THE POWER TO KNOW.

Getting Started with SAS® Profitability Management 2.3

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc 2014. *Getting Started with SAS® Profitability Management 2.3*. Cary, NC: SAS Institute Inc.

Getting Started with SAS® Profitability Management 2.3

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

Printing 1, February 2014

Contents

Chapter 1

Introduction

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 familiarize you 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 (SKUs), 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 resources 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/. If there are updates or corrections to this tutorial, you can find them there.

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 needed.

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 critical 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 statements that SAS Profitability Management provides

- are based on a custom-defined report layout and can support complex calculation logic to present your company s reporting needs
- are drillable for increasing levels of detail (revenue breakdowns or contributing costs details)
- are drillable based on dimensional hierarchies
- include a summary cube report that can be
  - created to include only specific dimensions
  - summarized by depths noted in any dimension
- include a detail cube report that can be
  - defined with filter logic for a single dimension member
  - run on the fly.

Accessibility Features of SAS Profitability Management

This product has not been tested for compliance with U.S. Section 508 standards and W3C Web content accessibility guidelines. If you have specific questions about the accessibility of SAS products, send them to **accessibility@sas.com** or call SAS Technical Support.

Chapter 2

The Baby Bank Model

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 Profitabillty 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 each with a behavior
6. Calculate the model
7. Define a report hierarchy
8. Define a report layout
9. Prepare reports
10. View the reports

Chapter 3

Populate the Input Directory

Retrieve the Tutorial Data

The data sets for use with the tutorial are provided in a ZIP file. You can access the ZIP file from two places:

- installed with the product
- from support.sas.com

Installed with the product

By default the ZIP file, tutorial.zip, is installed in the following directory:

```
C:\Program Files\SASHome\SASProfitabilityManagementDataTier\2.21\Tutorial
```

From support.sas.com

1. Go to the SAS Profitability Management Product Documentation Page at **`http://support.sas.com/documentation/onlinedoc/pm/index.html`**.
2. Download and extract the tutorial data sets.

Data sets provided

Different data sets are provided for 32–bit and 64–bit systems. The following data sets are provided:

association.sas7bdat	Holds the relationships between the behaviors (where costs reside) and the rules (which define the assignment logic to transactions).
behavior.sas7bdat	Holds all of the transactions and costs (total or unit)
callcenter_q4a.sas7bdat	Call center transaction details for the 4th quarter actual

callcenter_q4b.sas7bdat	Call center transaction details for the 4th quarter budget
dim_channel.sas7bdat	Hierarchy definition for the channel dimension
dim__customer.sas7bdat	Hierarchy definition for the customer dimension
dim__custtype.sas7bdat	Hierarchy definition for the customer type dimension
dim__period.sas7bdat	Hierarchy definition for the time period dimension
dim__product.sas7bdat	Hierarchy definition for the product dimension
dim__region.sas7bdat	Hierarchy definition for the region dimension
load_trans_q4a.sas7bdat	ABMCost transaction details 4th quarter actual
load_trans_q4b.sas7bdat	ABMCost transaction details 4th quarter budget
reportHierarchy.sas7bdat	Report hierarchy defining the drill-down detail
reportLayout.sas7bdat	Report layout defining the profit and loss calculations
revenue_q4a.sas7bdat	Revenue transaction details 4th quarter actual
revenue_q4b.sas7bdat	Revenue transaction details 4th quarter budget
rules.sas7bdat	Definition of the assignment rules logic

For additional details describing the source tables required for SAS Profitability Management, please see Chapter 15, “Summary of Model Elements,” on page 121.

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.

   For purposes of illustration, we assume that you create and name it as follows:

   C:\SAS\ProfitabilityManagement\Bank_In

You can place the directory where you like and name it what you like. However, do not use the system directories under C:\SAS\Lev1\AppData \SASProfitabilityMgmt2.2\.

2. Give yourself (or whoever is doing the tutorial) read, modify, and write acess to the directory. If you have an access problem in doing the tutorial, that this 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.

   For purposes of illustration, we assume that you create and name it as follows:

   C:\SAS\ProfitabilityManagement\Bank_Out

   You can place the directory where you like and name it what you like. However, do not use the system directories under C:\SAS\Lev1\AppData \SASProfitabilityMgmt2.2\.

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

Chapter 4

Set Up the Environment

Overview

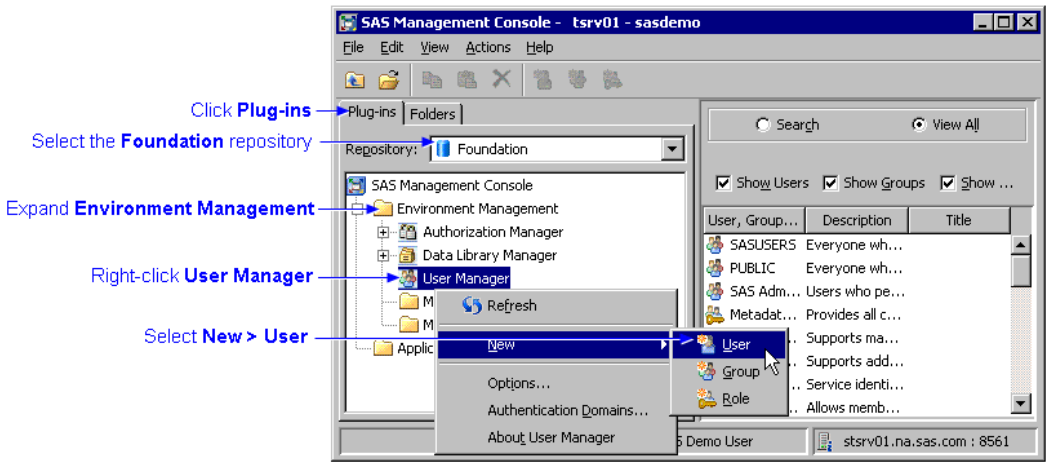
Setting up the SAS Profitability Management environment includes accomplishing the following tasks:

- Add users
- Identify input and output directories to SAS Profitability Management
- Register the tables in the input directory

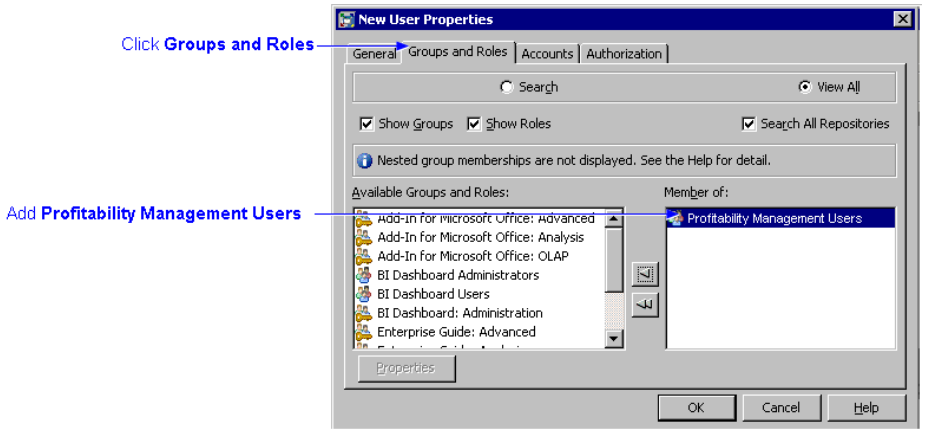
Add Users

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. Click the **Plug-ins** tab.
3. Select the **Foundation** repository.
4. Select **Environment Management**.
5. Right-click **User Manager**, and select **New** ⇨ **User**.

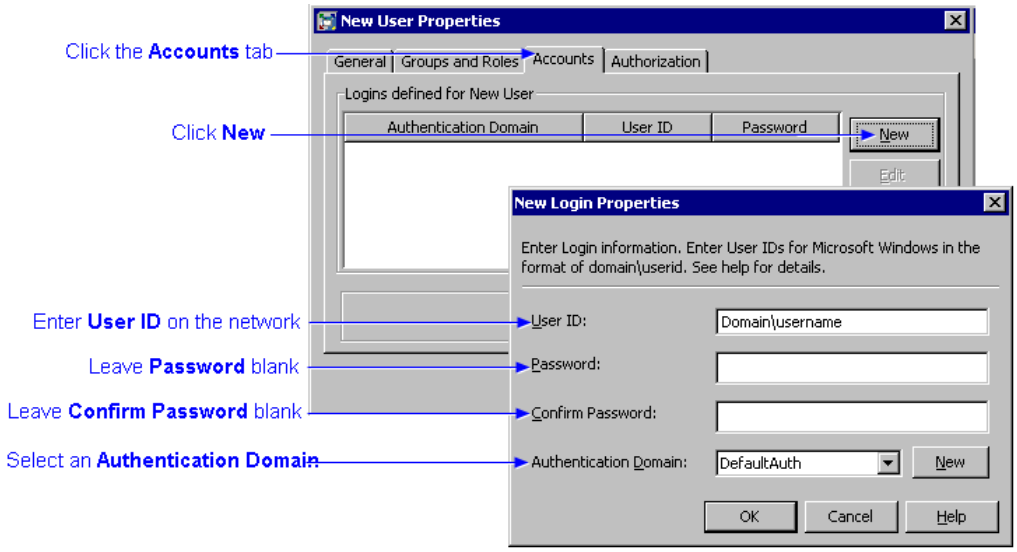


6. Name the new user, and enter other user information on the **General** tab.
7. Click the **Groups and Roles** tab, and add **Profitability Management Users** to the list of groups that the user is a member of.



8. Click the **Accounts** tab, and then click **New**.
   - Enter the new user's **User ID**, including domain, on the network.

     Note: For Windows, the name should be prefixed with the user's domain. For UNIX, no prefix is needed.
   - 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 an authentication domain.



9. You can leave the **Authorization** tab blank.
10. Click **OK**.

The user is created.

Identify Input and Output Directories to Profitability Management

Overview

Having created an input and output directory on the server, you must tell SAS Profitability Management about each one. To do so, you use SAS Management Console. For each directory (input and output) you must create both a library in SAS Management Console and a folder that is associated with each library.

Your input and output directories can contain either SAS data sets or database tables. If your output directory will contain database tables, then in addition to creating a library and its associated folder for the output directory, you must also create a Base SAS library and its associated folder to contain the analysis view. The analysis view, which is a SAS data set that is created during calculation, is a join of the transaction output tables into a single virtual fact table that the OLAP cube is built from. The following table summarizes what libraries and folders you must create depending on whether the files in your input and output directories are SAS data sets or database tables.

Files are SAS Data Sets	Files are Database Tables
Input Library:	**Input Library**:
a Base SAS library (and its associated folder) for an input directory that contains SAS data sets	a database library (and its associated folder) for an input directory that contains database tables

Files are SAS Data Sets	Files are Database Tables
Output Library: a Base SAS library (and its associated folder) for an output directory that contains SAS data sets	**Output Library**: a database library (and its associated folder) for an output directory that contains database tables
Analysis Library (optional): a Base SAS library (and its associated folder) for a directory that contains the analysis view, which is a SAS data set. A single library can serve as both the output library and the analysis library.	**Analysis Library**: a Base SAS library (and its associated folder) for an analysis directory that contains the analysis view, which is a SAS data set.

Using Base SAS Librairies

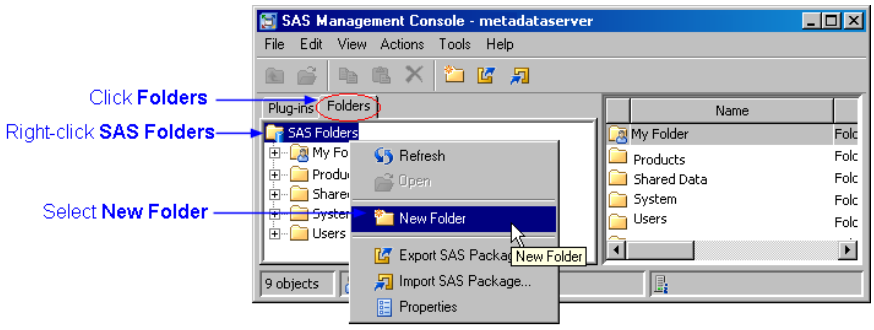
Creating Folders

The following instructions apply to using SAS data sets as your input and output files for SAS Profitability Management. In this case, your libraries are Base SAS libraries.

To understand folders, it might help to realize that they are independent of repositories. One SAS Metadata Server can have multiple repositories, but it has only one set of folders. While each library exists in a particular repository, its associated folder is independent of any repository.

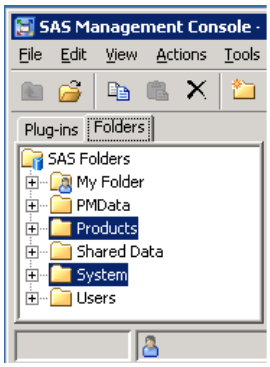
1. Log on to SAS Management Console as administrator, connecting to your SAS Metadata Server.
2. Click **Folders**.
3. Right-click **SAS Folders**, and select **New Folder**.

   The New Folder wizard opens. The folder you are creating contains metadata about the input (or output) library.



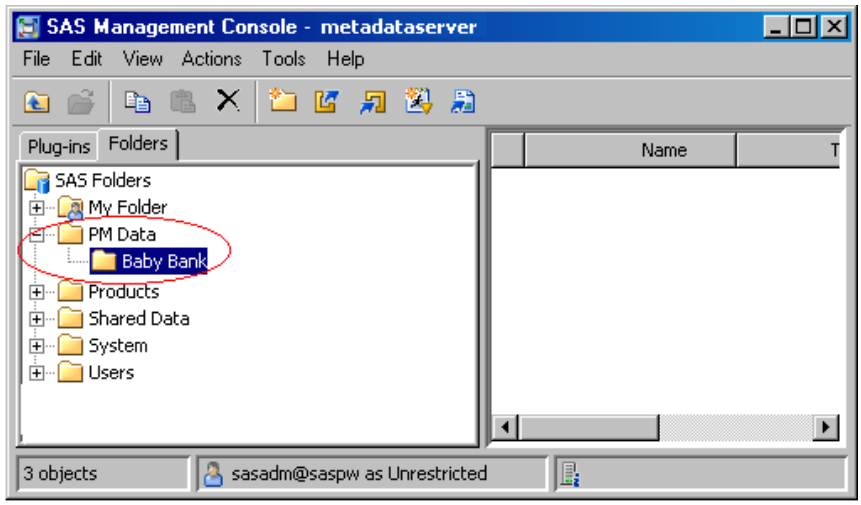
4. Name the folder. For this example, the folder is named **PM Data**.

   In this tutorial, the folder names and locations are only suggestions. You can name them as you like and locate them as you like. You should not, however, place folders for your input and output libraries under the already-existing **Products** folder or the already-existing **System** folder, as shown below.



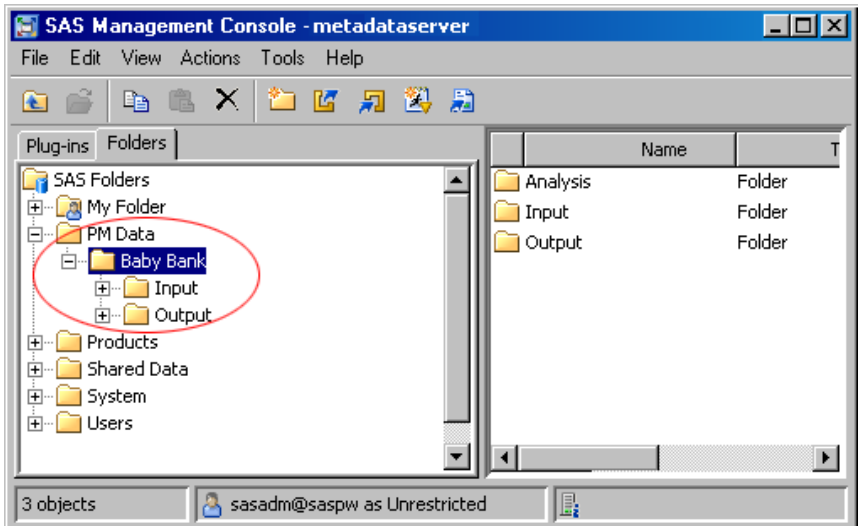
The important point is that you should have a separate folder for each model and you should create at least separate folders for input and output (a separate **Analysis** folder is optional).

5. Click **Finish**.
6. Create a folder named **Baby Bank** underneath **PM Data**. Your folders should now appear as follows:



7. Create the following folders underneath **Baby Bank**:
   - **Input**
   - **Output**

   Your folders should now appear as follows:

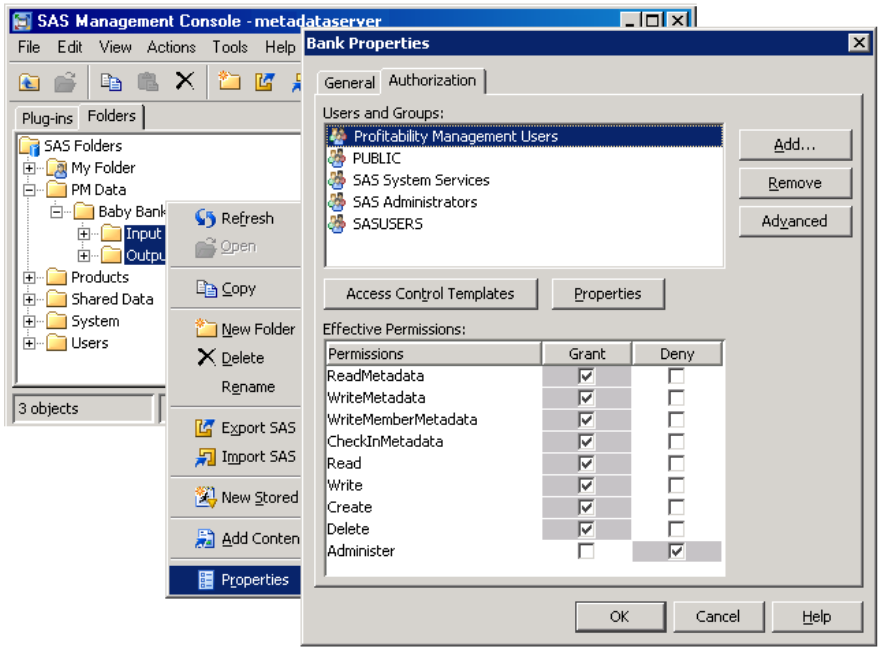


8. Grant permissions to the input and output libraries to the group, Profitability Management Users:

   a. Right-click each of the input and output folders in turn and select **Properties**.

   b. Click the **Authorization** tab.

   c. Grant all but Administer permission to the group, Profitability Management Users, as shown:

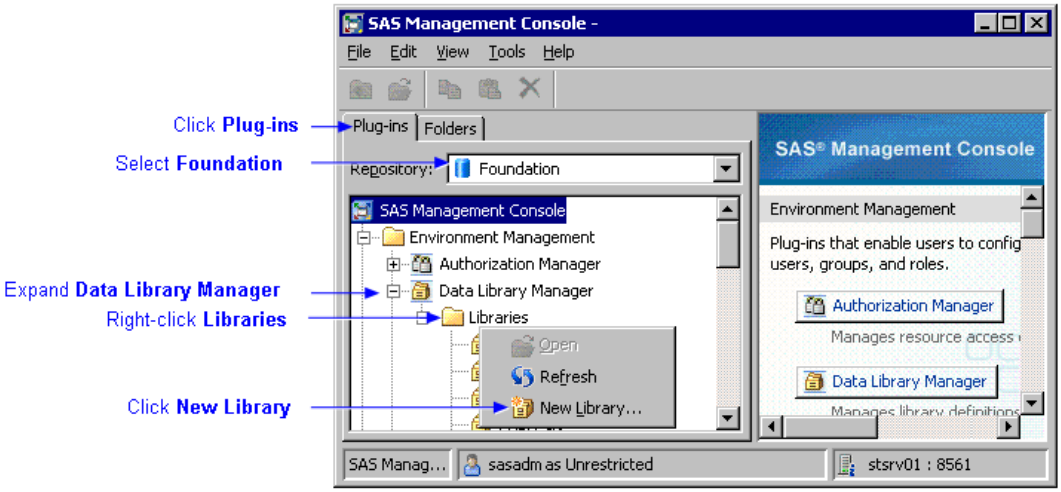


Creating the Input and Output Libraries

The following instructions apply to using SAS data sets as your input and output files for SAS Profitability Management. When you use SAS data sets, your libraries are Base SAS libraries.

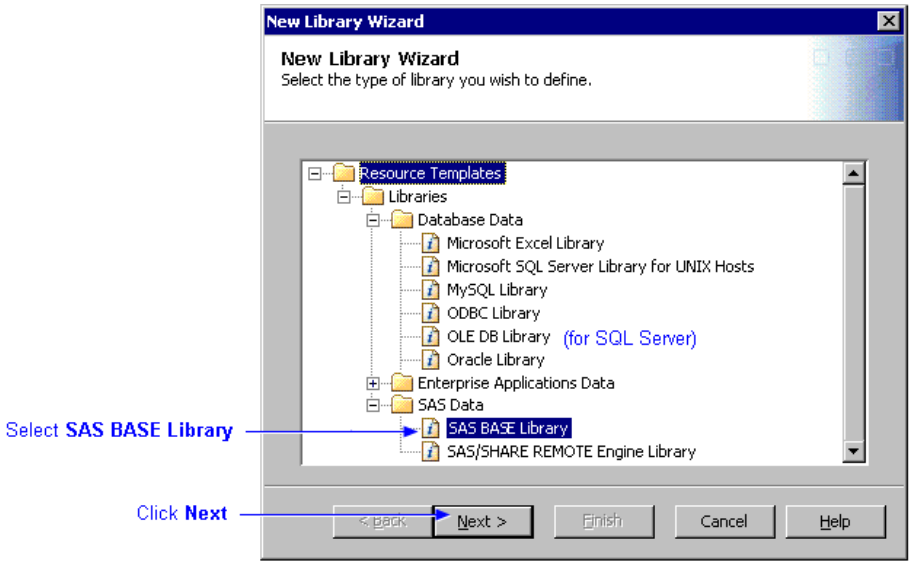
1. Click the **Plug-ins** tab.
2. Select the **Foundation** repository.
3. Expand **Data Library Manager**.

4. Right-click **Libraries**.
5. Click **New Library**.



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

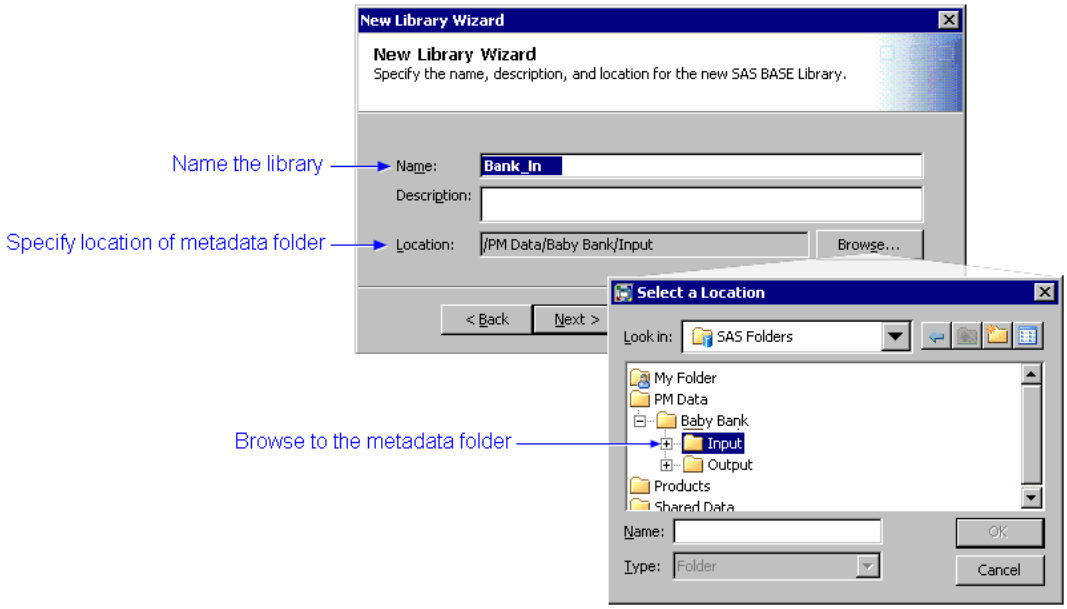
   Note: SAS Profitability Management supports any type of library for which you can create a libref. For this tutorial, however, we use a SAS Base engine library.



7. Name the library **Bank_In**.

   You can give the library any name you want (eight characters maximum). For convience, we give it the same name as the directory. The name is the text that appears in the navigation and display areas of SAS Management Console, but it is not the libref for the library. The name must be unique within the folder specified in the **Location** field. It must also be unique on all servers where the library is assigned.

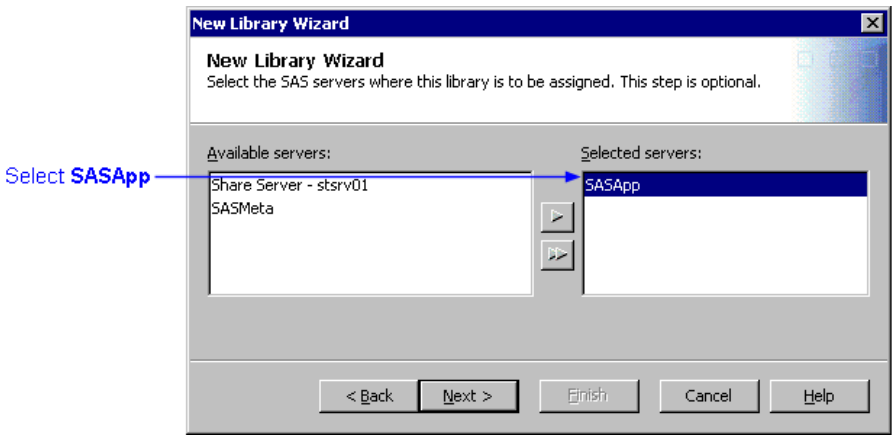
8. Specify the location of the library's metadata folder by clicking **Browse** and browsing to the **Input** folder that you just created.



Note: You can also create the folder from this window by clicking **New folder**.

9. Click **Next**.
10. Select **SASApp** as the SAS server that can access the library, and then click **Next**.

    Note: Although the window indicates that this step is optional, it is required for SAS Profitability Management. And, it is required that you choose **SASApp** as the workspace server.

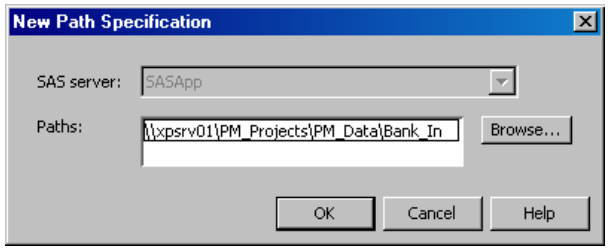


11. Specify **Bank_In** as the libref.

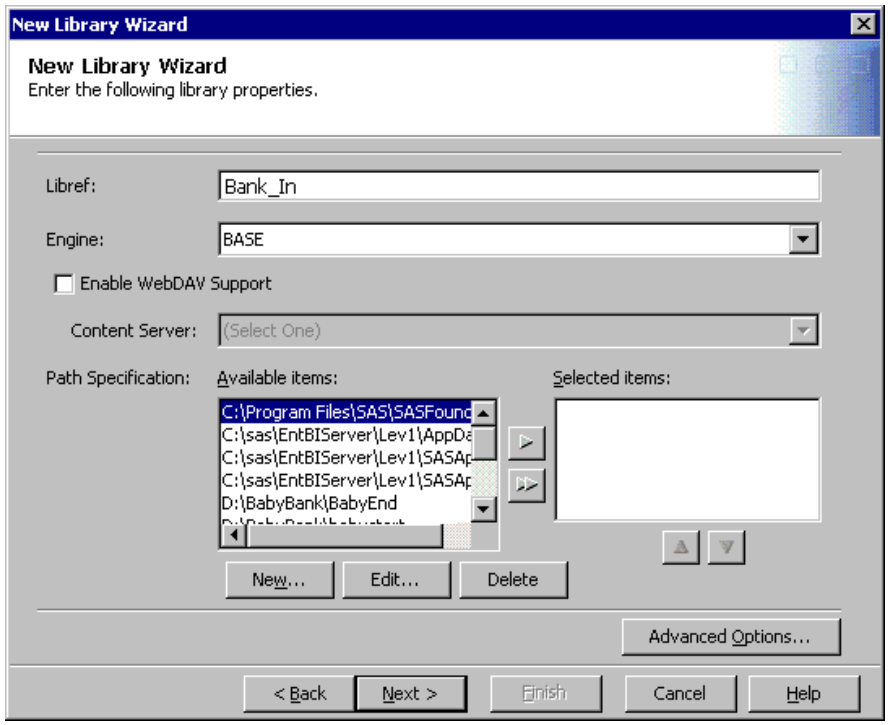
    The libref is used to reference a library without having to refer to its actual physical location.

12. Select the path of the library on the server. This is the actual physical location of the library.

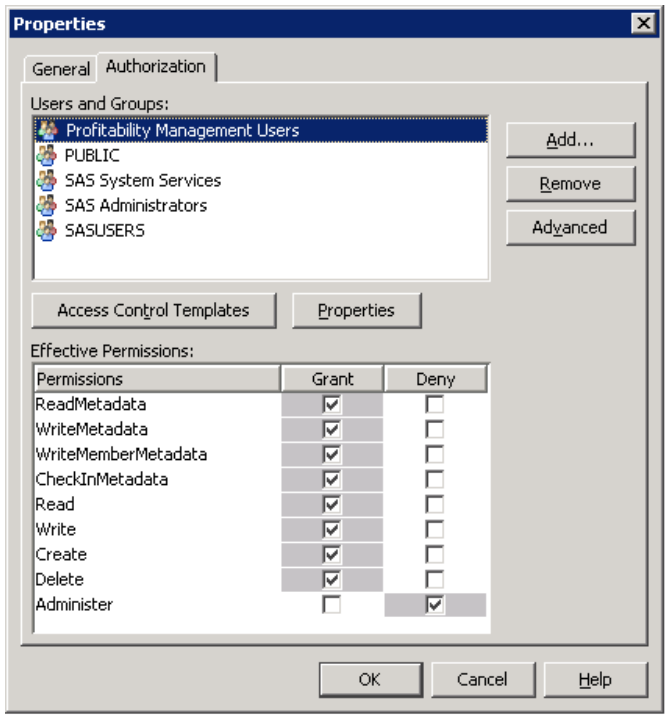
    Note: If the path does not appear in the list box of available paths, then click **New** to create the path. You can either browse to the path, if it is located on the metadata server machine, or you can simply type in the path, if it is on another server, as shown in the following:



For this tutorial we assume that you create the directory at C:\SAS \ProfitabilityManagement\Bank_In even though the directory can be anywhere on a server. However, do not use the system directories under C:\SAS\Lev1\AppData \SASProfitabilityMgmt2.2\, which are for system use.



13. Click **Next**.
14. Review the options you selected, and then click **Finish**.
15. Select **View** ⇨ **Refresh** from the menu bar to see the library listed.
16. Grant permissions to the input library to the group, Profitability Management Users:
    a. Right-click the library and select **Properties**.
    b. Click the **Authorzation** tab.
    c. Grant all but Administer permission to the group, Profitability Management Users, as shown in the following:



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.

Using Database Libraries

Overview

SAS Profitability Management supports any type of library for which you can create a libref, so your input and output directories can be connected to a database rather than containing SAS data sets. Although this tutorial assumes that you are using SAS data sets, the following instructions show you how to set up your environment to use database tables instead of SAS data sets for your input and output files. The following instructions assume that you use an ODBC connection to access your database. Consequently, setting up your database environment includes the following tasks:

- creating an ODBC connection
- creating a server
- creating a database library

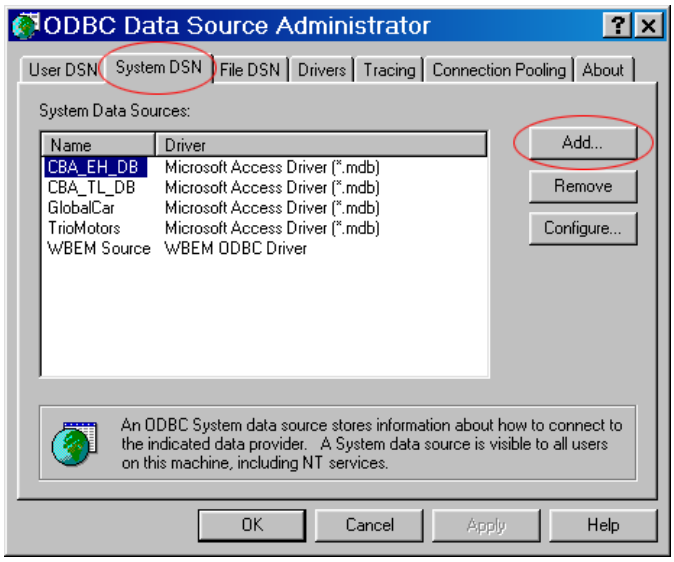
Creating an ODBC Connection

Do the following to create an ODBC connection:

1. Select **Start** ⇨ **Settings** ⇨ **Control Panel** ⇨ **Administrative Tools** ⇨ **Data Sources (ODBC)**.

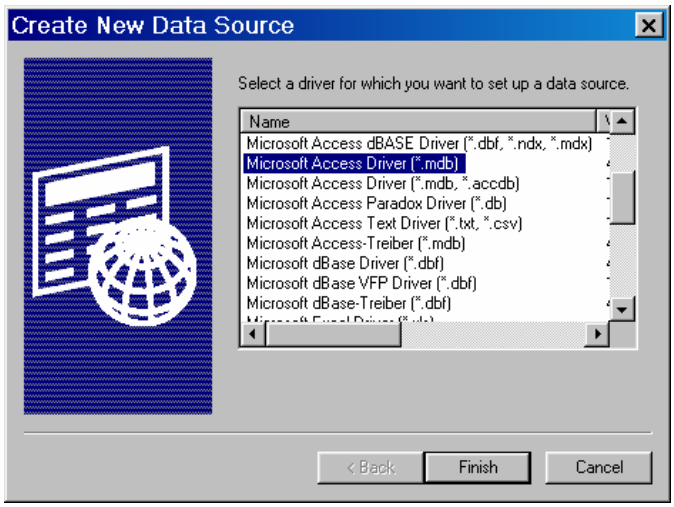
   The ODBC Data Source Administrator dialog opens.

2. Select the **System DSN** tab and click **Add**.



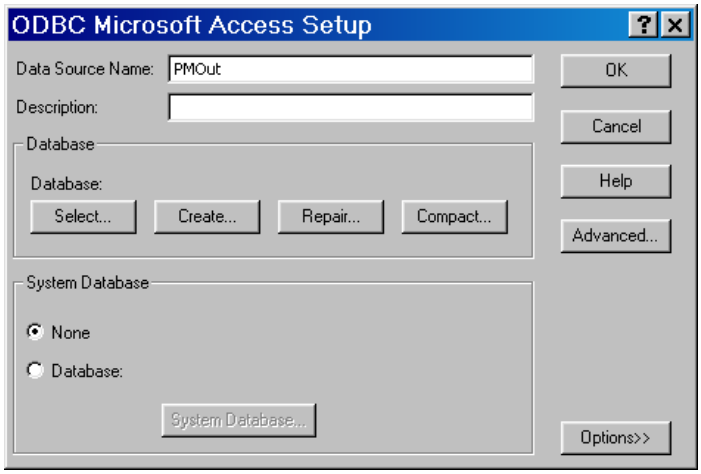
The Create New Data Source dialog opens.

3. Select **Microsoft Access Driver [*.mdb]** and click **Finish**.



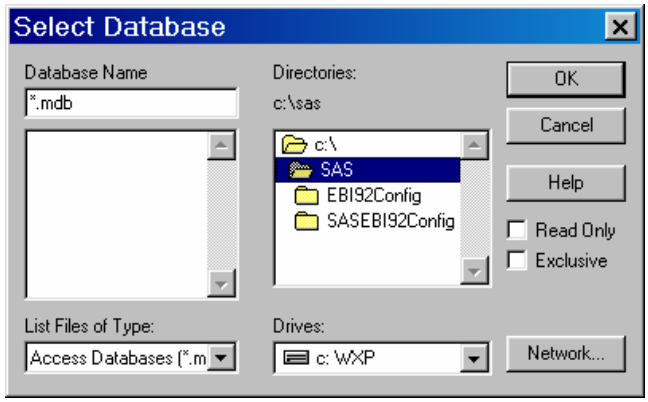
The ODBC Microsoft Access Setup dialog opens.

4. Type a name for your data source, for example **PMOut**, and click **Select**.



The Select Database dialog opens.

5. Browse to the location of your input or output database, and then click **OK**. Click **OK** again to close the ODBC Data Source Administrator dialog.

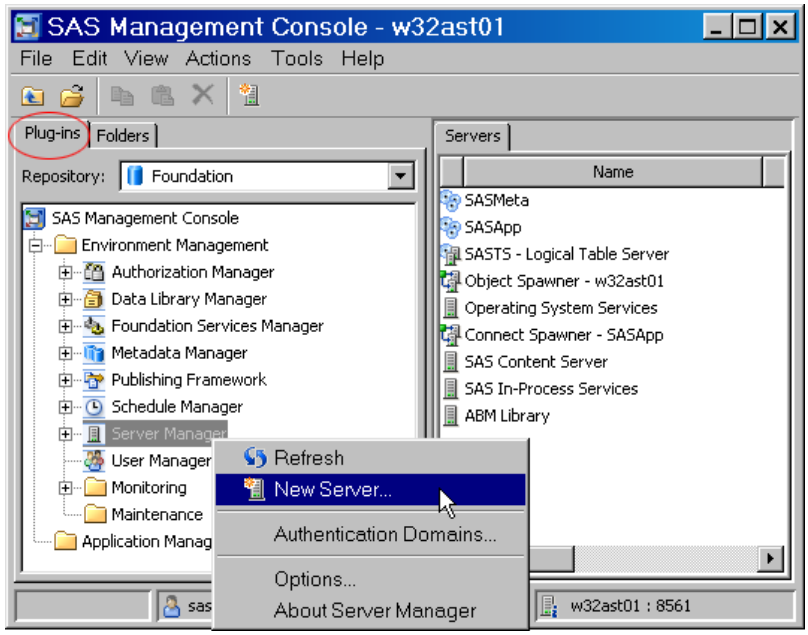


The ODBC connection is created.

Creating a Server

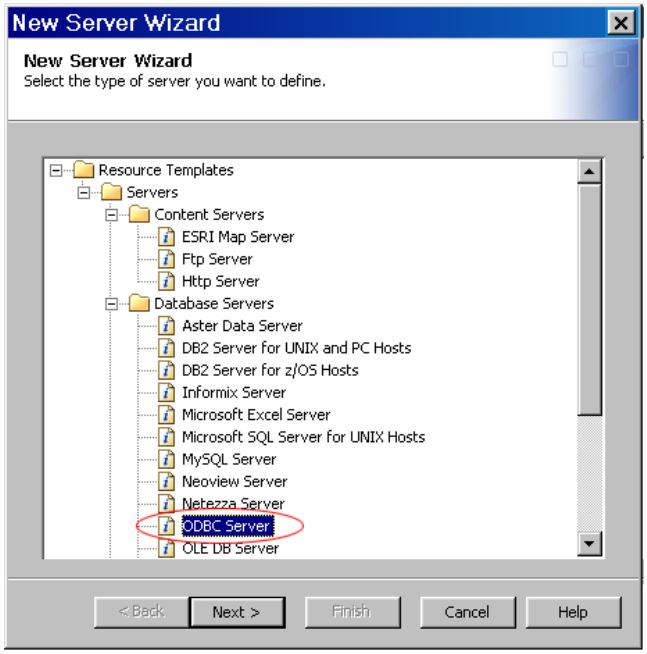
After creating the ODBC connection, use SAS Management Console to create a server.

1. Log on to SAS Management Console as administrator, connecting to your SAS Metadata Server.
2. On the **Plug-ins** tab, right-click **Server Manager** and select **New Server**.

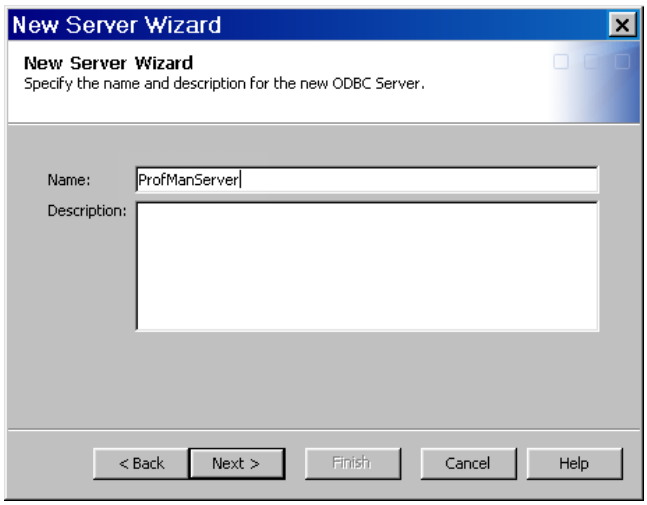


The New Server wizard opens.

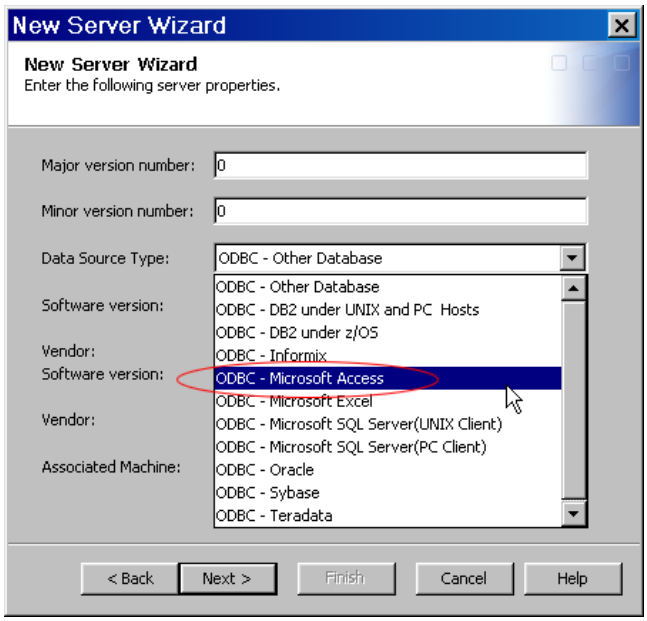
3. Select **ODBC Server** and click **Next**.



4. Name the server, for example **ProfManServer**, and then click **Next**.

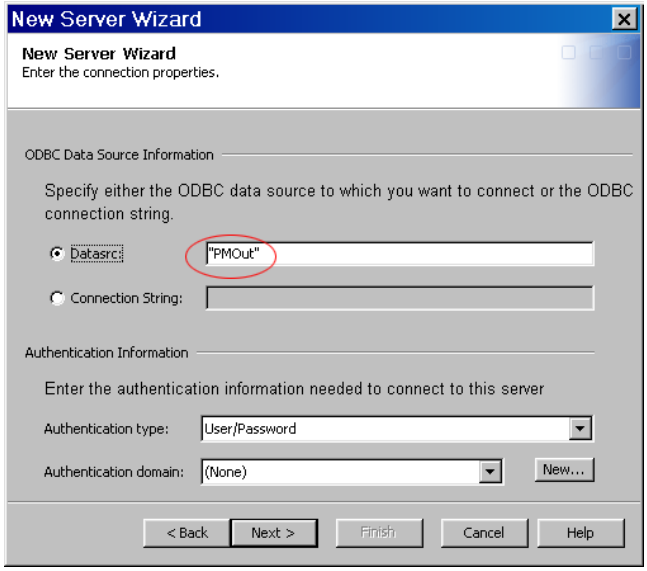


5. Select **ODBC Microsoft Access** as the data source type, and then click **Next**.

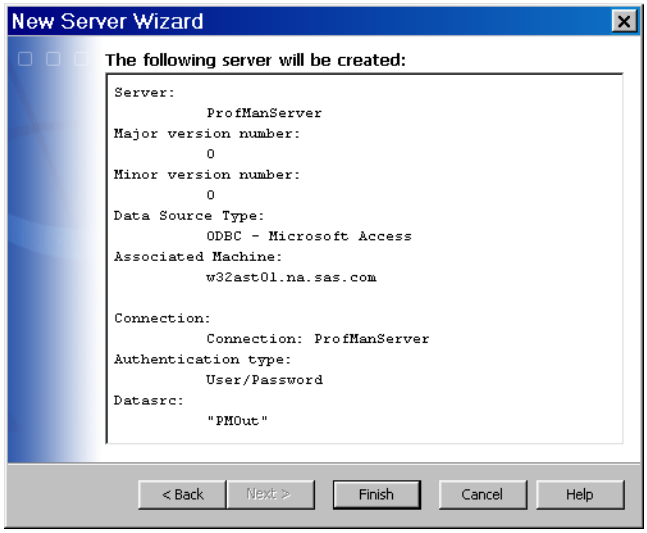


6. Select **Datasrc** and type the datasource name, **"PMOut"**, that you specified when creating the ODBC connection (be sure to include the quotation marks).

   Also specify the database user name and password, and then click **Next**.



7. Review your options, and then click **Finish**.

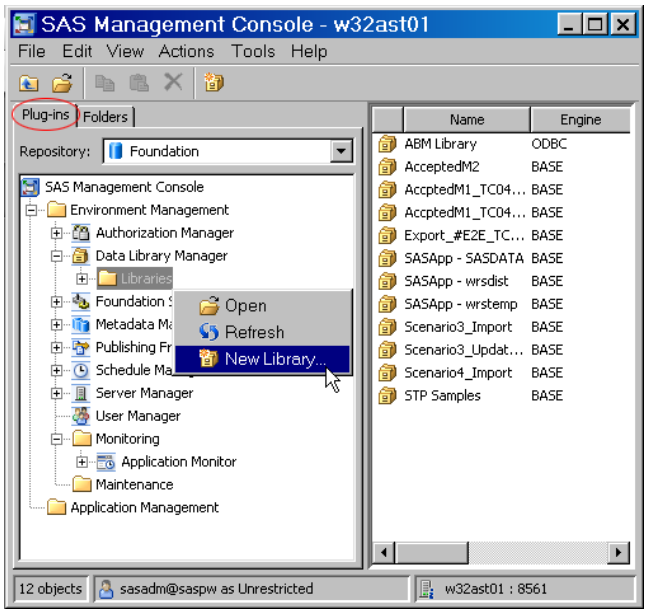


Creating a Database Library

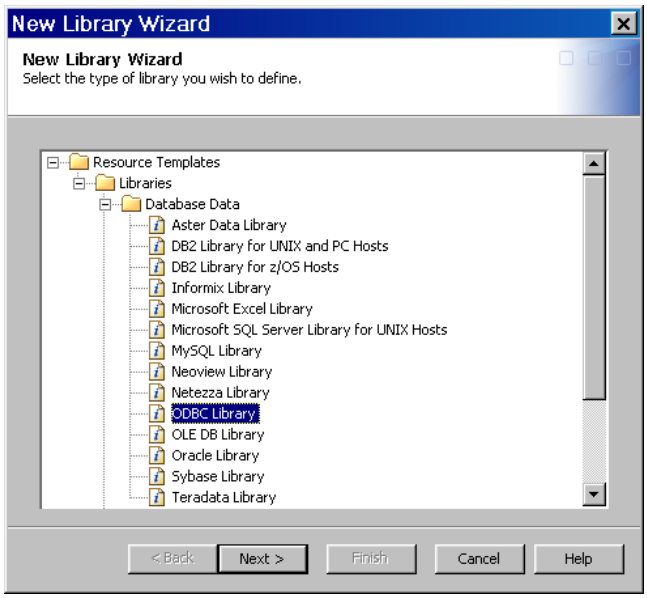
SAS Profitability Management supports any type of library for which you can create a libref, so your input and output libraries can be connected to a database rather than containing SAS data sets. The Analysis library, however, can contain only SAS data sets. So, if you choose to connect your input and output libraries to a database, you must create a separate Analysis library for SAS data sets.

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

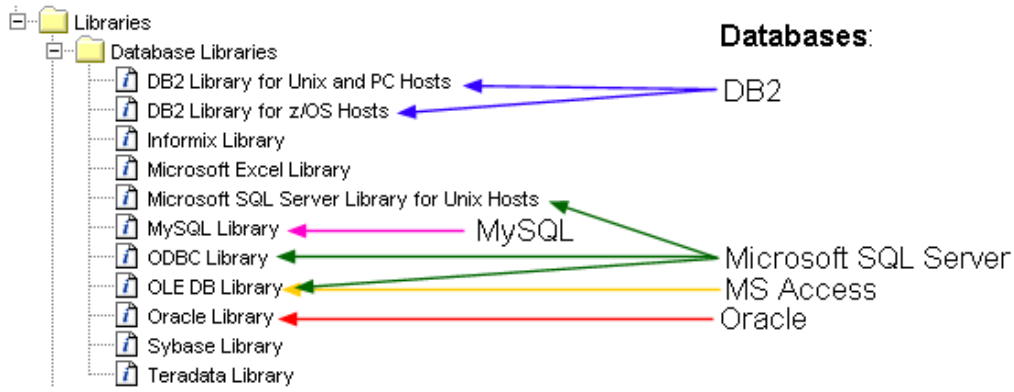
1. Log on to SAS Management Console with the user account that you just created.
2. Click **Plug-ins**.
3. Select the **Foundation** repository.
4. Expand **Data Library Manager**.
5. Right-click **Libraries**.
6. Click **New Library**.



7. Select among the database libraries for the type of library to be created, and then click **Next**.



In this graphic, the **ODBC Library** is selected, but the following library types are also available:



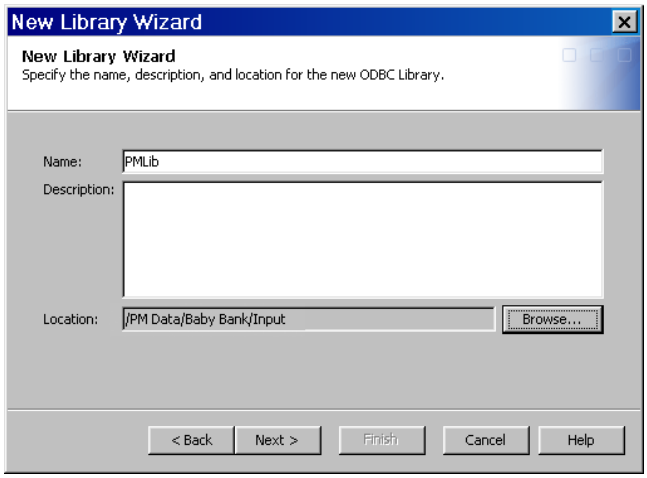
The following table shows the appropriate database library choice for each of the possible database types:

Database	Appropriate Database Library
DB2	DB2 Library for Unix and PC Hosts DB2 Library for z/OS Hosts
Microsoft SQL Server	ODBC Library (Windows) OLE DB Library (Windows) Microsoft SQL Server Library for UNIX Hosts
MySql	MySQL Library
Oracle	Oracle Library

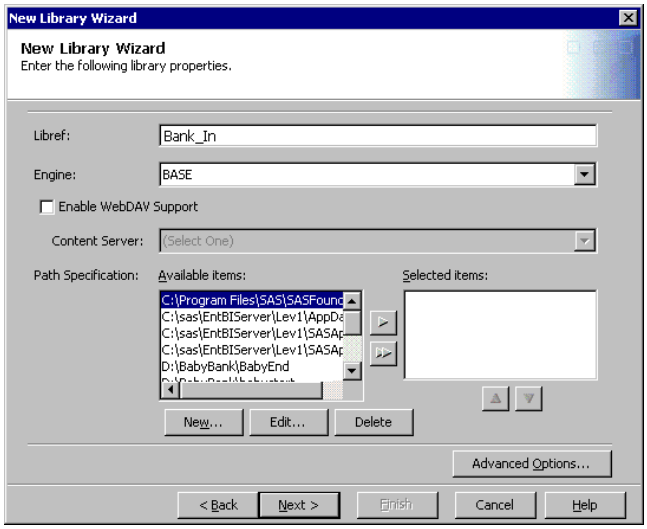
Database	Appropriate Database Library
MS Access	OLE DB Library (Windows)

8. Name the library and specify its associated folder, and then click **Next**.

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

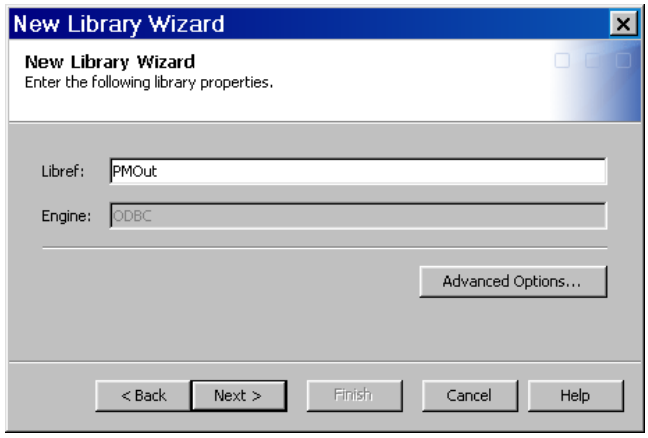


9. Select **SASApp** as the SAS server, and then click **Next**.

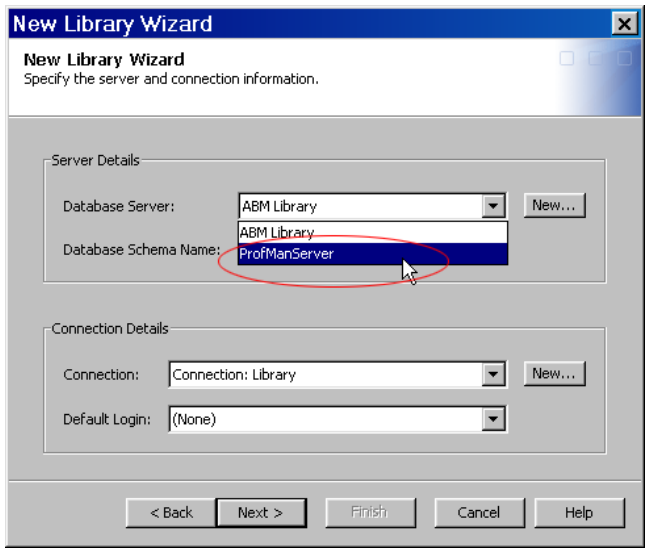


10. Type a **Libref** name, and then click **Next**.

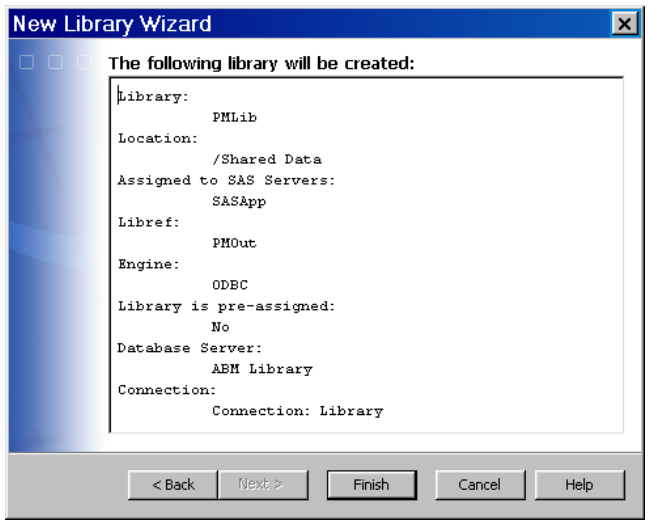
    The libref is used to reference a library without having to refer to its actual physical location.



11. Specify the **Database Server Name** that you specified when creating the server and specify the connection information. Click **Next**.



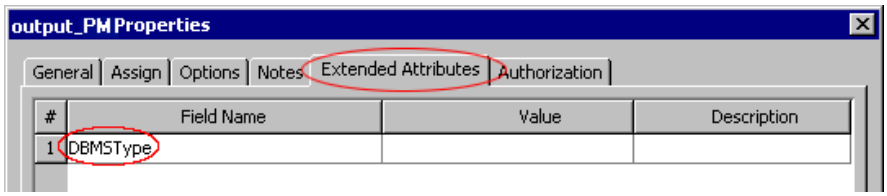
12. Review the library information, and then click **Finish**.



13. Click **View** ⇨ **Refresh** from the menu to see the library listed.
14. Right-click the newly created library and select **Properties**.

The Properties window opens.

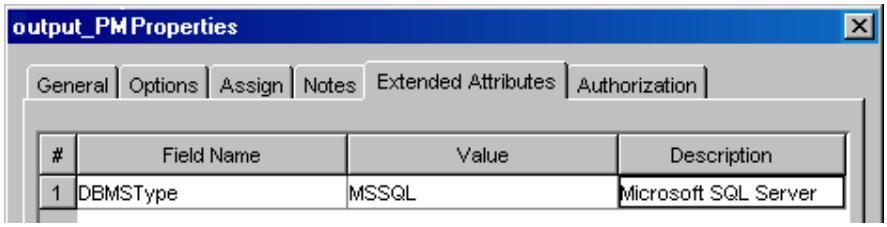
15. Select the **Extended Attributes** tab.
16. Click **New**.
17. In the Field Name column, type **DBMSType**.



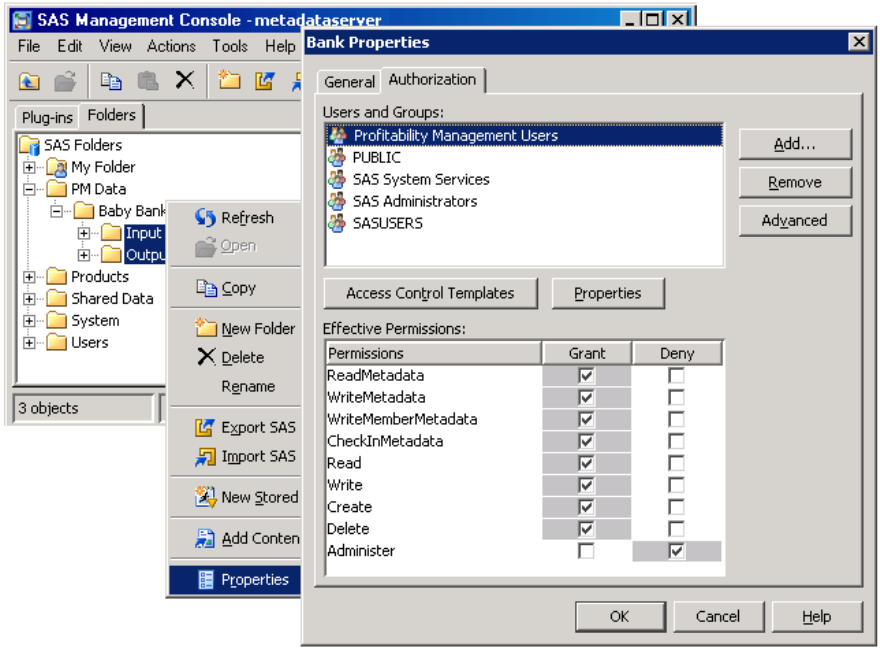
18. In the Value and Description columns, type one of the following pairs depending on your database (the Description is optional).

Value	Description
MSSQL	Microsoft SQL Server
Oracle	Oracle
DB2	DB2
MySql	MySql
Jet	MS Access

The following graphic shows as example in which Microsoft SQL Server is specified:



19. Give the group, Profitability Management Users, permission to use this library.

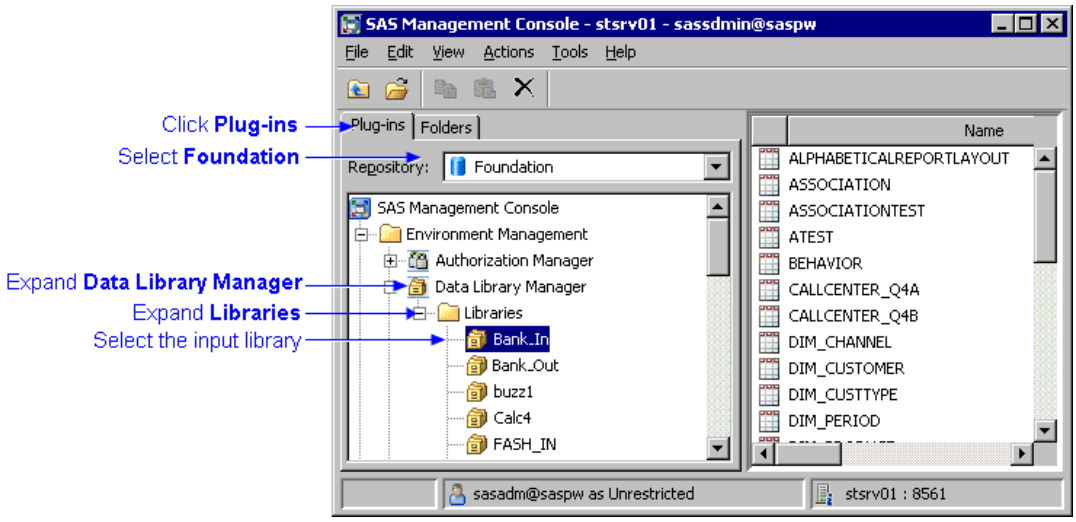


Register Tables in the Input Directory

The following instructions apply whether you use SAS data sets or database tables. Even though you used the operating system or a database management system to copy source files into the input directory, SAS does not yet know about them. Use SAS Management Console to store metadata regarding the files by registering them. Only structural metadata is retained by SAS Management Console, and not the actual content of the files.

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

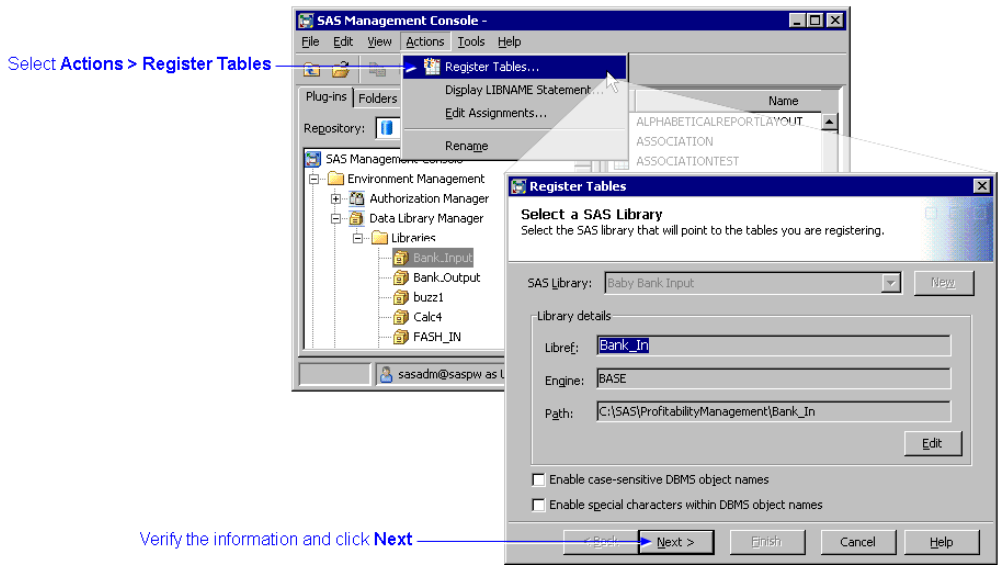
1. Log on to the SAS Management Console as administrator.
2. Click **Plug-ins**.
3. Select the **Foundation** repository.
4. Expand **Data Library Manager**.
5. Expand **SAS Libraries**.
6. Select the input library, **Bank_In**.



7. Select **Actions** ⇨ **Register Tables**.

   The Register Tables window opens.

   Verify that the library information is correct, and click **Next**.



   The log on dialog opens for SASApp.

8. Log on to SASApp.

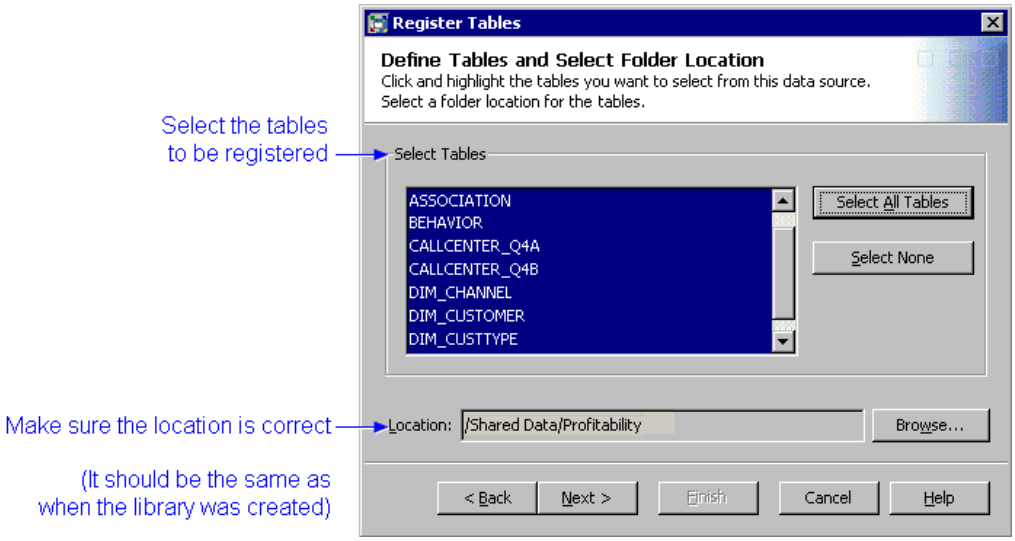
   Note: You should have specified SASApp as the workspace server for this library when the library was created. (You can change the server for the library from the Assign tab of the Properties window for the library in SAS Management Console.)

9. Select the tables to register.

Note: The name of a SAS table cannot contain a blank space or exceed 32 characters.

Make sure the location of the registration metadata is correct.

Note: The location should be the same as was specified when the library was created. (You can change the location of the library metadata from the **General** tab of the Properties window for the library in SAS Management Console.)



10. Click **Next**

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

Chapter 5

Create a New Profitability Model

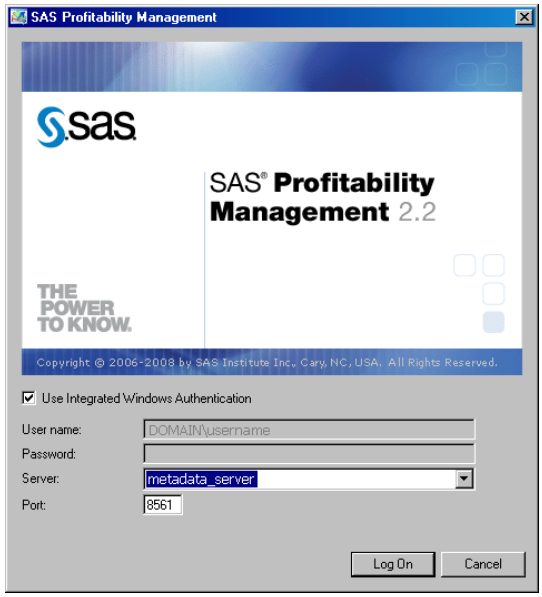
Introduction

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

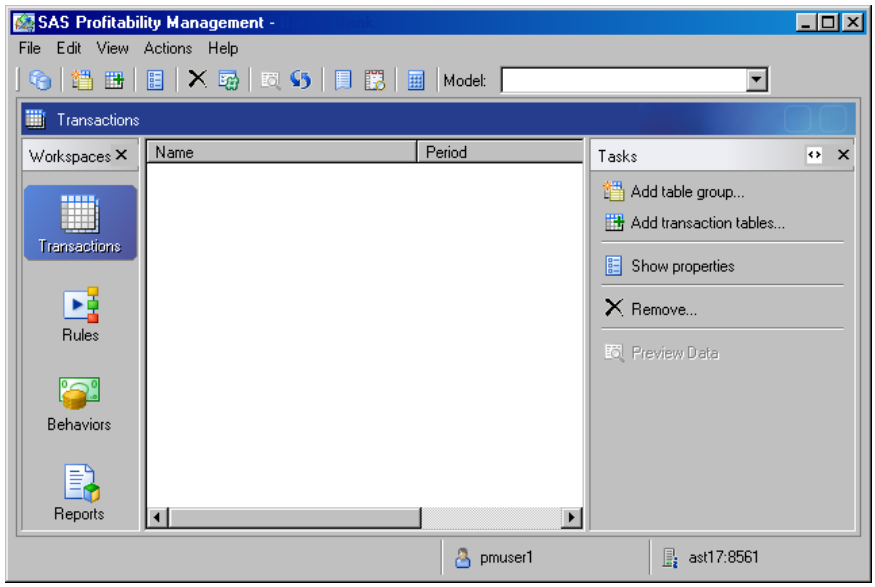
Open the Profitability Management Client Application

Open the SAS Profitability Management rich client application.



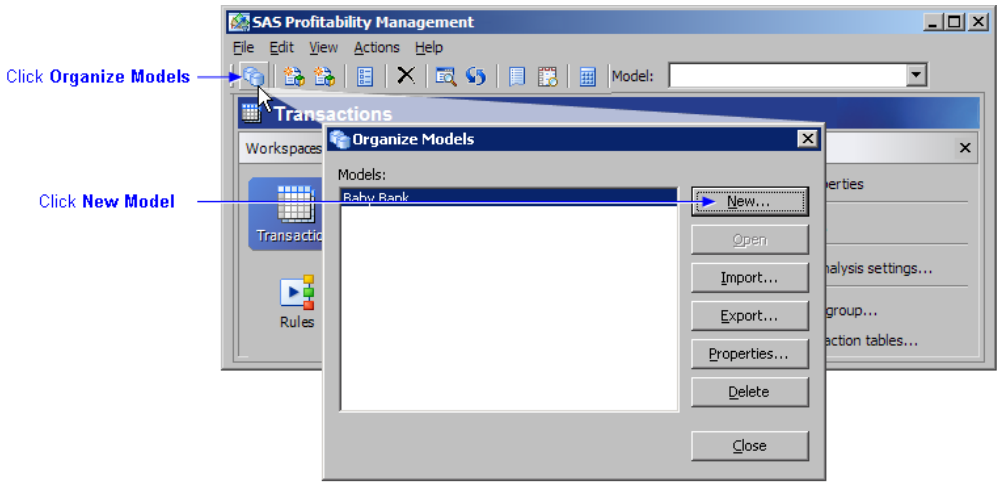
1. Specify the user ID and password that you created in the SAS Management Console.
2. Specify the SAS Profitability Management server. This is dependent on your installation.
3. Specify the port. 8561 is the default for the the SAS Metadata Server.
4. Click **Log On**.

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**.



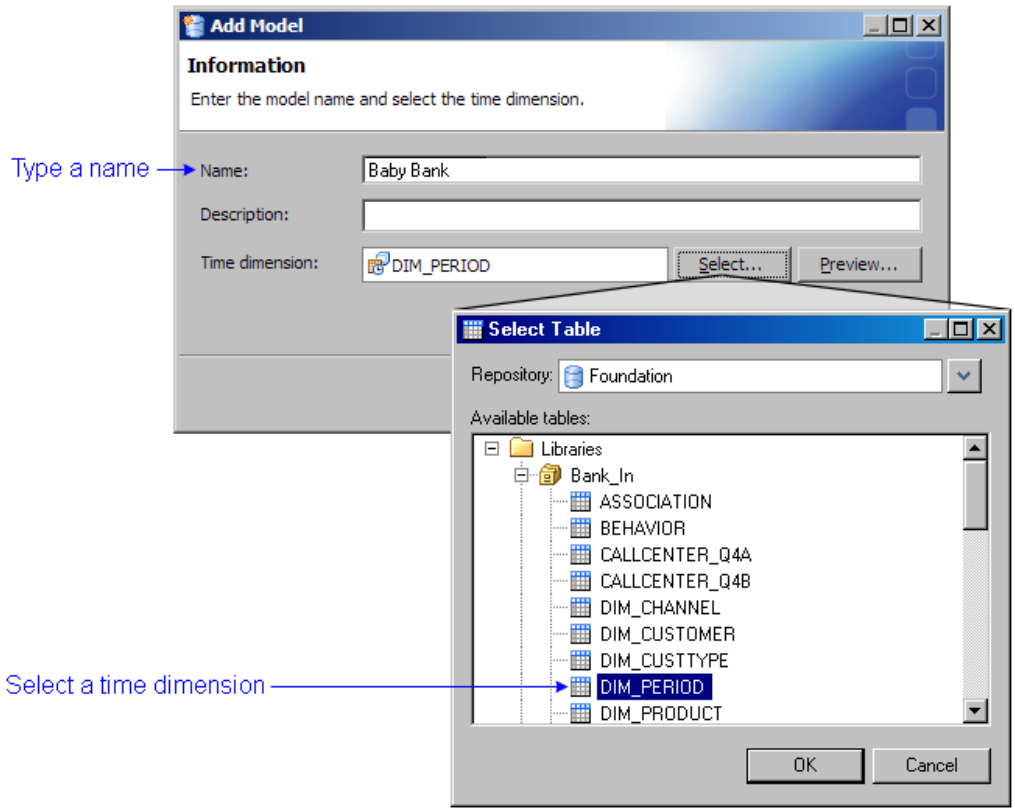
The Organize Models window opens.

Note: If this is the first time that the SAS Profitability Management client has ever been invoked, then the Organize Models window opens automatically.

Name the Model and Select the Time Dimension

1. Name the model **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).

ID	L1_Scenario	L2_Year	L3_Quarter
2006_Q4_Actual	Actual	2006	2006_Q4
2006_Q4_Budget	Budget	2006	2006_Q4

5. Click **Next**.

Select the Output Libraries

1. Specify **Baby_Bank** as 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 **Bank_Out** as 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.

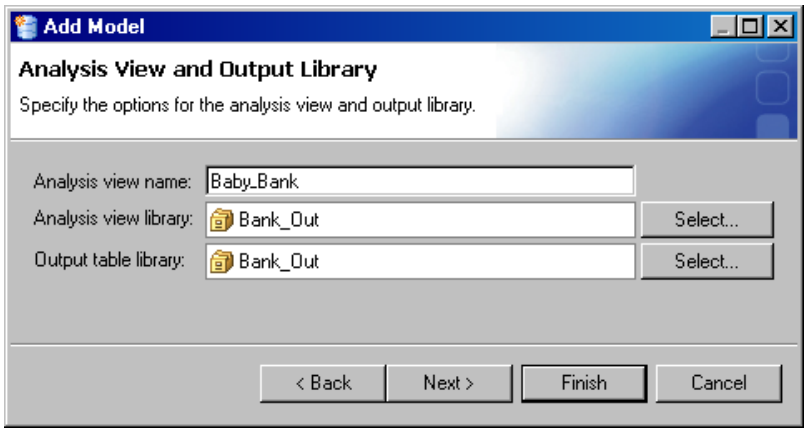
   Note: The Analysis view library must be a SAS Base Engine Library.

3. Also specify **Bank_Out** as 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 Libary, 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 "Using Database Libraries" on page 22.

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

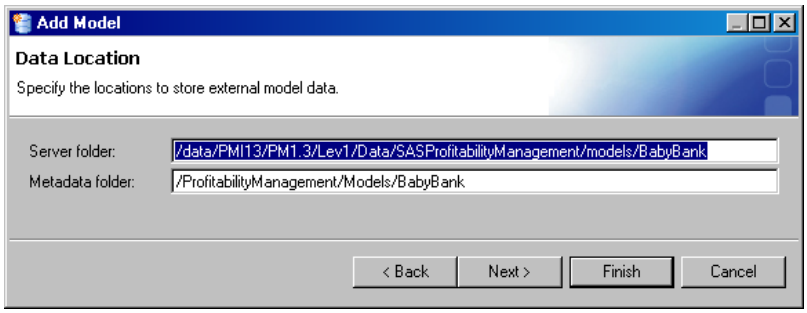


4. Click **Next**.

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

What is a Behavior Table?

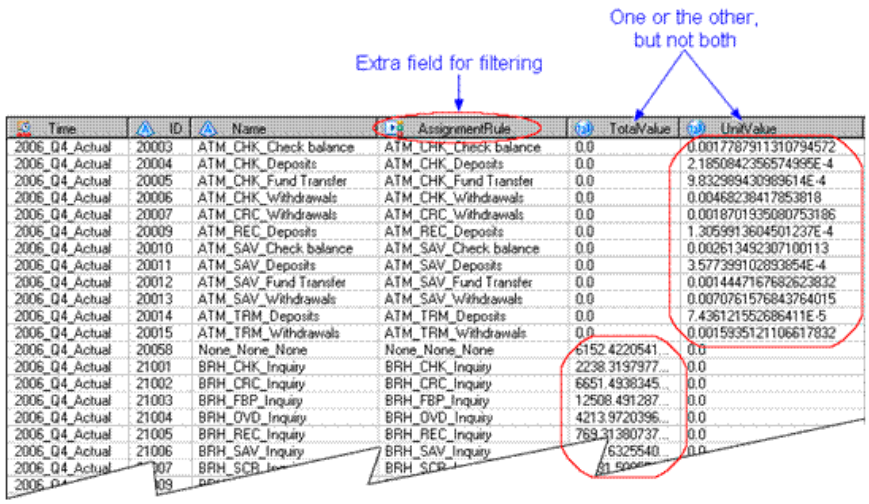
Behaviors, typically, are things that your customers do. For example, the customers of a bank can check their balance, make deposits, transfer funds, and make withdrawals.

Each row in a behavior table:

- represents the lowest level of drill-down in an OLAP view " so you should not create more behaviors than you expect to view.
- is assigned to a transaction by a rule " so you should create only as many behaviors as you have rules for assignment.

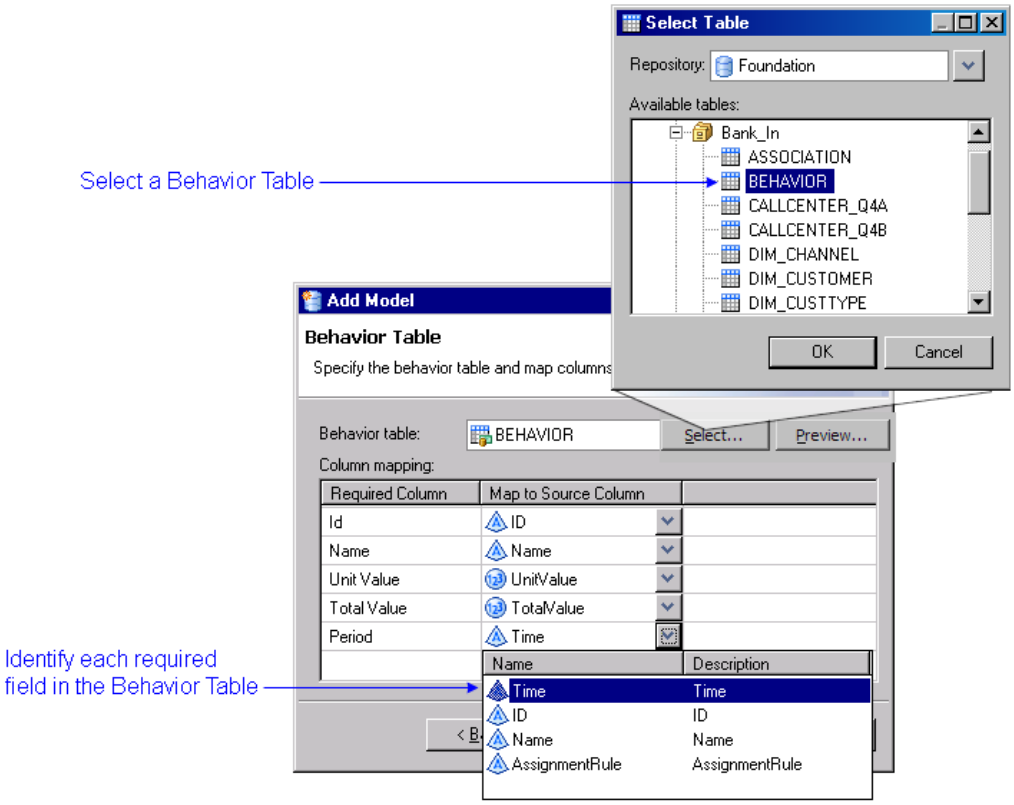
The following graphic shows part of the sample behavior table, BEHAVIOR. Notice the following:

- 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. Sucha field is not required for a standard model.



How to Do It

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

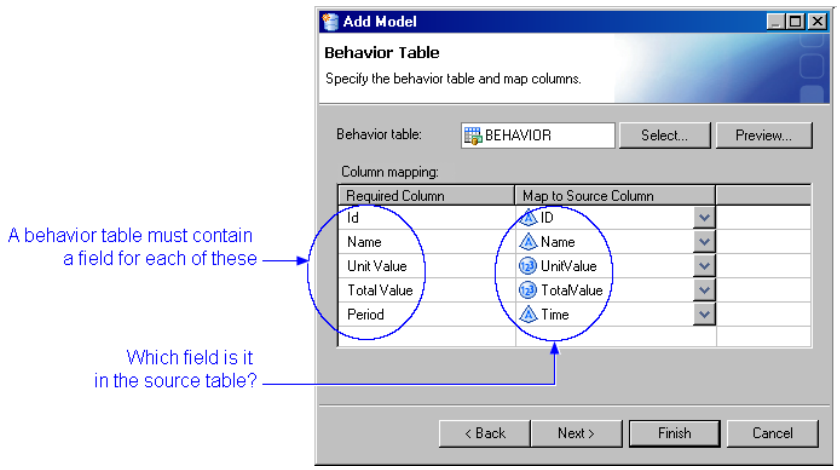


3. Click **OK**.
4. Identify fields in the behavior table.

   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.

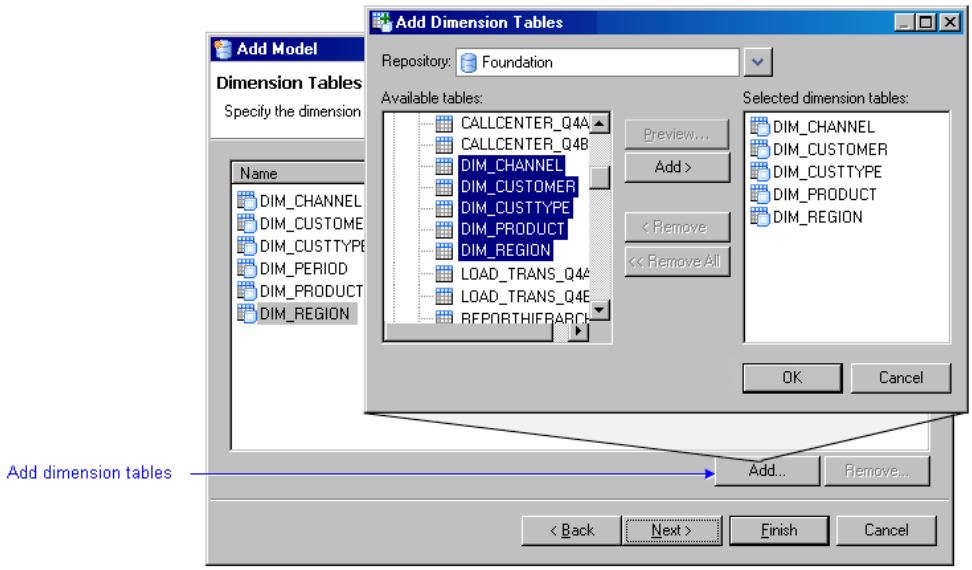


5. Click **Next**.

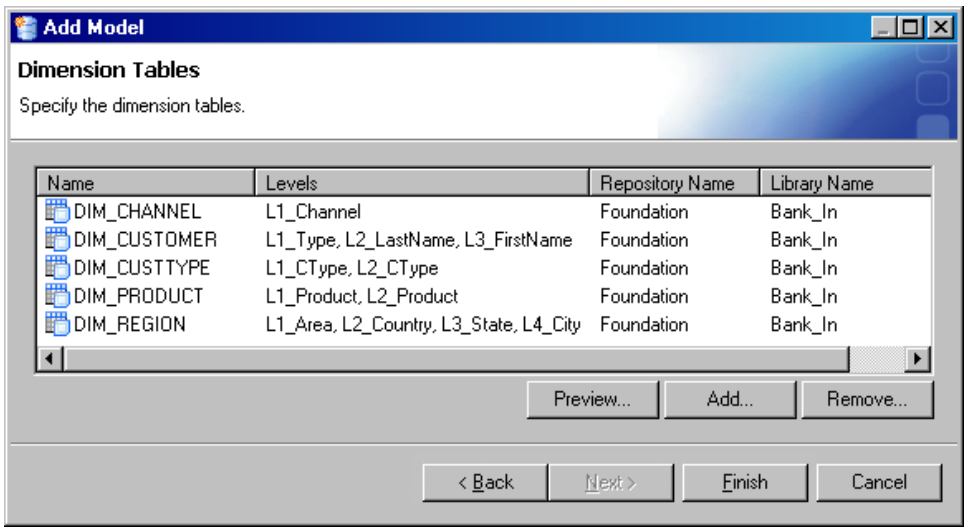
Identify Dimension Tables

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

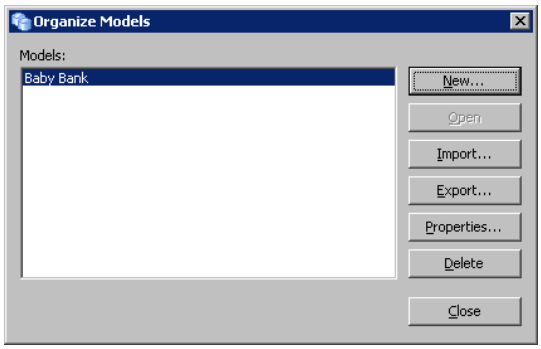
   Note: You can select multiple dimensions by using the Shift or Ctrl key.
3. Click **OK.**



The dimension tables are added to the model as shown in the following graphic.



4. Click **Finish**, and verify that the new model, **Baby Bank**, is added to the list of models.



5. Click **Close**.

Chapter 6

Define Transaction Table Groups

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 pre-defined rules definition table.

Define the ABMCost Group

Overview

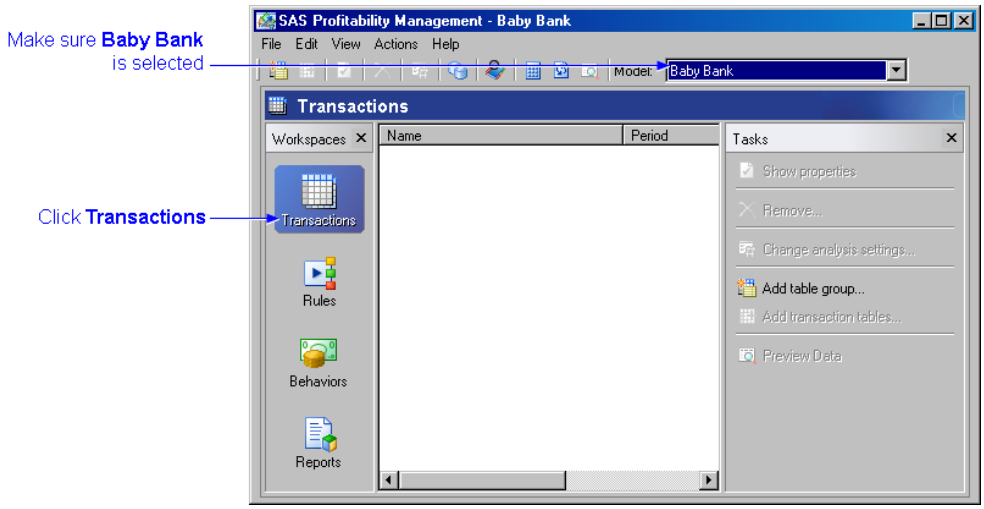
Defining transaction table groups is a two-part process:

- 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.

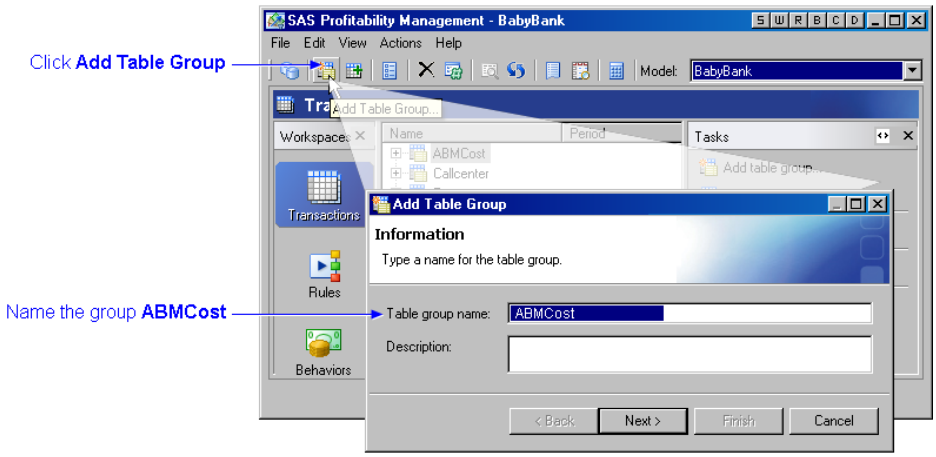
   Make sure that the **Baby Bank** model is selected.



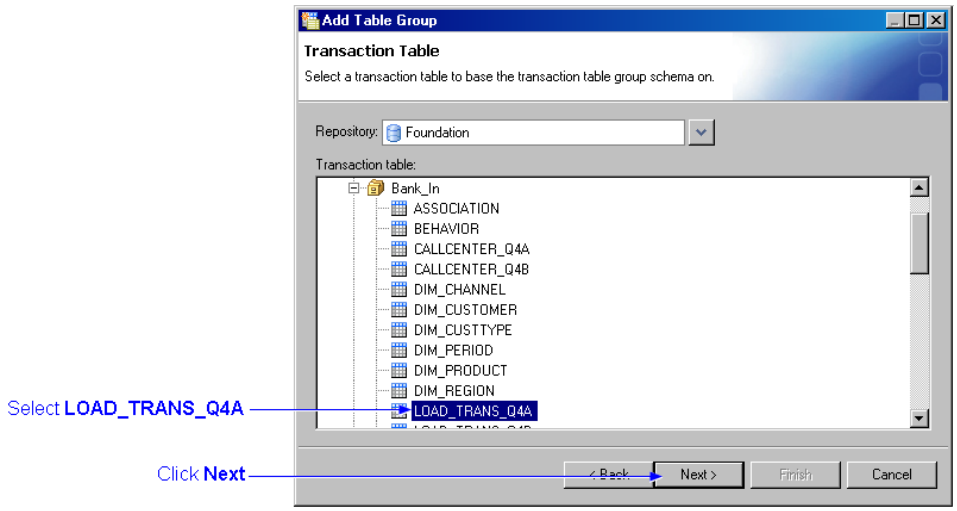
2. Click **Add table group** (or the Add table group icon ).

   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.



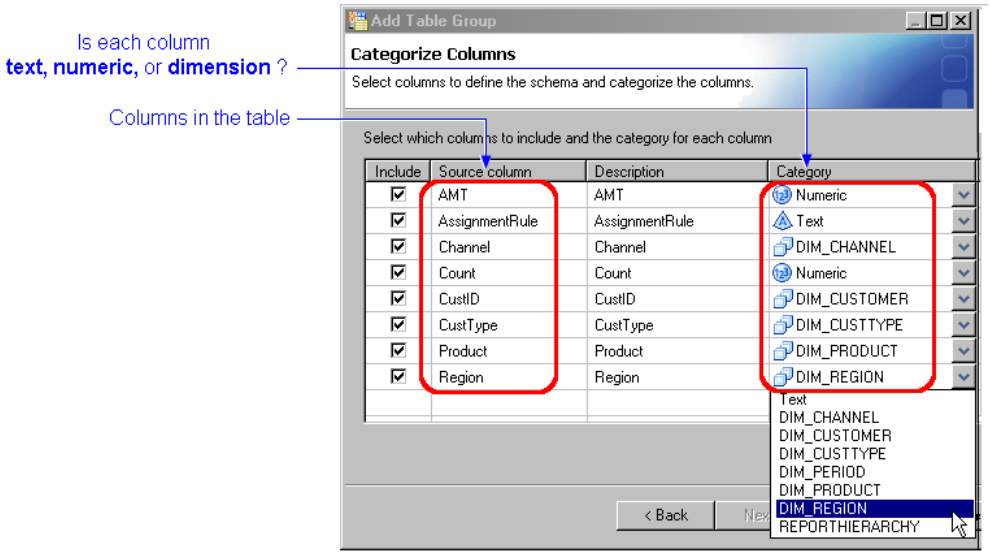
Note: All the tables in a table group share the same schema, and each table is associated with a different time period.

5. Click **Next**.

6. Specify whether each column in the table is

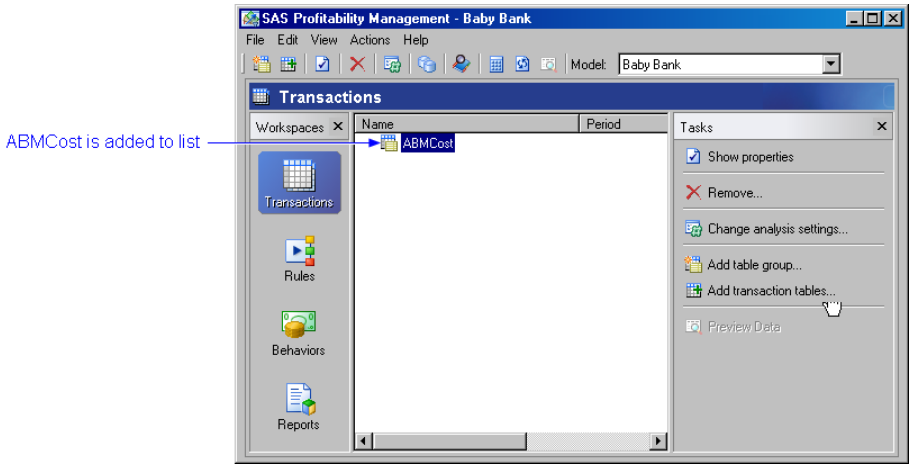
   - Text
   - Numeric
   - Dimension member (if a dimension member, specify which one).

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



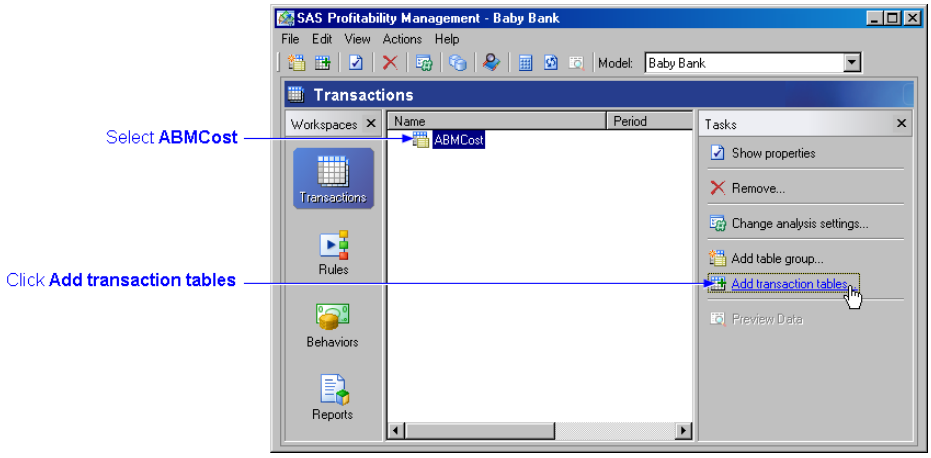
7. Click **Finish**.

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

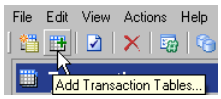


Add Tables to the Group

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



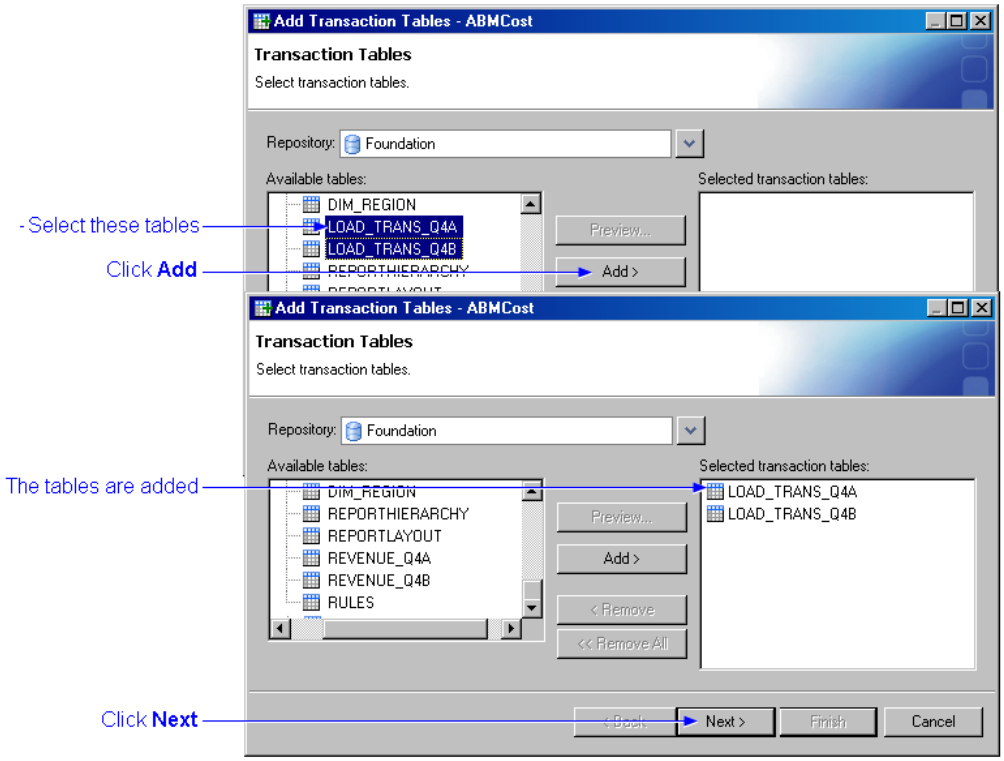
Or click the icon.



The Add Transaction Tables window opens.

2. Select **LOAD_TRANS_Q4A** and **LOAD_TRANS_Q4B**.
3. Click **Add**.

   The tables are added.

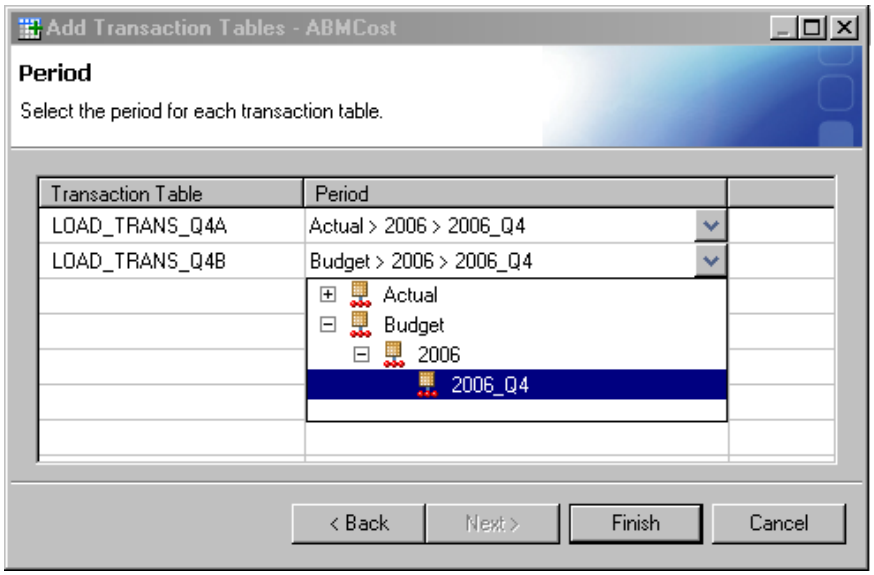


4. Click **Next**.

   The Period window opens.

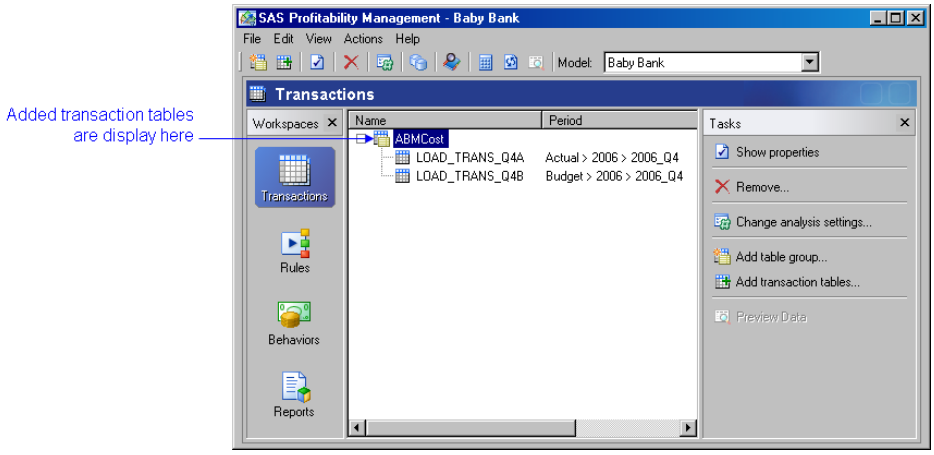
5. Associate each of the tables with a period as follows:

Table	Period
LOAD_TRANS_Q4A	Actual > 2006 > 2006_Q4
LOAD_TRANS_Q4B	Budget > 2006 > 2006_Q4



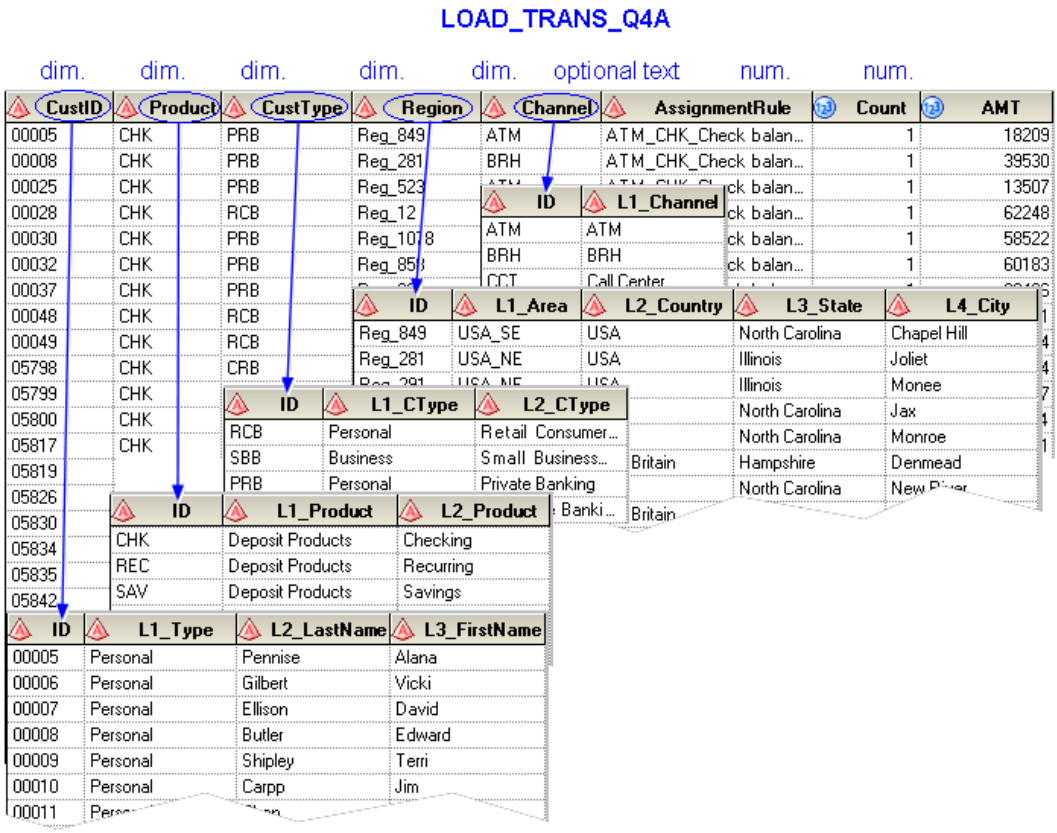
6. Click **Finish**.

   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 the CallCenter Group

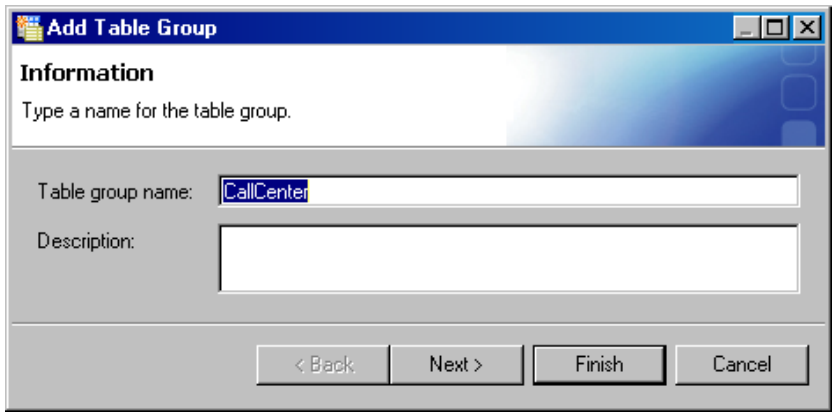
Overview

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

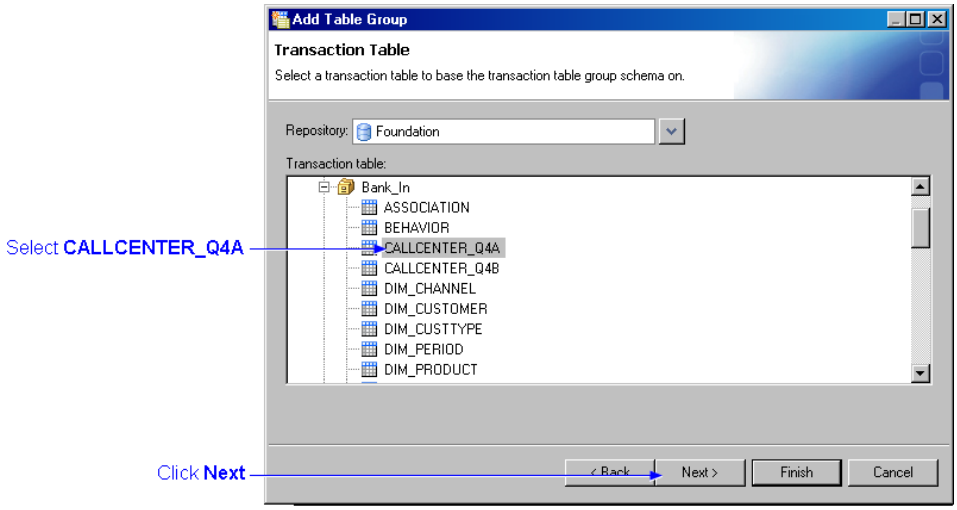
- Define the transaction table group.
- Add transaction tables to the group, and associate each transaction table with a time period.

Define the Transaction Table Group

1. Select the **Transaction** workspace.
2. Click **Add table group**.
3. Name the group **CallCenter**, and then click **Next**.

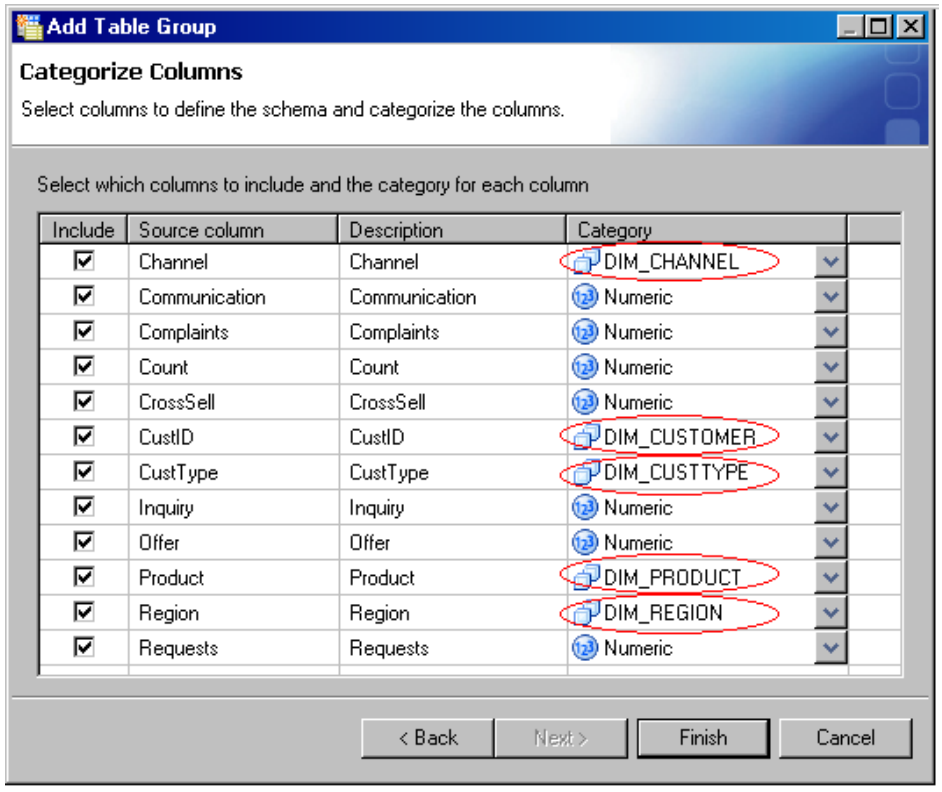


4. Select **CALLCENTER_Q4A** as the table whose schema serves as the schema for the table group.

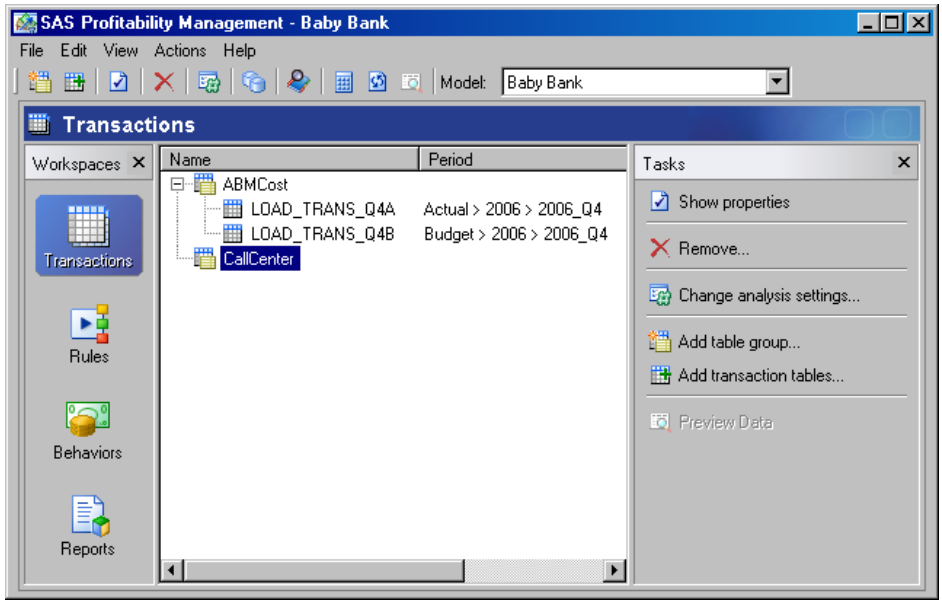


5. Click **Next**.

6. By clicking the drop-down arrow for each column, specify whether the column in the table is text, numeric, or a dimension member as shown in the following graphic, and then click **Finish**.



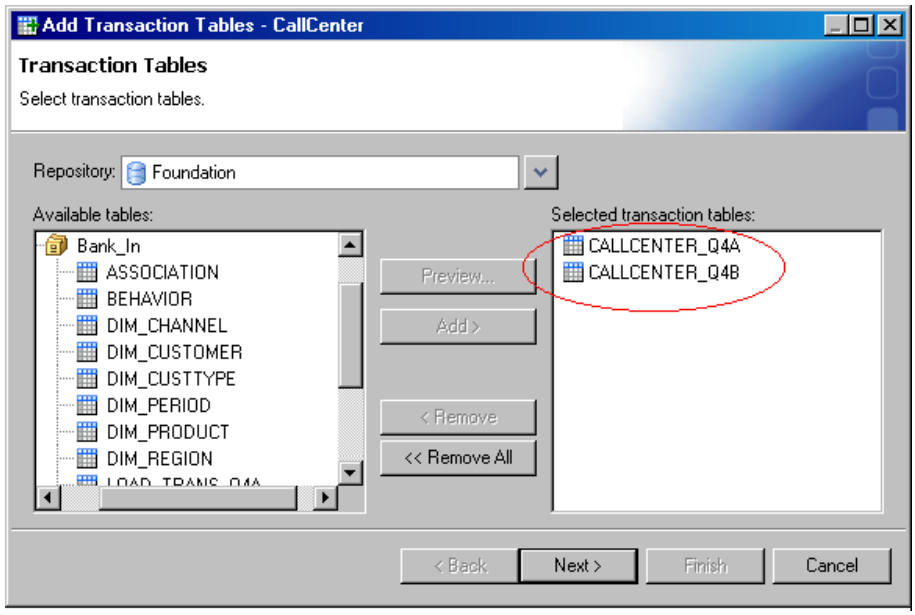
The table group, CallCenter, is added to the list.



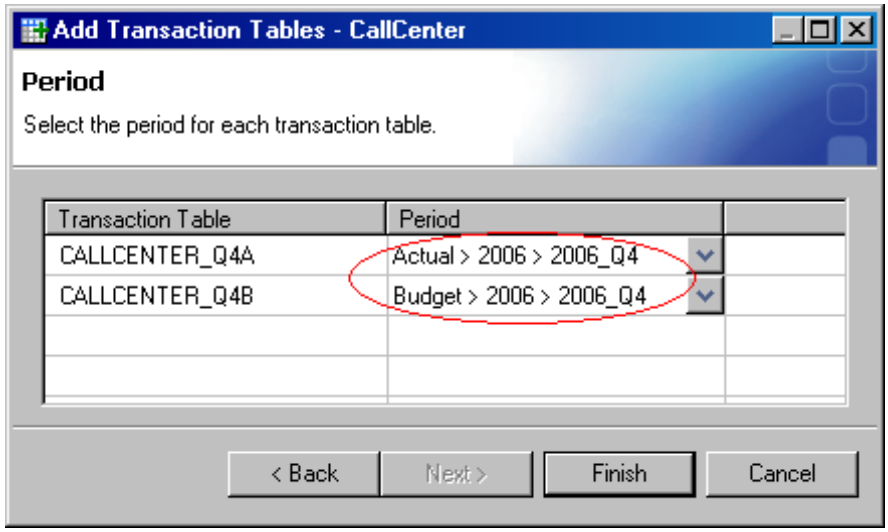
Add Tables to the Group

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**.

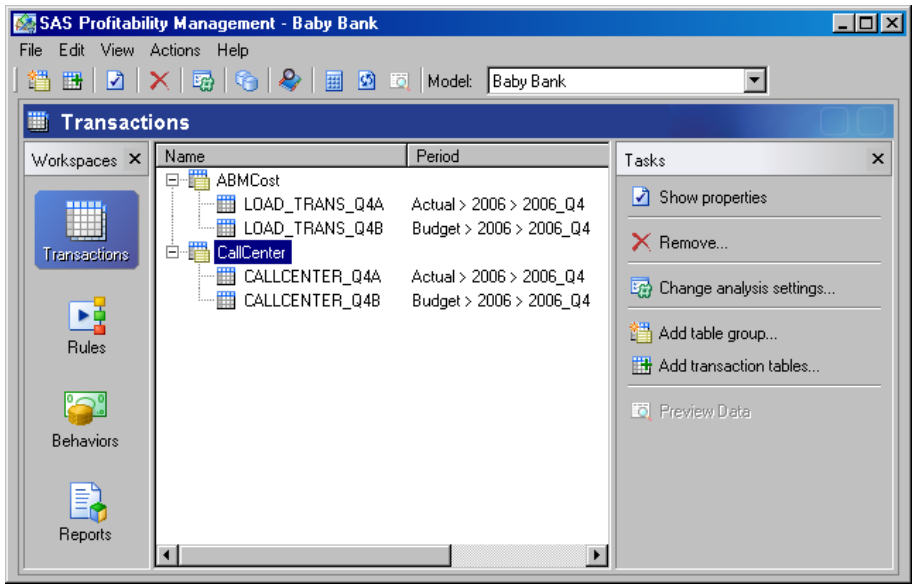


3. Associate each of the tables with a period as shown:



4. Click **Finish**.

   The tables are added to the group.



Transaction Table CALLCENTER_Q4A

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

CALLCENTER_Q4A

dim. Product	dim. CustType	dim. Region	dim. Channel	num. Communication	num. Complaints	num. Inquiry	num. Requests	num. CrossSell	num. Offer	num. Count
CHK	PRB	Reg_87	CCT	2	0	1	0	1	0	1
CHK	PRB	Reg_87	CCT	298	15	149	49	74	30	1
CHK	PRB	Reg_768	CCT	99	5	50	16	25	10	1
CHK	PRB	Reg_768	CCT	461	23	231	76	115	46	1
CHK	PRB	Reg_70	CCT	6	0	3	1	2	1	1
CHK	PRB	Reg_70	CCT	180	9	90	30	45	18	1
CHK	PRB	Reg_528	CCT	91	5	46	15	23	9	1
CHK	RCB	Reg_202	CCT	155	8	78	26	39	16	
CHK	RCB	Reg_870	CCT	186	9	93	31			
CHK	PRB	Reg_1101	CCT	456	23	228			46	1
CHK	PRB	Reg_218	CCT	313				78	31	1
CHK						156	52	78	31	1
					4	15	5	7	7	1
		Reg_135	CCT	122	18	61	20	31	31	1
UCR	PRB	Reg_1080	CCT	126	19	63	21	31	31	1
UCR	CRB	Reg_59	CCT	144	22	72	24	36	36	1
UCR	SBB	Reg_1090	CCT	72	11	36	12	18	18	1
UCR	SBB	Reg_1131	CCT	25	4	13	4	6	6	1
UCR	CRB	Reg_768	CCT	20	3	10	3	5	5	1
UCR	CRB	Reg_413	CCT	133	20	66	22	33	33	1
UCR	CRB	Reg_696	CCT	119	18	60	20	30	30	1
UCR	CRB	Reg_1003	CCT	5	1	2	1	1	1	1
UCR	SBB	Reg_1093	CCT	145	22	73	24	36	36	1

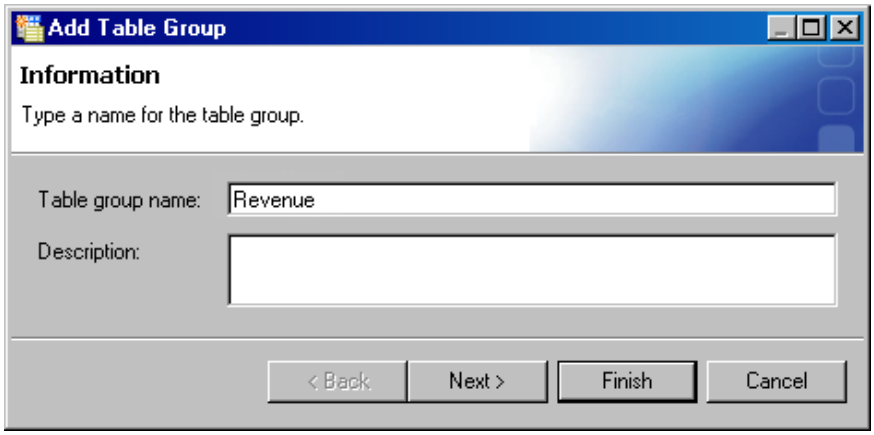
Define the Revenue Group

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

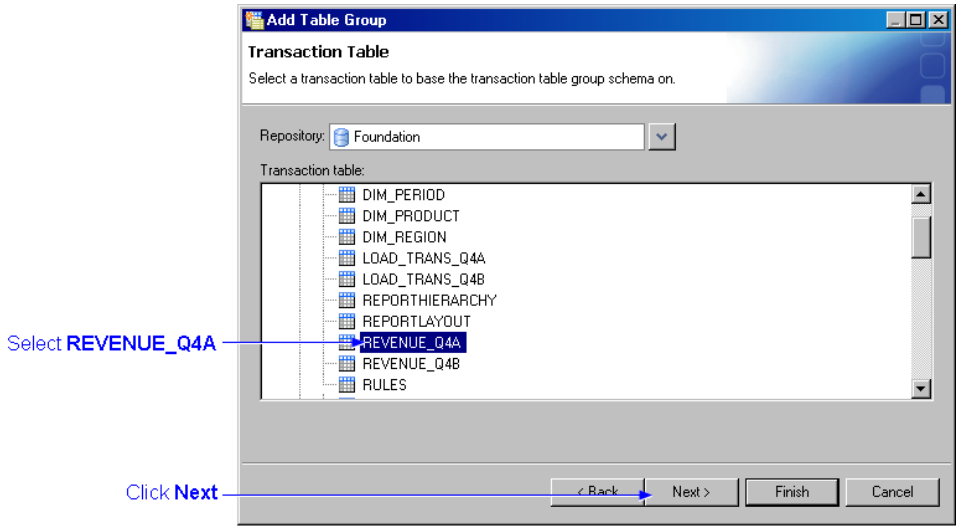
Define the Transaction Table Group

1. Click **Add table group**.

2. Name the group **Revenue**, and then click **Next**.

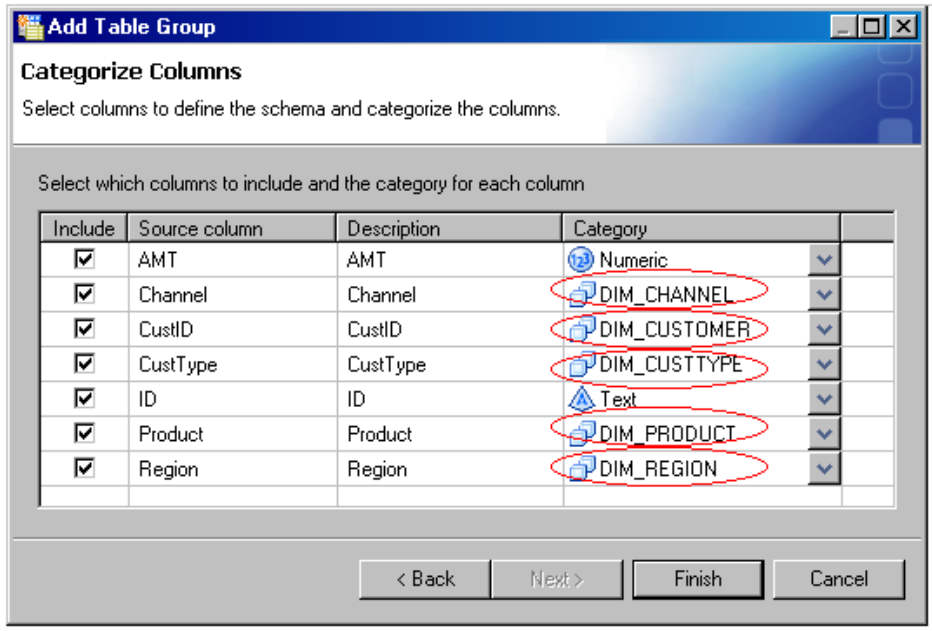


3. Select **REVENUE_Q4A** as the table whose schema serves as the schema for the table group.

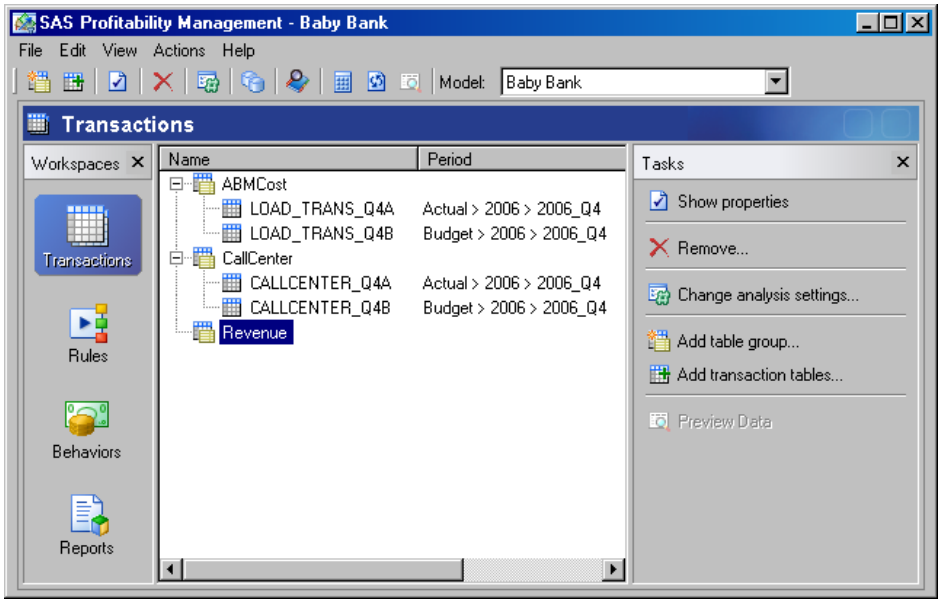


4. Click **Next**.

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**.

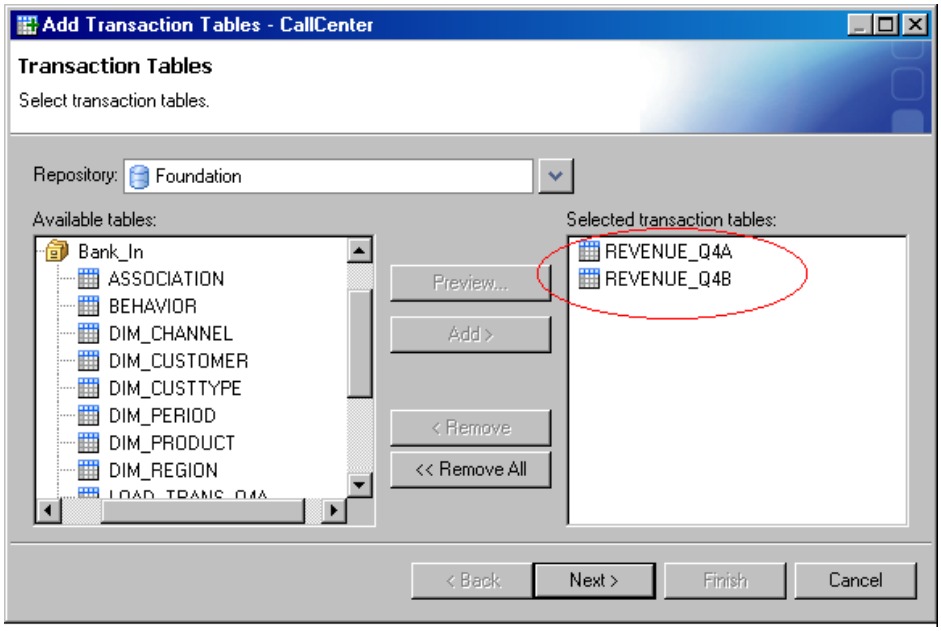


The table group, Revenue, is added to the list.

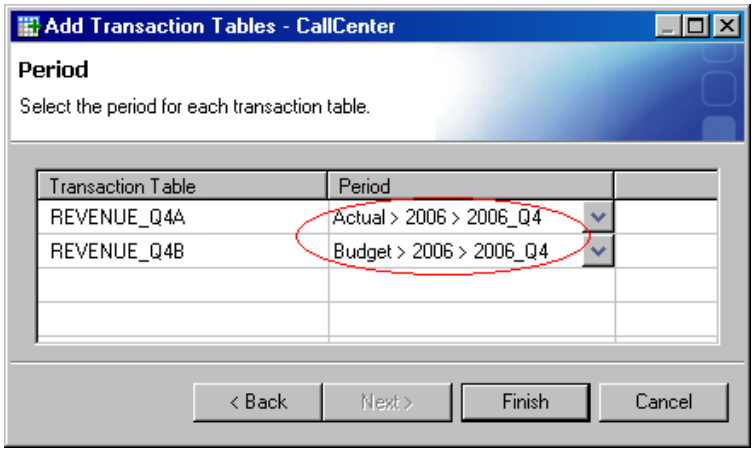


Add Tables to the Group

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**.

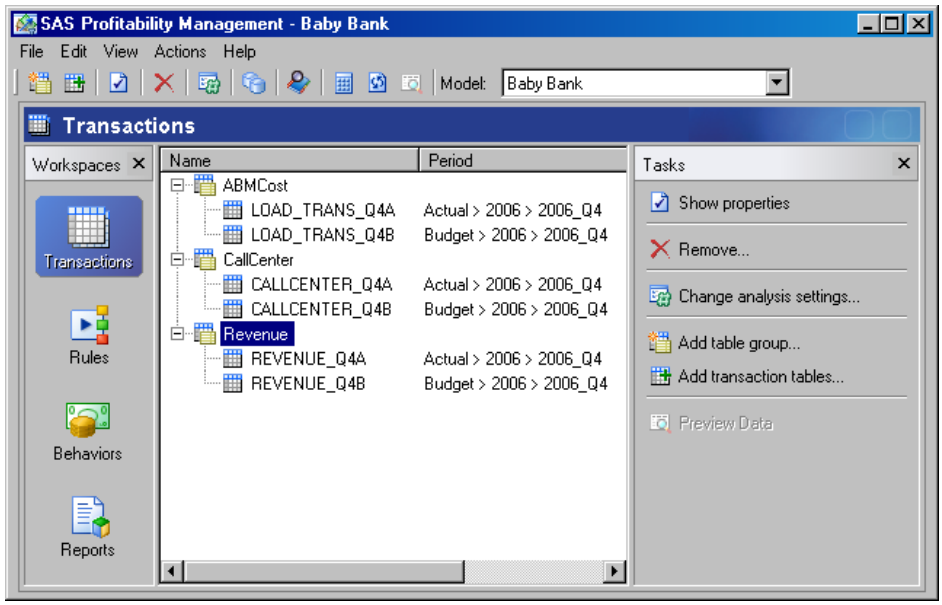


3. Associate each of the tables with a period as shown:



4. Click **Finish**.

   The tables are added to the group.



Transaction Table REVENUE_Q4A

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

REVENUE_Q4A

dim. Channel	dim. CustID	dim. Product	dim. CustType	dim. Region	num. AMT	text ID
ATM	00001	OVD	PRB	Reg_67	399.3	10002
ATM	00001	OVD	PRB	Reg_67	8.16	12002
ATM	00001	OVD	PRB	Reg_67	20.47	13001
ATM	00001	OVD	PRB	Reg_67	25.59	13002
ATM	00001	OVD	PRB	Reg_67	5.12	14001
ATM	00002	CRC	PRB	Reg_188	4359.35	10001
ATM	00002	CRC	PRB	Reg_188	1744.98	12001
ATM	00002	CRC	PRB	Reg_188	134.[illegible]	[illegible]
ATM	00002	CRC	PRB	Reg_188	[illegible]	[illegible]
ATM	[illegible]	[illegible]	[illegible]	[illegible]	221.31	13001
[illegible]	[illegible]	[illegible]	[illegible]	Reg_795	276.64	13002
[illegible]	05845	FBP	[illegible]	Reg_795	55.33	14001
CCT	05845	OTP	CRB	Reg_194	23.38	13001
CCT	05845	OTP	CRB	Reg_194	29.22	13002
CCT	05845	OTP	CRB	Reg_194	5.84	14001
CCT	05846	UCR	SBB	Reg_1093	6478.6	10002
CCT	05846	UCR	SBB	Reg_1093	332.17	13001
CCT	05846	UCR	SBB	Reg_1093	415.22	13002
CCT	05846	UCR	SBB	Reg_1093	83.04	14001
CCT	05847	FBP	SBB	Reg_269	21.04	13001
CCT	05847	FBP	SBB	Reg_269	26.3	13002
CCT	05847	FBP	SBB	Reg_269	5.26	14001
CCT	05847	OVD	CRB	Reg_448	3208.39	10002
CCT	05847	OVD	CRB	Reg_448	164.5	13001
CCT	05847	OVD	CRB	Reg_448	205.63	13002
CCT	05847	OVD	CRB	Reg_448	41.13	14001

Note: The text field **ID** in this table is an optional field that is used by a rule for filtering.

Chapter 7

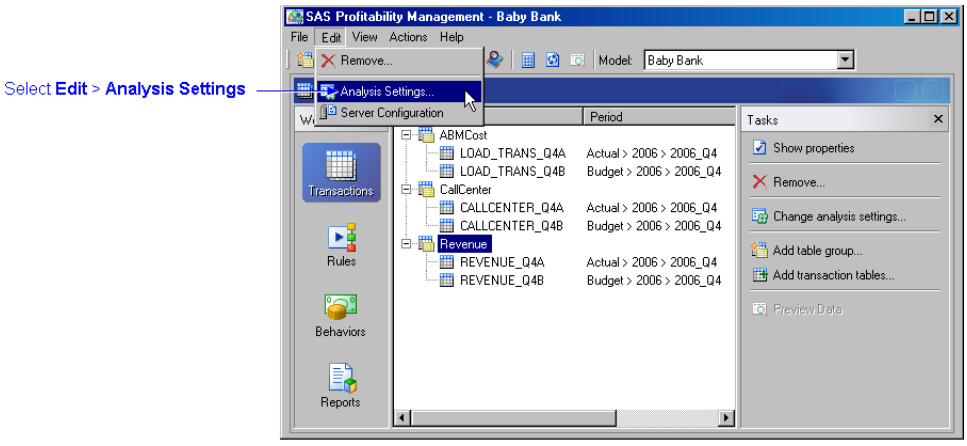
Change Analysis Settings

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.

- Select **Edit** > **Analysis** Settings.

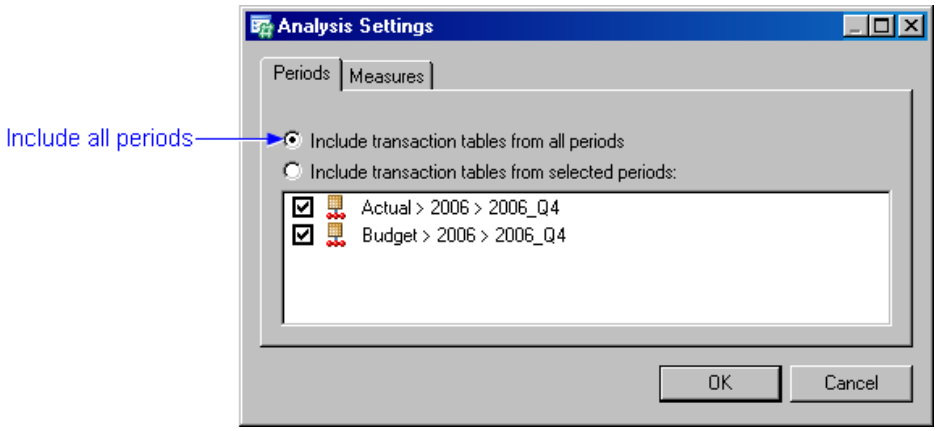
  The Analysis Settings window opens.



Select Periods for the Cube

- 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.

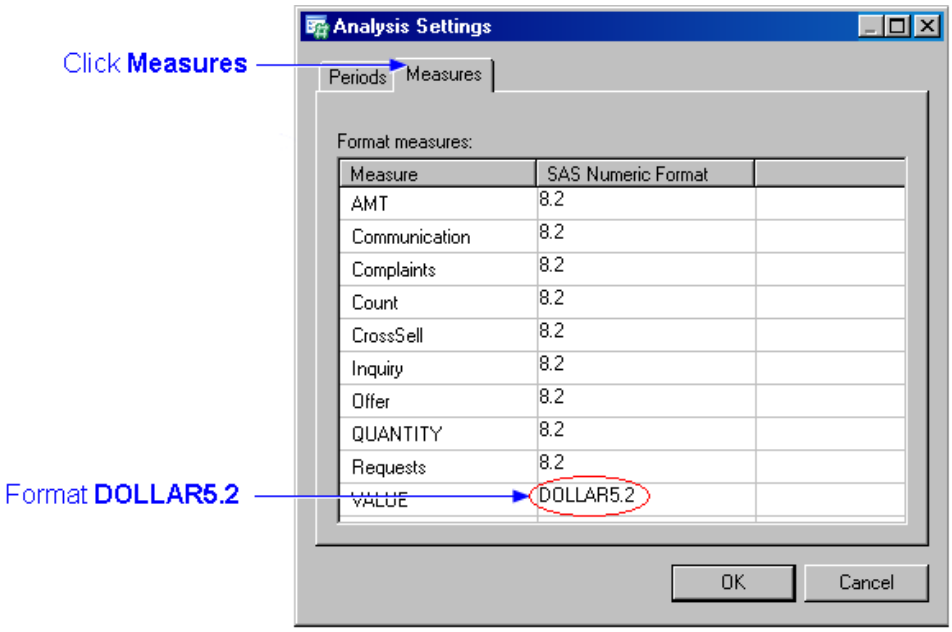
   The Measures tab lists all numeric properties available to the model. All numerical properties are available to be included in profit and loss reporting. This includes:

   - 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, select **DOLLAR12.2**.

   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 80.

   The default number format is 12.2 (twelve characters with two of them after the decimal).



The following SAS formats are supported:

Note: Not all of the following formats are available on the dropdown list. If a format is not on the list, you can type it into the SAS Numeric Format field.

BEST*w*	Best available format with width=w
COMMA*w.d*	Comma and decimal points with width=w and decimal=d
COMMAX*w.d*	Comma and decimal points with width=w and decimal=d (switches the role of comma and decimal points)
DOLLAR*w.d*	Dollar signs, comma, and decimal points with width=w and decimal=d
DOLLARX*w.d*	Dollar signs, comma, and decimal points with width=w and decimal=d (switches the role of comma and decimal points)
EURO*w.d*	Euro signs, comma, and decimal points with width=w and decimal=d
EUROX*w.d*	Euro signs, comma, and decimal points with width=w and decimal=d (switches the role of comma and decimal points)
PERCENT*w.d*	Percentage with width=w and decimal=d and a percentage sign
Z*w.d*	Prefixed with zero to get width=w and decimal=d
w.d	width=w and decimal=d

Chapter 8

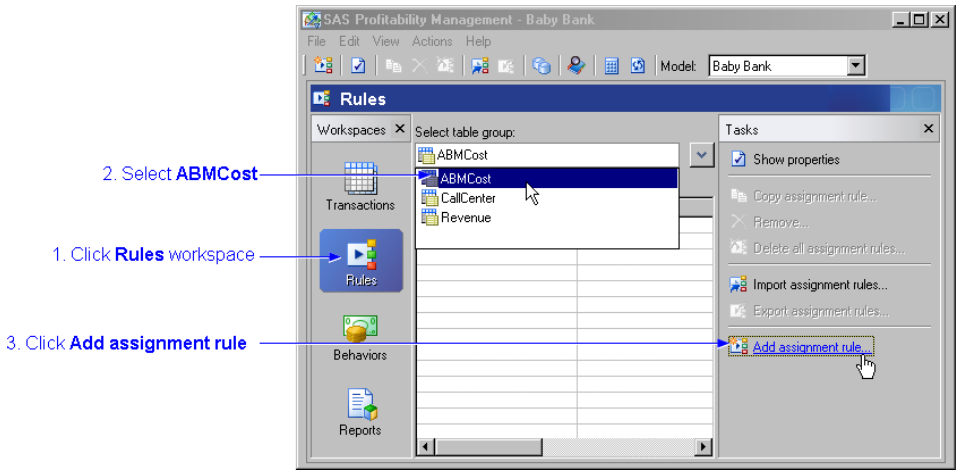
Define Rules

Introduction

An assignment rule associate a behavior with rows in a transaction table. 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.

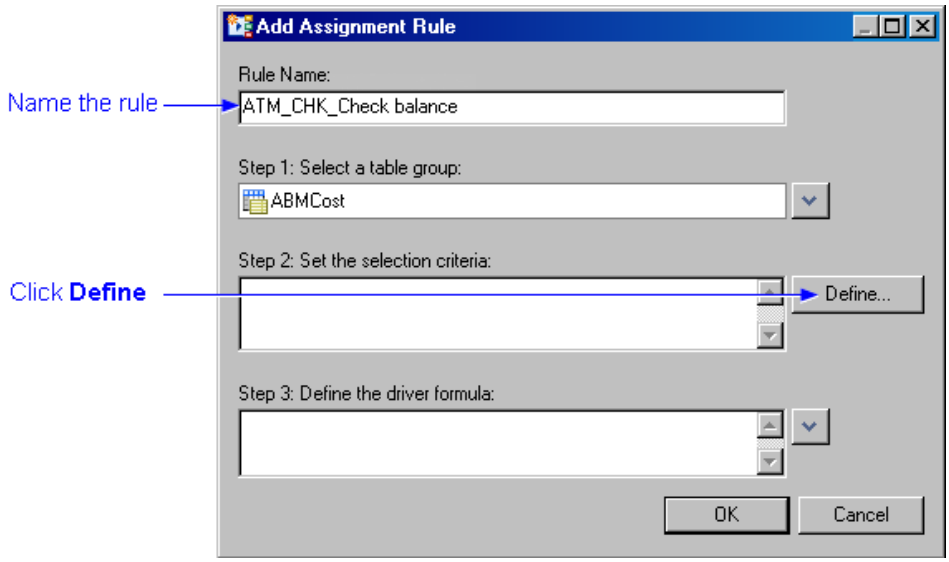
Define the First Rule

1. Click the **Rules** workspace.
2. Select **ABMCost** as the table group to use the rule.
3. Click **Add assignment rule**.



The Add Rule window opens.

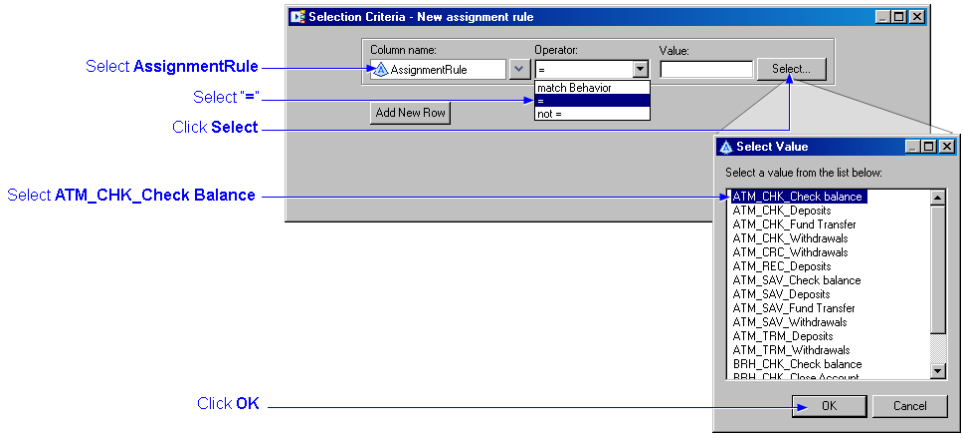
4. Name the rule **ATM_CHK_Check balance**.
5. Click **Define**.



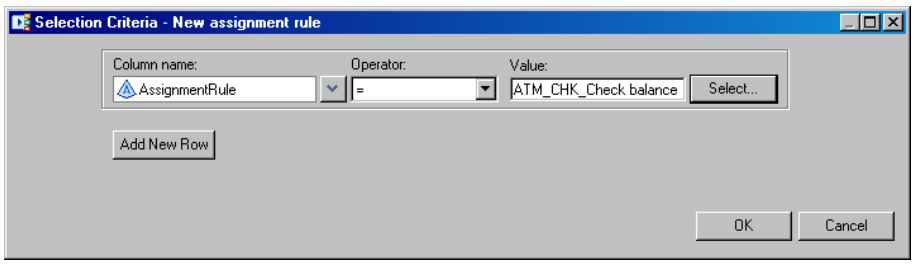
The Selection Criteria window opens.

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.)



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

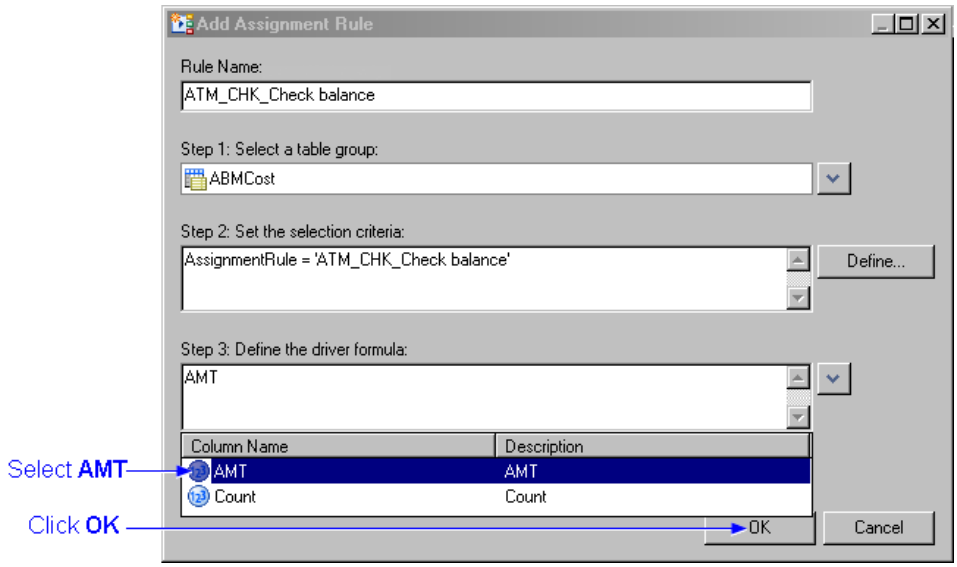


9. Click **OK**.

   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 chosen by the selection criterion, AssignmentRule = 'ATM_CHK_Check balance'.

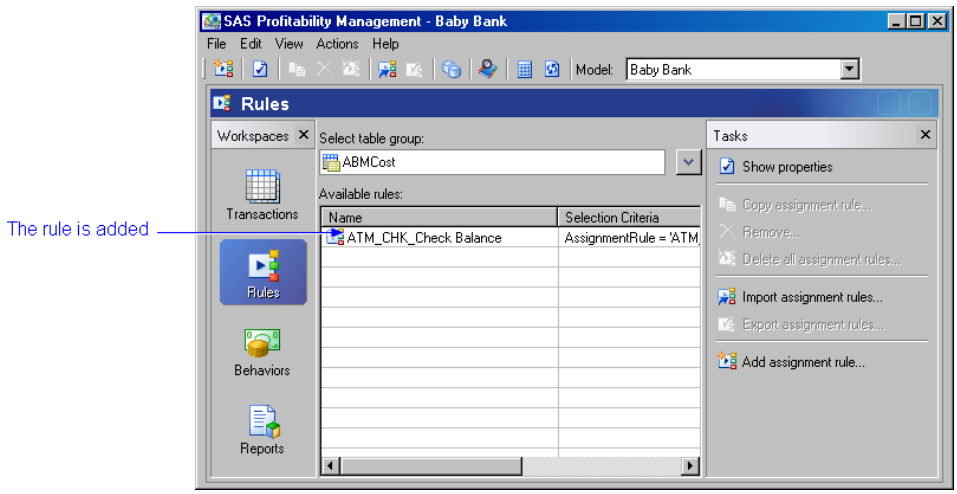


AMT represents the number of times a transaction (such as checking one's balance) occurred.

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.

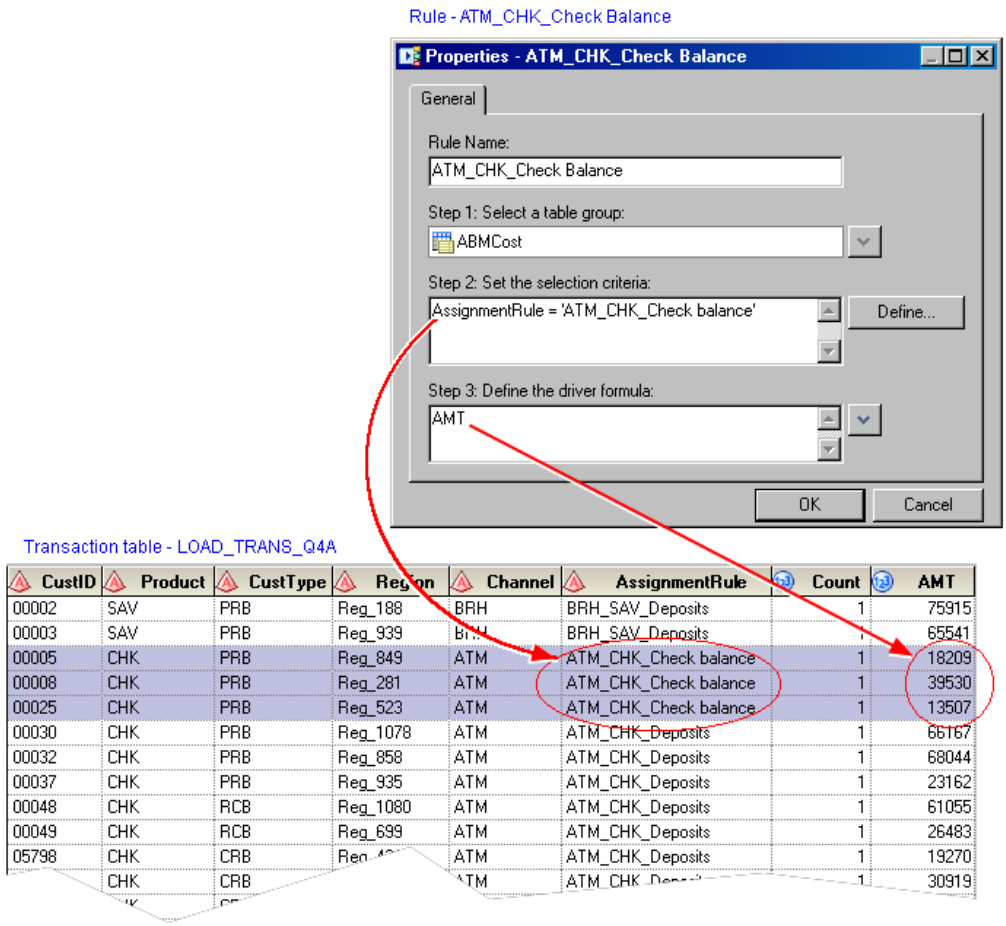


How Rules Work

An assignment rule associates a behavior with rows in a transaction table. 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 to include in a calculation
- includes a driver formula that specifies what quantity to include in the calculation.

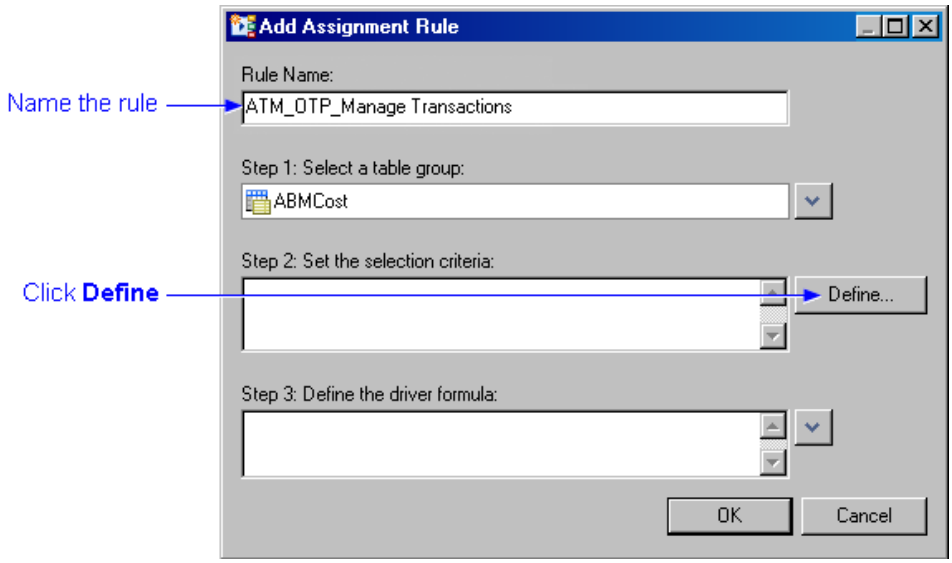
An assignment rule is implicitly associated with a single transaction table in a table group - namely, the transaction table whose period is specified in the behavior table row with which the rule is associated.



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.
4. Name the rule **ATM_OTP_Manage Transactions**.
5. Click **Define**.

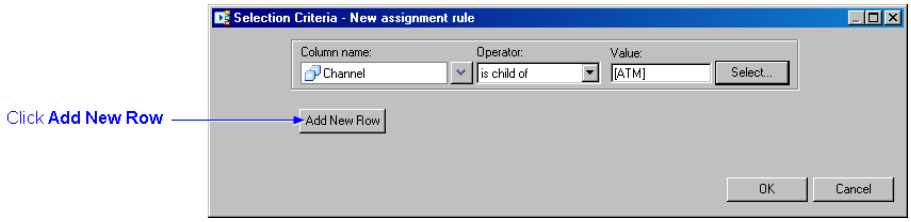


The Selection Criteria window opens.

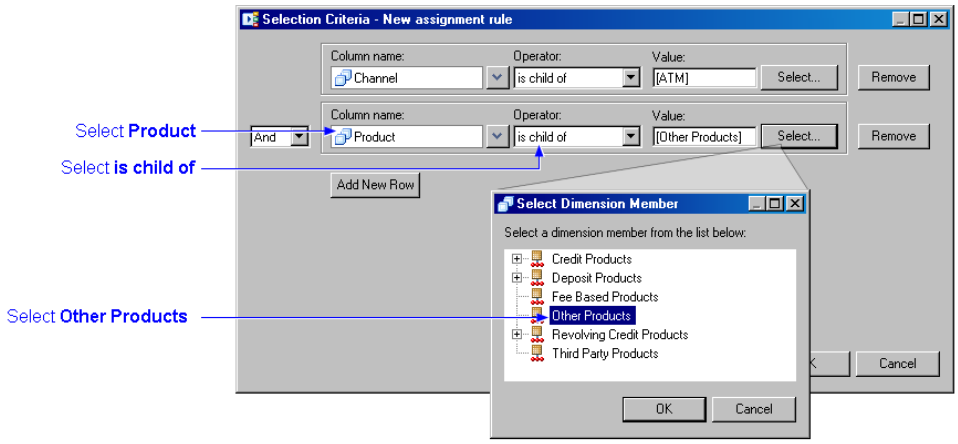
6. Select the dimension **Channel** from the drop-down list of column names to select.
7. Select **is child of** from the drop-down list of operators.

   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. For more information, see "Is Child Of" on page 133.
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.

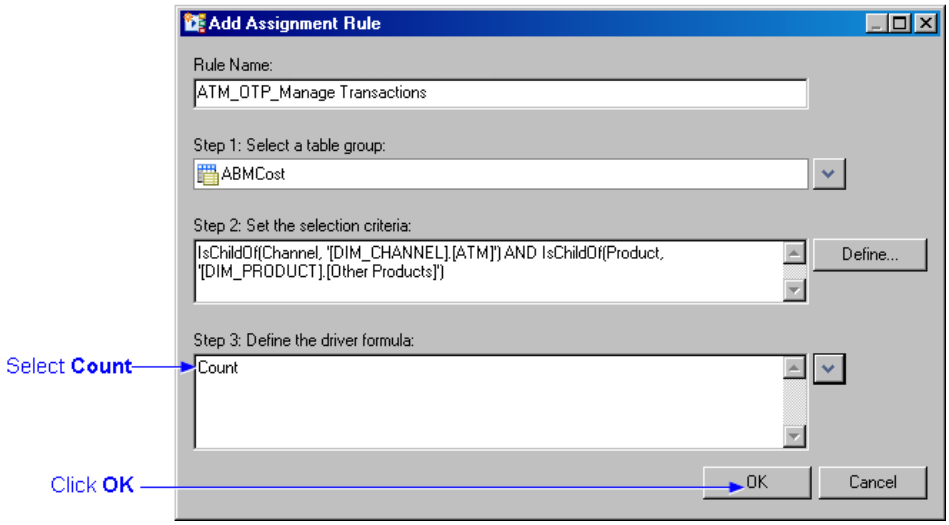


13. Click **OK**.

    You return to the Add Assignment Rule window.

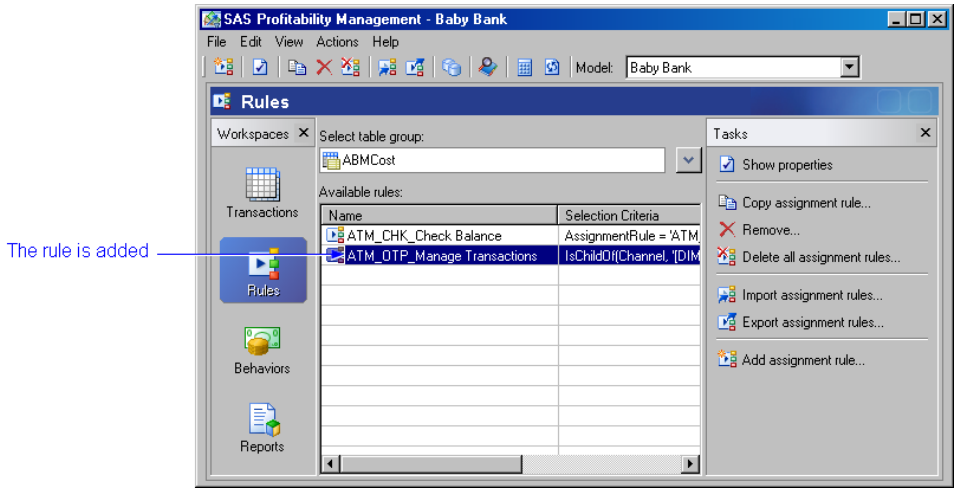
14. Select **Count** from the drop-down list of values for the driver formula.

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



15. Click **OK**.

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



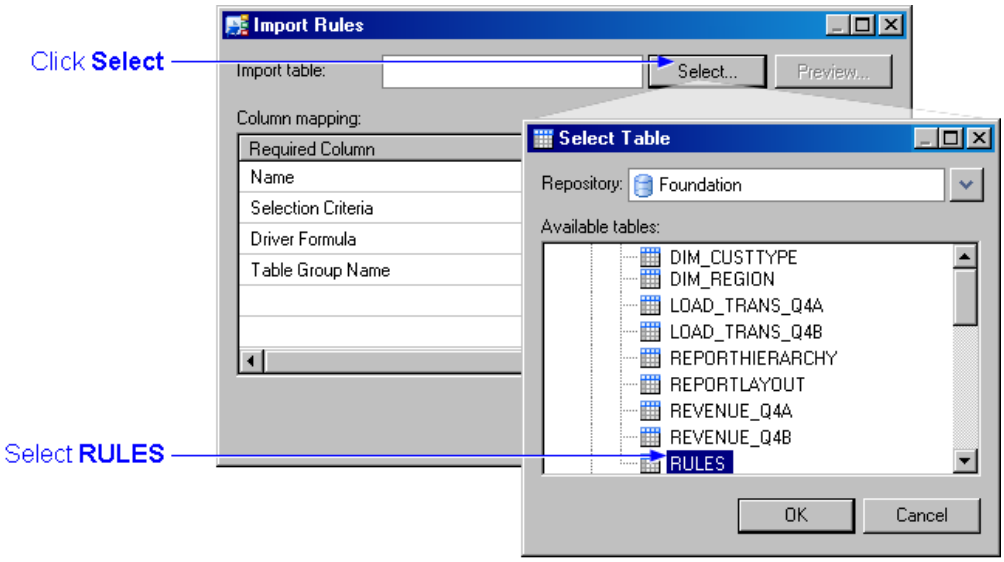
Import the Remaining Rules

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. Click **OK**.

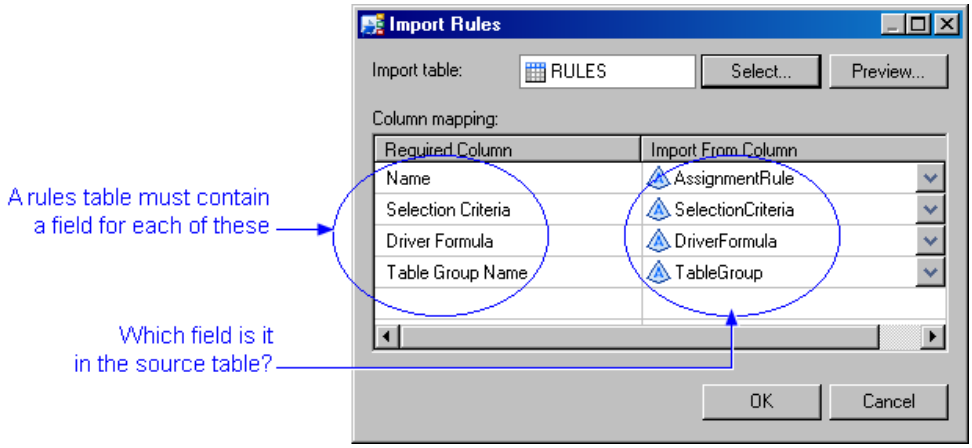
4. Identify fields in the rules table.

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

   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.

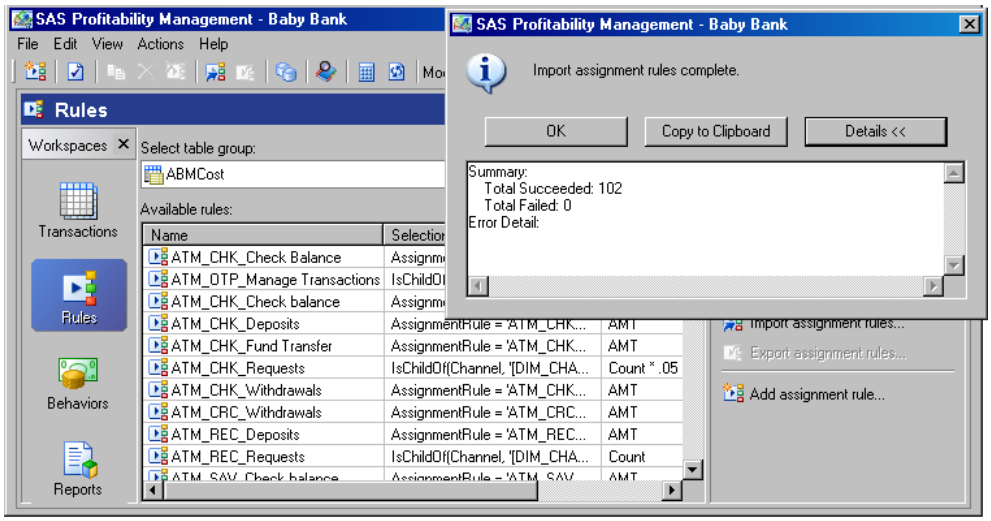
The sample rules table, RULES, contains these fields, so they are mapped automatically.



5. Click **OK**.

   The rules are imported.

   Note: It is considered an error if a rule already exists. It is not overwritten. A duplication error does not prevent other rules from being imported.



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.

TableGroup	AssignmentRule	SelectionCriteria	DriverFormula
ABMCost	ATM_CHK_Check balance	AssignmentRule = 'ATM_CHK_Check balance'	AMT
ABMCost	ATM_CHK_Deposits	AssignmentRule = 'ATM_CHK_Deposits'	AMT
ABMCost	ATM_CHK_Fund Transfer	AssignmentRule = 'ATM_CHK_Fund Transfer'	AMT
ABMCost	ATM_CHK_Requests	IsChildOf(Channel, '[DIM_CHANNEL].[ATM]') AND IsChildOf(Product, '[DIM_PRODUCT].[De...	Count * .05 +AMT*.01
ABMCost	ATM_CHK_Withdrawals	AssignmentRule = 'ATM_CHK_Withdrawals'	AMT
ABMCost	ATM_CRC_Withdrawals	AssignmentRule = 'ATM_CRC_Withdrawals'	AMT
ABMCost	ATM_REC_Deposits	AssignmentRule = 'ATM_REC_Deposits'	AMT
ABMCost	ATM_REC_Requests	IsChildOf(Channel, '[DIM_CHANNEL].[ATM]') AND IsChildOf(Product, '[DIM_PRODUCT].[De...	Count
ABMCost	ATM_SAV_Check balance	AssignmentRule = 'ATM_SAV_Check balance'	AMT
ABMCost	ATM_SAV_Deposits	AssignmentRule = 'ATM_SAV_Deposits'	AMT
ABMCost	ATM_SAV_Fund Transfer	AssignmentRule = 'ATM_SAV_Fund Transfer'	AMT
ABMCost	ATM_SAV_Requests	IsChildOf(Channel, '[DIM_CHANNEL].[ATM]') AND IsChildOf(Product, '[DIM_PRODUCT].[De...	Count
ABMCost	ATM_SAV_Withdrawals	AssignmentRule = 'ATM_SAV_Withdrawals'	AMT
ABMCost	ATM_TRM_Deposits	AssignmentRule = 'ATM_TRM_Deposits'	AMT
ABMCost	ATM_TRM_Requests	IsChildOf(Channel, '[DIM_CHANNEL].[ATM]') AND IsChildOf(Product, '[DIM_PRODUCT].[De...	Count
ABMCost	ATM_TRM_Withdrawals	AssignmentRule = 'ATM_TRM_Withdrawals'	AMT
ABMCost	BRH_CHK_Check balance	AssignmentRule = 'BRH_CHK_Cl...	AMT
[illegible]	[illegible]	[illegible]	AMT
CallCenter	CCT_SCR_Inquiry	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Product, '[DIM_PRODUC...	Inquiry
CallCenter	CCT_SCR_Requests	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Product, '[DIM_PRODUC...	Requests
CallCenter	CCT_TPP_Cross_Up Sell	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Product, '[DIM_PRODUC...	CrossSell
CallCenter	CCT_TRM_Complaints	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Product, '[DIM_PRODUC...	Complaints
CallCenter	CCT_TRM_Inquiry	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Product, '[DIM_PRODUC...	Inquiry
CallCenter	CCT_TRM_Requests	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Product, '[DIM_PRODUC...	Requests
CallCenter	CCT_UCR_Complaints	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Product, '[DIM_PRODUC...	Complaints
CallCenter	CCT_UCR_Cross_Up Sell	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Product, '[DIM_PRODUC...	CrossSell
CallCenter	CCT_UCR_Inquiry	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Product, '[DIM_PRODUC...	Inquiry
CallCenter	CCT_UCR_Requests	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Product, '[DIM_PRODUC...	Requests
Revenue	ATM Fees	ID = '12002'	AMT
Revenue	Certificates of Deposit Pay...	ID = '11002'	AMT
Revenue	Charge For Funds	ID = '13002'	AMT
Revenue	Checking Account Fees	ID = '12004'	AMT
Revenue	Credit Card Fees	ID = '12001'	AMT
Revenue	Credit Card interest Income	ID = '10001'	AMT
Revenue	Credit for Funds	ID = '13001'	AMT
Revenue	Investment Account Fees	ID = '12003'	AMT
Revenue	Investment Securities Pay...	ID = '11003'	AMT
Revenue	Loan Interest Income	ID = '10002'	AMT
Revenue	Mortgages Income	ID = '10003'	AMT
Revenue	Provision For Losses	ID = '14001'	AMT
Revenue	Savings Interest Payments	ID = '11001'	AMT

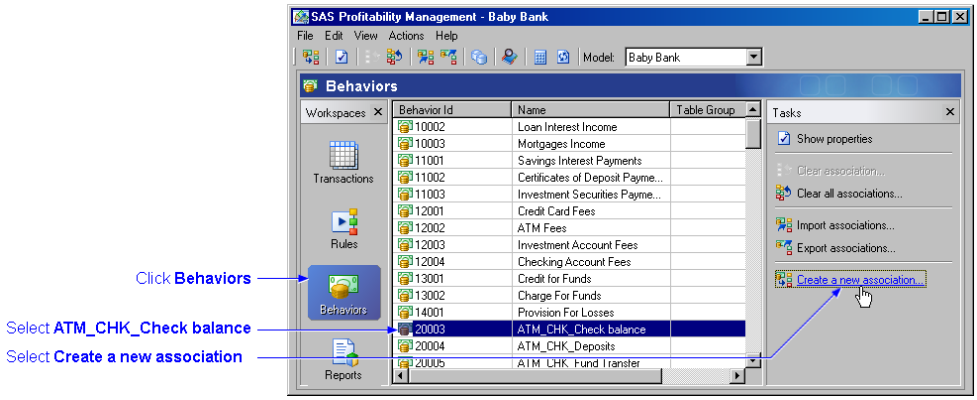
Chapter 9

Associate Behaviors With Rules

Associate Behaviors with Rules

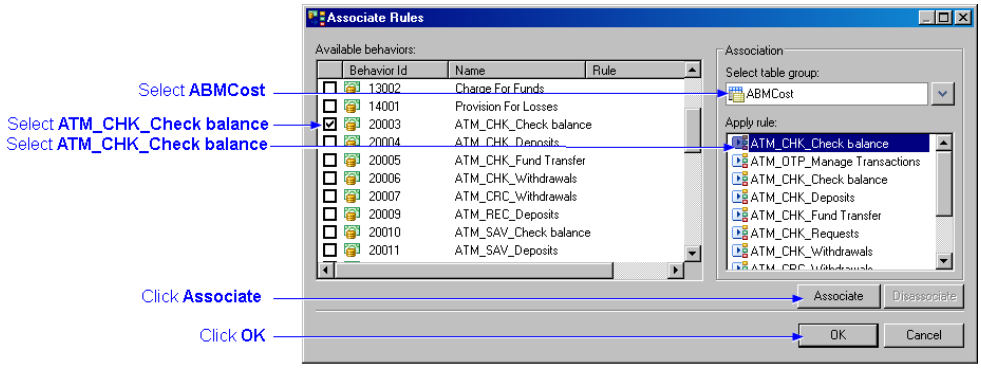
Each behavior must be associated with only one rule. Multiple behaviors can be associated with the same rule.

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

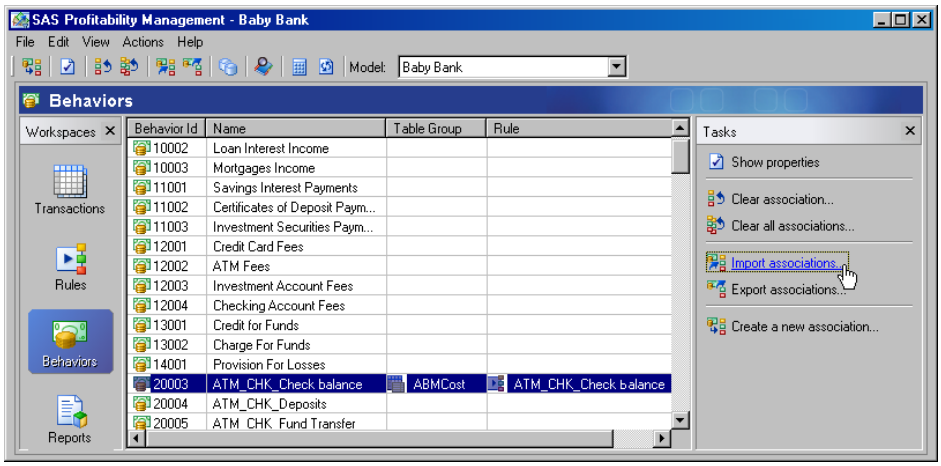


The Associate Rules window opens.

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

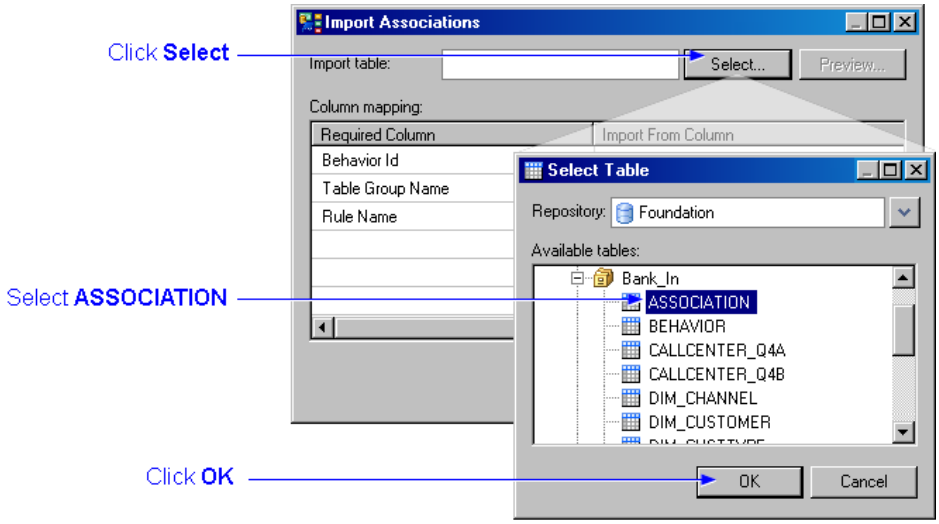
1. Click **Import associations**.

   The Import Associations window opens.

2. Click **Select**.

   The Select Table dialog opens.

3. Select the **ASSOCIATION** table to import.

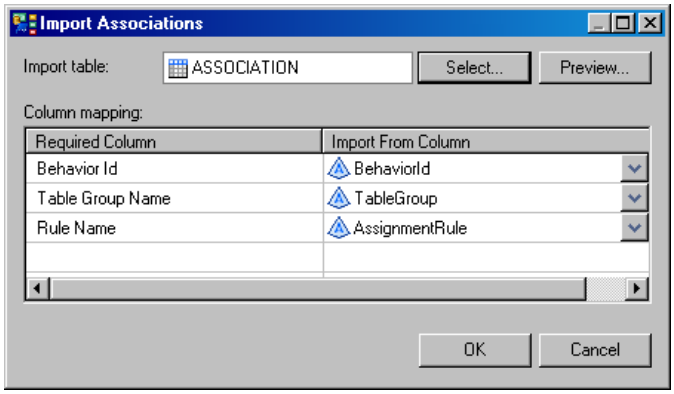


4. Click **OK**.
5. Identify fields in the ASSOCIATIONS table.

   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.

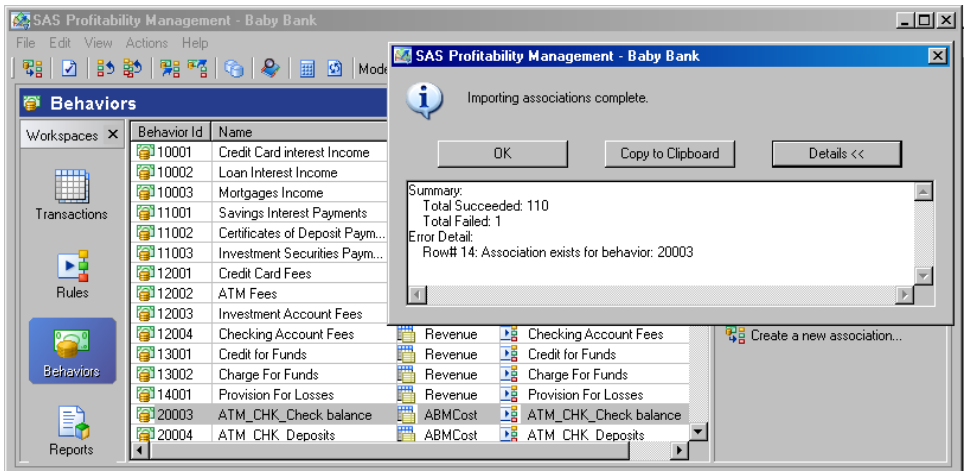
   The sample table, ASSOCIATIONS, contains these fields, so they are mapped automatically.



6. Click **OK**.

   Each behavior is associated with a rule.

   Note: If a behavior is already associated with a rule, an error message is issued and the association is not replaced. The error does not prevent the rest of the associations from being imported.


SAS Profitability Management - Baby Bank
File Edit View Actions Help
Behaviors
Workspaces
Transactions
Rules
Behaviors
Reports
Behavior Id
Name
10001 Credit Card interest Income
10002 Loan Interest Income
10003 Mortgages Income
11001 Savings Interest Payments
11002 Certificates of Deposit Paym...
11003 Investment Securities Paym...
12001 Credit Card Fees
12002 ATM Fees
12003 Investment Account Fees
12004 Checking Account Fees
13001 Credit for Funds
13002 Charge For Funds
14001 Provision For Losses
20003 ATM_CHK_Check balance
20004 ATM_CHK_Deposits
Revenue Checking Account Fees
Revenue Credit for Funds
Revenue Charge For Funds
Revenue Provision For Losses
ABMCost ATM_CHK_Check balance
ABMCost ATM_CHK_Deposits
Create a new association...
SAS Profitability Management - Baby Bank
Importing associations complete.
OK
Copy to Clipboard
Details <<
Summary:
Total Succeeded: 110
Total Failed: 1
Error Detail:
Row# 14: Association exists for behavior: 20003

Chapter 10

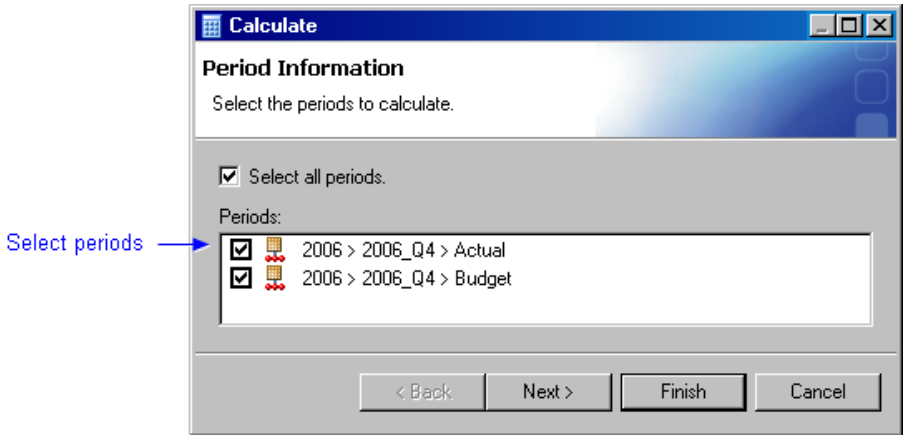
Calculate the Model

Calculate the Model

1. Select **Actions > Calculate Model**.

   The Calculate window opens.

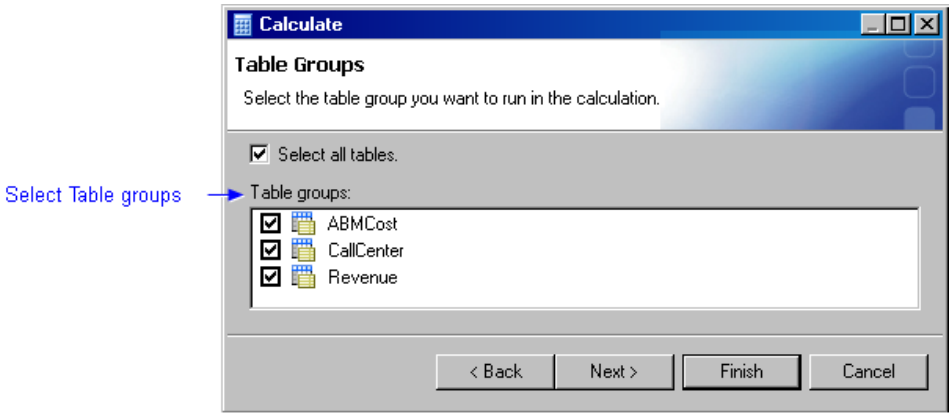
2. Select both periods to be calculated for the model, and then click **Next**.



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 59) 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**.



The Model Calculate Options window opens.

4. Select 1 as the number of concurrent sessions.

   SAS Profitability Management can start concurrent sessions to distribute the workload across multiple CPUs. One transaction is allocated to each session.

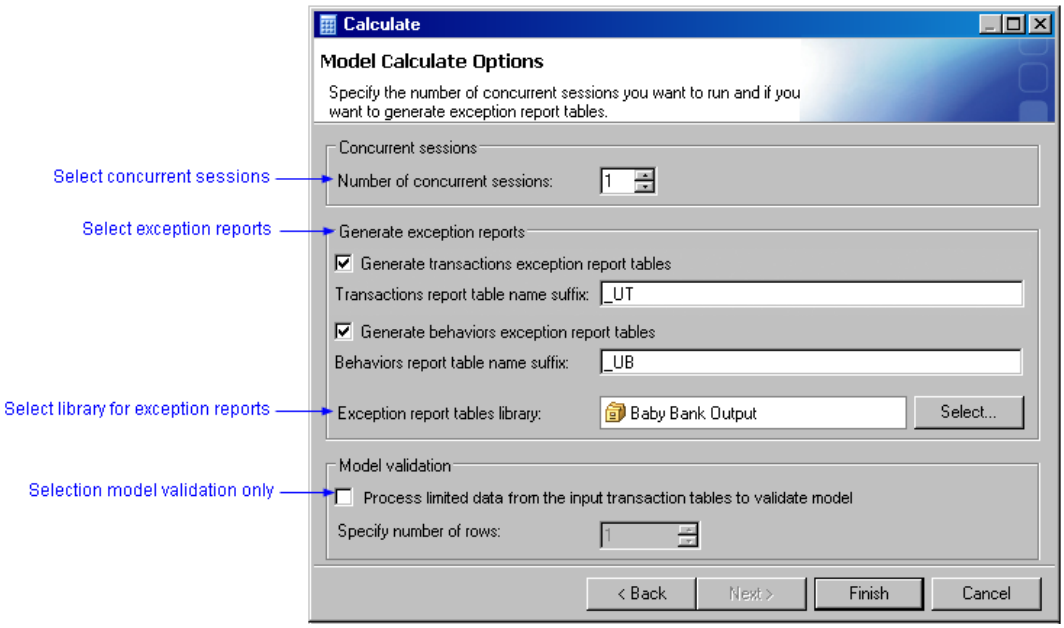
5. Select exception reports to be generated and specify their suffix.

   - A transaction exception report flags transaction rows that did not receive an assignment during the calculation.
   - A behaviors exception report flags behaviors that were not assigned to any transaction during the calculation.

6. If you want to generate exception reports, then specify the library to receive them.

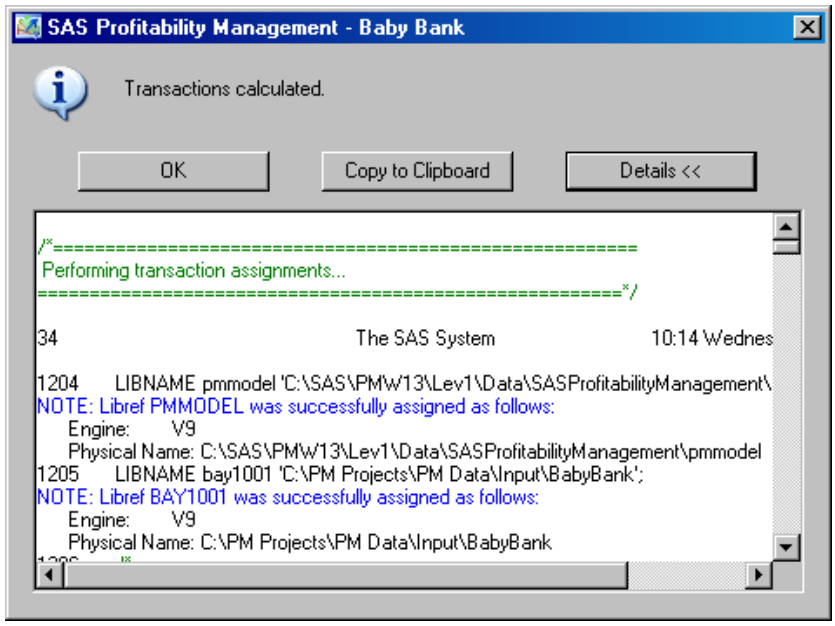
7. If you do not want a complete calculation but want only to validate the model, then you can select Process limited data from the input transaction tables to validate model and specify the number of rows to calculate.

   For now, uncheck this option because we want to do a complete calculation.



8. Click **Finish**.

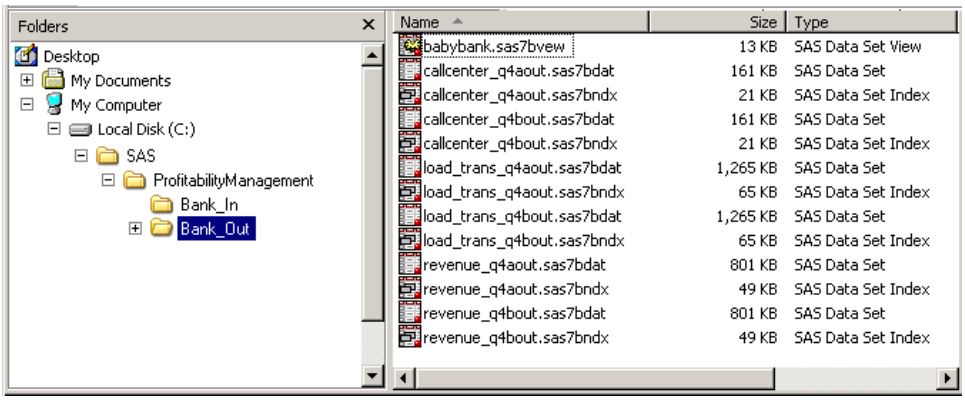
   When the calculation completes, you can view the SAS log.



9. Click **OK** to close the results window.
10. 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.sas7bvew**. 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

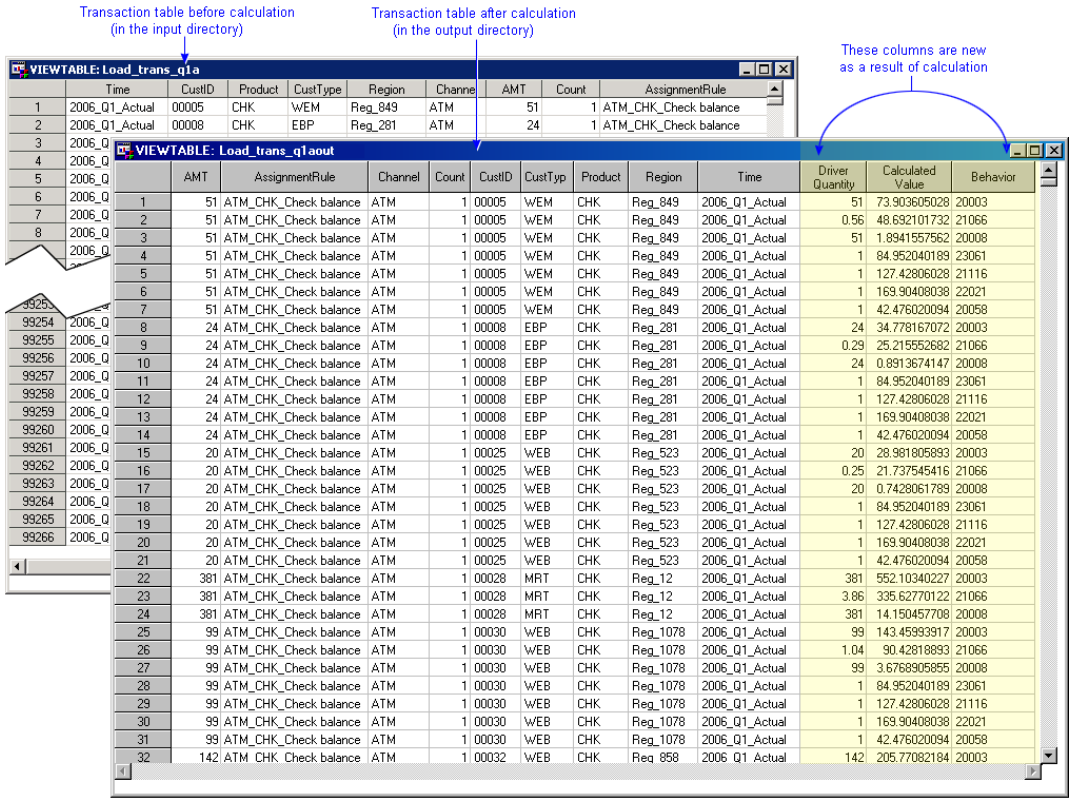
Overview

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.

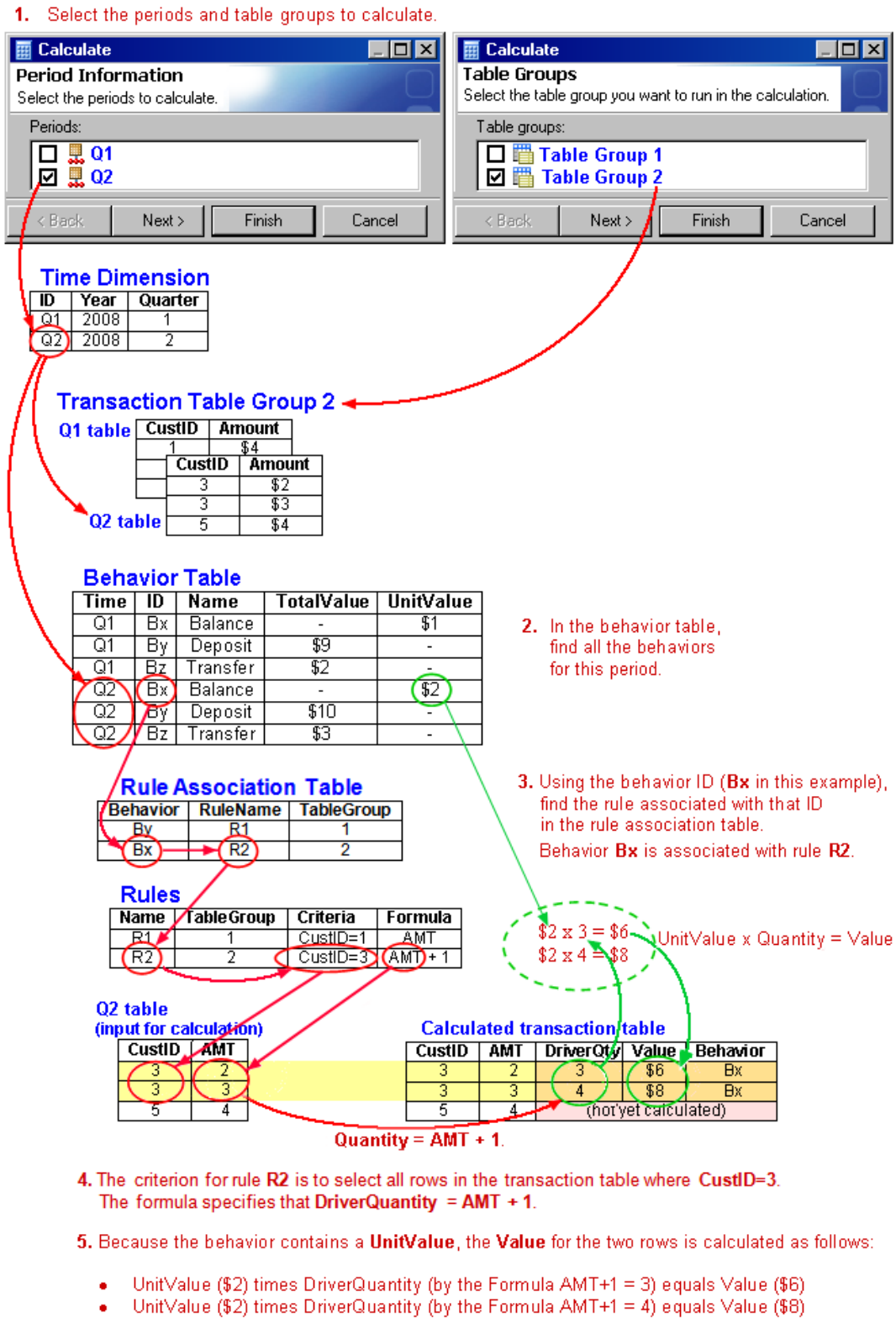


Calculation Using a Unit Value

When a behavior table row contains a unit value, the driver formula calculates the number of units that are involved in each transaction chosen 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:

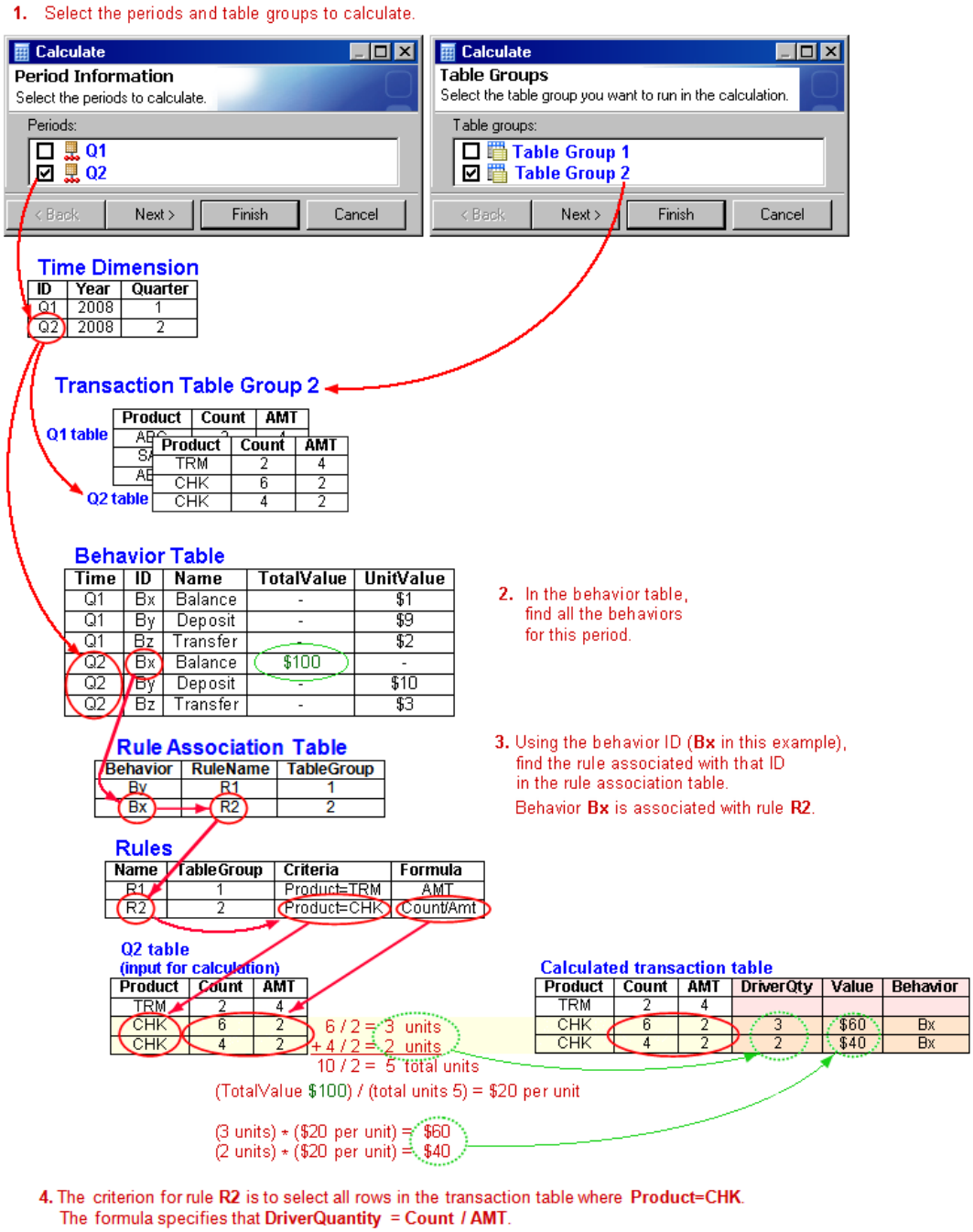


Calculation Using a Total Value

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

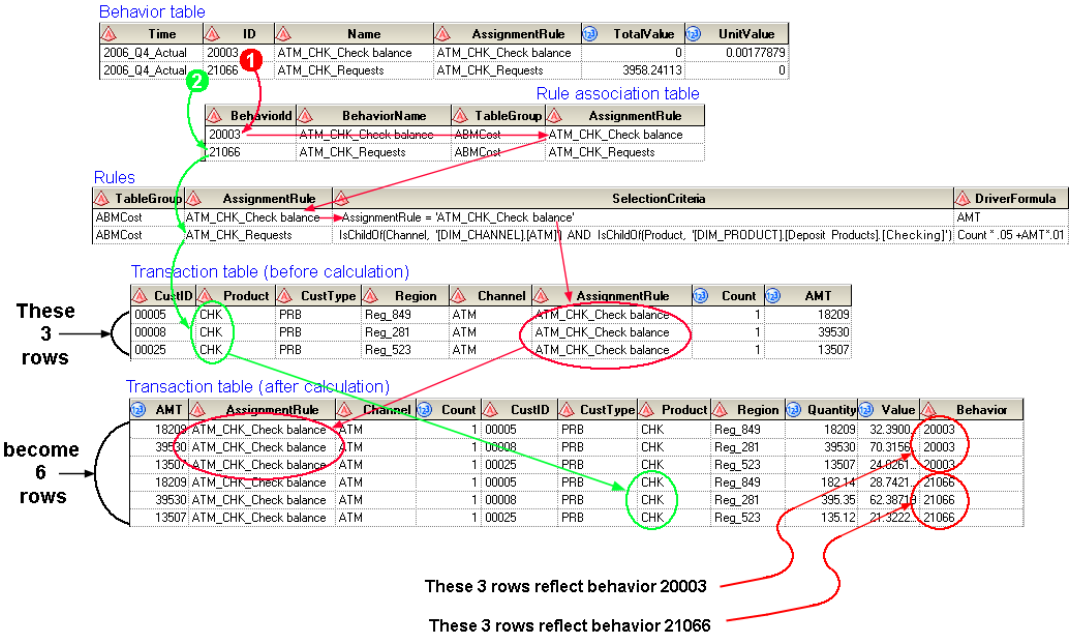
1. The **total number of units** for all transactions (chosen by the selection criteria) is calculated by adding the number of units (as determined by the driver formula) for all the transactions chosen by the selection criteria.
2. The **cost per unit** is calculated by dividing the total value (in 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:



Output Tables Grow in Length

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 this 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) selecting 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.


Behavior table
Time
ID
Name
AssignmentRule
TotalValue
UnitValue
2006_Q4_Actual
20003
ATM_CHK_Check balance
ATM_CHK_Check balance
0
0.00177879
2006_Q4_Actual
21066
ATM_CHK_Requests
ATM_CHK_Requests
3958.24113
0
Rule association table
BehaviorId
BehaviorName
TableGroup
AssignmentRule
20003
ATM_CHK_Check balance
ABMCost
ATM_CHK_Check balance
21066
ATM_CHK_Requests
ABMCost
ATM_CHK_Requests
Rules
TableGroup
AssignmentRule
SelectionCriteria
DriverFormula
ABMCost
ATM_CHK_Check balance
AssignmentRule = 'ATM_CHK_Check balance'
AMT
ABMCost
ATM_CHK_Requests
IsChildOf(Channel, "[DIM_CHANNEL].[ATM]") AND IsChildOf(Product, "[DIM_PRODUCT].[Deposit Products].[Checking]")
Count * .05 +AMT*.01
Transaction table (before calculation)
These 3 rows
CustID
Product
CustType
Region
Channel
AssignmentRule
Count
AMT
00005
CHK
PRB
Reg_849
ATM
ATM_CHK_Check balance
1
18209
00008
CHK
PRB
Reg_281
ATM
ATM_CHK_Check balance
1
39530
00025
CHK
PRB
Reg_523
ATM
ATM_CHK_Check balance
1
13507
Transaction table (after calculation)
become 6 rows
AMT
AssignmentRule
Channel
Count
CustID
CustType
Product
Region
Quantity
Value
Behavior
18209
ATM_CHK_Check balance
ATM
1
00005
PRB
CHK
Reg_849
18209
32.3900...
20003
39530
ATM_CHK_Check balance
ATM
1
00008
PRB
CHK
Reg_281
39530
70.3156...
20003
13507
ATM_CHK_Check balance
ATM
1
00025
PRB
CHK
Reg_523
13507
24.0261...
20003
18209
ATM_CHK_Check balance
ATM
1
00005
PRB
CHK
Reg_849
182.14
28.7421...
21066
39530
ATM_CHK_Check balance
ATM
1
00008
PRB
CHK
Reg_281
395.35
62.38719...
21066
13507
ATM_CHK_Check balance
ATM
1
00025
PRB
CHK
Reg_523
135.12
21.3222...
21066
These 3 rows reflect behavior 20003
These 3 rows reflect behavior 21066

Chapter 11

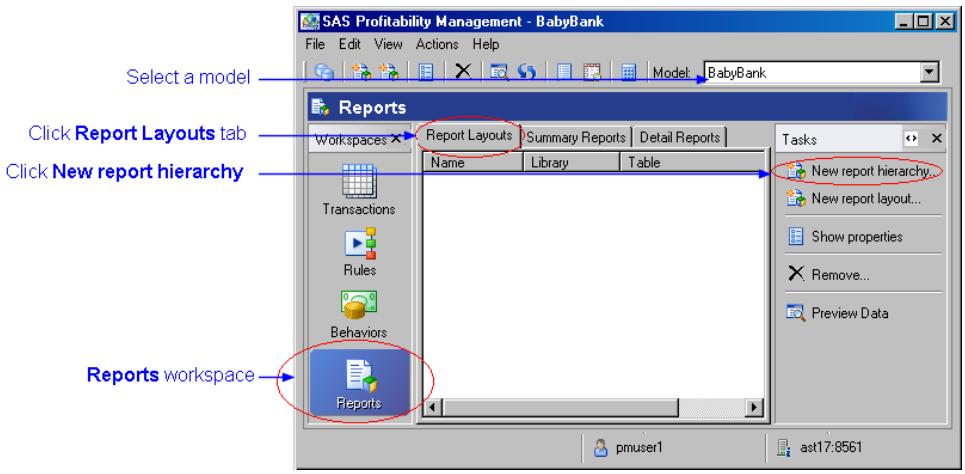
Define a Report Hierarchy

Define a Report Hierarchy

The report hierarchy table defines the dimension hierarchy for drilling into a profit-and-loss report. A model can have multiple report hierarchies. For more information on report hierarchies, see "Report Hierarchy" on page 136

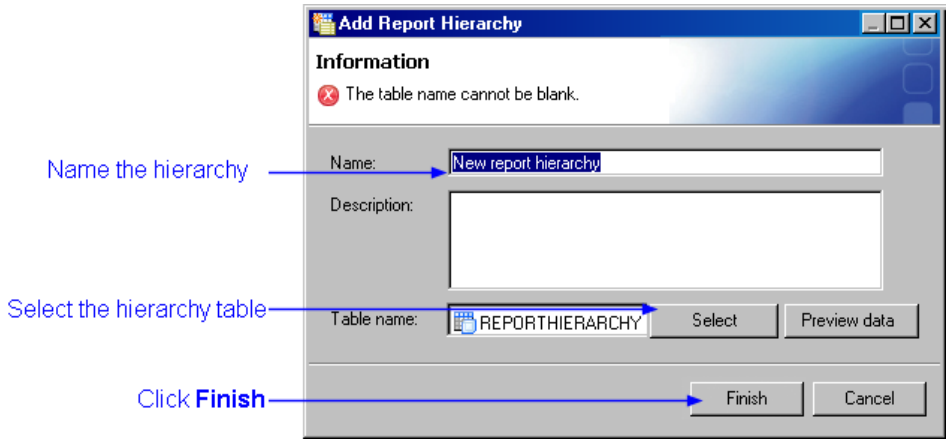
To define a report hierarchy:

1. Select a model.
2. Click the Report Layouts tab in the Reports workspace.
3. Click New report hierarchy (or select **File** ⇨ **New Report Hierarchy**.

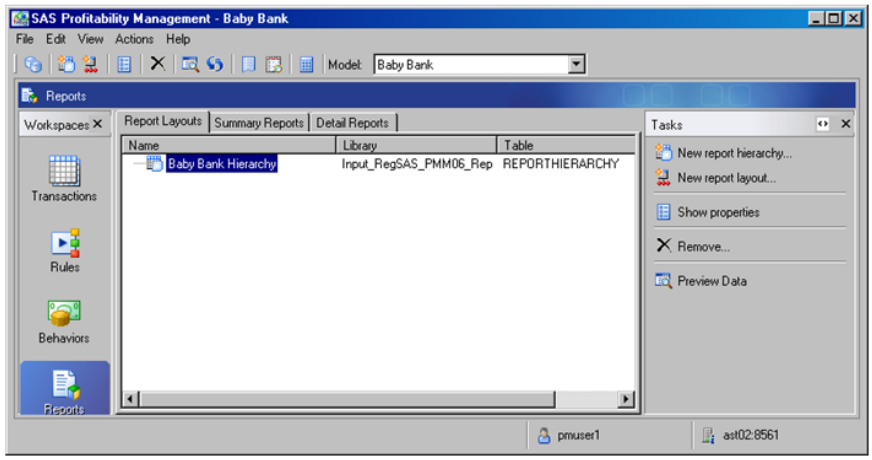


The Add Report Hierarchy window opens.

4. Name the hierachy **Baby Bank Hierarchy**.
5. Click Select, and select the **REPORTHIERARCHY** table.
6. Click **Finish**.



The report hierarchy is added to the model.



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) do not exist outside the report hierarchy table itself.

BEHAVIOR

Time	ID	Name	AssignmentRule	TotalValue	UnitValue
2006_Q4_Actual	10001	Credit Card interest Income	Credit Card interest Income	0	1
2006_Q4_Actual	10002	Loan Interest Income	Loan Interest Income	0	1
2006_Q4_Actual	10003	Mortgages Income	Mortgages Income	0	1
2006_Q4_Actual	11001	Savings Interest Payments	Savings Interest Payments	0	1
2006_Q4_Actual	11002	Certificates of Deposit Payments	Certificates of Deposit Pay...	0	1
2006_Q4_Actual	11003	Investment Securities Payments	Investment Securities Paym...	0	1
2006_Q4_Actual	12001	Credit Card Fees			
2006_Q4_Actual	12002	ATM Fees			
2006_Q4_Actual	12003	Investment Account Fees			
2006_Q4_Actual	12004	Checking Account Fees			
2006_Q4_Actual	13001	Credit for Funds			
2006_Q4_Actual	13002	Charge For Funds			
2006_Q4_Actual	14001	Provision For Losses			
2006_Q4_Budget	10001	Credit Card interest Income			
2006_Q4_Actual	23001	BRH_CHK_Open Account			
2006_Q4_Actual	23002	BRH_CRC_Open Account			
2006_Q4_Actual	23003	BRH_FBP_Open Account			
2006_Q4_Actual	23004	BRH_OVD_Open Account			
2006_Q4_Actual	23005	BRH_REC_Open Account			
2006_Q4_Actual	23006	BRH_SAV_Open Account			
2006_Q4_Actual	23007	BRH_SCR_Open Account			
2006_Q4_Actual	23008	BRH_TRM_Open Account			
2006_Q4_Actual	23009	BRH_UCR_Open Account			
2006_Q4_Actual	23017	BRH_CHK_Close Account			
2006_Q4_Actual	23018	BRH_CRC_Close Account			
2006_Q4_Actual	23019	BRH_REC_Close Account			
2006_Q4_Actual	23020	BRH_SAV_Close Account			
2006_Q4_Actual	23021	BRH_SCR_Close Account			
2006_Q4_Actual	23022	BRH_TRM_Close Account			
2006_Q4_Actual	23023	BRH_UCR_Close Account			
2006_Q4_Budget	23061	None_None_None			

REPORTHIERARCHY

ID	L1_Profit	L2_Profit	L3_Profit	L4_Profit	L5_Profit
10001	Interest Income	Credit Card interest Income			
10002	Interest Income	Loan Interest Income			
10003	Interest Income	Mortgages Income			
11001	Interest Expense	Savings Interest Payments			
11002	Interest Expense	Certificates of Deposit Payments			
11003	Interest Expense	Investment Securities Payments			
12001	Fees	Credit Card Fees			
12002	Fees	ATM Fees			
12003	Fees	Investment Account Fees			
12004	Fees	Checking Account Fees			
13001	Funds	Credit for Funds			
13002	Funds	Charge For Funds			
14001	Provision For Losses				
23000	Servicing Effort		(Channel)	(Product)	
23001	Servicing Effort	Cost to Acquire	BRH	CHK	BRH_CHK_Open Account
23002	Servicing Effort	Cost to Acquire	BRH	CRC	BRH_CRC_Open Account
23003	Servicing Effort	Cost to Acquire	BRH	FBP	BRH_FBP_Open Account
23004	Servicing Effort	Cost to Acquire	BRH	OVD	BRH_OVD_Open Account
23005	Servicing Effort	Cost to Acquire	BRH	REC	BRH_REC_Open Account
23006	Servicing Effort	Cost to Acquire	BRH	SAV	BRH_SAV_Open Account
23007	Servicing Effort	Cost to Acquire	BRH	SCR	BRH_SCR_Open Account
23008	Servicing Effort	Cost to Acquire	BRH	TRM	BRH_TRM_Open Account
23009	Servicing Effort	Cost to Acquire	BRH	UCR	BRH_UCR_Open Account
23017	Servicing Effort	Cost to Close	BRH	CHK	BRH_CHK_Close Account
23018	Servicing Effort	Cost to Close	BRH	CRC	BRH_CRC_Close Account
23019	Servicing Effort	Cost to Close	BRH	REC	BRH_REC_Close Account
23020	Servicing Effort	Cost to Close	BRH	SAV	BRH_SAV_Close Account
23021	Servicing Effort	Cost to Close	BRH	SCR	BRH_SCR_Close Account
23022	Servicing Effort	Cost to Close	BRH	TRM	BRH_TRM_Close Account
23023	Servicing Effort	Cost to Close	BRH	UCR	BRH_UCR_Close Account
23061	Servicing Effort	Cost to Sustain Business	None	None	None_None_None

Chapter 12

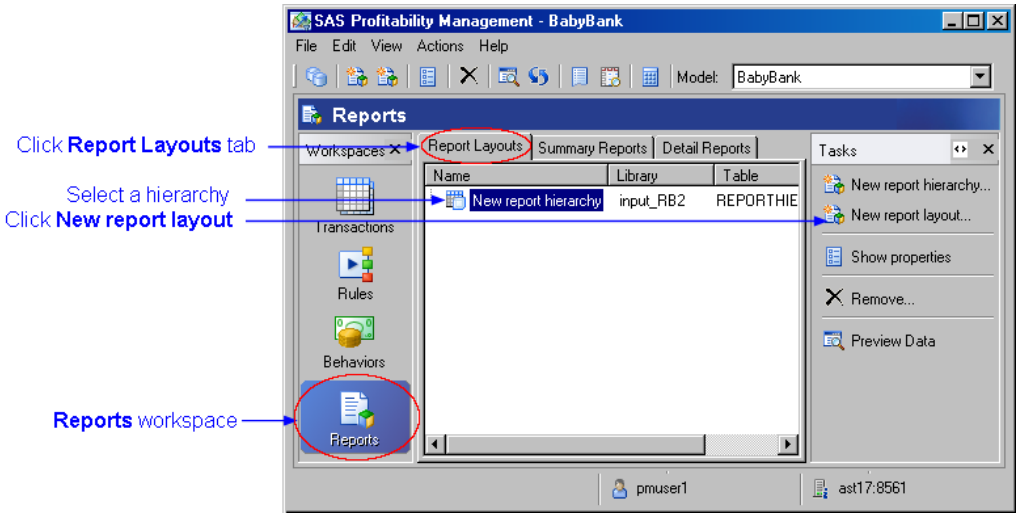
Define a Report Layout

Define a Report Layout

A report layout table defines the calculation formula for the profit-and-loss report, and it specifies what levels from the report hierarchy are to appear on the initial display of a report. A report hierarchy can have multiple report layouts so that you can create different reports from the same data. For more information on report layouts, see "Report Layout" on page 140.

To define a report layout:

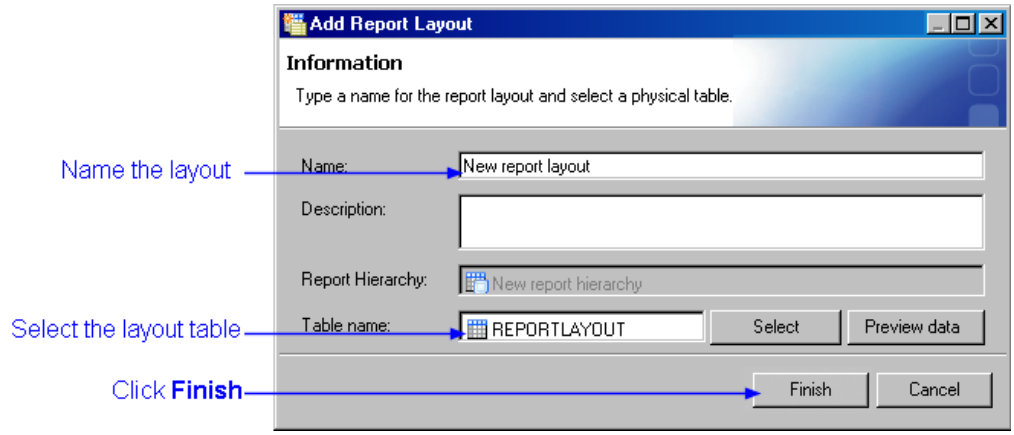
1. Select a model.
2. Click the Report Layouts tab in the Reports workspace.
3. Select the **Baby Bank Hierarchy**. Each hierarchy can have one or more layouts.
4. Click **New report layout** (or select **File** ⇨ **New Report Layout**).



The Add Report Layout window opens.

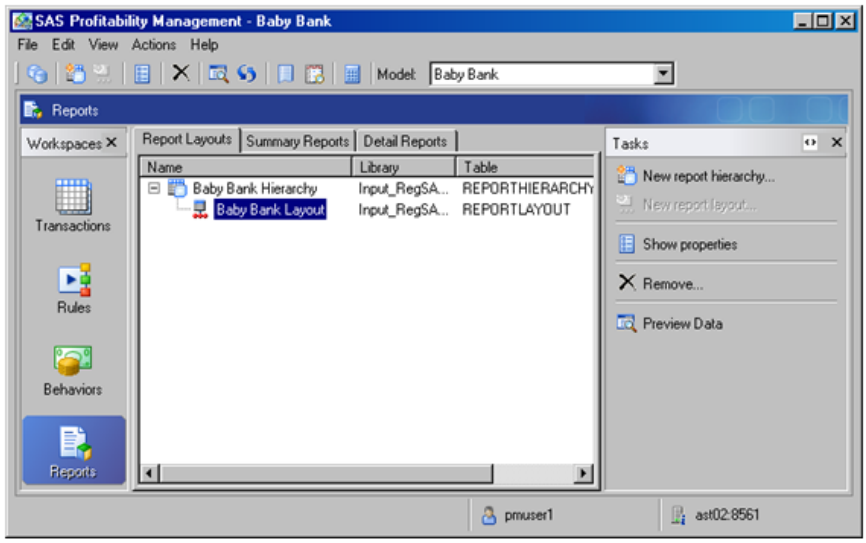
5. Name the layout **Baby Bank Layout**.
6. Click Select, and select the **REPORTLAYOUT** table to define the layout.

Note: The report layout must be compatible with its report hierarchy.



7. Click **Finish**.

   The report layout is added to the model.



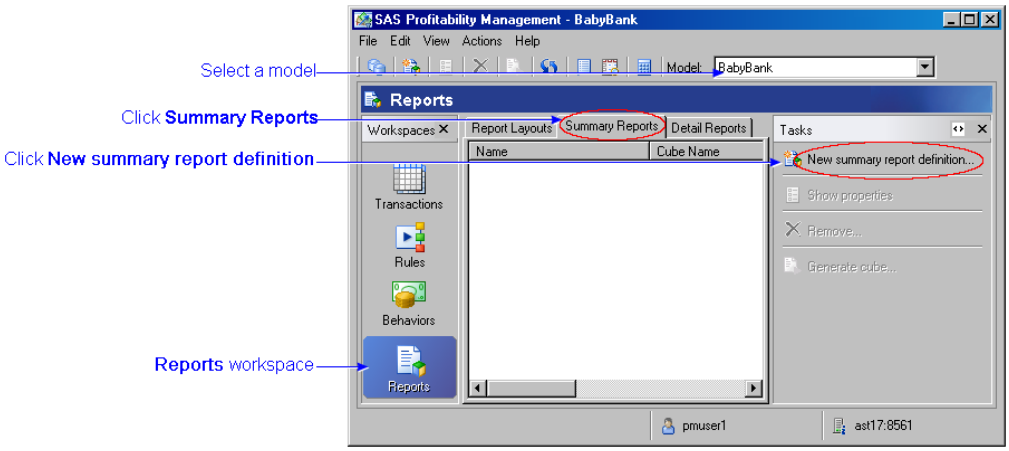
Chapter 13

Prepare Reports

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.

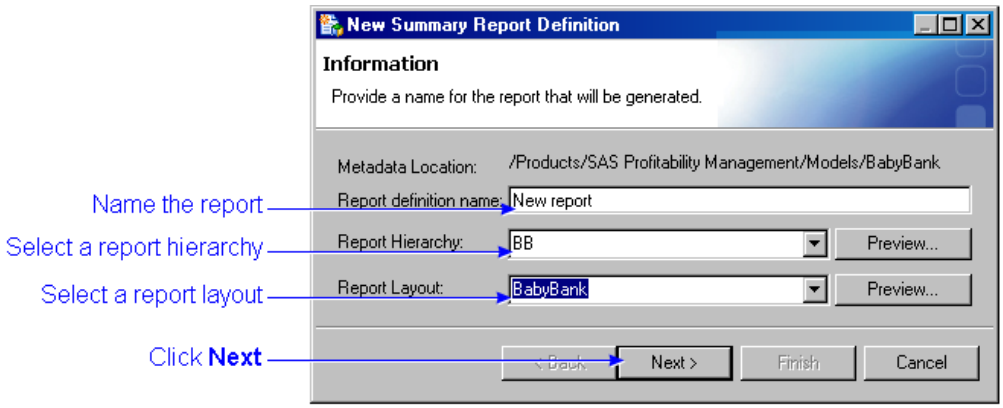
1. Select the **Reports** tab (making sure that the Baby Bank model is selected).
2. Click the Summary Reports tab.
3. Click **New summary report definition**.



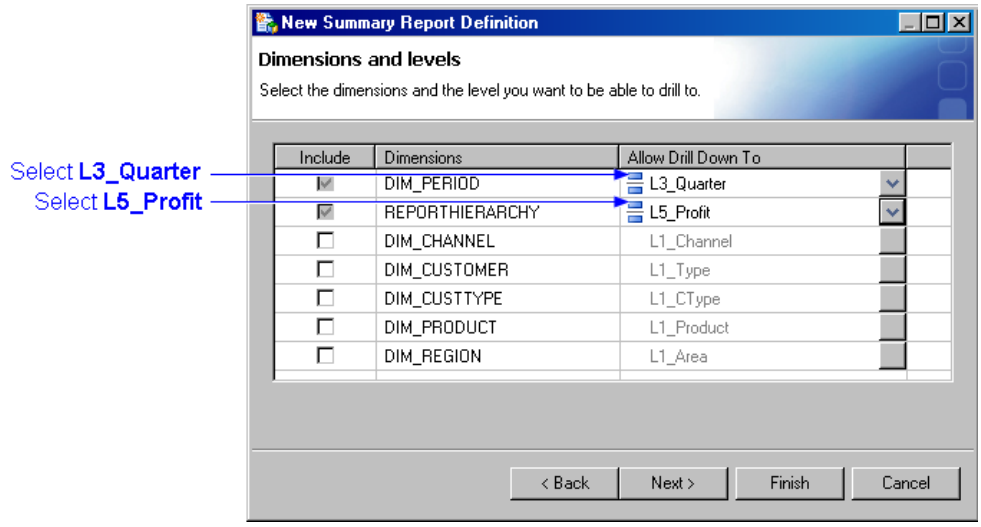
The New Summary Report Definition window opens.

4. Name the report, for example **Baby Bank Summary Report**.
5. Select the report hierarchy, **Baby Bank Hierarchy**, to determine the drill-down order of the report.
6. Select the report layout, **Baby Bank Layout**, to determine the appearance of the report.

   Note: After you select a report hierarchy, the drop-down list of report layouts lists only those layouts belonging to the selected report hierarchy.
7. Click **Next**.



8. 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.

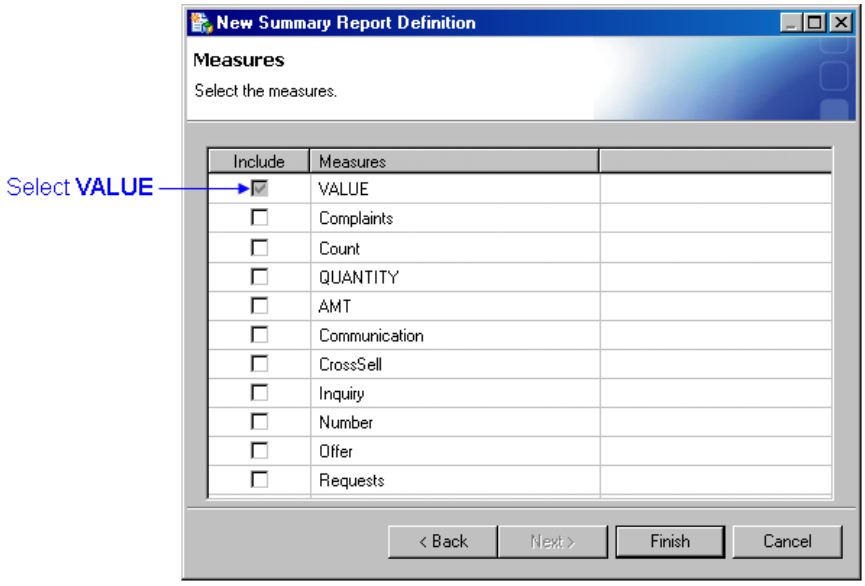
   All decisions on the number of dimensions and the depth of drill-down in any selected dimension have a dramatic impact on the reporting performance. It is wise to limit summary reports to specific areas where business decisions will be made.

   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.

9. Click **Next.**

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

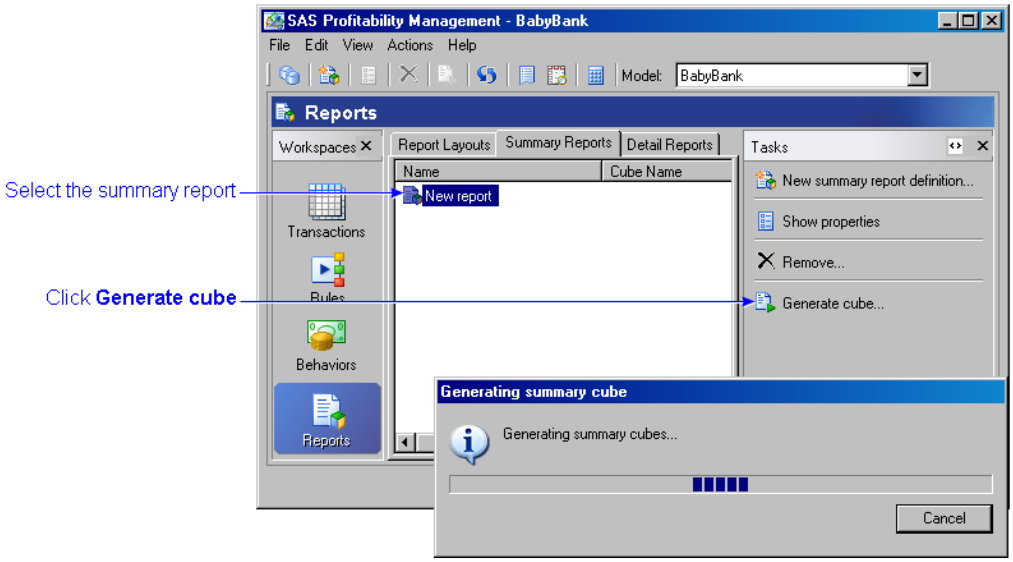
    The VALUE property is the default. It is the calculated cost for a transaction.



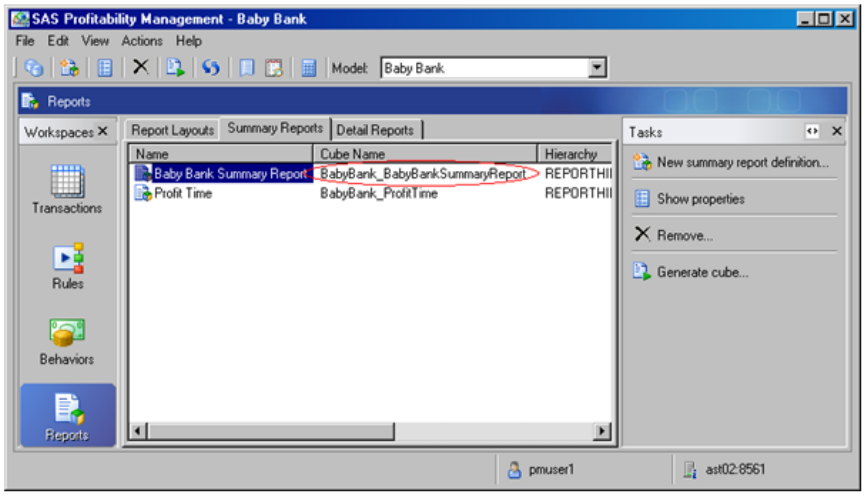
All numeric properties are available for inclusion in a summary report. The following are available:

- 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 thebehavior tablerow that is accessed by an assignmentrulecontains a unit value or a total value, as we shall see in greater length in “Calculation - a Conceptual View” on page 80.

11. Click **Finish**. The new report is added to the list of summary reports.
12. Select the new report in the list of summary reports, and then click **Generate cube**.



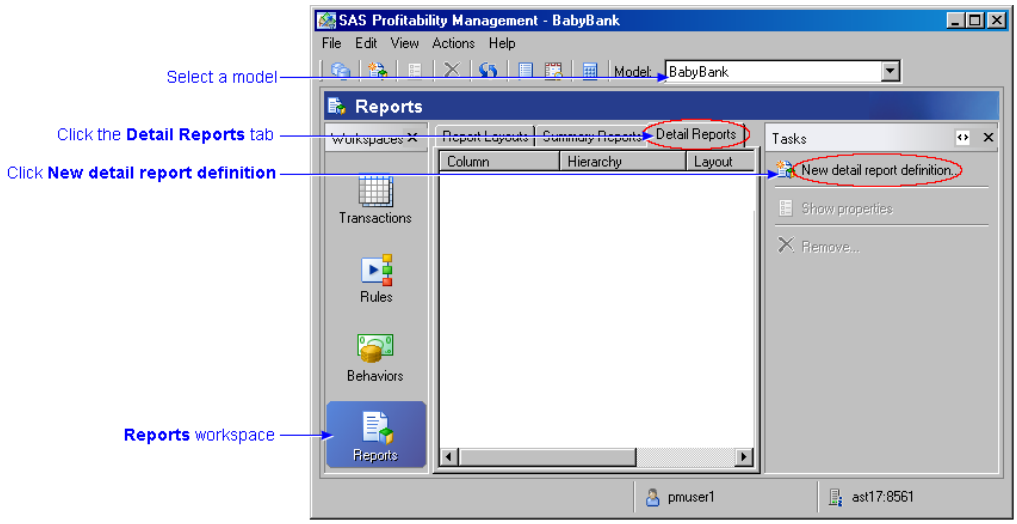
The cube is generated.



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.

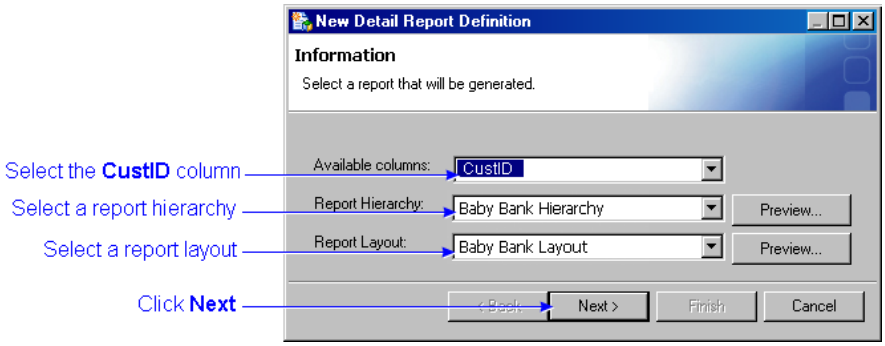
1. Select the **Reports** tab (and make sure that the Baby Bank model is selected).
2. Click the **Detail Reports** tab.
3. Click **New detail report definition**.



The New Detail Report Definition window opens.

4. Select **CustID** as the column to be featured in the detail report.
5. Select the report hierarchy, **Baby Bank Hierarchy**, containing the column.
6. Select the report layout, **Baby Bank Layout**, to determine the appearance of the report.

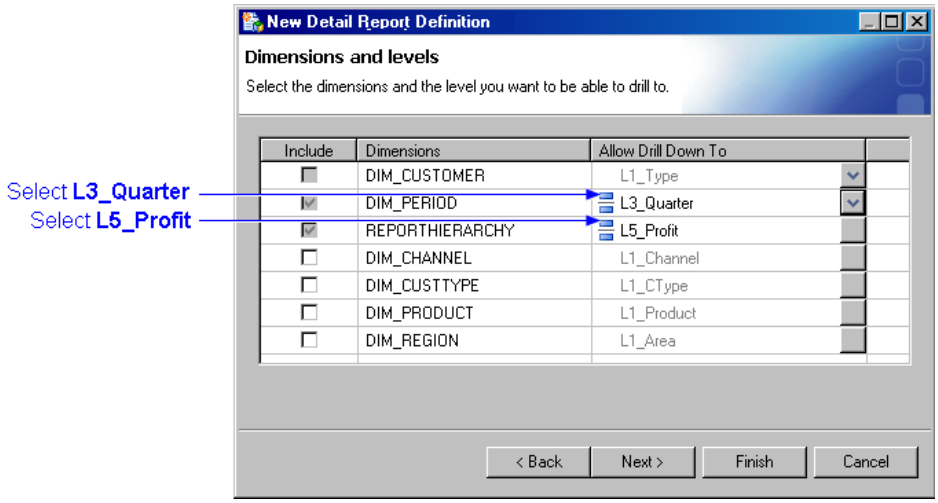
   Note: After you select a report hierarchy, the drop-down list of report layouts lists only those layouts belonging to the selected report hierarchy.



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.

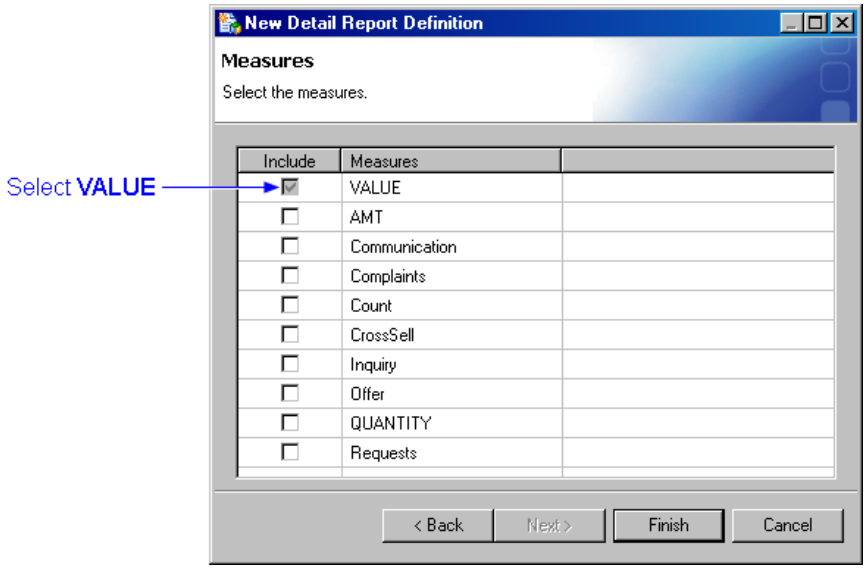
The goal for quick reporting response time is to try to filter your result such that the selected answer has fewer than 1,000 records. So, while there are other dimensions in the model, it would be unwise to select any dimension other than customer, because the filtered results would be more than 1,000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

7. Click **Next**.
8. 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.



9. Click **Next**.
10. Select **VALUE** as a measure to be included in the report.

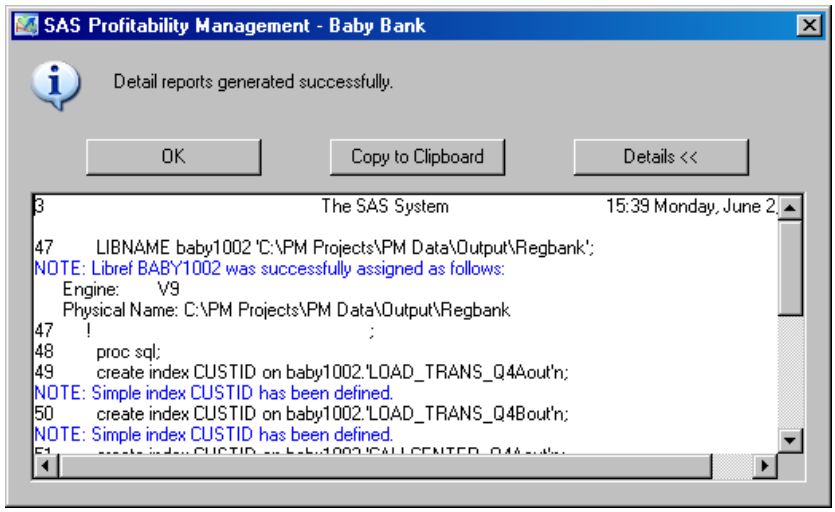
    The VALUE property is the default. It is the calculated cost for a transaction.



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

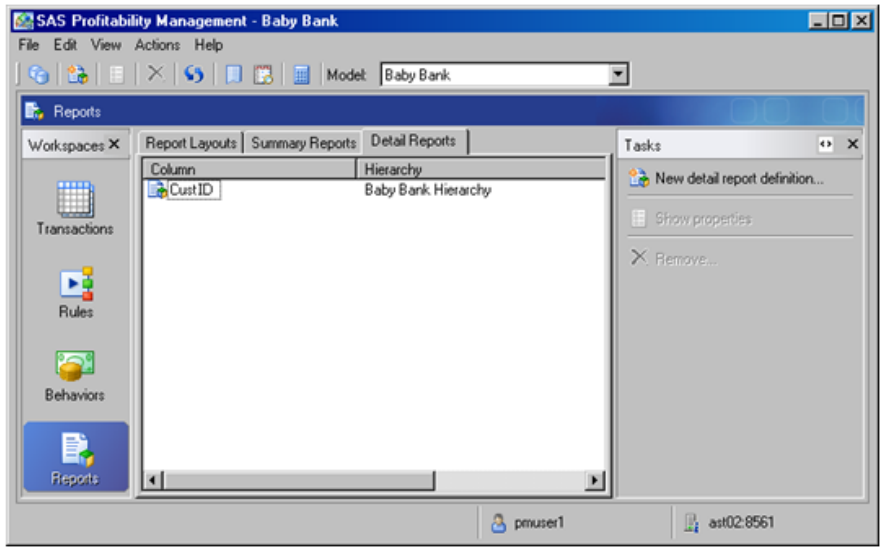
11. Click **Finish**.

    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.



12. Click **OK**.

    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.



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

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

Chapter 14

View the Reports

Viewing Reports on the Web

After using the SAS Profitability Management rich client to calculate a model and generate reports, you can view the reports on the Web. Anyone with a browser and access to the Web can view the reports. You use the SAS Profitability Management Web application to display a list of summary and detail reports available for viewing. Once you choose a report to view, SAS Web Report Studio is launched for you to view the report and edit it if you want. You can access SAS Web Report Studio help by selecting **Help** from the SAS Web Report Studio menu (not the browser menu). You can also view SAS Web Report Studio product documentation at **`http://support.sas.com/documentation/onlinedoc/wrs/`**.

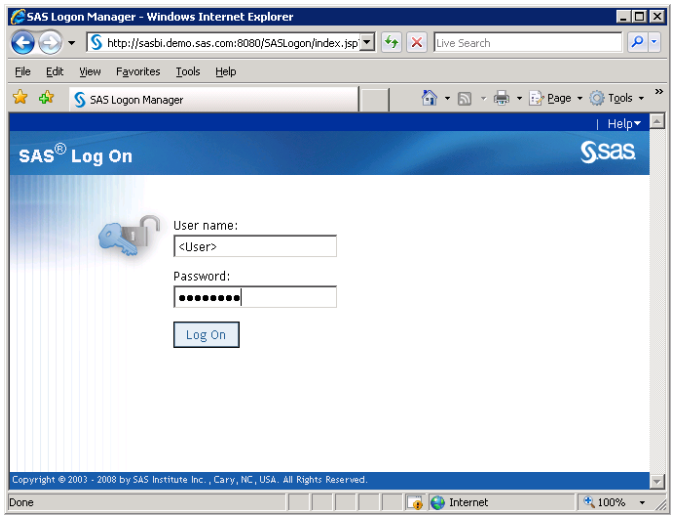
View the Summary Report

A summary report allows you to navigate all the dimensions of a SAS Profitability Management model.

Open the Baby Bank Report

1. Log on to the SAS Profitabilty Management Web application.

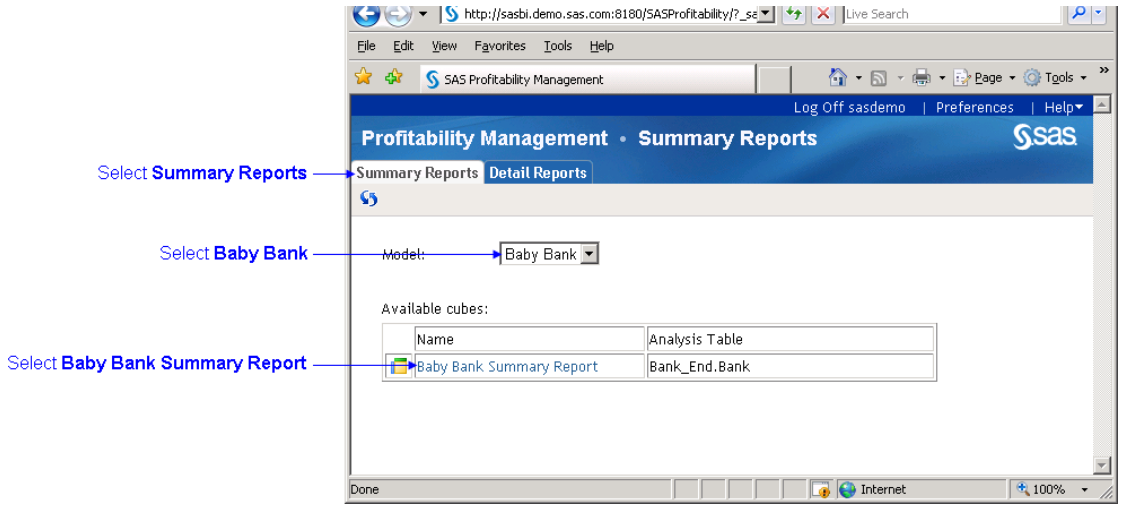
   Access it at: **http://**<<*your web server*>>**:8080/SASProfitability/**.



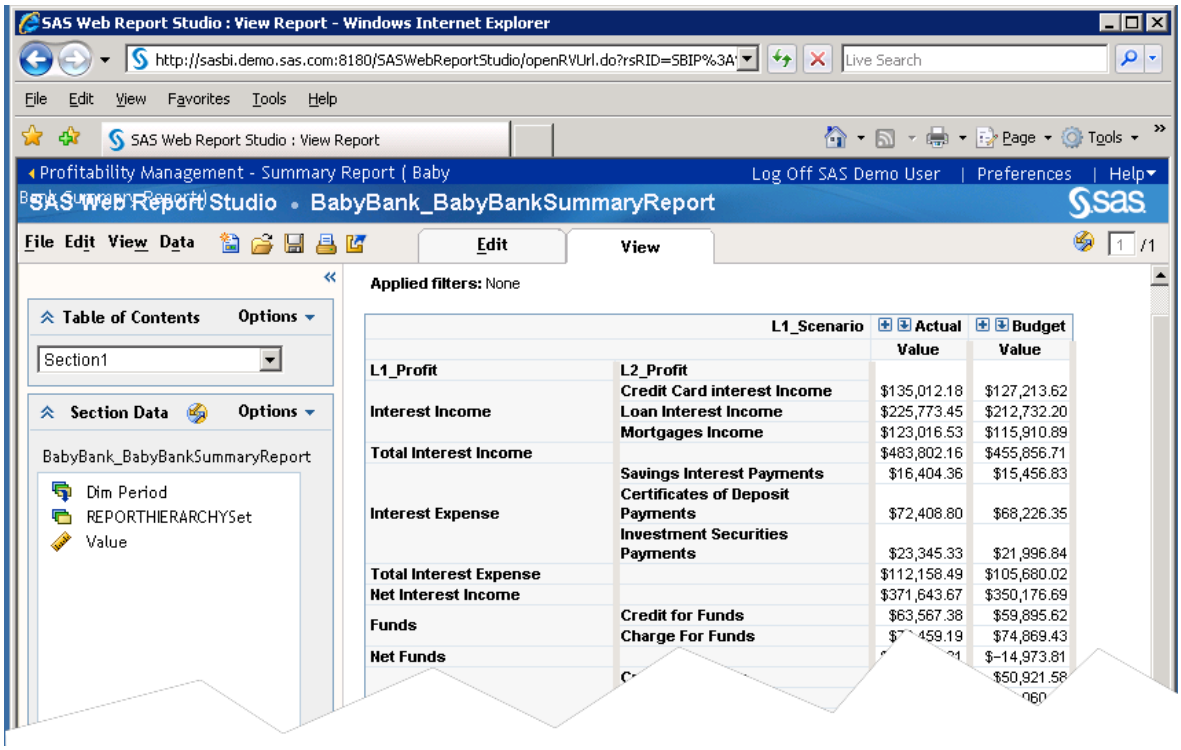
The user name and password you defined earlier using SAS Management Console—the same user name you have used up to now in accessing the SAS Profitability Management rich client.

SAS Web Report Studio 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 SAS Web Report Studio.



The profit and loss statement is displayed in table format. By default, the display shows:

Rows
: which are determined by the combination of the report layout table and the report hierarchy table.

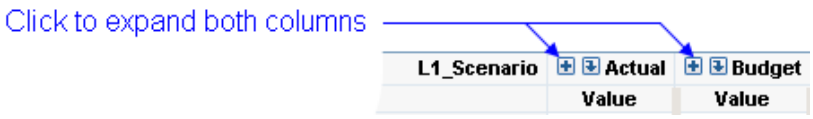
Columns
: which represent the time period dimension.

Note: Drag the column borders to resize the columns.

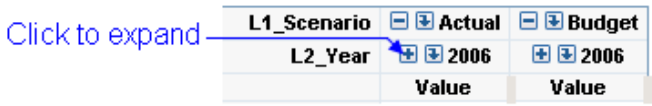
See More Columns of Data

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

1. Click the plus sign next to **Actual** and next to **Budget** to expand both columns.



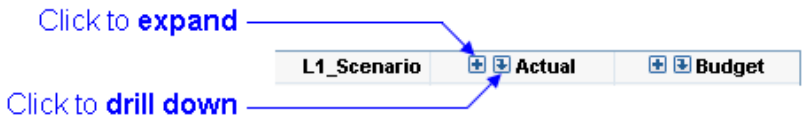
2. Click the plus sign next to **2006** to expand the year.



You should see a column for the actual results for the fourth quarter of 2006, as shown in the following graphic:

	L1_Scenario	Actual	Budget
	L2_Year	2006	2006
	L3_Quarter	2006_Q4	
		Value	Value
L1_Profit	L2_Profit		
Interest Income	Credit Card interest Income	$135,012.18	$127,213.62
	Loan Interest Income	$225,773.45	$212,732.20
	Mortgages Income	$123,016.53	$115,910.89
Total Interest Income		$483,802.16	$455,856.71
Interest Expense	Savings Interest Payments	$16,404.36	$15,456.83
	Certificates of Deposit Payments	$72,408.80	$68,226.35
	Investment Securities Payments	$23,345.33	$21,996.84
Total Interest Expense		$112,158	$105,680.02
Net Int… t Income		$371,6…	$350,176…
	…it for F…	$63…	…
	…Fo…	$7…	

Note: You click the plus sign to expand, and you 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.

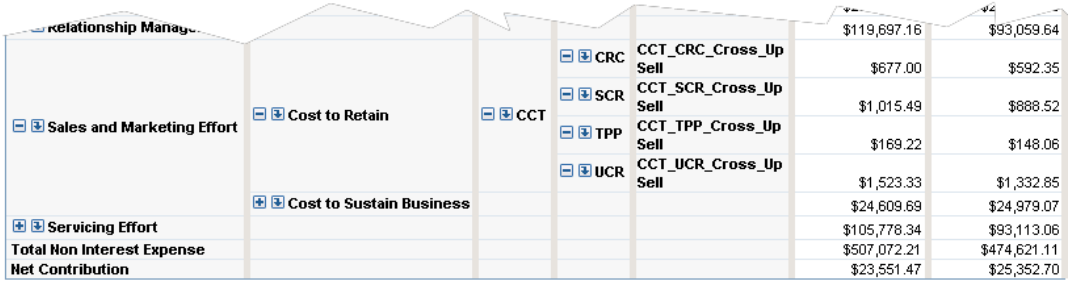
After expanding or drilling down, you can navigate in the opposite direction (collapsing) by clicking the minus sign.

See More Rows of Data

1. 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.

2. Continue clicking the plus sign to fully expand that branch of the **Sales and Marketing Effort**. Your display should look like the following:

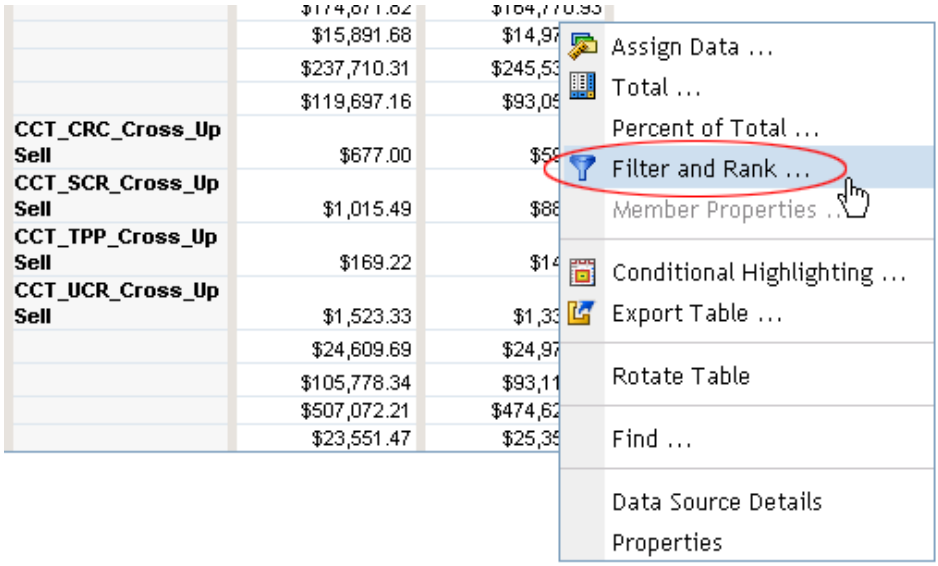


The levels available to drill down to additional detail in the cube depend on the number of dimension levels in the dimension hierarchy and the number of dimension levels that you select in the definition of the summary cube. For more information, see "Report Hierarchy" on page 136 and "Report Layout" on page 140.

Filter Data

You can select specific values to display using a filter.

1. Right-click the numeric columns in the report and select **Filter and Rank** from the drop-down menu.

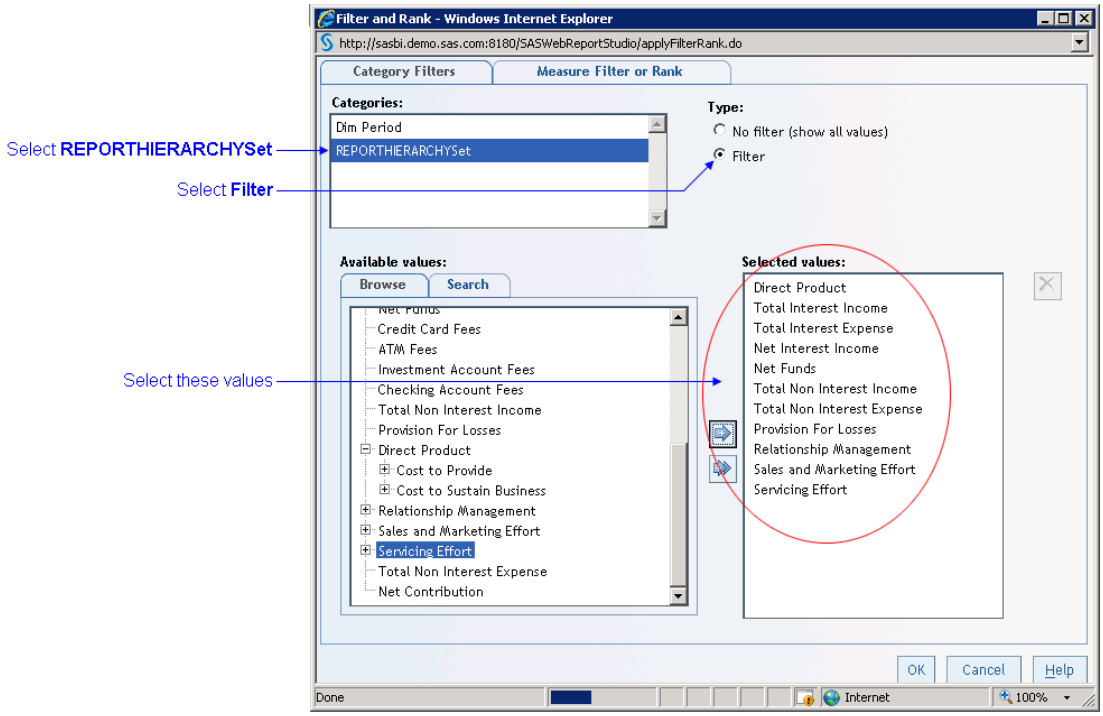


The Filter and Rank dialog opens.

2. Select **REPORTHIERARCHYSet** to filter the report by its rows, and click **Filter**.

   Then select the following rows (you will have to scroll to select them all), and click **OK**:

- Direct Product
- Total Interest Income
- Total Interest Expense
- Net Interest Income
- Net Funds
- Total Non Interest Income
- Total Non Interest Expense
- Provision For Losses
- Relationship Management
- Sales and Marketing Effort
- Servicing Effort

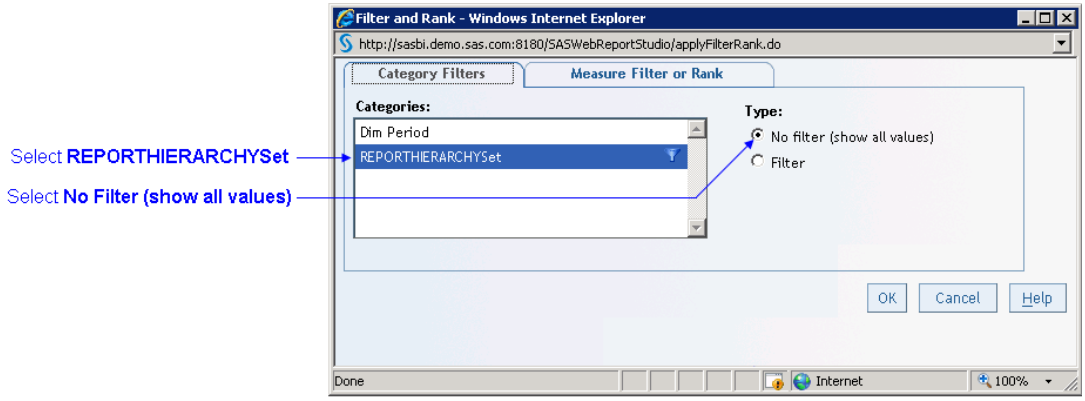


Your report should look like the following. It is filtered to show only the rows that you have selected.

Applied filters: REPORTHIERARCHYSet equal to Direct Product, Total Interest Income, Total Interest Expense, Net Interest Income, Net Funds, Total Non Interest Income, Total Non Interest Expense, Provision For Losses, Relationship Management, Sales and Marketing Effort, Servicing Effort

L1_Scenario	Actual	Budget
L2_Year	2006	2006
	Value	Value
L1_Profit		
Direct Product	$237,710.31	$245,533.75
Total Interest Income	$483,802.16	$455,856.71
Total Interest Expense	$112,158.49	$105,680.02
Net Interest Income	$371,643.67	$350,176.69
Net Funds	$-15,891.81	$-14,973.81
Total Non Interest Income	$174,871.82	$164,770.93
Total Non Interest Expense	$507,072.21	$474,621.11
Provision For Losses	$15,891.68	$14,973.81
Relationship Management	$119,697.16	$93,059.64
Sales and Marketing Effort	$27,994.72	$27,940.85
Servicing Effort	$105,778.34	$93,113.06

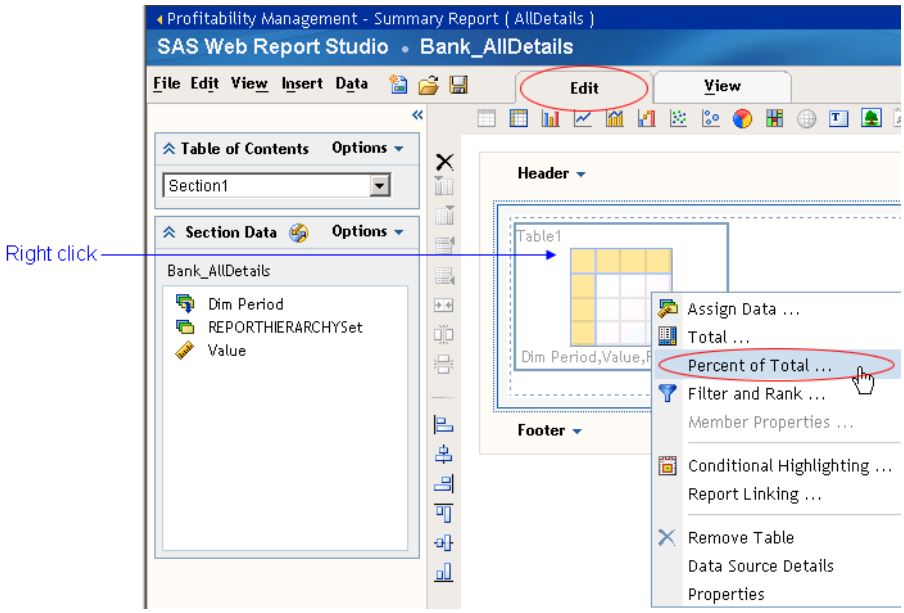
3. Clear the filter by doing the following:
   a. Right-click the numeric columns in the report and select **Filter and Rank** from the drop-down menu.
   b. Select **REPORTHIERARCHYSet**.
   c. Select **No Filter (show all values)**.
   d. Click **OK**.



Display Percentages

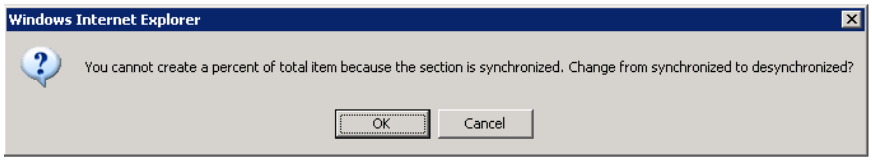
You can easily display the percentages for the different categories of income and expense.

1. Click the **Edit** tab.
2. Right-click the Table area and select **Percent of Total**.



A warning message appears telling you that you cannot display percentages without first changing the section from synchronized to desynchronized.

Note: By default, OLAP data is synchronized.

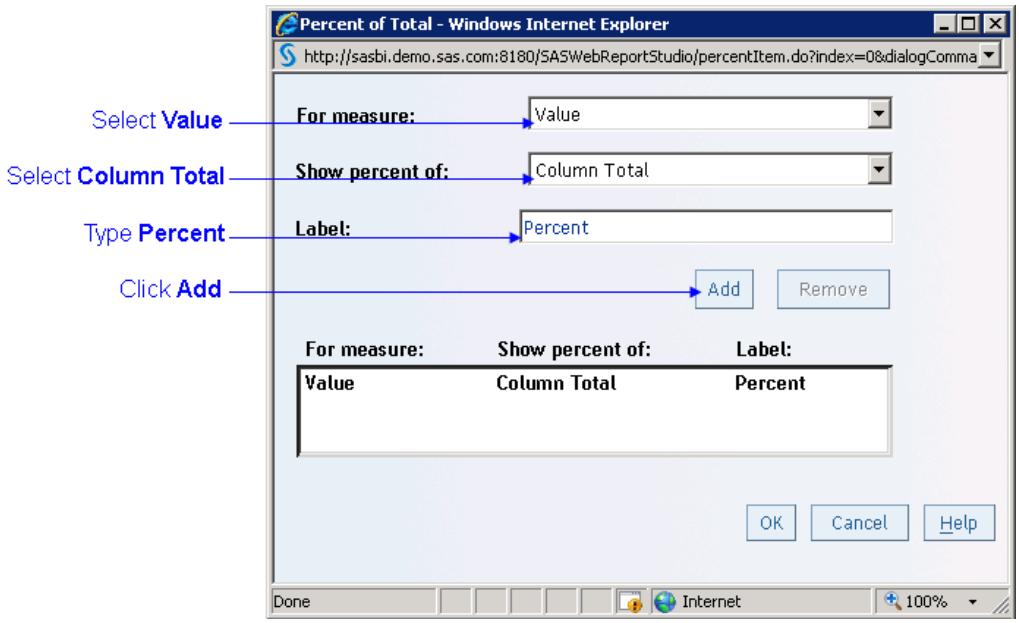


3. Click **OK** to change to desynchronized. Then right-click the Table icon again and select **Percent of Total**.

   The Percent of Total dialog box opens.

4. Specify the following, and then click **Add**:

Field	Value
For measure	Value
Show percent of	Column Total
Label	Percent



5. Click **OK** to close the Percent of Total dialog box.

   The following report is displayed:

		L1_Scenario	Actual		Budget	
			Value	Percent	Value	Percent
L1_Profit	L2_Profit					
Interest Income	Credit Card interest Income		$135,012.18	4.39%	$127,213.62	4.39%
	Loan Interest Income		$225,773.45	7.33%	$212,732.20	7.34%
	Mortgages Income		$123,016.53	4.00%	$115,910.89	4.00%
Total Interest Income			$483,802.16	15.72%	$455,856.71	15.73%
Interest Expense	Savings Interest Payments		$16,404.36	0.53%	$15,456.83	0.53%
	Certificates of Deposit Payments		$72,408.80	2.35%	$68,226.35	2.35%
	Investment Securities Payments		$23,345.33	0.76%	$21,996.84	0.76%
Total Interest Expense			$112,158.49	3.64%	$105,680.02	3.65%
Net Interest Income			$371,643.67	12.07%	$350,176.69	12.09%
Funds	Credit for Funds		$63,567.38	2.07%	$59,895.62	2.07%
	Charge For Funds		$79,459.19	2.58%	$74,869.43	2.58%
Net Funds			$-15,891.81	-0.52%	$-14,973.81	-0.52%
Fees	Credit Card Fees		$54,043.22	1.76%	$50,921.58	1.76%
	ATM Fees		$10,677.53	0.35%	$10,060.87	0.35%
	Investment Account Fees		$93,381.32	3.03%	$87,987.40	3.04%
	Checking Account Fees		$16,769.75	0.54%	$15,801.08	0.55%
Total Non Interest Income			$174,871.82	5.68%	$164,770.93	5.69%
Provision For Losses			$15,891.68	0.52%	$14,973.81	0.52%
Direct Product			$237,710.31	7.72%	$245,533.75	8.47%
Relationship Management			$119,697.16	3.89%	$93,059.64	3.21%
Sales and Marketing Effort			$27,994.72	0.91%	$27,940.85	0.96%
Servicing Effort			$105,778.34	3.44%	$93,113.06	3.21%
Total Non Interest Expense			$507,072.21	16.47%	$474,621.11	16.38%
Net Contribution			$23,551.47	0.77%	$25,352.70	0.88%
Total			**$3078139.26**	**100.00%**	**$2897178.17**	**100.00%**

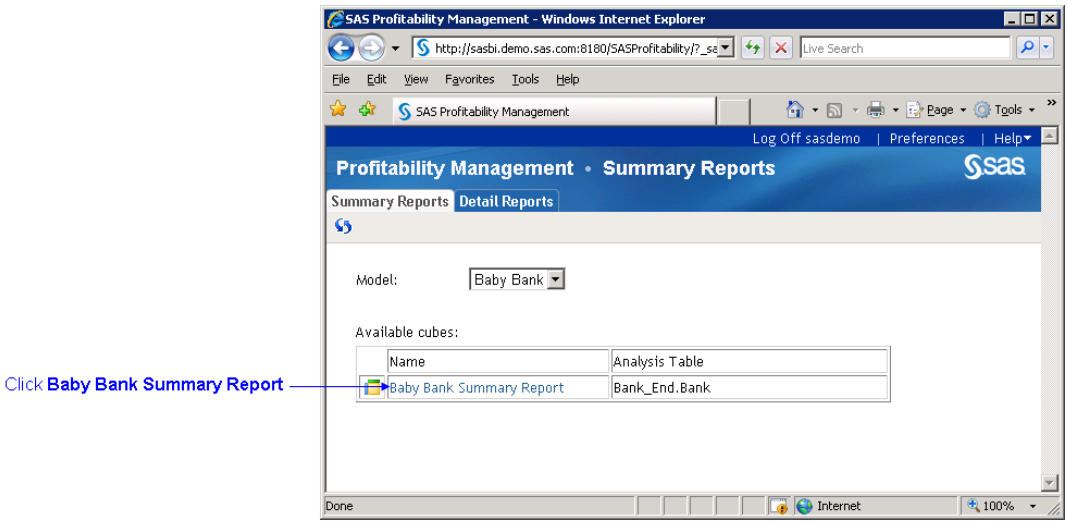
Add a Bar Chart to the Report

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.

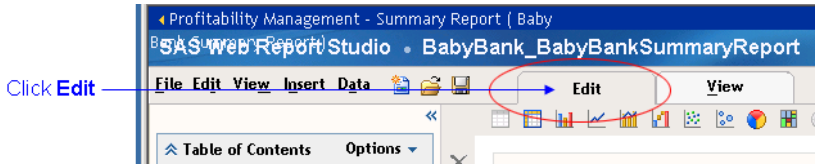
The list of available summary reports appears.

2. Click **Baby Bank Summary Report**.

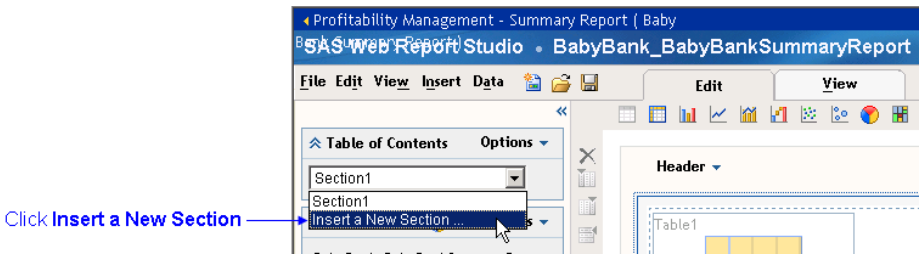


The Baby Bank Summary Report opens.

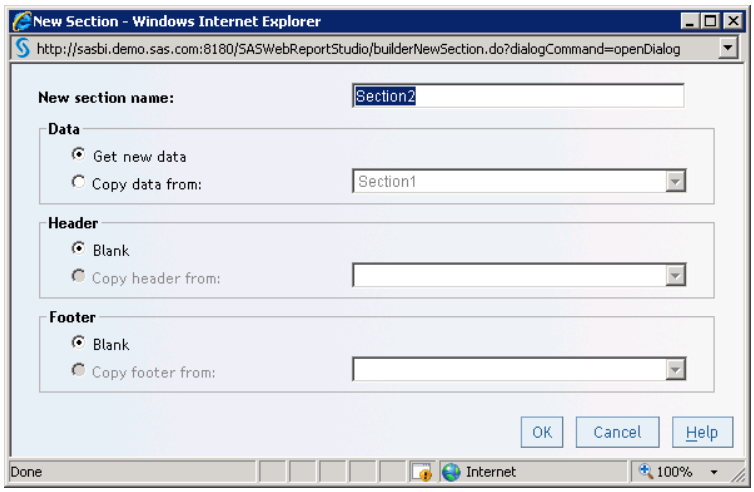
3. Click the **Edit** tab.



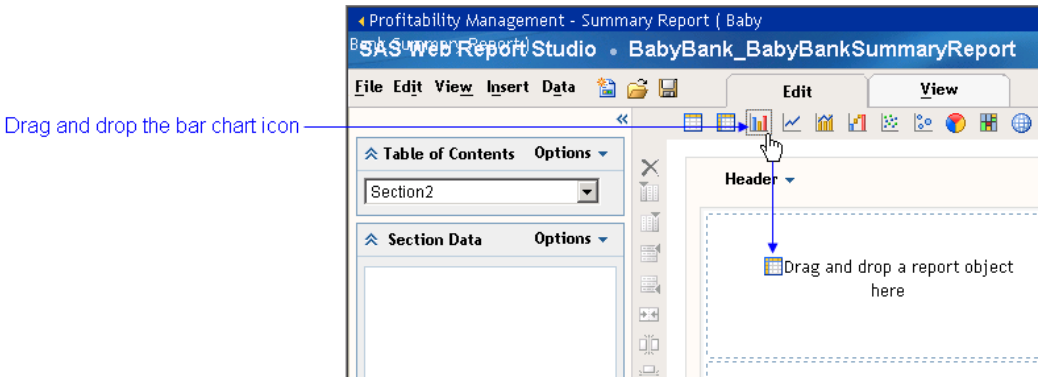
4. Click the down arrow under **Table of Contents** and select **Insert a New Section**.



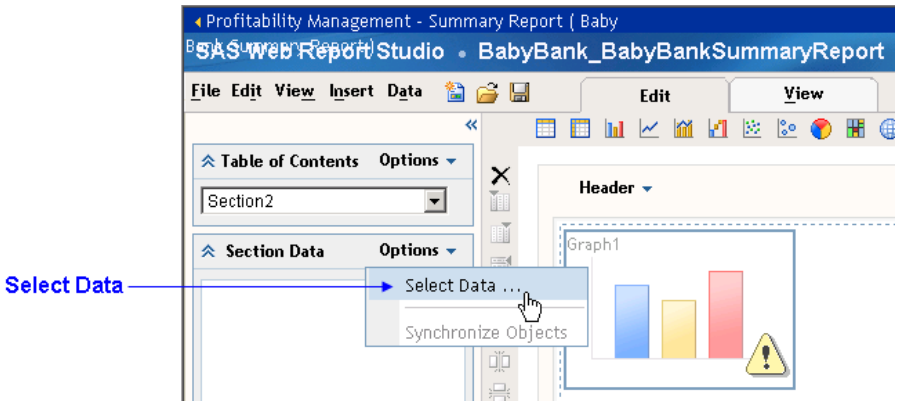
The New Section window opens. Click **OK** to accept the defaults.



5. Drag the Bar Chart icon ( ) and drop it onto the report area.

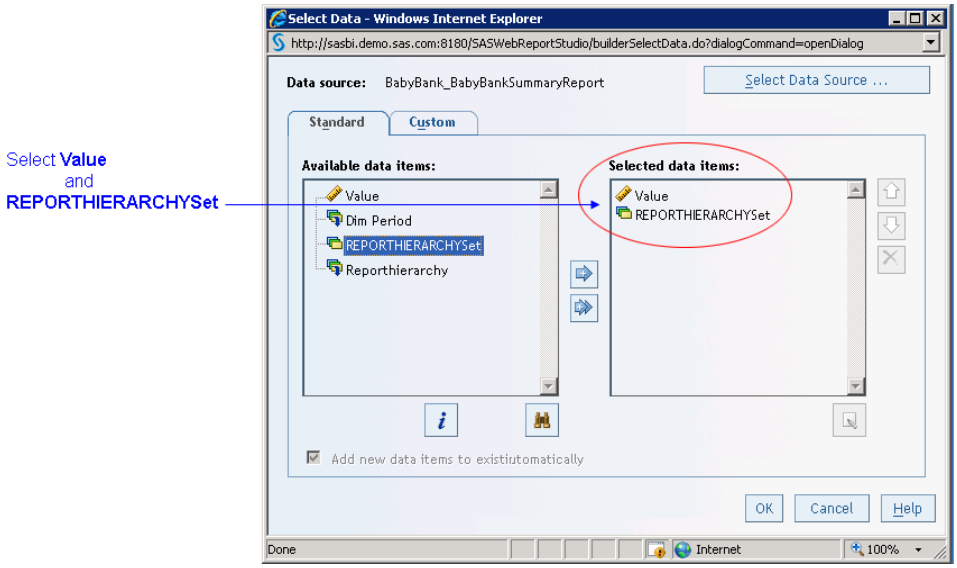


6. Under **Section Data**, choose **Select Data**.

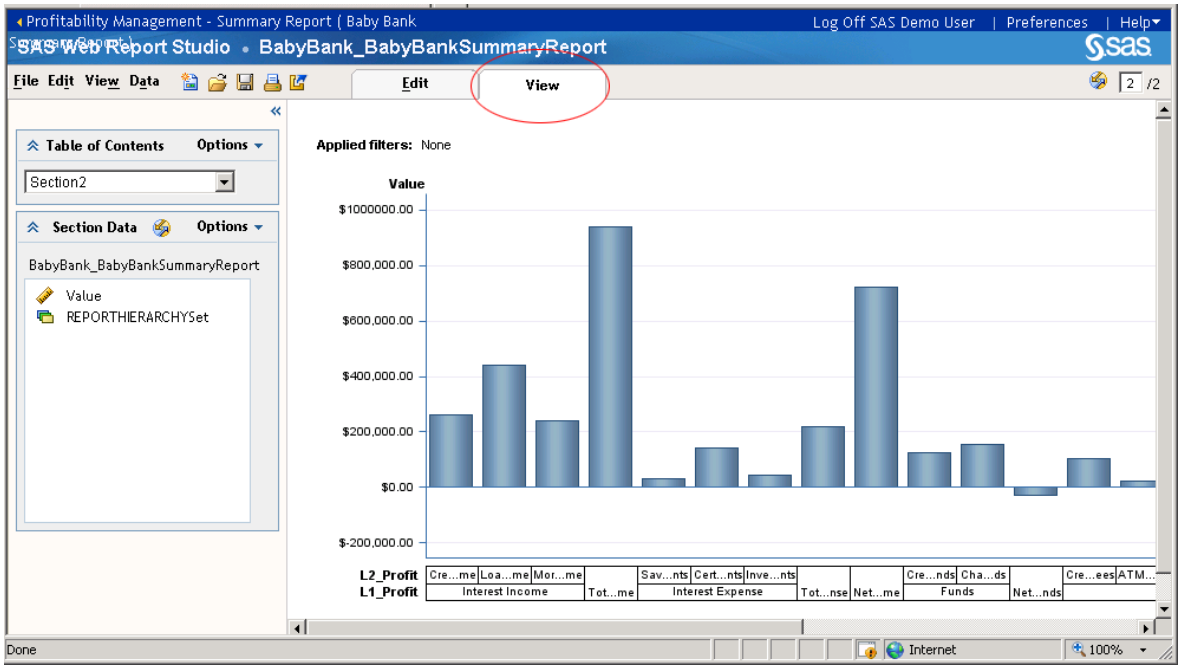


The Select Data window opens.

7. Select **Value** and **REPORTHIERARCHYSet**, and then click **OK**.

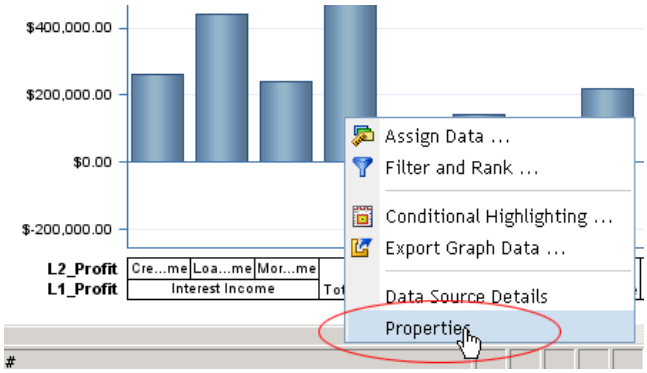


8. Click the **View** tab to see the resulting bar chart.



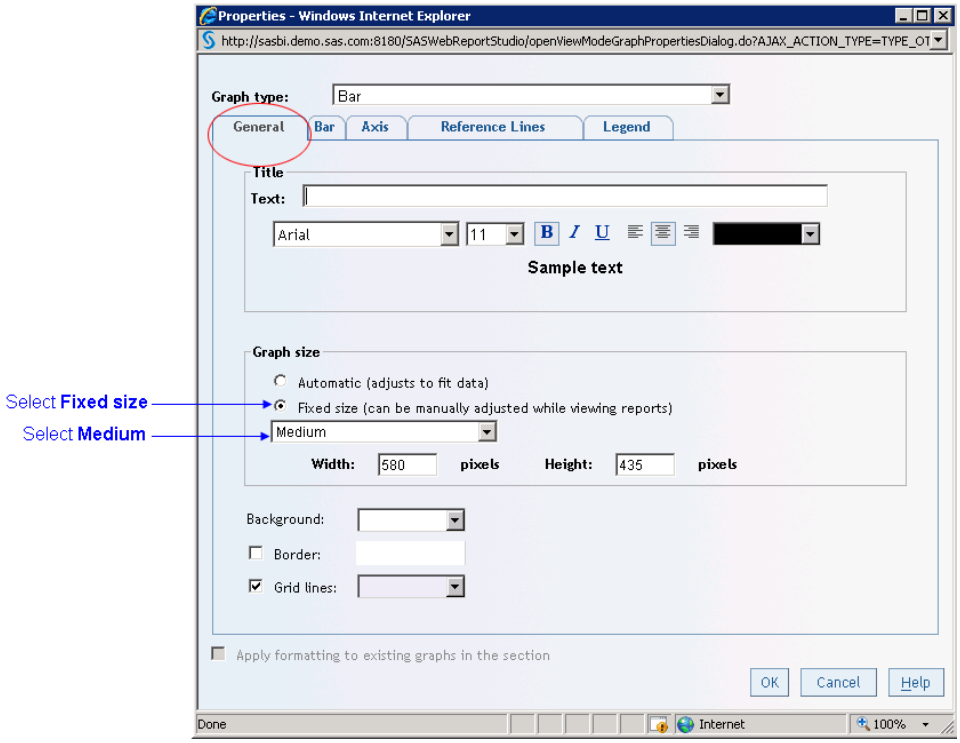
Customize the Bar Chart

1. Right-click the bar chart and select **Properties**.

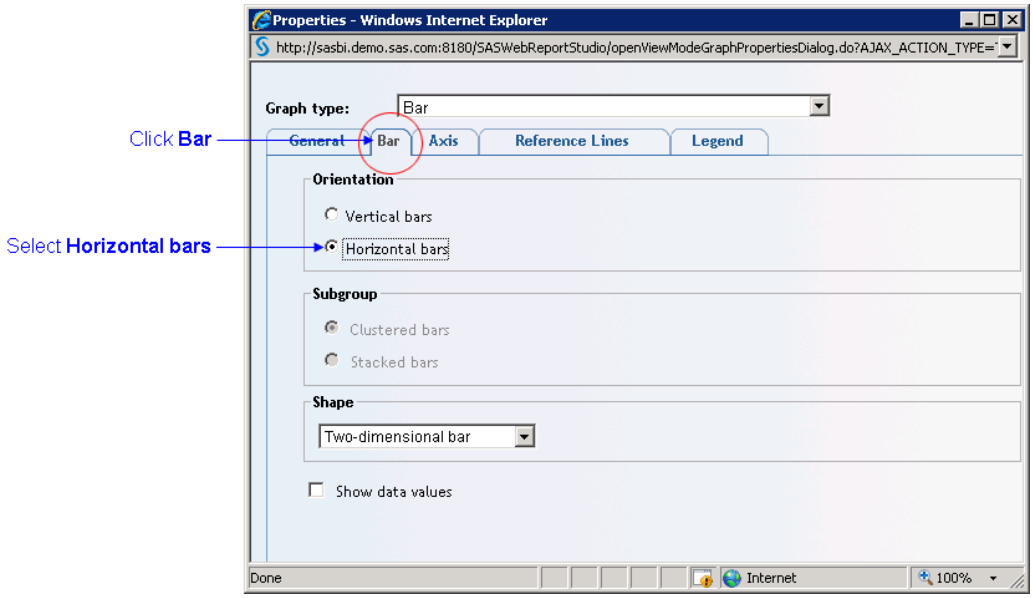


The Properties window opens.

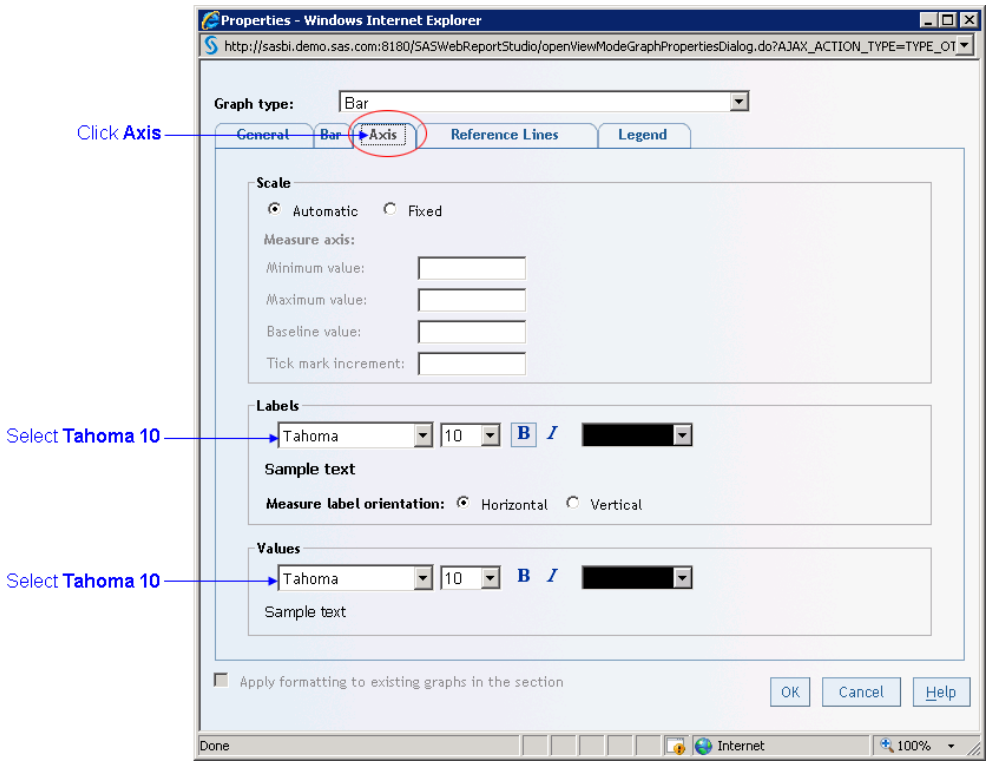
2. From the **General** tab, select **Fixed Size** and **Medium** for the graph size.



3. Click the **Bar** tab and select **Horizontal bars**.



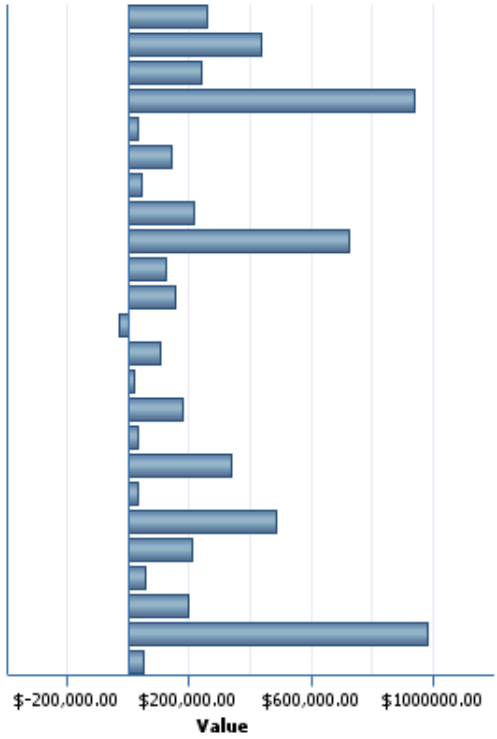
4. Click the **Axis** tab and select **Tahoma 10** point for both labels and values.



5. Click **OK**.

   The bar chart is redrawn to your specifications.

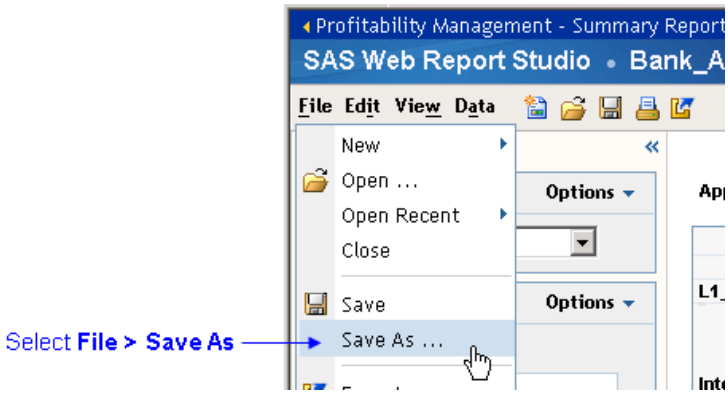
L1_Profit	L2_Profit
Interest Income	Credit Card interest Income
	Loan Interest Income
	Mortgages Income
Total Interest Income	
Interest Expense	Savings Interest Payments
	Certificates of Deposit Payments
	Investment Securities Payments
Total Interest Expense	
Net Interest Income	
Funds	Credit for Funds
	Charge For Funds
Net Funds	
Fees	Credit Card Fees
	ATM Fees
	Investment Account Fees
	Checking Account Fees
Total Non Interest Income	
Provision For Losses	
Direct Product	
Relationship Management	
Sales and Marketing Effort	
Servicing Effort	
Total Non Interest Expense	
Net Contribution	



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 then click **OK**.

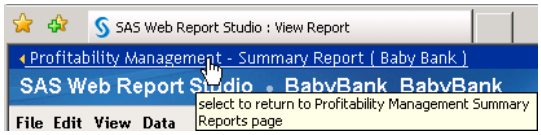
   The view is saved and can be opened again and shared with others for review.

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

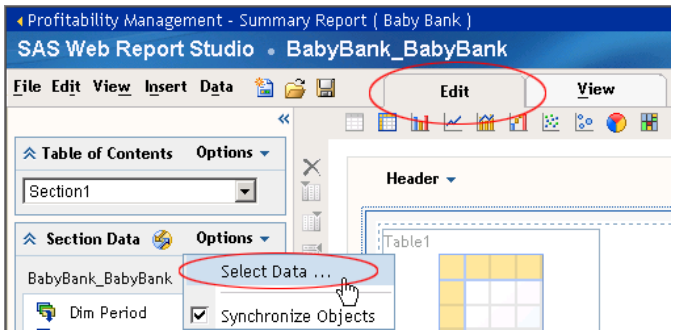
Add a Custom Dimension

So far we haven't used any of the custom dimensions that are included in the report. To illustrate adding a custom dimension, let's first start over again with a fresh view of the summary report.

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

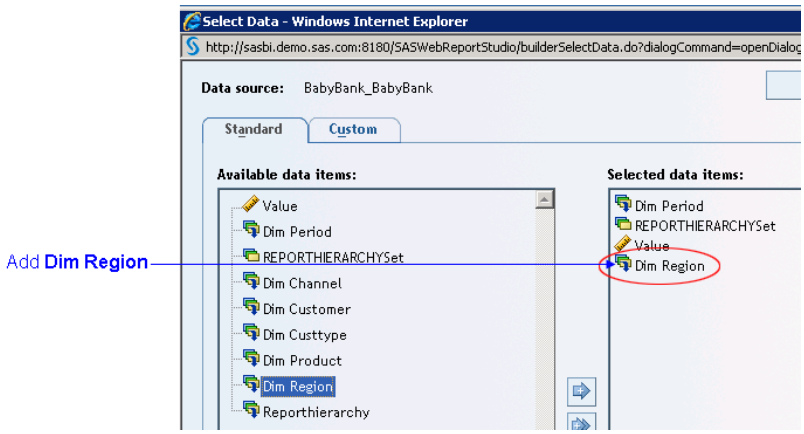


2. Open the Baby Bank summary report.
3. Click the **Edit** tab, and then select **Options** ⇨ **Select Data**.

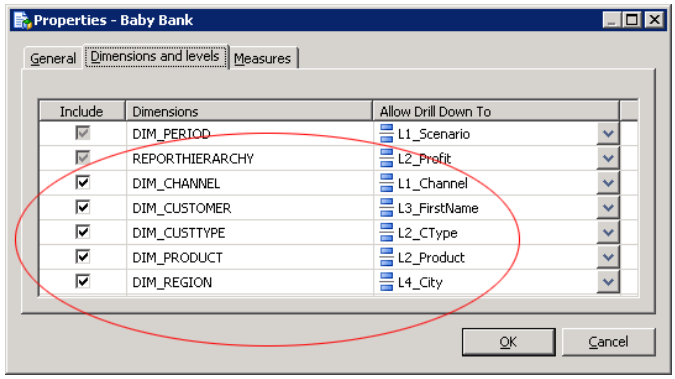


The Select Data window opens.

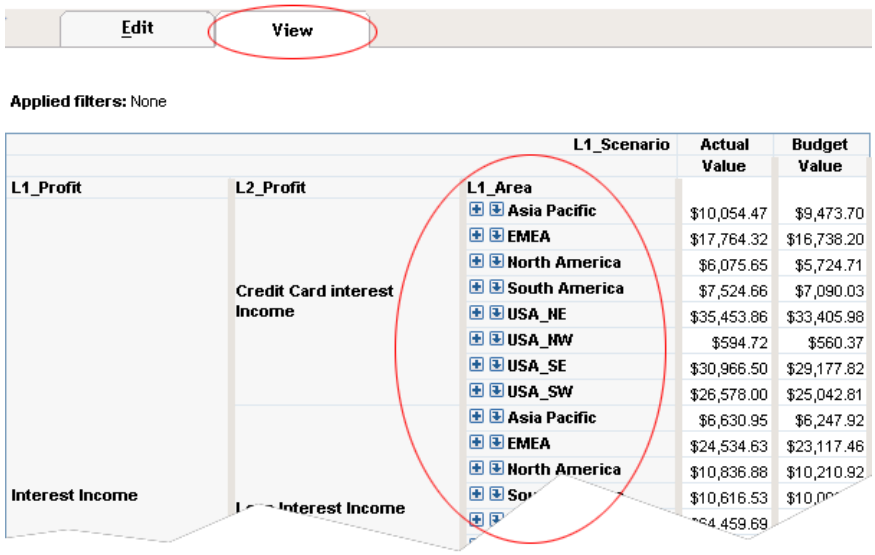
4. Select **Dim Region** and move it to the list of **Selected data items** on the right side of the window. Then click **OK**.



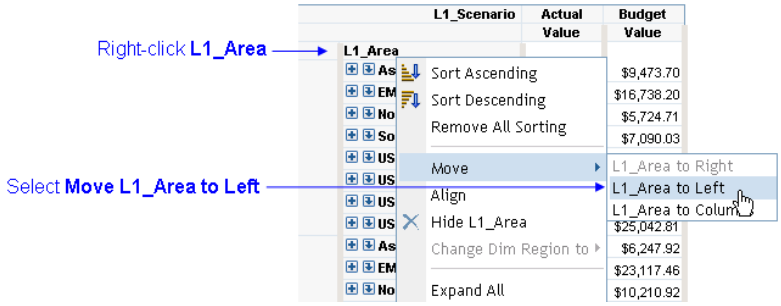
Note: For the custom dimensions to be available for selection, you must have chosen to include the dimensions in the generated cube as shown in cube Properties window below.



5. Click the **View** tab to see the resulting table. Notice the additional column L1_Area.

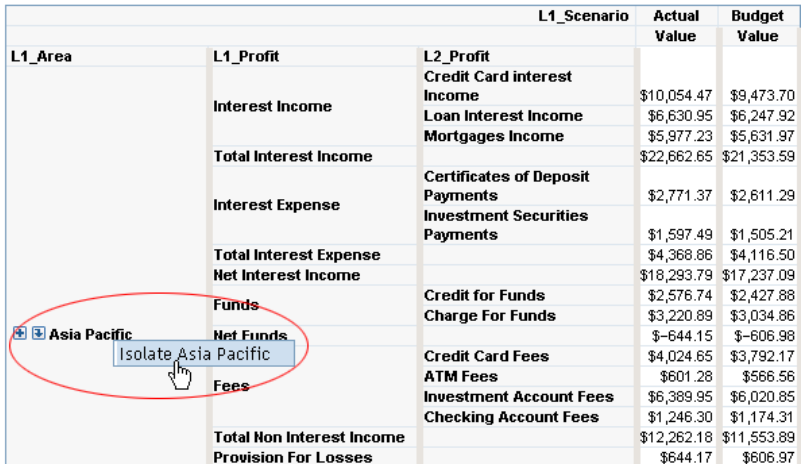


6. Right-click **L1_Area** and select **Move ⇨ L1_Area to Left**.



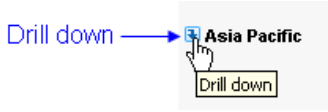
The L1_Area column is moved to the left of the **REPORTHIERARCHYSet**.

7. Right-click **Asia Pacific** and select **Isolate Asia Pacific**.



		L1_Scenario	Actual	Budget
			Value	Value
L1_Area	L1_Profit	L2_Profit		
Asia Pacific	Interest Income	Credit Card interest Income	$10,054.47	$9,473.70
		Loan Interest Income	$6,630.95	$6,247.92
		Mortgages Income	$5,977.23	$5,631.97
	Total Interest Income		$22,662.65	$21,353.59
	Interest Expense	Certificates of Deposit Payments	$2,771.37	$2,611.29
		Investment Securities Payments	$1,597.49	$1,505.21
	Total Interest Expense		$4,368.86	$4,116.50
	Net Interest Income		$18,293.79	$17,237.09
	Funds	Credit for Funds	$2,576.74	$2,427.88
		Charge For Funds	$3,220.89	$3,034.86
	Net Funds		$-644.15	$-606.98
	Fees	Credit Card Fees	$4,024.65	$3,792.17
		ATM Fees	$601.28	$566.56
		Investment Account Fees	$6,389.95	$6,020.85
		Checking Account Fees	$1,246.30	$1,174.31
	Total Non Interest Income		$12,262.18	$11,553.89
	Provision For Losses		$644.17	$606.97

8. Click the down arrow next to **Asia Pacific** to drill down.



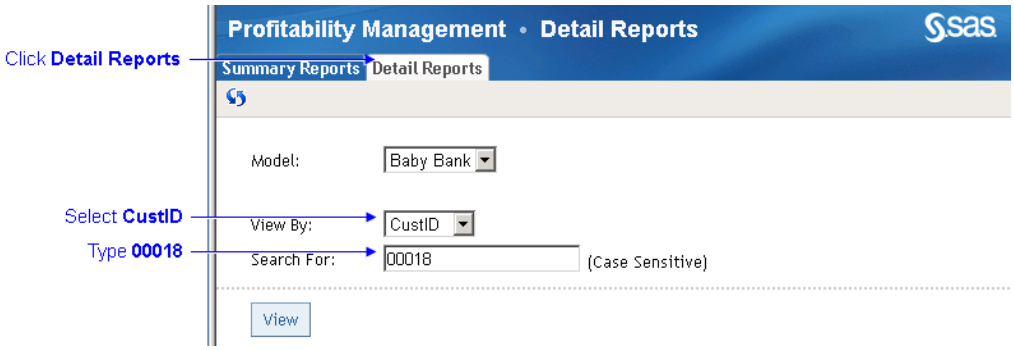
9. Continue drilling down until you reach the bottom of the region hierarchy. You can see the results for Sydney, which you would not be able to see without adding the region dimension to the report.

		L1_Scenario	Actual	Budget
			Value	Value
L4_City	L1_Profit	L2_Profit		
Sydney	Interest Income	Mortgages Income	$5,977.23	$5,631.97
	Total Interest Income		$5,977.23	$5,631.97
	Net Interest Income		$5,977.23	$5,631.97
	Funds	Credit for Funds	$285.17	$268.69
		Charge For Funds	$356.46	$335.87
	Net Funds		$-71.29	$-67.18
	Fees	ATM Fees	$113.68	$107.11
	Total Non Interest Income		$113.68	$107.11
	Provision For Losses		$71.29	$67.17
	Total Non Interest Expense		$71.29	$67.17
	Net Contribution		$5,948.33	$5,604.73

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 is displayed.

	L1_Scenario	Actual	Budget
		Value	Value
L1_Profit	L2_Profit		
Interest Income	Credit Card interest Income	$546.19	$514.64
	Loan Interest Income	$1,552.13	$1,462.47
Total Interest Income		$2,098.32	$1,977.11
Net Interest Income		$2,098.32	$1,977.11
Funds	Credit for Funds	$394.78	$371.97
	Charge For Funds	$493.48	$464.96
Net Funds		$-98.70	$-92.99
Fees	Credit Card Fees	$218.63	$206.00
	ATM Fees	$138.24	$130.26
Total Non Interest Income		$356.87	$336.26
Provision For Losses		$98.69	$92.99
Relationship Management		$97.30	$102.06
Sales and Marketing Effort		$10.46	$9.15
Total Non Interest Expense		$206.45	$204.20
Net Contribution		$2,150.04	$2,016.18

Notice that this report contains fewer rows than the summary report. And, if you open the detail report for customer 00040, notice that it contains a different number of rows again as shown in the following graphic:

Summary report

Interest Income	Credit Card interest Income
	Loan Interest Income
	Mortgages Income
Total Interest Income	
Interest Expense	Savings Interest Payments
	Certificates of Deposit Payments
	Investment Securities Payments
Total Interest Expense	
Net Interest Income	
Funds	Credit for Funds
	Charge For Funds
Net Funds	
Fees	Credit Card Fees
	ATM Fees
	Investment Account Fees
	Checking Account Fees
Total Non Interest Income	
Provision For Losses	
⊞ ⊞ Direct Product	
⊞ ⊞ Relationship Management	
⊞ ⊞ Sales and Marketing Effort	
⊞ ⊞ Servicing Effort	
Total Non Interest Expense	
Net Contribution	

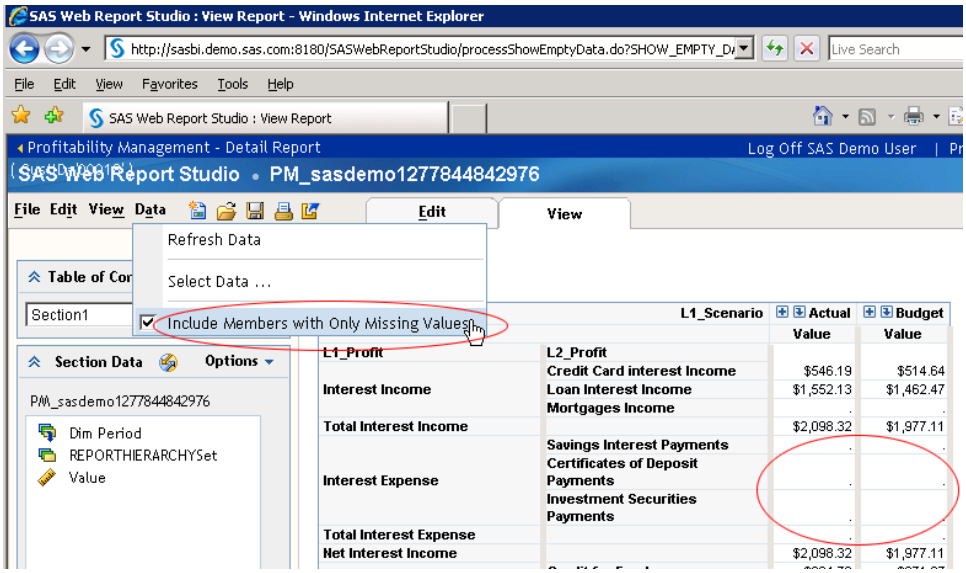
Detail Report (customer 00018)

Interest Income	Credit Card interest Income
	Loan Interest Income
Total Interest Income	
Net Interest Income	
Funds	Credit for Funds
	Charge For Funds
Net Funds	
Fees	Credit Card Fees
	ATM Fees
Total Non Interest Income	
Provision For Losses	
⊞ ⊞ Relationship Management	
⊞ ⊞ Sales and Marketing Effort	
Total Non Interest Expense	
Net Contribution	

Detail Report (customer 00040)

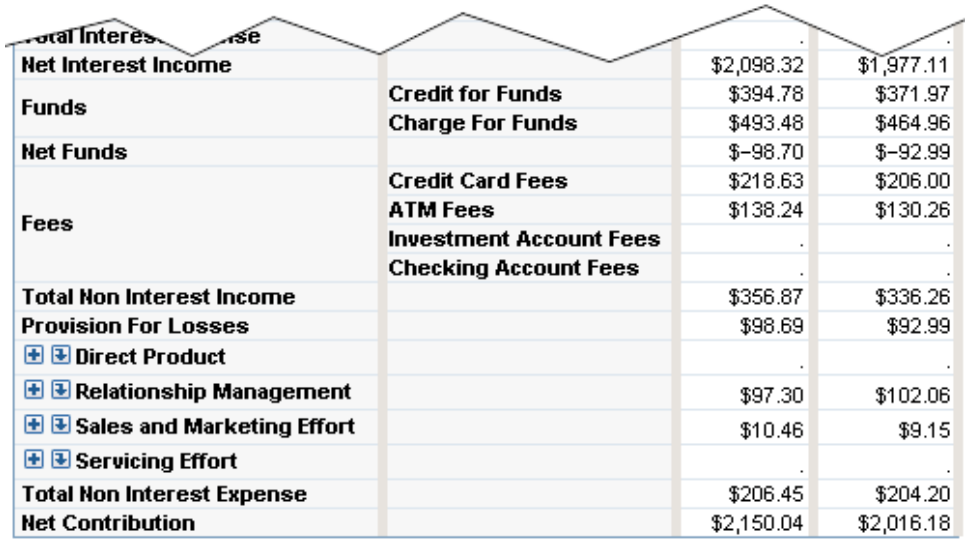
Interest Income	Credit Card interest Income
Total Interest Income	
Interest Expense	Savings Interest Payments
Total Interest Expense	
Net Interest Income	
Funds	Credit for Funds
	Charge For Funds
Net Funds	
Fees	Credit Card Fees
	ATM Fees
	Checking Account Fees
Total Non Interest Income	
Provision For Losses	
⊞ ⊞ Direct Product	
⊞ ⊞ Relationship Management	
⊞ ⊞ Sales and Marketing Effort	
⊞ ⊞ Servicing Effort	
Total Non Interest Expense	
Net Contribution	

What accounts for the difference is that rows with no value to show are suppressed. To show rows with no values, select **Data ⇨ Include Members with Only Missing Values**. If you don't select **Include Members with Only Missing Values**, then rows with no data are suppressed.



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

- She has a credit card and a loan and has been paying a significant amount of credit card interest.
- She has also been paying a lot of unsecured loan 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.
- She does not have any interest expense, which indicates that her savings, CDs, and investments are not with Baby Bank. So she might be a prime target for marketing those products.
- When you look at the bottom line, you see that customer 00018, Lynn York, is a very profitable customer for Baby Bank.



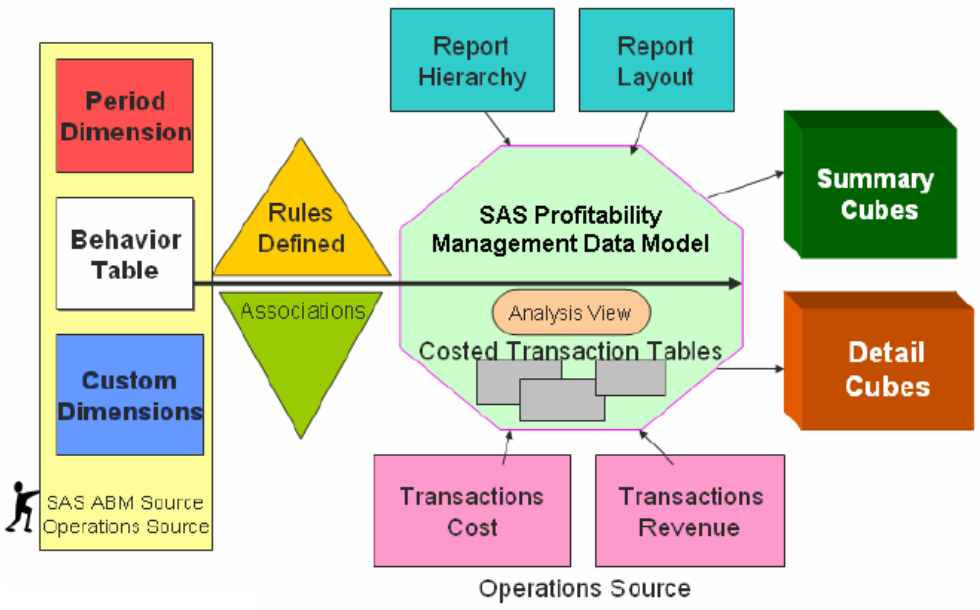
Net Interest Income		$2,098.32	$1,977.11
Funds	**Credit for Funds**	$394.78	$371.97
	Charge For Funds	$493.48	$464.96
Net Funds		$-98.70	$-92.99
Fees	**Credit Card Fees**	$218.63	$206.00
	ATM Fees	$138.24	$130.26
	Investment Account Fees	.	.
	Checking Account Fees	.	.
Total Non Interest Income		$356.87	$336.26
Provision For Losses		$98.69	$92.99
Direct Product		.	.
Relationship Management		$97.30	$102.06
Sales and Marketing Effort		$10.46	$9.15
Servicing Effort		.	.
Total Non Interest Expense		$206.45	$204.20
Net Contribution		$2,150.04	$2,016.18

Chapter 15

Summary of Model Elements

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, perform data validation 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 is 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. Each row in a behavior table

- represents the lowest level of drill down in an OLAP view - so you should not create more behaviors than you expect to view.
- is assigned to a transaction by a rule - so you should create only as many behaviors as you have rules for assignment.
- must be represented in the hierarchy table. The behavior table represents the leaf nodes of the hierarchy. The hiearchy table defines the parent nodes.

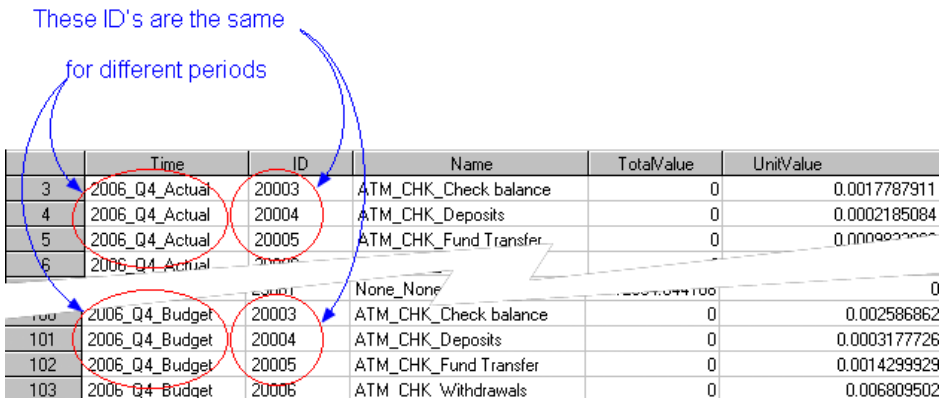
  Note: The following message received during cube generation ERROR: More unique keys were detected in the fact table than in the dimension table for the <hierarchy> dimension indicates that some behavior table rows are missing from the hierarchy dimension.

The behavior table contains the following fields:

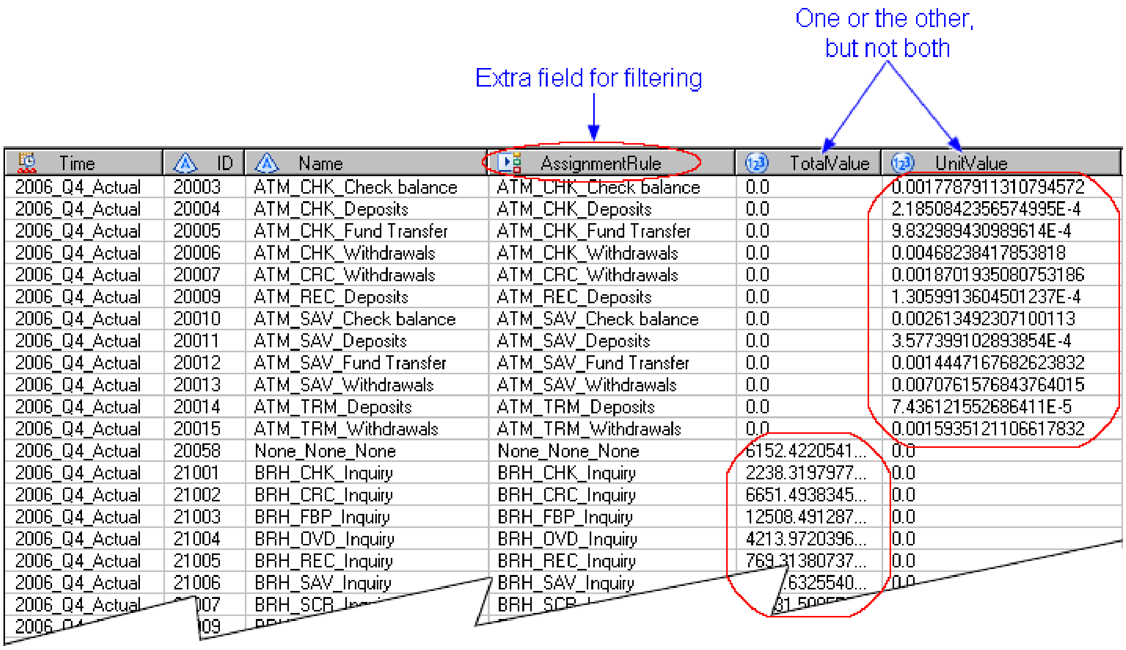
Name	Maximum Length	Description
Time	Char 32	The period for the transaction cost
ID	Char 32	The ID for the behavior
Name	Char 32	The name of the behavior
TotalValue	Num 8	The total source amount to be divided
UnitValue	Num 8	The unit cost for each transaction with this source

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).
- Different behaviors can have the same ID. This is frequently the case when the behaviors represent different time periods. Because rules are attached to behaviors by the behavior ID, having identical IDs allows you to attach the same rule to behaviors from different time periods. The following graphic shows an example of behaviors with the same ID.

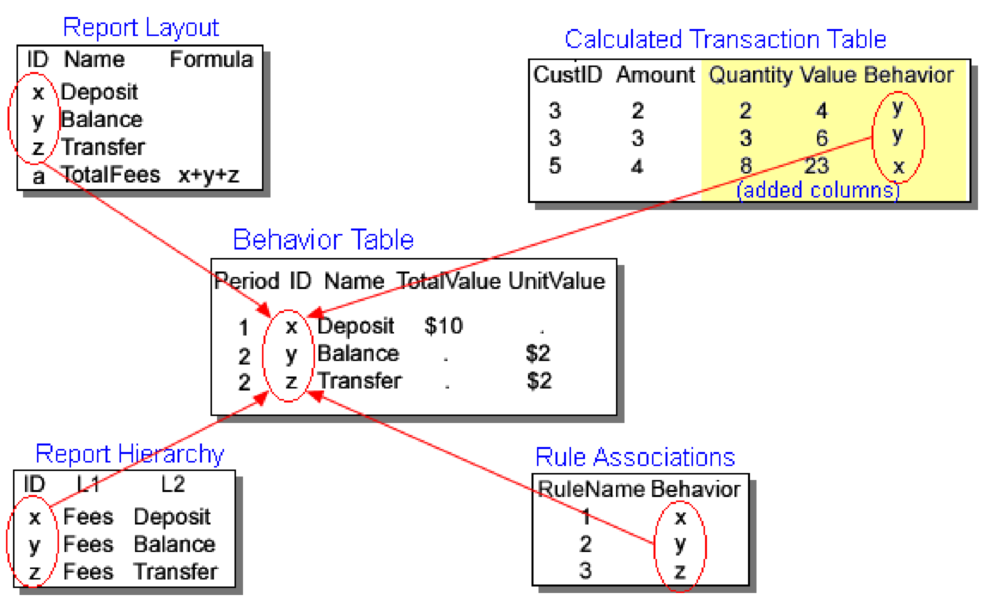


The following graphic shows the behavior table, BEHAVIORS, for the Baby Bank model:



Time	ID	Name	AssignmentRule	TotalValue	UnitValue
2006_Q4_Actual	20003	ATM_CHK_Check balance	ATM_CHK_Check balance	0.0	0.0017787911310794572
2006_Q4_Actual	20004	ATM_CHK_Deposits	ATM_CHK_Deposits	0.0	2.1850842356574995E-4
2006_Q4_Actual	20005	ATM_CHK_Fund Transfer	ATM_CHK_Fund Transfer	0.0	9.832989430989614E-4
2006_Q4_Actual	20006	ATM_CHK_Withdrawals	ATM_CHK_Withdrawals	0.0	0.00468238417853818
2006_Q4_Actual	20007	ATM_CRC_Withdrawals	ATM_CRC_Withdrawals	0.0	0.0018701935080753186
2006_Q4_Actual	20009	ATM_REC_Deposits	ATM_REC_Deposits	0.0	1.3059913604501237E-4
2006_Q4_Actual	20010	ATM_SAV_Check balance	ATM_SAV_Check balance	0.0	0.002613492307100113
2006_Q4_Actual	20011	ATM_SAV_Deposits	ATM_SAV_Deposits	0.0	3.577399102893854E-4
2006_Q4_Actual	20012	ATM_SAV_Fund Transfer	ATM_SAV_Fund Transfer	0.0	0.0014447167682623832
2006_Q4_Actual	20013	ATM_SAV_Withdrawals	ATM_SAV_Withdrawals	0.0	0.0070761576843764015
2006_Q4_Actual	20014	ATM_TRM_Deposits	ATM_TRM_Deposits	0.0	7.436121552686411E-5
2006_Q4_Actual	20015	ATM_TRM_Withdrawals	ATM_TRM_Withdrawals	0.0	0.0015935121106617832
2006_Q4_Actual	20058	None_None_None	None_None_None	6152.4220541...	0.0
2006_Q4_Actual	21001	BRH_CHK_Inquiry	BRH_CHK_Inquiry	2238.3197977...	0.0
2006_Q4_Actual	21002	BRH_CRC_Inquiry	BRH_CRC_Inquiry	6651.4938345...	0.0
2006_Q4_Actual	21003	BRH_FBP_Inquiry	BRH_FBP_Inquiry	12508.491287...	0.0
2006_Q4_Actual	21004	BRH_OVD_Inquiry	BRH_OVD_Inquiry	4213.9720396...	0.0
2006_Q4_Actual	21005	BRH_REC_Inquiry	BRH_REC_Inquiry	769.31380737...	0.0
2006_Q4_Actual	21006	BRH_SAV_Inquiry	BRH_SAV_Inquiry	6325540...	0.0
2006_Q4_Actual	007	BRH_SCR_In	BRH_SCR_I	31.5095	
2006_Q4	09				

The following graphic shows how a behavior table is related to other files:



Period Dimension

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 dimension table can contain the following fields:

Name	Maximum Length	Description
ID	Char 32	The identifying value for the period *Note:* The name must be ID.
L1_Scenario	Char 32	The top-level dimension member value (for example, Actual or Budget)
L2_Year	Char 32	The second-level dimension member value (for example, 2008)
L3_Quarter	Char 32	The third-level dimension member value (for example, 2008_Q4)

Note:

- The order of the columns is significant.
- The name of the columns is arbitrary, except for ID.
- The number of levels is arbitrary.

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

	ID	L1_Scenario	L2_Year	L3_Quarter
1	2006_Q4_Actual	Actual	2006	2006_Q4
2	2006_Q4_Budget	Budget	2006	2006_Q4

Custom Dimensions

Overview

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.

Note: For every custom dimension, the name of the ID column must be ID. The names of the other columns are your choice.

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:

L1_Channel
ATM
BRH
Call Center

The following graphic shows the entire table contents:

ID	L1_Channel
ATM	ATM
BRH	BRH
CCT	Call Center

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 bsiness. 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.

L1_Type	L2_LastName	L3_FirstName
Personal	Adcock	James
	Farm	Peter T.
	Aden	Shelly
Business	Logic Consulting	
	Metrogas SA	
	FASHBI Worldwide	

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

ID	L1_Type	L2_LastName	L3_FirstName
00040	Personal	Adcox	James
00041	Personal	Farm	Peter T.
00042	Personal	Aden	Shelly
00043	Personal	Desai	Sanjay
00044	Personal	Fosher	Monte
00045	Personal	Armstrong	Brian
00046	Personal	Devries	Tom
00047	Personal	Furrer	Allen
00048	Personal	Burdt	Lawrence J.
00049	Personal	Dickson	Charles
00050	Personal	Gunn	Henry
05798	Business	Logic Consulting	
05799	Business	Metrogas SA	
05800	Business	FASBHI Worldwide	
05801	Business	FASBHI Worldwide	
05802	Business	Post zip	
05803	Busi	Jess nd Consulting	
05804			

DIM_CUSTYPE

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

L1_CType	L2_CType
Personal	Retail Consumer Banking
	Small Business Banking
Business	Private Banking
	Corporate Banking

The following graphic shows the entire contents of DIM_CUSTYPE:

ID	L1_CType	L2_CType
RCB	Personal	Retail Consumer Banking
SBB	Business	Small Business Banking
PRB	Personal	Private Banking
CRB	Business	Corporate Banking

DIM_PRODUCT

The follow table represents the contents of DIM_PRODUCT:

L1_Product	L2_Product
Credit Products	Secured/Mortgages
	Unsecured

L1_Product	L2_Product
Deposit Products	Checking
	Recurring
	Savings
	Term
Fee-Based Products	
Other Products	
Revolving Credit Products	Credit Cards
	Overdrafts
Third-Party Products	

The following graphic shows the entire contents of DIM_PRODUCT:

ID	L1_Product	L2_Product
CRP	Credit Products	
DEP	Deposit Products	
FBP	Fee Based Products	
OTP	Other Products	
RCP	Revolving Credit Products	
TPP	Third Party Products	
SCR	Credit Products	Secured/Mortgages
UCR	Credit Products	Un secured
CHK	Deposit Products	Checking
REC	Deposit Products	Recurring
SAV	Deposit Products	Savings
TRM	Deposit Products	Term
CRC	Revolving Credit Products	Credit Cards
OVD	Revolving Credit Products	Overdrafts

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.

Area	Country	State	City
North America	Canada	Ontario	Don Mills
			Metcalfe
		Quebec	St. Hubert Quebec
			St. Laurent
			Town of Mont Royal
		Saskatchewan	Sheffield
	Mexico	Coahuila	Saltillo
		Mexico	Cuauhtemoc
			Colinia Granada
			Polanco
South America	Brazil	Brazil	Uberlandia - Mg
		Pe Brazil	Itapissuma
		Sao Paulo	Ribeirao Preto
			Sao Paulo
	Costa Rica	Costa Rica	San Jose
	Ecuador	Ecuador	Guayaquil
	Uruguay	Montevideo	Montevideo
	Venezuela	Caracas	Altamira

The following graphic shows the corresponding part of the actual table contents:

ID	L1_Area	L2_Country	L3_State	L4_City
Reg_156	North America	Canada	Ontario	Don Mills
Reg_163	North America	Canada	Ontario	Metcalfe
Reg_183	North America	Canada	Quebec	St. Hubert Quebec
Reg_186	North America	Canada	Quebec	St. Laurent
Reg_188	North America	Canada	Quebec	Town Of Mont Royal
Reg_191	North America	Canada	Saskatchewan	Sheffield
Reg_193	North America	Mexico	Coahuila	Saltillo
Reg_194	North America	Mexico	Mexico	Cuauhtemoc
Reg_196	North America	Mexico	Mexico	Colinia Granada
Reg_202	North America	Mexico	Mexico	Polanco
Reg_215	South America	Brazil	Brazil	Uberlandia - Mg
Reg_218	South America	Brazil	Pe Brazil	Itapissuma
Reg_220	South America	Brazil	Sao Paulo	Ribeirao Preto
Reg_221	South America	Brazil	Sao Paulo	Sao Paulo
Reg_229	South America	Costa Rica	Costa Rica	San Jose
Reg_230	South America	Ecuador	Ecuador	Guayaquil
Reg_236	South America	Uruguay	Montevideo	Montevideo
Reg_237	South America	Venezuela	Caracas	Altamira
Reg_245	USA_NE	USA	Connecticut	Enfield
Reg_246	USA_NE	USA	Connecticut	Hartford
Reg_248	USA_NE	USA	Connecticut	Milford

Rule Definition Table

Overview

A rule provides

- selection criteria that determine which rows are selected in the transaction tables for calculation of values (revenue or cost).
- a 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 they can be imported from a 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:

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

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).

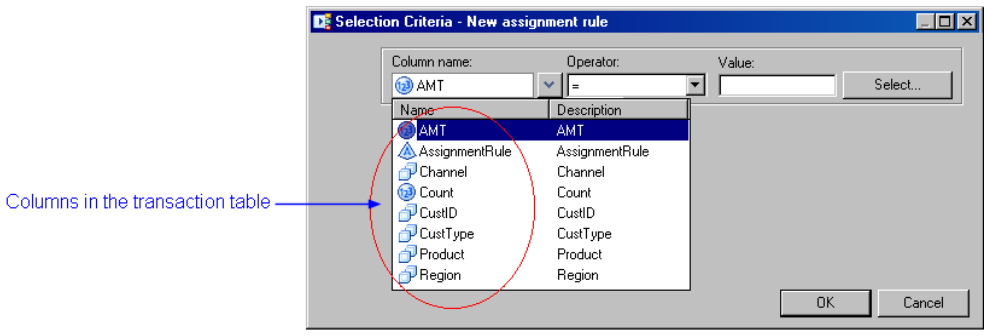
The following graphic shows the rule import table, RULES, for the Baby Bank model:

	TableGroup	AssignmentRule	SelectionCriteria	DriverFormula
1	ABMCost	buzz	AssignmentRule = 'buzz'	Count/Amt
2	ABMCost	ATM_CHK_Check balance	AssignmentRule = 'ATM_CHK_Check balance'	AMT
3	ABMCost	ATM_CHK_Deposits	AssignmentRule = 'ATM_CHK_Deposits'	AMT
4	ABMCost	ATM_CHK_Fund Transfer	AssignmentRule = 'ATM_CHK_Fund Transfer'	AMT
5	ABMCost	ATM_CHK_Requests	IsChildOf(Channel, '[DIM_CHANNEL].[ATM]') AND IsChildOf(Product, '[DI...	Count * .05 +AMT * .01
6	ABMCost	ATM_CHK_Withdrawals	AssignmentRule = 'ATM_CHK_Withdrawals'	AMT
7	ABMCost	ATM_CRC_Withdrawals	AssignmentRule = 'ATM_CRC_Withdrawals'	AMT
8	ABMCost	ATM_REC_Deposits	AssignmentRule = 'ATM_REC_Deposits'	AMT
9	ABMCost	ATM_REC_Requests	IsChildOf(Channel, '[DIM_CHANNEL].[ATM]') AND IsChildOf(Product, '[DI...	Count
10	ABMCost	ATM_SAV_Check balance	AssignmentRule = 'ATM_SAV_Check balance'	AMT
11	ABMCost	ATM_SAV_Deposits	AssignmentRule = 'ATM_SAV_Deposits'	AMT
12	ABMCost	ATM_SAV_Fund Transfer	AssignmentRule = 'ATM_SAV_Fund Transfer'	AMT
13	ABMCost	ATM_SAV_Requests	IsChildOf(Channel, '[DIM_CHANNEL].[ATM]') AND IsChildOf(Product, '[DI...	
14	ABMCost	ATM_SAV_Withdrawals	AssignmentRule = 'ATM_SAV_Withdrawals'	
15	ABMCost	ATM_TRM_Deposits	Assign... ATM_TRM_Deposits'	Complaints
16	ABMCost	ATM_TRM_Request...	...Center]') AND IsChildOf(Produ...	CrossSell
17	ABMCost		IsChildOf(Ch... ...CHANNEL].[Call Center]') AND IsChildOf(Produ...	Inquiry
		...Requests	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Produ...	Requests
	...nter	CCT_TPP_Cross_Up Sell	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Produ...	CrossSell
84	CallCenter	CCT_TRM_Complaints	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Produ...	Complaints
85	CallCenter	CCT_TRM_Inquiry	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Produ...	Inquiry
86	CallCenter	CCT_TRM_Requests	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Produ...	Requests
87	CallCenter	CCT_UCR_Complaints	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Produ...	Complaints
88	CallCenter	CCT_UCR_Cross_Up Sell	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Produ...	CrossSell
89	CallCenter	CCT_UCR_Inquiry	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Produ...	Inquiry
90	CallCenter	CCT_UCR_Requests	IsChildOf(Channel, '[DIM_CHANNEL].[Call Center]') AND IsChildOf(Produ...	Requests
91	Revenue	ATM Fees	ID = '12002'	AMT
92	Revenue	Certificates of Deposit Payme...	ID = '11002'	AMT
93	Revenue	Charge For Funds	ID = '13002'	AMT
94	Revenue	Checking Account Fees	ID = '12004'	AMT
95	Revenue	Credit Card Fees	ID = '12001'	AMT
96	Revenue	Credit Card interest Income	ID = '10001'	AMT
97	Revenue	Credit for Funds	ID = '13001'	AMT
98	Revenue	Investment Account Fees	ID = '12003'	AMT
99	Revenue	Investment Securities Payme...	ID = '11003'	AMT
100	Revenue	Loan Interest Income	ID = '10002'	AMT
101	Revenue	Mortgages Income	ID = '10003'	AMT
102	Revenue	Provision For Losses	ID = '14001'	AMT
103	Revenue	Savings Interest Payments	ID = '11001'	AMT

Rule Selection Criteria

Overview

When you define a rule, you select columns in a transaction table and specify conditions on those columns for a row to be selected.

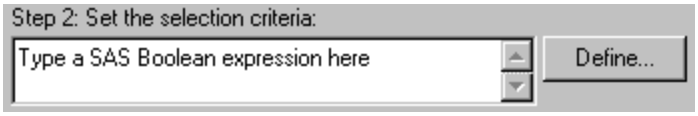


The following table shows what sort of conditions you can specify depending on the type of column that you select:

Table 15.1 *temp title*

If the selected column is:	You can use the following operators:
Text	=, not = match Behavior
Numeric	=, not =, <, <=, >, >= match Behavior
Dimension	is child of match Behavior

Instead of selecting columns from a drop-down list, you can simply type a SAS Boolean expression into the selection criteria field.



Following is an example of a SAS boolean expression that you would have to type into the selection criteria field.

(IF customer ='C00650' THEN (0.54*Distance) ELSE (0.23*Distance)) > 1000

Note: Every field referenced in such an expression is in the transaction table.

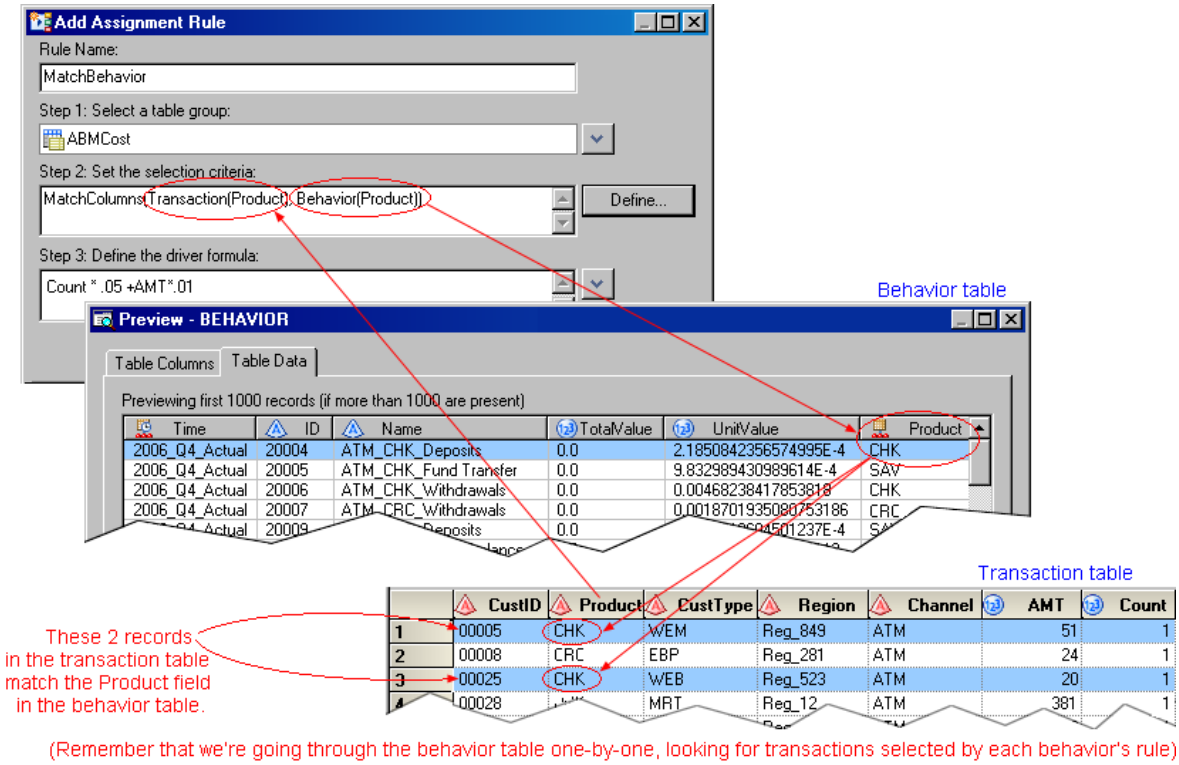
Match Behavior

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 Product column 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.



Is Child Of

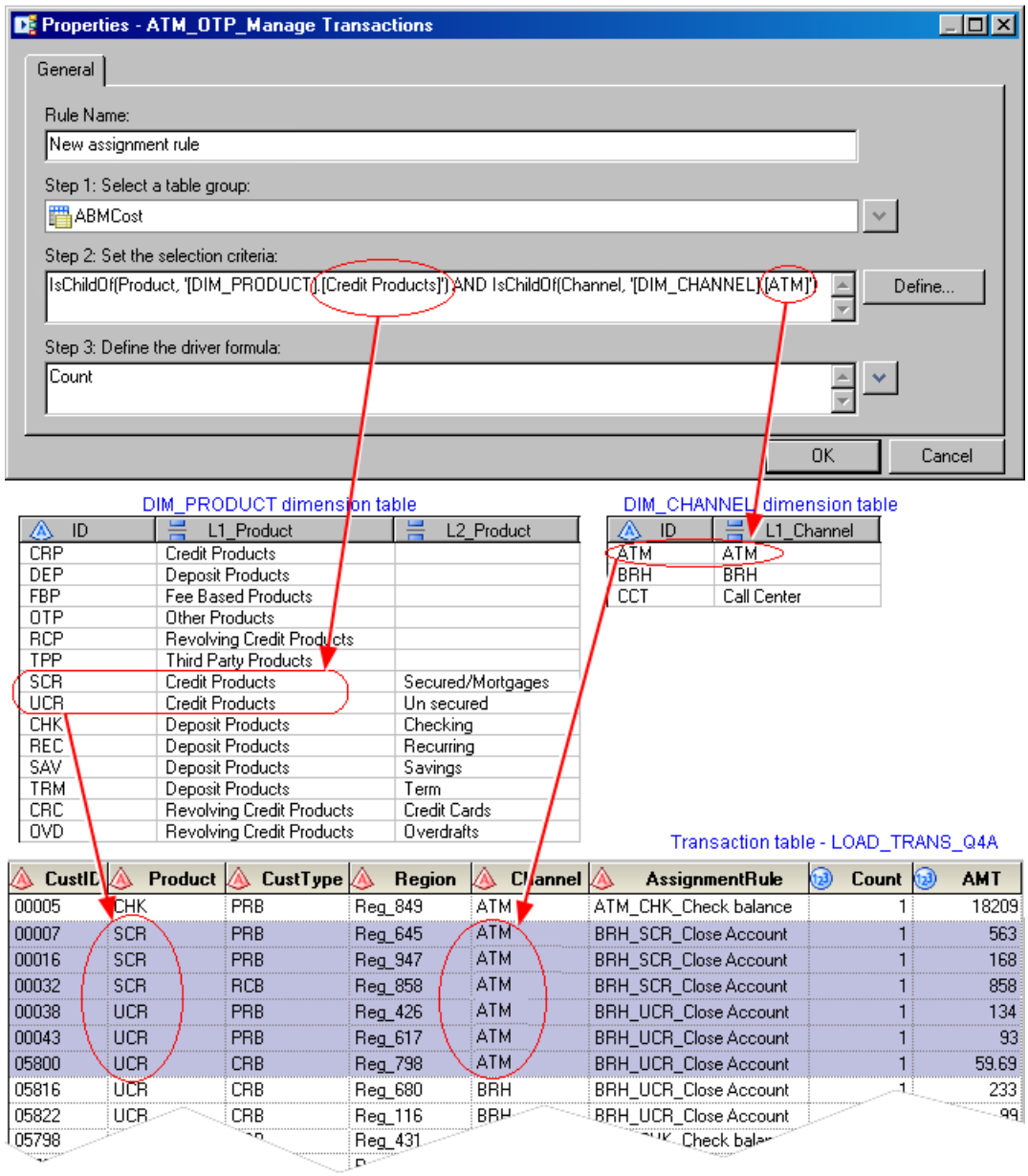
Syntax	**IsChildOf (source column, [dimension table].[value])**
Example	IsChildOf(channel, '[dim_channel].[ATM]')
Example	IsChildOf(channel,'[dim_channel].[ATM]') And IsChildOf(product,'[dim_product].[deposit products].[checking]')

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 choose 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.



Rule Driver Formula

Overview

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) *Note:* The parentheses after the ELSE operator are required because ELSE binds tighter than the * numeric operator (omitting the parentheses would result in the expression being parsed differently than intended).

Rule/Behavior Associations

The association table correlates 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.

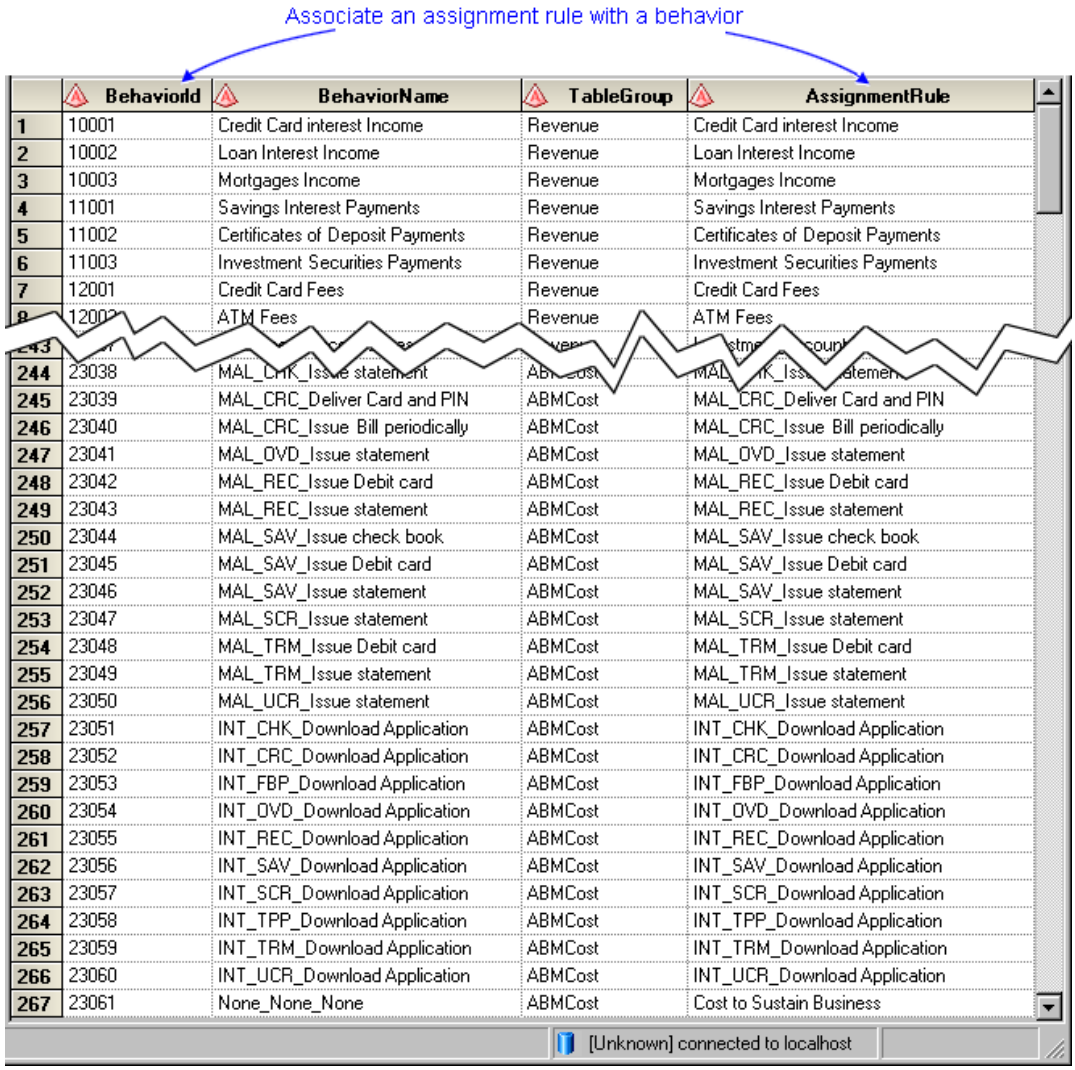
The rule association table contains the following fields:

Name	Maximum Length	Description
BehaviorId	Char 32	The ID of the behavior table row
TableGroup	Char 64	The table group to which the rule is applied
AssignmentRule	Char 64	The name of the associated assignment rule. This name must be unique within the model

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 association table, ASSOCATIONS, for the Baby Bank model:



Report Hierarchy

Overview

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 of the business. A model can have multiple report hierarchies.

The report hierarchy table contains the following fields:

Position	Name	Maximum Length	Description
1	**ID**	Char 32	The ID of the behavior table row for this member *Note:* The name of the ID column must be ID.
2	**L1_Profit**	Char 32	1st level dimension members for the hierarchy
3	**L2_Profit**	Char 32	2nd level dimension members for the hierarchy
4	**L3_Profit**	Char 32	3rd level dimension members for the hierarchy
5	**L4_Profit**	Char 32	4th level dimension members for the hierarchy
6	**L5_Profit**	Char 32	5th level dimension members for the hierarchy

Note:

- The order of the columns is significant.
- The name of the columns is arbitrary, except for the ID column.
- The number of levels is arbitrary.

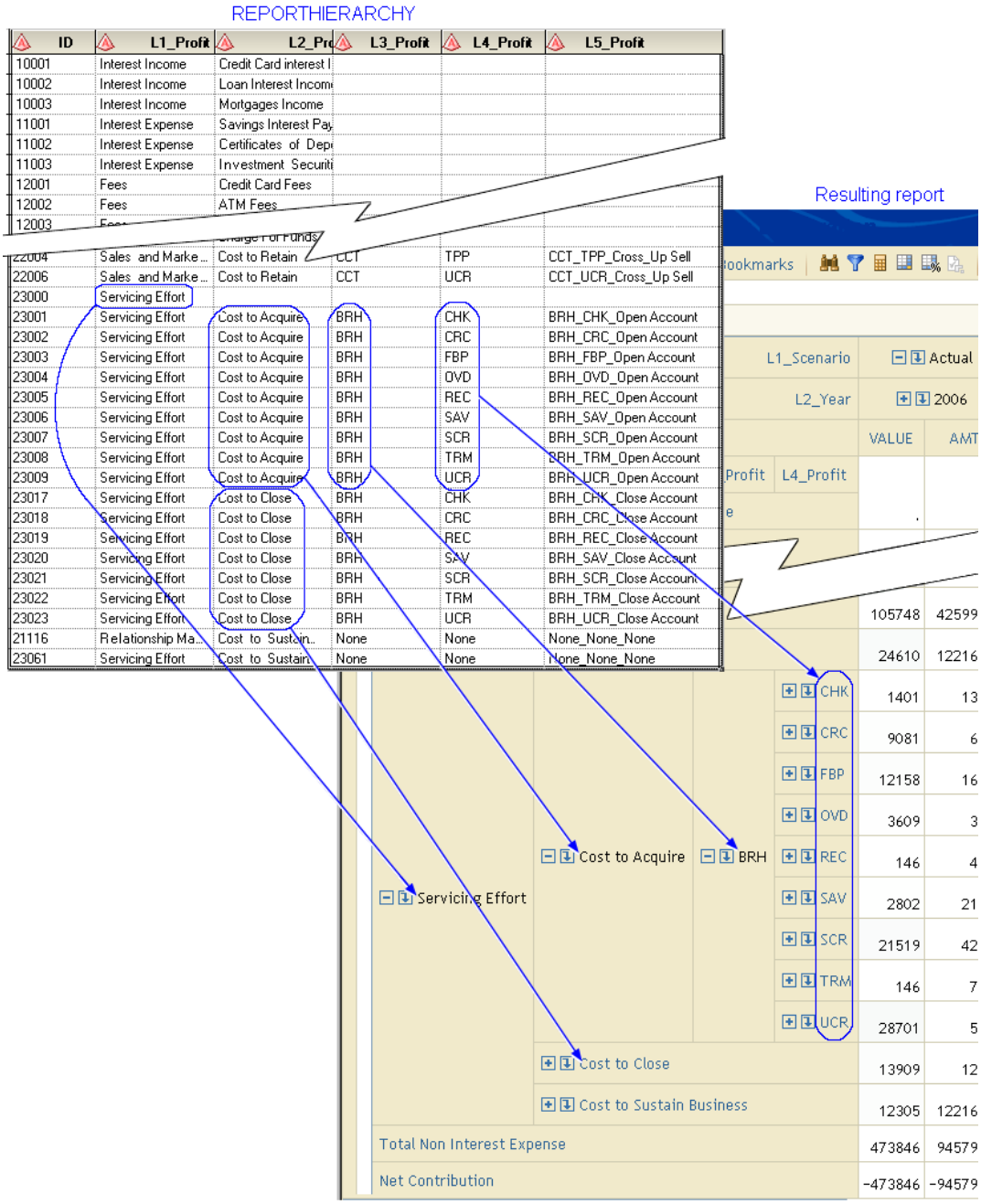
The following graphic shows the report hierarchy table for the Baby Bank model. Notice that it is ragged—different items have different drill-down depths.

	ID	L1_Profit	L2_Profit	L3_Profit	L4_Profit	L5_Profit
1	10001	Interest Income	Credit Card interest Inco...			
2	10002	Interest Income	Loan Interest Income			
3	10003	Interest Income	Mortgages Income			
4	11001	Interest Expense	Savings Interest Payments			
5	11002	Interest Expense	Certificates of Deposit Pa...			
6	11003	Interest Expense	Investment Securities Pa...			
7	12001	Fees	Credit Card Fees			
8	12002	Fees	ATM Fees			
9	12003	Fees	Investment Account Fees			
10	12004	Fees	Checking Account Fees			
11	13001	Funds	Credit for Funds			
12	13002	Funds	Charge For Funds			
13	14001	Provision For Losses				
14	20000	Direct Product				
15	20003	Direct Product	Cost to Provide	ATM	CHK	ATM_CHK_Check balance
16	20004	Direct Product	Cost to Provide	ATM	CHK	ATM_CHK_Deposits
17	20005	Direct Product		ATM		
18	20006					
			Cost to Retain		TPP	CCT_TPP_Cross_Up Sell
		Sales and Marketing Effort	Cost to Retain	CCT	UCR	CCT_UCR_Cross_Up Sell
96	23000	Servicing Effort				
97	23001	Servicing Effort	Cost to Acquire	BRH	CHK	BRH_CHK_Open Account
98	23002	Servicing Effort	Cost to Acquire	BRH	CRC	BRH_CRC_Open Account
99	23003	Servicing Effort	Cost to Acquire	BRH	FBP	BRH_FBP_Open Account
100	23004	Servicing Effort	Cost to Acquire	BRH	OVD	BRH_OVD_Open Account
101	23005	Servicing Effort	Cost to Acquire	BRH	REC	BRH_REC_Open Account
102	23006	Servicing Effort	Cost to Acquire	BRH	SAV	BRH_SAV_Open Account
103	23007	Servicing Effort	Cost to Acquire	BRH	SCR	BRH_SCR_Open Account
104	23008	Servicing Effort	Cost to Acquire	BRH	TRM	BRH_TRM_Open Account
105	23009	Servicing Effort	Cost to Acquire	BRH	UCR	BRH_UCR_Open Account
106	23017	Servicing Effort	Cost to Close	BRH	CHK	BRH_CHK_Close Account
107	23018	Servicing Effort	Cost to Close	BRH	CRC	BRH_CRC_Close Account
108	23019	Servicing Effort	Cost to Close	BRH	REC	BRH_REC_Close Account
109	23020	Servicing Effort	Cost to Close	BRH	SAV	BRH_SAV_Close Account
110	23021	Servicing Effort	Cost to Close	BRH	SCR	BRH_SCR_Close Account
111	23022	Servicing Effort	Cost to Close	BRH	TRM	BRH_TRM_Close Account
112	23023	Servicing Effort	Cost to Close	BRH	UCR	BRH_UCR_Close Account
113	21116	Relationship Management	Cost to Sustain Business	None	None	None_None_None
114	23061	Servicing Effort	Cost to Sustain Business	None	None	None_None_None

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.

Report Hierarchy and the 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:

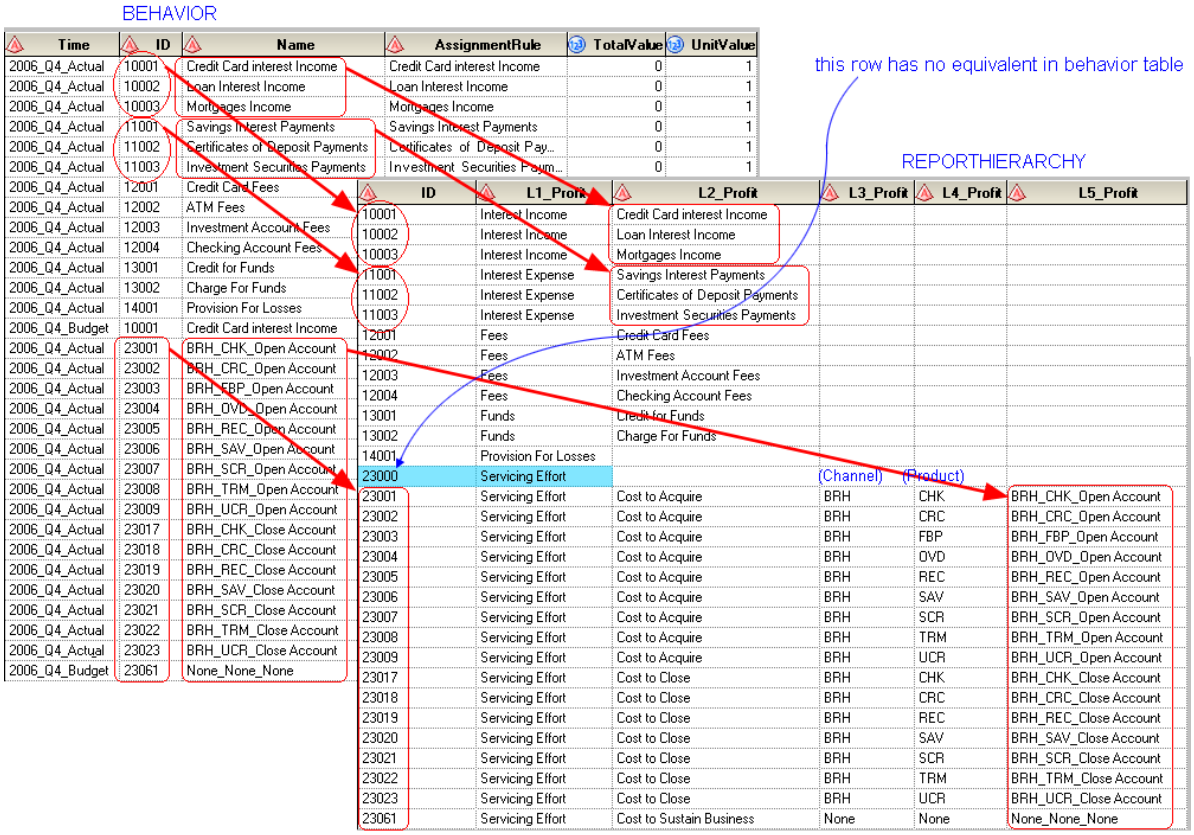


Behavior Table and Report Hierarchy

The following graphic illustrates the relationship between the behavior table and report hierarchy. In it, 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. In short, the report hierarchy table is ragged (variable length rows).

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.

It is also important to note that some rows in the report hierarchy have no corresponding row in the behavior table (although every row in the behavior table has a corresponding row in the report hierarchy). For example, the row with the ID of 23000 in the report hierarchy has no corresponding row in the behavior table. That row in the report hierarchy serves to group the items logically below it in the hierarchy.



Time	ID	Name	AssignmentRule	TotalValue	UnitValue
2006_Q4_Actual	10001	Credit Card interest Income	Credit Card interest Income	0	1
2006_Q4_Actual	10002	Loan Interest Income	Loan Interest Income	0	1
2006_Q4_Actual	10003	Mortgages Income	Mortgages Income	0	1
2006_Q4_Actual	11001	Savings Interest Payments	Savings Interest Payments	0	1
2006_Q4_Actual	11002	Certificates of Deposit Payments	Certificates of Deposit Pay...	0	1
2006_Q4_Actual	11003	Investment Securities Payments	Investment Securities Paym...	0	1
2006_Q4_Actual	12001	Credit Card Fees			
2006_Q4_Actual	12002	ATM Fees			
2006_Q4_Actual	12003	Investment Account Fees			
2006_Q4_Actual	12004	Checking Account Fees			
2006_Q4_Actual	13001	Credit for Funds			
2006_Q4_Actual	13002	Charge For Funds			
2006_Q4_Actual	14001	Provision For Losses			
2006_Q4_Budget	10001	Credit Card interest Income			
2006_Q4_Actual	23001	BRH_CHK_Open Account			
2006_Q4_Actual	23002	BRH_CRC_Open Account			
2006_Q4_Actual	23003	BRH_FBP_Open Account			
2006_Q4_Actual	23004	BRH_OVD_Open Account			
2006_Q4_Actual	23005	BRH_REC_Open Account			
2006_Q4_Actual	23006	BRH_SAV_Open Account			
2006_Q4_Actual	23007	BRH_SCR_Open Account			
2006_Q4_Actual	23008	BRH_TRM_Open Account			
2006_Q4_Actual	23009	BRH_UCR_Open Account			
2006_Q4_Actual	23017	BRH_CHK_Close Account			
2006_Q4_Actual	23018	BRH_CRC_Close Account			
2006_Q4_Actual	23019	BRH_REC_Close Account			
2006_Q4_Actual	23020	BRH_SAV_Close Account			
2006_Q4_Actual	23021	BRH_SCR_Close Account			
2006_Q4_Actual	23022	BRH_TRM_Close Account			
2006_Q4_Actual	23023	BRH_UCR_Close Account			
2006_Q4_Budget	23061	None_None_None			

ID	L1_Profit	L2_Profit	L3_Profit	L4_Profit	L5_Profit
10001	Interest Income	Credit Card interest Income			
10002	Interest Income	Loan Interest Income			
10003	Interest Income	Mortgages Income			
11001	Interest Expense	Savings Interest Payments			
11002	Interest Expense	Certificates of Deposit Payments			
11003	Interest Expense	Investment Securities Payments			
12001	Fees	Credit Card Fees			
12002	Fees	ATM Fees			
12003	Fees	Investment Account Fees			
12004	Fees	Checking Account Fees			
13001	Funds	Credit for Funds			
13002	Funds	Charge For Funds			
14001	Provision For Losses				
23000	Servicing Effort		(Channel)	(Product)	
23001	Servicing Effort	Cost to Acquire	BRH	CHK	BRH_CHK_Open Account
23002	Servicing Effort	Cost to Acquire	BRH	CRC	BRH_CRC_Open Account
23003	Servicing Effort	Cost to Acquire	BRH	FBP	BRH_FBP_Open Account
23004	Servicing Effort	Cost to Acquire	BRH	OVD	BRH_OVD_Open Account
23005	Servicing Effort	Cost to Acquire	BRH	REC	BRH_REC_Open Account
23006	Servicing Effort	Cost to Acquire	BRH	SAV	BRH_SAV_Open Account
23007	Servicing Effort	Cost to Acquire	BRH	SCR	BRH_SCR_Open Account
23008	Servicing Effort	Cost to Acquire	BRH	TRM	BRH_TRM_Open Account
23009	Servicing Effort	Cost to Acquire	BRH	UCR	BRH_UCR_Open Account
23017	Servicing Effort	Cost to Close	BRH	CHK	BRH_CHK_Close Account
23018	Servicing Effort	Cost to Close	BRH	CRC	BRH_CRC_Close Account
23019	Servicing Effort	Cost to Close	BRH	REC	BRH_REC_Close Account
23020	Servicing Effort	Cost to Close	BRH	SAV	BRH_SAV_Close Account
23021	Servicing Effort	Cost to Close	BRH	SCR	BRH_SCR_Close Account
23022	Servicing Effort	Cost to Close	BRH	TRM	BRH_TRM_Close Account
23023	Servicing Effort	Cost to Close	BRH	UCR	BRH_UCR_Close Account
23061	Servicing Effort	Cost to Sustain Business	None	None	None_None_None

Report Layout

Overview

The report layout defines the calculation logic for the profit and loss statement. The report layout controls how the OLAP viewer displays the cube when it is first opened. More specifically, the report layout determines:

- The order in which columns are displayed
- Whether fields are initially displayed expanded or collapsed.

  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.

A report hierarchy can have multiple report layouts so that you can create different reports from the same data.

A report layout table contains the following fields:

Position	Name	Maximum Length	Description
1	**ID**	Char 32	The ID of the behavior table row for this member
2	**Name**	Char 32	The name of the behavior for display on the report
3	**Formula**	Char 255	For calculated rows, the formula for the calculation based on the IDs. The formula can include addition, subtraction, multiplication, and division (+, -, *, /). *Note:* Formula fields must *not* also appear in the report hierarchy.
4	**RowOrder**	Numeric 8	The position of the row in the profit-and-loss report

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.

	ID	Name	Formula	RowOrder
1	10001	Credit Card interest Income		1
2	10002	Loan Interest Income		2
3	10003	Mortgages Income		3
4	TOT-INT-INC	Total Interest Income	[10001]+[10002]+[10003]	4
5	11001	Savings Interest Payments		5
6	11002	Certificates of Deposit Payments		6
7	11003	Investment Securities Payments		7
8	TOT-INT-EXP	Total Interest Expense	[11001]+[11002]+[11003]	8
9	NET-INT-INC	Net Interest Income	[TOT-INT-INC]-[TOT-INT-EXP]	9
10	13001	Credit for Funds		10
11	13002	Charge For Funds		11
12	NET-FUNDS	Net Funds	[13001]-[13002]	12
13	12001	Credit Card Fees		13
14	12002	ATM Fees		14
15	12003	Investment Account Fees		15
16	12004	Checking Account Fees		16
17	TOT-NON-INT-INC	Total Non Interest Income	[12001]+[12002]+[12003]+[12004]	17
18	14001	Provision For Losses		18
19	20000	Direct Product		19
20	21000	Relationship Management		20
21	22000	Sales and Marketing Effort		21
22	23000	Servicing Effort		22
23	TOT-NON-INT-EXP	Total Non Interest Expense	[14001]+[20000]+[21000]+[22000]+[23000]	23
24	NET_CONTRIB	Net Contribution	[NET-INT-INC]+[NET-FUNDS]+[TOT-NON-INT-INC]-[TOT-NON-INT-EX...	24

Report Layout and the Resulting Report

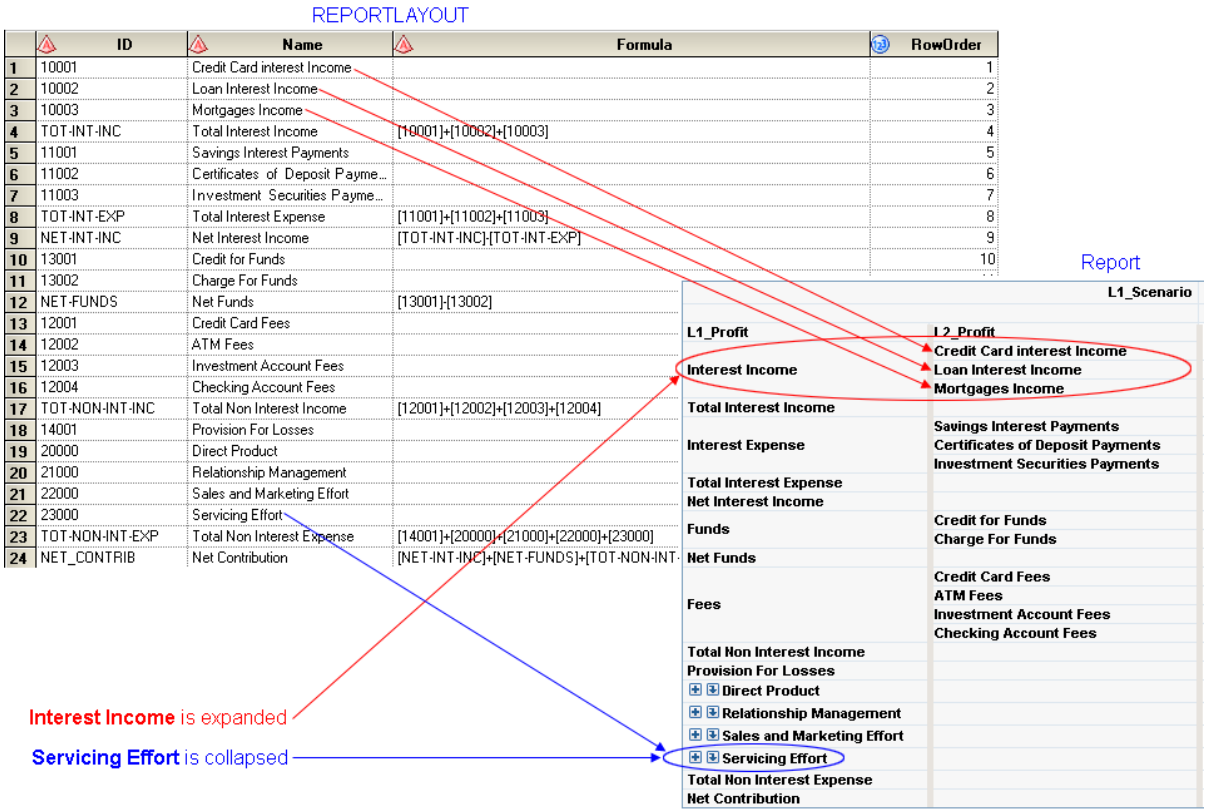
The report layout controls how the OLAP viewer displays the cube when it is first opened. You can see in the following graphic that the report layout table controls:

- the order in which columns are displayed. REPORTLAYOUT causes income variables to display before expense variables.
- whether dimensions initially display expanded or collapsed. You can see in the following graphic that the Servicing Effort dimension is displayed collapsed because

it is included in REPORTLAYOUT without any of its children showing. By contrast, the Interest Income dimension 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.

- 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).

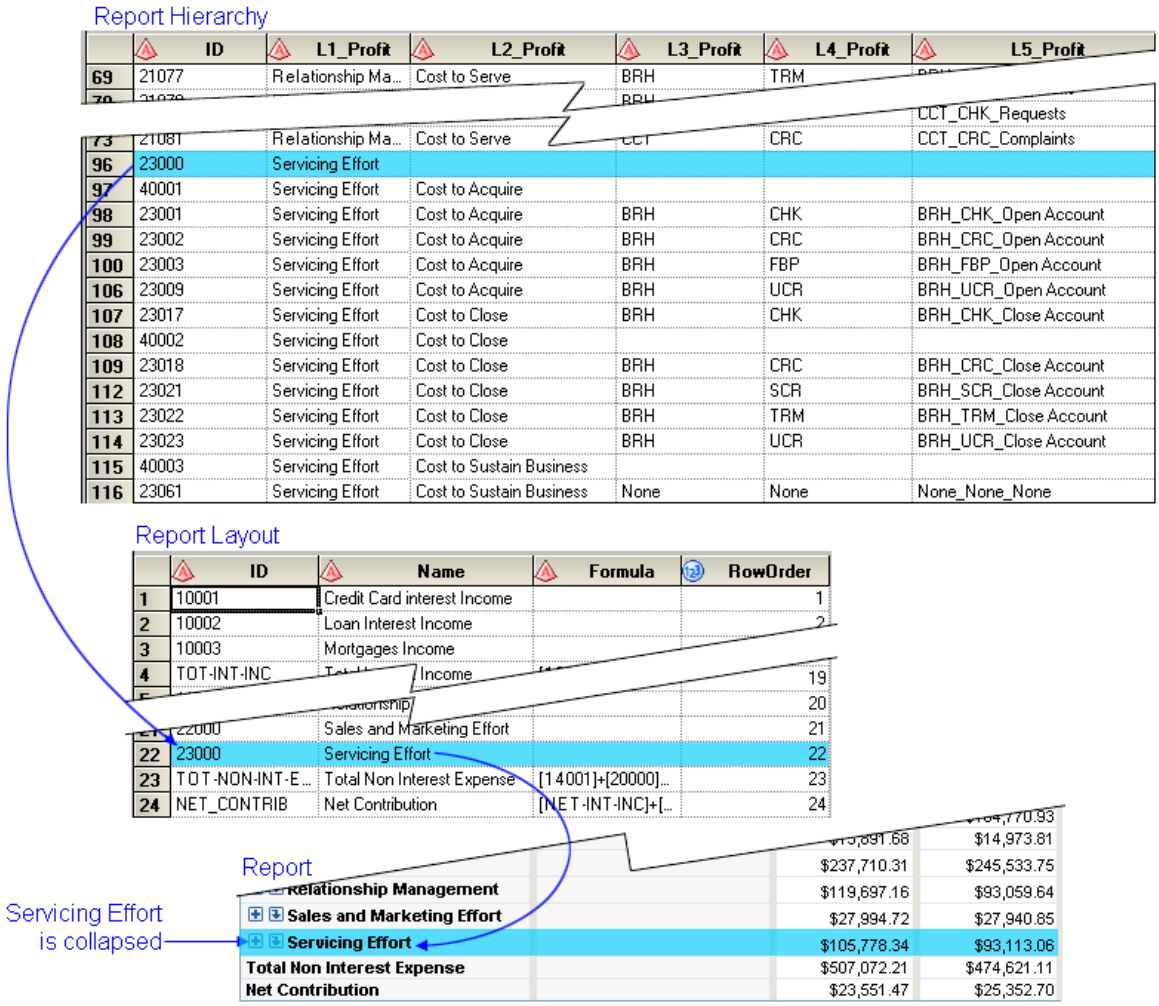


	ID	Name	Formula	RowOrder
1	10001	Credit Card interest Income		1
2	10002	Loan Interest Income		2
3	10003	Mortgages Income		3
4	TOT-INT-INC	Total Interest Income	[10001]+[10002]+[10003]	4
5	11001	Savings Interest Payments		5
6	11002	Certificates of Deposit Payme...		6
7	11003	Investment Securities Payme...		7
8	TOT-INT-EXP	Total Interest Expense	[11001]+[11002]+[11003]	8
9	NET-INT-INC	Net Interest Income	[TOT-INT-INC]-[TOT-INT-EXP]	9
10	13001	Credit for Funds		10
11	13002	Charge For Funds		
12	NET-FUNDS	Net Funds	[13001]-[13002]	
13	12001	Credit Card Fees		
14	12002	ATM Fees		
15	12003	Investment Account Fees		
16	12004	Checking Account Fees		
17	TOT-NON-INT-INC	Total Non Interest Income	[12001]+[12002]+[12003]+[12004]	
18	14001	Provision For Losses		
19	20000	Direct Product		
20	21000	Relationship Management		
21	22000	Sales and Marketing Effort		
22	23000	Servicing Effort		
23	TOT-NON-INT-EXP	Total Non Interest Expense	[14001]+[20000]+[21000]+[22000]+[23000]	
24	NET_CONTRIB	Net Contribution	[NET-INT-INC]+[NET-FUNDS]+[TOT-NON-INT-	

L1_Scenario	
L1_Profit	L2_Profit
Interest Income	Credit Card interest Income
	Loan Interest Income
	Mortgages Income
Total Interest Income	
Interest Expense	Savings Interest Payments
	Certificates of Deposit Payments
	Investment Securities Payments
Total Interest Expense	
Net Interest Income	
Funds	Credit for Funds
	Charge For Funds
Net Funds	
Fees	Credit Card Fees
	ATM Fees
	Investment Account Fees
	Checking Account Fees
Total Non Interest Income	
Provision For Losses	
Direct Product	
Relationship Management	
Sales and Marketing Effort	
Servicing Effort	
Total Non Interest Expense	
Net Contribution	

Let's take a closer look at how the report layout determines whether dimensions are initially displayed expanded or collapsed. The OLAP viewer always starts at the highest level of a dimension. The report layout specifies to what level that dimension is expanded when the report is initially opened.

The following graphic shows a report layout that causes the report to be collapsed up to the highest level of the Servicing Effort dimension. Notice that Servicing Effort has five hierarchical levels in the report hierarchy. The report layout specifies only the highest level—Servicing Effort. Therefore the resulting report shows the Servicing Effort dimension completely collapsed.

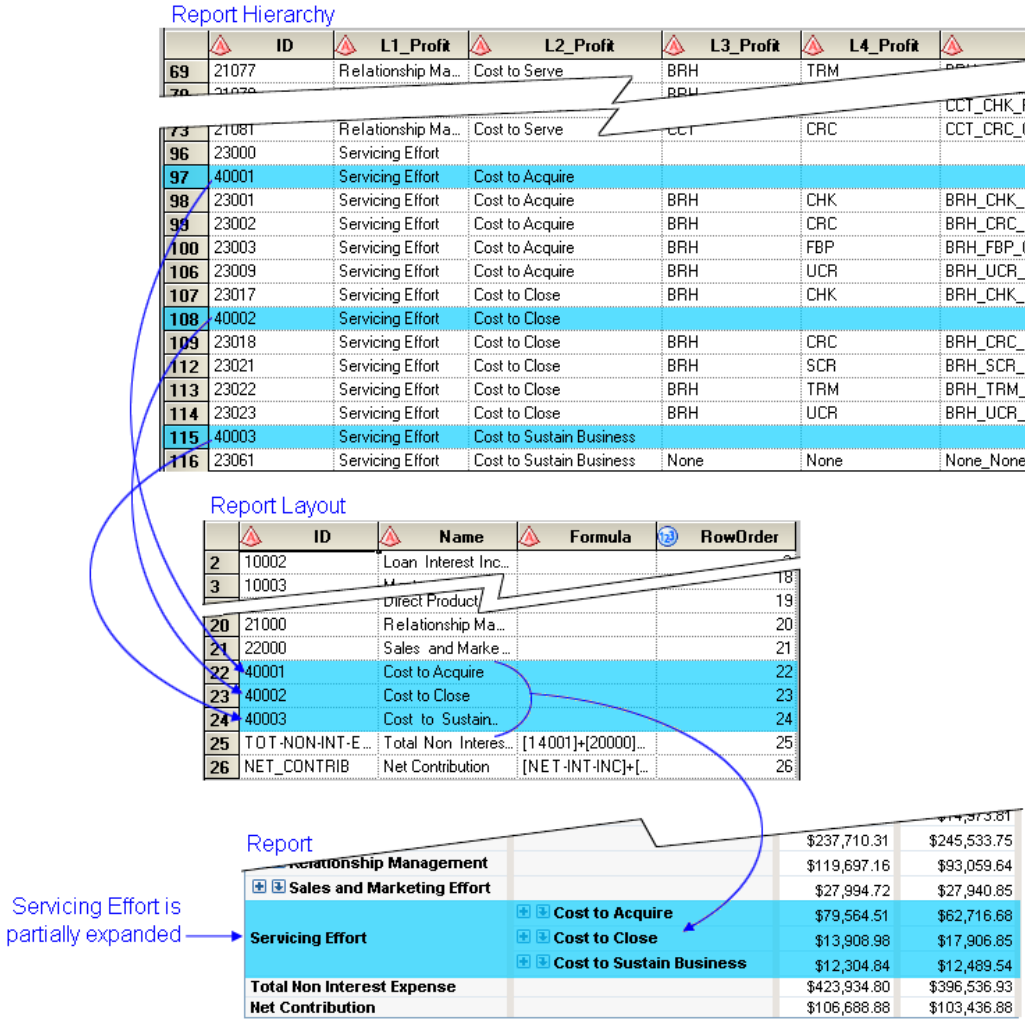


	ID	L1_Profit	L2_Profit	L3_Profit	L4_Profit	L5_Profit
69	21077	Relationship Ma...	Cost to Serve	BRH	TRM	
						CCT_CHK_Requests
96	23000	Servicing Effort				
97	40001	Servicing Effort	Cost to Acquire			
98	23001	Servicing Effort	Cost to Acquire	BRH	CHK	BRH_CHK_Open Account
99	23002	Servicing Effort	Cost to Acquire	BRH	CRC	BRH_CRC_Open Account
100	23003	Servicing Effort	Cost to Acquire	BRH	FBP	BRH_FBP_Open Account
106	23009	Servicing Effort	Cost to Acquire	BRH	UCR	BRH_UCR_Open Account
107	23017	Servicing Effort	Cost to Close	BRH	CHK	BRH_CHK_Close Account
108	40002	Servicing Effort	Cost to Close			
109	23018	Servicing Effort	Cost to Close	BRH	CRC	BRH_CRC_Close Account
112	23021	Servicing Effort	Cost to Close	BRH	SCR	BRH_SCR_Close Account
113	23022	Servicing Effort	Cost to Close	BRH	TRM	BRH_TRM_Close Account
114	23023	Servicing Effort	Cost to Close	BRH	UCR	BRH_UCR_Close Account
115	40003	Servicing Effort	Cost to Sustain Business			
116	23061	Servicing Effort	Cost to Sustain Business	None	None	None_None_None

	ID	Name	Formula	RowOrder
1	10001	Credit Card interest Income		1
2	10002	Loan Interest Income		2
3	10003	Mortgages Income		
4	TOT-INT-INC	Income		19
				20
	22000	Sales and Marketing Effort		21
22	23000	Servicing Effort		22
23	TOT-NON-INT-E...	Total Non Interest Expense	[14001]+[20000]...	23
24	NET_CONTRIB	Net Contribution	[NET-INT-INC]+[...	24

		$237,710.31	$245,533.75
Relationship Management		$119,697.16	$93,059.64
Sales and Marketing Effort		$27,994.72	$27,940.85
Servicing Effort		$105,778.34	$93,113.06
Total Non Interest Expense		$507,072.21	$474,621.11
Net Contribution		$23,551.47	$25,352.70

By contrast, the next graphic shows the report layout specifying an intermediate level of the Servicing Effort dimension. Consequently, in the resulting report, Servicing Effort is partially expanded down to the level specified in the report layout. By clicking the plus sign or down arrow, a user can navigate to lower levels of the Servicing Effort dimension.



Report Hierarchy

	ID	L1_Profit	L2_Profit	L3_Profit	L4_Profit	L5_Profit
69	21077	Relationship Ma...	Cost to Serve	BRH	TRM	
73	21081	Relationship Ma...	Cost to Serve	CCT	CRC	CCT_CHK_Requests CCT_CRC_Complaints
96	23000	Servicing Effort				
97	40001	Servicing Effort	Cost to Acquire			
98	23001	Servicing Effort	Cost to Acquire	BRH	CHK	BRH_CHK_Open Account
99	23002	Servicing Effort	Cost to Acquire	BRH	CRC	BRH_CRC_Open Account
100	23003	Servicing Effort	Cost to Acquire	BRH	FBP	BRH_FBP_Open Account
106	23009	Servicing Effort	Cost to Acquire	BRH	UCR	BRH_UCR_Open Account
107	23017	Servicing Effort	Cost to Close	BRH	CHK	BRH_CHK_Close Account
108	40002	Servicing Effort	Cost to Close			
109	23018	Servicing Effort	Cost to Close	BRH	CRC	BRH_CRC_Close Account
112	23021	Servicing Effort	Cost to Close	BRH	SCR	BRH_SCR_Close Account
113	23022	Servicing Effort	Cost to Close	BRH	TRM	BRH_TRM_Close Account
114	23023	Servicing Effort	Cost to Close	BRH	UCR	BRH_UCR_Close Account
115	40003	Servicing Effort	Cost to Sustain Business			
116	23061	Servicing Effort	Cost to Sustain Business	None	None	None_None_None

Report Layout

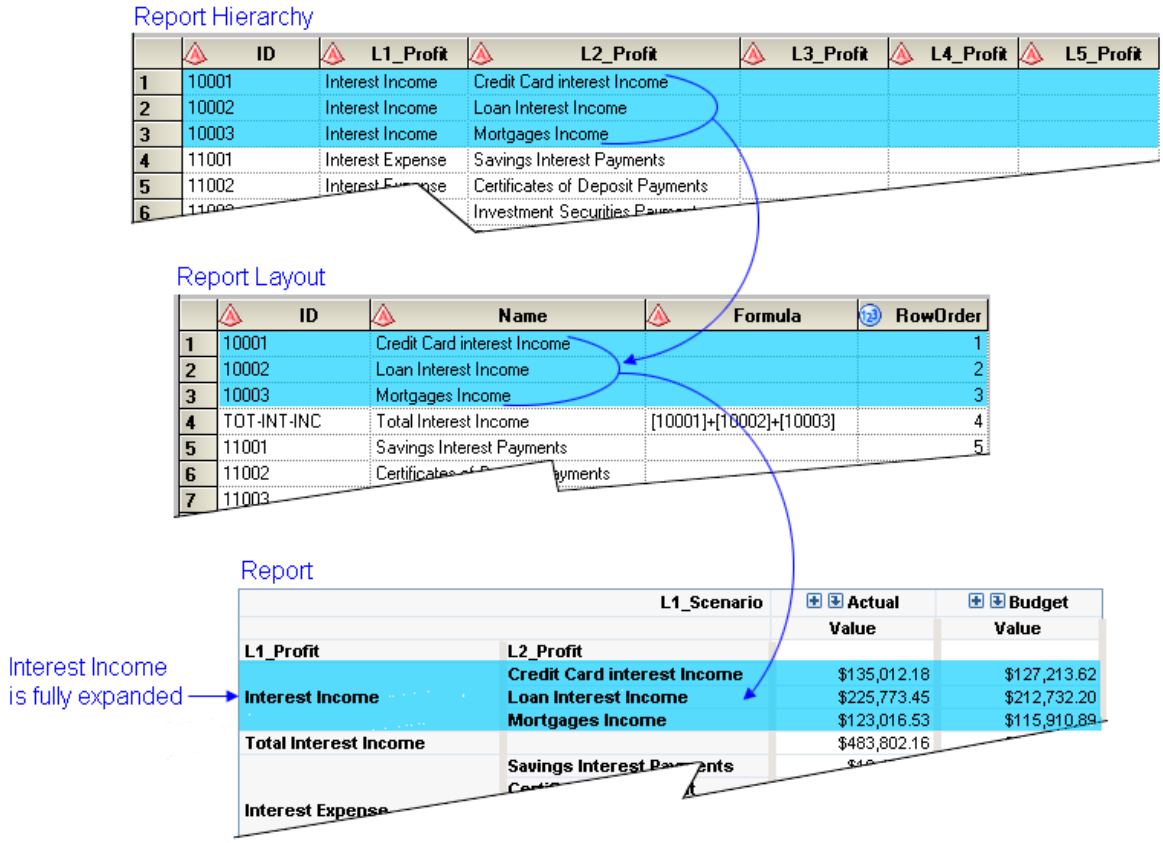
	ID	Name	Formula	RowOrder
2	10002	Loan Interest Inc...		
3	10003			18
		Direct Product		19
20	21000	Relationship Ma...		20
21	22000	Sales and Marke...		21
22	40001	Cost to Acquire		22
23	40002	Cost to Close		23
24	40003	Cost to Sustain...		24
25	TOT-NON-INT-E...	Total Non Interes...	[14001]+[20000]...	25
26	NET_CONTRIB	Net Contribution	[NET-INT-INC]+[...	26

Report

		$237,710.31	$245,533.75
Relationship Management		$119,697.16	$93,059.64
Sales and Marketing Effort		$27,994.72	$27,940.85
Servicing Effort	Cost to Acquire	$79,564.51	$62,716.68
	Cost to Close	$13,908.98	$17,906.85
	Cost to Sustain Business	$12,304.84	$12,489.54
Total Non Interest Expense		$423,934.80	$396,536.93
Net Contribution		$106,688.88	$103,436.88

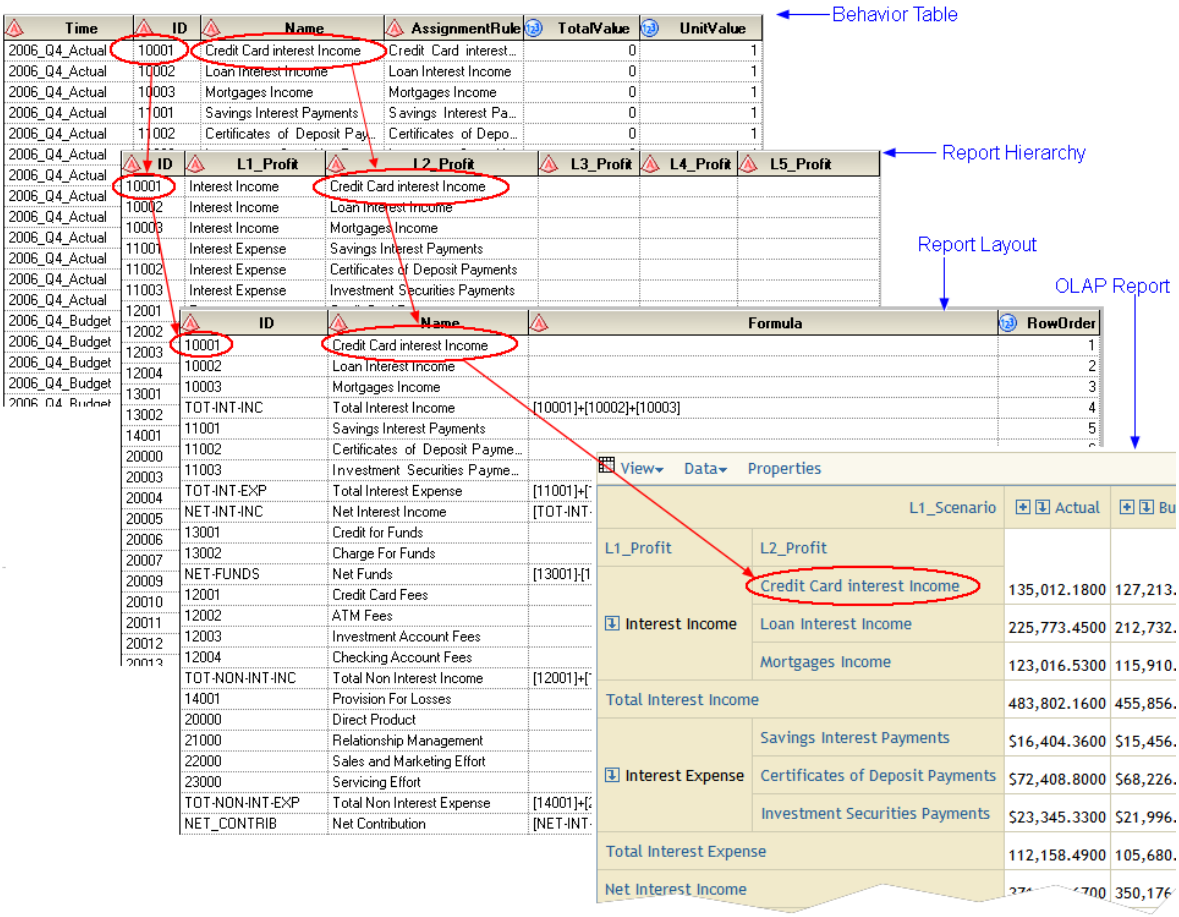
The following graphic shows another dimension, Interest Income, that is fully expanded in a report. The Interest Income dimension has only three members at its lowest level. In the following graphic, you can see the effect of specifying, in the report layout, the lowest level members of the Interest Income dimension. It fully expands that dimension in the resulting report.

Note: To expand a dimension fully in a report, specify in the report layout the lowest level members of the dimension—regardless of how many levels are in the dimension hierarchy.



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. Note that the ID in the behavior table matches the ID in the report hierarchy table and the report layout table. SAS Profitability Management also currently requires that the associated name in the report hierarchy table match the name in the report layout table, but the primary key is the **ID** field.



Transaction Tables

Overview

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 table 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.

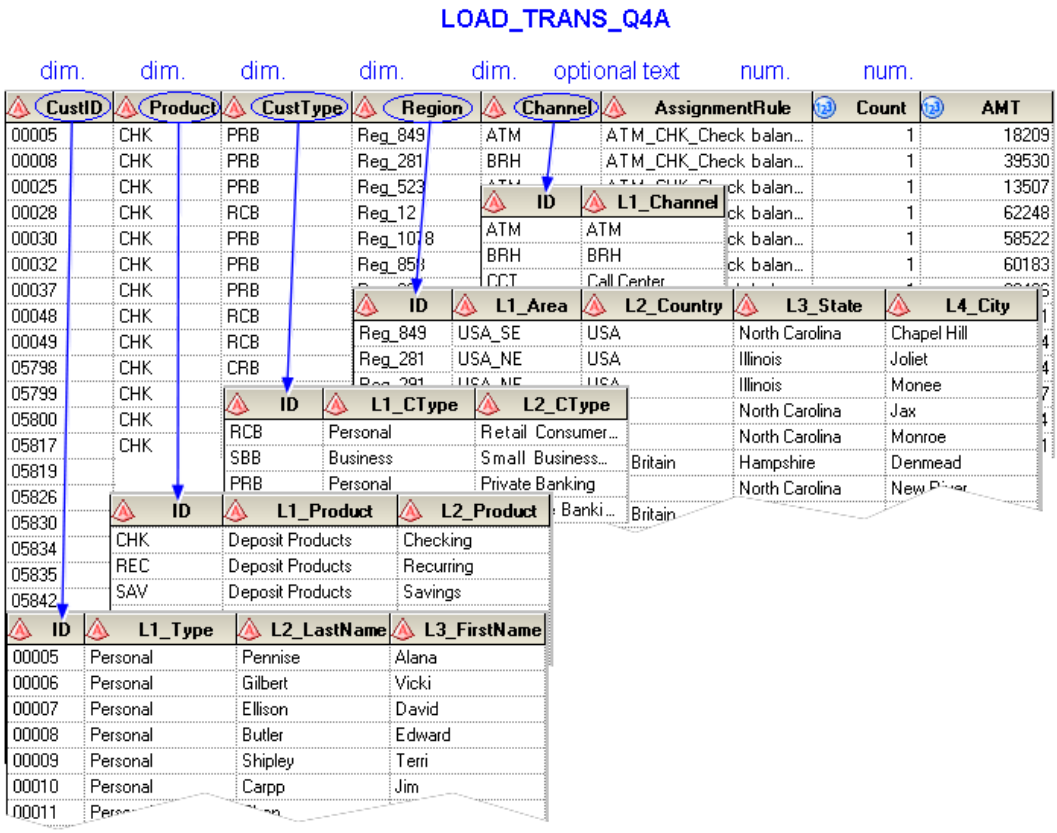
A transaction table contains the following columns and must conform to the following rules:

- 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.

Name	Maximum Length	Description
Dimensions (1 to *n*)	Char 32	Each value identifies the row (in a custom dimensions table) for this transaction. The number of dimensions is optional (minimum 1).
Driver quantities (1 to *n*)	Numeric 8	A rule's driver formula uses these values to calculate the driver quantity for a transaction. The number of columns is optional (minimum 1).
Optional columns (0 to *n*)	Optional	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.

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.



ABMCost Group

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

- LOAD_TRANS_Q4A for the fourth quarter actual period
- LOAD_TRANS_Q4B for the fourth quarter budget period.

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

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

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

LOAD_TRANS_Q4A

	CustID	Product	CustType	Region	Channel	AssignmentRule	Count	AMT
	dim.	dim.	dim.	dim.	dim.	text	num.	num.
1	00005	CHK	PRB	Reg_849	ATM	ATM_CHK_Check b...	1	18209
2	00008	CHK	PRB	Reg_281	ATM	ATM_CHK_Check b...	1	39530
3	00025	CHK	PRB	Reg_523	ATM	ATM_CHK_Check b...	1	13507
4	00028	CHK	RCB	Reg_12	ATM	ATM_CHK_Check b...	1	62248
5	00030	CHK	PRB	Reg_1078	ATM	ATM_CHK_Check b...	1	58522
6	00032	CHK	PRB	Reg_858	ATM	ATM_CHK_Check b...	1	60183
7	00037	CHK	PRB	Reg_935	ATM	ATM_CHK_Check b...	1	20486
8	00048	CHK	RCB	Reg_1080	ATM	ATM_CHK_Check b...	1	54001
9	00049	CHK	RCB	Reg_699	ATM	ATM_CHK_Check b...	1	23424
10	05798	CHK	CRB	Reg_431	ATM	ATM_CHK_Check b...	1	17044
11	05799	CHK	CRB	Reg_551	ATM	ATM_CHK_Check b...	1	27347
12	05800	CHK	CRB	Reg_798	ATM	ATM_CHK_Check b...	1	
13	05817	CHK	CRB	Reg_230	ATM	ATM_CHK_Check b...		
14	05819	CHK	PRB	Reg_32	ATM	ATM_CHK_Chec	1	725
15	05826	CHK	CRB	Reg_163	AT	pen Ac...	1	770
16	05830	CHK	CRB		BRH	BRH_TRM_Open Ac...	1	-104.75
17	05834	CHK	SBB	Reg_1029	BRH	BRH_TRM_Open Ac...	1	1535
18	05835		PRB	Reg_635	BRH	BRH_TRM_Open Ac...	1	606
19		TRM	RCB	Reg_842	BRH	BRH_TRM_Open Ac...	1	1339
386	00003	TRM	PRB	Reg_939	BRH	BRH_TRM_Withdra...	1	15762
387	00014	TRM	PRB	Reg_1034	BRH	BRH_TRM_Withdra...	1	16431
388	00021	TRM	PRB	Reg_1035	BRH	BRH_TRM_Withdra...	1	49710
389	00024	TRM	PRB	Reg_434	BRH	BRH_TRM_Withdra...	1	20289
390	00048	TRM	RCB	Reg_1080	BRH	BRH_TRM_Withdra...	1	21561
391	05800	TRM	CRB	Reg_798	BRH	BRH_TRM_Withdra...	1	-2930
392	05801	TRM	SBB	Reg_1029	BRH	BRH_TRM_Withdra...	1	42981
393	05816	TRM	PRB	Reg_635	BRH	BRH_TRM_Withdra...	1	16966
394	05831	TRM	RCB	Reg_842	BRH	BRH_TRM_Withdra...	1	37497
395	00038	UCR	PRB	Reg_426	BRH	BRH_UCR_Close A...	1	134
396	00043	UCR	PRB	Reg_617	BRH	BRH_UCR_Close A...	1	93
397	05800	UCR	CRB	Reg_798	BRH	BRH_UCR_Close A...	1	59.69
398	05816	UCR	CRB	Reg_680	BRH	BRH_UCR_Close A...	1	233
399	05822	UCR	CRB	Reg_116	BRH	BRH_UCR_Close A...	1	99
400	05831	UCR	CRB	Reg_958	BRH	BRH_UCR_Close A...	1	30
401	00038	UCR	PRB	Reg_426	BRH	BRH_UCR_Open Ac...	1	1119
402	00043	UCR	PRB	Reg_617	BRH	BRH_UCR_Open Ac...	1	771
403	05800	UCR	CRB	Reg_798	BRH	BRH_UCR_Open Ac...	1	496.75
404	05816	UCR	CRB	Reg_680	BRH	BRH_UCR_Open Ac...	1	1938
405	05822	UCR	CRB	Reg_116	BRH	BRH_UCR_Open Ac...	1	827
406	05831	UCR	CRB	Reg_958	BRH	BRH_UCR_Open Ac...	1	254

CallCenter Group

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

- CALLCENTER_Q4A for the fourth quarter actual period
- CALLCENTER _Q4B for the fourth quarter budget period.

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

Field	Name	Description
1	**Product**	ID of product dimension
2	**CustType**	ID of customer-type dimension
3	**Region**	ID of region dimension
4	**Channel**	ID of channel dimension
5	**Communication**	Number of calls

Field	Name	Description
6	**Complaints**	Number of complaints
7	**Inquiry**	Number of inquiries
8	**Requests**	Number of requests
9	**CrossSell**	Number of sales calls
10	**Offer**	Number of offers
11	**Count**	A value of 1 indicates an evenly assigned item

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

CALLCENTER_Q4A

dim.	dim.	dim.	dim.	num.	num.	num.	num.	num.	num.	num
Product	CustType	Region	Channel	Communication	Complaints	Inquiry	Requests	CrossSell	Offer	Count
CHK	PRB	Reg_87	CCT	2	0	1	0	1	0	1
CHK	PRB	Reg_87	CCT	298	15	149	49	74	30	1
CHK	PRB	Reg_768	CCT	99	5	50	16	25	10	1
CHK	PRB	Reg_768	CCT	461	23	231	76	115	46	1
CHK	PRB	Reg_70	CCT	6	0	3	1	2	1	1
CHK	PRB	Reg_70	CCT	180	9	90	30	45	18	1
CHK	PRB	Reg_528	CCT	91	5	46	15	23	9	1
CHK	RCB	Reg_202	CCT	155	8	78	26	39	16	
CHK	RCB	Reg_870	CCT	186	9	93	31			
CHK	PRB	Reg_1101	CCT	456	23	228			46	1
CHK	PRB	Reg_218	CCT	313				78	31	1
CHK						156	52	78	31	1
					4	15	5	7	7	1
		Reg_135	CCT	122	18	61	20	31	31	1
UCR	PRB	Reg_1080	CCT	126	19	63	21	31	31	1
UCR	CRB	Reg_59	CCT	144	22	72	24	36	36	1
UCR	SBB	Reg_1090	CCT	72	11	36	12	18	18	1
UCR	SBB	Reg_1131	CCT	25	4	13	4	6	6	1
UCR	CRB	Reg_768	CCT	20	3	10	3	5	5	1
UCR	CRB	Reg_413	CCT	133	20	66	22	33	33	1
UCR	CRB	Reg_696	CCT	119	18	60	20	30	30	1
UCR	CRB	Reg_1003	CCT	5	1	2	1	1	1	1
UCR	SBB	Reg_1093	CCT	145	22	73	24	36	36	1

Revenue Group

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

- REVENUE_Q4A for the fourth quarter actual period
- REVENUE _Q4B for the fourth quarter budget period.

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

Table 15.2 *Revenue Group*

Field	Name	Description
1	**Channel**	ID of channel dimension
2	**CustID**	ID of customer dimension

Field	Name	Description
3	**Product**	ID of product dimension
4	**CustType**	ID of customer type dimension
5	**Region**	ID of region dimension
6	**AMT**	Calculated specific revenue
7	**ID**	Behavior ID for revenue items

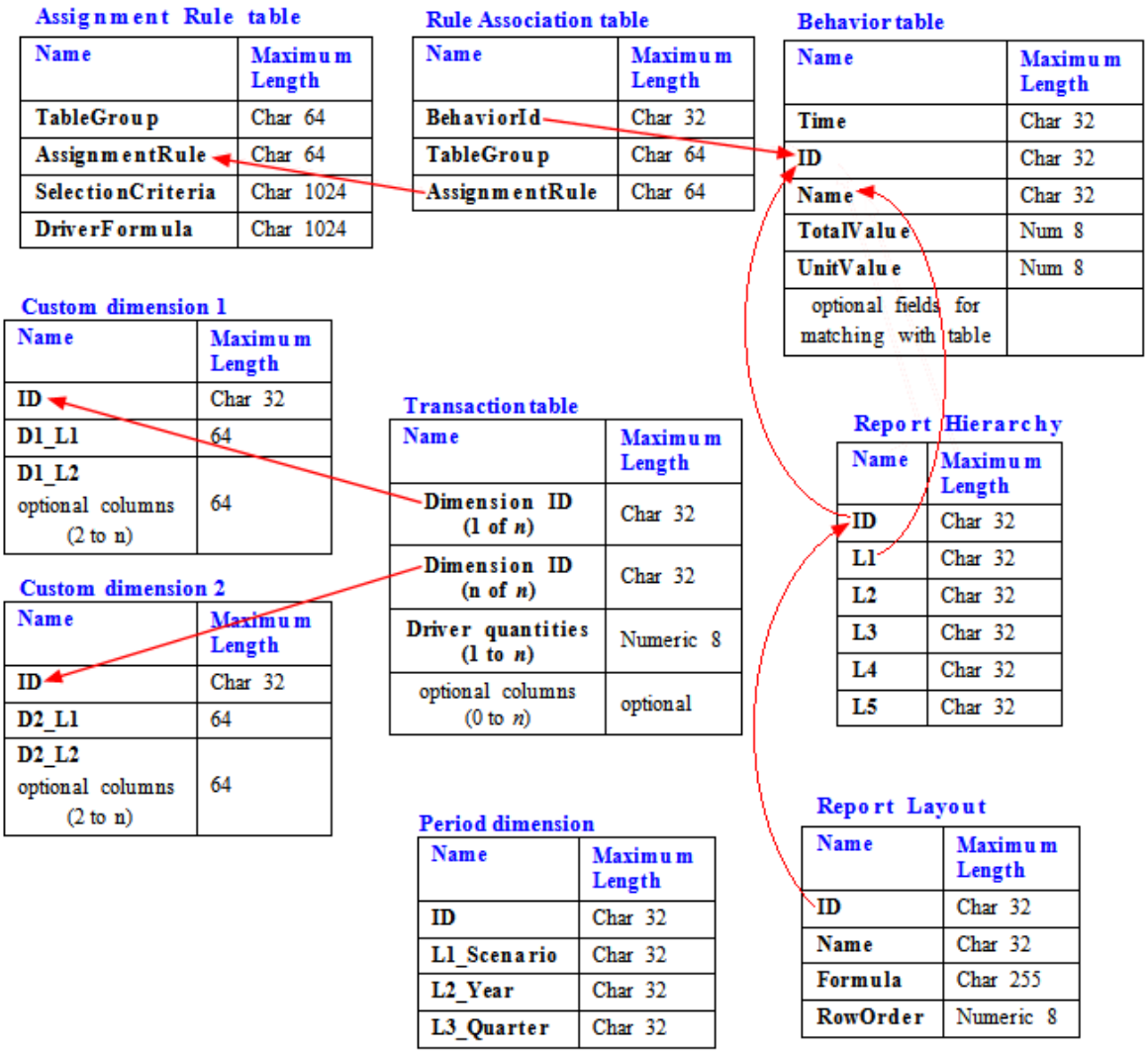
The following graphic shows a portion of the contents of REVENUE_Q4A:

REVENUE_Q4A

Channel (dim.)	CustID (dim.)	Product (dim.)	CustType (dim.)	Region (dim.)	AMT (num.)	ID (text)
ATM	00001	OVD	PRB	Reg_67	399.3	10002
ATM	00001	OVD	PRB	Reg_67	8.16	12002
ATM	00001	OVD	PRB	Reg_67	20.47	13001
ATM	00001	OVD	PRB	Reg_67	25.59	13002
ATM	00001	OVD	PRB	Reg_67	5.12	14001
ATM	00002	CRC	PRB	Reg_188	4359.35	10001
ATM	00002	CRC	PRB	Reg_188	1744.98	12001
ATM	00002	CRC	PRB	Reg_188	134.32	
ATM	00002	CRC	PRB	Reg_188		
ATM					221.31	13001
				Reg_795	276.64	13002
	05845	FBP	CRB	Reg_795	55.33	14001
CCT	05845	OTP	CRB	Reg_194	23.38	13001
CCT	05845	OTP	CRB	Reg_194	29.22	13002
CCT	05845	OTP	CRB	Reg_194	5.84	14001
CCT	05846	UCR	SBB	Reg_1093	6478.6	10002
CCT	05846	UCR	SBB	Reg_1093	332.17	13001
CCT	05846	UCR	SBB	Reg_1093	415.22	13002
CCT	05846	UCR	SBB	Reg_1093	83.04	14001
CCT	05847	FBP	SBB	Reg_269	21.04	13001
CCT	05847	FBP	SBB	Reg_269	26.3	13002
CCT	05847	FBP	SBB	Reg_269	5.26	14001
CCT	05847	OVD	CRB	Reg_448	3208.39	10002
CCT	05847	OVD	CRB	Reg_448	164.5	13001
CCT	05847	OVD	CRB	Reg_448	205.63	13002
CCT	05847	OVD	CRB	Reg_448	41.13	14001

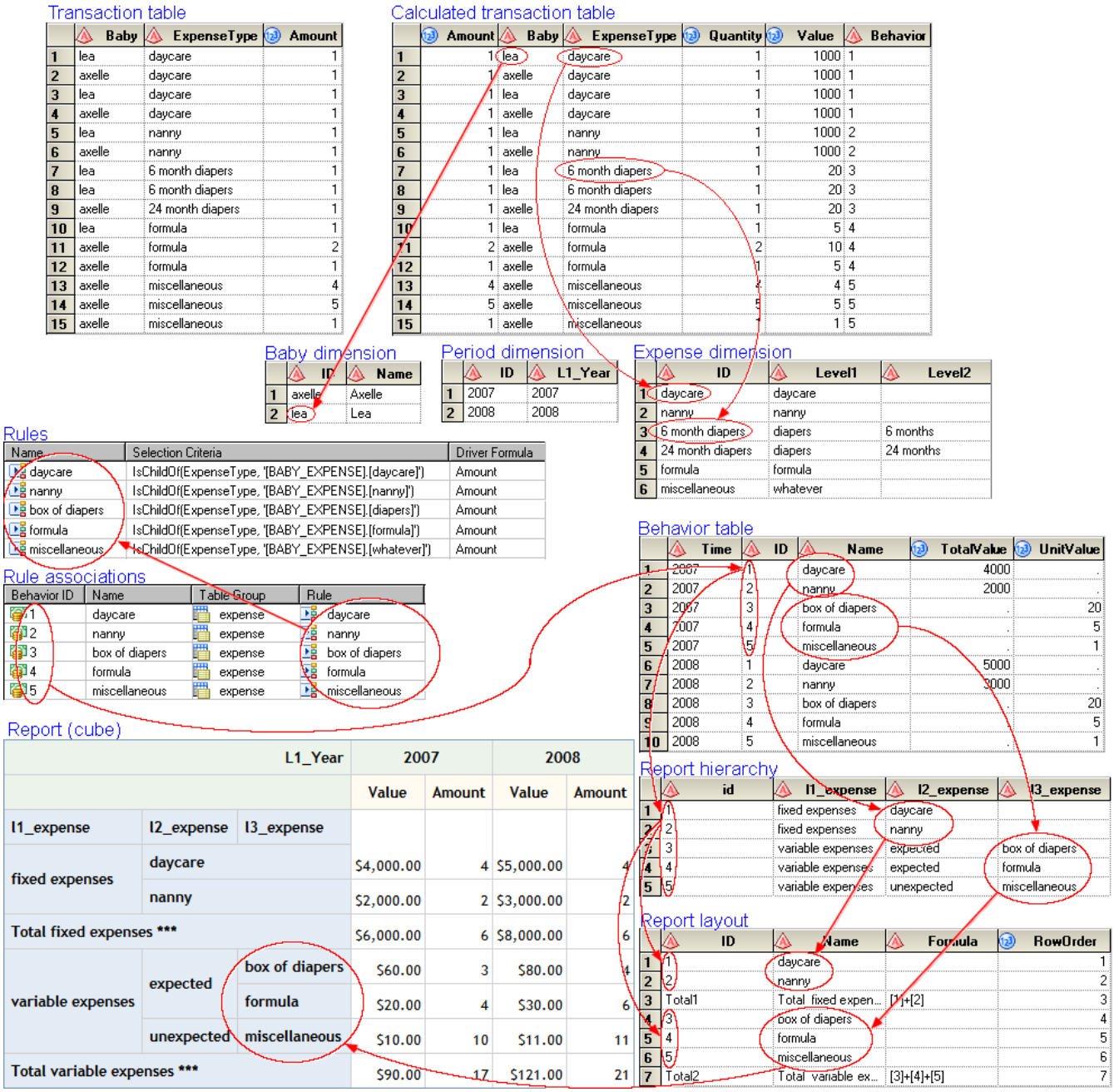
Data Model Definition

The following graphic shows the the tables that make up the data model, with the columns for each field, and the relations among them.



A Complete Model

The following graphic shows all the elements of a complete model, even smaller than Baby Bank. This one can be called Babies Bank. It calculates expenses for two babies, Lea and Axelle, during the course of two years (only one of which is shown here).



Transaction table

	Baby	ExpenseType	Amount
1	lea	daycare	1
2	axelle	daycare	1
3	lea	daycare	1
4	axelle	daycare	1
5	lea	nanny	1
6	axelle	nanny	1
7	lea	6 month diapers	1
8	lea	6 month diapers	1
9	axelle	24 month diapers	1
10	lea	formula	1
11	axelle	formula	2
12	axelle	formula	1
13	axelle	miscellaneous	4
14	axelle	miscellaneous	5
15	axelle	miscellaneous	1

Calculated transaction table

	Amount	Baby	ExpenseType	Quantity	Value	Behavior
1	1	lea	daycare	1	1000	1
2	1	axelle	daycare	1	1000	1
3	1	lea	daycare	1	1000	1
4	1	axelle	daycare	1	1000	1
5	1	lea	nanny	1	1000	2
6	1	axelle	nanny	1	1000	2
7	1	lea	6 month diapers	1	20	3
8	1	lea	6 month diapers	1	20	3
9	1	axelle	24 month diapers	1	20	3
10	1	lea	formula	1	5	4
11	2	axelle	formula	2	10	4
12	1	axelle	formula	1	5	4
13	4	axelle	miscellaneous	4	4	5
14	5	axelle	miscellaneous	5	5	5
15	1	axelle	miscellaneous	1	1	5

Baby dimension

	ID	Name
1	axelle	Axelle
2	lea	Lea

Period dimension

	ID	L1_Year
1	2007	2007
2	2008	2008

Expense dimension

	ID	Level1	Level2
1	daycare	daycare	
2	nanny	nanny	
3	6 month diapers	diapers	6 months
4	24 month diapers	diapers	24 months
5	formula	formula	
6	miscellaneous	whatever	

Rules

Name	Selection Criteria	Driver Formula
daycare	IsChildOf([ExpenseType], '[BABY_EXPENSE].[daycare]')	Amount
nanny	IsChildOf([ExpenseType], '[BABY_EXPENSE].[nanny]')	Amount
box of diapers	IsChildOf([ExpenseType], '[BABY_EXPENSE].[diapers]')	Amount
formula	IsChildOf([ExpenseType], '[BABY_EXPENSE].[formula]')	Amount
miscellaneous	IsChildOf([ExpenseType], '[BABY_EXPENSE].[whatever]')	Amount

Rule associations

Behavior ID	Name	Table Group	Rule
1	daycare	expense	daycare
2	nanny	expense	nanny
3	box of diapers	expense	box of diapers
4	formula	expense	formula
5	miscellaneous	expense	miscellaneous

Behavior table

	Time	ID	Name	TotalValue	UnitValue
1	2007	1	daycare	4000	.
2	2007	2	nanny	2000	.
3	2007	3	box of diapers	.	20
4	2007	4	formula	.	5
5	2007	5	miscellaneous	.	1
6	2008	1	daycare	5000	.
7	2008	2	nanny	3000	.
8	2008	3	box of diapers	.	20
9	2008	4	formula	.	5
10	2008	5	miscellaneous	.	1

Report (cube)

		L1_Year	2007		2008	
			Value	Amount	Value	Amount
l1_expense	l2_expense	l3_expense				
fixed expenses	daycare		$4,000.00	4	$5,000.00	4
	nanny		$2,000.00	2	$3,000.00	2
Total fixed expenses ***			$6,000.00	6	$8,000.00	6
variable expenses	expected	box of diapers	$60.00	3	$80.00	4
		formula	$20.00	4	$30.00	6
	unexpected	miscellaneous	$10.00	10	$11.00	11
Total variable expenses ***			$90.00	17	$121.00	21

Report hierarchy

	id	l1_expense	l2_expense	l3_expense
1	1	fixed expenses	daycare	
2	2	fixed expenses	nanny	
3	3	variable expenses	expected	box of diapers
4	4	variable expenses	expected	formula
5	5	variable expenses	unexpected	miscellaneous

Report layout

	ID	Name	Formula	RowOrder
1	1	daycare		1
2	2	nanny		2
3	Total1	Total_fixed expen...	[1]+[2]	3
4	3	box of diapers		4
5	4	formula		5
6	5	miscellaneous		6
7	Total2	Total_variable ex...	[3]+[4]+[5]	7

Chapter 16

A Minimal Model

Introduction

The Baby Bank model in this tutorial is intended to be small enough to be easily understood and yet large enough to suggest a realistic business model. Although the tables are only a fraction of the size of the tables of a real business model, they are still too large to see each one easily in its entirety, much less all of them at once. By contrast, this chapter develops a model that is so minimal that you can see all of the tables in their entirety at the same time. The model represents the budget of a single individual. And, instead of showing an entire personal budget, it shows only salary minus tax withholding and food expenses. The following graphic shows the final report to be produced.

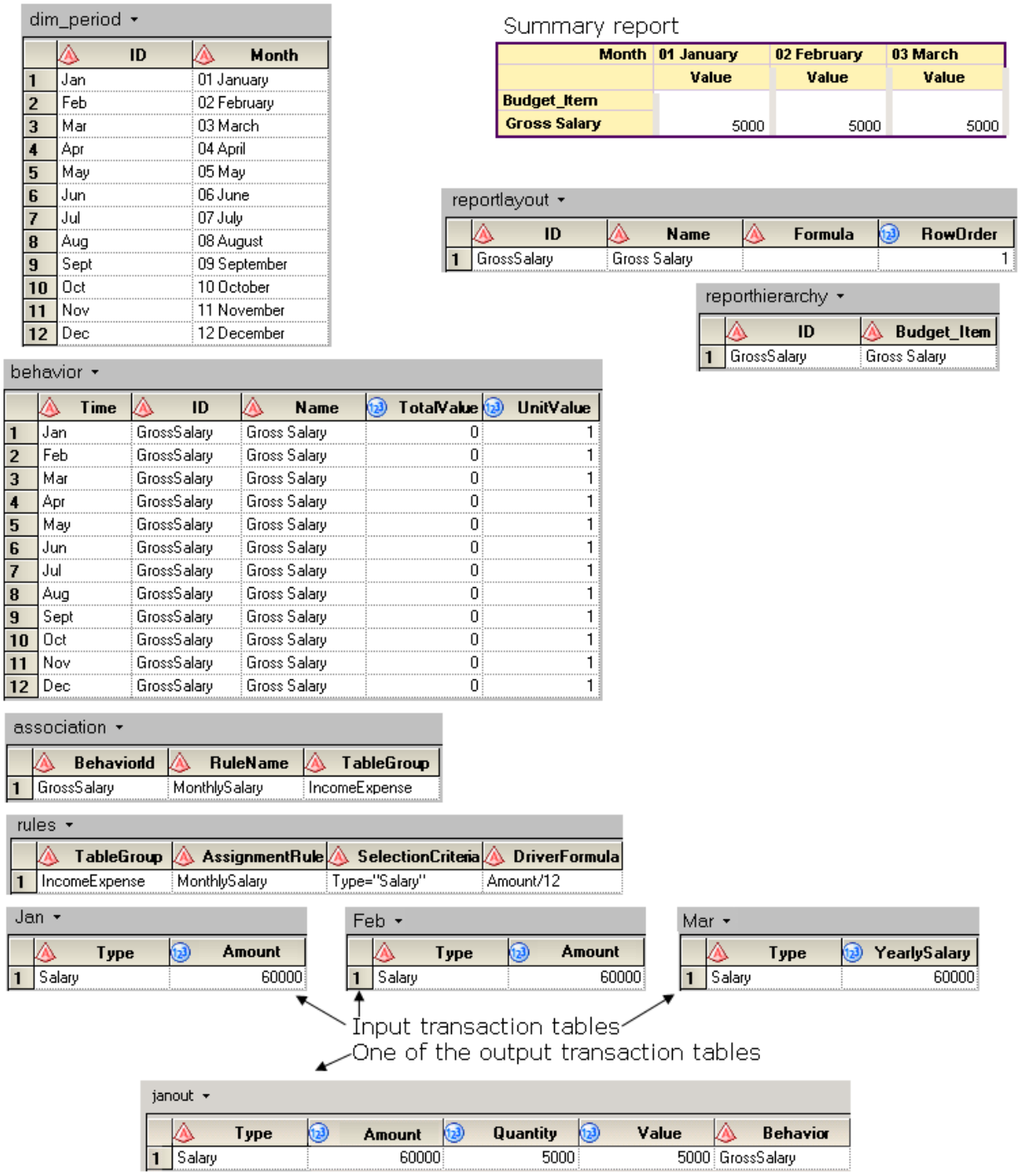
	Month	01 January	02 February	03 March
		Value	Value	Value
Budget_Item	L3			
Gross Salary		5000	5000	5000
Withholding		1000	1000	1000
Net Salary		4000	4000	4000
Food	At home	37	62	60
	El Rodeo	.	18	32
	Hunam	.	25	24
	Panzanella	32	.	.
	Weaver Street	31	.	22
Total Food		100	105	138
Bottom Line		3900	3895	3862

Step 1: Calculating Gross Monthly Salary

We create this model step by step. Once the model has been completely developed, we will examine the relationships among its tables.

As a first step, we create a minuscule model to produce a report showing only monthly gross salary. In this model, the annual salary is contained in each month's transaction table. The rule, Monthly Salary, divides the annual salary by 12 to calculate the salary for a month. By including the annual salary in each month's transaction table, the annual salary can change from one month to the next. We do not show any change in the three months represented here.

The following graphic shows all the tables that make up this minuscule model and the resulting summary report.



dim_period

	ID	Month
1	Jan	01 January
2	Feb	02 February
3	Mar	03 March
4	Apr	04 April
5	May	05 May
6	Jun	06 June
7	Jul	07 July
8	Aug	08 August
9	Sept	09 September
10	Oct	10 October
11	Nov	11 November
12	Dec	12 December

Summary report

Month	01 January	02 February	03 March
	Value	Value	Value
Budget_Item			
Gross Salary	5000	5000	5000

reportlayout

	ID	Name	Formula	RowOrder
1	GrossSalary	Gross Salary		1

reporthierarchy

	ID	Budget_Item
1	GrossSalary	Gross Salary

behavior

	Time	ID	Name	TotalValue	UnitValue
1	Jan	GrossSalary	Gross Salary	0	1
2	Feb	GrossSalary	Gross Salary	0	1
3	Mar	GrossSalary	Gross Salary	0	1
4	Apr	GrossSalary	Gross Salary	0	1
5	May	GrossSalary	Gross Salary	0	1
6	Jun	GrossSalary	Gross Salary	0	1
7	Jul	GrossSalary	Gross Salary	0	1
8	Aug	GrossSalary	Gross Salary	0	1
9	Sept	GrossSalary	Gross Salary	0	1
10	Oct	GrossSalary	Gross Salary	0	1
11	Nov	GrossSalary	Gross Salary	0	1
12	Dec	GrossSalary	Gross Salary	0	1

association

	BehaviorId	RuleName	TableGroup
1	GrossSalary	MonthlySalary	IncomeExpense

rules

	TableGroup	AssignmentRule	SelectionCriteria	DriverFormula
1	IncomeExpense	MonthlySalary	Type="Salary"	Amount/12

Jan

	Type	Amount
1	Salary	60000

Feb

	Type	Amount
1	Salary	60000

Mar

	Type	YearlySalary
1	Salary	60000

Input transaction tables

One of the output transaction tables

janout

	Type	Amount	Quantity	Value	Behavior
1	Salary	60000	5000	5000	GrossSalary

Step 2: Deducting Taxes

In a second step, we complicate the model slightly to subtract the tax withheld from each monthly salary to produce a report showing both gross salary and net salary. The rule to calculate the withholding once again calculates the monthly salary and then multiplies the product by .20 for a 20% deduction.

The following graphic shows all of the tables that make up the model and the resulting summary report.

dim_period

	ID	Month
1	Jan	01 January
2	Feb	02 February
3	Mar	03 March
4	Apr	04 April
5	May	05 May
6	Jun	06 June
7	Jul	07 July
8	Aug	08 August
9	Sept	09 September
10	Oct	10 October
11	Nov	11 November
12	Dec	12 December

Summary report

Month	01 January	02 February	03 March
	Value	Value	Value
Budget_Item			
Gross Salary	$5,000.00	$5,000.00	$5,000.00
Withholding	$1,000.00	$1,000.00	$1,000.00
Net Salary	$4,000.00	$4,000.00	$4,000.00

reportlayout

	ID	Name	Formula	RowOrder
1	GrossSalary	Gross Salary		1
2	Deduction	Withholding		2
3	NetSalary	Net Salary	[GrossSalary]-[Deduction]	3

behavior

	Time	ID	Name	TotalValue	UnitValue
1	Jan	GrossSalary	Gross Salary	0	1
2	Feb	GrossSalary	Gross Salary	0	1
3	Mar	GrossSalary	Gross Salary	0	1
4	Apr	GrossSalary	Gross Salary	0	1
5	May	GrossSalary	Gross Salary	0	1
6	Jun	GrossSalary	Gross Salary	0	1
7	Jul	GrossSalary	Gross Salary	0	1
8	Aug	GrossSalary	Gross Salary	0	1
9	Sept	GrossSalary	Gross Salary	0	1
10	Oct	GrossSalary	Gross Salary	0	1
11	Nov	GrossSalary	Gross Salary	0	1
12	Dec	GrossSalary	Gross Salary	0	1
13	Jan	Deduction	Withholding	0	1
14	Feb	Deduction	Withholding	0	1
15	Mar	Deduction	Withholding	0	1
16	Apr	Deduction	Withholding	0	1
17	May	Deduction	Withholding	0	1
18	Jun	Deduction	Withholding	0	1
19	Jul	Deduction	Withholding	0	1
20	Aug	Deduction	Withholding	0	1
21	Sept	Deduction	Withholding	0	1
22	Oct	Deduction	Withholding	0	1
23	Nov	Deduction	Withholding	0	1
24	Dec	Deduction	Withholding	0	1

reporthierarchy

	ID	Budget_Item
1	GrossSalary	Gross Salary
2	Deduction	Withholding

association

	BehaviorId	RuleName	TableGroup
1	GrossSalary	MonthlySalary	IncomeExpense
2	Deduction	DeductTax	IncomeExpense

rules

	TableGroup	AssignmentRule	SelectionCriteria	DriverFormula
1	IncomeExpense	MonthlySalary	Type="Salary"	Amount/12
2	IncomeExpense	DeductTax	Type="Salary"	(Amount/12)*.20

Jan

	Type	Amount
1	Salary	60000

Feb

	Type	Amount
1	Salary	60000

Mar

	Type	Amount
1	Salary	60000

Input transaction tables

One of the putput transaction tables

janout

	Type	Amount	Quantity	Value	Behavior
1	Salary	60000	5000	5000	GrossSalary
2	Salary	60000	1000	1000	Deduction

Step 3: Completing the Model

The Entire Model at a Glance

Next, we complete the model to produce the report promised at the beginning of this chapter. The following graphic shows all of the tables that make up the model. Because, up to now, we have not shown transaction tables for the months of April through December, we have dropped those months from the behavior table—and consequently from the association table, which associates behaviors with rules. This makes it easier to see each entire table.

dim_period

	ID	Month
1	Jan	01 January
2	Feb	02 February
3	Mar	03 March
4	Apr	04 April
5	May	05 May
6	Jun	06 June
7	Jul	07 July
8	Aug	08 August
9	Sept	09 September
10	Oct	10 October
11	Nov	11 November
12	Dec	12 December

Summary report

		Month	01 January	02 February	03 March
			Value	Value	Value
Budget_Item	L3				
Gross Salary			5000	5000	5000
Withholding			1000	1000	1000
Net Salary			4000	4000	4000
Food		At home	37	62	60
		El Rodeo	.	18	32
		Hunam	.	25	24
		Panzanella	32	.	.
		Weaver Street	31	.	22
Total Food			100	105	138
Bottom Line			3900	3895	3862

reportlayout

	ID	Name	Formula	RowOrder
1	GrossSalary	Gross Salary		1
2	Deduction	Withholding		2
3	NetSalary	Net Salary	[GrossSalary]-[Deduction]	3
4	Fd	Food		4
5	TotalFood	Total Food	[Fd]	5
6	BottomLine	Bottom Line	[NetSalary]-[Fd]	6

behavior

	Time	ID	Name	TotalValue	UnitValue
1	Jan	GrossSalary	Gross Salary	0	1
2	Feb	GrossSalary	Gross Salary	0	1
3	Mar	GrossSalary	Gross Salary	0	1
4	Jan	Deduction	Withholding	0	1
5	Feb	Deduction	Withholding	0	1
6	Mar	Deduction	Withholding	0	1
7	Jan	House	At home	0	
8	Feb	House	At home		
				0	1
		ws	Weaver Street	0	1
12	Mar	ws	Weaver Street	0	1
13	Jan	panza	Panzanella	0	1
14	Feb	panza	Panzanella	0	1
15	Mar	panza	Panzanella	0	1
16	Jan	hunam	Hunam	0	1
17	Feb	hunam	Hunam	0	1
18	Mar	hunam	Hunam	0	1
19	Jan	rodeo	El Rodeo	0	1
20	Feb	rodeo	El Rodeo	0	1
21	Mar	rodeo	El Rodeo	0	1

reporthierarchy

	ID	Budget_Item	L3
1	GrossSalary	Gross Salary	
2	Deduction	Withholding	
3	Fd	Food	
4	House	Food	At home
5	ws	Food	Weaver Street
6	panza	Food	Panzanella
7	hunam	Food	Hunam
8	rodeo	Food	El Rodeo

rules

	name	selectionCriteria	driverFormula	tableGroupName
1	MonthlySalary	Type="Salary"	Amount/12	IncomeExpense
2	DeductTax	Type="Salary"	(Amount/12)*.20	IncomeExpense
3	FoodExpense	MatchColumns(Transaction(Provider), Behavior(ID))	Amount	IncomeExpense

Jan

	Type	Amount	Provider
1	Salary	60000.00	
2	Food	25.15	House
3	Food	18.50	ws
4	Food	12.10	House
5	Food	32.00	panza
6	Food	12.00	ws

An input transaction table

An output transaction table

association

	BehaviorId	RuleName	TableGroup
1	GrossSalary	MonthlySalary	IncomeExpense
2	Deduction	DeductTax	IncomeExpense
3	House	FoodExpense	IncomeExpense
4	ws	FoodExpense	IncomeExpense
5	panza	FoodExpense	IncomeExpense
6	hunam	FoodExpense	IncomeExpense
7	rodeo	FoodExpense	IncomeExpense

janout

	Amount	Provider	Type	Quantity	Value	Behavior
1	60000.00		Salary	5000	5000	GrossSalary
2	60000.00		Salary	1000	1000	Deduction
3	25.15	House	Food	25.15	25.15	House
4	18.50	ws	Food	18.5	18.5	ws
5	12.10	House	Food	12.1	12.1	House
6	32.00	panza	Food	32	32	panza
7	12.00	ws	Food	12	12	ws

dim_food

	ID	Place	Restau_Type	Restaurant
1	House	At home		
2	Restau	Restaurant		
3	Vegetarian	Restaurant	Vegetarian	
4	ws	Restaurant	Vegetarian	Weaver Street
5	panza	Restaurant	Vegetarian	Panzanella
6	Chinese	Restaurant	Chinese	
7	hunam	Restaurant	Chinese	Hunam
8	Mexican	Restaurant	Mexican	
9	rodeo	Restaurant	Mexican	El Rodeo

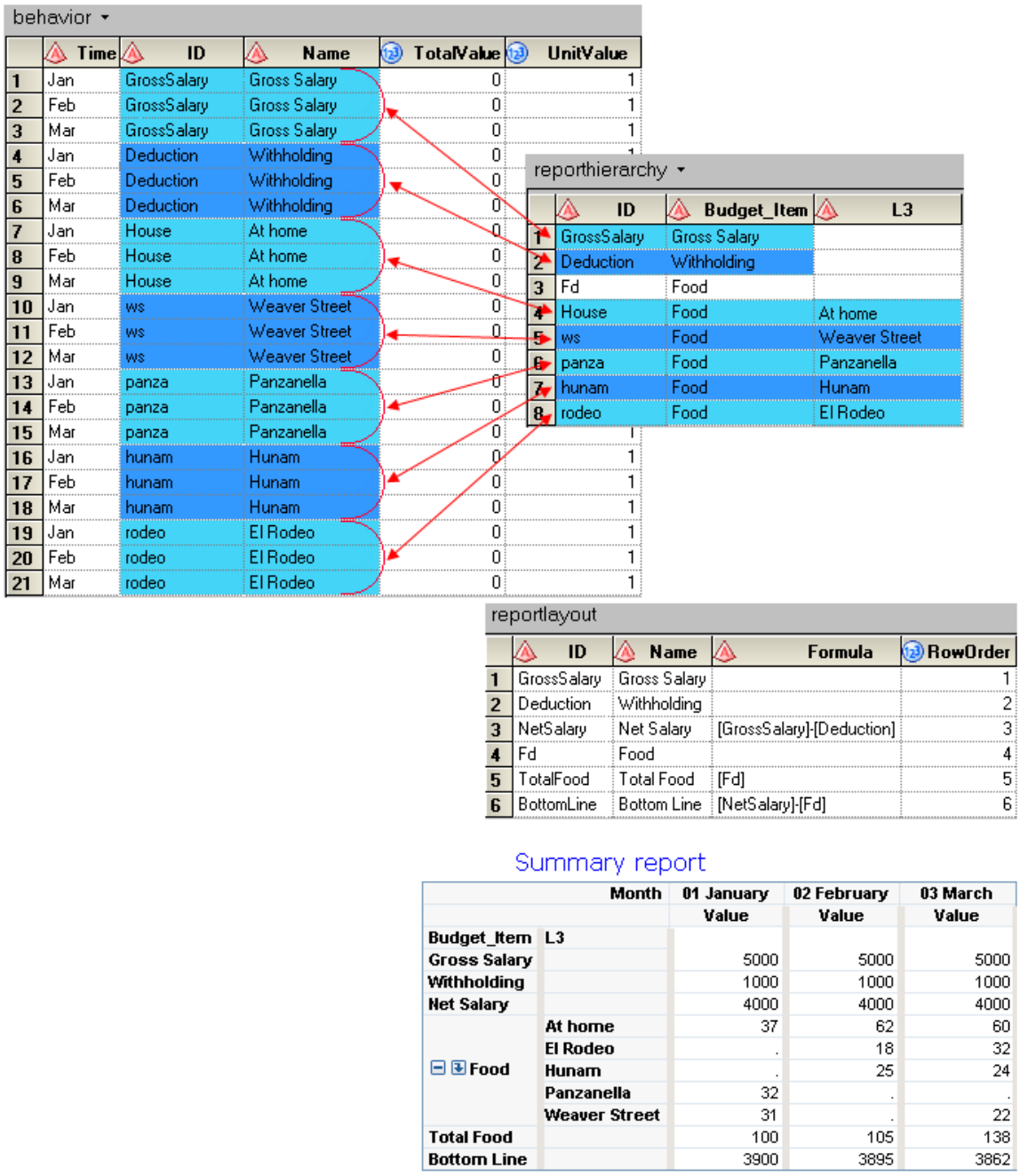
Having seen the model in its entirety, we can take a closer look at its constituent tables.

Behavior Table and Report Hierarchy

Now we can look at the tables in detail. First, consider the relationship between the behavior table and the report hierarchy. For every row in the behavior table, there is a corresponding row in the report hierarchy. However, there are rows in the report hierarchy for which there is no corresponding row in the behavior table. These rows (such as the Fd row) establish higher levels in the hierarchy. Such rows can establish intermediate levels, but the report hierarchy shown here has no intermediate levels.

Note that multiple rows in the behavior table have the same ID. The ID identifies the period in which a revenue or expense is incurred, and of course there are multiple revenues and expenses in a single period.

For convenience of reference, we also show the report layout and resulting summary report.



behavior

	Time	ID	Name	TotalValue	UnitValue
1	Jan	GrossSalary	Gross Salary	0	1
2	Feb	GrossSalary	Gross Salary	0	1
3	Mar	GrossSalary	Gross Salary	0	1
4	Jan	Deduction	Withholding	0	1
5	Feb	Deduction	Withholding	0	1
6	Mar	Deduction	Withholding	0	1
7	Jan	House	At home	0	1
8	Feb	House	At home	0	1
9	Mar	House	At home	0	1
10	Jan	ws	Weaver Street	0	1
11	Feb	ws	Weaver Street	0	1
12	Mar	ws	Weaver Street	0	1
13	Jan	panza	Panzanella	0	1
14	Feb	panza	Panzanella	0	1
15	Mar	panza	Panzanella	0	1
16	Jan	hunam	Hunam	0	1
17	Feb	hunam	Hunam	0	1
18	Mar	hunam	Hunam	0	1
19	Jan	rodeo	El Rodeo	0	1
20	Feb	rodeo	El Rodeo	0	1
21	Mar	rodeo	El Rodeo	0	1

reporthierarchy

	ID	Budget_Item	L3
1	GrossSalary	Gross Salary	
2	Deduction	Withholding	
3	Fd	Food	
4	House	Food	At home
5	ws	Food	Weaver Street
6	panza	Food	Panzanella
7	hunam	Food	Hunam
8	rodeo	Food	El Rodeo

reportlayout

	ID	Name	Formula	RowOrder
1	GrossSalary	Gross Salary		1
2	Deduction	Withholding		2
3	NetSalary	Net Salary	[GrossSalary]-[Deduction]	3
4	Fd	Food		4
5	TotalFood	Total Food	[Fd]	5
6	BottomLine	Bottom Line	[NetSalary]-[Fd]	6

Summary report

	Month	01 January	02 February	03 March
		Value	Value	Value
Budget_Item	L3			
Gross Salary		5000	5000	5000
Withholding		1000	1000	1000
Net Salary		4000	4000	4000
Food	At home	37	62	60
	El Rodeo	.	18	32
	Hunam	.	25	24
	Panzanella	32	.	.
	Weaver Street	31	.	22
Total Food		100	105	138
Bottom Line		3900	3895	3862

Report Hierarchy, Report Layout, and Behavior Table

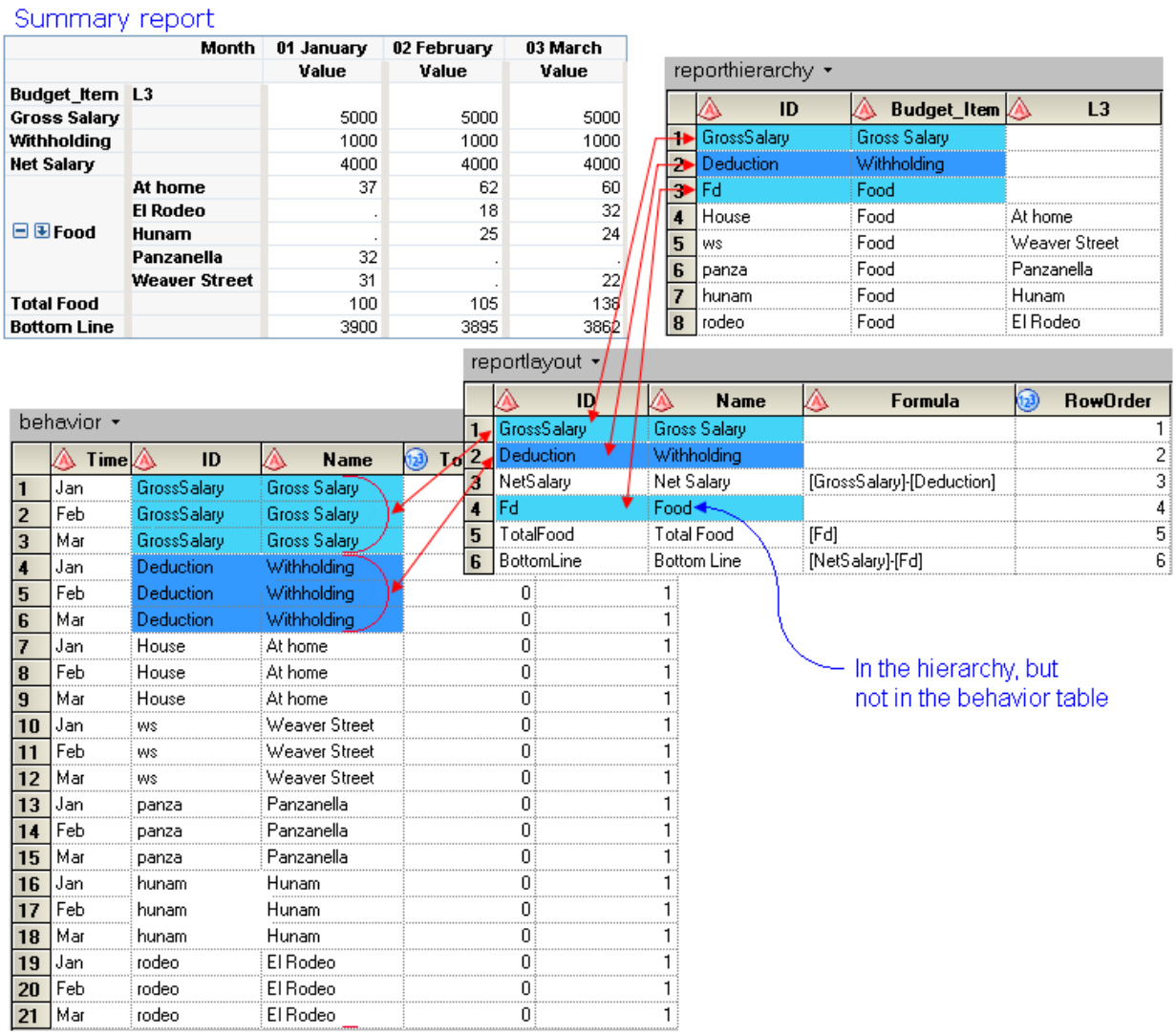
Overview

A model can have multiple hierarchies, and each hierarchy can have multiple layouts. In this section, we look at the relationship between our model's report hierarchy, behavior table, and three alternative report layouts.

First Report Layout

In the following graphic, every row in the report layout that is not a formula row has a corresponding row in the report hierarchy. However, not every row in the report

hierarchy has a corresponding row in the report layout. And some of the rows in the report layout have a corresponding row in the behavior table.



Summary report

	Month	01 January	02 February	03 March
		Value	Value	Value
Budget_Item	L3			
Gross Salary		5000	5000	5000
Withholding		1000	1000	1000
Net Salary		4000	4000	4000
Food	At home	37	62	60
	El Rodeo	.	18	32
	Hunam	.	25	24
	Panzanella	32	.	.
	Weaver Street	31	.	22
Total Food		100	105	138
Bottom Line		3900	3895	3862

reporthierarchy

	ID	Budget_Item	L3
1	GrossSalary	Gross Salary	
2	Deduction	Withholding	
3	Fd	Food	
4	House	Food	At home
5	ws	Food	Weaver Street
6	panza	Food	Panzanella
7	hunam	Food	Hunam
8	rodeo	Food	El Rodeo

reportlayout

	ID	Name	Formula	RowOrder
1	GrossSalary	Gross Salary		1
2	Deduction	Withholding		2
3	NetSalary	Net Salary	[GrossSalary]-[Deduction]	3
4	Fd	Food		4
5	TotalFood	Total Food	[Fd]	5
6	BottomLine	Bottom Line	[NetSalary]-[Fd]	6

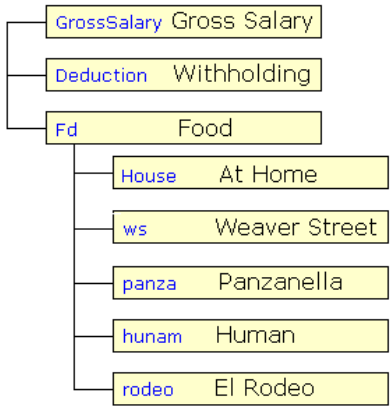
behavior

	Time	ID	Name	To	
1	Jan	GrossSalary	Gross Salary		
2	Feb	GrossSalary	Gross Salary		
3	Mar	GrossSalary	Gross Salary		
4	Jan	Deduction	Withholding		
5	Feb	Deduction	Withholding	0	1
6	Mar	Deduction	Withholding	0	1
7	Jan	House	At home	0	1
8	Feb	House	At home	0	1
9	Mar	House	At home	0	1
10	Jan	ws	Weaver Street	0	1
11	Feb	ws	Weaver Street	0	1
12	Mar	ws	Weaver Street	0	1
13	Jan	panza	Panzanella	0	1
14	Feb	panza	Panzanella	0	1
15	Mar	panza	Panzanella	0	1
16	Jan	hunam	Hunam	0	1
17	Feb	hunam	Hunam	0	1
18	Mar	hunam	Hunam	0	1
19	Jan	rodeo	El Rodeo	0	1
20	Feb	rodeo	El Rodeo	0	1
21	Mar	rodeo	El Rodeo	0	1

However, not every row in the report layout has a corresponding row in the behavior table. There are three types of rows in a report layout that do not have a corresponding row in the behavior table:

- Formula rows (for example, the row with the formula `[GrossSalary]-[Deduction]`)
- Label rows (A row whose formula is "." creates a label in the report. This particular report layout does not contain any label rows.)
- Rows that correspond to rows in the hierarchy table that are not leaf rows (for example, the row with the ID of Fd).

  For clarity, let's view the report hierarchy as a tree rather than as a table. The following graphic shows the report hierarchy as a tree.



Notice that GrossSalary, Deduction, and Fd are all siblings. However, only GrossSalary and Deduction are leaf nodes. Fd has five children, so it is not a leaf node and, therefore, not in the behavior table. The behavior table contains nothing but leaf nodes.

Second Report Layout

The following graphic shows the same report hierarchy with an alternative report layout. This layout has only one row that has a corresponding row in the report hierarchy—the Fd row. Whereas every row in the report layout that is not a formula or label row has a corresponding row in the report hierarchy, not every row in the report hierarchy has a corresponding row in the report layout. And, again, because the Fd row in the layout is not a leaf node in the hierarchy, it is not represented in the behavior table.

Because the Fd row in the report layout is not a leaf node in the report hierarchy, the report opens with that item collapsed. In the following graphic the Food row is initially displayed collapsed, and when a user clicks on the plus sign, the children of the Fd row are displayed.

Summary report (initial view)

Month	01 January	02 February	03 March
	Value	Value	Value
Budget_Item			
⊞ Food	100	105	138
Total Food	100	105	138

Summary report (expanded)

	Month	01 January	02 February	03 March
		Value	Value	Value
Budget_Item	L3			
⊟ Food	At home	37	62	60
	El Rodeo	.	18	32
	Hunam	.	25	24
	Panzanella	32	.	.
	Weaver Street	31	.	22
Total Food		100	105	138

reporthierarchy

	ID	Budget_Item	L3
1	GrossSalary	Gross Salary	
2	Deduction	Withholding	
3	Fd	Food	
4	House	Food	At home
5	ws	Food	Weaver Street
6	panza	Food	Panzanella
7	hunam	Food	Hunam
8	rodeo	Food	El Rodeo

reportlayoutFood

	ID	Name	Formula	RowOrder
1	Fd	Food		1
2	TotalFood	Total Food	[Fd]	2

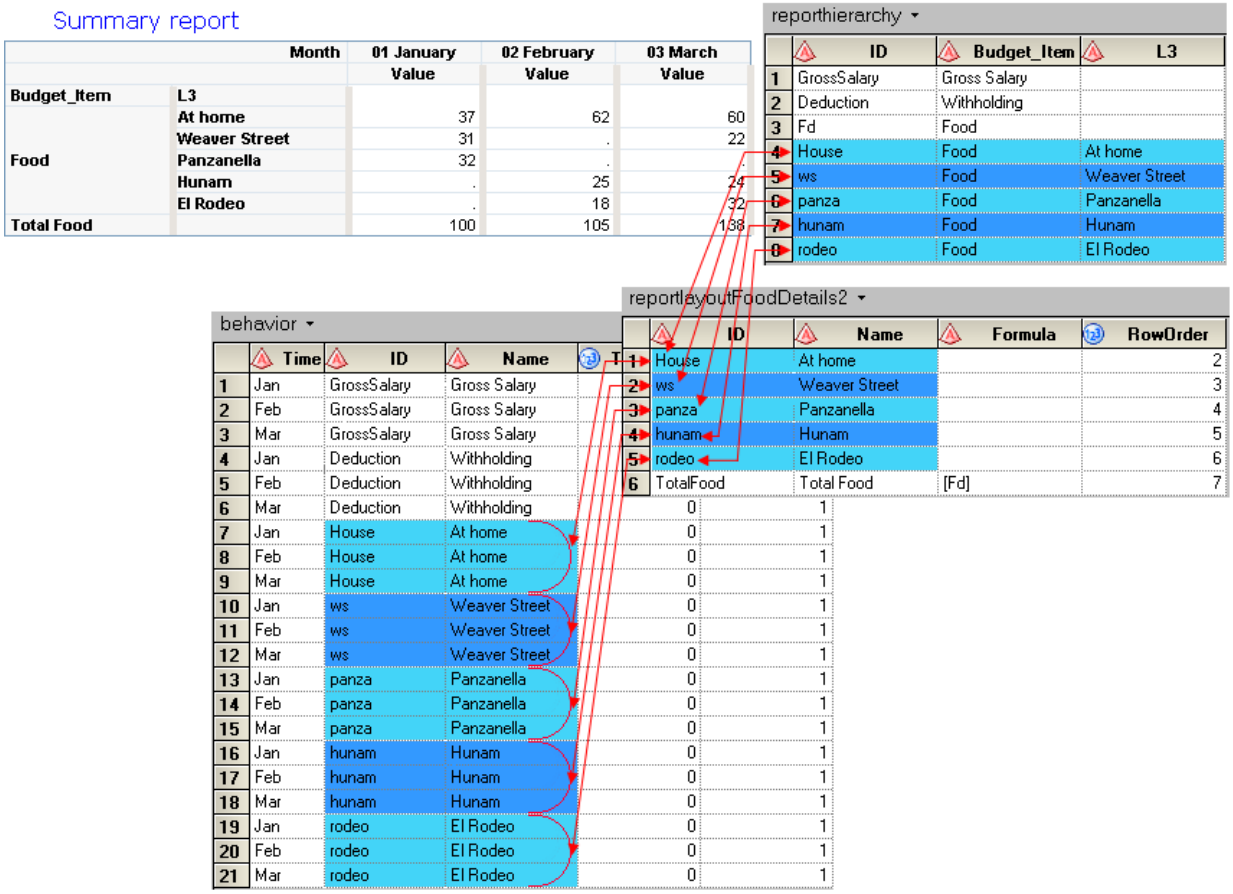
behavior

	Time	ID	Name		
1	Jan	GrossSalary	Gross Salary		
2	Feb	GrossSalary	Gross Salary	0	1
3	Mar	GrossSalary	Gross Salary	0	1
4	Jan	Deduction	Withholding	0	1
5	Feb	Deduction	Withholding	0	1
6	Mar	Deduction	Withholding	0	1
7	Jan	House	At home	0	1
8	Feb	House	At home	0	1
9	Mar	House	At home	0	1
10	Jan	ws	Weaver Street	0	1
11	Feb	ws	Weaver Street	0	1
12	Mar	ws	Weaver Street	0	1
13	Jan	panza	Panzanella	0	1
14	Feb	panza	Panzanella	0	1
15	Mar	panza	Panzanella	0	1
16	Jan	hunam	Hunam	0	1
17	Feb	hunam	Hunam	0	1
18	Mar	hunam	Hunam	0	1
19	Jan	rodeo	El Rodeo	0	1
20	Feb	rodeo	El Rodeo	0	1
21	Mar	rodeo	El Rodeo	0	1

Third Report Layout

In this final example we look at a third alternative report layout. In this layout, all the non-formula rows are leaf items in the report hierarchy. Thus, not only do these rows in the report layout have corresponding rows in the report hierarchy, they also have corresponding rows in the behavior table. And, again you can see that not every row in the report hierarchy has a corresponding row in the report layout—GrossSalary and Deduction in the report hierarchy do not have corresponding rows in the report layout.

Also, notice the effect on the summary report of including leaf nodes in the report layout. Initially the report displays with these items expanded. In general, a report opens expanded to the lowest level that is represented in the report layout. If there are levels in the report hierarchy lower than the lowest levels in the report layout, then those levels must be expanded by a person viewing the report to be visible.



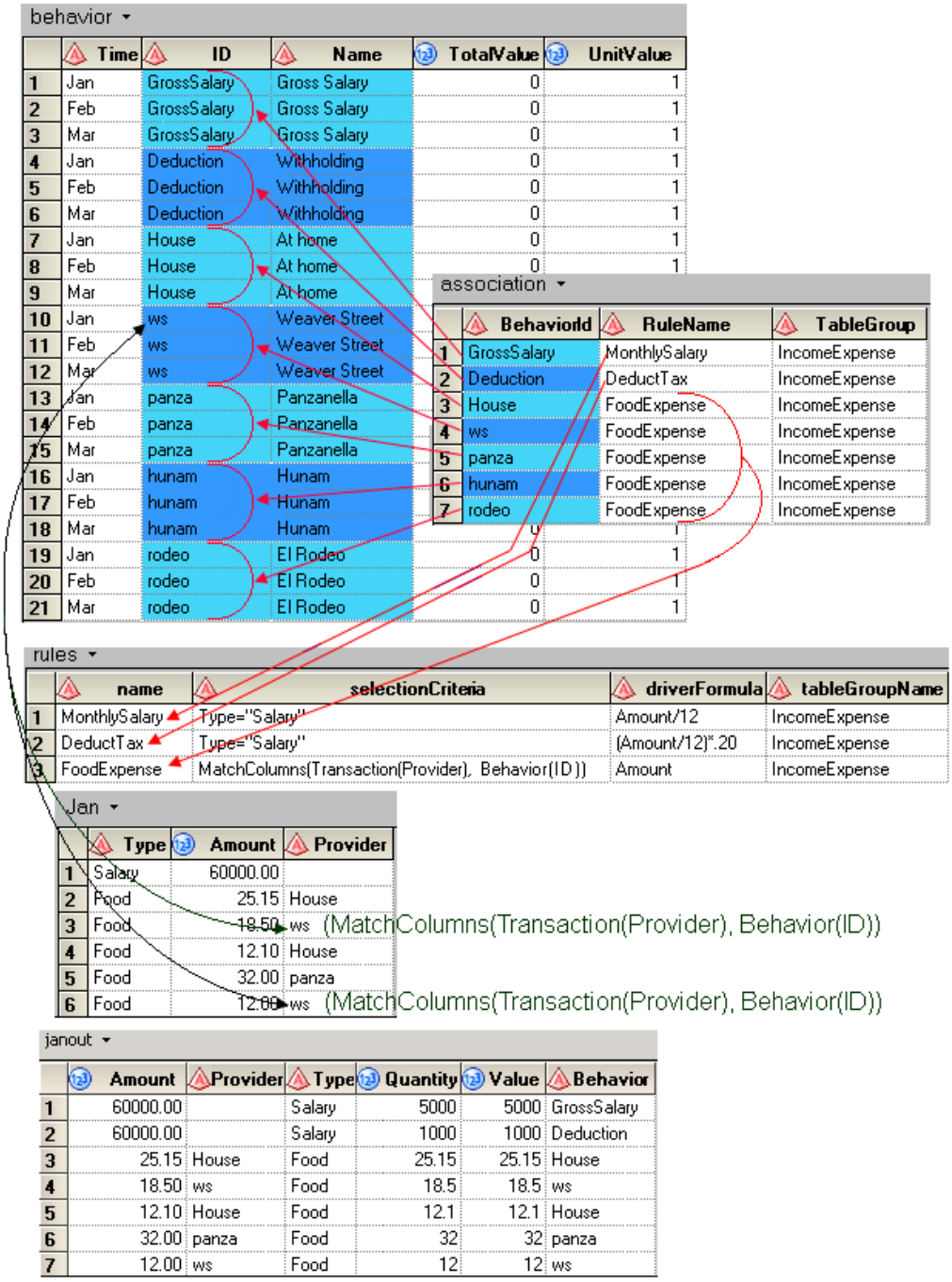
Behavior Table, Rule Association Table, and Rules

Looking at the behavior table, the rule association table, and the rules table, we can see that every row in the behavior table is associated, via the rule association table, with a row in the rules table. Multiple behaviors can be associated with the same rule.

The rule, FoodExpense, provides a good example of the use of Match Behavior in the selection criteria.To calculate how much was spent at Weaver Street restaurant, for example, it is not sufficient to select all transactions that are Food expenses. You must select only those Food expenses that were incurred at Weaver Street. This is accomplished by using the selection criterion **`MatchColumns(Transaction(Provider), Behavior(ID))`**. For the month of January, the behavior whose ID is **`ws`** (Weaver Street) matches two transactions whose value in the Provider column is **`ws`**.

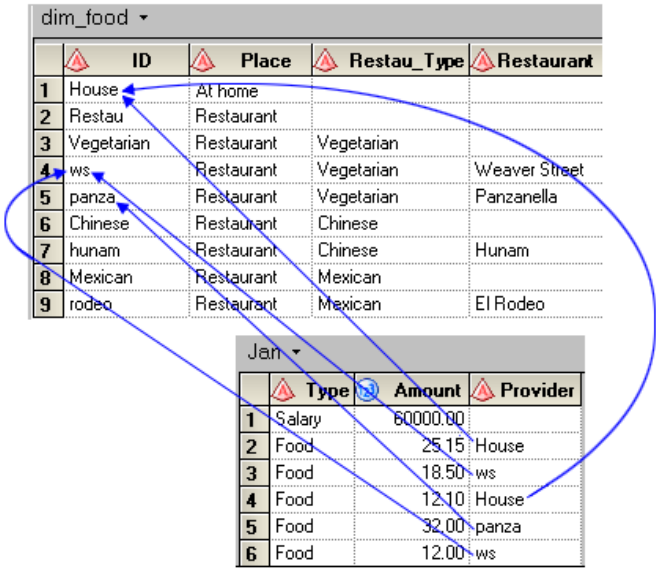
You can see how each row in the output transaction table is produced by the application of a rule to a row of the input transaction table.

Note: This example of MatchColumns is atypical because the column in the behavior table being compared for a match is one of the required behavior table columns. Usually, the column is one that you will have specifically added to the behavior table to enable a match comparison.



Dimension Table and Transaction Tables

The model includes a custom dimension, dim_food, which contains more information about food expenses than the report hierarchy. The Provider column in each row of the transaction table points to one of the IDs in the dimension table.



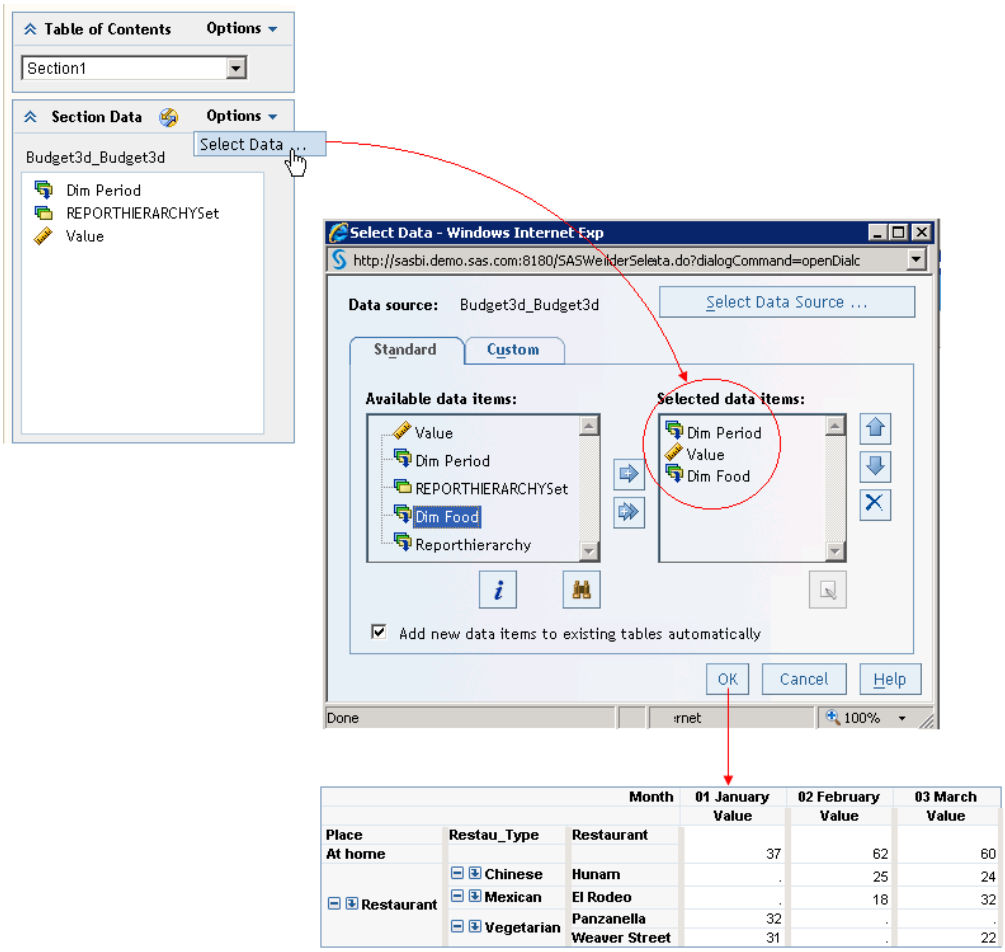
dim_food

	ID	Place	Restau_Type	Restaurant
1	House	At home		
2	Restau	Restaurant		
3	Vegetarian	Restaurant	Vegetarian	
4	ws	Restaurant	Vegetarian	Weaver Street
5	panza	Restaurant	Vegetarian	Panzanella
6	Chinese	Restaurant	Chinese	
7	hunam	Restaurant	Chinese	Hunam
8	Mexican	Restaurant	Mexican	
9	rodeo	Restaurant	Mexican	El Rodeo

Jan

	Type	Amount	Provider
1	Salary	60000.00	
2	Food	25.15	House
3	Food	18.50	ws
4	Food	12.10	House
5	Food	32.00	panza
6	Food	12.00	ws

Thus, because rows in the transaction table can be identified as belonging to a custom dimension, you can open that dimension table in the cube viewer, SAS Web Report Studio, to see details regarding the dimension. The following graphic shows the report that results when you select Dim Food in the cube viewer. In the report you can see how much was spent on what type of restaurant—vegetarian, Mexican, Chinese—information that was not available in the original report.



			Month	01 January	02 February	03 March
				Value	Value	Value
Place	Restau_Type	Restaurant				
At home				37	62	60
Restaurant	Chinese	Hunam		.	25	24
	Mexican	El Rodeo		.	18	32
	Vegetarian	Panzanella		32	.	.
		Weaver Street		31	.	22

An Additional Hierarchy and Layout

Notice that in the report on the Food dimension (shown again here) there is no monthly total for food expenses. Unlike with a report hierarchy, with a custom dimension there is no associated report layout that contains rows to perform a calculation.

Food dimension report

		Month	01 January	02 February	03 March
			Value	Value	Value
Place	Restau_Type	Restaurant			
At home			37	62	60
Restaurant	Chinese	Hunam	.	25	24
	Mexican	El Rodeo	.	18	32
	Vegetarian	Panzanella	32	.	.
		Weaver Street	31	.	22

If you want to perform a calculation on the information contained in a dimension table, then you can create an additional report hierarchy and report layout for the model. Each model can have multiple hierarchies and multiple layouts to produce multiple reports. The following graphic shows an additional hierarchy and layout used to create a report that shows only food expenses and the total spent each month.

As you can see, the report hierarchy duplicates much of the information in the custom dimension in order to produce a report showing total food expenses. Even though this example does not show it, the custom dimension can contain additional information (such as the tip left at each restaurant or the number of diners at each meal) that is not incorporated into the report hierarchy. Thus, even though the report hierarchy might contain much of the information of a custom dimension, you might still want to open the custom dimension in SAS Web Report Studio to see more information.

			Month	01 January	02 February	03 March
				Value	Value	Value
Budget_Item	**Genre**	**Location**				
At home				37	62	60
Restaurant	**Chinese**	**Hunam**		.	25	24
	Mexican	**El Rodeo**		.	18	32
	Vegetarian	**Panzanella**		32	.	.
		Weaver Street		31	.	22
Total Food				100	105	138

reporthierarchy2

	ID	Budget_Item	Genre	Location
1	GrossSalary	Gross Salary		
2	Deduction	Withholding		
3	House	At home		
4	Restau	Restaurant		
5	Vegetarian	Restaurant	Vegetarian	
6	ws	Restaurant	Vegetarian	Weaver Street
7	panza	Restaurant	Vegetarian	Panzanella
8	Chinese	Restaurant	Chinese	
9	hunam	Restaurant	Chinese	Hunam
10	Mexican	Restaurant	Mexican	
11	rodeo	Restaurant	Mexican	El Rodeo

reportlayout2

	ID	Name	Formula	RowOrder
1	House	At home		1
2	Restau	Restaurant		2
3	TotalFood	Total Food	[House]+[Restau]	3

dim_food

	ID	Place	Restau_Type	Restaurant
1	House	Home		
2	Restau	Restaurant		
3	Vegetarian	Restaurant	Vegetarian	
4	ws	Restaurant	Vegetarian	Weaver Street
5	panza	Restaurant	Vegetarian	Panzanella
6	Chinese	Restaurant	Chinese	
7	hunam	Restaurant	Chinese	Hunam
8	Mexican	Restaurant	Mexican	
9	rodeo	Restaurant	Mexican	El Rodeo

Changing the Period Dimension

Notice that the reports show no total for the year. To display an annual total, you can change the period dimension by adding a column and row for the year, as shown in the following graphic.

dim_period

Filter and Sort | Query Builder | Data | Describ

	ID	L1_Year	Month
1	2010	2010	
2	Jan	2010	01 January
3	Feb	2010	02 February
4	Mar	2010	03 March
5	Apr	2010	04 April
6	May	2010	05 May
7	Jun	2010	06 June
8	Jul	2010	07 July
9	Aug	2010	08 August
10	Sept	2010	09 September
11	Oct	2010	10 October
12	Nov	2010	11 November
13	Dec	2010	12 December

The following graphic shows two views of the original report using the new period dimension. At the top of the graphic you can see the collapsed report showing the total for the year. And, at the bottom of the graphic, you see the expanded report showing the total for each month.

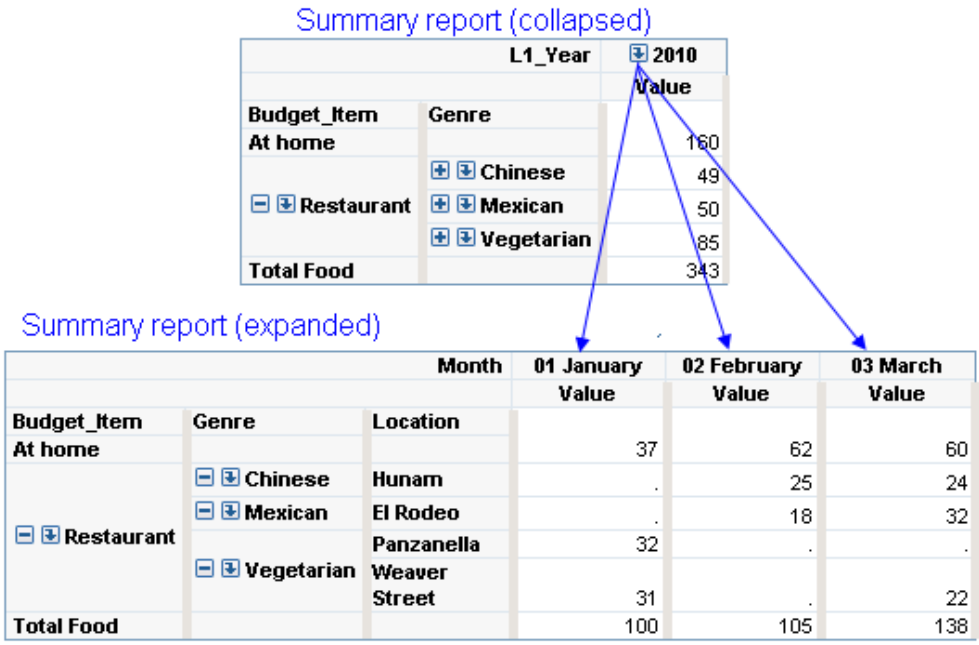
Summary report (collapsed)

	L1_Year	2010
		Value
Budget_Item	L3	
Gross Salary		15000
Withholding		3000
Net Salary		12000
Food	At home	160
	El Rodeo	50
	Hunam	49
	Panzanella	32
	Weaver Street	53
Total Food		343
Bottom Line		11657

Summary report (expanded)

	Month	01 January	02 February	03 March
		Value	Value	Value
Budget_Item	L3			
Gross Salary		5000	5000	5000
Withholding		1000	1000	1000
Net Salary		4000	4000	4000
Food	At home	37	62	60
	El Rodeo	.	18	32
	Hunam	.	25	24
	Panzanella	32	.	.
	Weaver Street	31	.	22
Total Food		100	105	138
Bottom Line		3900	3895	3862

In the following graphic, two views of a report show the additional hierarchy and layout that concentrate on food expenses. At the top of the graphic is the collapsed report showing only the total expense for each type of restaurant for the entire year. At the bottom of the graphic is an expanded report showing each individual restaurant and the total for the month.



An Alternative Behavior Table

This section shows a model that uses an alternative behavior table. The behavior table that we have used up to now contains much information on food expenses, with a row for each possible restaurant expense and a row for eating at home. This level of detail in the behavior table is appropriate if you want to see such data in the resulting OLAP report. A behavior table represents the lowest level of drill down in an OLAP view. However, if you want to see only total food expenses, then the behavior table used up to now is too complicated. It can be replaced by the alternative behavior table (shown in the following graphic) that contains a single row for each month of food expense. The graphic also shows a new report hierarchy with the alternative behavior table and a new report layout—each of which is displayed next to the corresponding original report hierarchy and report layout.

Original behavior table

behavior

	Time	ID	Name	TotalValue	UnitValue
1	Jan	GrossSalary	Gross Salary	0	1
2	Feb	GrossSalary	Gross Salary	0	1
3	Mar	GrossSalary	Gross Salary	0	1
4	Jan	Deduction	Withholding	0	1
5	Feb	Deduction	Withholding	0	1
6	Mar	Deduction	Withholding	0	1
7	Jan	House	At home	0	
8	Feb	House	At home		
				0	1
		ws	Weaver Street	0	1
12	Mar	ws	Weaver Street		
13	Jan	panza	Panzanella		
14	Feb	panza	Panzanella		
15	Mar	panza	Panzanella		
16	Jan	hunam	Hunam		
17	Feb	hunam	Hunam		
18	Mar	hunam	Hunam		
19	Jan	rodeo	El Rodeo		
20	Feb	rodeo	El Rodeo		
21	Mar	rodeo	El Rodeo		

Alternative behavior table

behavior

	Time	ID	Name	TotalValue	UnitValue
1	Jan	GrossSalary	Gross Salary	0	1
2	Feb	GrossSalary	Gross Salary	0	1
3	Mar	GrossSalary	Gross Salary	0	1
4	Jan	Deduction	Withholding	0	1
5	Feb	Deduction	Withholding	0	1
6	Mar	Deduction	Withholding	0	1
7	Jan	Fd	Food	0	1
8	Feb	Fd	Food	0	1
9	Mar	Fd	Food	0	1

Original report hierarchy

reporthierarchy

	ID	Budget_Item	L3
1	GrossSalary	Gross Salary	
2	Deduction	Withholding	
3	Fd	Food	
4	House	Food	At home
5	ws	Food	Weaver Street
6	panza	Food	Panzanella
7	hunam	Food	Hunam
8	rodeo	Food	El Rodeo

New report hierarchy for the alternative behavior table

reporthierarchy

	ID	Budget_Item
1	GrossSalary	Gross Salary
2	Deduction	Withholding
3	Fd	Food

Original report layout

reportlayout

	ID	Name	Formula	RowOrder
1	GrossSalary	Gross Salary		1
2	Deduction	Withholding		2
3	NetSalary	Net Salary	[GrossSalary]-[Deduction]	3
4	Fd	Food		
5	TotalFood	Total Food	[Fd]	
6	BottomLine	Bottom Line	[NetSal	

New report layout for the alternative behavior table

reportlayout

	ID	Name	Formula	RowOrder
1	GrossSalary	Gross Salary		1
2	Deduction	Withholding		2
3	NetSalary	Net Salary	[GrossSalary]-[Deduction]	3
4	Fd	Food		4
5	BottomLine	Bottom Line	[NetSalary]-[Fd]	6

The following graphic shows the summary report that results from the alternative behavior table, report hierarchy, and report layout as contrasted with the original summary report resulting from the original behavior table. Notice that detailed information on food expenses is missing from the new report because that information is missing from the behavior table.

Original report

	Month	01 January	02 February	03 March
		Value	Value	Value
Budget_Item	L3			
Gross Salary		5000	5000	5000
Withholding		1000	1000	1000
Net Salary		4000	4000	4000
Food	At home	37	62	60
	El Rodeo	.	18	32
	Hunam	.	25	24
	Panzanella	32	.	.
	Weaver Street	31	.	22
Total Food		100	105	138
Bottom Line		3900	3895	3862

This data is missing from the new report

New report

Month	01 January	02 February	03 March
	Value	Value	Value
Budget_Item			
Gross Salary	5000	5000	5000
Withholding	1000	1000	1000
Net Salary	4000	4000	4000
Food	100	105	138
Bottom Line	3900	3895	3862

The following graphic shows all of the tables that make up the new model containing the alternative behavior table as well as the resulting summary report.

dim_period

	ID	L1_Year	Month
1	2010	2010	
2	Jan	2010	01 January
3	Feb	2010	02 February
4	Mar	2010	03 March
5	Apr	2010	04 April
6	May	2010	05 May
7	Jun	2010	06 June
8	Jul	2010	07 July
9	Aug	2010	08 August
10	Sept	2010	09 September
11	Oct	2010	10 October
12	Nov	2010	11 November
13	Dec	2010	12 December

Summary report

Month	01 January	02 February	03 March
	Value	Value	Value
Budget_Item			
Gross Salary	5000	5000	5000
Withholding	1000	1000	1000
Net Salary	4000	4000	4000
Food	100	105	138
Bottom Line	3900	3895	3862

behavior

	Time	ID	Name	TotalValue	UnitValue
1	Jan	GrossSalary	Gross Salary	0	1
2	Feb	GrossSalary	Gross Salary	0	1
3	Mar	GrossSalary	Gross Salary	0	1
4	Jan	Deduction	Withholding	0	1
5	Feb	Deduction	Withholding	0	1
6	Mar	Deduction	Withholding	0	1
7	Jan	Fd	Food	0	1
8	Feb	Fd	Food	0	1
9	Mar	Fd	Food	0	1

reportlayout

	ID	Name	Formula	RowOrder
1	GrossSalary	Gross Salary		1
2	Deduction	Withholding		2
3	NetSalary	Net Salary	[GrossSalary]-[Deduction]	3
4	Fd	Food		4
5	BottomLine	Bottom Line	[NetSalary]-[Fd]	6

reporthierarchy

	ID	Budget_Item
1	GrossSalary	Gross Salary
2	Deduction	Withholding
3	Fd	Food

rules

	name	selectionCriteria	driverFormula	tableGroupName
1	MonthlySalary	Type="Salary"	Amount/12	IncomeExpense
2	DeductTax	Type="Salary"	(Amount/12)*.20	IncomeExpense
3	FoodExpense	Type="Food"	Amount	IncomeExpense

association

	BehaviorId	RuleName	TableGroup
1	GrossSalary	MonthlySalary	IncomeExpense
2	Deduction	DeductTax	IncomeExpense
3	Fd	FoodExpense	IncomeExpense

dim_food

	ID	Place	Restau_Type	Restaurant
1	House	At home		
2	Restau	Restaurant		
3	Vegetarian	Restaurant	Vegetarian	
4	ws	Restaurant	Vegetarian	Weaver Street
5	panza	Restaurant	Vegetarian	Panzanella
6	Chinese	Restaurant	Chinese	
7	hunam	Restaurant	Chinese	Hunam
8	Mexican	Restaurant	Mexican	
9	rodeo	Restaurant	Mexican	El Rodeo

Jan

	Type	Amount	Provider
1	Salary	60000.00	
2	Food	25.15	House
3	Food	18.50	ws
4	Food	12.10	House
5	Food	32.00	panza
6	Food	12.00	ws

An input transaction table

Report showing the food dimension (dim_food)

		Month	01 January	02 February	03 March
			Value	Value	Value
Place	Restau_Type	Restaurant			
At home			37	62	60
Restaurant	Chinese	Hunam	.	25	24
	Mexican	El Rodeo	.	18	32
	Vegetarian	Panzanella	32	.	.
		Weaver Street	31	.	22

An output transaction table

jan1out

	Amount	Provider	Type	Quantity	Value	Behavior
1	60000.00		Salary	5000	5000	GrossSalary
2	60000.00		Salary	1000	1000	Deduction
3	25.15	House	Food	25.15	25.15	Fd
4	18.50	ws	Food	18.5	18.5	Fd
5	12.10	House	Food	12.1	12.1	Fd
6	32.00	panza	Food	32	32	Fd
7	12.00	ws	Food	12	12	Fd

Notice that in this model a user can open a report showing the food dimension, dim_food, as discussed in "Dimension Table and Transaction Tables" on page 168.

		Month	01 January	02 February	03 March
			Value	Value	Value
Place	Restau_Type	Restaurant			
At home			37	62	60
Restaurant	Chinese	Hunam	.	25	24
	Mexican	El Rodeo	.	18	32
	Vegetarian	Panzanella	32	.	.
		Weaver Street	31	.	22

The same report is also available in our previous model because both models contain exactly the same custom dimension, dim_food. This food report's availability illustrates the fact that even though some information may be missing from the behavior table, it can be accessed in custom dimensions, which are also viewable as OLAP views.

Chapter 17

Finishing Up

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 custerm 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, decision-makers 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 - presenting 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

  http://www.sas.com/

- SAS Worldwide Training

  http://support.sas.com/training/index.html

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

  http://www.bettermanagement.com/

- 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.

  http://www.sas.com/solutions/index.html

- 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

  http://www.sas.com/technologies/bi/index.html

- 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

  http://www.sas.com/technologies/analytics/index.html

- 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.

  http://www.sas.com/solutions/crm/index.html