4</td>
<td>Region</td>
<td>ID of region dimension</td>
</tr>
<tr>
<td>5</td>
<td>Channel</td>
<td>ID of channel dimension</td>
</tr>
<tr>
<td>6</td>
<td>AssignmentRule</td>
<td>Text string for use in filtering rows with that string. Rows that are selected are assigned the behavior cost</td>
</tr>
<tr>
<td>7</td>
<td>Count</td>
<td>Value of 1 used in driver</td>
</tr>
<tr>
<td>8</td>
<td>AMT</td>
<td>Number of occurrences, used in driver</td>
</tr>
</tbody>
</table>

The following graphic shows a portion of the contents of LOAD_TRANS_Q4A:
CallCenter Group

The CallCenter table group consists of the following two transaction tables:

- CALLCENTER_Q4A – for the 4th quarter actual period
- CALLCENTER_Q4B – for the 4th quarter budget period.

The schema for the CallCenter group consists of the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Product</td>
<td>ID of product dimension</td>
</tr>
<tr>
<td>2</td>
<td>CustType</td>
<td>ID of customer-type dimension</td>
</tr>
<tr>
<td>3</td>
<td>Region</td>
<td>ID of region dimension</td>
</tr>
<tr>
<td>4</td>
<td>Channel</td>
<td>ID of channel dimension</td>
</tr>
<tr>
<td>5</td>
<td>Communication</td>
<td>number of calls</td>
</tr>
<tr>
<td>6</td>
<td>Complaints</td>
<td>Number of complaints</td>
</tr>
<tr>
<td>7</td>
<td>Inquiry</td>
<td>Number of inquiries</td>
</tr>
<tr>
<td>8</td>
<td>Requests</td>
<td>Number of requests</td>
</tr>
<tr>
<td>9</td>
<td>CrossSell</td>
<td>Number of sales calls</td>
</tr>
<tr>
<td>10</td>
<td>Offer</td>
<td>Number of offers</td>
</tr>
<tr>
<td>11</td>
<td>Count</td>
<td>A value of 1 indicates an evenly assigned item</td>
</tr>
</tbody>
</table>

The following graphic shows a portion of the contents of CALLCENTER_Q4A:

Revenue Group

The Revenue table group consists of the following two transaction tables:
REVENUE_Q4A – for the 4th quarter actual period
REVENUE_Q4B – for the 4th quarter budget period.

The schema for the Revenue group consists of the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Channel</td>
<td>ID of channel dimension</td>
</tr>
<tr>
<td>2</td>
<td>CustID</td>
<td>ID of customer dimension</td>
</tr>
<tr>
<td>3</td>
<td>Product</td>
<td>ID of product dimension</td>
</tr>
<tr>
<td>4</td>
<td>CustType</td>
<td>ID of customer type dimension</td>
</tr>
<tr>
<td>5</td>
<td>Region</td>
<td>ID of region dimension</td>
</tr>
<tr>
<td>6</td>
<td>AMT</td>
<td>Calculated specific revenue</td>
</tr>
<tr>
<td>7</td>
<td>ID</td>
<td>Behavior ID for revenue items</td>
</tr>
</tbody>
</table>

The following graphic shows a portion of the contents of REVENUE_Q4A:
Baby Bank Conclusions

You have successfully created a SAS Profitability Management model, calculated transaction tables, and generated cubes. You have reviewed profit and loss reports at both a summary and a detail level. You have reviewed those results to draw conclusions about your customers' behavior and the action Baby Bank can take to focus specific attention on customer needs and how they impact the corporate profitability of Baby Bank. Baby Bank can now analyze its customers, channels, products, customer types, and regions to plan for a successful and profitable future.

SAS Profitability Management enables organizations to use more accurate profitability measures to make better decisions for customers, products, and channels. With SAS Profitability Management, decisionmakers can define and redefine the segmentation reports that they need on the fly. SAS Profitability Management enables business managers to track the profit performance of customer groups or individual customers, product groups or individual SKUs, channels or specific branches – providing drill-down and at-a-glance views into revenue, cost, and other metrics so they can identify and investigate problems that can improve the bottom line.

Additional Features

Enhance your SAS Profitability Management Solution with

- SAS Activity-Based Management enables strategic and operational decisions that maximize profit, reduce costs, and streamline processes by determining the cost of those processes and the profitability of products, customers, and business segments. In SAS Activity-Based Management, you can mark accounts as behaviors and then publish the behaviors for use in SAS Profitability Management.
- SAS Customer Profitability for telecommunications is a component of SAS Telecommunications Intelligence Solutions, a Suite of integrated solutions that are built on an enterprise data architecture optimized for telecommunications providers.
- SAS Customer Intelligence for Banking can help you understand an individual customer’s behavior at every touch-point throughout the life cycle of the relationship. By integrating data across channels, product silos, and external data
and market sources, you create a holistic picture of the current, potential, and future value that each customer delivers, as opposed to fragmented facts on customer risk, behavior, account activities, and operational costs. Using predictive analytics, you can forecast customer behaviors such as attrition and credit and load risk so you can devise more effective cross-sell and up-sell strategies.

What to Do Next: Useful Links

- SAS Worldwide Web for links to everything SAS

- SAS Worldwide Training
  [http://support.sas.com/training/index.html](http://support.sas.com/training/index.html)

- BetterManagement – for useful business domain white papers and web casts

- SAS Solutions – Links to other powerful business solutions from SAS

  Focused Solutions for your Business Challenges: To lead with confidence and outpace competitors, you need to make accurate decisions faster than ever. SAS equips your organization for success by helping you answer more questions, for more people, across more departments than any other analytic applications suite provider.
  

- SAS Business Intelligence:

  SAS Business Intelligence gives you the information, when you need it, in the format you need. By integrating data from across your enterprise and delivering self-service reporting and analysis, IT spends less time responding to requests, and business users spend less time looking for information – so more time is spent on making better, more informed decisions
  

- SAS Analytics:

  SAS Analytics give you THE POWER TO KNOW® how to integrate data from across your enterprise and then quickly transform that data into shared insights. We offer a comprehensive suite of analytics software to help you reduce uncertainty, predict with precision, and optimize performance
  

- SAS Merchandise Intelligence

  Only SAS Merchandise Intelligence provides real intelligence at every step of the merchandising life cycle. With this collection of software and services, you can
maximize the profitability of the merchandising process while improving customer loyalty and satisfaction levels. Retailers get reporting, planning, forecasting, and optimization at critical points through the planning process, which leads to faster and better decisions.

http://www.sas.com/industry/retail/merchandise/index.html

SAS Customer Intelligence

Only SAS Customer Intelligence provides the vital knowledge needed to help organizations build an integrated platform for enterprise marketing management. With SAS Customer Intelligence, campaigns and programs implemented across channels will be effective, consistent, and timely. They will target the right customers with the right offers. And with the power of SAS predictive analytics, you can be confident that actual results will match predicted ones, even before you spend anything on a new campaign.

We welcome your feedback.

- If you have comments about this book, please send them to yourturn@sas.com. Include the full title and page numbers (if applicable).

- If you have comments about the software, please send them to suggest@sas.com.
Whether you are new to the workforce or an experienced professional, you need to distinguish yourself in this rapidly changing and competitive job market. SAS® Publishing provides you with a wide range of resources to help you set yourself apart.

**SAS® Press Series**
Need to learn the basics? Struggling with a programming problem? You’ll find the expert answers that you need in example-rich books from the SAS Press Series. Written by experienced SAS professionals from around the world, these books deliver real-world insights on a broad range of topics for all skill levels.

**SAS® Documentation**
To successfully implement applications using SAS software, companies in every industry and on every continent all turn to the one source for accurate, timely, and reliable information—SAS documentation. We currently produce the following types of reference documentation: online help that is built into the software, tutorials that are integrated into the product, reference documentation delivered in HTML and PDF—free on the Web, and hard-copy books.

**SAS® Learning Edition 4.1**
Get a workplace advantage, perform analytics in less time, and prepare for the SAS Base Programming exam and SAS Advanced Programming exam with SAS® Learning Edition 4.1. This inexpensive, intuitive personal learning version of SAS includes Base SAS® 9.1.3, SAS/STAT®, SAS/GRAF®H, SAS/QC®, SAS/ETS®, and SAS® Enterprise Guide® 4.1. Whether you are a professor, student, or business professional, this is a great way to learn SAS.