Paper 63-28

Data Warehouse Administrator: Step by Step

Francesca Pierri, Università di Perugia – C.A.S.I., Perugia, Italy

ABSTRACT

The word "information" has become one of the most important in human life. During the last few decades much work has been done to collect and analyze data: every office has developed its own Information System, and is very proud to show its results to management. Today's problem is not how to get information, but rather how to relate each piece of information to the other. To do that, you need to read all the data residing in different databases, and to organize it from a different point of view: you need a Data Warehouse System. SAS/Warehouse Administrator is a tool that provides a visual environment to manage data warehouses. It helps you in the creation and maintenance of your warehouse, and is very handy to use. This tutorial will show what is needed to start a Data Warehouse Project, and the sequence to be followed. It will explain nomenclature that is very popular among data warehouse developers. A simple example is provided to better illustrate the theory. The intended audience should have beginning familiarity with the SAS Base product.

INTRODUCTION

A Data Warehouse Project is a big investment requiring time, open-mindedness and cooperation among people with different working experiences. When you start thinking about a data Warehouse, you are definitely convinced it is a good idea and will be useful, but not everybody thinks the same. Some people need to be convinced about that. Only when they are convinced will you become the warehouse architect that draws the project and then works closely with a specialized working group.

Cooperation is essential: obtaining other people's help is already a good result. For instance, if you want to build a house, you need to know where to find the land for it, or probably you know what you want to put inside the house, but do not know where to find it. In other words, even though you are a good architect, you are not necessarily a skilled bricklayer!

The most important thing in a Data Warehouse Project is the objective: what you are required to know, and how your results may be visualized. In order to do that, you must organize and plan your data requirements. Once the first step has been completed, you can start taking the second step: studying the data you actually have, and understanding where to find the information you need.

The following sections describe the sequences suggested for building a Data Warehouse.

DATA FLOW THROUGH A WAREHOUSE

Data sources might reside in databases, flat files, SAS datasets, and so on: SAS/Warehouse Administrator is able to read a wide range of data. All the information needed in the project will be extracted from data sources and stored in the Warehouse environment.

The data definition is saved in one or more ODD Groups, and inside each group there will be an ODD (Operational Data Definition) for each data table. The ODDs will be the input for your Warehouse.

You can conceive of your warehouse as an environment where you organize source data to answer questions that are different from those for which the data were collected. You will assign a name to your warehouse, and you will define one or more Subjects. To understand what a Subject is, think of the analysis result of your project and give that name to the Subject. Inside it, you will define a Staging Area or Data Group, where ODDs will populate Data Tables according to the architect's warehouse design. Basically, each Data Table represents a table where the input (ODDs) is organized according to a new logic, in order to populate, in the next step, what is called a Detail Table. The name itself explains the meaning: a Detail Table is a table where you have all the detailed data useful to the analysis you want to do; to create it, you can put together some of the Data Tables you have previously defined.

Once data stores (Data Tables and Detail Tables) are available, they can be used as input to summary data stores, and can be exploited with SAS software or other software products.

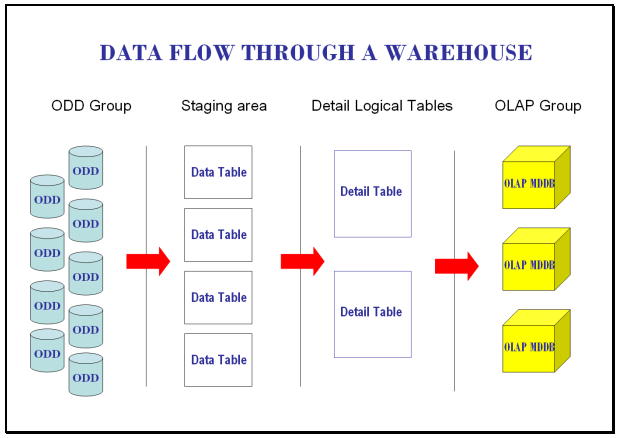

Figure 1

DATA WAREHOUSE ENVIRONMENT

A Data Warehouse project requires at least a Data Warehouse Environment. It is simply a metadata record that specifies the SAS library _MASTER, which is a repository.

Here the administrator will principally define

- all the libraries the project needs; for instance, a library for each kind of data: data sources, transformed data, detail data and summarized data.
- the host/hosts which are involved in the project
- DBMS connection profiles, which are the information needed to access data in a database management system other than SAS

OPERATIONAL DATA DEFINITION

Once you have defined your Data Warehouse objective, you will have surely looked at the required data, and you will define their location in the warehouse environment. To do that, you will add an Operational Data Definition Group (ODD Group), which is a simple grouping element for Operational Data Definitions (ODDs).

An ODD is a metadata record which provides access to data sources by

- registering the location of a SAS table or view,
- registering the location of a DBMS table with the help of a SAS/ACCESS LIBNAME definition
- executing user-written code that extracts information from a data source.

DATA WAREHOUSES AND SUBJECTS

A data warehousing project needs at least one Warehouse and one or more Subjects within the Warehouse Environment.

A Data Warehouse is a metadata record that specifies the SAS library _DWMD, which is a repository that mainly stores metadata for any data stores defined at the Warehouse level.

A Subject is a grouping element for data related to one topic within a Data Warehouse. Each Subject can group different kinds of data such as detail data, data tables or summarized data.

After adding a Data Warehouse and a Subject, you are ready to

define Data Groups and Detail Data stores in the Subject.

STAGING AREA

The Staging Area is where you organize the data sources previously defined in the ODD (Operational Data Definition), it is an intermediate data store.

In the Warehouse you look at the data from a different point of view. Often you can not use the ODDs as they are. You need to join source tables, create or rename variables, assign formats, and so on.

You can create a Data Group, a grouping element that allows you to organize a variety of data stores, which include Data Tables.

A Data Table is a metadata record that specifies a SAS Table or view. It can serve multiple purposes, and is frequently used to define an intermediate data store, such as a look-up table included as part of a join.

Once you have created a Data Group and assigned it a name, you are able to define the properties for a Data Table: variable name, type, length, description, format, physical storage and access location. Afterwards you associate a process to the Data Table through the Process Editor Job. Here you prepare the data to be loaded into the Data Table and define the steps to load it.

The source code can be generated by SAS/Warehouse Administrator, or it may be user written and saved into a catalog.

DETAIL DATA STORE

Detail data is information near to the fact level and is the natural input for a summary data store. Typically, detail data, or a Detail Table, is created by joining multiple Data Tables, in order to provide useful detail data for your project.

You can use a Detail Logical Table, a metadata record, as a grouping element for Detail Tables. You can create only one Detail Logical Table for each Subject.

Detail Tables are added to Detail Logical Tables: you define the table's properties, a metadata record, and then the process to define and load the table. SAS/Warehouse Administrator will, or the user may, write the code to do it.

SUMMARY DATA STORE

In a Data Warehouse Project the objective is to analyze the data. You work and organize source tables from ODDs to Detail Tables, because you know which kinds of analyses and reports or graphs you must be able to produce, and the user will have more quick access to the data if you summarize them.

SAS/Warehouse Administrator allows you to summarize and store data in OLAP (Online Analytical Processing) tables or OLAP MDDBs, and to organize them in OLAP Groups.

You can also store summarized data in Data Tables, writing the summarization code in the Load Step for the Data Table.

A DATA WAREHOUSE PROTOTYPE

The previous sections show what are the phases to follow in a Data Warehouse Project.

In the following sections you will see how you can develop a prototype with SAS/Warehouse Administrator, illustrated with a real prototype project, that of the University of Perugia.

The Data Warehouse objective is the study of the various kinds of workers absences during the time period and inside some classification variables such as the gender, the age, the qualification or economic position, the date of appointment, the department.

The month is the Data Warehouse granularity.

The data source is an Oracle database, the university workers' presences and absences database.

SAS/ACCESS to Oracle allows one to access Oracle tables as SAS views or tables, so that inexperienced Oracle users are able to read and understand Oracle table contents and structures.

Let's look at the sequence followed to develop the prototype.

First of all, define the metadata to access Oracle tables, and then create a staging area where you define the following Data Tables:

- Absences: join two ODDs and create new variables
- Person: join multiple ODDs, create new class variables (like age group)
- Time: a user table where you can define for each date the year, the semester, the quarter, the month.

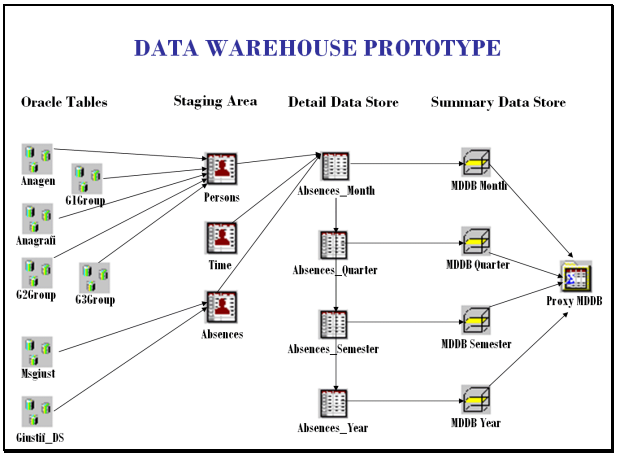

Figure 2

Now create a process for each Data Table. SAS/Warehouse Administrator software helps you to do it. If you like to add something else to the generated code, you can save it in a catalog, modify it, and tell SAS to use the user-written code instead of Warehouse-generated code. You can submit the code, test it, and view the result and log.

The next step is the Detail Tables definition. Do it with the Warehouse objective in mind: you will create a detail table joining the three Data Tables - Person, Time and Absences - and choosing the variables you need for the project. If you forget something, you can add it later. Do not include things you do not need, because they occupy space. The project detail time granularity is the month. So you create a Detail table for the month, one for the quarter, one for the semester and another for the year, all with the same structure. After that, you create a Process for each Detail Table.

The last step is the data summarization. Here has been created an OLAP Group that groups four OLAP MDDBs (Multi-Dimensional Data Bases), one for each time hierarchy (month, quarter, semester, year). In order to create an OLAP MDDB, you need SAS/MDDB software licensed on the machine where the OLAP MDDB will be stored. When you create an MDDB, you are required to define the class and analysis variables, the dimensions in the data, and the hierarchical relationships within the dimensions. Then you need to determine the crossing, which is a unique list of class variables that define a summarization level to be stored in one or more OLAP summary data stores. A crossing represents a grouping on which summary statistics are calculated. It is physically stored data, which provides the quick response when displaying a report in an OLAP.

There are several methods to determine crossing. Do not forget that an OLAP MDDB requires at least one crossing, in which all the class columns are included.

It is useful to read some theory books before undertaking an OLAP project.

A process is associated with each OLAP MDDB. As described earlier, SAS/Warehouse Administrator software helps you with generating the code.

After you create summary data stores, you can exploit them with tools designed to work with summary data.

STEP BY STEP

Open the SAS/Warehouse Administrator Desktop

Open SAS, type *dw* in the sas command window, and press the Return Key.

Add a new Environment

Put the cursor on an empty area on the desktop, click the right mouse button, select **Add Item** and then **Data Warehouse Environment**. The following window (Figure 3) requires you to write

at least the physical path where the Environment's metadata repository (_MASTER) will be stored.

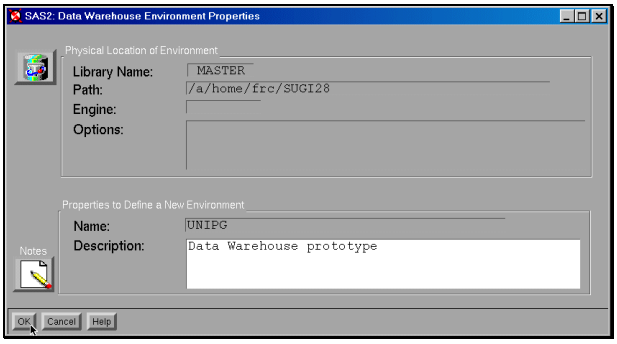

Figure 3

Then you will be prompted to enter a *Name* (UNIPG in the example) and a *Description*.
Click the OK button, and the new Environment is added to the SAS/Warehouse Administrator desktop (Figure 4).

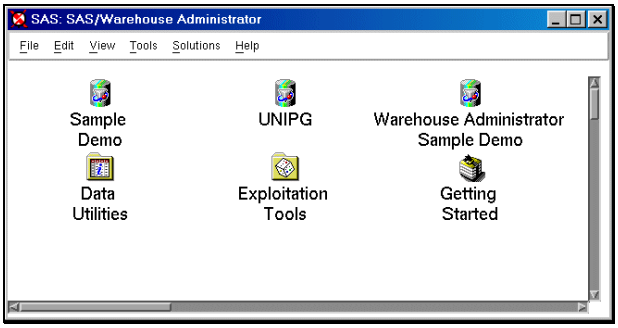

Figure 4

Make a double click on UNIPG to open the DW Environment

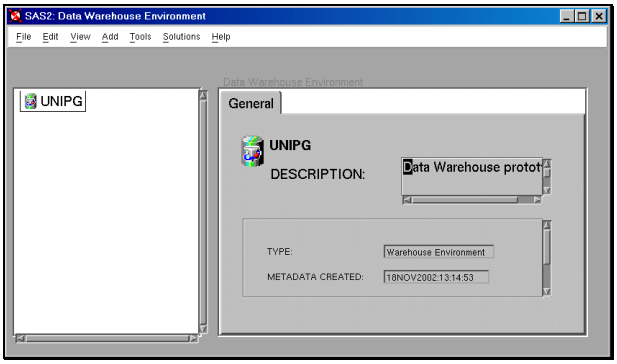

Figure 5

Define the Data Warehouse Environment
From the Menu bar select **File**, then **Setup** (Figure 6). The window in Figure 7 will appear. Here you can define your libraries, hosts and DMBS connections. If you click the button beside **SAS Libraries** and then the **Add** button at the bottom left of the window, a Properties window for the library displays for you to enter the appropriate information, such as the name of the library, the library reference (libref) and the path (Figure 8).
When you click the OK button (Figure 8), you come back and see the new library defined (Figure 9).
If you click the button besides **Host** and then the **Add** button at the bottom left of the window, a Properties window for the host displays for you to enter the appropriate information, such as the name of the host, whether it is local or remote, and the SAS version you are using (Figure 10-12).
If you click on the OK button, you come back to the previous window and see the host named previously defined added to the list.

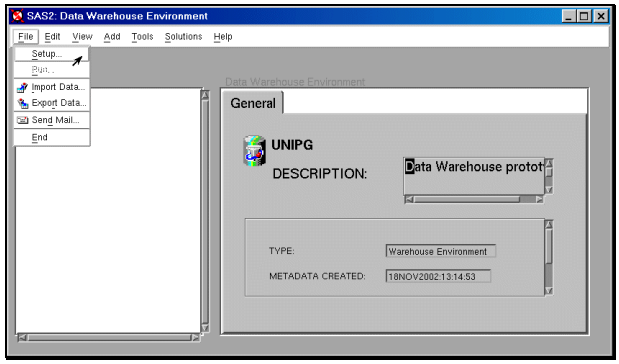

Figure 6

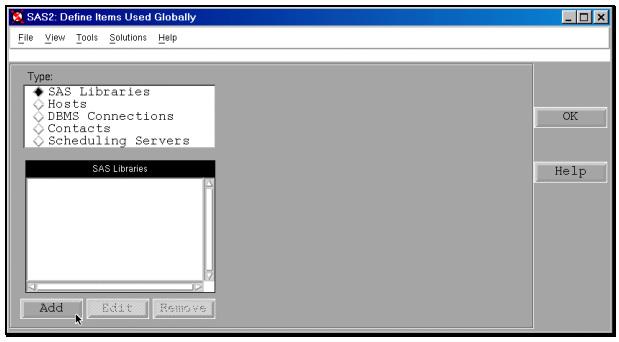

Figure 7

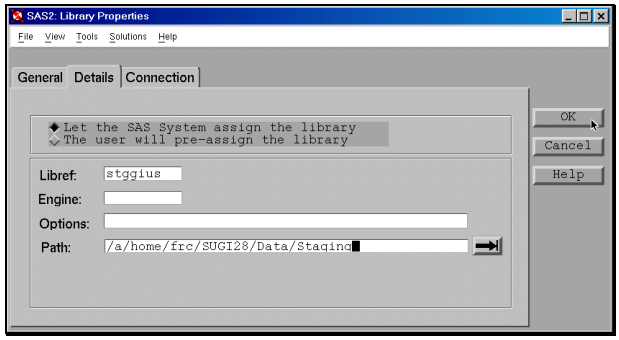

Figure 8

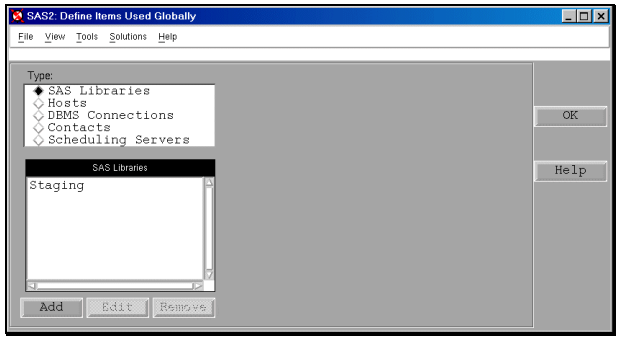

Figure 9

Now you can define the Oracle library: click on the button beside **DBMS Connections** and on the **Add** button. A Properties window is opened, and you can write the appropriate information, such as the name connection, the user schema and password (Figure 13-14).
When you click on the OK button, the new window displays the connection previously defined (Figure 15). If you click again on the OK button, you come back to the DW Environment.

Define the Data Warehouse
From the Data Warehouse Environment you can start defining your

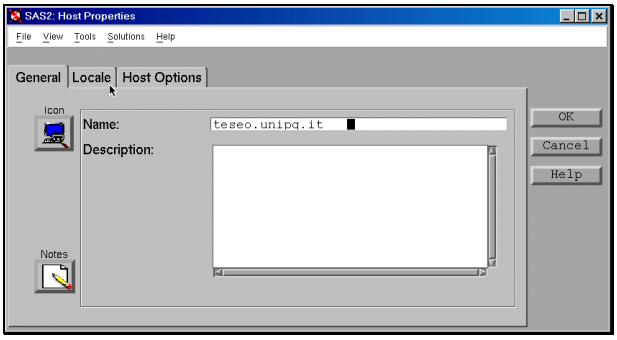

Figure 10

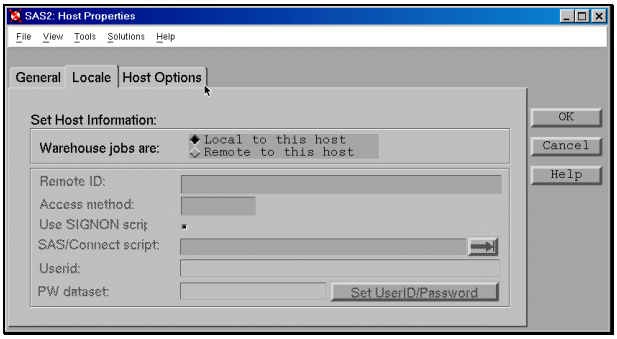

Figure 11

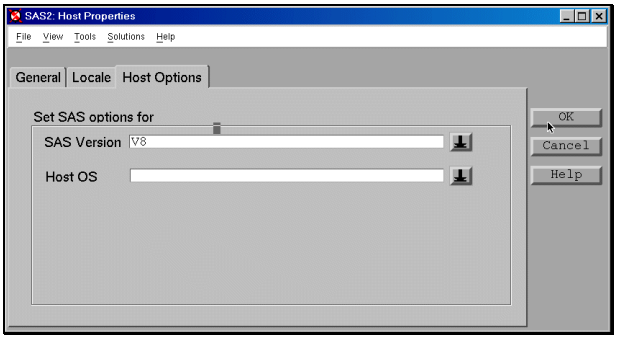

Figure 12

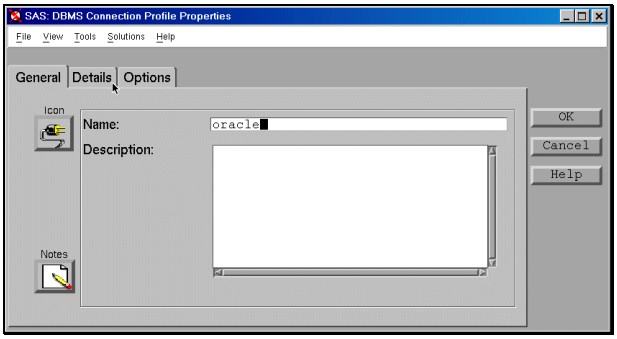

Figure 13

Warehouse. Put the cursor on the Environment, press the right mouse button, select **Add Item** and then **Data warehouse** (Figure 16).

A Properties window for the Warehouse displays for you to enter the DW name (Figure 17), and the location data for the metadata library _DWMD (Figure 18).

When you click on the OK button, you come back to the Explorer window, and you can see the data warehouse name (Figure 19).

Define an ODD Group

Put the cursor on the Environment, press the right mouse button, select **Add Item** and then **Operational Data Definition Group**

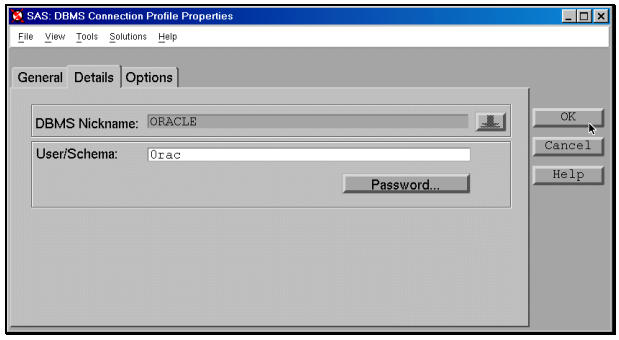

Figure 14

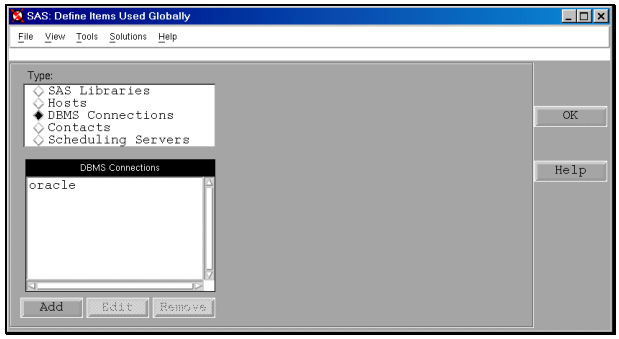

Figure 15

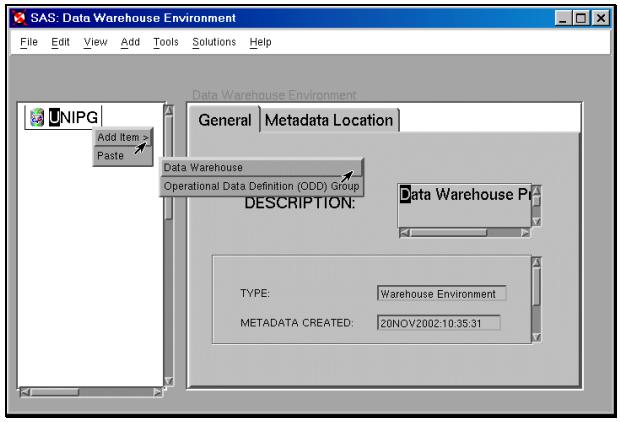

Figure 16

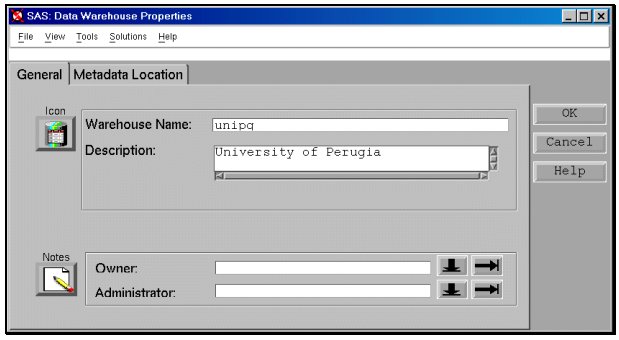

Figure 17

(Figure 20): in the Explorer window an ODD Group is added under the Environment. To define its properties click on it with the right mouse button and select **Properties**. Here you can define the ODD Group name, the owner and the administrator.

Create an ODD That Registers the Location of an Oracle table

Click again with the right mouse button on the ODD Group, select **Add item,** and then **ODD** (Figure 21). In the Explorer window a new ODD is added under the ODD Group previously defined. To update

the default metadata for the ODD, press the right mouse button, and select **Properties** (Figure 22).
A Properties window is opened and you can enter the appropriate information, such as the ODD name (Figure 23) , and its location (Figure 24).

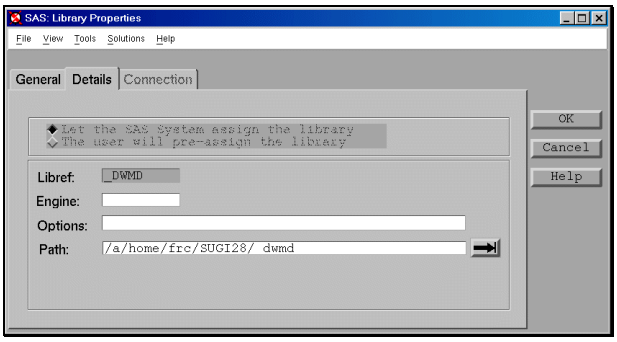

Figure 18

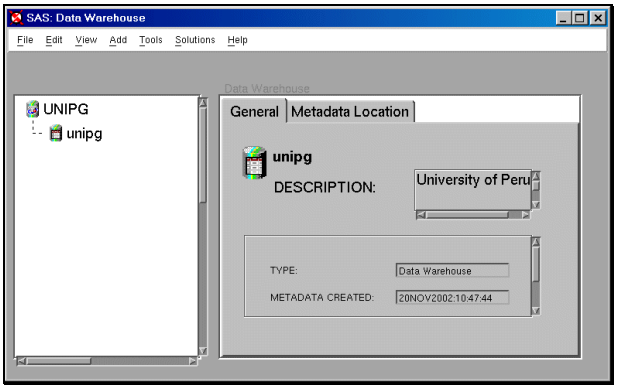

Figure 19

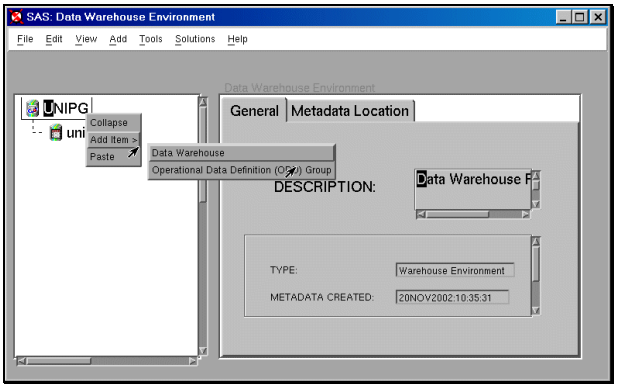

Figure 20

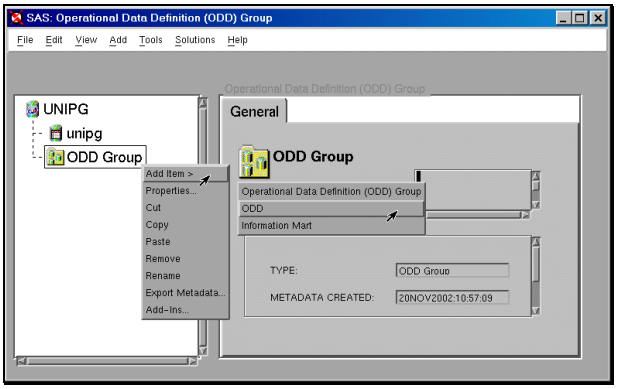

Figure 21

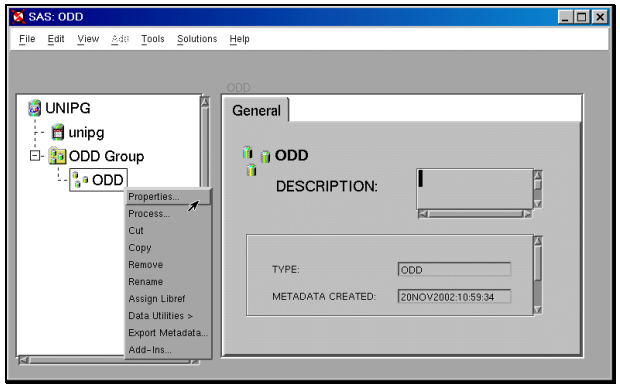

Figure 22

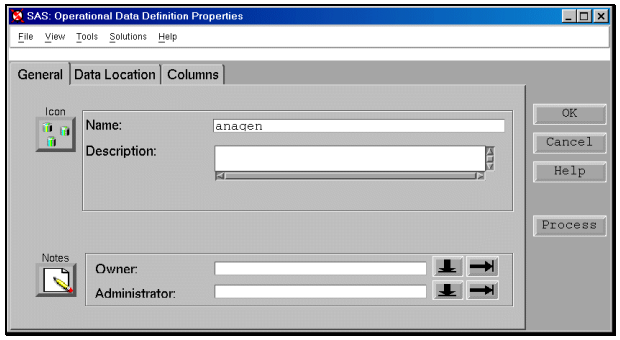

Figure 23

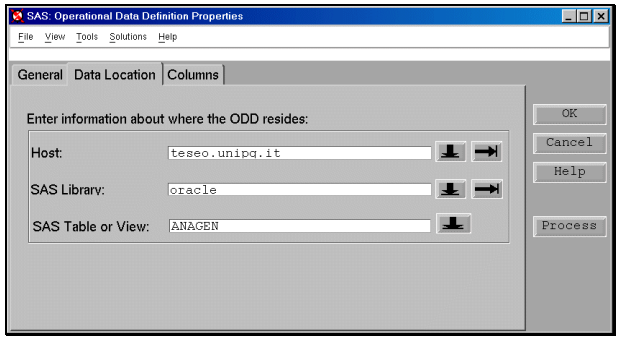

Figure 24

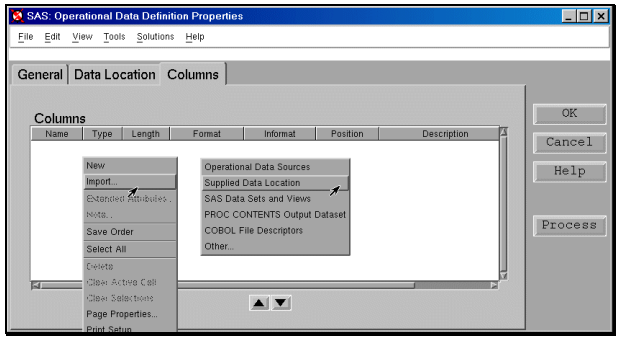

Figure 25

To import the Oracle table column names, click on **Columns**. A white window of properties is displayed. Press the right mouse button on the window area, select **Import** and then **Supplied Data Location** (Figure 25). All of the columns from the data source specified on the **Data Location** tab (Figure 24) are imported (Figure 26).
Click on the OK button, come back to the Explorer window and define the other ODDs you need for your project (Figure 27).

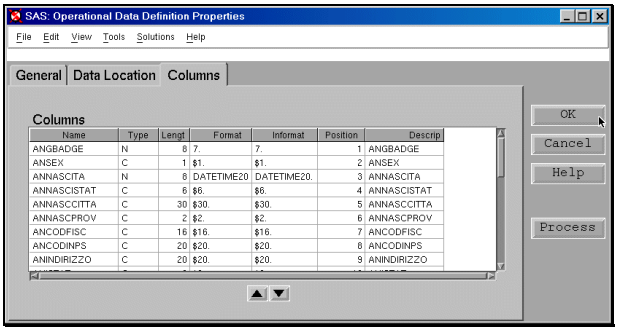

Figure 26

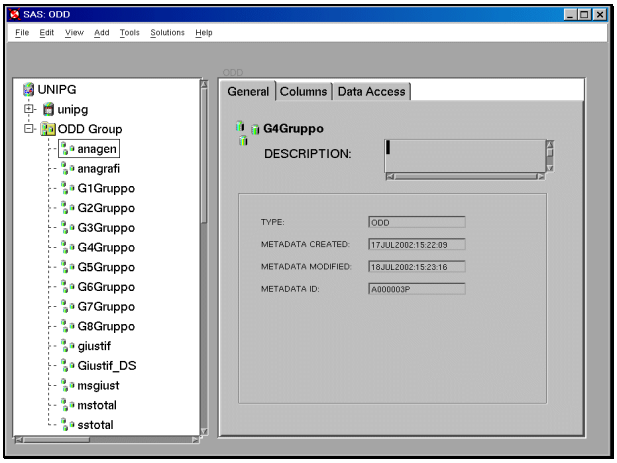

Figure 27

Define a Subject

In the Explorer window put the cursor on the Data Warehouse, press the right mouse button, select **Add Item**, then **Subject** (Figure 28).

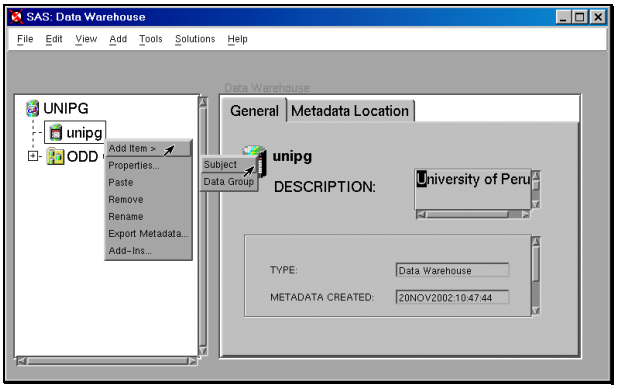

Figure 28

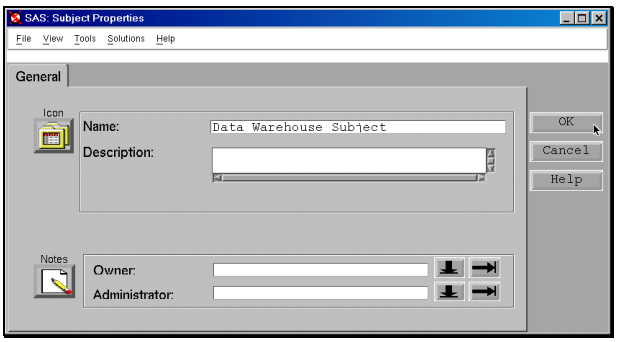

Figure 29

Update its default metadata: put the cursor on the Subject icon, press the right mouse button and select **Properties**. Here you can assign a name to your Subject (Figure 29).

Click on the OK button and come back to the Explorer window.

Create a Data Group

Put the cursor on the Subject, press the right mouse button, select **Add Item**, then **Data Group** (Figure 30). A new Data Group is added in the Explorer window, click on its icon, select **Properties**, assign a name (Figure 31), click on the OK button, and come back to the Explorer window (Figure 32).

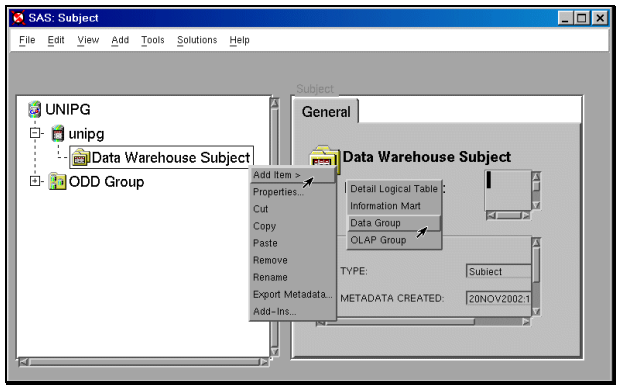

Figure 30

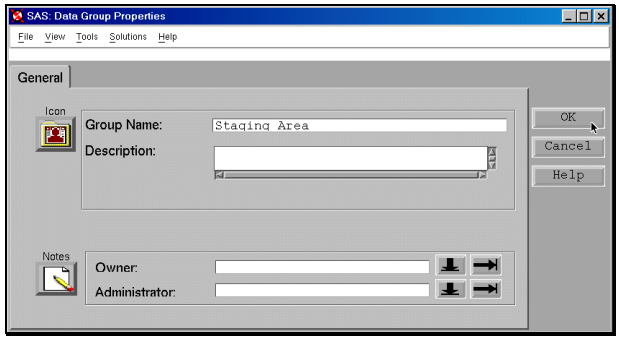

Figure 31

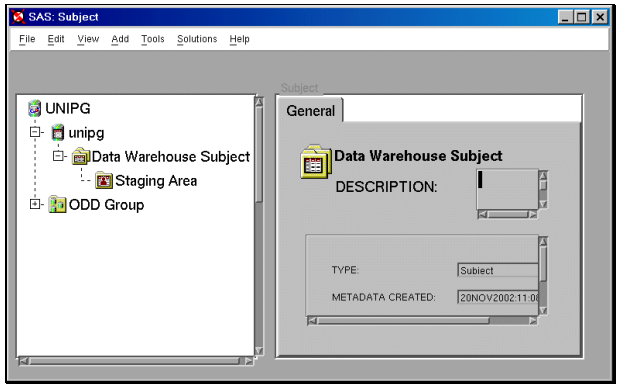

Figure 32

Create a Data Table

Once you have a Data Group, you can add tables. In the following the Absences table is created, and you can repeat the same sequence to create the other tables required.

Put the cursor on the Data Group icon (Staging Area), press the right mouse button, select **Add item**, then **Data Table** (Figure 33). In the Explorer window a new Data Table is added. Put the cursor on its icon, press the right mouse button, and select **Properties** (Figure 34). Define the Table name in the General Properties window, the Column names, length, type, format, …in the Columns window (Figure 35), the Physical storage attributes and the Access location (Figure 36).

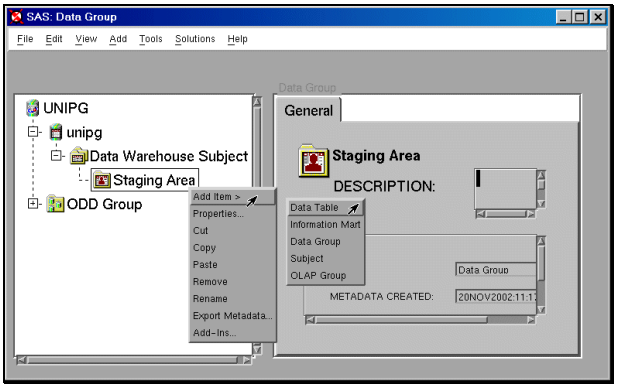

Figure 33

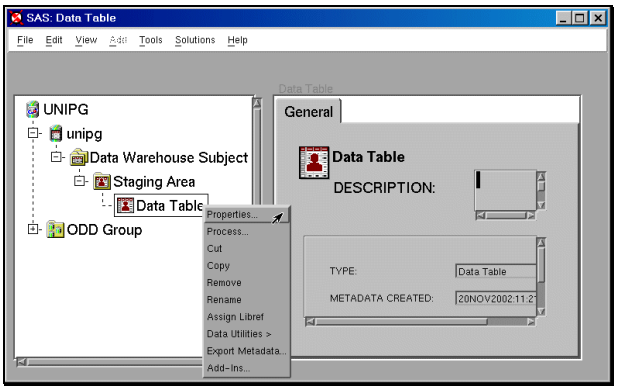

Figure 34

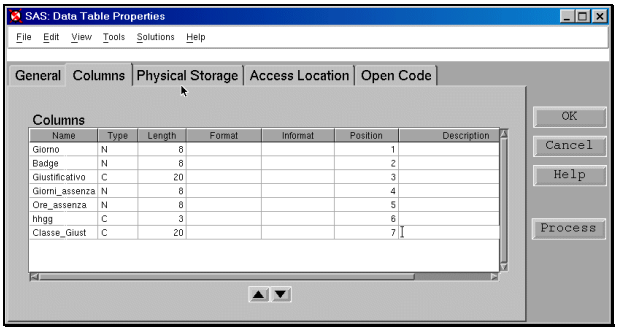

Figure 35

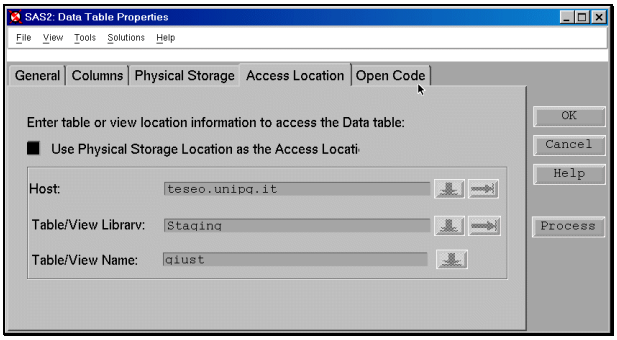

Figure 36

Define and Create a Job for the Absences Data Table

Once you have defined the Data table metadata, you can associate it with a process to create and load the table.

Put the cursor on the Data Table called **Absences**, press the right mouse button, and select **Process**. When SAS/Warehouse Administrator asks you if you want to create a job, select **Yes**. The software adds a job, opens the Process Editor window, and displays

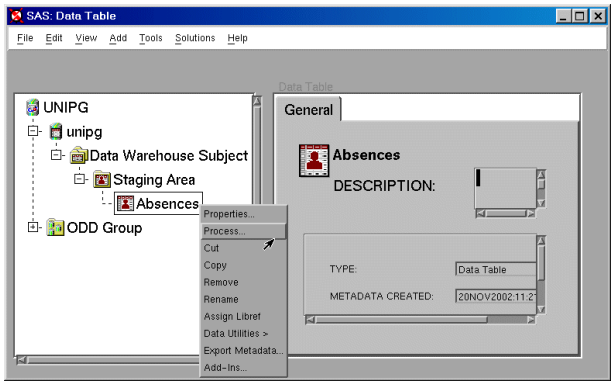

Figure 37

a job icon for the output table in the Process View (Absences). To add the input click the output table with the right mouse button, select **Add**, then **Inputs** (Figure 38). When the Selector window displays the Table type (Figure 39), select ODD, and then select the ODDs you want as inputs (msgiust, Giustif_DS), and click on the OK button.

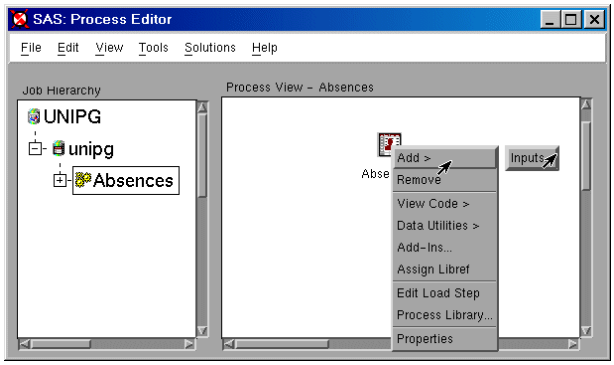

Figure 38

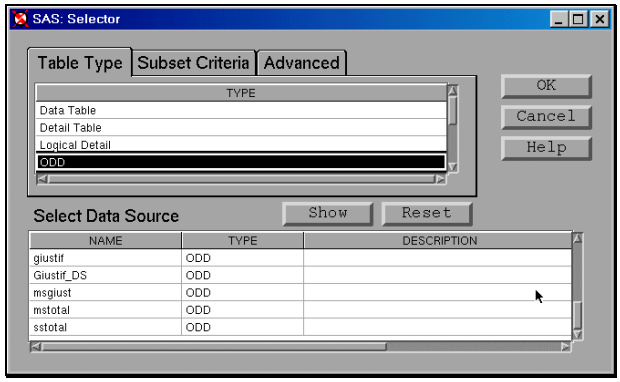

Figure 39

Now in the Process View (Figure 40) you see an output, two inputs, and a mapping process, which is a metadata record automatically added and used to generate or retrieve a routine that maps columns from the input to the output. To update the default metadata, put the cursor on the mapping icon, click the right mouse button, and select **Properties**. The Mapping Process Window displays for you to enter the appropriate information.

Create Detail Logical Tables

In the SAS/Warehouse Administrator Explorer put the cursor on the Subject, press the right mouse button, select **Add Item**, then **Detail Logical Table** (Figure 41). In the Explorer a Detail Logical Table is added, and you can modify its properties by clicking on its icon with the right mouse button and selecting **Properties**. When the Properties window is opened, you can assign a new name.

Now you are ready to define all the Detail Tables you need.

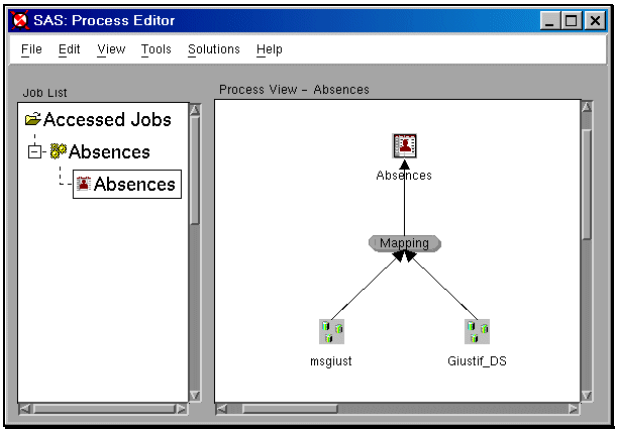

Figure 40

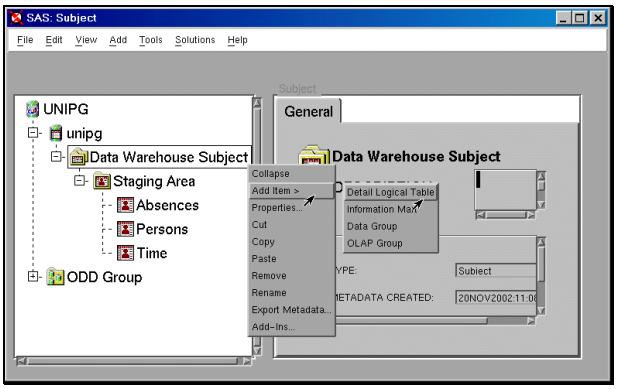

Figure 41

Create Detail Table

Put the cursor on the Detail Logical Table icon, press the right mouse button and select **Add New Table** (Figure 42).

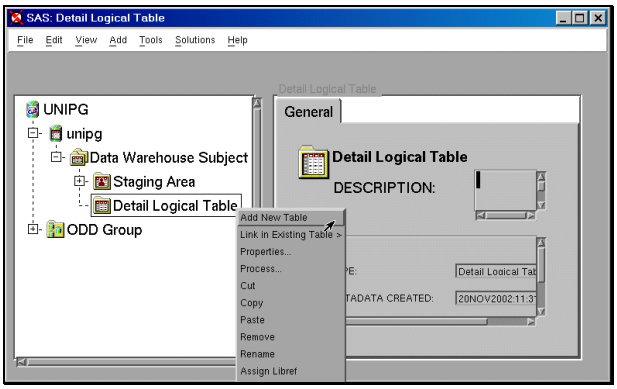

Figure 42

In the Explorer a new Detail Table is added, and you are ready to update the default metadata. Put the cursor on its icon, click the right mouse button, and select **Properties**(Figure 43).

A Detail Table Properties window (Figure 44) displays for you to enter the appropriate information, such as the name, the columns, the table allocation, and so on. You can follow the same steps described earlier to create a Data Table.

Define and Create a Job for a Detail Table

Once you have updated the metadata record for the Detail Table, you need to associate it with a process to create and load the Table. Put the cursor on the Detail Table icon, press the right mouse button, and select **Process**. When the SAS/Warehouse Administrator software asks you if you want to add a process, answer **Yes**, and a Process Editor window will be opened.

In the Process View you see your output icon, the Detail Table. Click

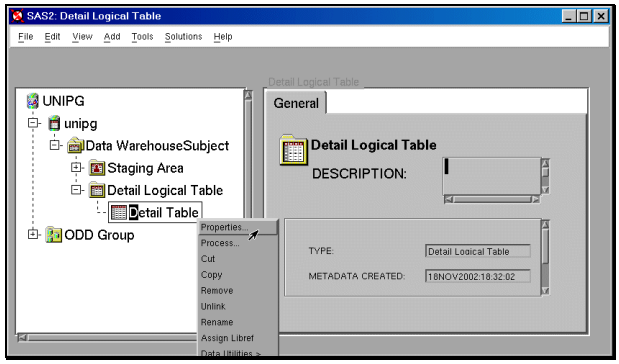

Figure 43

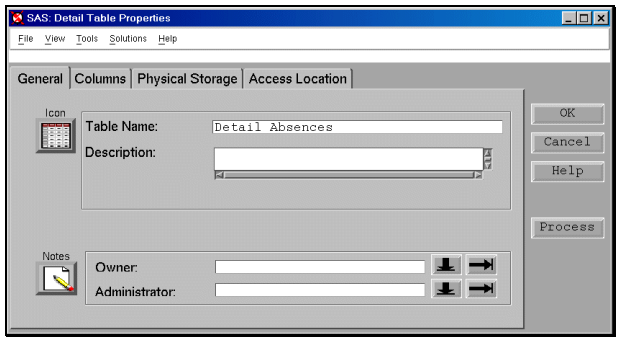

Figure 44

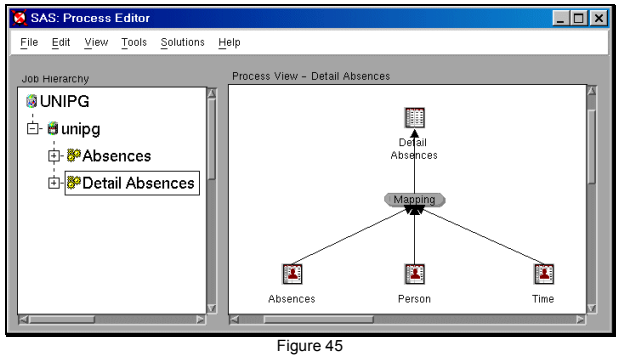

Figure 45

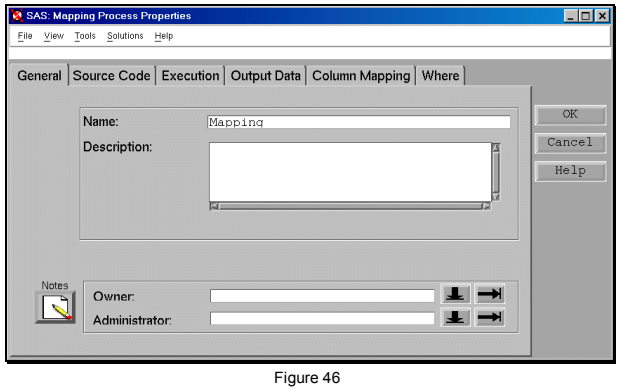

Figure 46

on it with the right mouse button, and select **Add** and then **Inputs**.

The Selector window is displayed, and here you can select the type of table you want to add as input source. In this case your input are Data Tables, so click on it, and the Data Tables list will be displayed. Click on your Tables input, and press the OK button.

In the Process View you see the inputs added to the output through a mapping process (Figure 45).

Click on the mapping icon with the right mouse button, and select **Properties.** When the Mapping Process window displays, enter the

appropriate information (Figure 46).

Create an OLAP Group

In the SAS/Warehouse Administrator Explorer, put the cursor on the Subject, press the right mouse button, select **Add Item**, then **OLAP Group** (Figure 47). In the Explorer window an OLAP Group is added under the Subject. To update the default metadata for the

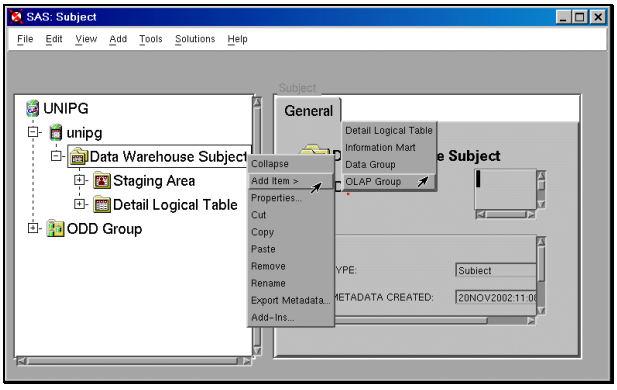

Figure 47

OLAP Group, put the cursor on its icon, press the right mouse button, and select **Properties**. The OLAP Group Properties window displays for you to enter the appropriate information, such as the group's name, and the group type (Figure 48). If you specify HOLAP as group type, SAS/Warehouse Administrator generates a *proxy MDDB*, which is a physical file that represents the structure of the data in an OLAP Group, and which will provide a more efficient access to the OLAP MDDBs.

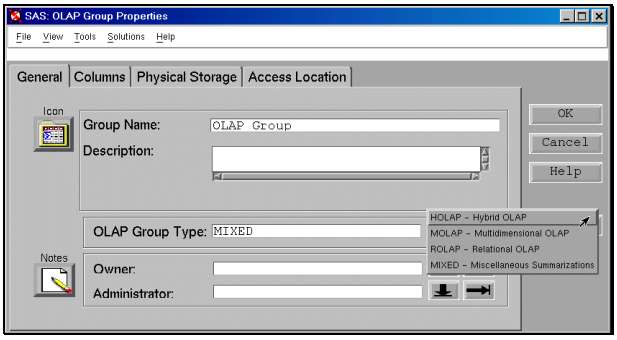

Figure 48

Define an OLAP MDDB

In the SAS Warehouse Administrator Explorer, put the cursor on the OLAP Group, press the right mouse button, select **Add Item**, then **OLAP MDDB** (Figure 49).

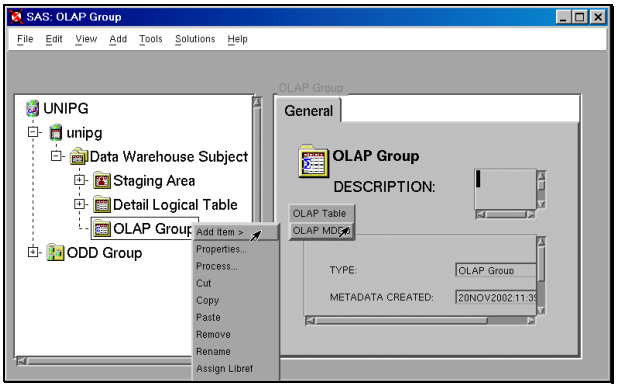

Figure 49

In the Explorer Window an OLAP MDDB is added under the OLAP Group. Put the cursor on its icon, press the right mouse button, and select **Properties**. The OLAP Properties window is opened, and you can enter the required information. Specify an OLAP MDDB name (*OLAP MDDB Absences*), specify the columns to be included in the OLAP MDDB, the storage format (MDDB), and where the OLAP MDDB is stored (Figure 50).

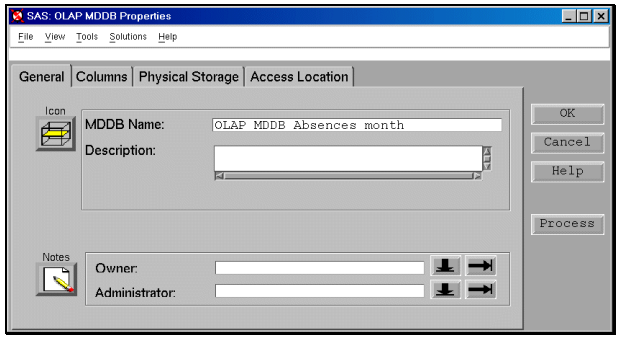

Figure 50

Define Class and Statistical Columns for the MDDB

Select **Columns** in the OLAP MDDB Properties, and a new window is opened. There are two sides: Columns and OLAP roles. Click with the right mouse button inside the Columns chart, select **Import** (Figure 51), and **Detail Tables**. When the Import Column Metadata window is opened, select the **Detail Table** used as input, and the columns required. Click the OK button, and come back to the OLAP MDDB Properties window.

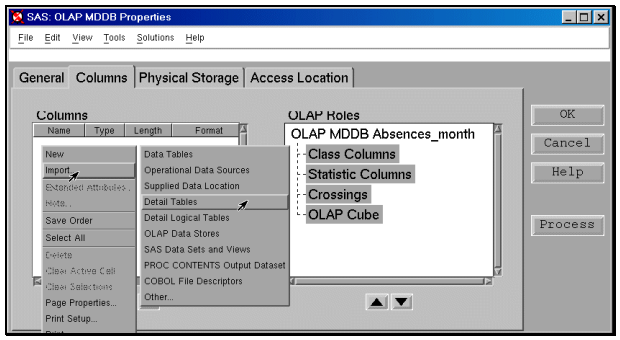

Figure 51

To assign class columns (Figure 52):

- select the columns from the list under **Columns**
- select the **Class Columns** label under **OLAP Roles**
- click the right arrow to add the columns to the summary role

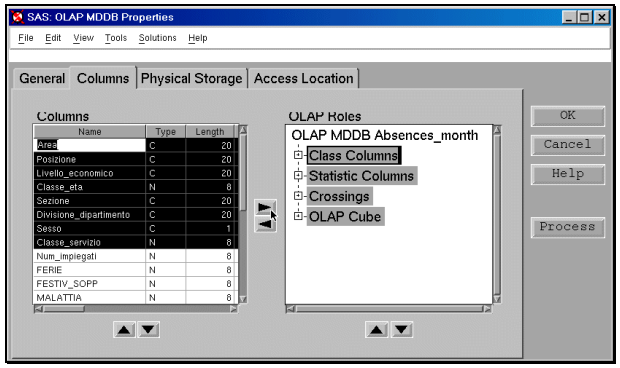

Figure 52

Do the same to assign the statistic columns. The default statistic assignment is SUM. In the example all the variables are class variables except the absences' hours and days that are the analysis variables assigned to the statistic columns.

Define Dimension and Hierarchies for the MDDB

Select the **OLAP Cube** label in the OLAP Roles, click the right mouse button, and select **New**: a new OLAP Cube is added.

Select the **Dimension** label in the **OLAP Roles**, click the right mouse button, and select **New** (Figure 53). You can specify a dimension name and description, clicking with the right mouse button on the Dimension object, and selecting **Properties**. To create the time dimension, you can call it Time and then define its hierarchies: year, semester, quarter, month.

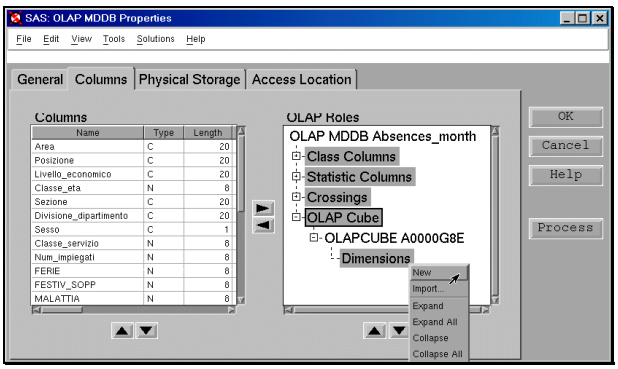

Figure 53

Select the Hierarchies label (Figure 54) associated with the appropriate dimension (**Time**), click the right mouse button, and select **New**. Click the right mouse button on the Hierarchy object created, and select **Properties**.

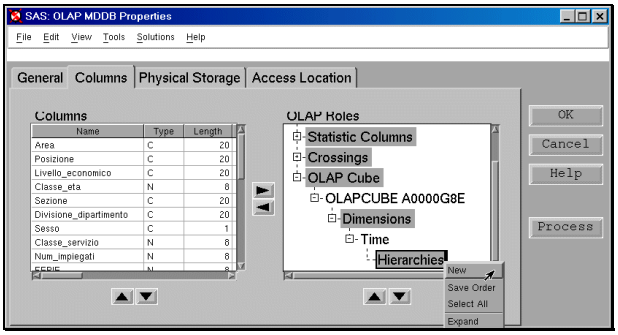

Figure 54

Specify the appropriate columns for the hierarchies:

- select the class columns from the list under Columns
- select the Columns label associated with the hierarchy object
- click the right arrow to add the column to the object

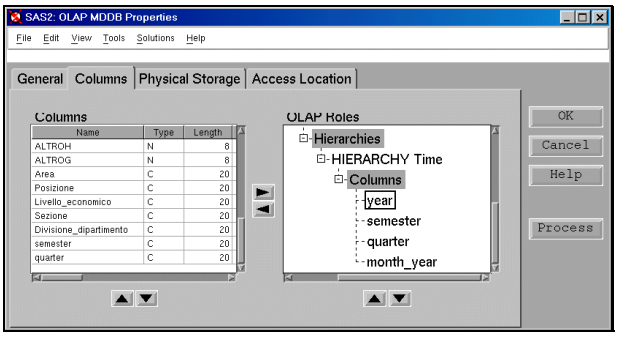

Figure 55

Define Crossing for the MDDB

Since a MOLAP or HOLAP application can access only the data referenced in a crossing, you are required to define at least a crossing including all your Class columns (NWAY crossing). Then you can define other crossings according to your application requirements (Stairstep crossing, Dimensional crossing,..).

To define an NWAY crossing, on the **Crossing** label (inside the OLAP Roles chart) press the right mouse button, and select **Create NWAY Crossing** (Figure 56): the crossing is automatically created.

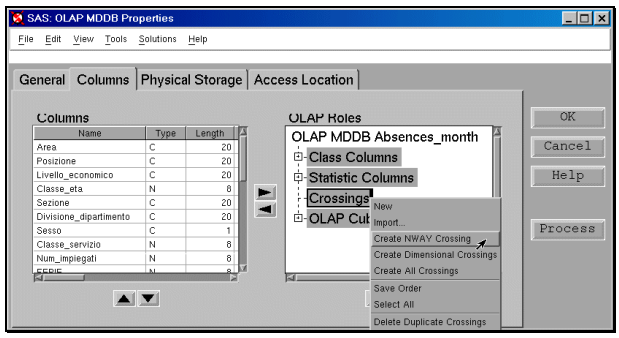

Figure 56

If you want to create a crossing to represent the most accessed class columns, click the Crossing label as earlier, and select **New**. You have created a new crossing, click on it with the right mouse button, select **Properties,** and here assign a name to the crossing and specify the class columns to include.

Click on the OK button, and come back to the Environment Window.

Create process for the OLAP MDDB

In the Environment Window click with the right mouse button on the OLAP MDDB, and select **Process**. As described above, you will define a Process (Figure 57) in the Process Editor window, specifying the input and the mapping properties, SAS/Warehouse administrator software generates the code.

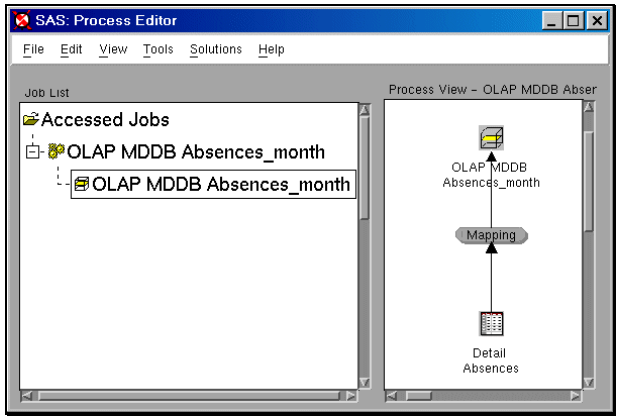

Figure 57

CONCLUSION

SAS/Warehouse Administrator is a friendly tool that helps you develop and maintain a Data Warehouse. You can work much better if you keep clearly in mind the phases to be followed. Thus, plan your warehouse, and then create it with the software tool. It is advisable to start with a small project, build a single Data Mart, become familiar with the software you are using, show the first results, and then refine the original plan and complete it.

CONTACT INFORMATION

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

Francesca Pierri
Università di Perugia, C.A.S.I.
Via G. Duranti, 1/A - S. Lucia Canetola
06125 Perugia, Italy
Work Phone: 001 39 075 585 3794
Fax: 001 39 075 585 3615
Email: frc@unipg.it