**ABSTRACT**
Creating a "one-row-per-subject" data mart is a fundamental task when preparing data for data mining. To answer the underlying business question, the analyst or data-mart programmer is challenged to distill the relevant information from various data sources.

Creating a data mart involves more than reading columns from a source table to the data mart. It also includes the aggregation or transposition of observations from "multiple-row-per-subject" tables like transactional tables and time histories. This process is a critical success factor for being able to answer the business question or to have good predictors available for a target event or target value.

This paper shows the details of input data structures for a "one-row-per-subject" data mart. It also discusses the "one-row-per-subject" paradigm from a technical and a business point of view, and shows how data, some of which include hierarchical dependencies, is aggregated into a "single-row-per-subject." A comprehensive example shows how a "one-row-per-subject" data mart is created from various data sources.

**INTRODUCTION**
Besides selecting and tuning the appropriate statistical model, data availability and data preparation are fundamental to the successful performance of an analytical model. Depending on the business questions and on the properties of the statistical model, different data mart structures, like the one-row-per-subject data mart that is necessary for predictive modeling or the multiple-row-per-subject data mart that is required for market basket analysis, must be assembled.

Answering a business problem usually includes the following steps:

1. Define the business problem. A lot of collaboration and discussion between business people, analysts, and IT people is needed in order to evaluate what exactly is needed.
2. Based on the definition of the business problem, determine what data is available to help you answer the business question(s), particularly observational data rather than data that are specifically gathered for an analysis.
3. Determine how to prepare the data for analytical modeling and the analytical data mart.
4. Feed the analytical data mart into the analytical procedure to analyze relationships, predict values and events, or segment subjects, to name a few tasks.

There are two main factors that influence the quality of an analytical model:

- The selection and tuning of the appropriate analytical model
- The availability and preparation of data that is suitable to answer the business question.

**SUCCESS FACTORS FOR ANALYTICS**
In statistical theory and literature there is much focus on the selection and tuning of an analytical model. There are many books and courses that focus on predictive modeling--training neural networks, creating decisions trees, or building regression models. For all these tasks a lot of knowledge is needed and has proven to be a critical success factor for good analytical results.

Besides the importance of analytical skills, however, having good and relevant data to help answer business questions is also very important.
SUCCESS FACTORS FOR DATA

The availability of the data for creating the analytical data mart is an important consideration to evaluate early in a project. Determining whether historic snapshots of the data that are necessary for an analysis will be available is a long-term planning issue and is not something that can be determined at the beginning of the data preparation process.

The adequate preparation of the data is an important success factor for analysis. Even if the right data sources are available and can be accessed, appropriately preparing the data is an essential pre-requisite for a good model. In the author’s experience with numerous projects, a missing or improperly derived variable or aggregation cannot be covered up by any clever parameterization of the model.

This paper focuses on the adequate preparation of the data. This paper contains the content of selected chapters from a soon to be released book Data Preparation for Analytics written by Gerhard Svolba (read more about this book in the section “Recommended Reading”). Following is a discussion of conceptual relationships and data structures when creating a one-row-per-subject data mart. The creation process is further illustrated by a case study.

THE ONE-ROW-PER-SUBJECT PARADIGM

ANALYSIS SUBJECT

Analysis subjects are entities that are being analyzed and the analysis results are interpreted in their context. Analysis subjects are therefore also the basis for the structure of analysis tables.

Examples of analysis subjects are:

- Persons: Depending on the area of the analysis the analysis subjects have more specific names like patients in medical statistics, customers in marketing analytics, or applicants in credit scoring.
- Animals: Piglets, for example, are analyzed in feeding experiments; rats are analyzed in pharmaceutical experiments.
- Parts of the body system: In medical research analysis subjects can also be parts of the body system like arms (left arm compared to the right arm), shoulders or hips. Note that from a statistical point of view the validity of the assumptions of the respective analysis methods have to be checked, if dependent observations per person are used in the analysis.
- Objects: Things like cash machines in cash demand prediction, cars in quality control in the automotive industry, or products in product analysis can be subjects.
- Legal entities: Examples are companies, contracts, accounts, and applications.
- Geography: Regions or plots in agricultural studies or reservoirs in the maturity prediction of fields in the oil and gas industries are examples.

Analysis subjects are therefore the heart of each analysis. The features of analysis subjects are measured, processed, and analyzed. In deductive statistics the features of the analysis subjects in the sample are used to infer the properties of the similar subjects in the larger population.

IMPORTANCE AND FREQUENT OCCURRENCE OF THIS TYPE OF DATA MART

The one-row-per-subject data mart is important for statistical and data mining analyses. Many business questions are answered based on a data mart of this type. The following list presents business examples:

- Prediction of events such as
  - Campaign response
  - Contract renewal
  - Contract cancellation (attrition or churn)
  - Insurance claim
  - Default on credit or loan payback
  - Identification of risk factors for a certain disease

- Prediction of values such as
  - Amount of loan that can be paid back
  - Customer turnover and profit for the next period
  - Time intervals like the remaining lifetime of a customer or patient
- Time until next purchase
- Value of next insurance claim

- Segmentation of customers

The following analytical methods require a one-row-per-subject data mart:

- Regression analysis
- Analysis-of-variance analysis
- Neural networks
- Decision trees
- Survival analysis
- Cluster analysis
- Principle components analysis and factor analysis

PUTTING ALL INFORMATION INTO ONE

The above application examples coupled with the underlying algorithm requires that all the data source information be aggregated into one row. Central to this type is the fact that we have to put all information per analysis subject into one row. We will therefore call this the **one-row-per-subject data mart paradigm** in that all information has to be put into a single row. Multiple observations per analysis subject must not appear in additional rows, but have to be converted over into additional columns.

Obviously, putting all information into one row is very simple if we have no multiple observations per analysis subject. The values are then simply read into the analysis data mart and derived variables are being added.

If we have multiple observations per analysis subject it is more effort to put all information into one row. Here we have to cover two aspects:

- The technical aspect—how multiple-row-per-subject data can be converted into one-row-per-subject data.
- The business aspect—which aggregations, derived variables, and so on can best be condensed from the multiple rows into columns.

The process of taking data from tables that have one-to-many relationships and putting them into a rectangular one-row-per-subject analysis table has many names: transposing, denormalizing, “making flat”, pivoting, and others.

Figure 1 illustrates the creation of a one-row-per-subject data mart in terms of an entity relationship diagram.

![Figure 1. Creation of a One-Row-Per-Subject Data Mart]
THE TECHNICAL POINT OF VIEW

From a technical perspective, you can solve the task of putting all information into one row using two main techniques:

- Transposing. Here we transpose the multiple rows per subject into columns. This technique can also be named as the “pure way,” as we take all data from the rows and represent it in columns.
- Aggregating. Here we aggregate the information from the columns into an aggregated value per analysis subject. We perform information reduction by trying to express the content of the original data into descriptive measures that are derived from the original data.

**Transposing**

Here is a very simple example with three multiple observations per subject (Table 1).

<table>
<thead>
<tr>
<th>PatNr</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Male</td>
</tr>
<tr>
<td>2</td>
<td>Female</td>
</tr>
<tr>
<td>3</td>
<td>Male</td>
</tr>
</tbody>
</table>

Table 1. Base Table with Static Information per Patient (analysis subject)

Next, we have a table with repeated measurements per patient (Table 2).

<table>
<thead>
<tr>
<th>PatNr</th>
<th>Measurement</th>
<th>Cholesterol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>212</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>220</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>240</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>150</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>145</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>148</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>301</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>280</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>275</td>
</tr>
</tbody>
</table>

Table 2. Table with Multiple Observations per Patient (analysis subject)

We have to bring data from the table with multiple observations per patient into a form that can be joined to the base table on a 1:1 basis. We therefore transpose Table 2 by patient number and bring all repeated elements into columns (Table 3).

<table>
<thead>
<tr>
<th>PatNr</th>
<th>Cholesterol 1</th>
<th>Cholesterol 2</th>
<th>Cholesterol 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>212</td>
<td>220</td>
<td>240</td>
</tr>
<tr>
<td>2</td>
<td>150</td>
<td>145</td>
<td>148</td>
</tr>
<tr>
<td>3</td>
<td>301</td>
<td>280</td>
<td>275</td>
</tr>
</tbody>
</table>

Table 3. Multiple Observations in a Table Structure with One Row per Patient

This information is then joined with the base table and we retrieve our analysis data mart in the requested data structure (Table 4).

<table>
<thead>
<tr>
<th>PatNr</th>
<th>Gender</th>
<th>Cholesterol 1</th>
<th>Cholesterol 2</th>
<th>Cholesterol 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Male</td>
<td>212</td>
<td>220</td>
<td>240</td>
</tr>
<tr>
<td>2</td>
<td>Female</td>
<td>150</td>
<td>145</td>
<td>148</td>
</tr>
<tr>
<td>3</td>
<td>Male</td>
<td>301</td>
<td>280</td>
<td>275</td>
</tr>
</tbody>
</table>

Table 4. The Final One-Row-Per-Subject Data Mart
Notice that we brought all information 1:1 from the multiple-row-per-subject data set to the final data set by transposing the data. This data structure is suitable for analyses such as repeated measurements for analysis of variance.

**Aggregation**

If we aggregate the information from the multiple-row-per-subject data set, for example, by using descriptive statistics such as the median, the minimum and maximum, or the interquartile range, we do not bring the original data 1:1 to the one-row-per-subject data set, but instead analyze a condensed version of the data. The data can then look like that shown in Table 5.

<table>
<thead>
<tr>
<th>PatNr</th>
<th>Gender</th>
<th>Cholesterol_Mean</th>
<th>Cholesterol_Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Male</td>
<td>224</td>
<td>14.4</td>
</tr>
<tr>
<td>2</td>
<td>Female</td>
<td>148</td>
<td>2.5</td>
</tr>
<tr>
<td>3</td>
<td>Male</td>
<td>285</td>
<td>13.8</td>
</tr>
</tbody>
</table>

Table 5. Aggregated Data from Tables 1 and 2

We see that aggregations do not produce as many columns as transpositions do when we condense the information. The forms of aggregations, however, have to be carefully selected, because omitting an important and--from a business point of view--relevant aggregation means that information is lost. There are no "rules" for selecting the best aggregation measures for all the different business questions. Domain-specific knowledge is key when selecting them. In predictive analysis, for example, we try to create candidate predictors that reflect the properties of the underlying subject and its behavior as well as possible.

**THE BUSINESS POINT OF VIEW - TRANSPOSING OR AGGREGATING ORIGINAL DATA**

In the above section we saw two ways to bring multiple observation data into a one-row-per-subject structure: transposing and aggregating. Selecting the appropriate method depends on the business question and the analytical method.

We have already mentioned in the section above that there are analyses in which all data for a subject is needed in columns but no aggregation of observations makes sense. For example the repeated measurement analysis of variance needs the repeated measurement values transposed into columns. In cases where we want to calculate indicators describing the course of a time series--for example, the mean usage in months 1 to 3 before cancellation divided by the mean usage in months 4 to 6--the original values are needed in columns.

There are, however, a lot of situations in which a 1:1 transfer of data is technically not possible nor is it practical. In these situations aggregations make much more sense. Let’s look at the following questions:

- If we have the cholesterol values expressed in columns, is this the information we need for further analysis?
- What shall we do if we have not only three observations for one measurement variable per subject, but 100 repetitions for 50 variables? Transposing all these data would lead to 5000 variables.
- Do we in the analysis data mart really need the data 1:1 from the original table, or do we more precisely need the information in concise aggregations?
- In predictive analysis, do we need data that has good predictive power for the modeling target and suitably describes the relationship between target and input variables instead of the original data?

The answers to these questions depend on the business questions, the modeling technique, the data itself, and so on. However there are many cases, especially in data mining analysis, where clever aggregations of the data make much more sense than a 1:1 transposition. Determining how to best aggregate the data requires business and domain knowledge and a good understanding of the data.

Having said that, putting all information into one row is not just data management; we show in this paper ways to cleverly prepare data from multiple observations. Following is an overview of methods that enable the information to be extracted and represented from the multiple observations per subject.
When aggregating *measurement data* for a one-row-per-subject data mart we can use the following:

- simple descriptive statistics like sum, minimum, or maximum
- frequency counts and number of distinct values
- measures of location or dispersion like mean, median, standard deviation, the quartiles, or special quantiles
- concentration measures on the basis of cumulative sums
- statistical measures describing trends and relationships like regression coefficients or correlations

When analyzing multiple observations with *categorical data* that is aggregated per subject we can use the following:

- total counts
- frequency counts per category and percentages per category
- distinct counts
- concentration measures on the basis of cumulative frequencies

**HIERARCHIES: “AGGREGATING UP” AND “COPYING DOWN”**

**General**

In the case of *repeated measurements* over time we are usually aggregating this information by using the summary measurements described above. The aggregated values, such as means or sums, are then used at the subject level as a property of the corresponding subject.

In the case of *multiple observations because of hierarchical relationships*, we are also aggregating information from a lower hierarchical level to a higher one. However, we can also have information at a higher hierarchical level that has to be available or expressed to lower hierarchical levels. Also, information has to be shared between members of the same hierarchy, like the number of members at this hierarchy level.

**Example**

Consider an example in which we have the following relationships:

- In one HOUShold lives one or more CUSTOMER(s).
- One CUSTOMER can have one or more CREDIT CARD(s).
- On each CREDIT CARD a number of TRANSACTION(s) are associated.

For marketing purposes we do a segmentation of customers; therefore, CUSTOMER is chosen as the appropriate subject analysis level. When looking at the hierarchies, we aggregate information from the entity of CREDIT CARD and TRANSACTION to the CUSTOMER level and make information that is stored on the HOUSEHOLD level available to the entity CUSTOMER.

We next show how data from different hierarchical levels is made available to the analysis data mart. Note that for didactical purposes, we reduce the number of variables to a few.

**Information from the Same Level**

At the customer level we have the following variables directly available to our analysis data mart:

- customer birthday (on that basis the derived variable AGE is calculated)
- gender
- customer since (on that basis the duration of customer relationship is calculated)
Aggregating Up
In our example, we create the following variables on customer level that are aggregations from the lower levels CREDIT CARD and TRANSACTION.

- number of cards per customer
- number of transactions over all cards for the last 3 months
- sum of transaction values over all cards for the last 3 months
- average transaction value for the last 3 months (calculated on the basis of the values over all cards)

Dividing Down
At the HOUSEHOLD level the following information is available, that is copied down to the customer level.

- number of persons in the household
- geographic region type of the household with the values RURAL, HIGHLY RURAL, URBAN, HIGHLY URBAN.

Graphical Representation
Figure 2 shows the graphical representation in the form of an entity relationship diagram and the respective variable flows.

![Entity Relationship Diagram - Information Flows Between Hierarchies](image.png)

Conclusion
We see that information that is finally used at the analysis level CUSTOMER is retrieved from several levels. We have used simple statistical aggregations to illustrate this. In the following case study we go into more detail on this topic and also present the respective coding examples.

CASE STUDY – BUILDING A CUSTOMER DATA MART

INTRODUCTION
This section provides an example of how a one-row-per-subject table can be built using a number of data sources. In these data sources we only have in the CUSTOMER table one row per subject; in the other tables we have multiple rows per subject.
The task in this example is to combine these data sources to create a table that holds one row per customer. Therefore we have to aggregate the multiple observations per subject to one observation per subject.

Tables like this are frequently needed in data mining analyses such as prediction or segmentation. In our example, we assume that we can create data to predict the cancellation event of a customer.

THE BUSINESS CONTEXT

General
We want to create a data mart that reflects the properties and behaviors of customers. This table will be used to predict which customer has a high probability to leave the company.

Data Sources
In this example we will build a customer data mart for marketing purposes. Our basis is data from the following data sources:

- **Customer data**: demographic and customer baseline data
- **Account data**: information customer accounts
- **Leasing data**: data on leasing information
- **Call Center data**: data on Call center contacts
- **Score data**: data of value segment scores

From these data sources we want to build one table with one row per customer. We therefore have the classic one-row-per-subject paradigm in which we need to bring the repeated information from, for example, the account data to several columns.

Considerations for Predictive Modeling
Determining whether a customer has left the company can be derived from its value segment. The SCORES table (see Table 10) holds for each customer the ValueSegment for the previous month. If ValueSegment has the value ‘8. LOST’ then the customer cancelled his contract in the previous month. Otherwise ValueSegment has values such as ‘1. GOLD’, ‘2. SILVER’ which represent the customer value grading.

We define our snapshot date as June 1, 2003. This means that in our analysis we are allowed to use all information that is known on or before this date. We use the month June 2003 as the *offset window* and the month July 2003 as the *target window*. This means we want to predict which data cancellation events took place in July 2003. Therefore we have to consider the ValueSegment in the SCORES table per August 1, 2003. The offset window is used to train the model to predict an event that does not take place immediately after the snapshot date but after some time. This makes sense, because retention actions can not start immediately after the snapshot data. They start after data is loaded into the data warehouse and scored. Then the retention actions are defined.

Reasons for Multiple Observations per Customer
We have different reasons for multiple observations per customer in the different tables:

- In the SCORES table data, for example, we have multiple observations because of scores for different points in time.
- In the ACCOUNT table data we have already aggregated data over time, but we have multiple observations per customer because of different contract types.
- The CALLCENTER table is a classic form of a transactional data structure. We have on row per call center contact.
THE DATA
We have five tables available:

1. CUSTOMER
2. ACCOUNTS
3. LEASING
4. CALLCENTER
5. SCORES

These tables can be merged by the unique CUSTID variable. Notice that, for example, the ACCOUNT table already holds data that are aggregated from a transactional ACCOUNT table. We already have one row per account type that holds aggregated data over time.

The data model and the list of available variables can be seen in Figure 3.

Figure 3. Data Model for Creating the CUSTOMER DATA MART

The following tables give a snapshot of the contents of the tables (Tables 6 to 10).

Table 6. CUSTOMER Table

<table>
<thead>
<tr>
<th>CustID</th>
<th>Birthdate</th>
<th>Title</th>
<th>Branch</th>
<th>CustomerSince</th>
<th>Gender</th>
<th>MarriedStatus</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000002</td>
<td>2 Dec 1958</td>
<td>F1</td>
<td>01 Jan 2009</td>
<td>Male</td>
<td>Married</td>
</tr>
<tr>
<td>2</td>
<td>1000005</td>
<td>25 Jun 1947</td>
<td>F4</td>
<td>01 Apr 1999</td>
<td>Male</td>
<td>Single</td>
</tr>
<tr>
<td>3</td>
<td>1000006</td>
<td>10 Dec 1945</td>
<td>F4</td>
<td>01 Sep 1996</td>
<td>Female</td>
<td>Married</td>
</tr>
<tr>
<td>4</td>
<td>1000007</td>
<td>2 Jun 1934</td>
<td>F1</td>
<td>01 Sep 1997</td>
<td>Male</td>
<td>Married</td>
</tr>
<tr>
<td>5</td>
<td>1000008</td>
<td>1 Dec 1957</td>
<td>Dr</td>
<td>01 Jan 1996</td>
<td>Male</td>
<td>Single</td>
</tr>
<tr>
<td>6</td>
<td>1000009</td>
<td>1 Nov 1959</td>
<td>F2</td>
<td>01 Jul 2001</td>
<td>Male</td>
<td>Single</td>
</tr>
<tr>
<td>7</td>
<td>1000010</td>
<td>12 Aug 1962</td>
<td>F4</td>
<td>01 May 1996</td>
<td>Male</td>
<td>Single</td>
</tr>
<tr>
<td>8</td>
<td>1000011</td>
<td>12 May 1969</td>
<td>F2</td>
<td>01 Feb 1999</td>
<td>Male</td>
<td>Single</td>
</tr>
<tr>
<td>9</td>
<td>1000012</td>
<td>1 Feb 1967</td>
<td>F2</td>
<td>01 Dec 2001</td>
<td>Male</td>
<td>Married</td>
</tr>
<tr>
<td>10</td>
<td>1000013</td>
<td>1 Aug 2003</td>
<td>F2</td>
<td>01 Jul 2003</td>
<td>Male</td>
<td>Single</td>
</tr>
<tr>
<td>11</td>
<td>1000014</td>
<td>3 Dec 1936</td>
<td>F2</td>
<td>01 Jan 2002</td>
<td>Male</td>
<td>Single</td>
</tr>
<tr>
<td>12</td>
<td>1000015</td>
<td>12 Aug 1969</td>
<td>F2</td>
<td>01 Oct 2003</td>
<td>Male</td>
<td>Divorce</td>
</tr>
<tr>
<td>13</td>
<td>1000016</td>
<td>10 Jul 1961</td>
<td>F1</td>
<td>01 Nov 1993</td>
<td>Male</td>
<td>Married</td>
</tr>
<tr>
<td>14</td>
<td>1000017</td>
<td>20 Jan 1972</td>
<td>F1</td>
<td>01 Dec 1995</td>
<td>Male</td>
<td>Married</td>
</tr>
</tbody>
</table>
Table 7. ACCOUNTS Table

<table>
<thead>
<tr>
<th>CustID</th>
<th>AccountID</th>
<th>Balance</th>
<th>Interest</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000002</td>
<td>16853</td>
<td>150.42</td>
<td>5.00 Saving Account</td>
</tr>
<tr>
<td>2</td>
<td>1000002</td>
<td>16853</td>
<td>1558.42</td>
<td>5.00 Loan</td>
</tr>
<tr>
<td>3</td>
<td>1000003</td>
<td>13296</td>
<td>377.31</td>
<td>6.00 Savings Account</td>
</tr>
<tr>
<td>4</td>
<td>1000006</td>
<td>21992</td>
<td>2376.43</td>
<td>2.00 Funds</td>
</tr>
<tr>
<td>5</td>
<td>1000007</td>
<td>16799</td>
<td>1112.72</td>
<td>5.00 Funds</td>
</tr>
<tr>
<td>6</td>
<td>1000007</td>
<td>16789</td>
<td>1812.72</td>
<td>5.00 Savings Account</td>
</tr>
<tr>
<td>7</td>
<td>1000008</td>
<td>15987</td>
<td>3590.65</td>
<td>2.00 Funds</td>
</tr>
<tr>
<td>8</td>
<td>1000009</td>
<td>11907</td>
<td>1191.82</td>
<td>4.00 Build motor saving</td>
</tr>
<tr>
<td>9</td>
<td>1000009</td>
<td>11907</td>
<td>1191.82</td>
<td>4.00 Funds</td>
</tr>
<tr>
<td>10</td>
<td>1000009</td>
<td>11907</td>
<td>1191.82</td>
<td>4.00 Loan</td>
</tr>
<tr>
<td>11</td>
<td>1000014</td>
<td>10920</td>
<td>1090.46</td>
<td>5.00 Saving Account</td>
</tr>
<tr>
<td>12</td>
<td>1000014</td>
<td>10920</td>
<td>1100.46</td>
<td>4.00 Savings Account</td>
</tr>
<tr>
<td>13</td>
<td>1000014</td>
<td>10920</td>
<td>1100.46</td>
<td>1.00 Build motor saving</td>
</tr>
<tr>
<td>14</td>
<td>1000015</td>
<td>16853</td>
<td>2601.09</td>
<td>5.00 Saving Account</td>
</tr>
</tbody>
</table>

Table 8. LEASING Table

<table>
<thead>
<tr>
<th>Leasing Polizei</th>
<th>CustID</th>
<th>StartDate</th>
<th>EndDate</th>
<th>Value</th>
<th>AnnualRate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10001</td>
<td>1000002</td>
<td>13SEP1999</td>
<td>02APR2000</td>
<td>521763</td>
</tr>
<tr>
<td>2</td>
<td>10002</td>
<td>1000005</td>
<td>11FEB2002</td>
<td>17APR2003</td>
<td>560226</td>
</tr>
<tr>
<td>3</td>
<td>10003</td>
<td>1000006</td>
<td>26FEB2002</td>
<td>17APR2003</td>
<td>765661</td>
</tr>
<tr>
<td>4</td>
<td>10004</td>
<td>1000007</td>
<td>07OCT1999</td>
<td>07OCT2000</td>
<td>950047</td>
</tr>
<tr>
<td>5</td>
<td>10005</td>
<td>1000008</td>
<td>02OCT2000</td>
<td>02OCT2003</td>
<td>5276</td>
</tr>
<tr>
<td>6</td>
<td>10006</td>
<td>1000009</td>
<td>16OCT1998</td>
<td>16OCT2002</td>
<td>24331</td>
</tr>
<tr>
<td>7</td>
<td>10007</td>
<td>1000009</td>
<td>03OCT2001</td>
<td>18JUL2001</td>
<td>348232</td>
</tr>
<tr>
<td>8</td>
<td>10008</td>
<td>1000014</td>
<td>14MAC2000</td>
<td>21JUL2000</td>
<td>464272</td>
</tr>
<tr>
<td>9</td>
<td>10009</td>
<td>1000014</td>
<td>01JUL1997</td>
<td>19OCT2000</td>
<td>393984</td>
</tr>
<tr>
<td>10</td>
<td>10010</td>
<td>1000015</td>
<td>01JUL1997</td>
<td>01JUL1998</td>
<td>651408</td>
</tr>
<tr>
<td>11</td>
<td>10011</td>
<td>1000018</td>
<td>27AGU1995</td>
<td>28AGU2000</td>
<td>451405</td>
</tr>
</tbody>
</table>

Table 9. CALLCENTER Table

<table>
<thead>
<tr>
<th>CustID</th>
<th>ContactID</th>
<th>Date</th>
<th>Agent</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100003</td>
<td>15JUL2003</td>
<td>55 Telebanking</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>100014</td>
<td>02APR2003</td>
<td>94 Complaint</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>100014</td>
<td>02MAR2003</td>
<td>56 Complaint</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>100010</td>
<td>12JUN2003</td>
<td>25 Telebanking</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>100028</td>
<td>21FEB2003</td>
<td>25 Telebanking</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>100034</td>
<td>20MAY2003</td>
<td>25 Telebanking</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>100036</td>
<td>15JUL2003</td>
<td>81 Telebanking</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>100037</td>
<td>09JAN2003</td>
<td>32 Complaint</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>100037</td>
<td>10JUL2003</td>
<td>70 Complaint</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>100040</td>
<td>23APR2003</td>
<td>31 Complaint</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>100040</td>
<td>19MAY2003</td>
<td>68 Complaint</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>100041</td>
<td>10JUL2003</td>
<td>12 Telebanking</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>100042</td>
<td>04JUL2003</td>
<td>99 Telebanking</td>
<td></td>
</tr>
</tbody>
</table>

Table 10. SCORES Table
Transactional Data or Snapshot

As we can see from the data examples only tables SCORES and CALLCENTER are real transactional tables. When processing data from these tables, we need to make sure that no observations after our snapshot data 01JUN2003 are used.

The other tables we consider as snapshots from 01JUN2003—we do not have to worry about the correct time span for these tables.

THE PROGRAMS

We will create the CUSTOMER DATA MART in two steps:

1. We will create aggregations on a subject level of the ACCOUNT, LEASING, CALLCENTER, and SCORES tables.
2. We will join the resulting tables together and create the CUSTOMER DATA MART table.

Transforming the ACCOUNTS Table

With the ACCOUNTS table, we want to perform two aggregations:

- derive sums per customer
- calculate an account type split based on the balance

We use PROC MEANS to create the output data set ACCOUNTMTP that contains both aggregations per customer and aggregations per customer and account type. Notice that we have intentionally omitted the NWAY option in order to have the full set of _TYPE_ categories in the output data set.

```
PROC MEANS DATA = accounts NOPRINT;
CLASS CustID Type;
VAR Balance Interest;
OUTPUT OUT = AccountTmp(RENAME = (_FREQ_ = NrAccounts))
  SUM(Balance) = BalanceSum
  MEAN(Interest) = InterestMean;
RUN;
```

In the next step we use a DATA step with multiple output data sets to create:

- ACCOUNTSUM: that holds the aggregations on the customer level
- ACCOUNTTYPETMP: that is our basis for the account type split, which we need to transpose later on.

Notice that the creation of two output data sets is very elegant and we can rename the respective variables with a data set option. Based on the value of the _TYPE_ variable, we decide to which data set we output the respective observations.

```
DATA AccountSum(KEEP = CustID NrAccounts BalanceSum InterestMean)
  AccountTypeTmp(KEEP = CustID Type BalanceSum);
SET AccountTmp;
Type = COMPRESS(Type);
IF _TYPE_ = 2 THEN OUTPUT AccountSum;
ELSE IF _TYPE_ = 3 THEN OUTPUT AccountTypeTmp;
RUN;
```
Finally we need to transpose the data set with the ACCOUNTTYPES to have a one-row-per-customer structure:

```
PROC TRANSPOSE DATA = AccountTypeTmp
   OUT = AccountTypes(DROP = _NAME_ _LABEL_);
BY CustID;
VAR BalanceSum;
ID Type;
RUN;
```

**Transforming the LEASING Table**

For the leasing data we simply summarize the value and the annual rate per customer with the following statement:

```
PROC MEANS DATA = leasing NWAY NOPRINT;
CLASS CustID;
VAR Value AnnualRate;
OUTPUT OUT = LeasingSum(DROP = _TYPE_
   RENAME = (_FREQ_ = NrLeasing))
   SUM(Value) = LeasingValue
   SUM(AnnualRate) = LeasingAnnualRate;
RUN;
```

**Transforming the CALLCENTER Table**

We first define the snapshot date for the data in a macro variable.

```bash
%let snapdate = "01JUN2003"d;
```

Then we aggregate the CALLCENTER data over all customers using PROC FREQ.

```
PROC FREQ DATA = callcenter NOPRINT;
   TABLE CustID / OUT = CallCenterContacts(DROP = Percent RENAME = (Count = Calls));
   WHERE datepart(date) < &snapdate;
RUN;
```

Next we calculate the frequencies for the category COMPLAINT.

```
PROC FREQ DATA = callcenter NOPRINT;
   TABLE CustID / OUT = CallCenterComplaints(DROP = Percent RENAME = (Count = Complaints));
   WHERE Category = 'Complaint' and datepart(date) < &snapdate;
RUN;
```

**Transforming the SCORES Table**

We create three output data sets, one with the value segment in the actual (snapshot date) month, one with the value segment of one month ago, and one with the value segment in a future month in order to calculate the target variable.

```
DATA ScoreFuture(RENAME = (ValueSegment = FutureValueSegment))
   ScoreActual;
DATA ScoreLastMonth(RENAME = (ValueSegment = LastValueSegment));
SET Scores;
DROP Date;
IF Date = &snapdate THEN OUTPUT ScoreActual;
ELSE IF Date = INTNX('MONTH',&snapdate,-1) THEN OUTPUT ScoreLastMonth;
ELSE IF Date = INTNX('MONTH',&snapdate,2) THEN OUTPUT ScoreFuture;
RUN;
```
Notice that we used the INTNX function, which enables us to define the relevant points in time in a very elegant way. In this example the first INTNX function evaluates to 01MAY2003 and the second evaluates to 01AUG2003. We have already explained in the “Considerations for Predictive Modeling” section that this is needed for an offset-month.

Creating the CUSTOMER DATA MART

Finally we create the CUSTOMER DATA MART table by joining all intermediate result tables to the CUSTOMER table.

First we define a macro variable for the snapshot date in order to have a reference point for the calculation of age. (Notice that we repeated this macro variable assignment for didactical purposes here.)

```sas
%let snapdate = "01JUN2003"d;
```

Next we start defining our CUSTOMER DATA MART table. We use ATTRIB statements to have a detailed definition of the data set variables by specifying their order, their format, and their label. Notice that we aligned the FORMAT and LABEL statements for a clearer visual picture.

```sas
DATA CustomerMart;
ATTRIB /* Customer Baseline */
CustID FORMAT = 8. LABEL = "Customer ID"
Birthdate FORMAT = DATE9. LABEL = "Date of Birth"
Alter FORMAT = 8. LABEL = "Age (years)"
Gender FORMAT = $6. LABEL = "Gender"
MaritalStatus FORMAT = $10. LABEL = "Marital Status"
Title FORMAT = $10. LABEL = "Academic Title"
HasTitle FORMAT = 8. LABEL = "Has Title? 0/1"
Branch FORMAT = $5. LABEL = "Branch Name"
CustomerSince FORMAT = DATE9. LABEL = "Customer Start Date"
CustomerMonths FORMAT = 8. LABEL = "Customer Duration (months)"
;
ATTRIB /* Accounts */
HasAccounts FORMAT = 8. LABEL = "Customer has any accounts"
NrAccounts FORMAT = 8. LABEL = "Number of Accounts"
BalanceSum FORMAT = 8.2 LABEL = "All Accounts Balance Sum"
InterestMean FORMAT = 8.1 LABEL = "Average Interest"
Loan FORMAT = 8.2 LABEL = "Loan Balance Sum"
SavingAccount FORMAT = 8.2 LABEL = "Saving Account Balance Sum"
Funds FORMAT = 8.2 LABEL = "Funds Balance Sum"
LoanPct FORMAT = 8.2 LABEL = "Loan Balance Proportion"
SavingAccountPct FORMAT = 8.2 LABEL = "Saving Account Balance Proportion"
FundsPct FORMAT = 8.2 LABEL = "Funds Balance Proportion"
;
ATTRIB /* Leasing */
HasLeasing FORMAT = 8. LABEL = "Customer has any leasing contract"
NrLeasing FORMAT = 8. LABEL = "Number of leasing contracts"
LeasingValue FORMAT = 8.2 LABEL = "Totals leasing value"
LeasingAnnualRate FORMAT = 8.2 LABEL = "Total annual leasingrate"
;
ATTRIB /* Call Center */
HasCallCenter FORMAT = 8. LABEL = "Customer has any call center contact"
Calls FORMAT = 8. LABEL = "Number of call center contacts"
Complaints FORMAT = 8. LABEL = "Number of complaints"
ComplaintPct FORMAT = 8.2 LABEL = "Percentage of complaints"
;
ATTRIB /* Value Segment */
ValueSegment FORMAT = $10. LABEL = "Currently Value Segment"
LastValueSegment FORMAT = $10. LABEL = "Last Value Segment"
ChangeValueSegment FORMAT = 8.2 LABEL = "Change in Value Segment"

```
Next we merge the data set that we created in the previous steps and create logical variables for some tables.

```
MERGE Customer (IN = InCustomer)
    AccountSum (IN = InAccounts)
    AccountTypes
    LeasingSum (IN = InLeasing)
    CallCenterContacts (IN = InCallCenter)
    CallCenterComplaints
    ScoreFuture
    ScoreActual
    ScoreLastMonth;
BY CustID;
IF InCustomer;
```

In the next step we replace missing values with zero (0). These can be causally interpreted as zeros, for example, when transposing account types per customer and a customer does not have a certain account type.

```
ARRAY vars {*} Calls Complaints LeasingValue LeasingAnnualRate
    Loan SavingAccount Funds Buildassocsaving NrLeasing;
DO i = 1 TO dim(vars);
    IF vars{i}=. THEN vars{i}=0;
END;
DROP i;
```

Finally we calculate derived variables for the data mart.

```
/* Customer Baseline */
HasTitle = (Title ne "");
Alter = (&Snapdate-Birthdate)/365.25;
CustomerMonths = (&Snapdate- CustomerSince)/(365.25/12);
/* Accounts */
HasAccounts = InAccounts;
LoanPct = Loan / BalanceSum * 100;
SavingAccountPct = SavingAccount / BalanceSum * 100;
FundsPct = Funds / BalanceSum * 100;
/* Leasing */
HasLeasing = InLeasing;
/* Call Center */
HasCallCenter = InCallCenter;
ComplaintPct = Complaints / Calls *100;
/* Value Segment */
Cancel = (FutureValueSegment = '8. LOST');
ChangeValueSegment = (ValueSegment = LastValueSegment);
RUN;
```

Notice that we calculate the target variable CANCEL based on the fact that a customer has the ValueSegment entry '8. LOST'.

When ValueSegment entries change, we simply compare the current and the last ValueSegment entry. It would be more accurate to differentiate between an increase and a decreased ValueSegment entry. This would, however, be more complicated because we would need to consider all segment codes manually. Here it is more accurate to have the underlying numeric score available, which can be used for a simple numeric comparison.
The Resulting Data Set

The resulting data set looks like the following (Tables 11 to 13):

### Table 11. Customer Baseline Data in the CUSTOMER DATA MART

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Date of Birth</th>
<th>Age (years)</th>
<th>Gender</th>
<th>Marital Status</th>
<th>Academic Title</th>
<th>Highest?P</th>
<th>Branch Name</th>
<th>Customer Start Date</th>
<th>Customer Duration (months)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000002</td>
<td>26DEC1959</td>
<td>44</td>
<td>Male</td>
<td></td>
<td>0</td>
<td>F1</td>
<td>01JAN2000</td>
<td>41</td>
</tr>
<tr>
<td>2</td>
<td>1000005</td>
<td>25JUN1947</td>
<td>56</td>
<td>Male</td>
<td></td>
<td>1</td>
<td>F1</td>
<td>01APR1999</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>1000006</td>
<td>10DEC1945</td>
<td>57</td>
<td>Female</td>
<td></td>
<td>0</td>
<td>F1</td>
<td>01SEP1996</td>
<td>81</td>
</tr>
<tr>
<td>4</td>
<td>1000007</td>
<td>02JUN1934</td>
<td>69</td>
<td>Married</td>
<td></td>
<td>0</td>
<td>F1</td>
<td>01SEP1997</td>
<td>69</td>
</tr>
<tr>
<td>5</td>
<td>1000008</td>
<td>15DEC1957</td>
<td>45</td>
<td>Male</td>
<td></td>
<td>1</td>
<td>F1</td>
<td>01JAN1996</td>
<td>89</td>
</tr>
<tr>
<td>6</td>
<td>1000009</td>
<td>11MAR1959</td>
<td>44</td>
<td>Male</td>
<td></td>
<td>0</td>
<td>F2</td>
<td>01JUL2001</td>
<td>23</td>
</tr>
<tr>
<td>7</td>
<td>1000014</td>
<td>23AUG1952</td>
<td>51</td>
<td>Single</td>
<td></td>
<td>0</td>
<td>F4</td>
<td>01MAR1996</td>
<td>85</td>
</tr>
<tr>
<td>8</td>
<td>1000015</td>
<td>12MAR1997</td>
<td>44</td>
<td>Male</td>
<td></td>
<td>0</td>
<td>F2</td>
<td>01FEB1999</td>
<td>52</td>
</tr>
<tr>
<td>9</td>
<td>1000016</td>
<td>11FEB1967</td>
<td>36</td>
<td>Married</td>
<td></td>
<td>0</td>
<td>F2</td>
<td>01FEB2001</td>
<td>28</td>
</tr>
</tbody>
</table>

### Table 12. Account Data and Leasing Data in the CUSTOMER DATA MART

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Customer has any accounts?</th>
<th>Number of Accounts</th>
<th>Average Interest</th>
<th>Loan Balance Sum</th>
<th>Saving Account Balance Sum</th>
<th>Fonds Balance Sum</th>
<th>Loan Balance Propotion</th>
<th>Saving Account Balance Propotion</th>
<th>Fonds Balance Propotion</th>
<th>Customer has any leasing contract?</th>
<th>Number of leasing contracts</th>
<th>Total leasing value</th>
<th>Total annual leasing cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3100.84</td>
<td>5.0</td>
<td>1550.42</td>
<td>1550.42</td>
<td>0.0</td>
<td>50.0</td>
<td>50.0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>521763.0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3775.31</td>
<td>6.0</td>
<td>2376.43</td>
<td>2376.43</td>
<td>0.0</td>
<td>100.0</td>
<td>100.0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>852152.0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2376.43</td>
<td>5.0</td>
<td>1812.72</td>
<td>1812.72</td>
<td>0.0</td>
<td>50.0</td>
<td>50.0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>560362.0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3626.44</td>
<td>5.0</td>
<td>3350.65</td>
<td>3350.65</td>
<td>0.0</td>
<td>100.0</td>
<td>100.0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>5276.0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3575.46</td>
<td>5.0</td>
<td>1191.82</td>
<td>1191.82</td>
<td>0.0</td>
<td>100.0</td>
<td>100.0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>591963.0</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>4</td>
<td>3000.92</td>
<td>5.0</td>
<td>3000.92</td>
<td>3000.92</td>
<td>0.0</td>
<td>100.0</td>
<td>100.0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>564729.0</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1</td>
<td>2001.09</td>
<td>5.0</td>
<td>2001.09</td>
<td>2001.09</td>
<td>0.0</td>
<td>100.0</td>
<td>100.0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>303584.0</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>1</td>
<td>2001.09</td>
<td>5.0</td>
<td>2001.09</td>
<td>2001.09</td>
<td>0.0</td>
<td>100.0</td>
<td>100.0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>303584.0</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>2</td>
<td>2001.09</td>
<td>5.0</td>
<td>2001.09</td>
<td>2001.09</td>
<td>0.0</td>
<td>100.0</td>
<td>100.0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>303584.0</td>
</tr>
</tbody>
</table>

### Table 13. CALLCENTER Data, SCORE Data and the Target Variable in the CUSTOMER DATA MART

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Customer has any call center contact?</th>
<th>Number of call center contacts</th>
<th>Number of complaints</th>
<th>Percentage of complaints</th>
<th>Customer Value Segment</th>
<th>Last Value Segment</th>
<th>Change in Value Segment</th>
<th>Customer cancelled</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.2 SILBER</td>
<td>2 SILBER</td>
<td>1.00</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3 BRONZE</td>
<td>3 BRONZE</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3 BRONZE</td>
<td>3 BRONZE</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2 SILBER</td>
<td>2 SILBER</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.00 2 SILBER</td>
<td>2 SILBER</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3 BRONZE</td>
<td>4 LEAD</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>100.0 3 BRONZE</td>
<td>1 GOLD</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3 BRONZE</td>
<td>4 LEAD</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2 SILBER</td>
<td>2 SILBER</td>
<td>1.00</td>
<td>0</td>
</tr>
</tbody>
</table>

THE RESULTS AND THEIR USAGE

We created a data mart that can be used to predict which customers have a high probability to leave the company and to understand the reasons why customers cancel their service. We can perform univariate and multivariate descriptive analysis on the customer attributes. We can develop predictive modeling to determine the likelihood that future customers will cancel.

Notice that we only used data that we found in our five tables. We aggregated and transposed some of the data to a one-row-per-subject structure and joined these tables together. The derived variables that we defined give a good picture of the customer attributes and provide the best possible inference of customer behavior.

CONCLUSION

This paper shows that data preparation is a key success factor for analytical projects. Besides multiple-row-per-subject data marts or longitudinal data marts, the one-row-per-subject data mart is frequently used and is central for statistical and data mining analyses.
In many cases the available source data have a one-to-many relationship to the subject itself. Therefore data needs to be transposed or aggregated. The selection of the transposition and aggregation methods is primarily driven by business considerations of what will give meaningful derived variables.

This paper shows you ways to create a one-row-per-subject data mart from a conceptual and a coding point of view.

Anyone who is interested in more details on Data Preparation for Analytics is referred to my book with the same name, which will be published by SAS Press (Book# 60502) in September 2006 (see the “Recommended Reading” section). In this book the business point of view, the concepts, and coding examples for data preparation in analytics is extensively discussed.

RECOMMENDED READING
The author of this paper has written a SAS Press publication titled Data Preparation for Analytics. It is currently under review. The book provides a more detailed discussion on the topic of this paper. This book is written for SAS users and anyone involved in the data preparation process for analytics. The focus of the book is to give practical advice in the form of SAS coding tips and tricks, as well as conceptual background on data structures and considerations from the business point of view.

This book will enable the reader to do the following:

- view analytic data preparation in light of its business environment and consider the underlying business questions in data preparation
- learn about data models and relevant analytic data structures like the one-row-per-subject data mart, the multiple-row-per-subject data mart, and the longitudinal data mart
- use powerful SAS macros to change between the various data mart structures
- consider the specifics of predictive modeling for data mart creation
- learn how to create meaningful derived variables for all data mart types
- illustrate how to create a one-row-per-subject data mart from various data sources and data structures
- learn the concepts and considerations when preparing data for time series analysis
- illustrate how scoring can be done with various SAS procedures themselves and with SAS® Enterprise Miner™
- learn about the power of SAS for analytic data preparation and the data mart requirements of various SAS procedures
- benefit from many do’s and don’ts and practical examples for data mart building

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author:

Gerhard Svolba
SAS Austria
Mariahilfer Str. 116
Vienna, Austria
Email: Gerhard.Svolba@aut.sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.