ANALYTIC PLATFORM

Quick Implementation Guide

Table of Contents

List of Procedures

Introduction

E.piphany, Inc. is a leading provider of intelligent customer interaction software for the Customer Economy. E.piphany E.5, an integrated suite of software solutions, blends web-based analytic and operational CRM to unify all inbound and outbound marketing, sales and service customer interactions. This enables a single, enterprise-wide view of each customer to help global businesses better understand and proactively serve customers in real time.

This chapter provides a broad introduction to the E.piphany Analytic Platform, describes the manuals and online help systems that comprise the Analytic Platform documentation suite, and summarizes the purpose and organization of this manual.

For a summary of features that are new for this release, late-breaking information about installation and upgrade, and information on fixed or outstanding product issues, see the *E.piphany Analytic Platform Release Notes*.

For information on supported platforms, see the Documentation section of the E.piphany Support Portal, http://support.epiphany.com.

E.piphany Analytic Platform

E.piphany Analytic Platform is based on a modern architecture that provides scalability to billions of records to handle large enterprise requirements. It delivers the high performance necessary to scale up to thousands of users while delivering fast, interactive response times.

The Analytic Platform's multi-tier architecture employs an ultra-thin web browser client, a middle-tier application server implemented in Java, and an underlying industry-standard relational database server. This enables it to integrate customer data from a wide variety of sources to provide a comprehensive view of customers and their behavior.

These sources include

- Touchpoint systems
- ERP/operational applications
- E-commerce web logs
- Legacy systems
- Data warehouses
- Data marts
- Third party data sources

The keystones of Analytic Platform architecture are the following major themes:

- **Integrated and comprehensive technology stack.** Analytic Platform applications are built on an integrated platform that provides an end-to-end technology stack. This stack provides the following capabilities:
  - Data integration
  - Data management
  - Advanced analysis (including OLAP, data mining and list analytics)
  - Campaign planning and design
  - Campaign execution

  The entire platform is seamlessly integrated via rich, common metadata, enabling unparalleled productivity and simplicity in implementation as well as use.

- **Easy to use, web-based architecture.**

  The Analytic Platform's thin web browser client is easy for business users to learn, easy for implementers to deploy, and scalable to large numbers of users.

- **Designed for rapid and incremental deployment.**

  The Analytic Platform is explicitly designed to enable rapid initial implementation and to quickly respond to changing business needs. Adaptive growth is a direct consequence of the investment in rich, common metadata that spans the system from data integration, to data mart construction and management, all the way up to the design of end-user interfaces.

- **Architected for high performance and enterprise scalability.**

  The Analytic Platform achieves high performance and massive scalability without the use of any proprietary or esoteric technologies. (These are typically hard to manage and often result in increased cost of ownership.) Instead, it runs entirely on industry standard technologies:

  - relational database management systems (RDBMS) from IBM, Microsoft and Oracle
  - Java
  - Microsoft and Netscape browsers
  - ODBC and JDBC data access technologies

The Analytic Platform is an integrated environment that includes a scalable data mart, an integrated metadata repository, an extensible data extraction infrastructure, powerful data transformation technologies, a scalable application server, high-performance analytic components that deliver ROLAP, data mining and list analysis capabilities, a complete campaign management system, an execution engine for e-mail marketing, and an engine for Real-Time.

The only external software components required are a relational database engine, a web server, and a web browser.

Localized versions of E.piphany Analytic Platform software support datamart applications in which the entire web-based interface appears in a supported local language. A localized datamart can contain data values in any one of the currently supported languages, locales, and code sets.

Analytic Platform Product Documentation

The E.piphany Analytic Platform product documentation includes give manuals and two online help systems. Each of these manuals is updated whenever needed. Most are updated for each maintenance release, but occasionally one or two will not need updating, and thus bear an earlier version number. If you are not sure whether or not you have all the correct documentation for the release your company purchased, please contact E.piphany Customer Support as described on page 7.

Quick Implementation Guide

The *E.piphany Analytic Platform Quick Implementation Guide* is designed to help beginning implementers get started with the E.piphany Analytic Platform, using the sample datamart shipped with the product. It includes:

- An overview of the product architecture.
- Information on datamart design and creation.
- An explanation of how extraction works.
- Step-by-step procedures for creating Rows and Columns web pages, reporting and analysis, lists, and campaigns.

Installation Guide

The *E.piphany Analytic Platform Installation Guide* is intended for database administrators who install, configure, and maintain the Analytic Platform.

It is designed to be used in conjunction with our product release notes, plus the appropriate installation and configuration manuals for your existing hardware and software.

Depending on the database server platform you select, these can be any of the following:

DB2 on AIX	*Quick Beginnings: DB2 for UNIX* and *DB2 Administration Guide*, Volumes 1 through 3, which can be viewed or downloaded for free from: http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v7pubs.d2w/en_main *Installation Guide* and the *Performance Management Guide* for AIX, which can be viewed or downloaded from: http://www.rs6000.ibm.com/cgi-bin/ds_form
Oracle	The appropriate installation guide for Oracle on your operating system The installation guide for your operating system if that operating system is not preinstalled The instructions for configuring kernel resources *Net8 Getting Started* *Oracle8i Reference*
SQL Server	*SQL Server Administration Guide* The appropriate administration guide for your Windows version
All platforms	The installation and configuration instructions for your RAID disk-management equipment and software.

NOTE: *Please follow the configuration recommendations that are suggested in this guide. If specific instructions do not appear for a particular configuration step or option, default values are acceptable.*

CONFIGURING APPLICATIONS

Configuring Applications is intended for implementers and Professional Services personnel who are charged with creating applications that will run on an AP datamart. The manual overviews AP functionality, architecture, and administration, and provides in-depth technical information on how to configure the AP topics required for Campaign Management and analysis.

DATAMART GUIDE

The AP Datamart Guide provides in-depth technical information on how to configure and populate the datamart used by all AP applications, including E.piphany Enterprise Insight applications.

VIEWING RELEASE NOTES AND MANUALS ONLINE

You can view our product documentation and Release Notes on any computer that has Acrobat Reader running.

When you install documentation on Windows, the installation wizard adds shortcuts to the Windows Start menu for easy viewing access. The documentation is installed in a directory that matches the current version number. (The default location of this directory is **C:\Program Files\E.piphany\Docs*version number***.)

For example, to view the installed documentation for Analytic Platform V. 5.0.6, navigate to:

Start > Epiphany > Analytic Platform Documentation > V. 5.0.6

Choose either Documentation or **Release Notes**.

To install documentation on a UNIX host:

1. Copy the **documentation.tar** file to a destination directory within a hard-disk partition.
2. Navigate to that directory and enter:

   ```
   tar xvf AP50Doc.tar
   ```

   This command installs a PDF file for each manual in the documentation set.

Contacting Customer Support

E.piphany Customer Support is available from 6:00 am U.S. Eastern Time to 6:00 pm U.S. Pacific Time, Monday through Friday (excluding U.S. holidays).

Voice:	866.376.9266 (toll-free)	throughout the Americas
	650.358.1700	
	+61.3.9017.8017	Asia/Pacific
	+44 (0) 118.963.7951	Europe, Middle East and Africa (EMEA)
Fax:	650.570.6290	
Email:	support@epiphany.com	

Contacting Technical Communication

The E.piphany Technical Communication group is always happy to get feedback from users on the completeness and usability of our product manuals. To contact Technical Communication, send e-mail to:

documentation@epiphany.com

You will receive a response in one business day.

Overview

This chapter provides an introduction to Analytic Platform product architecture and outlines the major steps involved in the implementation process.

Analytic Platform

Analytic Platform provides a highly customizable, reliable, and robust framework for delivering a variety of enterprise-wide, customer-focused applications to users throughout an organization.

Analytic Platform applications enable you to use a web-based interface to review, analyze, and act on data that is collected throughout an enterprise. The data that you choose to make available to users is extracted (copied) from a variety of source systems and placed in an EpiCenter datamart. The thin-client interface runs on standard web browsers, requires no applets or plug-ins, and operates in a variety of supported languages and locales. Analytic Platform applications provide you with the following capabilities:

- On-line analytical processing (OLAP)
- Data mining
- List management
- Campaign management

The applications are deployed through the Analytic Server, an application server that allows the applications to work together in tightly integrated fashion. Depending on the source systems from which data is extracted, Analytic Platform applications can be applied to such diverse areas as

- Sales reporting and analysis
- E-commerce management
- Supply-chain management
- Demographic analysis

Analytic Server treats individual application components as building blocks for creating sophisticated solutions to complex business problems. Navigation paths, called topics, link applications together in coordinated sequences and allow them to share data. E.piphany provides a number of prepackaged templates for topics, called topic masters, that you can customize to meet the exact needs of your business.

System Architecture

The main components of Analytic Platform include the EpiCenter datamart, source systems, Analytic Manager, Analytic Server, Epichannel, and related utilities. For information on each component hardware and software requirements and installation instructions, refer to the *Installation Guide*.

Analytic Platform Back End

The Analytic Platform back end consists of the EpiCenter datamart, source systems, Analytic Manager, EpiChannel, and related utilities.

EpiCenter datamart

The backbone of Analytic Platform is the **EpiCenter datamart**. The EpiCenter datamart contains data that is extracted from source databases. The data is stored in a format that is optimized for reporting and analysis. You can use the EpiCenter datamart to collect data from a wide variety of data sources, including any data source with which you can establish an Open Database Connectivity (ODBC) connection. Typical source systems from which you can extract data include:

- On-line transaction processing (OLTP) systems
- On-line demographic databases
- Web logs
- Web-site traffic-analysis tools

The EpiCenter datamart resides on recognized, industry-standard relational database servers. Refer to the *Installation Guide* for a list of supported database servers. The datamart also includes extensions for list management and campaign analysis.

EPICENTER

An **EpiCenter** is an EpiMeta database with its associated internal links to EpiMart, and EpiOp databases.

EPIMETA

The **EpiMeta** database contains metadata. The metadata defines the following:

- Schema for the datamart.
- Extraction jobs that periodically update the fact and dimension tables of the datamart.
- Semantic rules by which updated data is merged into datamart tables.
- Scheduling information for extraction jobs and other periodic tasks.
- Set of accelerator tables and indexes to maintain.
- Specifications for each of the measures, attributes, filters, and web pages that you configure.
- Topics by which web pages are linked.
- User and security information.
- Reports that users have saved.

EPIMART

The **EpiMart** database contains all the data that has been extracted from the source systems. Users can select from the data and presentation elements that a web page displays to construct queries that read and report on data that resides in this database. The EpiMart includes:

- Standard datamart fact and dimension tables.
- Accelerators that improve response times for user queries, including indexes, aggregate tables, and integer maps for character columns.

EPIOP

The **EpiOp** database contains

- Campaign history tables, which act as a secondary source system for campaign analysis.
- Lists, which include the results of previously executed queries.
- Extraction, Scheduler, and user query logs.

To facilitate building of EpiCenters, E.piphany provides the Analytic Manager. You use Analytic Manager to define the datamart structure, manage data extraction, and configure the front end applications.

ANALYTIC PLATFORM METADATA

All of the control information for the datamart is stored in a single metadata repository, the EpiMeta database. This database uses a relational model that includes numerous tables and referential integrity constraints. The use of a single metadata repository ensures that all components of the system receive notice of a change simultaneously.

Because the EpiMeta database defines the structure of the datamart and the Analytic Platform applications that operate on that data, Analytic Manager includes export/import commands that allow you to back up metadata, or to transfer subsets of metadata between datamarts.

In traditional client/server application environments, changes to the schema of a database table can adversely affect programs that operate on that table. The goal of the Analytic Platform adaptive architecture is to allow on-the-fly changes to the datamart schema while preserving the proper operation of Analytic Platform applications.

For example, the semantic instances that apply business rules to extracted data are SQL programs that must be changed in response to schema changes. References to tables and columns are parameterized within these instances so that EpiChannel can update them at execution with current schema information. Analytic Platform applications and other system components also refer to current metadata for information about the schema of the datamart, the configuration of applications, and the structure of topics.

Source systems

Source systems contain the raw data from which information is extracted and incorporated into the datamart. Source data can be extracted from any number of data sources.

Analytic Manager

Analytic Manager is a Windows NT or Windows 2000 graphical utility that enables you to build and maintain the EpiCenter datamart and the Analytic Platform applications that the datamart supports.

All development activities associated with building an Analytic Platform are done through Analytic Manager folders. The Analytic Manager uses a tree structure and provides the following folders with sub-branches under each folder.

TABLE 1: ANALYTIC MANAGER FOLDERS

FOLDER NAME	PURPOSE
Configuration	User configurable settings, transaction types, measure units, strings, and EpiMart state information
Schema	Define all elements of EpiCenter star schema
Extraction	Specify extraction steps, extraction jobs, and semantic transformations to build and update the datamart

TABLE 1: ANALYTIC MANAGER FOLDERS (CONTINUED)

FOLDER NAME	PURPOSE
Measures	Define measures and configure presentation objects that allow measures to appear in a web page
Presentation	Define attributes and attribute layouts, create filters, web pages, and topics (the user interface that is presented to the user)
Security/Storage	Define global security and storage options
Solutions	Pre-defined web presentations

EPICHANNEL

EpiChannel is Analytic Platform's fully integrated, graphical, and high performance extract-transform-load (ETL) technology. Epichannel extracts data from source databases, copies it into the datamart, and restructures it as your datamart specifications require.

EpiChannel can utilize either a single-phase or a two-phase ETL process. Single-phase "streaming" extractions can move bulk data directly from source systems into the datamart, without staging in intermediate tables. Alternatively, the two-phase process allows for more complex ETL operations that require comparison to existing values in the datamart. Phase one extracts data from source systems, performs record level transformations, and populates Analytic Platform staging tables. Phase two extracts data from the staging tables, and uses built-in metadata-driven semantic transformation templates to prepare the data into an analysis-ready form while intelligently loading or merging it into the datamart. This separation of source-specific data transformations from analysis-specific business logic transformations provides significant benefits such as minimizing the impact on operational source systems, easy integration with third party ETL and cleansing tools, and faster implementation.

The EpiChannel utility initiates extraction jobs that do the following:

- Pull data from individual source systems into staging tables.

- Apply semantic transformations to ensure that data is updated in a manner that is consistent with your business rules.
- Incorporate the data into your datamart.
- Build accelerators such as aggregate tables and indexes, which are used to improve the response time for reports, as well as integer maps and fact clusters, which are used to construct lists and campaigns.

EpiChannel also allows you to call third-party programs, such as data-cleansing tools and other utilities, as part of your extraction process. For more information on EpiChannel, refer to chapter 8, "Running Jobs with EpiChannel" of *Datamart Guide*.

SCHEDULER SERVICE

The Analytic Platform includes a built-in **Scheduler** that can be used to run extraction jobs, generate campaigns, and refresh reports on a scheduled basis. The Scheduler runs as a Windows NT or Windows 2000 service configured for automatic startup, on the same machine as the Analytic Server. You can also run the Scheduler from a command line, enabling integration with other applications or custom code. For more information on the Scheduler, refer to chapter 7, "Configuring Extraction Jobs" of *Datamart Guide*.

ANALYTIC PLATFORM FRONT END

The Analytic Platform front end consists of the Analytic Server, web server , web browser, and web-based applications (Analytic Platform application components). The front end enables users to perform sophisticated data analyses using a simple web-browser interface.

At the front end, each user opens a web page on any computer that supports a Java Script-enabled browser and then follows a sequence of preconfigured links to reach a desired application. For example, the user completes a form that specifies the contents of a report and then clicks a button to initiate a query. The type of query that is issued, and the analytical calculations that the Analytic Server performs on the results, depend on the application (web page) from which the query originates.

ANALYTIC SERVER

The **Analytic Server** is a robust middle-tier application server that supports Analytic Platform applications and connects the applications to the datamart. The server enables users to interact with the datamart through the web based user interface.

The Analytic Server is a Java application that manages user connections and database queries in a multi threaded fashion. It is implemented as a collection of Java classes that run inside a Java Virtual Machine (JVM). The JVM that runs the Analytic Server can be invoked as a Windows NT or Windows 2000 service, or directly as a jview command.

Analytic Server performs the following functions:

- Accepts the requests that each user enters through the web page for an application.
- Constructs an optimized query for each request that takes advantage of available accelerators, and cached results from previous queries.
- Forwards each query to the database server.
- Instantiates appropriate Java classes to perform application-specific calculations.
- Caches the results of each query.
- Formats the results of each query for web-based display.
- Forwards the formatted results to the web browser.

ANALYTIC PLATFORM APPLICATIONS

Analytic Platform applications run on any computer that provides Internet Explorer or Netscape Navigator (4.0 or higher browsers). These front end application components enable database query and analysis. The application components connect to the datamart through the Analytic Server.

Analytic Platform application components can be presented individually or as complete end user solutions. They are designed to be simple to use, with no training required. All application components are exposed through an ultra-thin web browser client - they use HTML and Java Script only, and do not require the use of slow-to-download and platform-dependent Java applets, Active-X applets, or plug-ins. This allows users anytime, anywhere access, with good performance over even low-bandwidth links including dialup analog connections.

A user interacts with Analytic Platform by means of web pages, which provide the user interface for an application. These web pages can be configured easily and managed by the administrator without requiring any knowledge of HTML or an HTML editor. The Analytic Platform includes a complete development environment for web site navigation. Web pages are grouped into related sets of activities using **topics**. A topic consists of a set of web pages and the links you create between those pages. Links allow you to create structured or unstructured navigation between web pages. You can guide users through the steps and tools appropriate to a specific problem. Web pages can be configured so that nothing is more than a click away. This makes Analytic Platform applications very intuitive to use by all types of users.

Analytic Platform supports the following web-based applications. These applications allow users to gather and analyze data, and take action based on that data.

TABLE 2: ANALYTIC PLATFORM APPLICATIONS

Alert	This application allows users to define alert messages and conditional triggers that activate display of those messages on an Executive Dashboard web page. Alert triggers can also control the display of other dashboard components.
Basic and Advanced Rows and Columns	This OLAP application displays selected data in tabular format and allows users to *drill down* to increasing degrees of specificity.
Bayes Classifier	This data-mining application creates statistical models that users can use to classify data.
Community Clusters	This data-mining application identifies groupings with common characteristics.
Cumulative Projections	This application projects cumulative results for a current time period based on previous time periods and current results so far.
Campaign Treatment Strategies	This application defines reusable treatments that can be applied across multiple campaigns.
Campaigns	This application creates lists and campaigns that are directed toward individuals or members of groups.
Dashboard	This application allows users to display summary charts and tables, conditional alerts, hypertext links, and plain-text messages. Users with wireless personal data assistant (PDA) devices can peruse the dashboard components that those devices support.
High/Low Clusters	This data-mining application identifies particularly high or low values.
Influences	This data-mining application identifies predictive relationships.
Lifecycles	This application projects the life cycle for a new product based on information about previous similar products.
Modeling	This data-mining application identifies predictive relationships for use in creating scored lists.

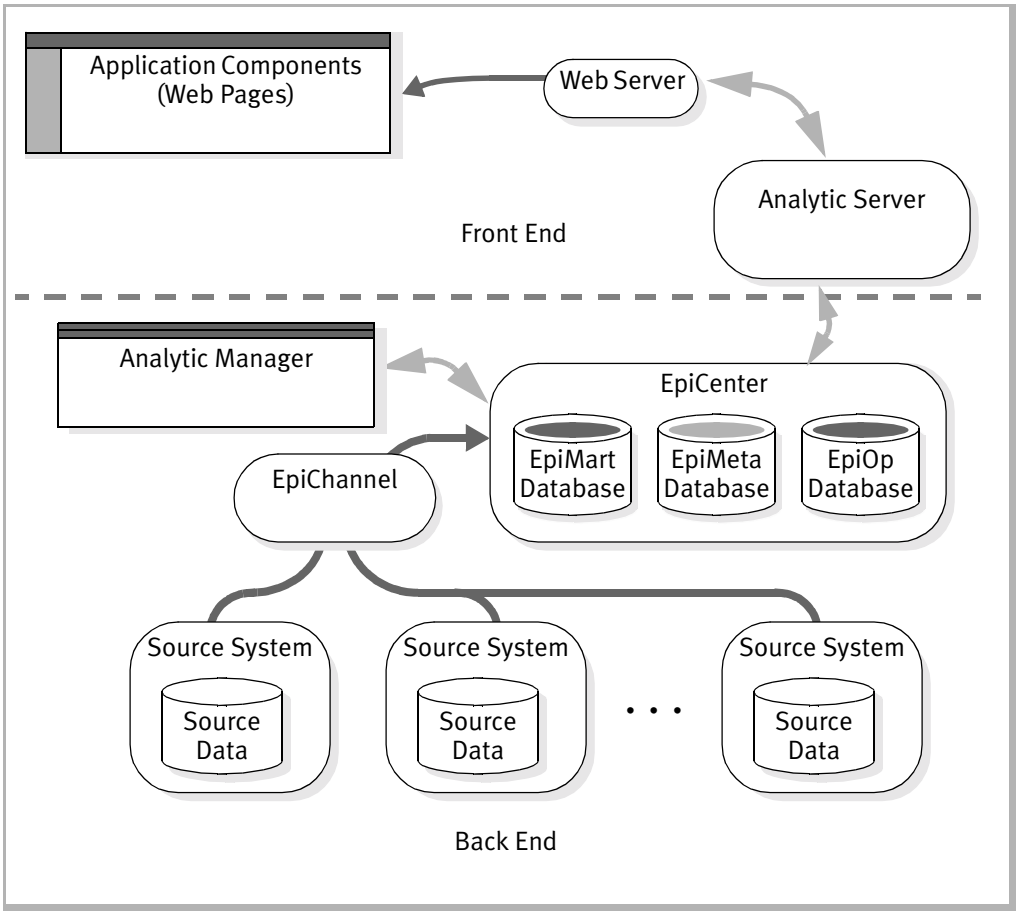
TABLE 2: ANALYTIC PLATFORM APPLICATIONS (CONTINUED)

Profiling	This OLAP application illustrates comparisons by selecting appropriate charts, such as pie charts, bar charts, histograms, and others.
Report Gallery	This application allows users to browse among saved reports, lists, and campaigns. Users can also store links to web pages that reside outside the Analytic Platform application.
Scoring	This data-mining application ranks list members according to a scoring model.
Trends	This application identifies trends and projects future values accordingly.

When a user interacts with the Analytic Platform through a web page and makes a request, the web browser passes the user's request to the web server, which in turn routes the request to the Analytic Server. The Analytic Server optimizes the query by selecting appropriate aggregate tables and indexes, then passes the optimized query to the EpiCenter datamart over a JDBC connection. When the datamart returns the result over that same connection, the Analytic Server instantiates the appropriate application-specific Java classes to perform analytical calculations, and then formats the finished results for presentation in HTML format. Analytic Server then forwards these formatted results to the requesting user's web browser for display.

Figure 1, on page 21 provides a high-level graphical representation of the Analytic Platform components.

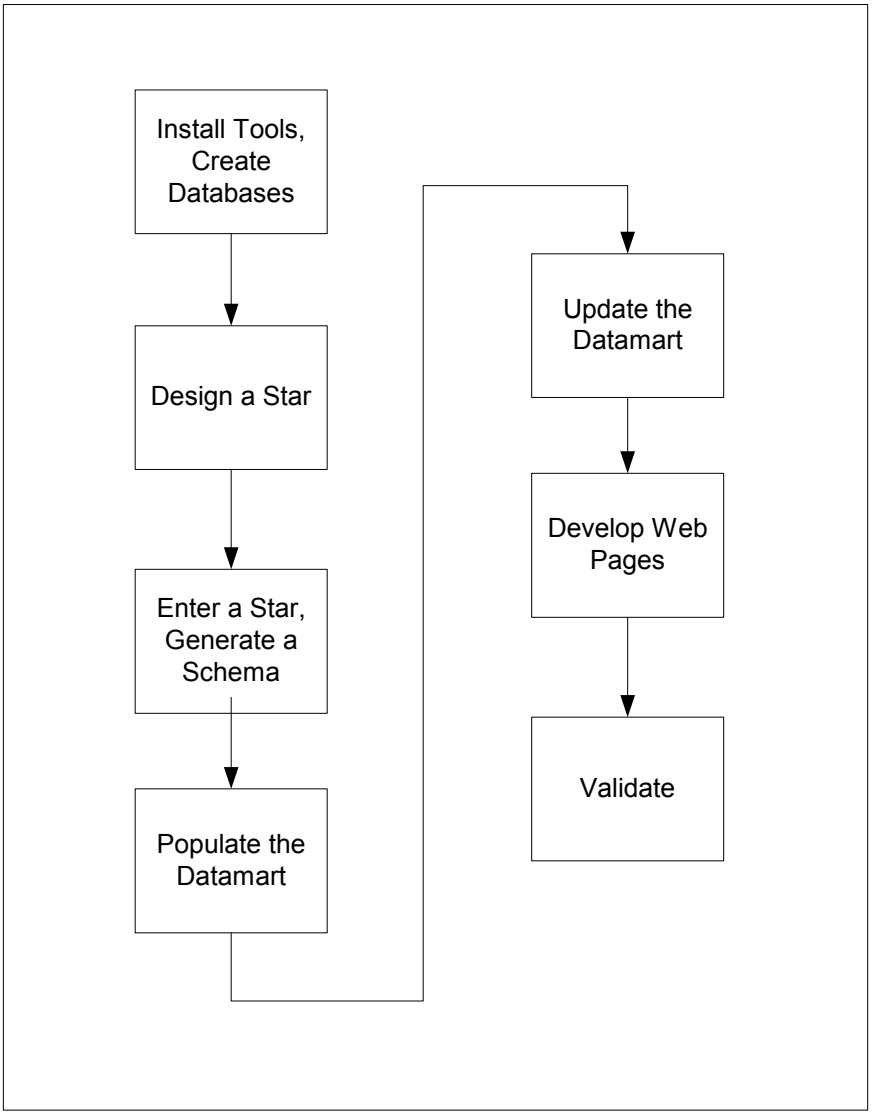
FIGURE 1: ANALYTIC PLATFORM ARCHITECTURE



ANALYTIC PLATFORM IMPLEMENTATION OVERVIEW

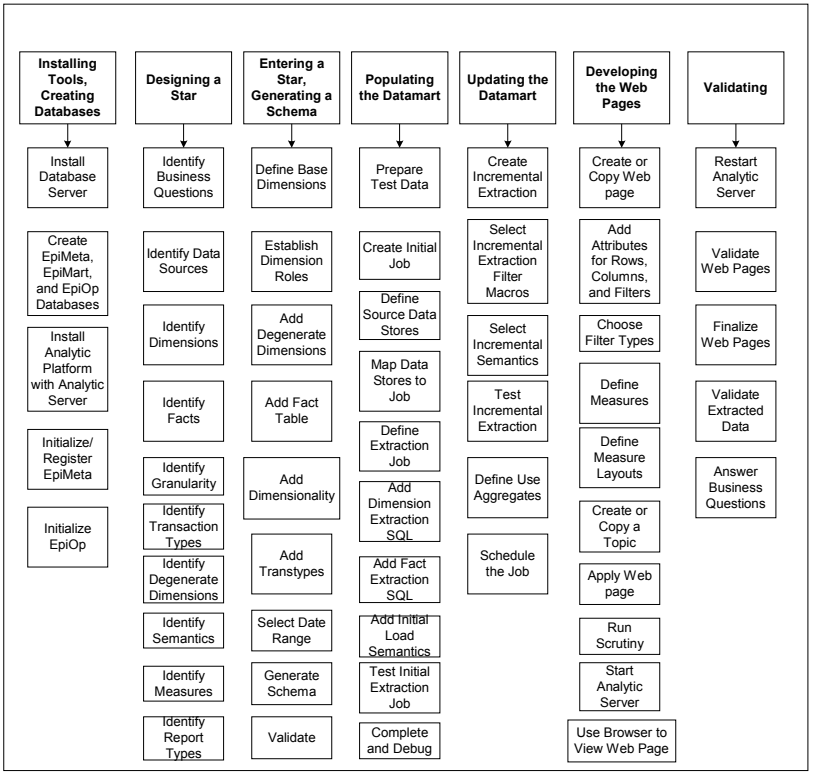
The major steps involved in an Analytic Platform implementation process are described in Figure 2, on page 22.

FIGURE 2: IMPLEMENTATION PROCESS OVERVIEW



The procedures that constitute each phase of the implementation process are described in Figure 3, on page 23.

FIGURE 3: IMPLEMENTATION PROCESS PROCEDURES



The procedures that you follow to implement the Analytic Platform are described in the following chapters:

- Chapter 2 provides an overview of datamart design, the EpiCenter database schema, building a star schema, and the sample datamart.
- Chapter 3 describes how to create databases for the EpiCenter, initialize the datamart, and generate a star schema. To create a datamart, you can use data from your own source systems or from a sample datamart. The procedures described in this book use examples from the sample datamart. The sample datamart is described in Appendix A, "Sample Datamart."
- Chapter 4 describes the extraction process (defining the data stores, defining the extraction job, and running the extraction job).

- Chapter 5 describes how to create a simple Rows and Columns web page to browse the datamart and display reports in tabular form.
- Chapter 6 describes how to create a full-featured Rows and Columns web page.
- Chapter 7 describes how to use a simple Rows and Columns web page to create other web pages such as Profiling, High/Low clusters, Trends, Lifecycles, Cumulative Projections, Influences, and Community Clusters. These web pages are commonly used for Reporting and Analysis.
- Chapter 8 describes how to create web pages for List and Campaign Management, update the datamart for list management, and complete a basic Campaign Management topic.
- Chapter 9 provides an overview of aggregates and describes how to define and build aggregates.

Designing a Datamart

This chapter provides an introduction to datamart design, the EpiCenter database schema, building a star schema, and the sample datamart.

Datamart Design

Designing a datamart involves determining the structure and scope of the datamart, specifying the level of detail to track, and developing some sample queries.

The structure of the business data you model is important because it provides the focus, direction, and purpose of your datamart. For end users, you model business data in terms of Facts and Dimensions. Facts are measurable events or quantities, such as the dollar amount of a transaction or a count of web-site hits. Dimensions are descriptive information about facts, such as the names, dates, and locations in which a transaction occurs.

You have to make decisions regarding the data you want to include in the datamart. This involves determining what kinds of questions users prefer to ask the datamart and what data is required to answer the questions.

You also have to make decisions regarding the level of detail you want to track (granularity). This involves answering questions about the level of tracking. For example, is it at the level of a state, a city, or a specific store within the city?

For focus and to aid in proof of concept, work with your users and develop a few sample queries. Such queries will help you define what the datamart model is able to report. It is also important to test whether the facts and dimensions that you have selected can answer the core business questions.

EPICENTER

An **EpiCenter** is an EpiMeta database with its associated internal links to EpiMart, and EpiOp databases. The EpiMeta database contains metadata tables. These tables store the schema for EpiMart tables and configuration data for the EpiCenter application component. The EpiMart tables contain the actual datamart tables (fact, dimension, and staging tables). It is where the extracted, organized data, such as customer, product, and order records is kept. The EpiOp database contains EpiCenter-generated data, such as campaign data, query logs, extraction logs, and saved lists.

To facilitate building of EpiCenters, E.piphany provides the Analytic Manager. Analytic Manager enables you to define the schema for the EpiMart tables, define the extraction process, set system security, configure web pages, and instantiate end user navigation paths for topics from pre-defined Analytic Platform solution templates.

EPICENTER DATABASE SCHEMA

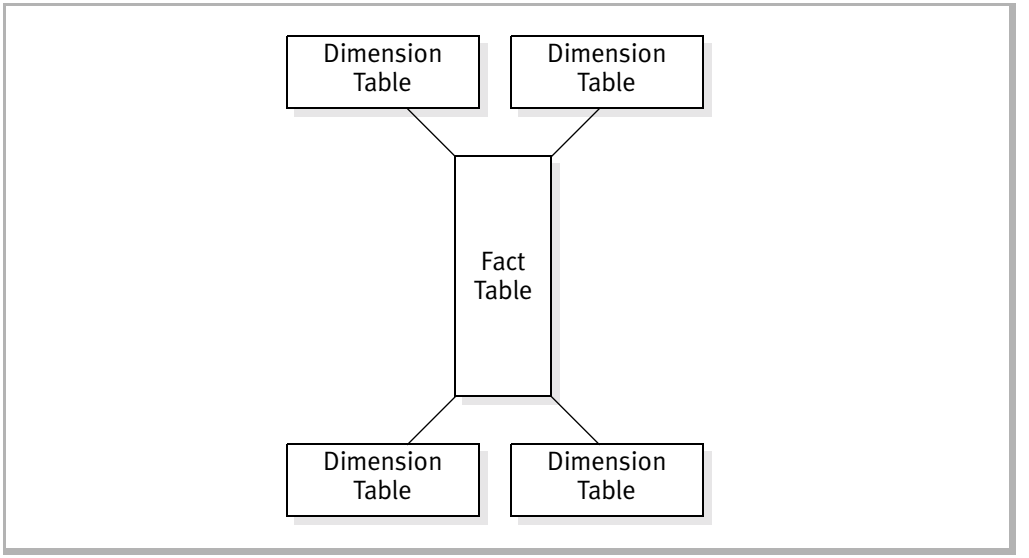
A **database schema** is a structure for organizing data into tables. The EpiCenter schema allows users to access information rapidly, in a form that allows them to understand and take appropriate action. The schema allows users to generate reports, charts, lists, and campaigns. Each datamart requires its own schema or table layout, which depends on the data that you choose to include.

The EpiCenter database schema is based on the dimensional data-warehouse model. A data warehouse transforms raw data from your organization's source system databases into a format that is optimized for query and analysis. Although the data in a source database is generally organized to facilitate the rapid processing of large numbers of individual transactions, the data in a data warehouse is organized so that it can be quickly sorted by any measure that the business uses.

For more information on the dimensional data-warehouse model, refer to *The Data Warehouse Toolkit*, by Ralph Kimball.

An EpiCenter datamart implements a dimensional data warehouse that is organized in the form of a **star schema**. A basic star schema appears as follows:

FIGURE 4: A BASIC STAR SCHEMA



At the center of a standard star schema is a fact table that contains quantitative data. Radiating outward from the fact table like the points of a star are a number of dimension tables.

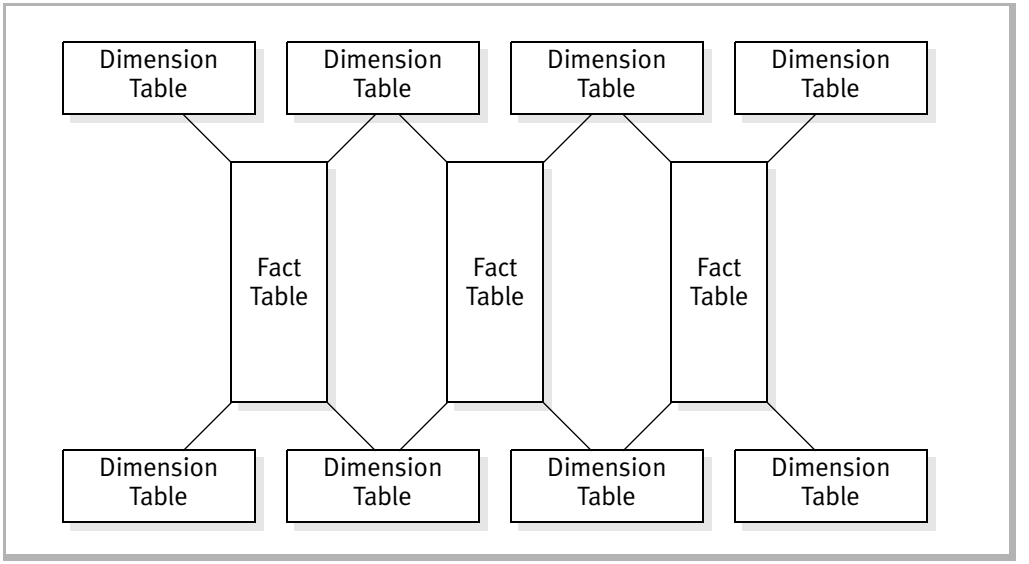
Dimension tables contain attribute data, such as the names of customers and territories. Each dimension table contains descriptive information about a particular aspect of a fact or facts. For example, one dimension table can contain information about the location of a store where a transaction takes place. That same store can account for hundreds or thousands of transactions. Another dimension table can contain information about customers, each of whom can have engaged in transactions at several different stores.

Fact tables are often "long and thin;" that is, they have many rows and few columns. Dimension tables, however, are often "short and fat," with few rows and many columns. The fact table is connected or joined to each of the dimension tables, but the dimension tables are connected only to the fact table.

This schema differs from that of many conventional relational databases, where many tables are interjoined.

An EpiCenter enables you to create a star schema with multiple fact tables, each of which is connected to a subset of the available dimensions. A sample star schema is as follows.

FIGURE 5: AN EPICENTER STAR SCHEMA

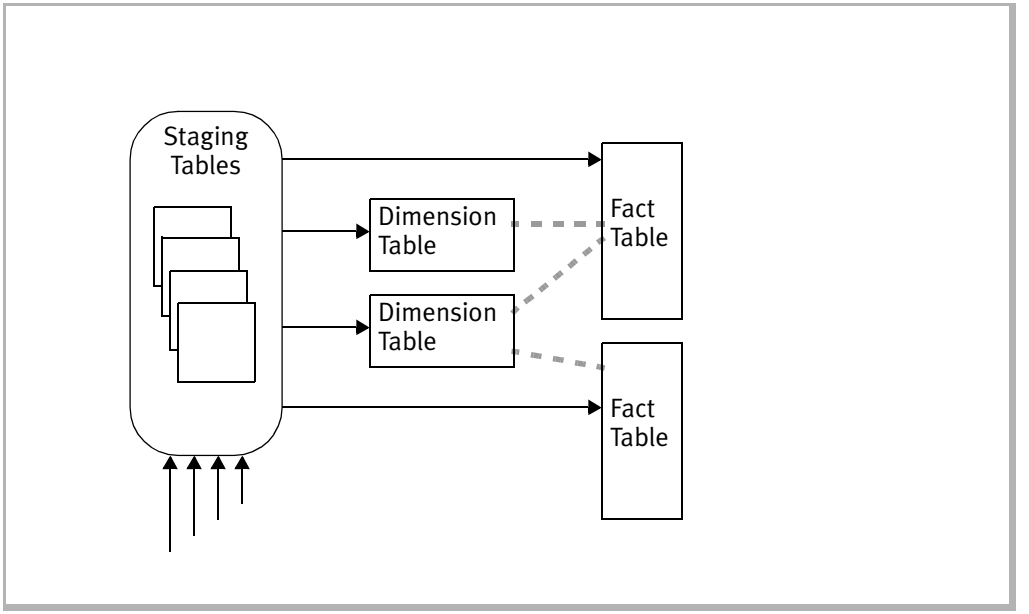


The facts in an EpiCenter datamart are organized into one or more fact tables. Dimension tables can be shared by multiple fact tables.

EpiMart Database

The **EpiMart database** includes fact, dimension, and staging tables. The fact and dimension tables contain the actual datamart data. **Staging tables** are the first entry point of raw data from the source systems into the EpiMart database. Each staging table serves as an interim stop for data that is to be stored in EpiMart database tables.

Figure 6: EpiMart Database Tables

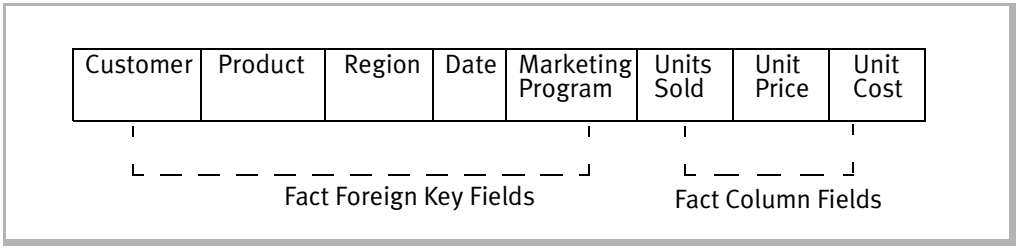


Fact tables contain records of discrete business facts, such as orders, shipments, or mailings. Each row in the fact table contains the actual fact values, as well as references (by means of foreign keys) to the dimension values that correspond to that fact.

For example, a fact in a sales table can be a customer's purchase of some quantity of a single item. Fact quantities are such things as number of units purchased and unit price. The dimension values corresponding to the fact are such things as customer, product purchased, and date of purchase.

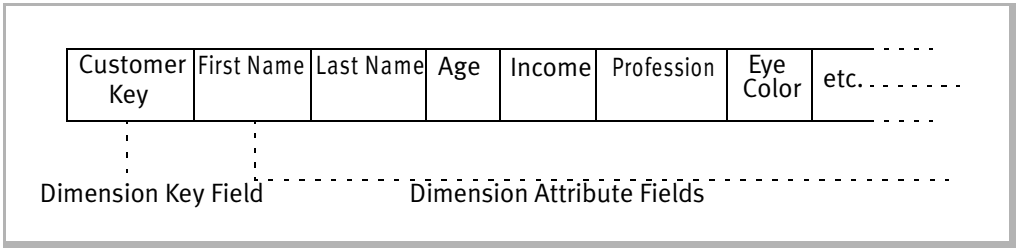
A typical fact-table row, with fields for the fact values and for the foreign keys to the dimension tables is as follows:

FIGURE 7: A FACT-TABLE ROW



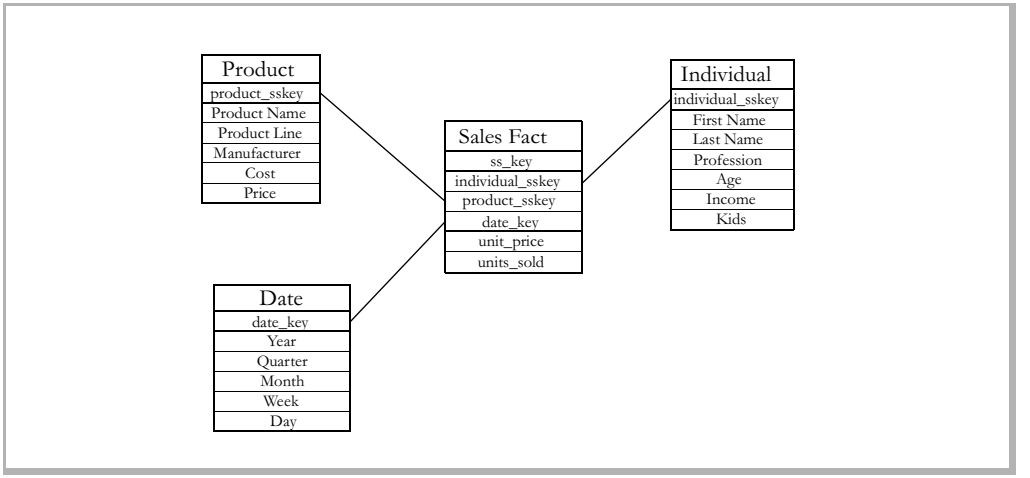
Dimensions are the aspects by which the data is available for analysis, such as customer or product. A dimension row contains a single dimension key, along with the attributes of that dimension element. The dimension key is the key that is used as a foreign key in the fact table. Dimensions often have large numbers of attributes. A typical dimension table row is as follows:

FIGURE 8: A DIMENSION-TABLE ROW



The fact and dimension tables discussed so far can be arranged in a star schema. In this star, you can perform calculations on fact-table quantities based on dimension-attribute values.

FIGURE 9: A SIMPLE STAR-SCHEMA



For example, you may be interested in the effect of age and income on the purchasing patterns of customers in the western region. You can then do a simple query that generates a table of unit purchases per customer (a calculation on facts) in the western region (a dimension-attribute value), with the results arranged by age and income groups (that is, arranged by dimension-attribute values).

BUILDING THE STAR SCHEMA

The star schema is the blueprint for building an EpiCenter datamart. It defines all of the fields in the dimension and fact tables. Before you initialize your datamart, you have to define the appropriate facts and dimensions for your datamart and sketch the star schema.

DEFINING DIMENSIONS

Dimensions provide the framework around which you perform measurements and analyses on the business data in the datamart. Dimensions typically include such things as time, transaction type, stores, vendors, customers, geography, accounts, and product.

Dimensions do not contain normalized data. In fact, it is often preferable to have denormalized data in the dimensions to speed up response time for user queries. Proper extraction ensures that any redundant data in a denormalized dimension remains consistent.

Time and Transaction Type are standard dimensions in EpiCenter datamarts. The Date dimension allows facts to be measured over various time periods. For example, you require a Date dimension to know which customers purchased a particular product during the month of February.

The Transtype (Transaction Type) dimension specifies a "slice" of a fact table. For example, the revenue generated from a customer purchase can be logged with a different transaction type from the refund generated from a customer return. Transtypes can be set up to handle all possible types of transactions within the business-data model.

DEFINING FACTS AND MEASURES

Facts are individual data points within the datamart. A useful rule of thumb for finding facts is to look at quantitative data. Facts are usually quantifiable things, such as "Dollar Amount Sold," "Units Sold," or "Customer Count."

Facts should be collected at a granular level. That is, a row in the fact table must refer to at most one element of any given dimension. For example, you can have a source database that records purchases by product category. You can want to record these purchases in a fact table that is connected to a product dimension. If your product dimension has a row for every specific product, then this source database will not give you data at a granular level.

Quantitative facts can be additive, nonadditive, or semiadditive:

- **Additive facts** can be added across every dimension. For example, in a fact table of sales transactions, it is possible to add up dollar sales by individual, product, date, and so on.
- **Nonadditive facts** cannot be added across any dimension. For example, if a sales fact table had a column for percentage discount, it would make no sense to add these percentages between transactions.
- **Semiadditive facts** can only be added across some dimensions. For example, a sales table can include a count of parking validations given out with a purchase. A single purchase can be split into multiple transaction records, one for each product purchased. This split makes the fact semiadditive.

  You can add these fact values along the product dimension, which allows you to find out how many parking validations were given out with purchases of some specific widget. However, you cannot add these fact values along the individual dimension. A single parking validation given to an individual can be recorded in multiple transaction records. If you were to try to add up all of the parking validations given to an individual, then any validation that appears in multiple transaction records would incorrectly be counted more than once.

Measures are computations performed on facts, and it is these computations that return the type of information that a user looks for in a report. These computations of measures have to be handled carefully in the case of nonadditive and semiadditive facts. Since additive facts are the easiest to manage, E.piphany recommends you arrange your fact table so that many facts are additive.

SAMPLE DATAMART DESIGN

The sample database contains sales and marketing information for a fictitious company. The fictitious corporate data is typical of the kind of data that you can have for a small company. The tables contain detailed information on customers, products, and manufacturers, as well as data about orders and sales calls.

For a list of the tables and columns in the sample database, see Appendix A, "Sample Datamart." Perform some SQL queries on the database to find out more about its structure and the data that it contains.

The sample database is provided as an Excel 5.0 file (**SampleSrc.xls**), in the **Resources/SampleSrc** subdirectory of the Analytic Platform installation image (CD-ROM or download). If you want to use the sample database as the source from which to create a sample datamart, you have to copy the file to your hard disk, and then load it into your SQL Server or Oracle8 database.

LOADING THE SAMPLE DATABASE IN SQL SERVER 7.0

1. In SQL Server Enterprise Manager, expand the directory tree to show the database server icon.
2. To create a database, expand the directory tree till you see the Databases folder.
3. Right-click the Databases folder and select New Database. The Database Properties dialog box appears. Make sure that Automatically Grow File is checked by default.
4. Enter the name **`sample_source_db`** and click OK to define the database.
5. Expand the Databases folder. The sample_source_db appears.
6. Right-click and from the All Tasks menu, select Import Data. SQL Server starts the DTS Import Wizard and the Import Wizard dialog box appears.
7. Click Next. The Choose a Data Source screen appears.
8. Select Microsoft Excel 5.0. as the source type.
9. In the File Name list box, enter the path of the sample database file that you just installed. If necessary, can click the "..." button and navigate directly to the file.
10. Click Next. The Choose a Destination Screen appears.
11. Make sure that your database server and the sample_source_db database are selected, and click Next. The Specify Table Copy or Query screen appears.
12. Select Copy Table(s) From the Source Database and click Next. The Select Source Tables screen appears.
13. Check the check boxes next to all tables whose names do not end with $.
14. Click Next. The Save, Schedule, or Replicate Package screen appears.

15. Check Run Immediately and uncheck all other check boxes.
16. Click Next. The Completing the DTS Wizard screen appears.
17. Click Finish to start loading the sample database. The Transferring Data progress dialog box appears. When the sample database is completely loaded, a confirmation message appears.
18. Click OK and click Done to exit the DTS Wizard.

NOTE: *If you encounter any problems with the above procedure, refer to SQL Server documentation. To load the sample database in Oracle, refer to your Oracle documentation.*

FACTS IN THE SAMPLE DATAMART

In the sample datamart, the candidates for facts are orders and marketing sales data. Orders and shipments (in both unit and dollar quantities) exhibit all of the qualities you expect from fact-table entries, so you can construct a fact table with rows for individual orders.

The sample datamart includes data for marketing sales calls and responses (in response_table). These are also quantifiable, but there is no clear way to include them in a fact table with the sales data. One possible solution is to build a separate EpiCenter for the marketing data. Building separate EpiCenters is not an entirely satisfactory solution because this arrangement makes it difficult to ask questions that involve both kinds of facts. A better solution is to create two fact tables, one for sales facts and one for marketing facts, within a single star. This multi-table solution is implemented in the sample datamart.

DIMENSIONS IN THE SAMPLE DATAMART

After you have specified your fact tables, decide how to arrange the remaining data of interest into dimension tables. E.piphany recommends that you not create a separate dimension for each source-database table. For example, the source data contains separate tables for product and product line. Since the product line is just an aspect of the specific product, include it as part of the product dimension.

Decisions about what data to combine into dimensions are dependent on the nature of your business and the types of queries that users will make. For example, the sample datamart includes manufacturer information in the product dimension. A different company can have dealings with a manufacturer in the roles of both supplier and customer. In such a situation, that company's users may want to make queries that combine customer and supplier data from these manufacturers. In that case, you can create a separate dimension for the manufacturer.

In addition to the product dimension, the sample datamart also contains dimensions for individuals, households, marketing programs, and sales representatives. Again, these decisions were made based on the nature of this fictitious business. A multi-level marketing company can, for example, have a single dimension that combined customers and sales representatives.

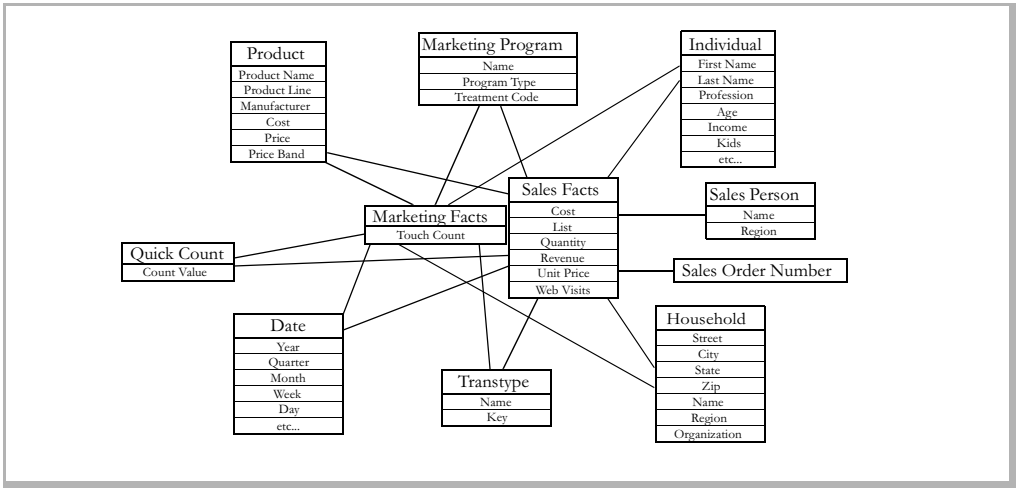
The sample datamart contains the Quick_Count dimension. This dimension enables you to do a query of something by "nothing" (that is, without breaking out a report by a second dimension). It is a dimension with a single value that is shared by every row in the fact table, so when a query is made by that value, every row in the fact table is returned. Such a dimension can be useful in Rows and Columns web pages, where a user can choose a specific attribute for the rows and nothing for the columns.

The sample datamart also contains a degenerate dimension called Sales Order Number. A degenerate dimension contains only a single column. Normally, a dimension is represented as an actual datamart table. A degenerate dimension, however, is represented as an additional text column in the fact table.

SAMPLE DATAMART STAR SCHEMA

The fact tables and dimensions discussed in relation to the sample datamart result in the following star schema.

FIGURE 10: THE SAMPLE DATAMART SCHEMA



The process of implementing the above star schema is described in Chapters 3-9.

Creating a Datamart

This chapter describes how to create EpiCenter databases, initialize the datamart, and generate a star schema. To create a simple datamart, you can use data from your own source systems or from the sample datamart described in Appendix A, "Sample Datamart." The procedures described in this book use examples from the sample datamart.

Creating EpiCenter Databases

The EpiCenter consists of three databases; one for the metadata (EpiMeta), one for the datamart data (EpiMart), and one for the operations data (EpiOp). E.piphany recommends that you give the databases names that end with meta, mart, and op (for example, sample_meta, sample_mart, and sample_op).

Note: *Before you create the databases, install your database server and Analytic Manager software. For installation instructions, refer to the Installation Guide.*

Creating Databases in SQL Server 7.0

1. Open SQL Server Enterprise Manager.

   Start>Programs>Microsoft SQL Server 7.0>Enterprise Manager

2. Expand the directory tree to show the database server icon.

3. Double click the database server icon to expand the directory tree. The Databases folder appears.
4. Right-click the Databases folder and select New Database. The Database Properties dialog box appears. Under File Properties, the Automatically Grow File is checked by default.
5. Enter the name **`sample_meta`** and click OK to define the database.
6. Expand the databases folder. The database icons appear in the right frame of the Console Root window.
7. Double-click the icon for your newly created database. The database_name Properties dialog box appears.
8. Check to make sure that the Automatically grow file box is checked.
9. Select the Transaction log tab. Make sure that the Automatically grow file box is checked.
10. Select the Options tab. Check the Select into with bulk copy, Truncate logs on checkpoint, Auto-create statistics, and Auto-update statistics check boxes.
11. Click OK.
12. Repeat steps 4through 11 to create two more databases and name them **`sample_mart`** and **`sample_op`**.

NOTE: *If you have a later version of SQL Server, refer to SQL Server documentation. To create databases in Oracle, refer to Oracle documentation.*

INITIALIZING THE DATAMART

After you create the databases, you initialize the datamart.

INITIALIZING THE DATAMART

1. Start Analytic Manager.

   Start\Programs\Epiphany*instance*\Analytic Manager.

NOTE: *Replace instance with the instance name that you entered when you installed Analytic Manager.*

2. Right-click the Servers folder and Select Register Server. The Server Properties dialog box appears.

FIGURE 11: THE SERVER PROPERTIES DIALOG BOX

3. Select the Server Type and enter the name of the database server.

4. Select an Authentication mode. Use the same mode that your database server is set up to use.
   - If you are using Windows NT Integrated Security, Select the Windows Authentication radio button.
   - If you are using SQL Server Authentication mode, Select the Use the following SQL Server login radio button, and specify the appropriate user name and password.
5. Click OK. If Analytic Manager displays the Select EpiCenter dialog box, click Cancel.

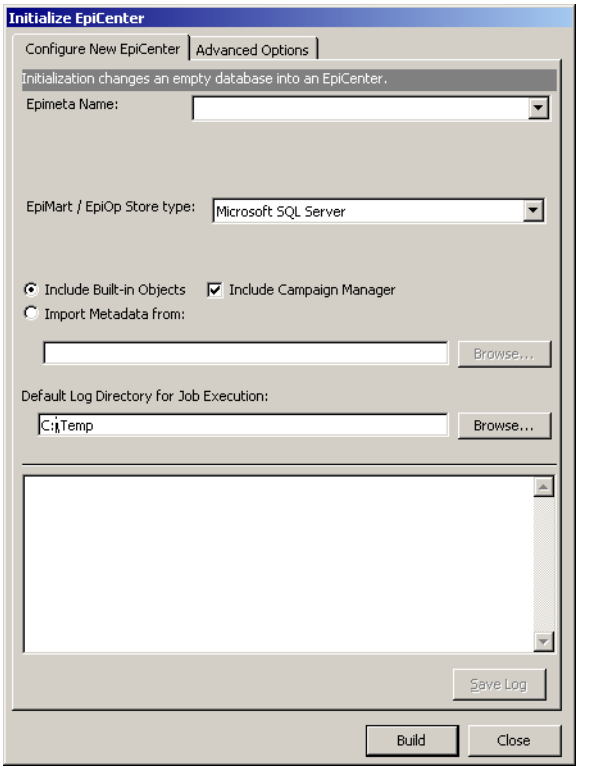
NOTE: *If you receive a message indicating that the server is already registered, click Cancel in the Server Properties dialog box. You do not have to re-register a server that has already been registered with an administrator account.*

When your database server is registered, Analytic Manager adds an icon with the name of the server, under the Servers folder.

6. Expand the Servers folder and select the new icon for your database server.

7. From the Menu bar, Select EpiCenter\Initialize EpiCenter. The Initialize EpiCenter dialog box appears.

FIGURE 12: INITIALIZE EPICENTER DIALOG BOX

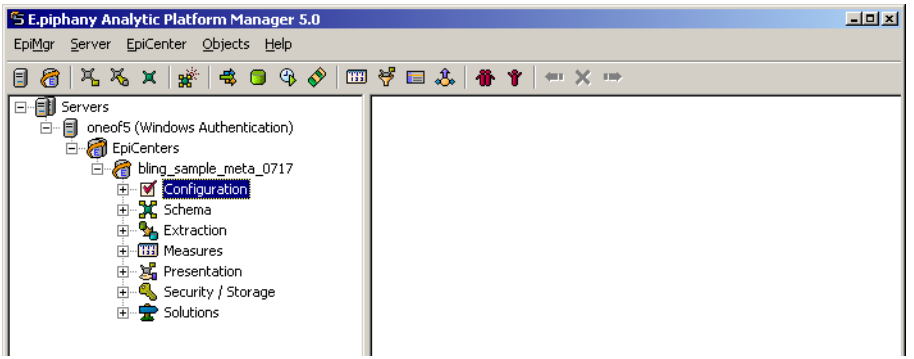


8. Select the appropriate database for your metadata from the Epimeta list box.
9. Select Include Build-in Objects and check Include Campaign Manager.
10. Select a default log directory. Use a directory different from the default of C:\TEMP.
11. Click Build.

    Analytic Manager begins the process of initializing the EpiCenter. This takes several minutes. When this process is complete, a confirmation message appears.

12. Click Close. The new icons appear in Analytic Manager main window. If you do not see these new icons, expand the folder that is named for your EpiMeta database.

FIGURE 13: ANALYTIC MANAGER MAIN WINDOW



Each of these icons represents a folder. Refer to Table 1 on page 14 for a description of these folders.

GENERATING A STAR SCHEMA

The **star schema** is the blueprint for building an EpiCenter because it defines all the fields in the dimension and fact tables. You create a star schema by doing the following:

- Creating Base Dimension Tables
- Creating Dimension Roles
- Defining Degenerate Dimensions
- Creating Fact Tables
- Generating the Schema

You define the star schema using the Analytic Manager Schema folder. When you expand the Schema folder, the icons for the sub-folders appear.

Each of the icons represents a subfolder that contains elements of the star schema .

TABLE 3: STAR SCHEMA ELEMENTS

ELEMENT	DESCRIPTION
Base Dimensions	Hold the dimension tables in the EpiCenter.
Dimension Roles	Hold the mappings from dimensions to fact-table.
Degenerate Dimensions	Contain fact table text fields that do not map to a base dimension.
Facts	Hold EpiCenter fact tables.
Schema Observatory	Provides a visual representation of EpiCenter schema.

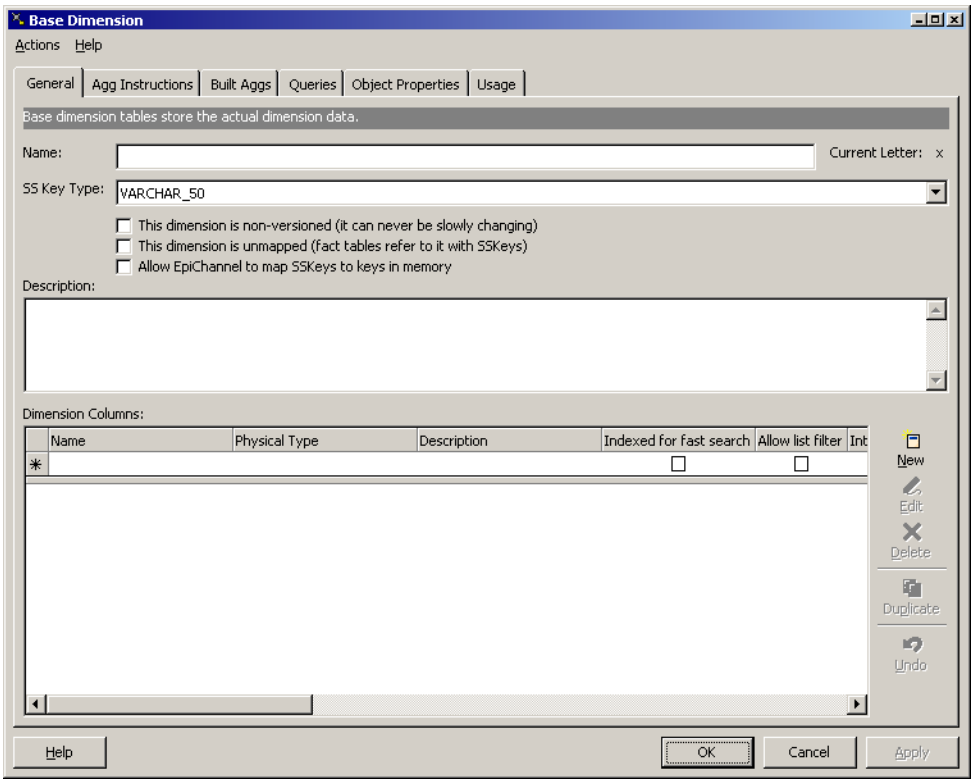
CREATING BASE DIMENSION TABLES

Base dimension tables are physical dimension tables that define the axes along which you can slice and dice your data. Base dimensions are global objects that can be used throughout the EpiCenter.

ADDING A BASE DIMENSION

1. In Analytic Manager, navigate to the Schema folder and select the Base Dimensions subfolder.
2. Right-click Base Dimensions and select New Base Dimension. The Base Dimension dialog box appears.

FIGURE 14: THE DIMENSION DIALOG BOX

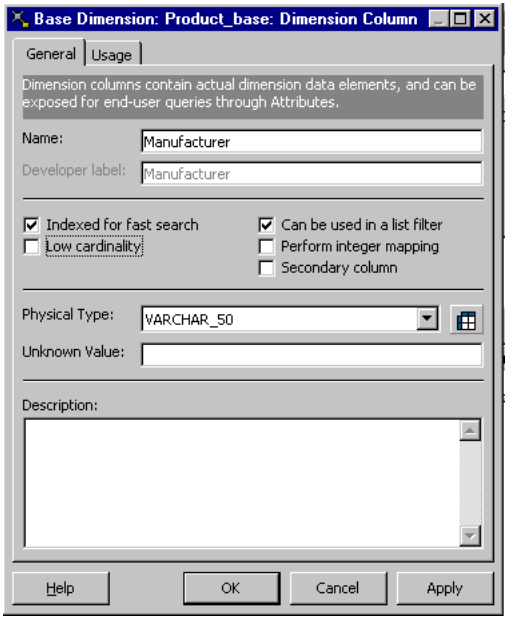


3. Enter the name and description of a base dimension. For the sample datamart, enter the name **`Product_base`**.
4. Select an SSKey Type.

The sskey type is a key that uniquely identifies each member of the dimension. The sskey type is always a string value, and it must be long enough to hold any possible key. The default value of VARCHAR_50 is sufficient for all base dimensions in the sample datamart.

5. Leave the This Dimension is Unmapped, This Dimension is Non-Versioned, and Allow EpiChannel to Map SSKeys to Keys in Memory check boxes unchecked.
6. Click New. Analytic Manager displays a message asking if you want to create a default dimension role.
7. Click No. The Dimension Column dialog box appears.

FIGURE 15: THE DIMENSION COLUMN DIALOG BOX



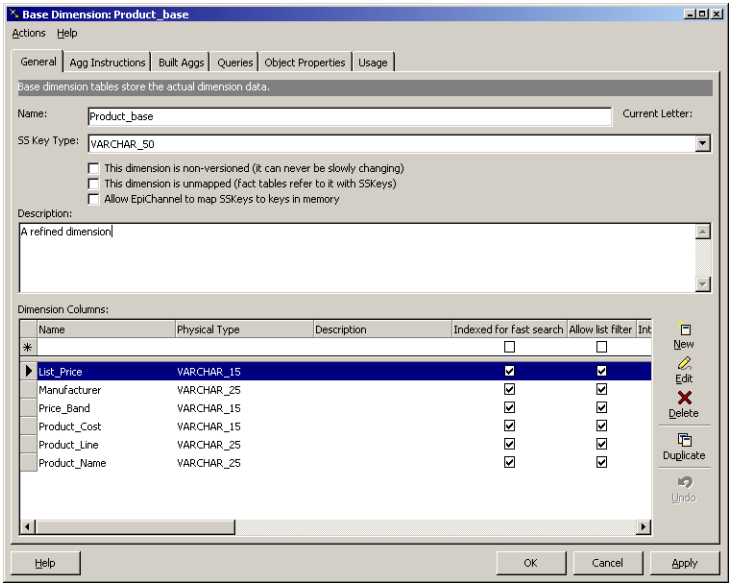
8. Enter the column name and a description. For the sample datamart, enter the name **Manufacturer**.
9. Select the appropriate Physical Type from the list box. For the Manufacturer column, select VARCHAR_50.

10. Check Indexed for fast search and Can be used in a list filter.
11. Enter a default value in the Unknown Value text box. If no value is entered in this text box, then the value UNKNOWN is used as a default. For the sample datamart, you do not enter a default value.
12. Click OK to define the column. The specification for the column you defined is listed in the Dimension dialog box.
13. Repeat steps 6 through 12 to define the following columns for the dimension:

```
List Price (VARCHAR_15)
Price_Band (VARCHAR_15)
Product_Cost (VARCHAR_15)
Product_Line (VARCHAR_25)
Product_Name (VARCHAR_50)
```

The completed Product_base dimension for the sample datamart is as follows.

FIGURE 16: THE BASE DIMENSION DIALOG BOX



NOTE: *INT refers to integers and VARCHAR refers to alphanumeric values. For more information on other data types, refer to the Datamart Guide.*

14. Click OK to complete creating the dimension table.
15. Now repeat Steps 2 through 14 to create the remaining dimensions. If you are using the sample datamart, add dimensions for Individual, Household, Quick Count, Marketing Program, and Sales Person. The details for these dimensions are illustrated in Appendix A, "Sample Datamart." (See Figures 124 through 129).

CREATING DIMENSION ROLES

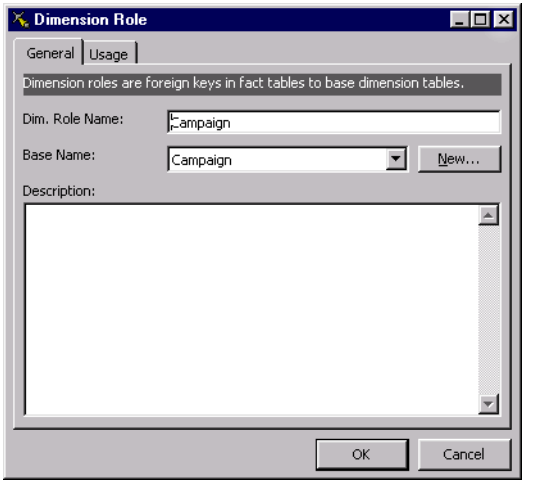
D**imension roles** specify the role that base dimensions can play for fact tables. A base dimension must have at least one associated dimension role to be used in a fact table. A base dimension can also have more than one associated dimension role.

For example, if there is significant overlap between suppliers and customers, you can have a single base dimension for both. You can then define separate supplier and customer dimension roles, both of which refer to the same dimension. This configuration allows a single fact table entry to refer to two elements of the dimension, once in the role of supplier and once in the role of customer.

CREATING A DIMENSION ROLE

1. In Analytic Manager, navigate to the Schema folder and select the Dimension Roles subfolder.
2. Right-click Dimension Roles and select New Dimension Role. The Dimension Role dialog box appears with the default Campaign dimension role.

FIGURE 17: THE DIMENSION ROLE DIALOG BOX



3. Enter a name for the dimension role. For the sample datamart, enter the name **`Product`**.
4. Select the Base Name from the list box. For the sample datamart, Select the Product_base dimension.
5. Enter a description.
6. Click OK to create the dimension role.

The datamart contains several pre-defined dimension roles that are used by the List Manager. In order to use the List Manager, the indiv and group dimension roles should refer to those base dimensions that contain individuals and groups of individuals. In the sample datamart, they should refer to the Individual_base and Household_base dimensions, respectively.

To assign a base dimension to the indiv dimension role:

1. Double-click the indiv dimension role. The Dimension Role: indiv dialog box appears. The name indiv is dimmed and cannot be changed.
2. Select a base dimension from the Base Name list box. For the sample datamart, Select the Individual_base dimension.
3. Click OK to assign the base dimension to the dimension role.

   Repeat steps 1-3 to assign a base dimension to the group dimension role. For the sample datamart, select the Household_base base dimension for the group dimension role.

   Define dimension roles for the other base dimensions that you created. Follow the procedure that you used to create your first dimension role (in the sample datamart, it is the Product dimension role). For a list of all the completed dimension roles for the sample datamart, see "The Dimension Roles," on page 244.

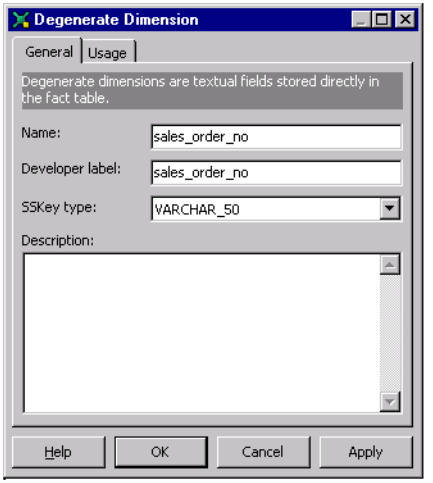
DEFINING DEGENERATE DIMENSIONS

Degenerate dimensions are fact-table text fields that do not map to a base dimension.

DEFINING A DEGENERATE DIMENSION

1. In Analytic Manager, navigate to the Schema folder and select the Degenerate Dimension subfolder.
2. Right-click Degenerate Dimension and select New Degenerate Dimension. The Degenerate Dimension dialog box appears.

FIGURE 18: THE DEGENERATE DIMENSION DIALOG BOX



3. Enter a name and description for the degenerate dimension. The sample datamart has a single degenerate dimension called sales_order_no.
4. Select an SSKey Type. The default value of VARCHAR_50 is acceptable for sales_order_no.
5. Click OK to define the degenerate dimension.
6. Repeat Steps 2-5 to define other degenerate dimensions. The sample datamart has no other degenerate dimensions.

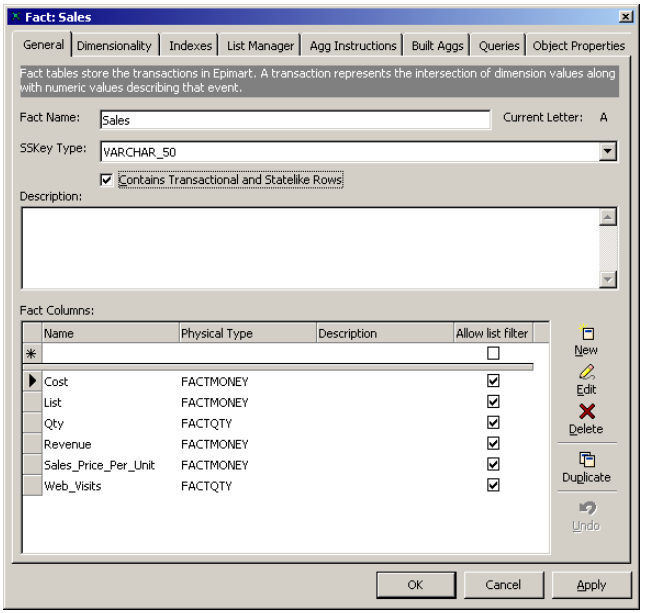
CREATING FACT TABLES

Fact tables are physical database tables that hold numeric data, degenerate dimensions and dimension-role foreign keys.When you create a fact table, you create a new base table, specify any dimension roles that the fact table will reference (including degenerate dimensions), and create at least one cluster.

CREATING A FACT TABLE

1. In Analytic Manager, navigate to the Schema folder and select the Facts subfolder.
2. Right-click Facts and select New Fact. The Fact dialog box appears.

FIGURE 19: THE FACT DIALOG BOX



3. Enter the Fact table name and description. For the sample datamart, enter the name **`Sales`**.

4. Click New Column. The Fact Column dialog box appears. Fact columns specify the types of numeric entries in your fact table.
5. Under Fact Columns, enter the column name and the physical type of the data it will hold. For the sample datamart, enter the name **`Cost`**, with a data type of FACTMONEY. Check the Allow List Filter check box.
6. Click OK to add the column and press Enter.
7. Add the following fact columns:
    - List
    - Qty
    - Revenue
    - Sales_Price_Per_unit.
    - Web_Visits

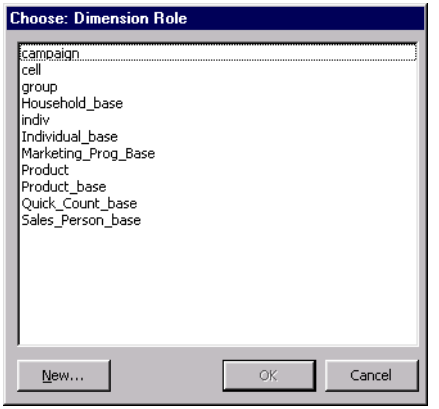
    Select FACTMONEY as the Physical Type and check Allow List Filter check box. For the Web_Visits column, select FACTQTY physical type and check Allow List Filter check box.
8. Create a second fact table called Marketing, with a **`Touchcount`** fact column and FACTQTY datatype. Check the Allow List Filter check box.

SPECIFYING DIMENSION ROLES

You specify dimension roles in the Fact Table dialog box as follows:

1. In the Fact Table dialog box, click the Dimensionality tab.
2. Click the Add icon. The Choose Dimension Role dialog box appears.

FIGURE 20: THE CHOOSE DIMENSION ROLE DIALOG BOX

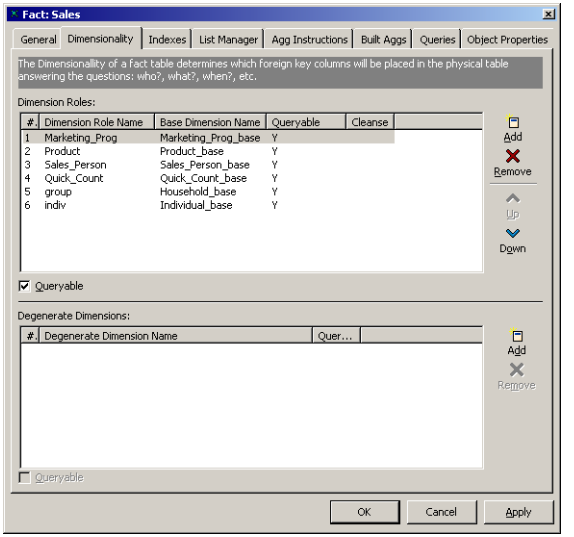


3. Select the dimension roles you want to add to the fact table. For the sample datamart, select the group, indiv, Marketing_Program, Product, and Quick_Count dimension roles.

NOTE: *If you want to select several dimension roles at the same time, hold down the Ctrl key.*

4. Click OK to add the dimension roles. The names of the dimension roles appear in the upper pane of the Fact Table dialog box.

FIGURE 21: THE FACT TABLE DIALOG BOX - DIMENSIONALITY TAB



5. With the dimension roles still selected, at the bottom of the Dimension Roles pane, check the Queryable check box .
6. Add any degenerate dimensions that the fact table uses. To add degenerate dimensions, click Add and then select the appropriate degenerate dimensions from the dialog box. If you are using the sample datamart, you do not have to add any degenerate dimensions to the Marketing table.

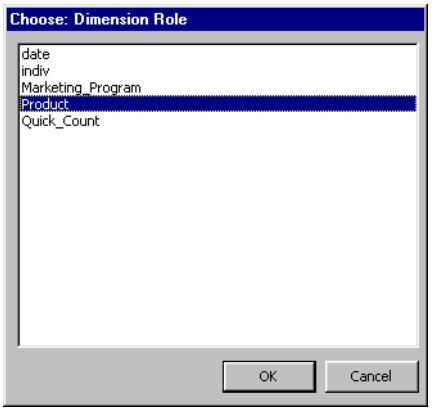
CLUSTERS

List management and campaign management use **clusters** on the fact tables you create. A fact cluster is an index on a List Manager table that indexes facts based on selected dimension. For more information on list management or campaign management, refer to Chapter 8, "Creating Web Pages for Lists and Campaigns."

CREATING A CLUSTER

1. In the Fact Table dialog box, click the Clusters tab.
2. Click New. The Choose Dimension Role dialog box appears.

FIGURE 22: THE SELECT DIMENSION ROLES DIALOG BOX



3. Select a dimension role. For the sample datamart, select the Product dimension role.
4. Click OK to add clusters for the selected dimension roles. The selected dimension role appears in the Clusters pane.
5. Click OK to complete defining the fact table.

INITIALIZING EPIOP

The EpiOp database stores the data that it generates. When you create a fresh EpiCenter, you have to initialize the EpiOp database. After you initialize the EpiOp database, you do not have to initialize it again.

INITIALIZING THE EPIOP DATABASE

1. In Analytic Manager, from the EpiCenter menu, select Initialize EpiOp. The Initializing EpiOp dialog box appears.
2. Check Trial Run and click Go to perform a trial run of the EpiOp initialization.
3. If the trial fun completes successfully, uncheck Trial Run and click Go to initialize the EpiOp database. The run results are displayed.
4. After initialization is completed successfully, click Close.

When you create an EpiCenter, you have to initialize the EpiOp database. Initialization changes an empty database into an EpiCenter.

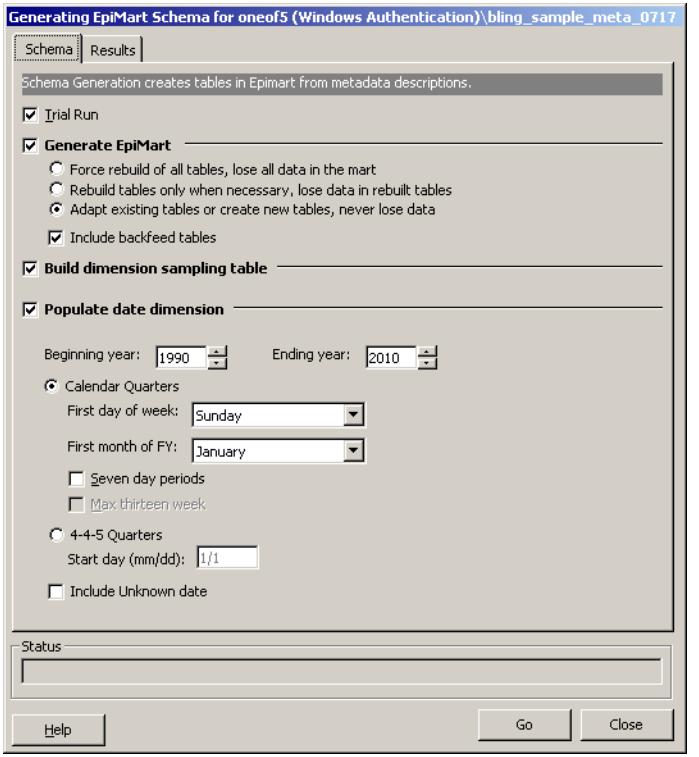
GENERATING THE SCHEMA

Generating the star schema creates the physical tables within the database with space for the fields or columns that you define. Generating the schema also creates the key fields that define the joins between tables. This way, the dimension tables are attached to the fact tables to create the star schema that you defined in Chapter 2.

GENERATING THE SCHEMA

1. In Analytic Manager, do one of the following:
   - Select EpiMeta. Right-click and select Generate Schema.
   - From the EpiCenter menu, select Generate Schema. The Schema Generation dialog box appears.

FIGURE 23: THE SCHEMA GENERATION DIALOG BOX



2. Verify that Generate EpiMart is checked, and select Rebuild Tables Only When Necessary.
3. Check Trial Run, and click Go to start the trial run.

   When you run schema generation with the Trial Run option, Analytic Manager tests the schema you have specified, but makes no changes at this time. A trial run allows you to check your specifications for consistency, and it gives you information on what changes will be made in your EpiCenter if you continue with the actual schema generation.

4. When the trial run is completed, click the Results tab to see details of the changes that would be made, or for an explanation of any problems that may have been encountered.
5. Return to the Schema tab and uncheck Trial Run. Make sure EpiMart Generation is still checked and Rebuild Tables Only When Necessary is selected.

   If you are building a new datamart or completely rebuilding an existing datamart (discarding all data previously in the datamart), you can select Force Rebuild of All Tables instead of Rebuild Tables Only When Necessary. This drops all existing EpiMart tables and recreates them using the parameters or values that you have specified.

   If you are making changes to an existing datamart that already contains data, the Adapt Existing Tables or Create New Tables option is safer than the Rebuild Tables Only When Necessary option, but it may not be sufficient for all changes that you have made to your schema.
6. Check Include Backfeed Tables.
7. If you are using the List Manager, check Build Sampling Table.
8. Check Populate Date Dimension, and enter appropriate time and date values for the desired range of dates. Enter the range of years covered by your database, as well as the details of your fiscal year calendar.

   For the sample datamart, use 1994-1997 with calendar quarters, weeks starting on Monday, and the fiscal year starting in January.
9. Check Include Unknown Date.
10. Click GO to generate the schema.
11. When schema generation is completed, click the Results tab to view the results.
12. Click Close.

Extracting Data into the Datamart

This chapter describes **extraction**, the the process of copying data from source databases into the EpiCenter datamart. The extraction process consists of defining the data stores, defining the extraction job, and running the extraction job.

Defining Source System Data Stores

Data stores are databases that are available for Analytic Manager. Analytic Manager has four pre-defined data stores. To view these data stores, in Analytic Manager navigate to the Extraction folder and expand the Data Stores folder. The expanded folder reveals the following four pre-defined data stores:

Table 4: Pre-Defined data Stores

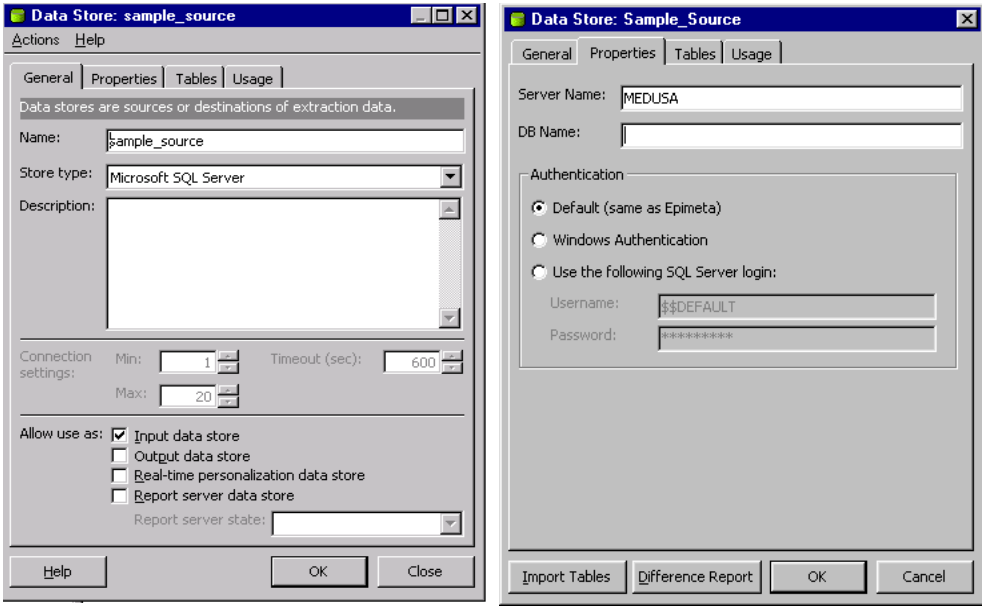
EpiMart	stores the data extracted from the source system.
EpiMeta	stores the metadata.
JobFileLog	stores the output created when a job is run, such as detailed extraction job logs and system call output.
ReportMeta	refers to the reporting metadata in a factory-server configuration. This data store is not used for the sample datamart.

You have to create additional data stores for any other input databases that you use. For the sample datamart, create a single input data store for the sample input database.

CREATING A NEW DATA STORE

1. In Analytic Manager, navigate to the **Extraction** folder, and select the **Data Stores** sub-folder.
2. Right-click and select **New Data Store**. The Data Store dialog box appears.

FIGURE 24: THE DATA STORE DIALOG BOX



3. Enter a name and description. For the sample datamart, enter **`Sample_Source`**.
4. Select the appropriate **Data Store Type**. For the sample datamart, the data store type is the same as the database server you use.
5. Under **Allow Use As**, check **Input Data Store** and uncheck **Output Data Store** to make the new data store input-only.
6. Select the **Properties** tab.

7. Enter the server name and database name. The name of the sample source database is **`sample_source_db`**.
8. Select an Authentication mode. Use the same mode that your database server is set up to use.
   - If you are using Windows NT Integrated Security, select the Windows Authentication radio button.
   - If you are using SQL Server Authentication mode, select the Use the following SQL Server login radio button, and specify the appropriate user name and password.
9. Click OK to define the data store. The icon for the data store you created appears under the Jobs subfolder.

DEFINING AN EXTRACTION JOB

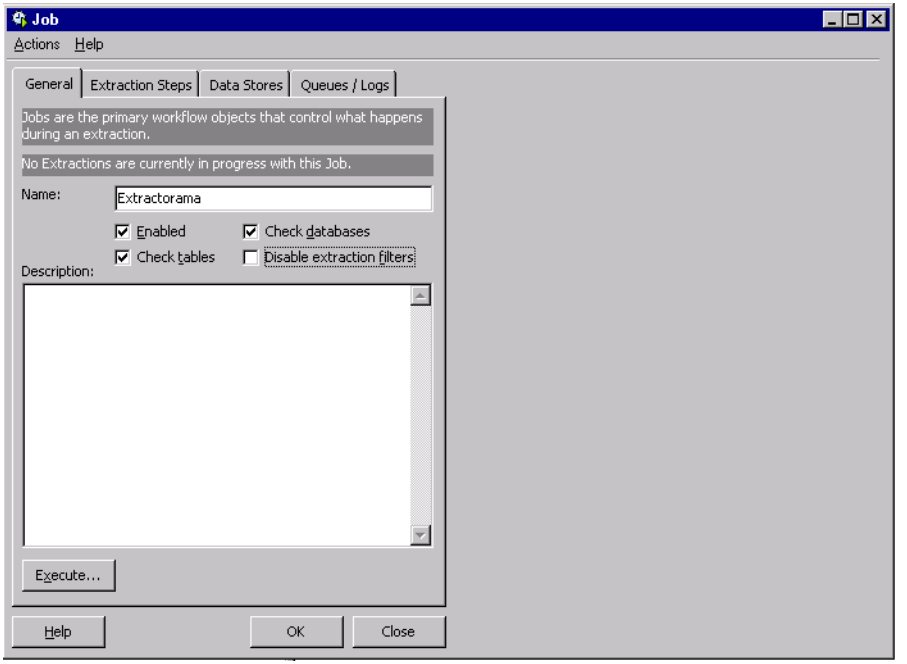
An **extraction job** is a sequence of extraction steps applied to a set of data stores. To define a job, you create the job, create any necessary extraction steps, create global and local extraction groups or job steps, assign extraction steps to groups, and assign data stores to job steps.

When you expand the Extraction folder, and the Jobs subfolder, you see the following predefined (default) jobs: Backfeed501, BackfeedRollOff501, Campaign501, CampaignUpgrade501, Default501, and FactRolloff501. These jobs can be used as starting points for many extraction jobs. For the sample datamart, create a new extraction job.

CREATING A NEW EXTRACTION JOB

1. In Analytic Manager, navigate to the Extraction folder, and select the Jobs sub-folder.
2. Right-click the Jobs folder, and select New Job. The Job dialog box appears.

FIGURE 25: THE JOB DIALOG BOX



3. Enter a name for the job. For the sample datamart, enter **`Extractorama`**.
4. Enter a description for the job (optional).
5. Make sure that the Check databases, Check tables, and Enabled check boxes are checked. The Check databases and Check tables direct the EpiChannel utility to make sure all referenced databases and tables are accessible before proceeding with the job.

6. Click OK to define the job.
7. Expand the Jobs sub-folder. The icon for the extraction job you created appears.

Extraction Steps and Groups

Extraction steps move data into the datamart. Extraction steps can be grouped into **extraction groups**. Extraction groups can contain other extraction groups and individual extraction steps.

A job consists of a sequence of job steps, each of which can be a local job group, a global group, or an individual extraction step.

Extraction Steps and Groups for Dimensions

You define extraction steps for dimensions and facts. This section describes how to define extraction steps and groups for dimensions. For information on extraction steps and groups for facts, refer to "Extraction Steps and Groups for Facts," on page 80.

Extraction steps for dimensions involves defining:

- Global Extraction Steps and Groups for Dimensions
- Truncation Steps for Dimensions
- Extraction Command Steps for Dimensions
- Semantics for Dimensions

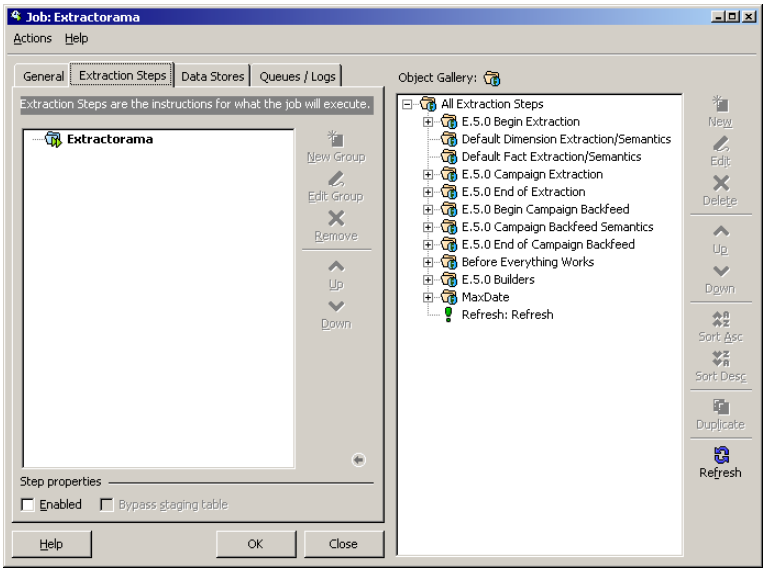
Global Extraction Steps and Groups for Dimensions

Global extraction steps and groups are created and modified under the Job dialog box Extraction Steps tab, but they exist independently of any job, and can be reused for various jobs. Local groups, on the other hand, are specific to an individual job; they can only be used for that particular job.

CREATING A GLOBAL EXTRACTION GROUP FOR DIMENSIONS

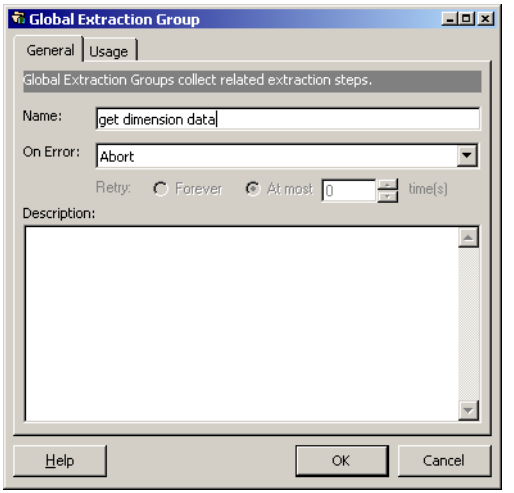
1. Open the **Extractorama** job. The Job dialog box appears.
2. Click the **Extraction Steps** tab. The **Object Gallery** is on the right of the Job dialog box. The available extraction steps and groups appear in the **Object Gallery**.

FIGURE 26: THE EXTRACTION STEPS TAB OF THE JOB DIALOG BOX



3. Right-click All Extraction Steps and from the New menu, select Group. The Global Extraction Group dialog box appears.

FIGURE 27: THE GLOBAL EXTRACTION GROUP DIALOG BOX



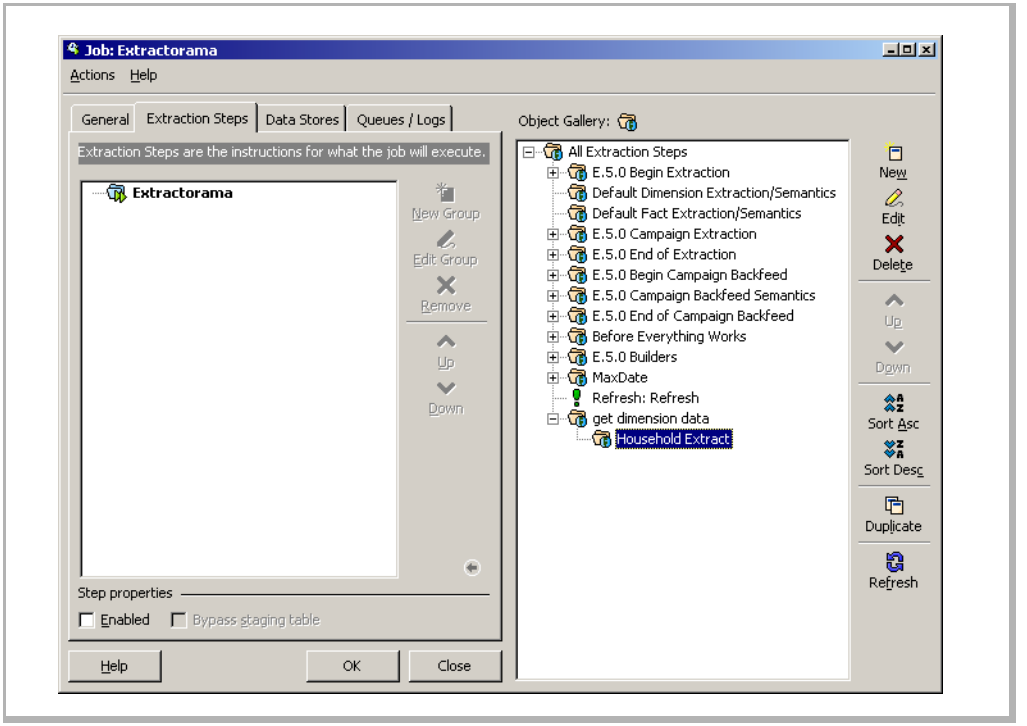
4. Enter a name. For the sample datamart, enter **`get dimension data`**.
5. In the On Error list box, select AutoRepeat.
6. Click OK to define the group. The group now appears under All Extraction Steps.

To create another global extraction group within the group you defined above, do the following:

1. Right-click the folder for the group you just defined. For the sample datamart, this group is called get dimension data.
2. From the New submenu, select Group. The Global Extraction Group dialog box appears.
3. Enter a name. For the sample datamart, enter **`Household Extract`**. In the On Error list box, select AutoRepeat.

4. Click OK to define the group. The group now appears under the get dimension data group.

FIGURE 28: GROUPS: GET DIMENSION DATA AND HOUSEHOLD EXTRACT



TRUNCATION STEPS FOR DIMENSIONS

Typically, the staging table for a fact or dimension is truncated (that is, all data in the table is deleted) before the table is populated with newly extracted data. This allows for the extraction restart ability. For more information on the restart ability, refer to chapter 5, "Analytic Platform Database Extraction" and chapter 7, "Configuring Extraction Jobs" of *Datamart Guide*. You create **truncation steps** for all fact and dimension tables that you define.

CREATING A TRUNCATION STEP

1. Right-click the **Household Extract** group that you defined. From the new menu, select **Truncation Step**. The Truncation dialog box appears.
2. Select the kind of table to be truncated (**Dimension**, **Fact**, **External Table**). For the sample datamart, select **Dimension**.
3. Select the name of the table to be truncated. For the sample datamart, select **Household_base**.
4. Select **AutoRepeat from the On Error** list box.
5. Click **OK** to define the step. The step appears under the **Household Extract** group.

FIGURE 29: EXTRACT GROUP WITH TRUNCATION STEP

EXTRACTION COMMAND STEPS FOR DIMENSIONS

The next step in moving data from the source database into the datamart is to move it into staging tables using SQL queries. Staging tables are tables for new data for your facts and dimensions that contain the keys that you specified when you defined your star schema.

NOTE: *The Analytic Platform uses internal staging tables during extraction. These staging tables are created when you initialize the EpiCenter.*

Analytic Manager provides the Extraction Wizard, a graphical tool for defining extraction commands. For instructions on using the Extraction Wizard, see "Using the Extraction Wizard for Defining Extraction Commands," on page 96. You can also use a SQL statement to define an extraction command. Analytic Manager provides SQL code templates for staging tables and external tables. Customize the templates based on the structure of your source system.

FORMULATING SQL QUERIES

When you formulate SQL queries, E.piphany recommends the following guidelines:

- The SQL code must be vendor independent.

  E.piphany provides a large collection of macros to facilitate the writing of vendor-independent SQL. Refer to the *Datamart Guide* for a complete description of all available macros.

- The data must not contain null values.

  Replace all null values with values that do not appear in your data. The string `'UNKNOWN'` is generally used for this purpose for string values. This replacement can be done with the `$$NVL` macro.

In addition to the fields that you have defined for the dimension, every dimension table also includes the following additional special fields:

*TableName*_sskey	The source-system key is a unique identifier for an element of a dimension. The key is used to associate a record with future changes to that record. (*TableName* is the name of the table—for example, the Household_base table has a Household_base_sskey field).
date_modified	Indicates the date when the dimension value was updated.

- Use `SELECT DISTINCT` when you write your SQL code to avoid duplicate dimension values.

- The sample datamart extraction commands extract all rows from the source database. If you are doing regular extractions from a source database, extract only values that have changed since the last extraction. E.piphany provides several macros, such as `$$COLUMN_FILTER` and `$$DATE_FILTER`, that make this process easy. For more information, refer to the *Datamart Guide*.

ADDING AN EXTRACTION COMMAND TO A NEW EXTRACTION GROUP

1. Open the Job dialog box for your new extraction job (for the sample datamart, open the Extractorama job), and click the Extraction Steps tab.
2. Expand the All Extraction Steps folder and right-click the extraction group you created (for the sample datamart, this is Household Extract). From the New submenu, select Extraction Command. Analytic Manager displays the Extraction Wizard.

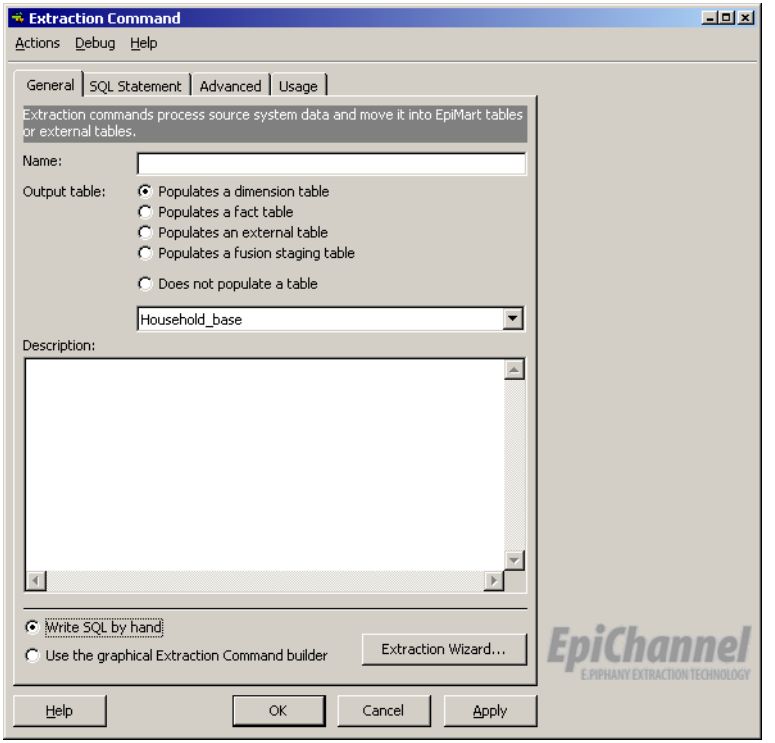
   You can use the Extraction Wizard to create the extraction command or you can use the standard dialog box interface. For more information on the Extraction Wizard, refer to chapter 7, "Extraction Wizard" of *Datamart Guide*.

   To switch to the dialog box interface, click the Create Extraction Command with Editor button. To turn the Wizard off, check the By default, don't start with this wizard check box.
3. Select Write SQL by Hand. The Extraction Command dialog box appears. (See Figure 30 on page 72.)

NOTE: *The steps that follow are tailored specifically to the dialog box interface with the Write SQL by hand radio button selected.*

4. Enter a name for the extraction step. For the sample datamart, enter **`Get Household Data.`**
5. Select Populates dimension table. Select the dimension that you want to populate from the Output table list box. For the sample datamart, select Household_base.

FIGURE 30: THE EXTRACTION COMMAND DIALOG BOX



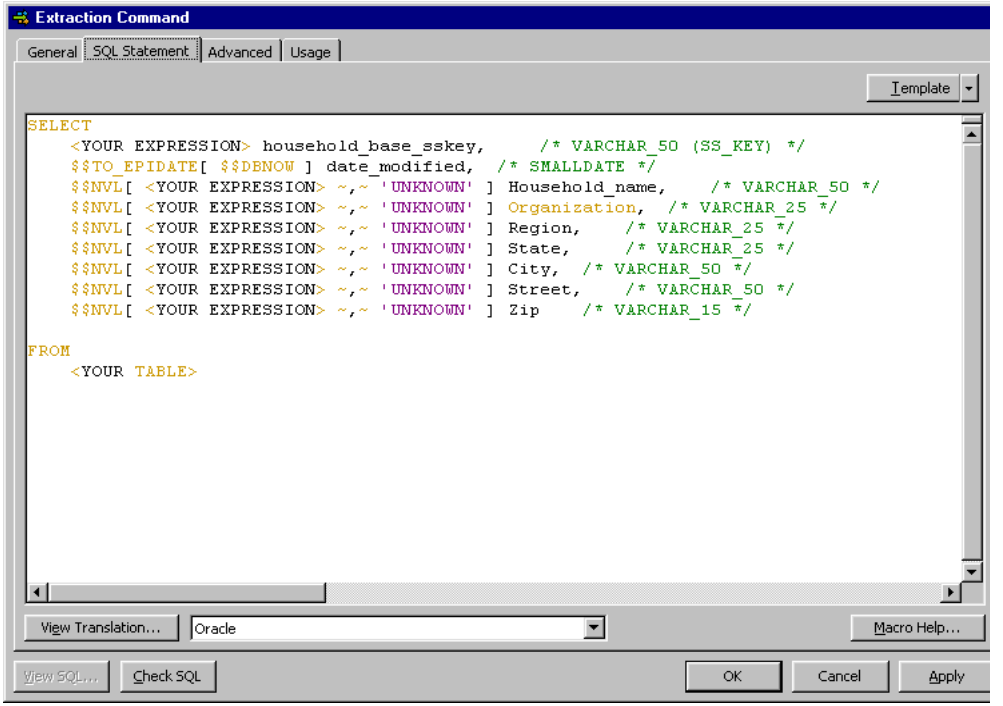
6. Select the Advanced tab. In the On Error list box, select inherit.
7. Select the SQL Statement tab. For the sample datamart, go to Step 8. For other datamarts, proceed to Step 9.

   The SQL code for the sample datamart is provided in the **SqlExtrnCode.doc** file located in the **Resources** subdirectory of the E.5 installation image (CD-ROM or download) and in Appendix A.
8. Copy and paste the "Get Household" section into the SQL pane. (Do not include the title.)

   Go to Step 12, unless you prefer to use the templates and alter the code yourself.

9. Click Template to fill the SQL pane with a template for your SQL code.

FIGURE 31: EXTRACTION COMMAND: SQL STATEMENT TAB WITH TEMPLATE



The SQL templates are based on the structures of the destination tables, derived from the schema you defined. If you change the schema and regenerate it, the template will be altered accordingly. In this case, update the SQL statement in the SQL pane before extracting.

As shown in Figure 31, on page 73 the template lists each column in the dimension. Preceding each column name is the term `<YOUR EXPRESSION>`. Replace `<YOUR EXPRESSION>` with an expression that selects the appropriate values.

In the template, the `<YOUR EXPRESSION>` term is placed inside an expression that uses the `$$NVL` macro to replace all null entries with the `'UNKNOWN'` string.

Each line ends with a comment that gives the physical type of the column.

10. Enter an expression for the source system key to replace the <YOUR EXPRESSION> term that precedes *dimension_name*_sskey (where *dimension_name* should be replaced by the name of the dimension). Each element in the dimension must have a unique source system key. To convert a numeric key to character format, use the $$_TOCHAR macro.

    For the sample datamart, use the Household_ID field of the source database table Household_table for the Household_base_sskey field.

11. Enter expressions for the remaining fields. If you are using the sample datamart, enter the appropriate expressions to get the following SQL code:

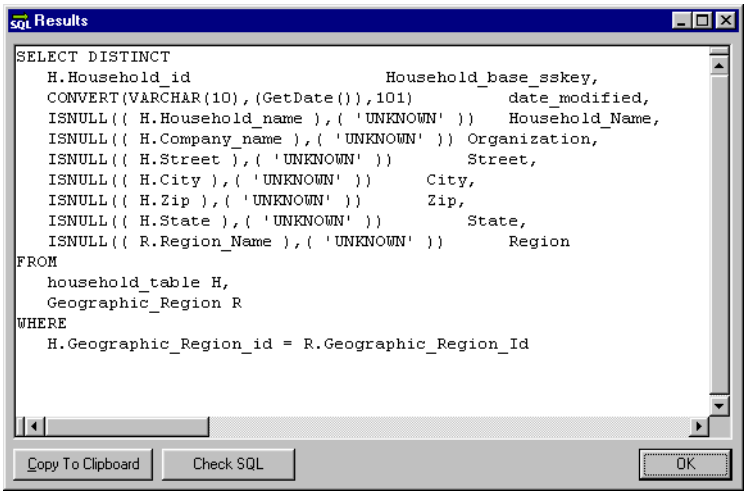
```
SELECT DISTINCT
   H.household_id   household_base_sskey, /* VARCHAR_50 (SS_KEY) */
   $$TO_EPIDATE[ $$DBNOW ] date_modified,      /* SMALLDATE */
   $$NVL[ H.household_name ~,~ 'UNKNOWN' ]
          Household_name,                      /* VARCHAR_50 */
   $$NVL[ H.Company_name ~,~ 'UNKNOWN' ]
          Organization,                        /* VARCHAR_25 */
   $$NVL[ R.Region_name ~,~ 'UNKNOWN' ]
          Region,                              /* VARCHAR_25 */
   $$NVL[ H.State ~,~ 'UNKNOWN' ]   State,     /* VARCHAR_25 */
   $$NVL[ H.Street ~,~ 'UNKNOWN' ]  Street,    /* VARCHAR_50 */
   $$NVL[ H.City ~'~ 'UNKNOWN' ]    City,      /*VARCHAR_50 */
   $$NVL[ H.Zip ~,~ 'UNKNOWN' ]     Zip        /* VARCHAR_15 */

FROM
   household_table H,
   Geographic_Region R

WHERE
   H.Geographic_Region_Id = R.Geographic_Region_Id
```

12. To see how the macros are translated into code for your server platform, select the appropriate server platform from the list box next to View Translation. Click View Translation. Analytic Manager displays the SQL Statement Check dialog box with the correct platform-specific translation.

FIGURE 32: THE SQL RESULTS DIALOG BOX

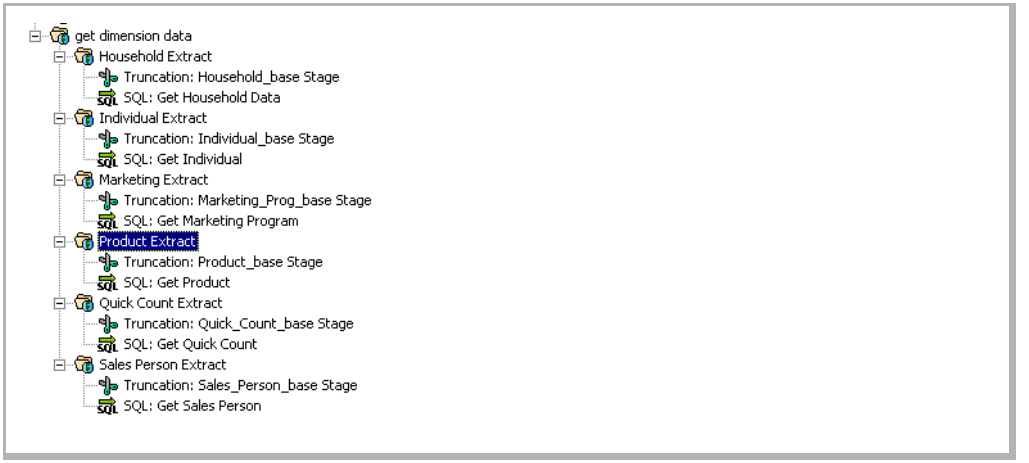


13. Check SQL to check your SQL code for syntactic correctness.
14. Click OK to close the SQL statement check dialog box.
15. Click OK in the Extraction Command dialog box to complete the new extraction step. The extraction step now appears under the appropriate group in the Object Gallery.

FIGURE 33: EXTRACT GROUP WITH TRUNCATION AND SQL STEP

16. Define extraction steps for the remaining dimensions. If you are using the sample datamart, create separate global groups to contain each of the extraction command steps and its accompanying truncation step. Put all of the groups in the get dimension data group.

FIGURE 34: GROUPS CONTAINED WITHIN THE GET DIMENSION DATA GROUP



17. For the On Error value, select inherit for the SQL steps, and select AutoRepeat for the other steps and groups.

    The completed group contains the following extraction steps:

    - Household Extract
      - Truncation: Household_base Stage
      - Extraction: Get Household
    - Individual Extract
      - Truncation: Individual_base
      - Extraction: Get Individual
    - Marketing Program Extract
      - Truncation: Marketing_prog_base
      - Extraction: Get Marketing Program

- Product Extract
  - Truncation: Product_base Stage
  - Extraction: Get Product
- Quick Count Extract
  - Truncation: Quick_Count_base Stage
  - Extraction: Get Quick Count
- Sales Person Extract
  - Truncation: Sales_Person Stage
  - Extraction: Get Sales_Person

SEMANTICS FOR DIMENSIONS

After defining extractors to populate the staging tables, you define the extraction steps that merge the data into your datamart. **Semantics** are pre-written standard rules for managing the merging of dimension and fact-table data.

At this point, you have not yet populated your datamart with data, so merging essentially consists of copying in the data from the staging tables. However, when you merge data into an existing datamart that already contains data, you have to do the following:

- Relate the new data to existing data.
- Assign artificial keys where needed.
- Compensate for referential-integrity errors.
- Convert amounts reported as balances to incremental (transactional) amounts.
- Track items through a pipeline of stages.
- Construct aggregates.
- Merge recent and historical data.

Analytic Manager performs the above processes automatically depending on the semantics specified in the extractors. E.piphany provides semantics for a wide range of business data models and you can select the correct semantic type for your data. For a description of the available semantic types, refer to the The *Datamart Guide*.

For all dimension semantics, any staging table row with a new sskey (that is, a row with an sskey that is not currently in any dimension table row) is appended to the dimension table. If an sskey appears more than once in the staging table, only the last row containing that sskey is merged into the dimension table. The semantics differ in how they handle this merging.

The available semantic types for dimensions include:

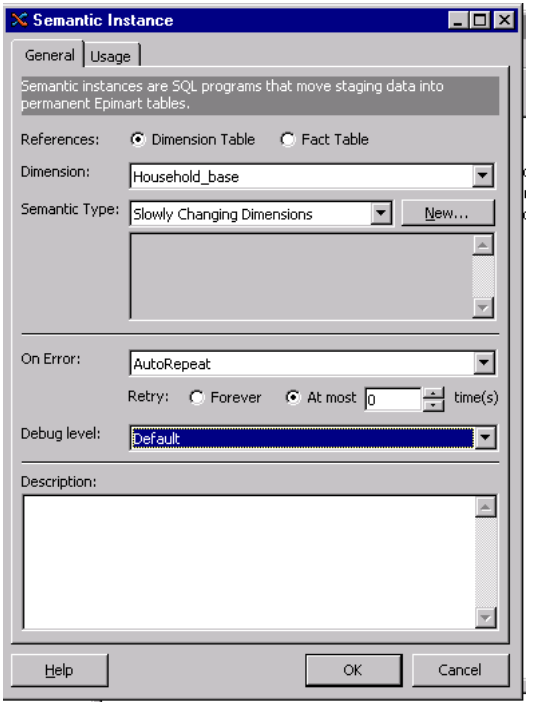
Latest Dimension Value	If a row in the staging table has the same sskey as a row in the datamart, the values in the staging table row replace those currently in the datamart.
First Dimension Value	If a row in the staging table has the same sskey as a row in the datamart, then that row of the staging table is discarded.
Slowly Changing Dimensions	If a row in the staging table has the same sskey as a row in the datamart but has different values in other columns, then that row of the staging table is appended to the datamart table.
Initial Load Dimension	The entire dimension table is reloaded from the staging table, without regard to the current contents of the dimension table.

DEFINING DIMENSION SEMANTICS

1. Open the new extraction job. For the sample datamart, open the Extratorama job. The Job dialog box appears.
2. Click the Extraction Steps tab.
3. Right-click All Extraction Steps, and select Group from the New submenu. The Global Extraction Group dialog box appears.
4. Enter a name for a new group for dimension semantics. For the sample datamart, enter dimension semantics. In the On Error list box, select AutoRepeat.

5. Click OK to define the new group. The new group appears under the All Extraction Steps group.
6. Right-click your new group and from the New submenu, select Semantic Instance. The Semantic Instance dialog box appears.

FIGURE 35: SEMANTIC INSTANCE

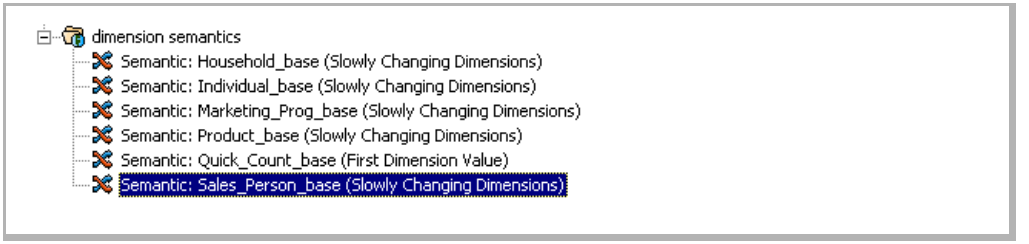


7. Select a dimension table from the Dimension list box. For the sample datamart, select Household_base.
8. Select an appropriate semantic type from the Semantic Type list box. For the sample datamart, select Slowly Changing Dimensions.

The source data for the sample datamart will not be updated, so your choice here will not have a noticeable effect. However, if you were planning to merge updated data into the Household_base dimension at some later time, then Slowly Changing Dimensions would be an appropriate choice.

9. In the On Error list box, select Auto Repeat.
10. Set the Debug level to default.
11. Click OK to define the semantic instance.
12. Repeat Steps 6-11 to define semantics for other dimensions for which you have created SQL extraction steps. (See Figure 36 on page 80.) For the sample datamart, select the Slowly Changing Dimensions semantic type for all dimensions except Quick_Count_base dimension. The Quick_Count_base dimension will not change, so you can assign it the First Dimension Value semantic.

FIGURE 36: DIMENSION SEMANTICS



EXTRACTION STEPS AND GROUPS FOR FACTS

You perform the same types of extraction steps for facts as you do for dimensions. For facts, you define the following:

- Global groups
- Truncation steps
- Extraction command steps
- Semantics

Defining Global Extraction Steps and Groups for Facts

The procedure for setting up extraction steps and groups for facts is similar to the procedure used for dimensions.

Defining Global Extraction Steps and Groups for Facts

1. Right-click the Extractorama extraction job, and select edit. The Job dialog box appears.
2. Click the Extraction Steps tab.
3. Right-click All Extraction Steps and from the New menu, select Group. The Global Extraction Group dialog box appears.
4. Enter a name for a new global group for facts. For the sample datamart, enter **`Get Fact Data`**.
5. In the On Error list box, select AutoRepeat.
6. Click OK. The new global group appears under the All Extraction Steps folder.
7. Create another global group, contained within the group you just created. For the sample datamart, enter the name **`Sales Extract`** for the new group. In the On Error list box, select AutoRepeat.

Truncation Steps for Facts

As with dimensions, for facts it is necessary to create a truncation step to accompany each extraction command step that you define, to allow for the extraction restart ability. For more information on restart ability, refer to chapters 5 and 7 of the *Datamart Guide*.

CREATING TRUNCATION STEPS FOR FACTS

1. Right-click the Sales Extract group and from the New menu, select Truncation Step. The Truncate dialog box appears.
2. Select the kind of table to be truncated (Dimension, Fact, External Table). For the sample datamart, select Fact.
3. Select the name of the table to be truncated. For the sample datamart, select Sales.
4. Select AutoRepeat from the On Error list box.
5. Click OK to define the step. The step appears under the Sales Extract group.
6. Click OK to complete creating the truncation step.

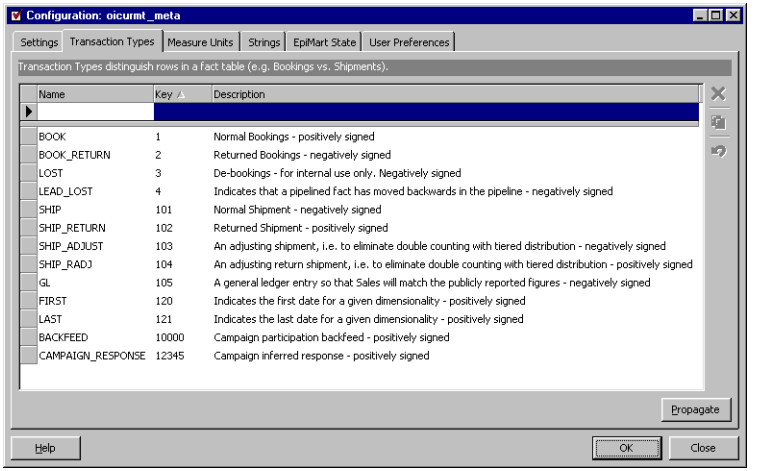
EXTRACTION COMMAND STEPS FOR FACTS

In addition to the extractors for dimensions, you also create extractors for facts. Fact tables have several special fields:

ss_key	The source-system key for the fact tables. Each fact contains an ss_key that uniquely identifies that fact. It is used to associate a record with future changes to that record. The ss_key has the same role in a fact table that *TableName*_sskey has in a dimension table.
date_key	The date key. This key indicates the date of the fact.

Transtype_key	The transaction type key. This is used to apply different aggregation math to different records. For the sample datamart, the default transaction types are sufficient. See Figure 37, on page 84 for the default transaction types. The transaction types that are used for the sample datamart are `1` for `BOOK` and `101` for `SHIP`. The Analytic Manager Configuration dialog box contains a list of available transaction types under the Transaction Types tab. You can also use this tab to define new transtypes.
Process_key	The process key. This key identifies whether a fact is transactional (value of 1) or state-like (value of 2).
*DimRoleName*_sskey	The source-system foreign key to a base dimension. There should be an sskey in the fact table for every dimension role. A fact must have a value for every dimension role in its dimensionality. If there is no actual value for some dimension role, use the `'UNKNOWN'` string.
*degenerate*_key	The key for degenerate dimensions, where *degenerate* is the name of the degenerate dimension.

FIGURE 37: DEFAULT TRANSACTION TYPES



A single fact table generally contains several transaction types. The source data for different types of transactions may be structured differently, so it is often convenient to create a separate extraction step for each transaction type.

CREATING AN EXTRACTION COMMAND STEP FOR A FACT TABLE

1. Right-click the new extract group you created and select Extraction Command from the New submenu. Analytic Manager displays the Extraction Wizard.
2. You can use the Extraction Wizard to create the extraction command or you can use the standard dialog box interface. See the *Datamart Guide* for details on the Extraction Wizard.

   To switch to the dialog box interface, click the Create Extraction Command with Editor button. To turn the Wizard off, check the By default, don't start with this wizard check box.

3. Enter a name for the extraction step. For the sample datamart, enter Get Bookings.
4. Select Populates fact table and select a fact table to populate. For the sample datamart, specify the Sales table.
5. Select Write SQL by Hand.

   The instructions that follow are tailored specifically to the dialog box interface with the Write SQL by hand radio button selected.
6. Select the Advanced tab. In the On Error list box, select inherit.
7. To specify the SQL code, click the SQL Statement tab. For the sample datamart, proceed to Step . For other datamarts, proceed to Step 9.

   SQL code for the sample datamart is provided in the **SqlExtrnCode.doc** file located in the **Resources** subdirectory of the E.5 installation image (CD-ROM or download) and Appendix A.
8. Copy and paste the section "Get Bookings" into the SQL pane. (Do not include the title.)

   Proceed to Step 11, unless you prefer to use the templates and alter the code yourself.
9. Click Template to fill the SQL pane with a template for your SQL code.

   The SQL templates are based on the structures of the destination tables, derived from the schema you defined. If you change the schema and regenerate it, the template will be altered accordingly. In such a case, update the SQL statement in the SQL pane before extracting.

10. Enter SQL for the fact. For the sample datamart, use:

```
SELECT
       SO.Sales_Order_Id $$CAT '-' $$CAT OI.Order_Item_Id  ss_key,
       $$TO_EPIDATE[SO.Booked_Date]            date_key,
       1                                       transtype_key,
       1                                       process_key,
       $$NVL[ C.Household_Id ~,~ 'UNKNOWN' ]   group_sskey,
       $$NVL[ C.Contact_Id ~,~ 'UNKNOWN' ]     indiv_sskey,
       $$NVL[ SO.Marketing_program_id ~,~ 'UNKNOWN' ]
                                       Marketing_Program_sskey,
       $$NVL[ OI.Product_Id ~,~ 'UNKNOWN' ]    Product_sskey,
       '1'                                     Quick_Count_sskey,
       $$NVL[ SO.Sales_Order_Id ~,~ 'UNKNOWN' ] sales_order_no_key,
       $$NVL[ SO.Sales_Person_Id ~,~ 'UNKNOWN' ]
                                               Sales_Person_sskey,
       $$NVL[ P.Product_Cost_PC ~,~ 0 ]        Cost,
       $$NVL[ P.List_Price_PC ~,~ 0 ]          List,
       $$NVL[ OI.Number_of_Units ~,~ 0 ]       Qty,
       $$NVL[ P.List_Price_PC*(1-SO.Discount)
          *OI.Number_of_Units~,~0]             Revenue,
       $$NVL[ P.List_Price_PC *
          (1-SO.Discount) ~,~ 0 ]          Sales_Price_Per_Unit,
       $$NVL[ SO.web_visits ~,~ 0 ]        Web_Visits

FROM
       sales_orders  SO,
       Order_items   OI,
       contact_table C,
       Product_table P

WHERE
       SO.Sales_Order_Id = OI.Sales_Order_Id
       AND    SO.Contact_Id = C.Contact_Id
       AND    OI.Product_Id = P.Product_Id
```

   Here `transtype_key` is set to `1`, since this is a booked order, and `process_key` is set to `1`, since this is a transactional entry.

11. Click View Translation to see how the macros are translated. Click OK.
12. Click OK in the Extraction Command dialog box to finish defining the extraction step.

13. Define the remaining fact extraction steps, if any. The SQL extraction code for the fact tables of the sample datamart can be found in Appendix A. There are two SQL extraction steps each for the Sales and Marketing fact tables, one for `BOOK` transaction types and one for `SHIP` transaction types. (See Figure 38.)

FIGURE 38: GROUPS CONTAINED WITHIN THE GET FACTS GROUP

The completed Get Facts extraction group has the following steps:

- Sales Extract
  - Truncation: Sales Stage
  - Extraction: Get Bookings
  - Extraction: Get Shippings
- Marketing Extract
  - Truncation: Marketing Stage
  - Extraction: Get Mailings
  - Extraction: Get Responses
- IndGroup Extract
  - Truncation: Ind_Group_Joiner Stage
  - Extraction: Get IndGroup Facts

SEMANTICS FOR FACTS

Fact semantics operate on two different types of facts. Transactional facts indicate some discrete event such as a purchase or a sales call; their **process_key** value equals 1. State-like facts describe the current state of some business quantity; their **process_key** value is 2.

The available semantic types for facts include:

Transactional	Transactional facts with new **sskey**s are added to the fact table. Transactional facts with **sskey**s that are already in the fact table are added only if they have a newer date. State-like facts are ignored.
Transactional/State-Like	Transactional facts are treated as in the Transactional semantic type. State-like facts with new **sskey**s are added to the fact table. State-like facts with **sskey**s that are already in the fact table are merged into the fact table using an adjustment record. This adjustment record contains the difference between the values of the two states.
Transactional/State-Like/Force Close	This semantic is like the Transactional/State-Like semantic, except that all open bookings must be found in the staging table. Any open booking (that is, any entry with **transtype_key** between `1` and `99`) that is in the fact table but not in the staging table is automatically closed.
Pipelined	This semantic is used for tracking facts as they move through several life cycle phases. It is generally used with custom transtype values that indicate these phases.
Initial Load Fact	The entire fact table is re-loaded from the staging table, without regard to the current contents of the fact table.

Reload Date Fact	This semantic is used to re-load all facts that occurred after the earliest date in the staging table.
Count Unjoined	This semantic informs you of the number of rows that have been transformed by an outer join. No actual updates are performed.
First / Last Fact	This semantic keeps track of the first and last fact values for an sskey. Only the first and last occurrence of a fact is recorded in the fact table.

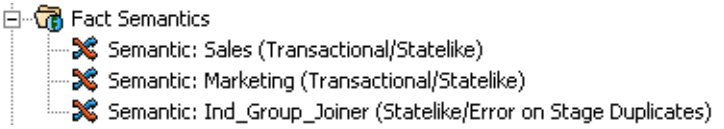
DEFINING FACT SEMANTICS

1. Double-click and open the Extractorama extraction job. In the Job dialog box, click the Extraction Steps tab.
2. Right-click All Extraction Steps, and from the New menu, select Group to add a new group for fact semantics. For the sample datamart, enter the name **`Fact Semantics`**. In the On Error list box, select AutoRepeat. Click OK to define the new group.
3. Right-click your new group and from the New menu, select Semantic Instance. The Semantic Instance dialog box appears.
4. In the References section, select Fact Table. Analytic Manager displays the Fact list box.
5. Select a fact table from the Fact list box. For the sample datamart, select Sales.
6. Select an appropriate semantic type from the Semantic Type list box. For the sample datamart, use Transactional/State-Like.

   You can also use Transactional, since there are no state-like facts for the sample datamart. The Transactional and Transactional/State-Like semantic types differ only with regard to state-like facts.
7. In the On Error list box, select AutoRepeat.

8. Click OK to define the semantic instance.
9. Repeat Steps 3 through 8 to define semantics for the Marketing table.

FIGURE 39: FACT SEMANTIC



RECORDING THE EXTRACTION DATE

The last_extract_date external table records the date of the most recent extraction. The use of this table is optional but E.piphany recommends its use. To enter the extraction date, you have to create another extraction command step. You do not have to create a truncation step for this extraction command step; a truncation step is provided.

CREATING EXTRACTION COMMAND STEP

1. Double-click and open the Extractorama extraction job. The Job dialog box opens.
2. Click the Extraction Steps tab.
3. Expand the All Extraction Steps folder.
4. Expand the E.5.0.1 Extraction Steps folder.
5. Select the MaxDate extraction group and right-click. From the New submenu, select Extraction Command. Analytic Manager displays the Extraction Wizard.

   You can use the Extraction Wizard to create the extraction command or you can use the standard dialog box interface. See the *Datamart Guide* for more information on the Extraction Wizard.

6. To switch to the dialog box interface, click the Create Extraction Command with Editor button. The instructions that follow are tailored specifically to the dialog box interface with the Write SQL by hand radio button selected.
7. Enter a name. For the sample datamart, enter **`Get Extraction Time`**.
8. Select Populates an external table.
9. From the list box, select last_extract_date.
10. Select Write SQL by Hand.
11. Select the Advanced tab. In the On Error list box, select Inherit.
12. Click the SQL Statement tab.
13. Click Template (or copy the SQL code provided in **SqlExtrnCode.doc** file located in the **Resources** subdirectory of the E.4 installation image (CD-ROM or download).

    Analytic Manager displays the SQL template for this table.
14. Enter SQL for the extraction date. This SQL is the code that extracts the maximum date from a table (the name of the table can be recorded in the tbl_name field), or it can be code that simply records the current date. For the sample datamart, the current date is sufficient.

    Use the following code:

    ```
    SELECT
       $$TO_EPIDATE[$$DBNOW] max_date,
       'UNKNOWN' tbl_name
    $$NO_FROM_LIST
    ```

    This data is not from a table, so use the `$$NO_FROM_LIST` macro for platform independence.
15. Click OK. The new extraction step is added to the MaxDate group.
16. Make sure that the new extraction step appears after the Truncation: last_extract_date step.
17. Click OK.

DEFINING LOCAL EXTRACTION GROUPS

The extraction steps in the Object Gallery of the Job dialog box are available for use in any job. A job consists of a sequence of local job steps, each of which can be a local job group or an extraction step. An extraction step can take the form of a global extraction group.

In order to use the extraction groups that you have defined in a job, you define local extraction groups and assign extraction steps to them.

ASSIGNING EXTRACTION STEPS

1. Open the Job dialog box for your new extraction job, and click the Extraction Steps tab. The left pane lists all steps in the job.

   When a job is newly-created, it contains only a single root-level job step, with the same name as the job. For the sample datamart, the root-level job step is Extractorama.

2. Drag the extraction group E.5.0.1 Begin Extraction from the E.5.0.1 Extraction Steps group in the Object Gallery onto the icon for the root-level job step. The extraction group now appears under the root-level job step in the left pane. This group contains a step that cancels any other extraction jobs that may be running.

3. Right-click on the root-level job step and select New Group. Analytic Manager displays the Local Extraction Group dialog box.

FIGURE 40: THE LOCAL EXTRACTION GROUP DIALOG BOX

4. Enter a name and description for the job step. For the sample datamart, enter **`Data Extraction`**.
5. Click OK to define the job step. Data Extraction now appears as the last step in the root-level job step, after the extraction group that you added previously.

6. Add extraction steps to the new local extraction group by dragging them from the object gallery and dropping them onto that group. For the sample datamart, add the get fact data and get dimension data groups, in that order.

NOTE: *Fact extraction steps are placed before dimension extraction steps in your job in order to ensure consistency. The reason for this order is that the source database can change during extraction. If dimensions were extracted before facts, a new dimension element and a new fact that refers to that element could arrive between dimension and fact extraction. The datamart would then have a fact table entry that refers to a non-existent dimension element.*

If you are using streaming extraction or memory-mapped dimensions, see the Datamart Guide for information on proper extraction order. The sample datamart does not use streaming extraction or memory-mapped dimensions.

7. Expand the local extraction group and verify that the extraction groups were added in the correct order. If a step is in the wrong location, you can move the step by selecting it and clicking the Up or Down button.
8. Add any other necessary Extraction Steps to the root-level job step. For the sample datamart, right-click on the root level job and select New Group. Enter the name Run Semantics for the new group. Add the dimension semantics and fact semantics extraction groups, in that order, to the Run Semantics group.

NOTE: *Run fact semantics after dimension semantics. The reason for this order is that fact table rows have foreign keys for dimension table rows, so the dimension table rows need to be merged into the datamart before any fact table rows that refer to them are merged.*

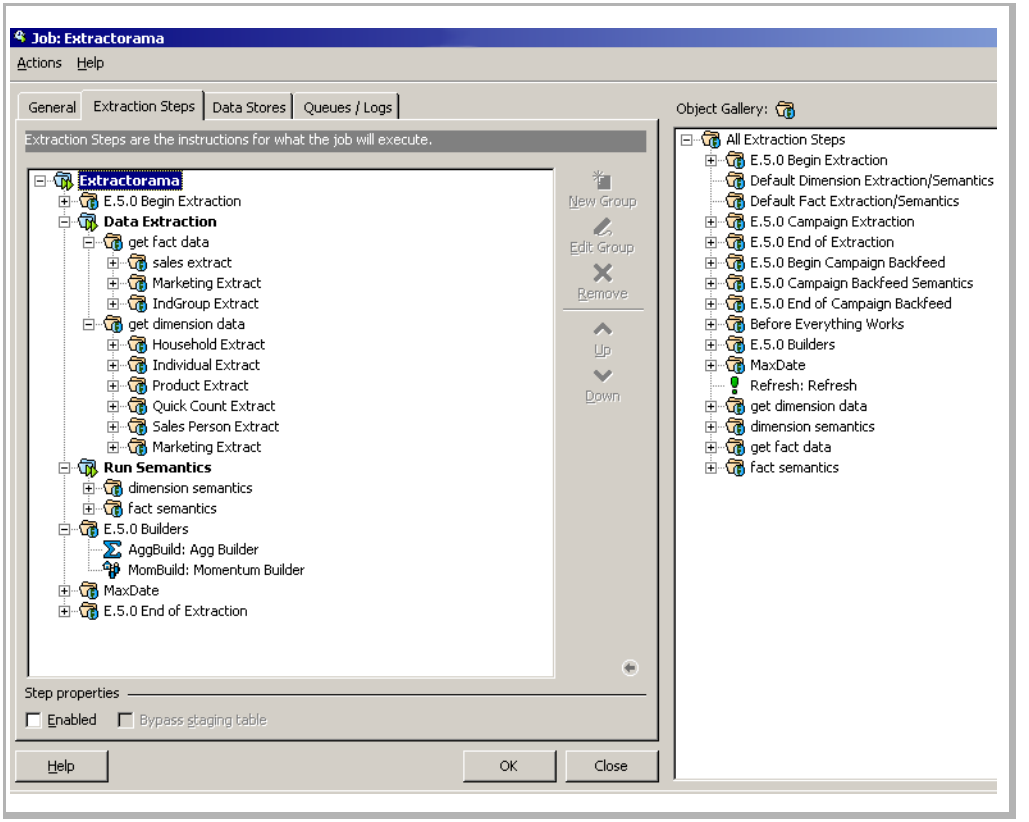
9. Add the E.5.0.1 Builders group, which contains the AggBuild: Agg Builder and MomBuild: Momentum Builder extraction steps, to your root-level job step.

   These extraction steps perform operations necessary for aggregate building, index building, list management, and campaign management. Aggregates are discussed in Chapter 9, "Building Aggregates." List management and campaign management are discussed in Chapter 8, "Creating Web Pages for Lists and Campaigns."

10. Add the pre-defined E.5.0.1 End of Extraction group to the root-level job step. This group contains several steps that are normally run at the end of an extraction job to finish the extraction process.
11. Click OK.

    For the sample datamart, the configured job appears as follows.

FIGURE 41: THE CONFIGURED SAMPLE JOB



Using the Extraction Wizard for Defining Extraction Commands

Analytic Manager also provides the **Extraction Wizard**, a graphical tool for defining extraction commands. For more information on the Extraction Wizard, refer to the *Datamart Guide*.

Importing External Tables

Before you use the graphical extraction command builder, you have to import the source tables into your datamart as external tables. Importing source-system tables as external tables makes the tables available to the graphical extraction command builder, it does not import any of the data in the tables.

Importing Source Tables as External Tables

1. In Analytic Manager, navigate to the Extraction folder . Right-click and select Import External Table to start the Import External Tables wizard.
2. In the Import External Table Definition screen, select the data store from which you would like to import table definitions. For the sample datamart, use the Sample_Source data store. Click Next.
3. On the second screen, select the tables that you want to import. Use the arrow buttons to move selected tables from the Tables/Views From Source System pane to the Tables/Views Selected as External Tables pane. Use the double arrow buttons to move all displayed tables and views. For the sample datamart, move all tables to the Tables/Views Selected as External Tables pane. Click Next.
4. On the third screen, you can edit the definitions of the external tables.For the sample datamart, you do not have to make any changes to the definitions. Click Next.
5. The fifth screen shows the definitions that will be added to metadata. Click Finish to add these definitions.

DEFINING AN EXTRACTION COMMAND USING THE EXTRACTION WIZARD

1. Double-click and open the Extractorama job. The Job dialog box appears.
2. Click the Extraction Steps tab.
3. In the Object Gallery, expand the All Extraction Steps folder. Expand the extraction group to which you want to add a new extraction command (for the sample datamart, this is Household Extract). For the sample datamart, delete the Get Household Data extraction step, which you will be replacing.
4. Right-click the extraction group to which you want to add a new extraction command (for the sample datamart, this is Household Extract). Right-click and and from the New menu, select Extraction Command. Analytic Manager displays the Extraction Wizard.
5. Click Next.
6. Select a type of table to populate. For the sample datamart, select Populate a dimension table. Click Next.
7. Select the destination table to populate. For the sample datamart, select Household_ base. Click Next.
8. Select the data store from which you populate this table. For the sample datamart, select sample_source. Click Next.
9. Select the source-system tables that you want to use to populate the selected EpiCenter datamart table. All external tables in the selected data store are available. For the sample datamart, select household_table. Click Next.
10. Enter a name for the extraction command. For the sample datamart, enter the name **`Get Household Data`**. Click Next.
11. The Extraction Wizard displays a summary of the extraction command for your review.

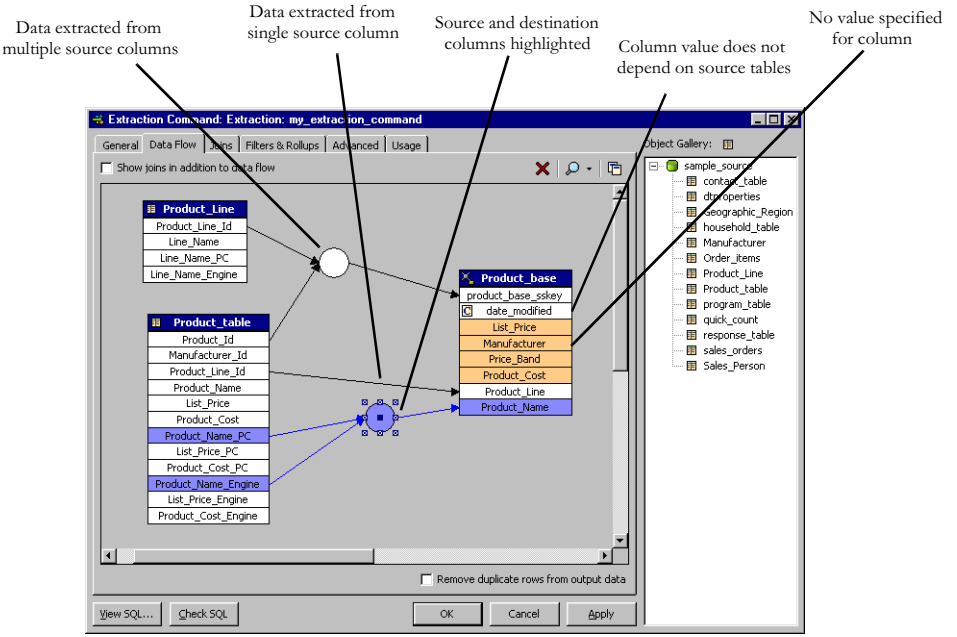
    Select the checkbox Open this Extraction Command in the Extraction Command dialog. This directs the Extraction Wizard to return to the Extraction Command dialog box, and you can edit your extraction command further.
12. Click Finish to define the extraction command.

DEFINING DATA FLOWS

When you use the Extraction Wizard to define extraction steps, Analytic Manager defines some of the **data flow** and **joins** based on column names and foreign keys in the selected source-system tables. You must define the remaining data flows and joins.

The Data Flow tab of the Extraction Command dialog box enables you to configure how data is copied from source-system tables to the destination tables. Arrows illustrate how data is copied from source-table columns to destination columns (See Figure 42).

FIGURE 42: EXTRACTION COMMAND DIALOG BOX: DATA FLOW TAB

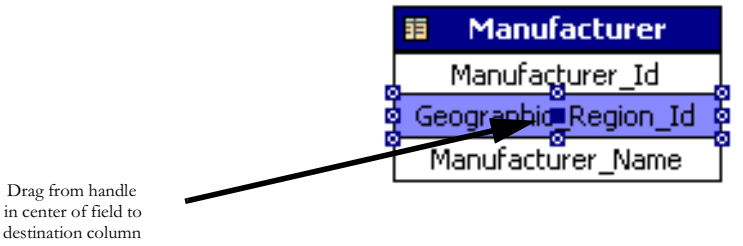


SPECIFYING A NEW DATA FLOW

1. In the Extraction Command dialog box, select the Data Flow tab. Select the name of a source-table column. For the sample datamart, select Household_id.

   When you select a column in this way, the column is highlighted and a dot appears in the middle. (See Figure 43.)

FIGURE 43: SELECTED SOURCE COLUMN



2. Click on the dot in the middle of the column and drag to the desired column of the destination table. For the sample datamart, drag to the household_base_sskey column.

   When you do this, an arrow appears from the source to the destination. To delete a data flow, right-click the arrow and select Delete.

   If you created an ID destination column, select the Household_id source column and map a data flow to the ID destination column.

3. Specify any additional data flows needed. For the sample datamart, map the the Company_name column to the Organization column and the Geographic_region_id column to the Region column.

To define a destination -column value that does not depend on your source data (such as a constant or the current date):

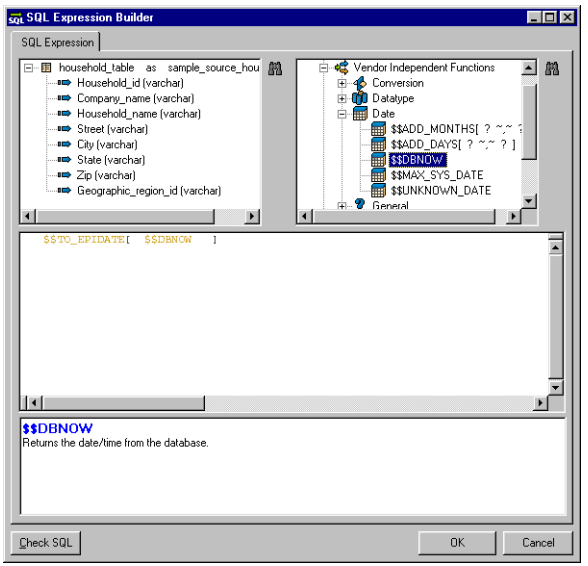
1. Double-click the destination column. For the sample datamart, double-click date_modified. Analytic Manager displays the SQL Expression Builder dialog box. (See Figure 44.)

2. You can use the upper left pane to add columns to the expression and the upper right pane to add vendor-independent or vendor-specific functions to the expression. The expression `$$TO_EPIDATE[ $$DBNOW ]` should already be entered in the Write SQL Expression pane for the date_modified column. For the sample datamart, you do not have to modify this expression.
3. Click the Check SQL button to verify that the SQL expression is correct. Analytic Manager displays the Expression check window.

NOTE: *In order to ensure correctness, SQL syntax is verified using the source database engine. Therefore, you must be connected to the source data store in order to check the syntax of your extraction command.*

4. Click OK to close the Expression check window, and click OK to close the SQL Expression Builder dialog box. Analytic Manager displays the letter C in the date_modified column.

FIGURE 44: SQL EXPRESSION BUILDER DIALOG BOX



5. Check Remove Duplicate Rows from Output Data. This will remove duplicate rows from your output data by performing a SELECT DISTINCT operation.
6. Select the Advanced tab. In the On error list box, select Inherit.
7. Click OK to finish defining the extraction command.

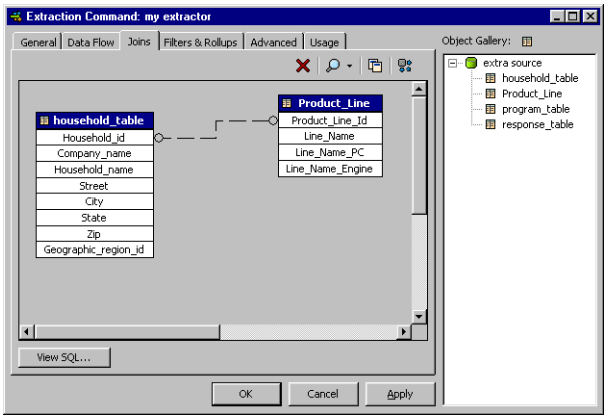
DEFINING JOINS

The Joins tab of the Extraction Command dialog box (see Figure 45) allows you to configure how source-system tables are related when extracting data. A dashed line between two source-table columns indicates that the values in these columns must be equal.

DEFINING A JOIN

1. Open the Extraction Command dialog box for the extraction command for which you define a join. For the sample datamart, this is Get Household Data. Select the Joins tab. Analytic Manager displays the source tables that have been selected for the extraction command.

FIGURE 45: EXTRACTION COMMAND DIALOG BOX: JOINS TAB



2. To create a join between two columns, select one of the columns, click on the dot in the middle of the column and drag to the second column. Analytic Manager displays a dashed line between the two columns. For the sample datamart, a join should already be defined between the Geographic_Region_Id column of the Geographic_Region table and the Geographic_region_id column of the household_table table.
3. Click the View SQL button to view the SQL code. Analytic Manager displays the SQL Statement check window. Verify that the SQL statement is correct, and click OK to close the SQL Statement check window.

Defining Filters and Rollups

The Filters & Rollups tab of the Extraction Command dialog box allows you to set criteria that limit the data that is extracted from the source system and roll up results along selected columns.

To add a new filter or rollup, Click the New button next to the appropriate pane in the Filters & Rollups tab and then use the SQL Expression Builder to construct an appropriate expression. For the sample datamart, you do not have to add filters or rollups to the `Get Household Data` extraction command.

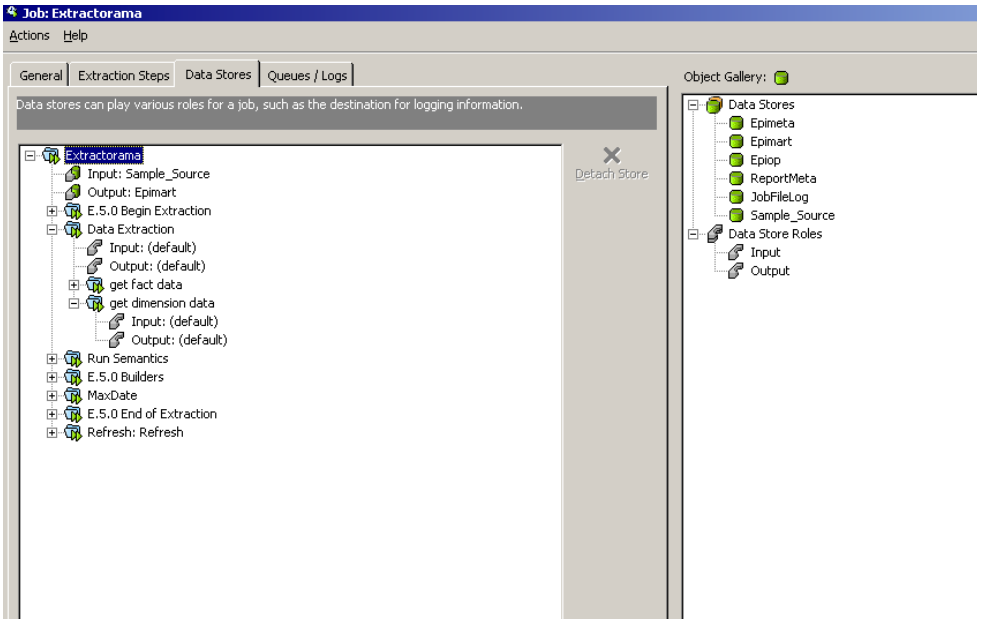
Assigning Data Stores

You now have to specify the databases that are used by the job. You specify these databases by assigning data stores to the job's data store roles. Each job step has its own collection of data store roles. If a job step does not have data stores assigned to some of these roles, then it makes use of the data stores assigned to those roles in its parent job step. The root-level job step must have data stores assigned to all data store roles.

ASSIGNING DATA STORES TO DATA STORE ROLES

1. Open the Job dialog box for the **Extractorama** job, and click the **Data Stores** tab. Here you can see all available data stores and data store roles in the right pane and data store role assignments for this job in the left pane. The **Data Stores** tab with several of the extraction steps and extraction groups expanded appears as follows.

FIGURE 46: THE DATA STORES TAB OF THE JOB DIALOG BOX

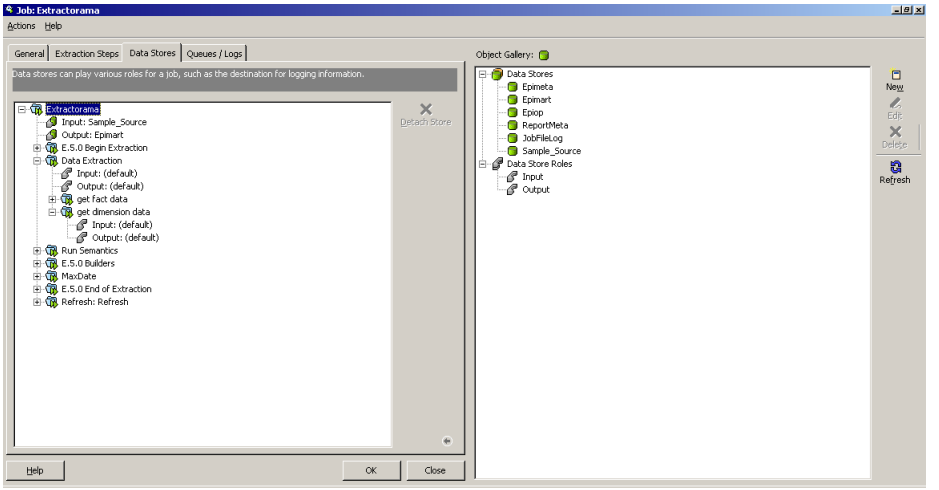


2. Drag an **Input Data Store** icon from the Object Gallery to the **Input Data Store Role** icon under the root-level job step. For the sample datamart, drag the **Sample_Source** icon to the **Input** icon just below the name **Extractorama** in the left pane. Analytic Manager now displays the Data Store icon attached to the Data Store Role icon. (To detach a data store, right-click the icon and select **detach**.)

3. Add the Output Data Store to the root-level Output Data Store Role. In a simple extraction job, the output is usually the EpiMart data store. For the sample datamart, drag the EpiMart icon to the Output Data Store Role.

   The Data Stores tab of the Job dialog box appears as follows.

FIGURE 47: THE DATA STORES TAB OF THE JOB DIALOG BOX WITH DATA STORES ASSIGNED



4. If you are not using the sample datamart, add further data stores to other Extraction Steps, as needed. For the sample datamart, the root-level data stores are sufficient.
5. Click OK to finish defining the job.

RUNNING THE EXTRACTION JOB

Prior to running your extraction job, disable the **MomentumEnabled flag**.This flag enables list and campaign management features. The list and campaign management features in the datamart were not implemented. Therefore, disable the MomentumEnabled flag.

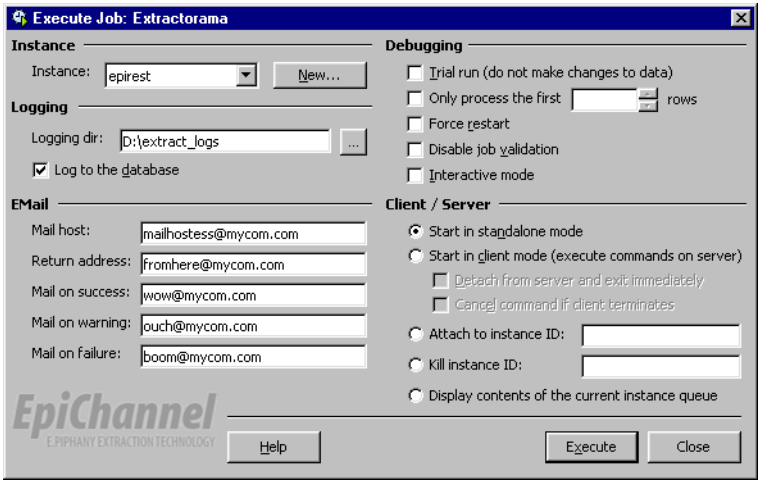
DISABLING THE MOMENTUMENABLED FLAG

1. In Analytic Manager, navigate to the Configuration folder.
2. Double-click the Settings subfolder. The Configuration dialog box appears.
3. Scroll down to the momentum_enabled key, click in the value column, and set the value to zero.
4. Click OK.

RUNNING THE EXTRACTION JOB

1. In Analytic Manager, select the Extractorama extraction job.
2. Right-click and select Execute. The Execute Job dialog box appears.

FIGURE 48: THE EXECUTE JOB DIALOG BOX



3. Make sure your EpiCenter instance is selected in the Instance list box.

NOTE: *If the list box is blank, check to make sure that your database server and your Analytic Manager are both set up to use the same authentication mode.*

4. Check Trial Run, and click Execute, to do a practice job execution run. (Analytic Manager tests the process, but makes no changes at this time.)

   When you click Execute, a console window appears and details of job progress are displayed in the console window. Detailed error messages are displayed if any errors are encountered.

   If any errors are reported, look at the log files to see what went wrong, and make the necessary corrections. In particular, be sure to correct any errors marked as `Fatal Exception`s.
5. Now right-click your extraction job and select Execute to re-display the Execute Job dialog box.
6. Uncheck Trial Run, then click Execute. Analytic Manager runs the job. The console window appears again, and you can see all of the extraction steps that are being taken.
7. If the job again ends with a return code of `0`, you have successfully populated your EpiMart.

NOTE: *To give users access to the data in your datamart, you have to create a web page. To create a web page, refer to Chapter 5, "Creating a Simple Web Page."*

TROUBLESHOOTING A FAILED EXTRACTION JOB

If your extraction job fails to complete successfully, refer to the general troubleshooting strategies in this section. For detailed troubleshooting instructions, refer to chapter 10, "Analytic Manager Utilities" of *Datamart Guide*.

RUN SCRUTINY

Scrutiny is a debugging utility that checks the metadata for consistency and suggests solutions for any problems that it finds. See "Run Scrutiny," on page 107 for detailed instructions. When Scrutiny finds errors, it describes the nature of the error and offers suggestions on how to correct the errors using Analytic Manager. Scrutiny also offers to correct the error for you.

E.piphany recommends that you follow the recommendation of fixing the problem yourself rather than having Scrutiny correct it for you. Make sure that you read the descriptions carefully before applying a Scrutiny fix. For additional information, refer to chapter 10, "Analytic Manager Utilities" of *Datamart Guide*.

After making the corrections that Scrutiny indicates, regenerate the schema . For more information, refer to "Creating a Datamart," on page 39. If you make changes to the schema and regenerate it, make sure that you update your SQL statements. After making any necessary corrections, run Scrutiny again to ensure that there are no remaining metadata errors.

EXAMINE THE EXTRACTION LOGS

If your EpiCenter passes the Scrutiny checks, then you look at the extraction log files. EpiChannel generates detailed logs for every extraction job. The console output for a failed extraction log ends with a line like the following:

```
**************Fatal Exception @ 11/7/00 5:59:03 PM**************
```

The above line is followed by a detailed description of the error.

The log files for the most recent extraction job can be found in the **Latest** directory of the log directory that you specify when you initialize your EpiCenter. This directory contains a file called **epiChannel_summary.log**, which contains the console output for that job.

A sample message that you may find in the log file is as follows:

- `Expected column "my_column" is missing from the extraction statement. Ensure that your extraction statement selects data for all columns in the * table you are attempting to populate.`

  This message indicates that the failed extraction step did not assign values to some column in one of the tables of your star schema. The error message indicates the destination column and the failed extraction step. Check the SQL code or data flow for the failed extraction step.

  Make sure that you have assigned values to all columns that are shown in the template or the graphical extraction command builder. If you are writing SQL code by hand, make sure that all column names are spelled correctly. for the sample datamart, make sure that your SQL code matches that in Appendix A.

- `A database error was encountered while attempting to execute the extraction sql.`

  This message most commonly indicates a SQL syntax error in the extraction code. The message is followed by additional details about the database error. In the case of a syntax error, the possible location of the error is indicated with the message "`Incorrect syntax near`...". Check the SQL code in the failed extraction step. If the extraction step uses E.piphany macros, click the Show Translation button in the SQL Statement dialog box to see the translation. Make sure that the translation is syntactically correct for the target database system.You can find more detailed logs in the file **epiChannel_detailed.log**, which can be found in the same directory as the **epiChannel_summary.log** file. After you make the corrections, run the extraction job again.

Creating a Simple Web Page

This chapter describes how to create a simple **Rows and Columns** web page. The Rows and Columns web page allows you to browse the datamart and display reports in tabular form. The web page provides easy access to data, simple navigation, drill-down reports, graphing, and filtering.

You create a simple Rows and Columns web page by doing the following:

- Creating Attributes
- Creating Attribute Layouts
- Creating Measures
- Creating Measure Layouts
- Defining the Initial Web Page
- Creating a New Topic
- Adding Users to web pages
- Running Scrutiny on Metadata

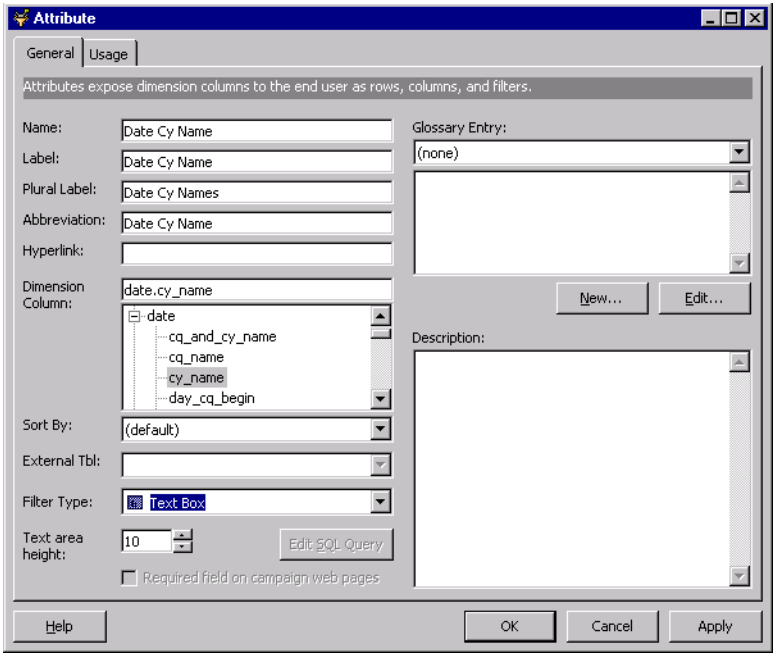
Creating Attributes

Attributes define how dimension-table columns are used in web pages. A Rows and Columns web page allows you to display the value of a measure based on two dimension-column values. These two values are chosen from lists that you define when you set up the web page. For example, you can display sales figures in a table with the customer's income in the columns and the customer's age in the rows. Income and age are both attributes of the customer dimension.

CREATING AN ATTRIBUTE

1. In Analytic Manager, expand the Presentation folder and select the Attributes subfolder.
2. Right-click Attributes, and select New Attribute. The Attribute dialog box appears.

FIGURE 49: THE ATTRIBUTE DIALOG BOX



3. In the Dimension Column field, expand the folder for the dimension role from which you want to select an attribute. For the sample datamart, expand the date dimension role.
4. Select the dimension column for which you want to create an attribute. For the sample datamart, select the cy_name (calendar year name) column.

5. Based on your selection, the Name, Label, Plural Label, and Abbreviation fields are filled in automatically.
6. Make any other changes you want to the Name, Label, Plural Label, and Abbreviation fields. Change the default name Date Cy Name to something more descriptive such as `Year`. When you change the attribute name, the other fields are also updated.
7. From the Filter Type list box, select Text Box filter type.

   Attributes can be configured as filters that you can apply to the data. When you use an attribute to only select a dimension column, the filter type is ignored. For more information on Filters, refer to Chapter 6.
8. Click OK.
9. Repeat steps 2-8 to add additional attributes. Set the Filter Type value of all of the attributes you create to Text Box.
10. For a complete list of the attributes in the sample datamart, refer to Table 9 on page 259.

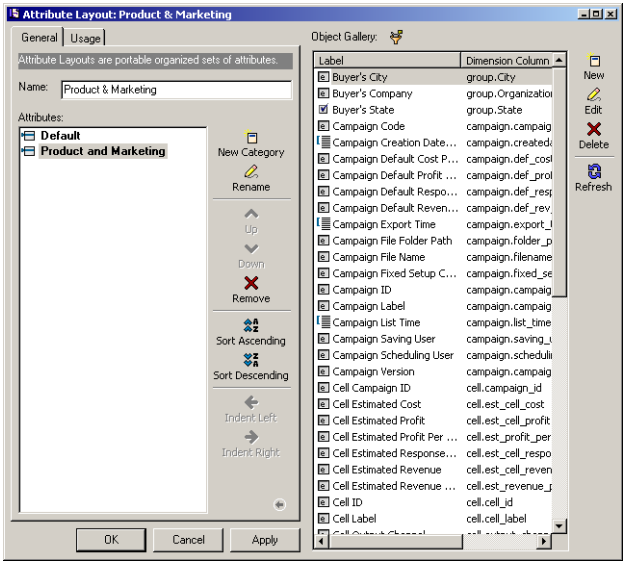
CREATING ATTRIBUTE LAYOUTS

Attribute layouts enable you to package attributes that are used together. You can use an attribute layout in any portion of a web page that calls for an attribute. The use of attribute layouts is optional.

CREATING AN ATTRIBUTE LAYOUT

1. In Analytic Manager, navigate to the Presentation folder and select the Attribute Layouts subfolder.
2. Right-click and select New Attribute Layout. The Attribute Layout dialog box appears.

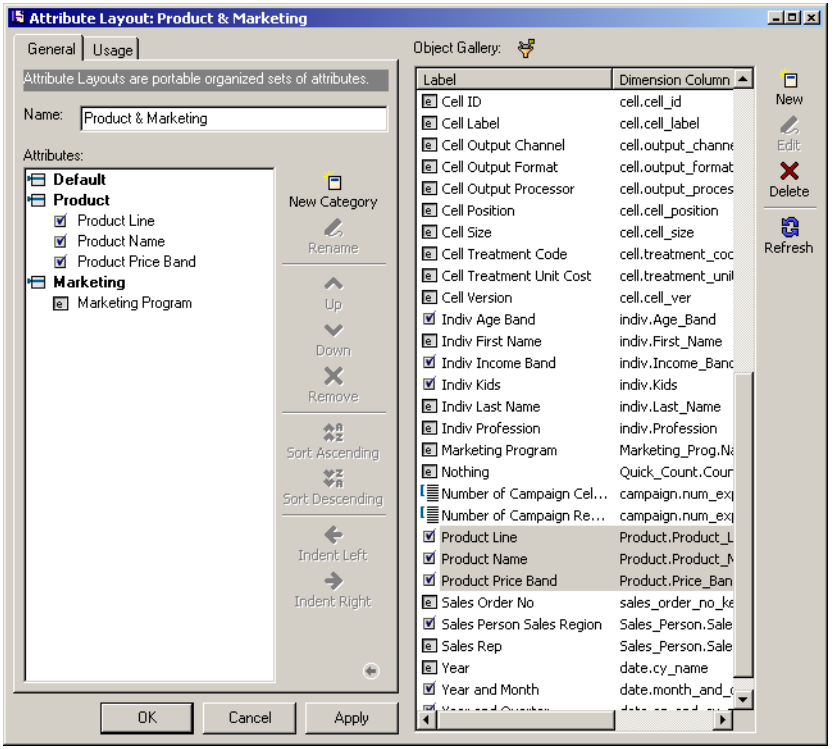
FIGURE 50: ATTRIBUTE LAYOUT DIALOG BOX



3. In the Name text box, enter a name for the attribute layout. For the sample datamart, enter **`Product & Marketing`**.
4. Click the New Category button that appears to the right of the Attributes field. For the sample datamart, enter another category called **`Marketing`**.
5. Change the name of the `Default` category to Product. Use the Rename button to change the name of a category.
6. To add an attribute to the attribute layout, drag the attribute from the Object Gallery pane and drop it on the category in which it should appear.

For the sample datamart, add all of the product attributes from the product dimension role to the product category. Add the Marketing Program attribute to the Marketing category.

FIGURE 51: ATTRIBUTE LAYOUT: PRODUCT & MARKETING



NOTE: *Use the* **Up** *and* **Down** *buttons to move an attribute up or down within a category, or move an entire category up or down with respect to other categories.*

7. Click **OK**.
8. Follow steps 2-7 to create **`Indiv Buyer`**, **`Group Buyer`**, **`Date`**, and **`Sales attribute`** layouts. For the completed layouts, Refer to Appendix A, Figures 136 through 139.

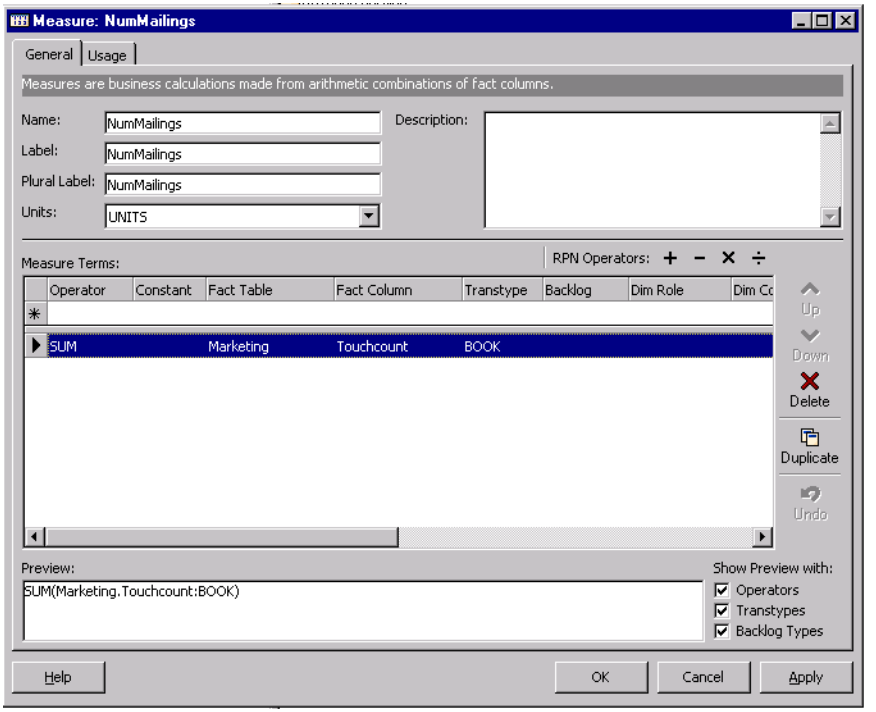
CREATING MEASURES

Measures display information from fact tables in your web page. A measure is an operation (such as a sum) performed along a column of a fact table. For example, you can define a measure for number of mailings. This measure adds up the touch_count booked columns of the chosen rows of the fact table.

CREATING A MEASURE

1. In Analytic Manager, expand the Measures folder.
2. Right-click the Measures subfolder and select New Measure. The Measure dialog box appears.

FIGURE 52: THE MEASURE DIALOG BOX



3. Enter a name and description for the new measure. For the sample datamart, enter the name **NumMailings**. This measure starts with a measure that counts the number of mailings sent.
4. Specify the units to be measured. For the NumMailings measure, mailings are sent in discrete units, therefore, select UNITS.
5. In the Measure Terms grid, click in the first row of the Operator column. A drop-down list box appears.
6. Select an Operator. For the sample datamart, select SUM.
7. Click the first row of the Fact Table column to display the drop-down list box. For the sample datamart, select the Marketing table, which contains the Touch_Count column.
8. Select the Fact Column to display the drop-down list box. For the sample datamart, select the Touch_Count column.
9. Click the Transtype column to display the drop-down list box. The sample datamart has mailings recorded as BOOK transactions and responses recorded as SHIP transactions. The NumMailings measure counts mailings sent, therefore, select the BOOK transaction type.
10. If you are not using the sample datamart, add any other measure terms that you require. Press Enter to refresh the grid in order to add a new measure term. No additional terms are needed for the NumMailings measure.
11. Click OK.
12. For the sample datamart, define another measure called **NumResponses**. Use the NumMailings values except for transtype. Set the transtype to SHIP.

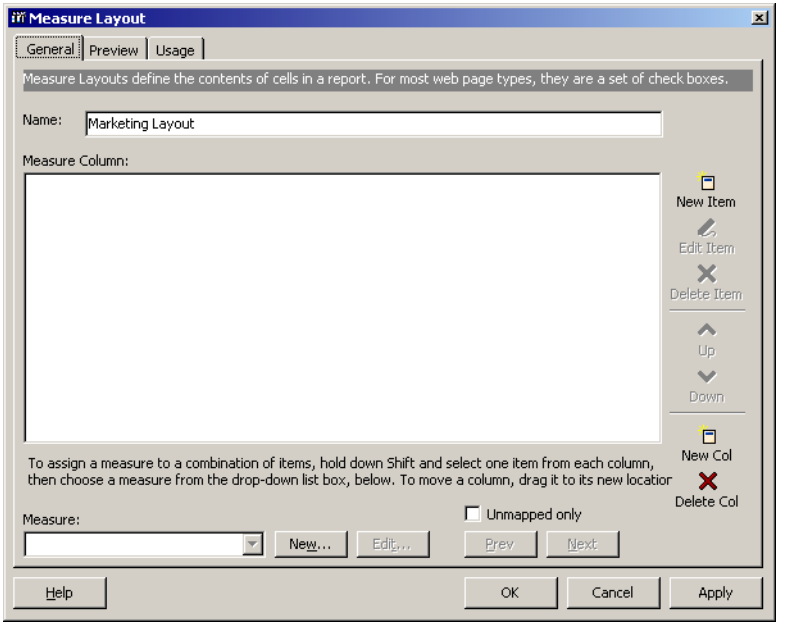
CREATING MEASURE LAYOUTS

Attributes specify how information is classified when it is displayed on a web page. The actual information displayed on the web page is calculated by a measure. A **measure layout** associates a measure with the web page.

DEFINING A MEASURE LAYOUT

1. In Analytic Manager, expand the **Measures** folder and select the **Measure Layouts** subfolder.
2. Right-click **Measure Layouts** and select **New Measure Layout**. The Measure Layout dialog box appears.

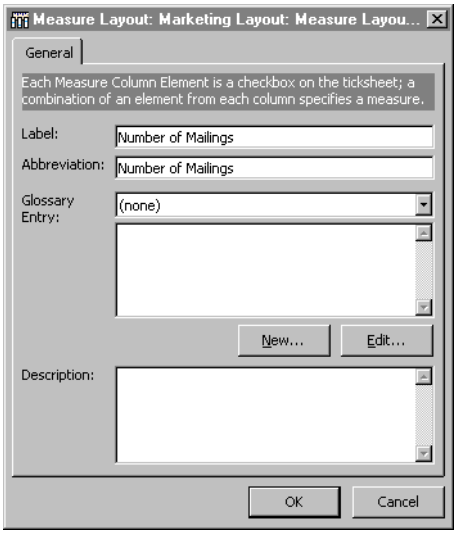
FIGURE 53: THE MEASURE LAYOUT DIALOG BOX



3. In the **Name** field, enter a name for the measure layout. For the sample datamart, enter the name **`Marketing Layout`**.

4. Click the New Item icon. The Measure Layout Column Item dialog box appears.

Figure 54: The Measure Layout Column Item



5. Enter a label for the measure column element. For the sample datamart, enter the label **`Number of Mailings`**. The description field is automatically populated.
6. For Glossary Entry, click the drop-down list box and select the entry you want.
7. To create a new Glossary entry, click New and enter the text for Glossary Entry Name, label and the Help text (optional).
8. Click OK to add the element to the measure columns list.
9. Select the name of the measure column element that you defined.
10. In the Measure list box, select a corresponding measure for that column element. For the sample datamart, select the NumMailings measure.
11. Click Apply. The specified measure is assigned to the selected measure column element.

12. Repeat this process for other measures that you defined. In the sample datamart, assign the NumResponses measure to a measure column element called Number of Responses.
13. Select the Preview tab in the Measure Layout dialog box to see how this measure layout is displayed on a web page.
14. Click OK.

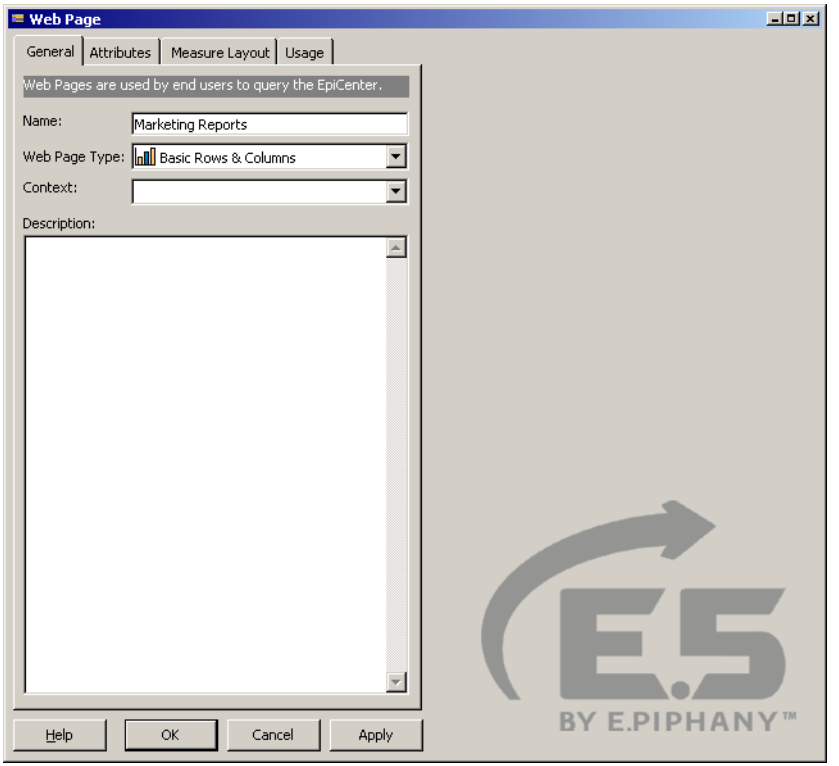
DEFINING THE INITIAL WEB PAGE

After you create attributes, attribute layouts, measures and measure layouts, define the initial web page.

DEFINING THE INITIAL WEB PAGE

1. In Analytic Manager, expand the **Presentation** folder and select the **Web Pages** subfolder.
2. Right-click **Web Pages** and select **New Web Page**. The Web Page dialog box appears.

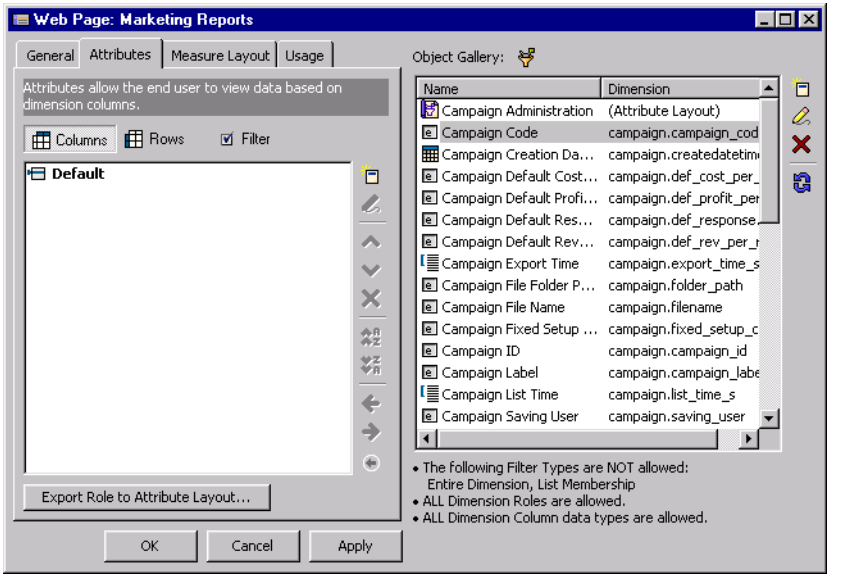
FIGURE 55: THE WEB PAGE DIALOG BOX: GENERAL TAB



3. Enter a name for the web page. The first web page in the sample datamart allows you to analyze marketing reports. Therefore, enter the name **`Marketing Reports.`**

4. In the Web Page Type list box, select Basic Rows & Columns.
5. To assign the attributes to the web page, click the Attributes tab. The Web Page dialog box opens. The attributes that are available for display in the columns of the web page appear in the Object Gallery.

FIGURE 56: THE WEB PAGE DIALOG BOX: ATTRIBUTES TAB



NOTE: *On a new web page, no attributes are displayed.*

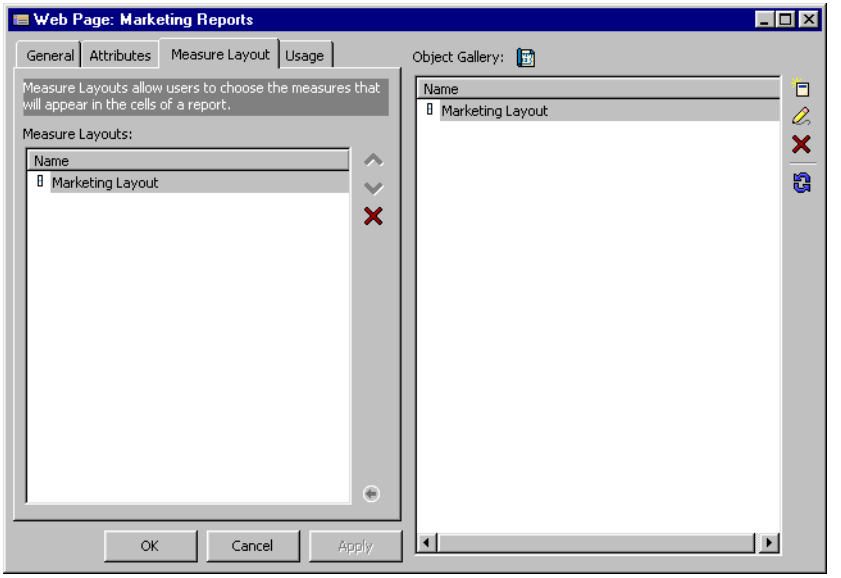
6. Add an attribute or attribute layout by selecting it in the Object Gallery and dragging it to the left pane of the Web Page dialog box. When you add an attribute, the attribute appears under the Default icon in the left pane.
7. For the sample datamart, select the Product & Marketing attribute layout that you created.

8. Add additional attributes for the columns. For the sample datamart, add all of the attribute layouts that you created. You can also add all the attributes that you created but did not add to an attribute layout, including the Nothing attribute.

NOTE: *You don't have to add the* ***Campaign*** *or* ***Cell*** *attributes.*

9. Click Rows at the top of the left pane and add all attributes that you want to make available for display in the rows of the web page results screen. For the sample datamart, select all of the attributes and attribute layouts that you added for the Column attribute role except for the Nothing attribute.
10. To add the measure layout that you created to the web page:
    a. In the Web Page dialog box, click the Measure Layout tab.
    b. Drag a measure layout from the Object Gallery on the right to the Measure Layouts pane on the left. For the sample datamart, use the Marketing Layout layout that you created.

FIGURE 57: THE WEB PAGE DIALOG BOX: MEASURE LAYOUT TAB



11. Click OK. The Web Page dialog box closes.

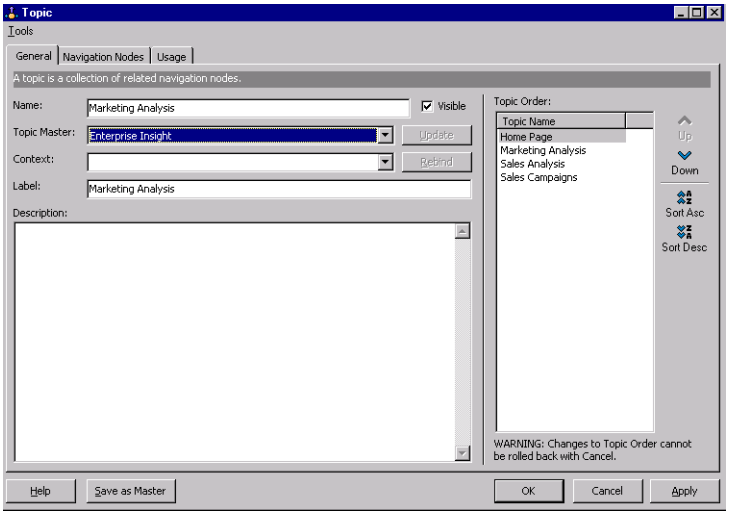
CREATING A NEW TOPIC

Topics enable you to group web pages into categories that make sense to users. Topics also provide navigation paths between web pages. For example, the sample datamart has Sales Analysis and Marketing Analysis topics. Only web pages that belong to a topic are accessible to users. Therefore, to use a web page that you create, you have to create a topic.

CREATE A NEW TOPIC

1. In Analytic Manager, expand the Presentation folder and select the Topics subfolder.
2. Right-click the Topics icon and select New Topic. The Topic dialog box appears.

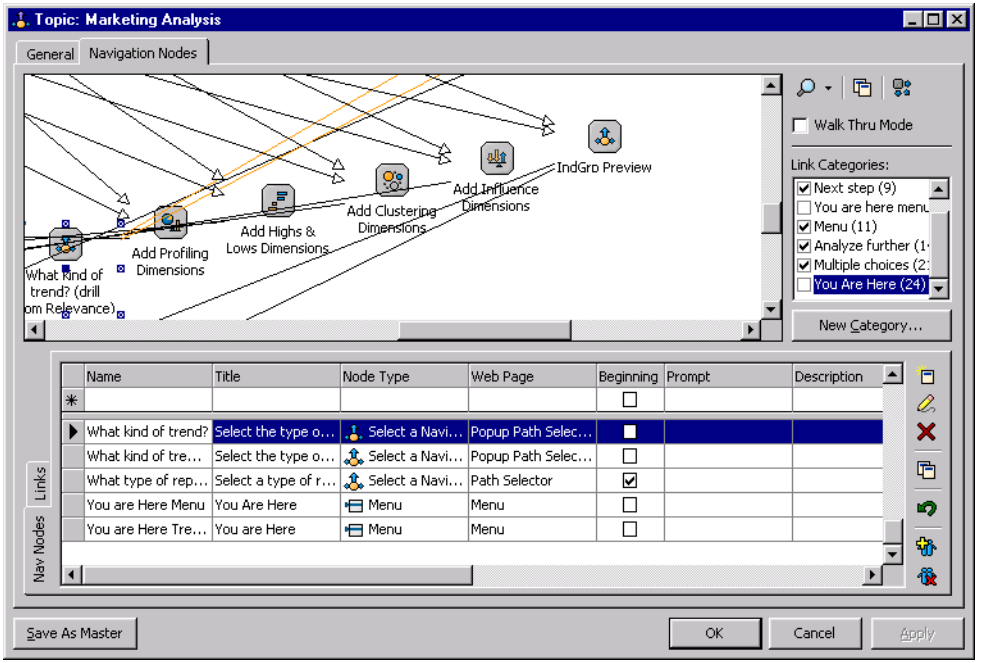
FIGURE 58: THE TOPIC DIALOG BOX: GENERAL TAB



3. In the Name field, enter a name for the topic. For the sample datamart, enter the name **`Marketing Analysis`**. When you enter the name, the Label field is filled in automatically. You can make changes to the label.
4. In the Topic Master field, use the drop-down list box to select the Enterprise Insight topic master. The topic master is a template that specifies the basic navigation structure of the topic that you create.
5. Click Apply to create the topic.

6. Select the Navigation Nodes tab to view and configure topic navigation. In this view you can see all of the navigation nodes in the topic. Use the scroll bars to view the nodes.

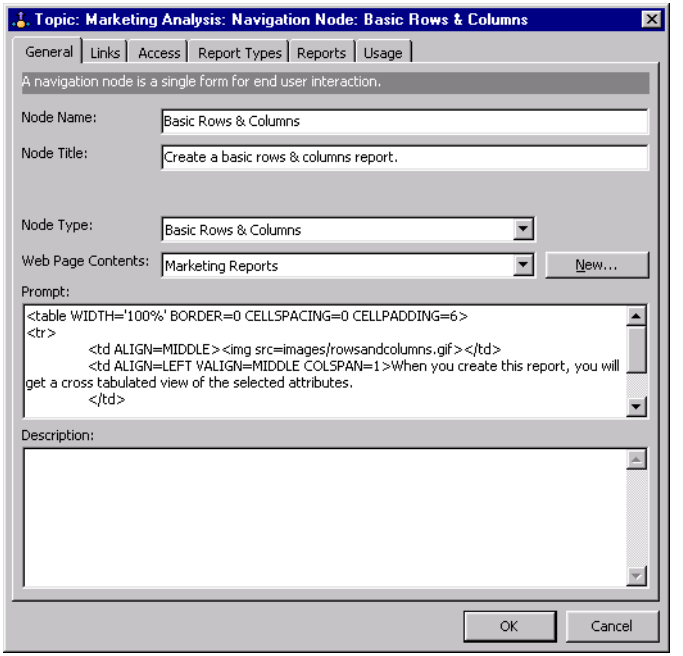
FIGURE 59: THE TOPIC DIALOG BOX: NAVIGATION NODES TAB



A navigation node is a potential destination that a user can visit. If an actual destination (a web page) has been assigned to a node, that destination is shown in the Web Page Contents column. Some nodes already have destinations assigned to them.

7. To edit a node, select the name of the node (for example, select Rows & Columns – Basic), right-click, and select Edit. The Navigation Node dialog box appears.

FIGURE 60: EDITING A NODE IN THE NAVIGATION NODE DIALOG BOX



8. Verify that the correct node type is selected in the Node Type list box. For the Rows & Columns node, Basic Rows & Columns should be selected.
9. From the Web Page Contents list box, select the Marketing Reports web page that you created.

NOTE: *The Prompt field contains HTML that formats the text displayed to the user at this node. You can edit the Prompt field. For the sample datamart, no changes are needed.*

10. Click OK to finish the basic setup for this node.

11. Click OK. The Navigation Node dialog box closes.

Adding Users to web pages

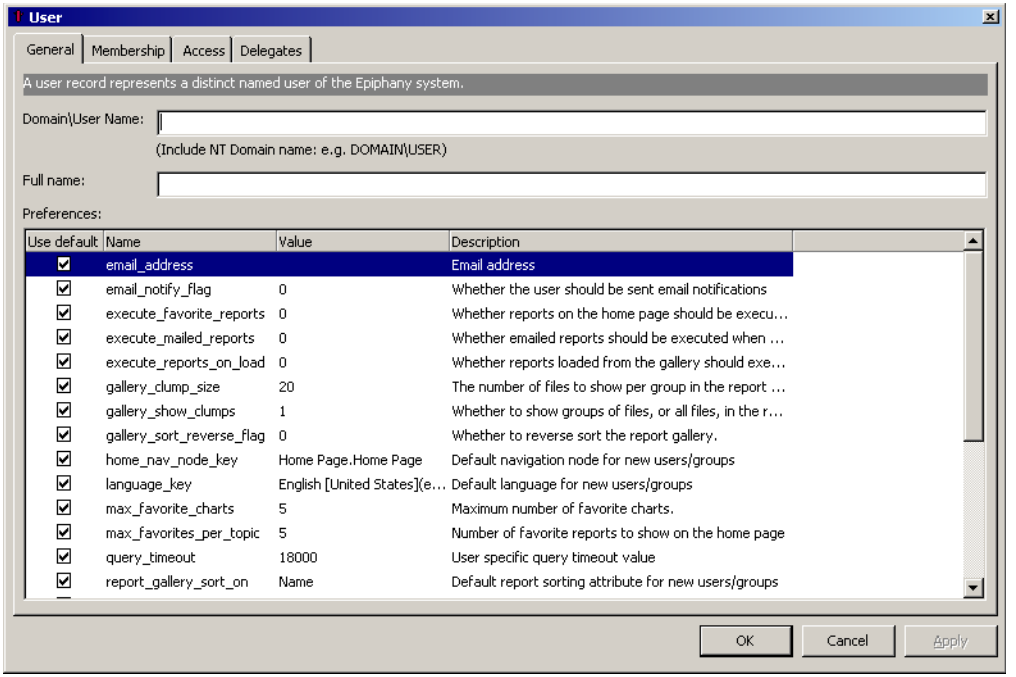
By default, users do not have access to a web page. Therefore, you have to add at least one user with access privileges.

Adding Users to a Web Page

1. In Analytic Manager, navigate to the Security/Storage folder and select the Users subfolder.
2. Right-click and select New User. The User dialog box appears.
3. In the User Name field, enter the user name.
4. In the Full Name field, enter the full name of the user.

5. Edit user Preferences as required. For the sample datamart, you do not have to make any changes to the preferences.

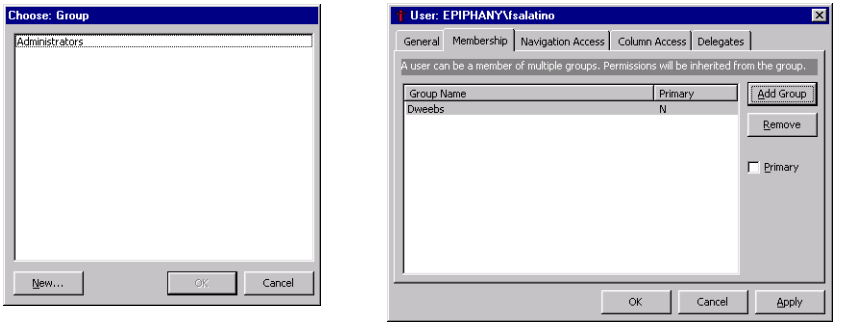
FIGURE 61: THE USER DIALOG BOX: GENERAL TAB



6. Click the Membership tab to assign the user to a group. Click Add. The Group dialog box appears (See Figure 62).
7. Select the name of a group. Use the pre-defined Administrators group for the sample datamart. You have to set up additional groups if users without administrator privileges will access the datamart.

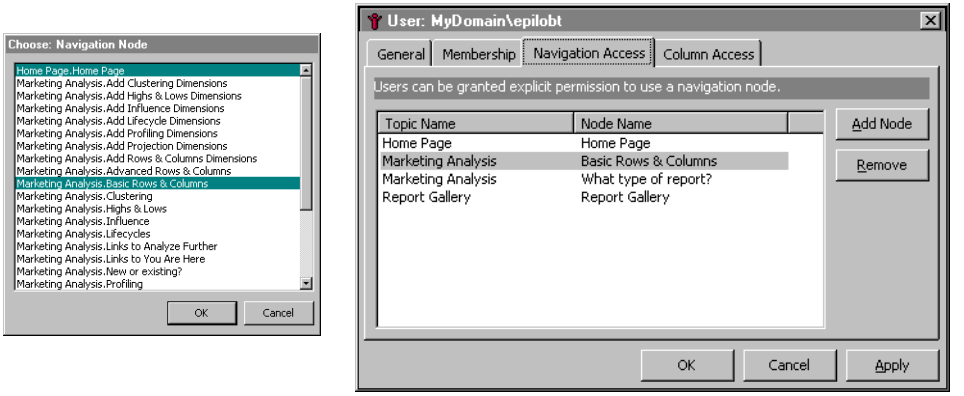
8. Click **OK**.

FIGURE 62: ADDING A USER TO A GROUP



9. Click the **Access** tab in the User dialog box.
10. Click **Add Node**. The Choose: Navigation Node dialog box appears. This dialog box lists all available topic nodes (See Figure 63).
11. Select the node to which you just assigned a web page, as well as all nodes that a user must traverse to reach that node. Hold down the **`Ctrl`** key to select more than one node.
12. Click **OK** to give the user access to those nodes.

FIGURE 63: GIVING A USER ACCESS TO A WEB PAGE



NOTE: *For every user, give access to the* Home Page *node of the* Home Page *topic because that node is the starting point for all users. Give every user access to the* Report Gallery *node of the* Report Gallery *topic.*

13. Give the user access to the What type of report? node and the Basic Rows & Columns node. Both these nodes are in the Marketing Analysis topic. The user reaches the Basic Rows & Columns node through the What type of report? node.
14. Click OK to finish setup for the user.

RUNNING SCRUTINY ON METADATA

Scrutiny is a tool that checks metadata for consistency. Inconsistencies in your metadata can prevent the Analytic Server from starting. The Analytic Server must be running for you to view your web page.

RUNNING SCRUTINY

1. In Analytic Manager, from the EpiCenter menu, select Run Scrutiny. The Scrutiny dialog box appears.
2. Make sure that Check EpiMeta is checked and Check EpiMart is not checked.
3. Check Show Warnings as Well as Errors.
4. Click Execute to run Scrutiny on your metadata.
5. When you run Scrutiny, a console window appears and the Scrutiny checks are displayed in it.
6. When you receive the prompt Press Enter to Continue, press Enter. This ensures that Scrutiny runs the complete set of tests.
7. When Scrutiny finds errors, it describes the nature of the error and offers suggestions on how you can correct the errors using Analytic Manager. Scrutiny also offers to correct the error for you.

Since the sample datamart is not yet complete, you may get messages from Scrutiny about inaccessible nodes, missing default reports, and queryability. You can ignore these messages for the time being.

You may also receive messages about other types of errors. E.piphany recommends that you follow the recommendation of fixing the problem yourself rather than having Scrutiny correct it for you. Be sure to read the descriptions carefully before applying a Scrutiny fix. Press **N** for No to decline Scrutiny's offer to correct the error for you. For more information on errors, refer to the *Datamart Guide*.

8. Continue with the responses until Scrutiny finishes running and the location of the log file is displayed in the console window.
9. Regenerate the schema after making the corrections that Scrutiny indicates. For instructions, refer to"Generating the Schema," on page 58 . If you regenerate the schema, you also have to run extraction again. Refer to "Extracting Data into the Datamart," on page 61. After making the necessary corrections, run Scrutiny again to ensure that there are no errors associated with your metadata.

STARTING THE ANALYTIC SERVER

The **Analytic Server** manages communications between the database server and the web server. When a user enters a request for data with one of E.piphany's web-based applications running in a web browser, the web server communicates that request to the Analytic Server. The Analytic Server then processes the request and issues an appropriate query to the database server. When the database server returns the results for that query, the Analytic Server renders them in an appropriate format for that application and passes them to the web server. The web server then transmits the updated web page to the user's web browser for display.

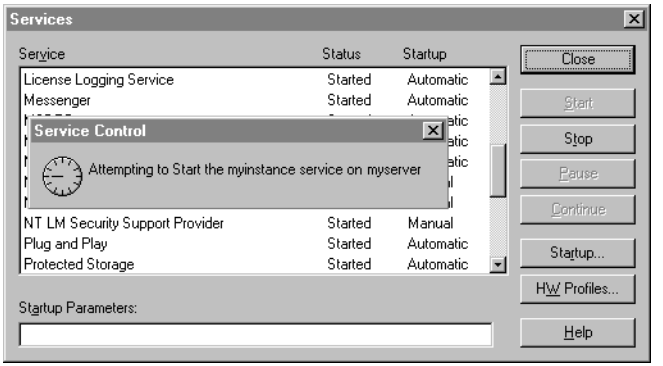
STARTING THE ANALYTIC SERVER

1. Go to the Windows Control Panel.

   Start>Settings>Control Panel

2. Double-click the Services icon. The Services dialog box appears.
3. Select the name of your EpiCenter instance from the list of services. Notice that it has a blank entry under Status and "Manual" under Startup.
4. Click the Start button to start the Analytic Server. Windows displays the Service Control panel with a message indicating that the service is attempting to start.

FIGURE 64: THE SERVICE CONTROL DIALOG BOX WITH THE SERVICE CONTROL PANEL



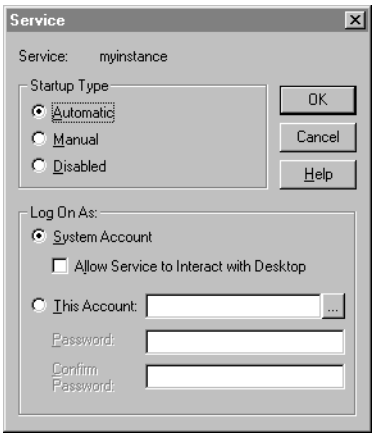
5. After a short wait, the Service Control dialog box disappears, and the Status column for your EpiCenter instance says "Started."

NOTE: *If there are errors in your datamart, you may get a dialog box saying that the service could not be started because of an internal Windows NT or Windows 2000 error. If you see this dialog box, review the logs to see what went wrong. The default installation places the Analytic Server log files in the following directory:*

C:\Program Files\Epiphany\instance\Web\WWWroot\logfiles

6. If you want to have your Analytic Server start automatically whenever you start Windows, you can configure it as follows:
    a. In the Service window, select your EpiCenter instance.
    b. Click the Startup button. Windows displays the Service dialog box.

FIGURE 65: THE SERVICE DIALOG BOX



    c. Specify the startup type to be Automatic.
    d. Click OK.
    e. Close the Service dialog box.
7. Close the Control Panel window.

VIEWING THE WEB PAGE

Once you create the web page, refresh the Analytic Server, and run Scrutiny, you can view the web page.

VIEWING THE WEB PAGE

1. Start your Analytic Server.
2. Start your web browser and point it to your EpiCenter instance. In the default setup, the URL takes the following form:

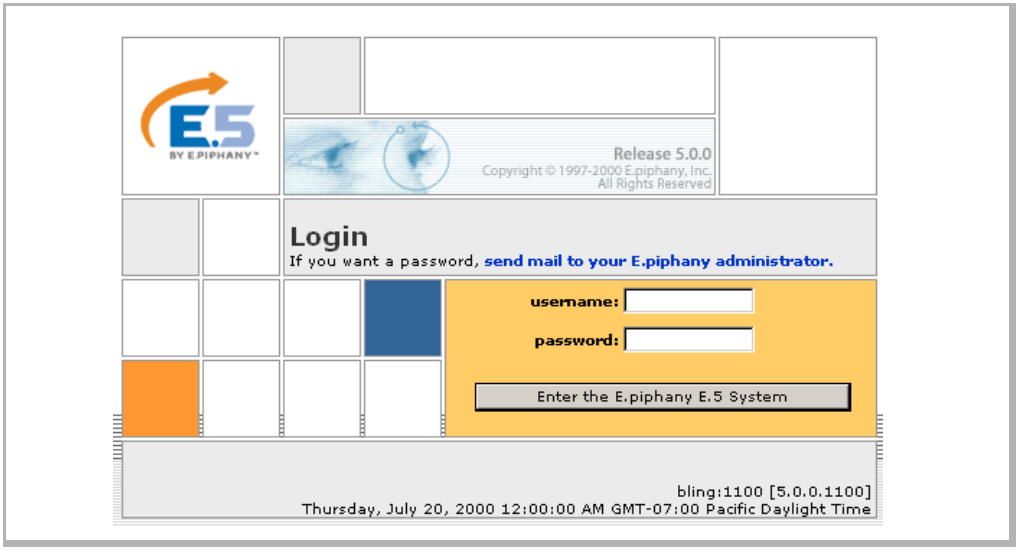
   http://myserver.mydomain/myinstance

   Replace *myserver.mydomain* with the full name of your host and *myinstance* with the name of your EpiCenter instance. For example, if the name of your host is epi_host, and the name of your domain is epiphany.com, and the name of your instance is epicenter_one, the URL would be:

   http://epi_host.epiphany.com/epicenter_one

   The URL takes you to the login screen.

FIGURE 66: THE LOGIN SCREEN



3. Enter the login and password for an authorized user. An authorized user is some one who was given permission to access the web page in E.piphany Analytic Manager.

4. Click Enter the E.piphany E.5 System to go to the home page. The home page lists all available topics. For the sample datamart, the topics that are available so far are Report Gallery and Marketing Analysis.
5. Click the name of your new topic to go to the starting page for the topic. In the sample datamart, click Marketing Analysis. See Figure 67 on page 134.
6. If you have created an Enterprise Insight topic, the default starting page is a path selector page. (See Figure 68 on page 135.) This page lists all available report types, but only those report types for which you have created web pages have active links.
7. Click the link for the web page that you created. In the sample datamart, click on the Rows & Columns link. (See Figure 68 on page 135.) .

FIGURE 67: THE HOME PAGE

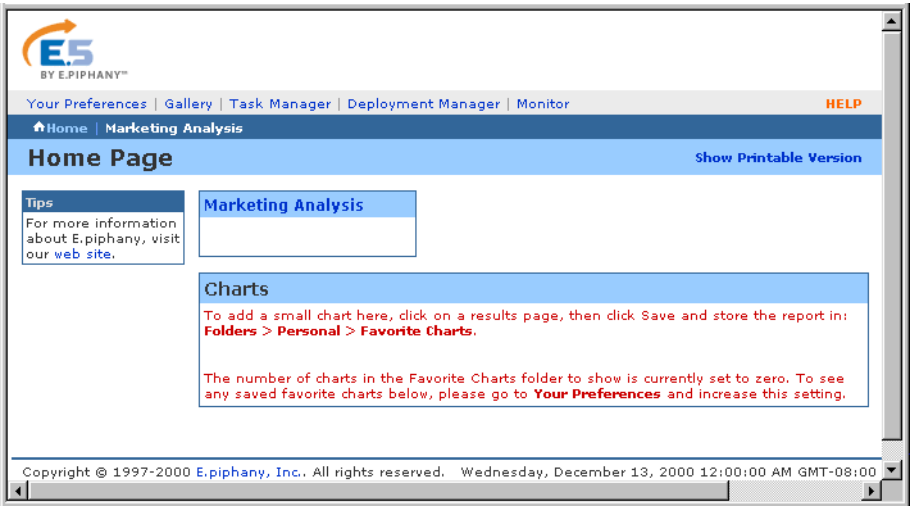
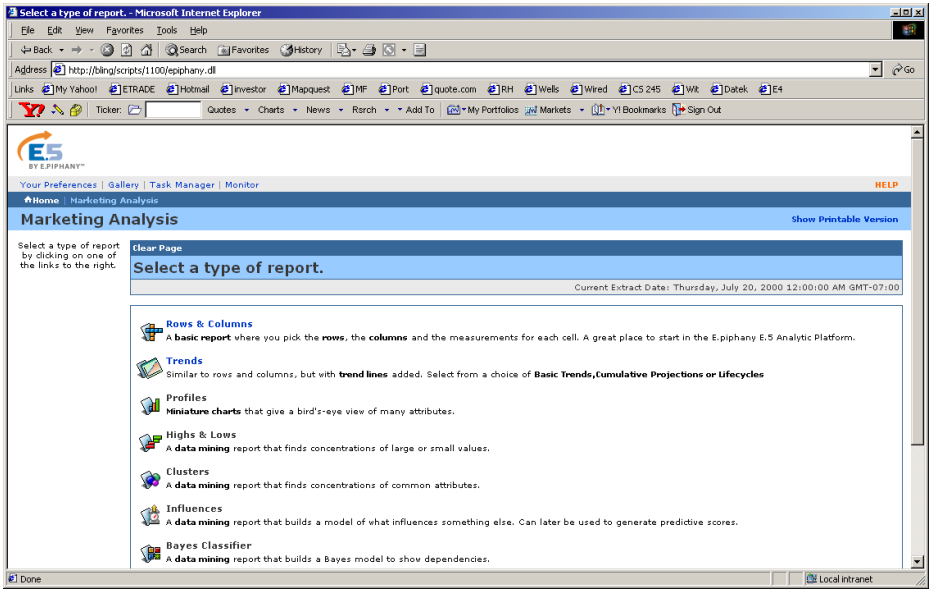


FIGURE 68: A PATH SELECTOR PAGE



8. Perform some sample queries on your EpiCenter. For online help, use the Help link on the upper side of the browser window.

Creating Rows and Columns Web Pages

This chapter describes how to create a **full-featured Rows and Columns** web page. Once you create a full-featured Rows and Columns web page, you can use it to create other web pages such as Profiling, High/Low clusters, Trends, Lifecycles, Cumulative Projections, Influences, and Community Clusters. These web pages are commonly used for reporting and Analysis.

You create a full-featured Rows and Columns web page by doing the following:

- Creating a New Web Page
- Creating Additional Measures
- Defining Complex Measure Layouts
- Adding Filters
- Configuring the Web Page
- Configuring a New Topic and Adding Security
- Adding a Default Saved Report

Creating a New Web Page

To create a new Rows and Columns web page, follow the instructions under "Defining the Initial Web Page," on page 118. For the sample datamart, call the new Rows and Columns web page Sales Reports. For the Sales Reports web page, define measures for booked revenue and shipped revenue. To define the measures, refer to "Creating Measures," on page 114.

Creating Additional Measures

A full-featured Rows and Columns web page can use measures that you have already defined or additional measures. Measures can also include operations (such as addition or subtraction) on measure terms. For the sample datamart, define a revenue backlog measure.

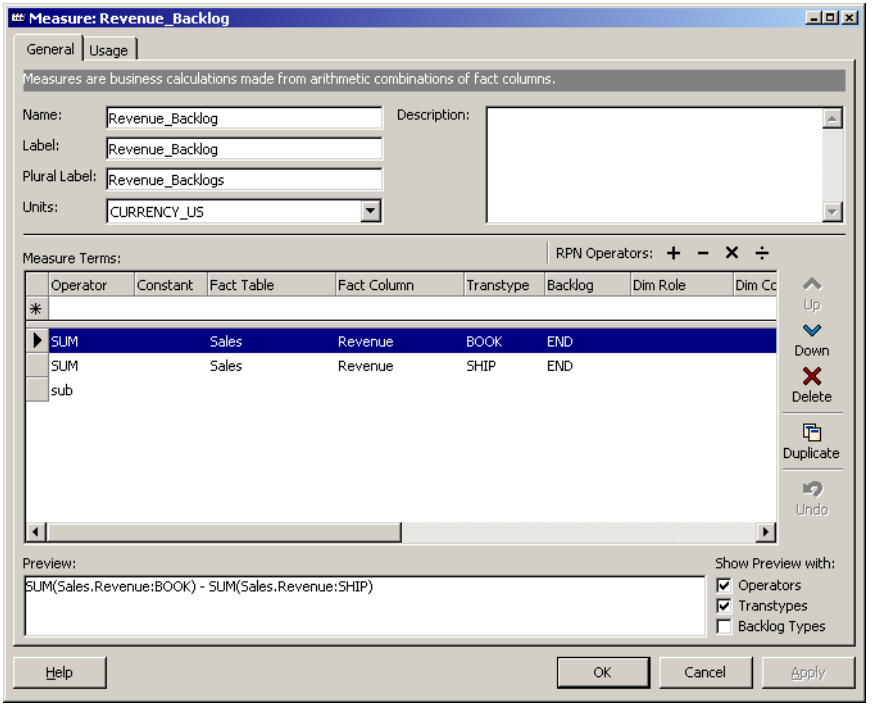
Backlog Measure

The **revenue backlog** is the difference between booked and shipped revenue. A **backlog measure** computes how values are accumulated over time, rather than for the current time period. For example, for a monthly calculation of revenue backlog, you compute the difference between all booked revenue through that month and all shipped revenue through that month and not just the difference between that month's booked and shipped revenue.

DEFINING AN ADDITIONAL MEASURE THAT CALCULATES THE DIFFERENCE OF BACKLOGS

1. In Analytic Manager, expand the Measures folder and select the Measure subfolder.
2. Right-click and select New Measure.The Measure dialog box appears.

FIGURE 69: A BACKLOG MEASURE



3. Enter the name and description. For the sample datamart, enter the name **`Revenue_Backlog`**.
4. For Units, select CURRENCY_US.
5. Define a measure term for the first backlog by using the following values:

Operator: SUM

Fact table: Sales

Fact column: Revenue

Transtype: BOOK

Backlog: End

NOTE: *BEGIN calculates the backlog as of the beginning of the time period and END calculates the backlog as of the end of the time period.*

6. Press Enter. The grid refreshes.
7. Add a second measure term by using the following values:

   Operator: SUM

   Fact table: Sales

   Fact column: Revenue

   Transtype: SHIP

   Backlog Type: End
8. Press Enter.
9. Click the – (minus) symbol above the list of measure terms to calculate the difference (the first term minus the second term).

   When you click this button, a third step with operator sub appears in the list of measure terms. The complete calculation appears in the Preview pane below the list of measure terms. Use the Operators, Transtypes, and Backlog types check boxes to change the amount of detail shown in the preview.
10. Click OK.

For additional information on calculations used to define complex measures, refer to the *Configuring Applications* guide.

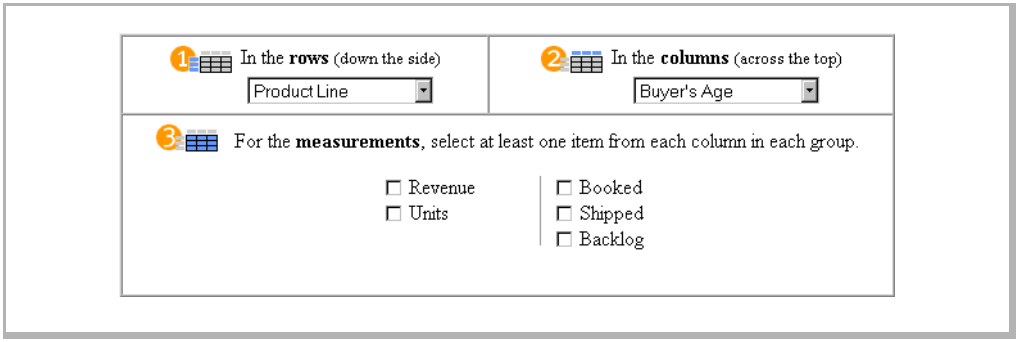
For the sample database, create additional measures for Units Booked, Units Shipped, and Unit Backlog. These are similar to the measures that you created for Revenue, except for qty column of the fact table and UNITS for the measure units. For more information, see Appendix A, "Sample Datamart."

DEFINING COMPLEX MEASURE LAYOUTS

You can configure a Rows and Columns web page with multiple columns of check boxes for measures. When you configure a web page in this way, you associate a separate measure with each combination of boxes that can be chosen from the columns.

For the sample datamart, you can have one column with Units and Revenue and another column with Booked, Shipped, and Backlog. You can then, for example, check Units and Revenue in the first column and Backlog in the second column. The resulting report would then show both Unit Backlog and Revenue Backlog with respect to the selected attributes.

FIGURE 70: MULTI-COLUMN CHECK BOXES ON A ROWS AND COLUMNS WEB PAGE



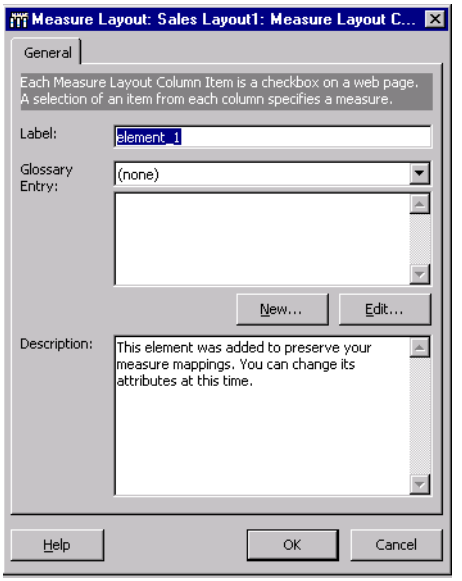
To use multi-column measure check boxes on a web page, you create a **multi-column measure layout**.

CREATING A MULTI-COLUMN MEASURE LAYOUT

1. In Analytic Manager, expand the Measures folder and select the Measure Layouts subfolder.
2. Right-click and select New Measure Layout. The Measure Layout dialog box appears.
3. Enter a name for the new measure layout. For the sample datamart, enter the name **`Sales Layout`**.
4. Click the New Item icon. The Measure Layout Column Item dialog box appears.
5. Enter a label for the measure column element. For the sample datamart, enter **`Revenue`**.
6. Click OK. The Measure Layout Column Item dialog box closes.
7. Repeat steps 4-6 and add another item called **`Units`**.

8. Click the New Column icon to add a new column. The Measure Layout dialog box opens. The dialog box shows a dummy label value as a placeholder for the first element you will add to the new column.

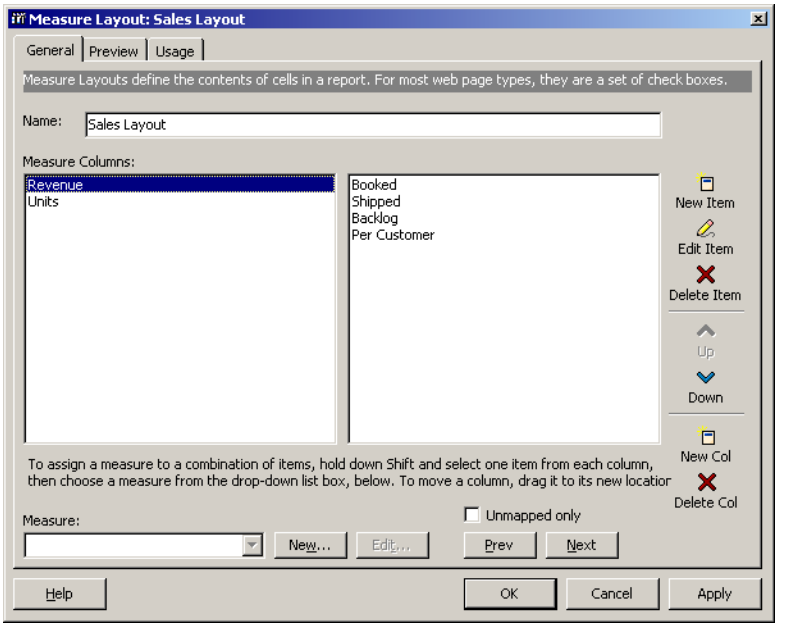
Figure 71: Dummy Values in the Measure Layout Column Item Dialog Box



9. Change the dummy Label value of element_1 to a label value of your choice. For the sample datamart, enter **Booked**.
10. Click OK. The new item is added to the second column. The Measure Layout dialog box now shows two columns. The first column contains two elements, Revenue and Units. The second column shows one element, Booked.

11. Click the New Item icon to add additional elements for the Booked column. For the sample datamart, add **`Shipped`** and **`Backlog`**.

FIGURE 72: THE MEASURE LAYOUT DIALOG BOX WITH TWO COLUMNS



12. To assign a measure to a combination of items, click an element in the first column, shift-click an element in the second column, and then select a measure from the Measure drop-down list-box.

    For the sample datamart do the following:

    a. Click Revenue in the first column.
    b. Shift-click Booked in the second column..
    c. Select Revenue Booked from the Measure drop-down list box in the lower left corner of the dialog box.

13. Repeat Step 12 for all possible combinations of items in the two columns.
14. Click OK to complete creating a multi-column measure layout.

Adding Filters

Filters allow you to select certain criteria and then generate reports for data that matches the criteria. For example, you may be interested in data for customers in certain age groups or at certain companies. When you add an attribute to a web page in a filter role, you can filter by that attribute values.

An attribute can be configured to present different filter interfaces. By default, attributes are configured to present a text box filter. A text box filter allows a user to select rows that have values in a given column matching what the user has typed into a text field.

This section describes how to configure attributes to present the following types of filter interfaces:

- List-Box filter
- Check-Box filter
- Radio-Button filter
- Multi-Column Check-Box filter
- Dynamic List-Box filter

Adding a List-Box Filter

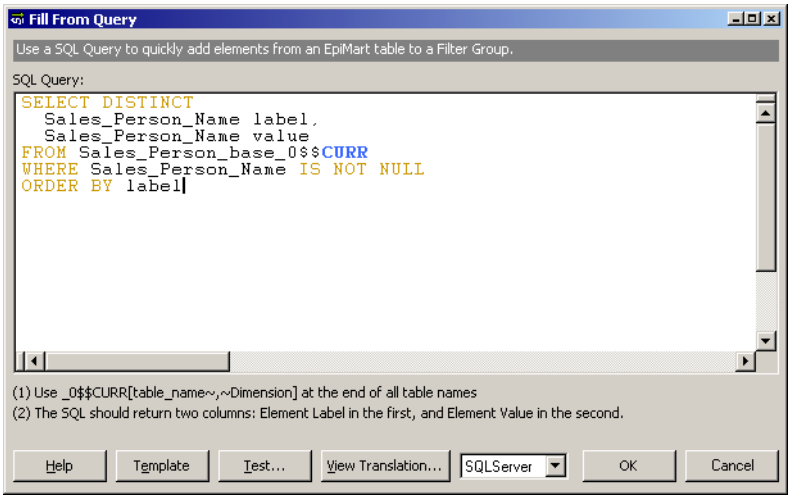
A **list-box filter** allows you to select from a pre-determined list of values for a dimension column.

Creating a List-Box Filter

1. In Analytic Manager, navigate to the Presentation folder and select the Attributes subfolder.
2. Open one of the attributes. For the sample datamart, open the Sales Rep attribute. The Attribute dialog box appears.
3. In the Attribute dialog box, under the General tab, change the Filter Type from Text Box to Listbox.

4. Select a Listbox Height. For the sample datamart, keep the default value.
5. Click the Filter Elements tab.
6. You can enter the values by hand, or you can use a SQL statement to fill in the values. To enter a value by hand, type a label and value under Elements and click Apply. For the sample database, fill in the values with a SQL statement. To fill in values with an SQL statement, do the following:
   a. Click Fill From Query. The Fill From Query dialog box appears.

FIGURE 73: THE FILL FROM QUERY DIALOG BOX



   b. Click Template. The Choose Dimension Columns to Query From dialog box displays the dimension columns for the query. For the sample datamart, leave the Sales_Person_base.Sales_Person_Name check box checked and click OK.

      A template SQL statement appears based on the dimension column that you selected. For the sample datamart, the template is as follows:

```
SELECT DISTINCT
sbp.Sales_Person_Name label,
sbp.Sales_Person_Name value
```

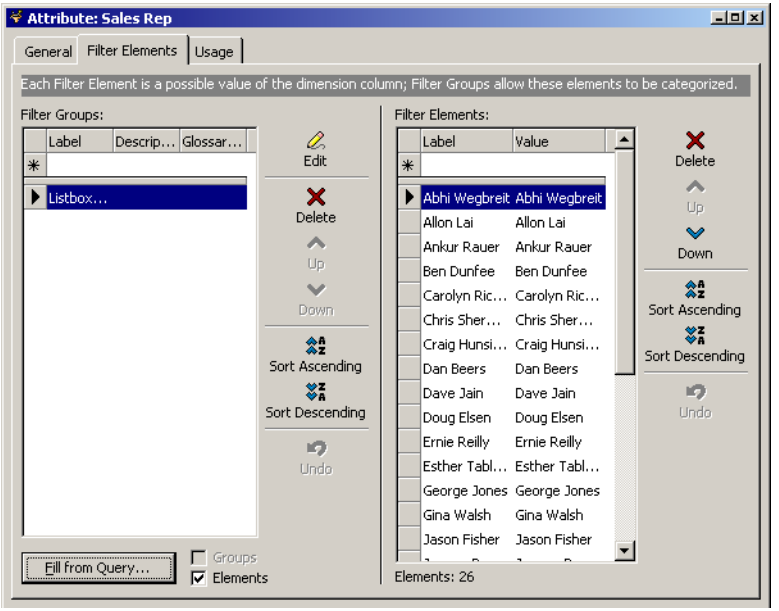
```
FROM
Sales_Person_base_0$$CURR[Sales_Person_base~,~Dimen-
sion] spb
WHERE sbp.Sales_Person_Name IS NOT NULL
GROUP BY label
```

c. Make any necessary changes to the query. For the sample datamart, no changes are required.

d. Click Test to see the results of the query.

e. Click OK. The results of your query appear in the Filter dialog box under Elements.

FIGURE 74: DEFINING FILTER ELEMENTS FOR A LIST-BOX FILTER



7. Click OK to define the filter.

Adding a Check-Box Filter

A **check-box filter** allows you to select any combination of values from a predetermined set by checking the corresponding check boxes.

Creating a Check-Box Filter

1. In Analytic Manager, navigate to the Presentation folder and select the Attributes subfolder.
2. Open one of the attributes. For the sample datamart, open the Product Line attribute. The Attribute dialog box appears.
3. Under the General tab of the Attribute dialog box, change the Filter Type from Text Box to Check Boxes.
4. Select the number of columns. for the sample datamart, use one column.
5. Click the Filter Elements tab.
6. Under Filter Group, check the Groups check box.
7. Click Fill from Query. The Fill From Query dialog box appears.
8. Click Template. The Choose Dimension Columns to Query From dialog box appears. In this dialog box, you select dimension columns for the query. For the sample datamart, make sure that the `Product_base.Product_Line` check box is checked.
9. Click OK.

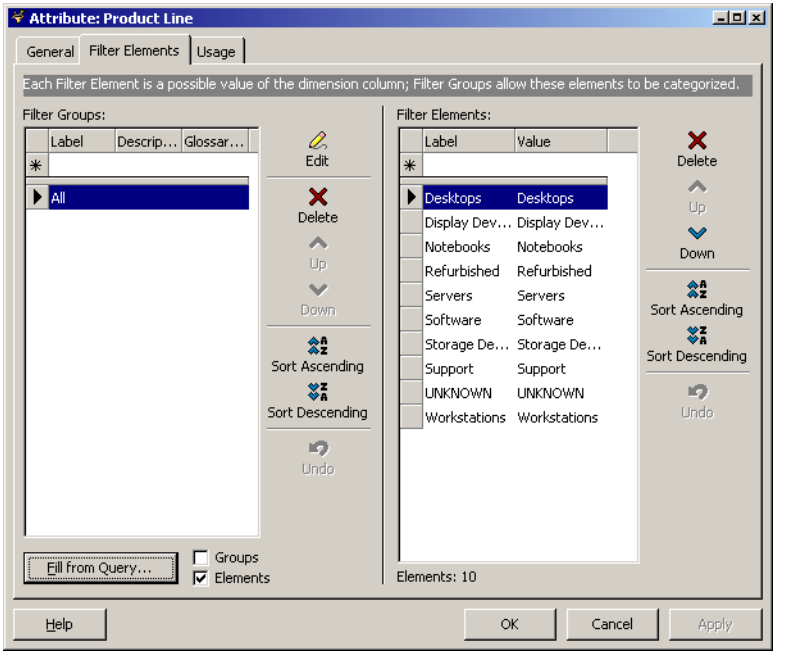
   For the sample datamart, the template is as follows:

```
SELECT DISTINCT
       'All'            group_name,
       pb.Product_Line  label,
       pb.Product_Line  value
FROM Product_base_0$$CURR[Product_base~,~Dimension] pb
WHERE pb.Product_Line IS NOT NULL
GROUP BY group_name, label
```

10. Make any necessary changes and test the SQL code. No changes are required for the sample datamart.

11. Click OK to create a filter group called All with the elements chosen by the query.

FIGURE 75: DEFINING A CHECK-BOX FILTER



12. Click OK to define the filter.

ADDING RADIO-BUTTON FILTERS

In a **radio-button filter**, only one of the elements can be selected at a time. To create a radio-button filter, follow the steps described for creating a check-box filter, except for Step 3. Under Step 3, select the Radio Buttons filter type.

The sample datamart does not have any radio button filters.

Adding a Multi-column Check-Box Filter

You can use groups to create more sophisticated **multi-column check-box filters**. For the sample datamart, create a check-box filter for Year and Quarter that has a separate row for each year.

Creating a Multi-Column Check-Box Filter

1. In Analytic Manager, navigate to the Presentation folder and select the Attributes subfolder.
2. Open the Year and Quarter attribute. The Attribute dialog box appears.
3. If you have not created this attribute, create a new attribute called Year and Quarter with date.cq_and_cy_name dimension column. To create an attribute, refer to "Creating Attributes," on page 109.
4. Change the Filter Type from Text Box to Check Boxes.
5. Select the number of columns. Set the number of columns to 4, so that there is one column for each quarter.
6. Click the Filter Elements tab.
7. Check the Groups check box next to the Fill From Query button.
8. Click Fill from Query. The Fill from Query dialog box opens.
9. Click Template.The Choose Dimension Columns to Query From dialog box lists all columns for query. For the sample datamart, check the Date.cq_and_cy_name check box and click OK.

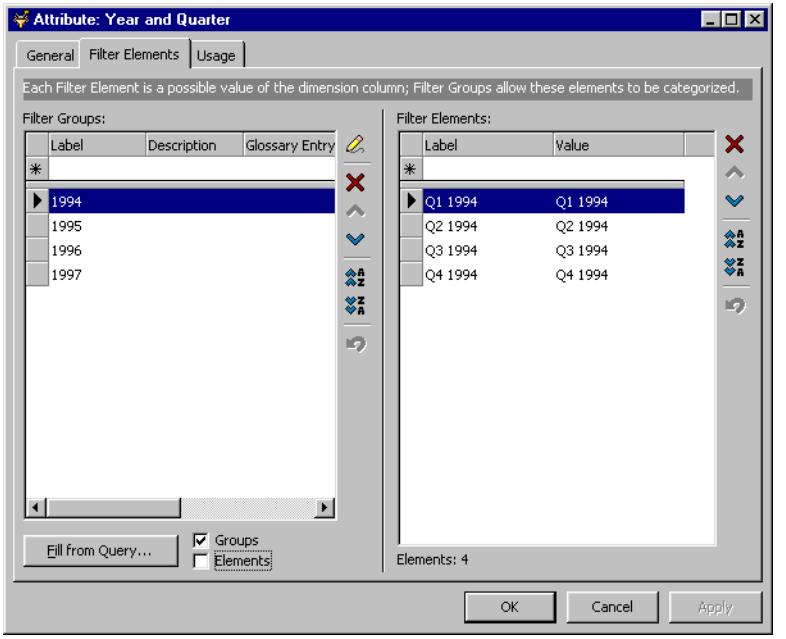
10. Change `'All'` to the name of the column that contains the value by which you want to group the check boxes. For the sample datamart, use the following query:

```
SELECT DISTINCT
       dat.cy_name          group_name,
       dat.cq_and_cy_name   label,
       dat.cq_and_cy_name   value
FROM Date_0 dat
WHERE cq_and_cy_name IS NOT NULL
GROUP BY group_name, label
```

    This query creates a filter group for each year that has elements for each quarter of the year.

11. Click **Test** to test the SQL code and click **OK**. When you click **OK**, several groups appear in the **Filter Group** column. When you click a group name, the corresponding elements appear in the **Elements** column.

FIGURE 76: DEFINING A MULTI-COLUMN CHECK BOX FILTER



12. If you have to change an element, select it in the Elements column, make changes to the Label and Value fields, and click the Update button.
13. Click OK to define the filter.

ADDING A DYNAMIC LIST- BOX FILTER

You can use a **dynamic list-box filter** to give users a choice of values that are dynamically generated from the current contents of the datamart.

CREATING A DYNAMIC LIST-BOX FILTER

1. In Analytic Manager, navigate to the Presentation folder and select the Attributes subfolder.
2. Open the attribute for which you want to create a dynamic list-box filter. For the sample datamart, open the Buyer's Company attribute. The Attribute dialog box appears.
3. Change the Filter Type from Text Box to Dynamic Listbox.
4. Select a Listbox Height. For the sample datamart, accept the default value.
5. Click Edit SQL Query.
6. Click Template in the Fill From Query dialog box. The Choose Dimension Columns to Query From dialog box appears. Select the dimension columns for the query. For the sample datamart, leave the Household_base.Organization check box checked and click OK.

   The following template appears:

```
SELECT DISTINCT
   hb.Organization  label,
   hb.Organization  value
FROM Household_base_0$$CURR[Household_base~,~Dimension] hb
WHERE hb.Organization IS NOT NULL
GROUP BY label
```

7. Make any necessary changes and test the SQL code. No changes are necessary for the sample datamart.
8. Click OK.
9. To preview the filter values that are generated from current datamart data, click the Filter Elements tab and then click Show Query Results. Note that the actual values displayed may change as the data in the datamart changes.
10. Click OK to define the filter.

ADDING ADDITIONAL FILTERS

For the sample datamart, define the filters described in Table 9 on page 259.

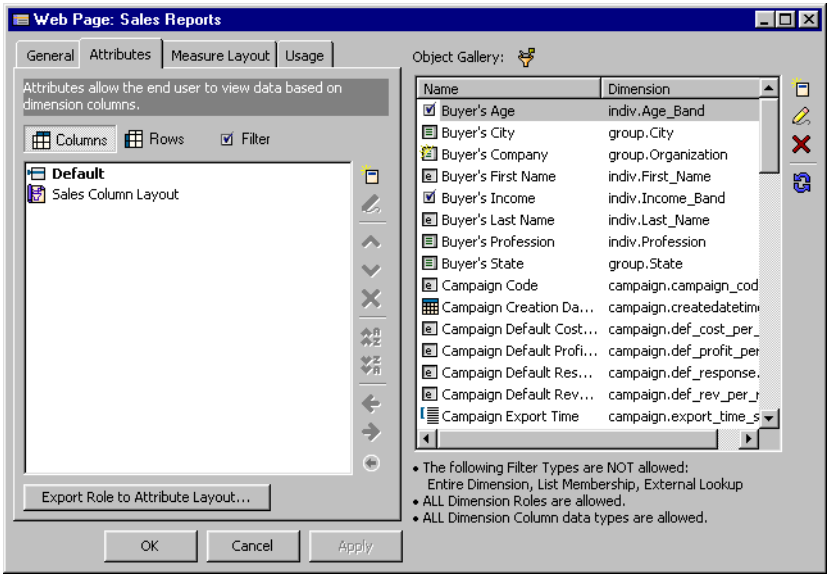
CONFIGURING THE WEB PAGE

To configure the Rows and Columns web page:

1. Double-click the web page that you created. For the sample datamart, open the Sales Reports web page. The Web Page dialog box appears.
2. Click the Attributes tab and add attributes and attribute layouts for the Columns and Rows, as you did with the first web page you created. For the sample datamart, add all of the columns shown in Figure 149 on page 261.

3. Add the filters that you configured in this chapter. To add the filters, click Filters and drag all of the desired filters from the Object Gallery to the left pane of the dialog box.

FIGURE 77: THE SALES REPORTS WEB PAGE DIALOG BOX CONFIGURED WITH FILTERS



4. Click the Measure Layout tab.
5. Drag the new measure layout from the Object Gallery to the Measure Layouts pane. For the sample datamart, use the Sales Layout.
6. Click OK to complete defining the web page.

Configuring a New Topic and Adding Security

1. Create a new Enterprise Insight topic by following the instructions listed under "Creating a New Topic," on page 122. For the sample datamart, name the new topic Sales Analysis.
2. Open the Topic. The Topic dialog box appears.
3. Click the Navigation Nodes tab and assign the new web page to the Rows & Columns – Basic, Rows & Columns – Advanced, and Rows & Columns – Add Dimensions nodes.
4. Set access permissions for the topic nodes by doing the following:
    a. Select the nodes to which you want to grant access. For the sample datamart, select all nodes in the topic.
    b. Click Add Permissions. The Choose Users and Groups dialog box appears.
    c. Select the groups or users to whom you want to grant access to the selected nodes.
    d. Click OK.
5. Click OK in the Topic Dialog box to finish defining the topic.

Adding a Default Saved Report

When a user goes to a navigation node, all options should have default values selected. In order to configure these defaults, you create a **default saved report**.

Creating a Default Saved Report

1. Navigate to the node in your web browser. For the sample datamart, this is the Rows and Columns page of the Sales Analysis topic.
2. Select default values on the web page. For the sample datamart, select different attributes from the Rows and Columns drop-down list-boxes and one item from each column under Measures.

3. Click Options. The Optional Parameters popup window appears.
4. Select one option from each group of radio buttons. You can also check any check boxes that you would like to have checked by default.
5. Click OK.
6. Click Save under Report Gallery. The Save to Report Gallery window appears.
7. Click the Folders link at the top of the window to go to the top level of the folder hierarchy.
8. Click the Public folder.
9. Click the Defaults folder to go to the Defaults subfolder of the Public folder.
10. Enter a name for the saved report. For the sample datamart, call it Sales Reports Default.
11. Click Save to save the report.

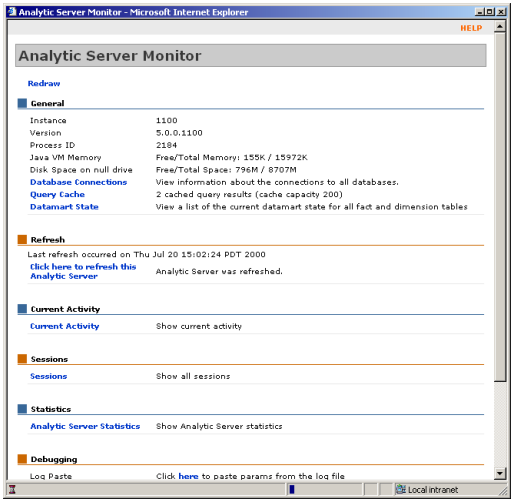
REFRESHING THE ANALYTIC SERVER

Before you use the new web page, you have to refresh the Analytic Server. You can refresh the Analytic Server using the web-based monitoring interface.

REFRESHING THE ANALYTIC SERVER USING THE WEB-BASED MONITORING INTERFACE

1. Log in to the Analytic Platform system as a user who belongs to the Administrators group.
2. Click the Monitor link at the top of the Home Page. The Analytic Server Monitor page appears.

FIGURE 78: THE APPSERVER MONITOR



3. In the Refresh section, click Click here to refresh this Analytic Server link.

   A confirmation dialog box appears.

4. Click OK. If the Analytic Server successfully refreshes, a confirmation message Analytic Server was refreshed appears.

FIGURE 79: THE REFRESH APPSERVER OPTION AND REFRESH SUCCESS MESSAGE

NOTE: *If you see the message* Failed to refresh Analytic Server, *check the log to determine the reason. The default installation places the Analytic Server log files in the following directory:*

```
C:\Program Files\Epiphany\instance\Web\WWWroot\logfiles
```

Replace instance with the name under which you have installed your Analytic Platform instance.

VIEWING WEB PAGES

1. Log in using the procedure described in "Viewing the Web Page," on page 133. The home page now shows the new topic that you created. Go to the Rows and Columns web page of the new topic.
2. Click the Filters button on the web page.
3. Click the Options button on the web page to see some of the Options that are available. You can configure options such as chart types and number of rows displayed. For more information on configuration options, refer to the online help.

Analyzing and Reporting Data

This chapter describes how to create web pages and topics for **analysis** and **reporting**. Analysis and reporting applications typically use topics that derive from the **Enterprise Insight** topic master. The topic master provides navigation nodes (locations) and links for a number of web page types that provide access to sophisticated analytical tools.

This chapter describes how to create the following web pages:

- Profiling
- High/Low Clusters
- Trends
- Lifecycles
- Cumulative Projections
- Influences
- Community Clusters

You can create the above analytic web pages by copying the completed Rows and Columns web page and making some minor changes. Therefore, before you begin working on the above web pages, make sure that you complete all the procedures associated with Rows and Columns web page. For more information on how to create Rows and Columns web pages, refer to Chapter 6, "Creating Rows and Columns Web Pages."

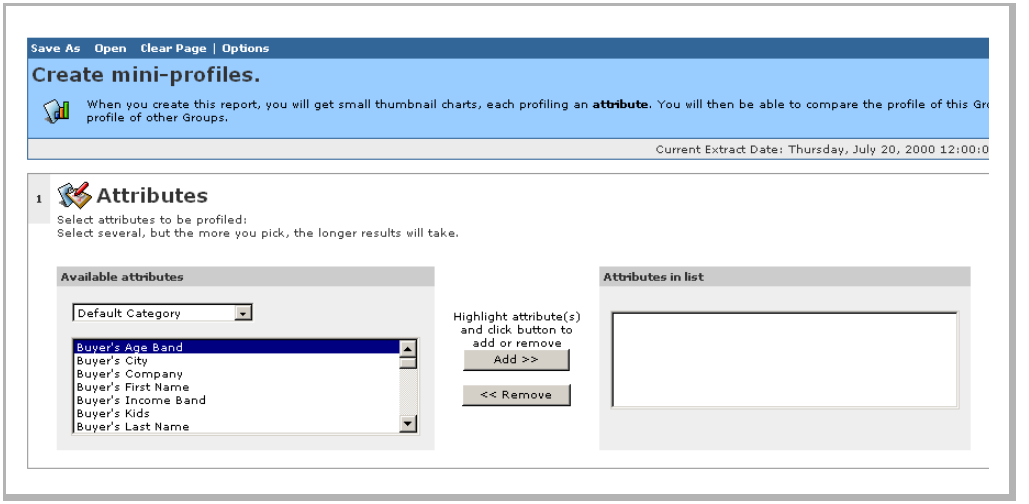
Profiling

A **Profiling** web page allows you to generate graphical profiles of the datamart data and compare these profiles between groups.

The web page allows you to generate charts that graphically profile various aspects of the business. The web page also provides a visual display of the variables of the business, which enables you to compare the profiles of different groups. With this web page, you can do a kind of visual data mining in order to identify patterns in the data. You can also compare profiles of two subsets of the data by producing a profile for one group and then defining another group for comparison.

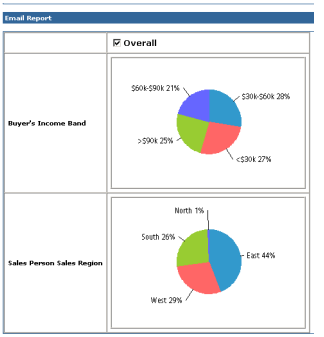
For the sample datamart, the Profiling web page appears as follows.

FIGURE 80: A PROFILING WEB PAGE



Both profiles use the same color for comparable values so that patterns or discrepancies can be easily identified. The rows of the comparison report are sorted so that the groups that show the biggest difference between the columns are at the top. Profiling web pages are also useful for a quick data-quality assessment. With the sample datamart web page, you can generate the following report.

FIGURE 81: A PROFILING REPORT



CREATING A PROFILING WEB PAGE

To create a Profiling web page, begin with the Rows and Columns web page that you created in Chapter 6, "Creating Rows and Columns Web Pages."

CREATING A PROFILING WEB PAGE

1. In Analytic Manager, expand the Presentation folder and select the web pages subfolder.
2. Select the Rows and Columns web page you created. For the sample datamart, select the Sales Reports web page. Right-click, and select Duplicate. The Duplicate Web Page dialog box appears.

3. Enter a name for the new web page. For the sample datamart, enter the name **`Sales Profiling`**.

FIGURE 82: THE DUPLICATE WEB PAGE DIALOG BOX

4. Click OK to create the duplicate web page. The new web page appears under web pages in the Presentation folder.
5. Double-click and open the new Sales profiling web page. The Web Page dialog box appears.
6. Select Profiling from the Web Page Type drop-down list box. A confirmation dialog box appears.
7. Click Yes.
8. Click the Attributes and Measure Layout tabs to check that everything has been copied to the new web page. Check both the attributes (under DimSingle) and the Filters in the Attributes tab.
9. Make any necessary changes to the web page. You do not have to make any changes to the Sales Profiling web page.
10. Click OK to save the changes.

Once you create a web page, you have to assign the web page to the appropriate topic node.

ASSIGNING A WEB PAGE TO A TOPIC NODE

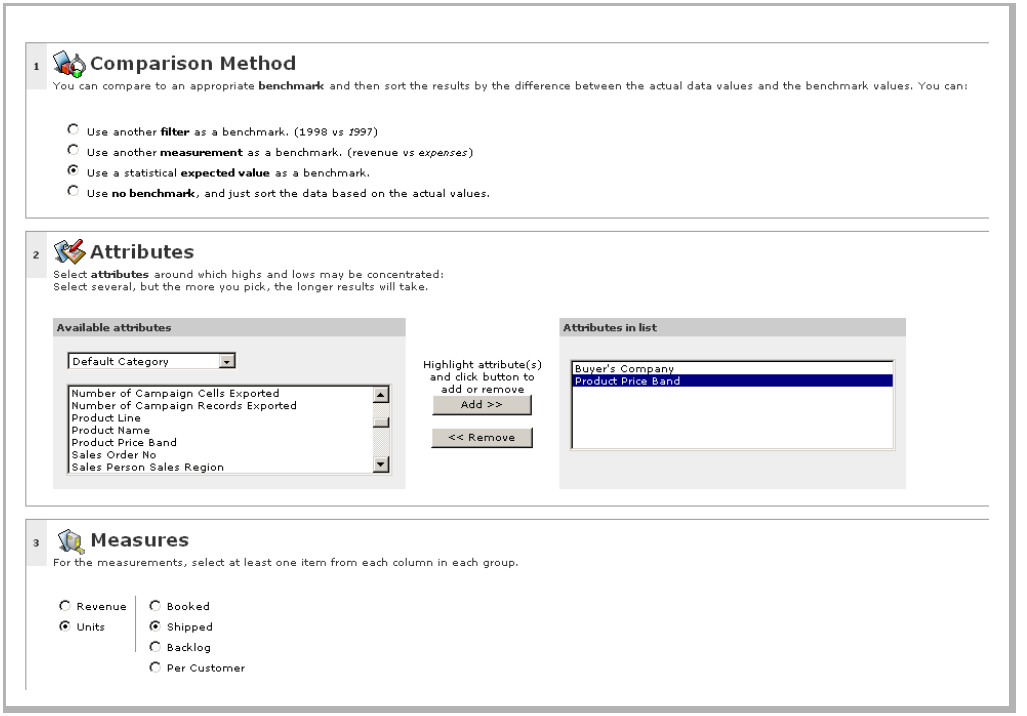
1. Open the topic you created in Chapter 6, "Creating Rows and Columns Web Pages." For the sample datamart, open the Sales Analysis topic.
2. Click the Navigation Nodes tab.
3. Select the Profiling row under Nav Nodes.
4. Click the web page column of the selected row. Analytic Manager displays a drop-down list box in the selected cell. Select the new Profiling web page from the drop-down list box. For the sample datamart, select the Sales Profiling web page.
5. Select Profiling – Add Dimensions row under Nav Nodes.
6. Click the web page column of the Profiling – Add Dimensions row. Analytic Manager displays a drop-down list box in the selected cell. Select the new Profiling web page from the drop-down list box. For the sample datamart, select the Sales Profiling web page.
7. Use the Add Permissions (Users and Groups) button to grant access to these nodes.
8. Click OK to complete editing the topic.
9. Refresh the Analytic Server. For more information, refer to "Refreshing the Analytic Server," on page 156.
10. Create a default saved report for this navigation node. For more information, refer to "Adding a Default Saved Report," on page 155.
11. View the web page.

HIGH/LOW CLUSTERS

The **High/Low Clusters** web page allows you to quickly scan the data to spot high and low values. The web page provides a quick review of the data by finding the highest and lowest values based on one or more attributes. With this web page, you can find data that differs from the rest, either by being lower or by being higher.

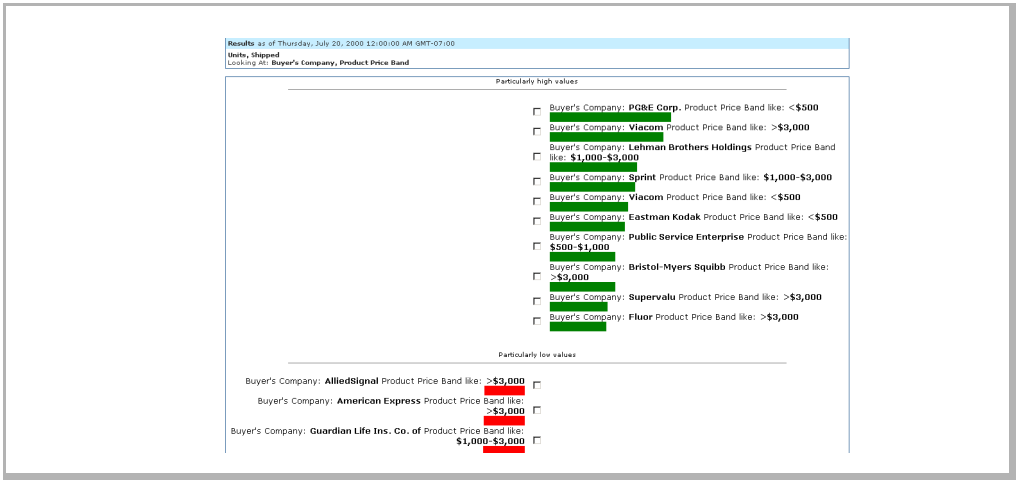
For the sample datamart, the High/Low Clusters web page appears as follows.

FIGURE 83: A HIGH/LOW CLUSTERS WEB PAGE



The web page generates the following report.

FIGURE 84: A HIGH/LOW CLUSTERS REPORT



The configuration for the High/Low Clusters web page is similar to that of the Profiling web page.

CREATING A HIGH/LOW CLUSTERS WEB PAGE

1. In Analytic Manager, expand the Presentation folder and select the web pages subfolder.
2. Select the Profiling web page you created. For the sample datamart, select the Sales Profiling web page. Right-click, and select Duplicate. The Duplicate Web Page dialog box appears
3. Enter the name **`Sales Highs and Lows`**.
4. Open the new web page and change the Web Page Type to High/Low Clusters.
5. Verify that all attributes, filters, and measure layouts have been copied correctly.
6. Click OK to save your changes.
7. Open your topic and click the Navigation Nodes tab. For the sample datamart, open the Sales Analysis topic.

8. Assign the web page to the Highs & Lows and Highs & Lows – Add Dimensions nodes.
9. Click OK to save your changes.
10. Refresh the Analytic Server. For more information, refer to "Refreshing the Analytic Server," on page 156.
11. Create a default saved report for this navigation node. For more information, refer to "Adding a Default Saved Report," on page 155.
12. View the web page.

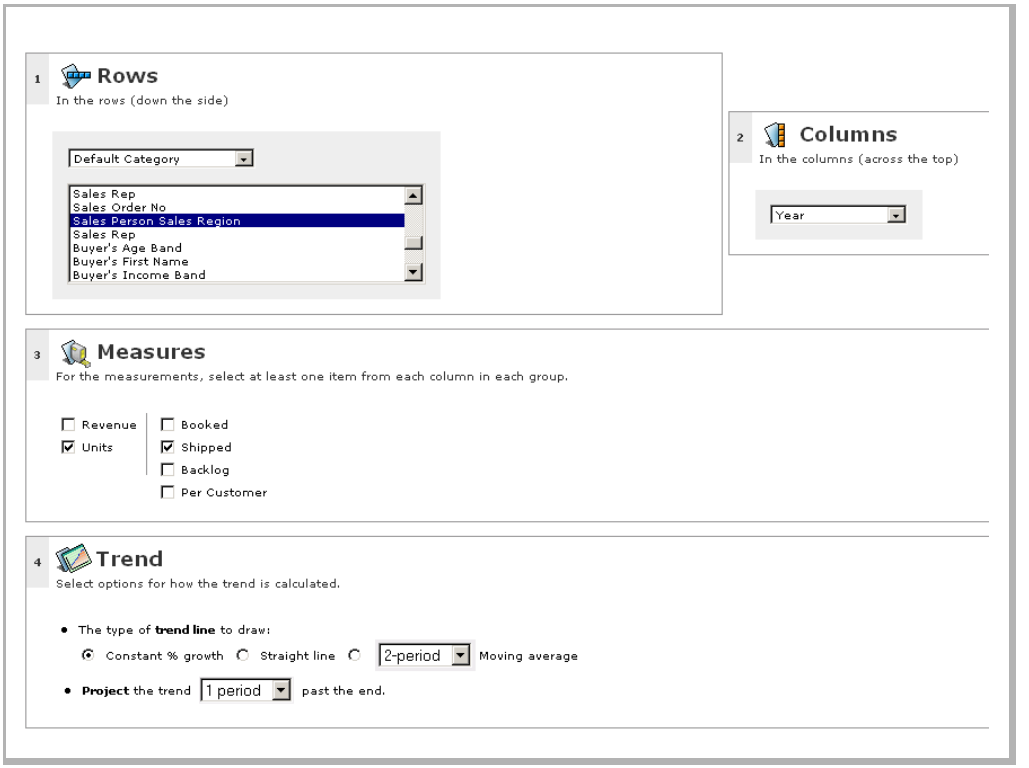
TRENDS

The **Trends** web page allows you to extrapolate data into the future. This is done by fitting a curve to the current data and extending the curve into the future with one of the following predictive techniques:

- Straight line (simple linear regression)
- Constant percent growth (linear regression on the logarithm of the data)
- *n*-period moving average

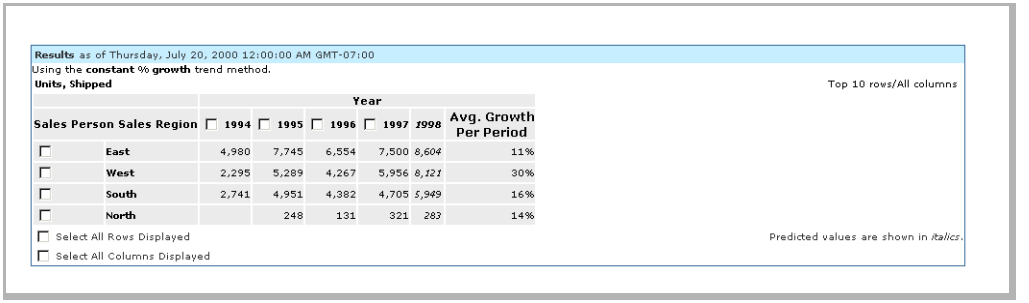
For the sample datamart, the Trends web page appears as follows.

FIGURE 85: A TRENDS WEB PAGE



The web page allows you to generate the following report.

FIGURE 86: A TRENDS REPORT



Results as of Thursday, July 20, 2000 12:00:00 AM GMT-07:00
Using the **constant** % **growth** trend method.
Units, Shipped

Top 10 rows/All columns

Sales Person Sales Region	Year 1994	1995	1996	1997	*1998*	Avg. Growth Per Period
East	4,980	7,745	6,554	7,500	*8,604*	11%
West	2,295	5,289	4,267	5,956	*8,121*	30%
South	2,741	4,951	4,382	4,705	*5,949*	16%
North		248	131	321	*283*	14%

Select All Rows Displayed
Select All Columns Displayed

Predicted values are shown in *italics*.

To create a Trends web page, you duplicate a full-featured Rows and Columns web page and make some changes to the attribute configuration. A Trends web page displays time periods in the columns, so you have to make sure that these are the only kinds of attributes that can be selected for the columns.

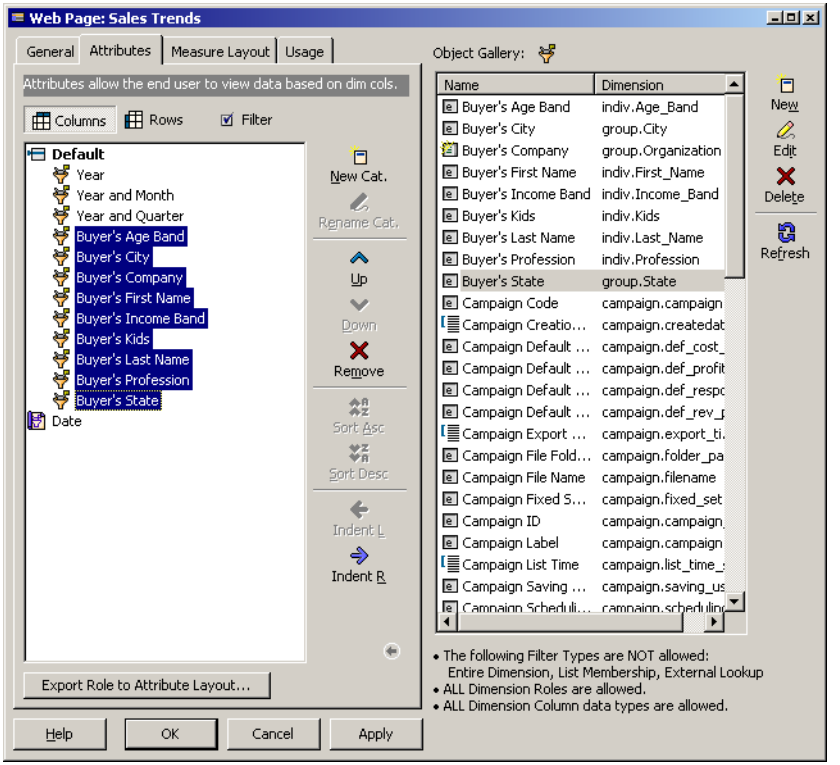
CREATING A TRENDS WEB PAGE

1. In Analytic Manager, expand the Presentation folder and select the web pages subfolder.
2. Select the Rows and Columns web page you created. Right-click, and select Duplicate. For the sample datamart, select the Sales Reports web page. The Duplicate Web Page dialog box appears
3. Enter a name for the new web page. For the sample datamart, enter the name **`Sales Trends`**.
4. Double-click the web page to open the page. In the Web Page dialog box, change the Web Page Type to Trends.
5. Click the Attributes tab. All the attributes that you created are available for both rows and columns.
6. Click Columns. In the left pane, select all of the attributes that are not from the date dimension.
7. For the sample datamart, if you created the attribute layouts, select all the layouts except the Date layout. Alternately, select all of the attributes except Year, Year and Quarter, and Year and Month.

NOTE: *You can select multiple attributes from the list by first selecting one and then holding down the* **`Ctrl`** *key while selecting additional attributes.*

8. Click the Delete icon to remove the selected attributes from the Column attribute role.

FIGURE 87: REMOVING ATTRIBUTES FROM THE COLUMN ATTRIBUTE ROLE



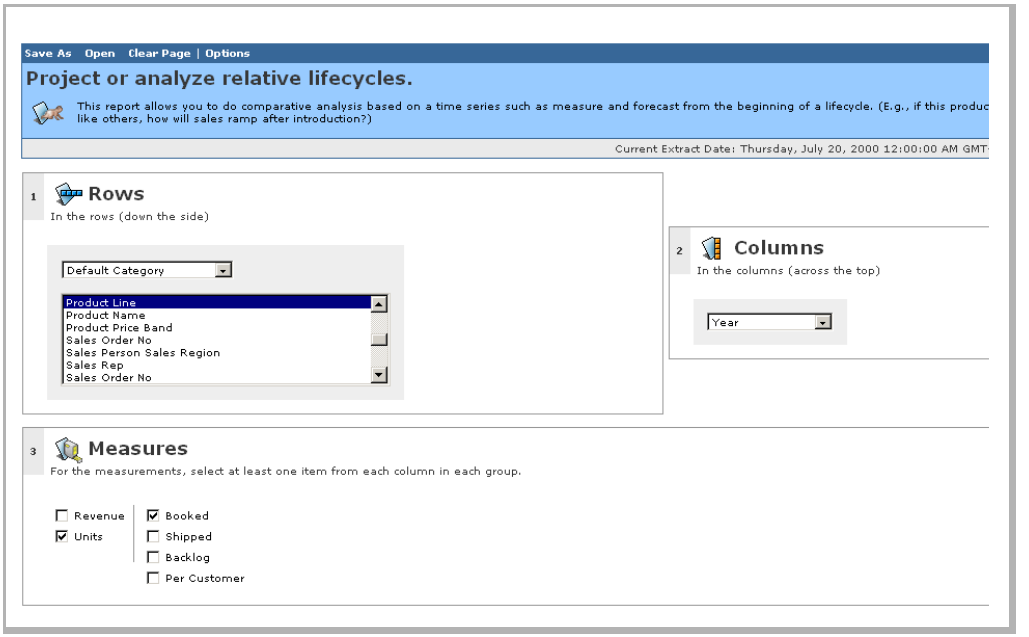
9. Click Yes in the confirmation dialog box.
10. Remove the Date dimension attributes, or the Date attribute layout, from the Row attribute role.
11. Click OK in the Web Page dialog box to save the changes.
12. Open your topic and assign the new web page to the Trends – Basic and Trends – Basic – Add Dimensions nodes.
13. Click OK to save the changes.

14. Refresh the Analytic Server. For more information, refer to "Refreshing the Analytic Server," on page 156.
15. Create a default saved report for this navigation node. For more information, refer to "Adding a Default Saved Report," on page 155.
16. View the web page.

LIFECYCLES

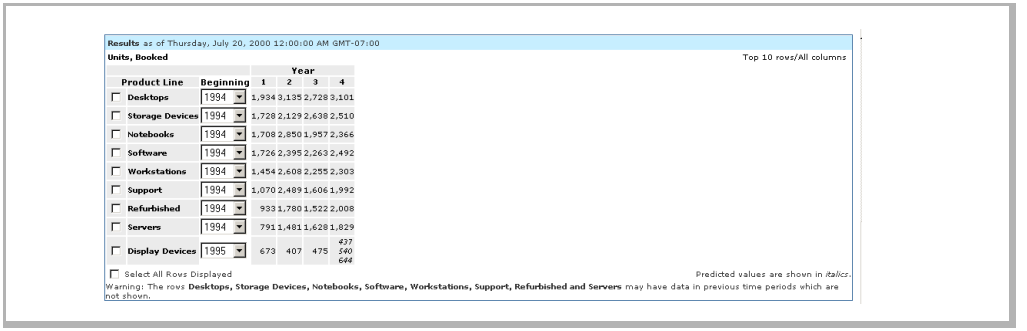
The **Lifecycles** web page allows you to predict the performance of new business entities based on the past performance of related entities. The web page also allows you to look at data from the product life cycle viewpoint. You can look at a product relative to some point in time (for example, the time since a product was introduced or the time since a customer was acquired) or use an older product to predict the life cycle of a new one. For the sample datamart, the Lifecycles web page appears as follows.

FIGURE 88: A LIFECYCLES WEB PAGE



The Life Cycles web page creates the following report.

FIGURE 89: A LIFECYCLES REPORT



Lifecycles tracks things across time. Therefore, the web page setup is similar to that for Trends.

CREATING A LIFECYCLES WEB PAGE

1. Duplicate the Trends web page. For the sample datamart, duplicate the Sales Trends web page and enter the name **`Sales Lifecycles`** for the new web page.
2. Open the Sales Lifecycles web page. The Web Page dialog box appears.
3. Change the Web Page Type to Lifecycles.
4. Verify that all attributes, filters, and measure layouts have been copied correctly. As with the Trends web page, the Column attribute role should have only date attributes assigned to it.
5. Click OK to save the changes.
6. Open the topic. In the Topic dialog box, assign the new web page to the Trends – Lifecycles and Trends – Lifecycles – Add Dimensions nodes.
7. Click OK to save the changes.
8. Refresh the Analytic Server. For more information, refer to "Refreshing the Analytic Server," on page 156.

9. Create a default saved report for this navigation node. For more information, refer to "Adding a Default Saved Report," on page 155.
10. View the web page.

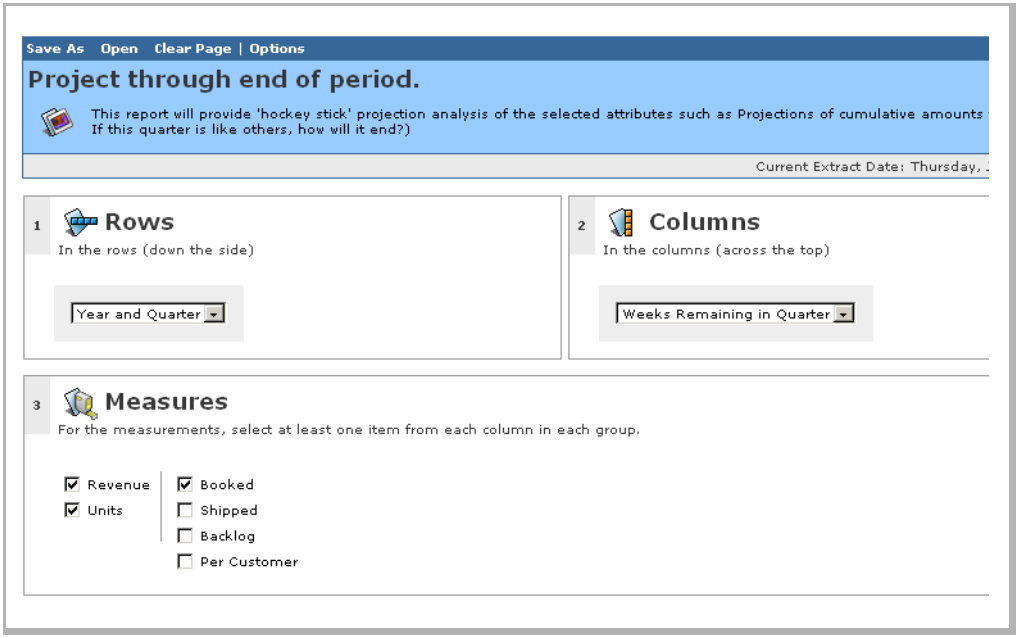
CUMULATIVE PROJECTIONS

The **Cumulative Projections** web page enables you to predict the remainder of the current quarter (or other time period) based on past performance data. The web page allows you to answer questions such as, "If this quarter looks like previous quarters, how is it likely to end?" Projections are based on ratios between periods within past quarters. Prediction for future periods in the current quarter are found by multiplying these ratios by results from the quarter so far. Three sets of predictions are computed:

- Min is based on the lowest ratio of any past quarter.
- Ave is based on the average of the ratios for past quarters.
- Max is based on the highest ratio of any past quarter.

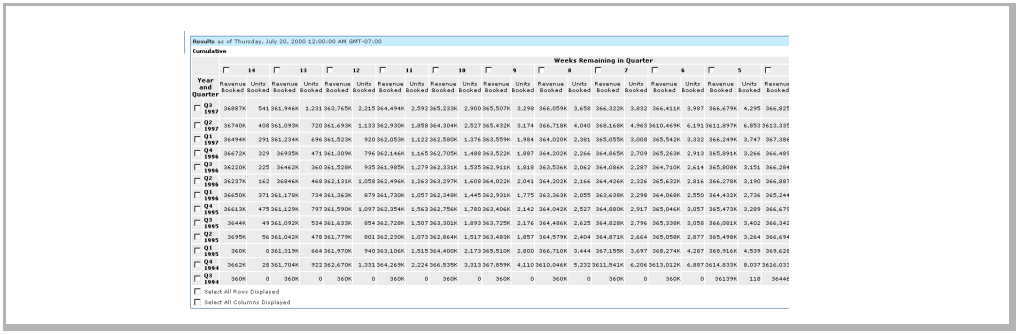
For the sample datamart, the Cumulative Projections web page appears as follows.

FIGURE 90: A CUMULATIVE PROJECTIONS WEB PAGE



The web page creates the following Cumulative Projections report.

FIGURE 91: A CUMULATIVE PROJECTIONS REPORT



The Cumulative Projections web page is set up slightly differently from the web pages discussed so far. This web page gives information about a time period (such as a quarter) from the point of view of a smaller time interval (such as a week). The results from this web page are displayed with the time period of interest in the rows, and with intervals remaining in the time period in the columns. For example, you might have Quarters in the rows and Weeks Remaining in Quarter in the columns.

You begin by creating new attributes for remaining time intervals.

1. In Analytic Manager, expand the Presentation folder and select the Attributes subfolder.
2. Right-click and select New Attribute.
3. Enter a name for the attribute. For the sample datamart, enter the name **`Weeks Remaining in Quarter`**.
4. For the Dimension Column, select an attribute from the date dimension. Select one of the attributes named *interval*_number_til_end_*timeperiod*. For the sample datamart, select week_number_til_end_cq. The attribute is not used for filtering. Therefore, leave the Filter Type set to Text Box.
5. Click OK to complete defining the attribute.
6. Define other applicable attributes. For the sample datamart, create an attribute called Days Remaining in Quarter with dimension column day_number_til_end_cq.

CREATING A CUMULATIVE PROJECTIONS WEB PAGE

1. Duplicate the full-featured Rows and Columns web page. For the sample datamart, duplicate the Sales Reports web page and enter the name **`Sales Projections`** for the new web page.
2. Open the new web page. The Web Page dialog box opens.
3. Change the Web Page Type to Cumulative Projections.
4. Click the Attributes tab, click Rows, and remove all attributes except the attribute for the time period of interest.

For the sample datamart, remove all attributes except Year and Quarter. If you created the attribute layouts, remove the date attribute layout, and add the Year and Quarter attribute to Rows.

5. Click Columns and remove all attributes.
6. Add Attributes that express time intervals remaining in the time period of interest for the column attribute role. For the sample datamart, add the Weeks Remaining in Quarter and Days Remaining in Quarter attributes that you created.
7. Verify that filters and measure roles have been copied correctly.
8. Click OK to save the changes.
9. Open your topic and assign the new web page to the Trends – Projections and Trends – Projections – Add Dimensions nodes.
10. Click OK to save the changes.
11. Refresh the Analytic Server. For more information, refer to "Refreshing the Analytic Server," on page 156.
12. Create a default saved report for this navigation node. For more information, refer to "Adding a Default Saved Report," on page 155.
13. View the web page.

INFLUENCES

The **Influences** web page allows you to build models of data to identify the factors with the greatest impact on the business. Influences models the impact of a set of attributes on another attribute or measure.

Influences builds models using two types of trees:

- Classification Tree: A classification tree predicts the value of a particular discrete-valued attribute based on the values of a set of attributes. For example, "How do the age, income, and city of residence of a customer influence whether the customer will buy an extended warranty?"

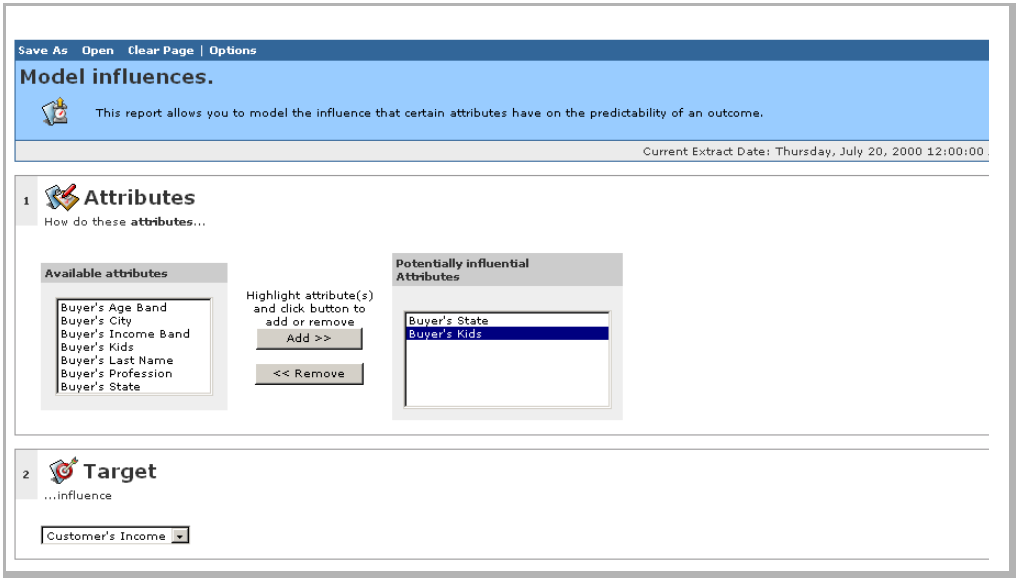
- Regression Tree: A regression tree predicts the value of a particular numeric measure based on the values of a set of attributes. For example, "How do age, income, and city of residence influence the total amount of revenue that a customer will generate?"

Before you configure an Influences web page, select a primary dimension, a target, and source attributes.

- The primary dimension is the entity or dimension about which Influences will make predictions. For example, if you are investigating how certain customer characteristics influence product purchases, the primary dimension would be the customer. You must specify the primary dimension when you configure your web page. For the sample datamart web page, select Individual as the primary dimension.
- The target is the entity that you are trying to predict. For example, if your goal is to predict customer's influence on product purchases, then the target is whether or not an existing customer bought a particular product. A target can be either an attribute or a numerically valued measurement. For the sample datamart, select the customer's income and revenue generated as targets.
- The source attributes are the attributes that might have an impact on the prediction. For example, to predict the customer's influence on product purchases, source attributes might be customers' age, income, and education. There must be a single value of the source attribute for every element of the primary dimension. For the sample datamart, use attributes from the group and indiv dimension roles.

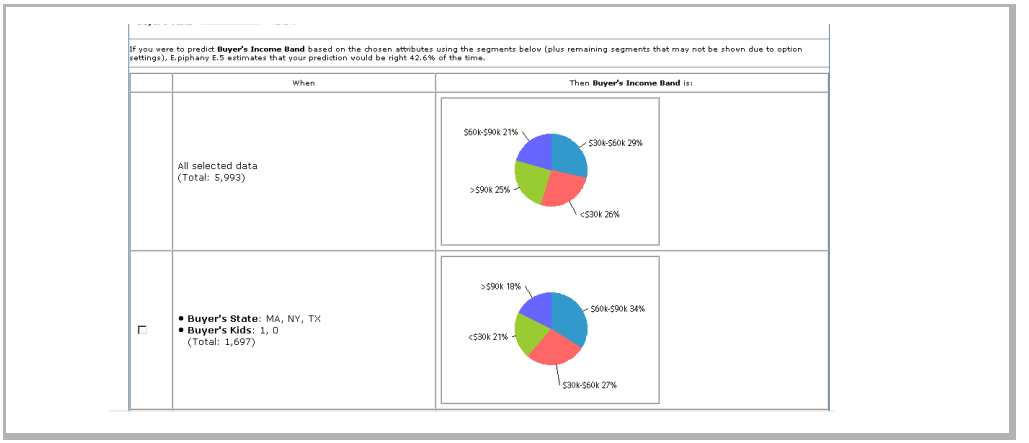
For the sample datamart, the Influences web page appears as follows.

FIGURE 92: AN INFLUENCES WEB PAGE



The web page creates the following Influences report.

FIGURE 93: AN INFLUENCES REPORT



CREATING AN INFLUENCES WEB PAGE

Influences web pages differ from the web pages that you have defined up to now in that they use measure sets rather than measure layouts. A measure set determines how measurements are to be made on a target.

You create an Influences web page by doing the following:

- Creating a Counting Measure
- Creating a Classification Measure Set
- Creating a Regression Measure Set
- Creating an Attribute for the Primary Dimension
- Creating and Configuring the Influences Web Page

CREATING A COUNTING MEASURE

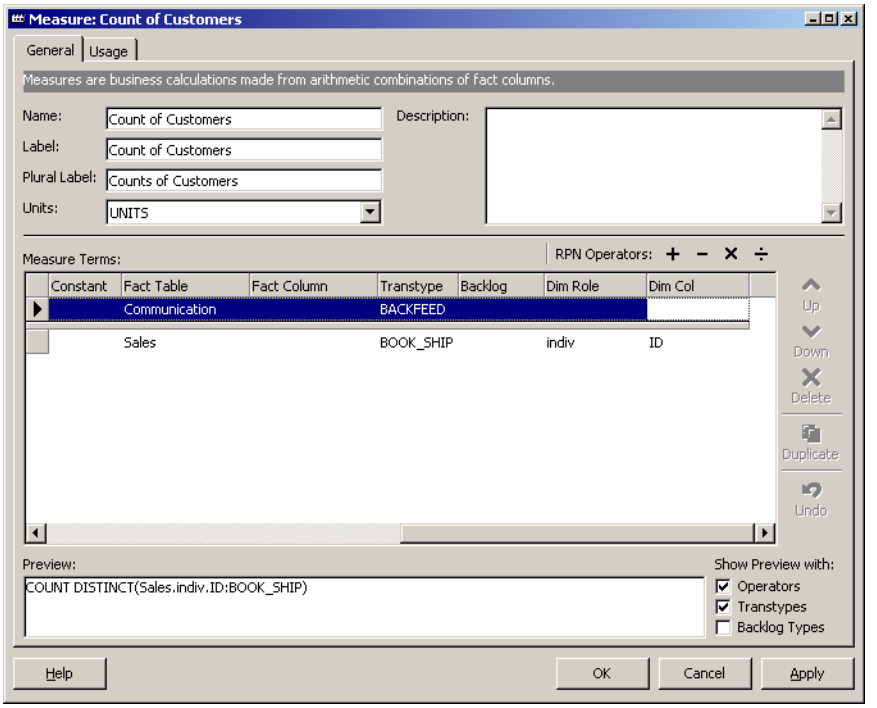
You create a measure to count the primary dimension. Based on the value of a measure, you can select to give more or less weight to any member of the primary dimension. For example, you can give more weight to customers who have made larger purchases. For the sample datamart, however, give each individual equal weight by creating a measure that counts distinct individuals in your fact table.

CREATING A COUNTING MEASURE

1. In Analytic Manager, navigate to the Measures folder. Right-click and select New Measure.
2. Give the measure a name and description, and set the measurement units to UNITS. For the sample datamart, enter the name **`Count of Customers`**.
3. In the Measure Terms grid, click in the first row of the Operator column. Analytic Manager displays a scroll bar in the Operator column. Select the COUNT DISTINCT operator to ensure that an individual is not counted more than once.
4. Click in the first row of the Fact Table column. Select the Sales Fact table. In this procedure, you count individuals who have records in the Sales Fact table.

5. Leave the null value selected in the Fact Column column because you are not using actual fact column values in this measure.
6. Click the Trans Type column and specify BOOK_SHIP, a transaction type set that includes both BOOK and SHIP transaction types.
7. Click the Dim Role column and specify indiv.
8. Click the Dim Col column and select ID. ID is a column of the Individual_base dimension that duplicates the sskey column, so it is guaranteed to have a unique value for each individual. The COUNT DISTINCT operator counts each individual exactly once.
9. Click Apply to add the measure term.

FIGURE 94: THE COUNT OF CUSTOMERS MEASURE



10. Click OK to define the measure.

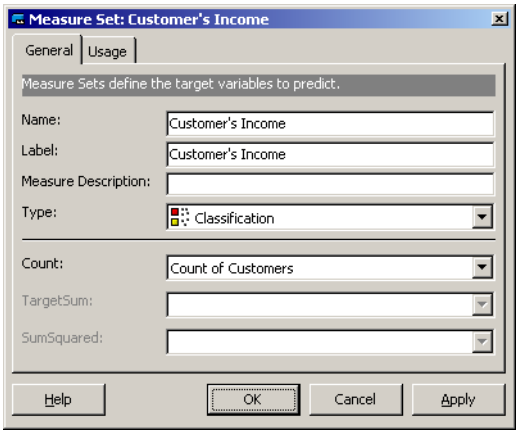
CREATING A CLASSIFICATION MEASURE SET

After you create a counting measure, you create a measure set for a classification target. For the sample datamart, create a measure set for Buyer's Income.

CREATING A CLASSIFICATION MEASURE SET

1. In Analytic Manager, navigate to the Measures folder. Select the Measure Sets subfolder.
2. Right-click and select New Measure Set.
3. Enter a name and description for the Measure Set. For the sample datamart, enter the name **`Customer's Income`**.
4. Select Classification from the Type drop-down list box.
5. Select the Count measure from the drop-down list box. For the sample datamart, select the Count of Customers measure that you created.

FIGURE 95: THE MEASURE SET DIALOG BOX



6. Click OK to define the Measure Set.

Creating a Regression Measure Set

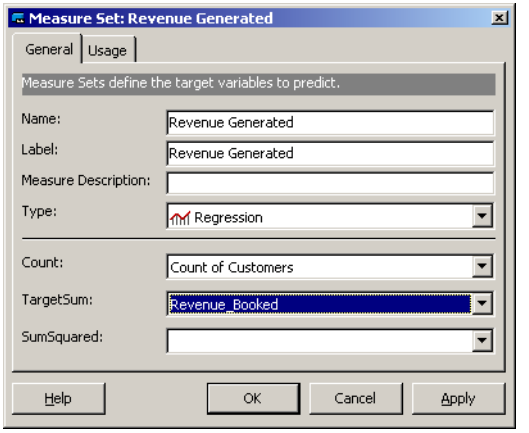
After you create a counting measure and a measure set for a classification target, you create a measure set for a regression target. For the sample datamart, create a measure set for revenue generated.

Creating a Regression Measure Set

1. In Analytic Manager, navigate to the Measures folder. Select the Measure Sets subfolder.
2. Right-click and select New Measure Set. The Measure Set dialog box appears.
3. Enter a name and description for the Measure Set. For the sample datamart, enter the name **`Revenue Generated`**.
4. Select Regression from the Type drop-down list box.
5. Select the Count measure from the drop-down list box. For the sample datamart, select the Count of Customers measure that you created.
6. Select the target measure from the TargetSum drop-down list box. The target measure is the measure whose value you want to predict. For the sample datamart, select the Revenue-booked measure.

7. Leave the SumSquared field empty.

FIGURE 96: DEFINING A REGRESSION MEASURE SET



8. Click OK to define the measure set.

CREATING AN ATTRIBUTE FOR THE PRIMARY DIMENSION

When you create an Influences web page, you have to specify the primary dimension. You can specify the primary dimension with a special kind of attribute that refers to an entire dimension role.

CREATING AN ATTRIBUTE FOR THE PRIMARY DIMENSION

1. In Analytic Manager, navigate to the Presentation folder and select the Attributes subfolder.
2. Right-click and select New Attribute. The Attribute dialog box appears.
3. Enter a name for the attribute. For the sample datamart, enter the name **`Individual Dimension.`**

4. Select Entire Dimension from the Filter Type drop-down list box. When you select this filter type, the Dimension Column pane changes to display only dimension role names.
5. Select the dimension column for which you want to create an attribute. For the sample datamart, select indiv.
6. Click OK to define the attribute.

CREATING AND CONFIGURING THE INFLUENCES WEB PAGE

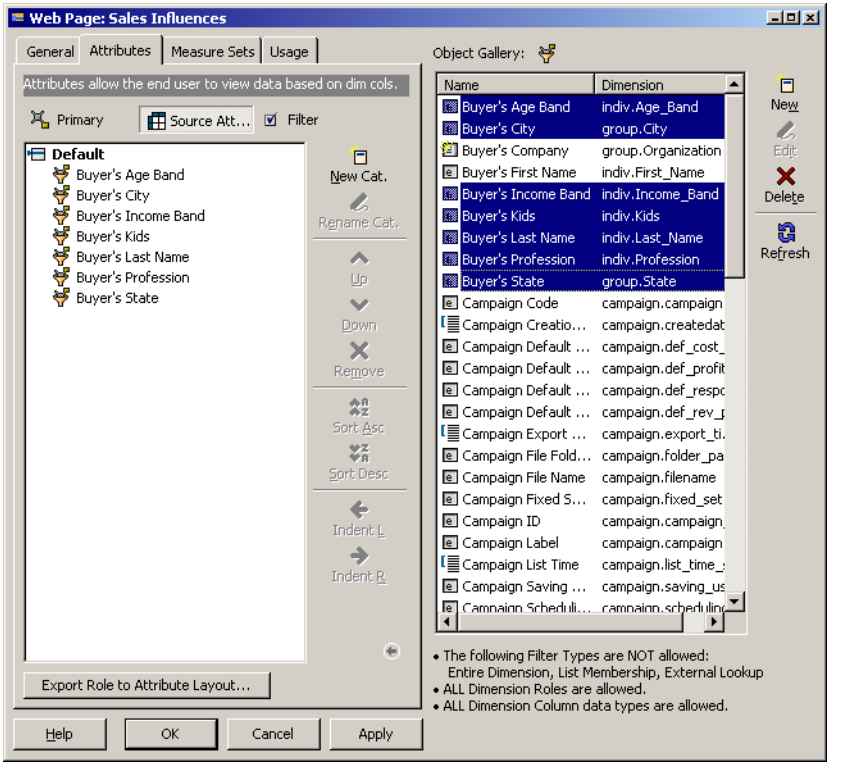
1. In Analytic Manager, navigate to the Presentation folder and select the web pages subfolder.
2. Right-click and select New web page.
3. Enter a name and description for the new web page. For the sample datamart, enter the name **`Sales Influences`**.
4. Set the Web Page Type to Influences.
5. Click the Attributes tab.
6. Click Primary and drag the new attribute for your primary dimension from the Object Gallery to the left pane. For the sample datamart, select the Individual Dimension attribute.
7. Add all attributes of interest that have unique values on the primary dimension. You add attributes by dragging them from the Object Gallery.
8. Verify that all attributes and filters have been copied correctly. For the sample datamart, use the same primary dimension and leave the attributes and filters unchanged.

NOTE: *Notice that some of these attributes are from the* Household *dimension. Using attributes from this dimension is not a problem if an individual cannot be a member of more than one household. What is necessary for consistency is that an individual can only be in households within a single city, state, and organization. If an individual can belong to households in several different cities, then* Household.State *would not have a single value for every individual.*

9. Click Filter to add filters. For the sample datamart, use the same filters that you used in the Sales Reporting web page. Add the Indiv, Group, Product, marketing_Program, Quick_Count, Sales_Person filters.
10. Click the Measure Sets tab.
11. Add the classification measure set to the Measure Sets pane by dragging it from the upper half of the Object Gallery. For the sample datamart, drag the Customer's Income measure set to the left pane.
12. Add the classification target to the measure set by dragging an attribute from the lower half of the Object Gallery to the Measure Sets pane and dropping it on the classification measure set that you just added.

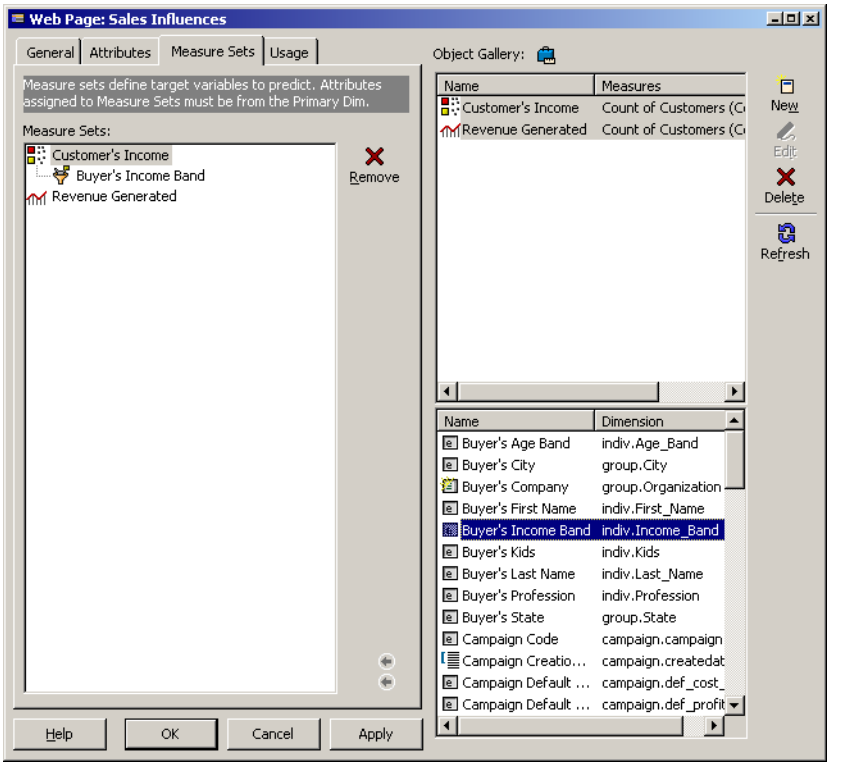
13. When you add the target in this way, the name of the target attribute appears under the name of the measure set. For the sample datamart, drop the Buyer's Income attribute on the Customer's Income measure set.

FIGURE 97: ATTRIBUTES FOR AN INFLUENCES WEB PAGE



14. Add the regression measure set to the **Measure Sets** pane by dragging it from the upper half of the **Object Gallery**. For the sample datamart, use the measure set called **Revenue Generated**. Regression measure sets do *not* have associated attributes.

FIGURE 98: MEASURE SETS FOR AN INFLUENCES WEB PAGE



15. Click **OK** to complete defining the web page.
16. Open your topic and assign the new web page to the **Influence** and **Influence – Add Dimensions** nodes.
17. Add access permissions for the web page.
18. Click **OK** to save the changes.

19. Refresh the Analytic Server. For more information, refer to "Refreshing the Analytic Server," on page 156.
20. Create a default saved report for this navigation node. For more information, refer to "Adding a Default Saved Report," on page 155.
21. View the web page.

BAYESIAN CLASSIFIER

Like an Influences web page, a **Bayesian Classifier** web page models the influence of a set of attributes on another attribute. You can create a Bayesian Classifier web page by starting with the Influences web page that you created.

CREATING A BAYESIAN CLASSIFIER WEB PAGE

1. Select the Influences web page. For the sample datamart, select the Sales Influences page. Right-click, and select Duplicate. The Duplicate Web Page dialog box appears.
2. Enter a name for the new web page and click OK. For the sample datamart, enter **`Sales Bayes`**.
3. Open the new Sales Bayes web page.
4. Click the General tab and select Bayesian Classifier from the Web Page Type drop-down list box. Click Yes in the confirmation dialog box.
5. Click the Measure Sets tab.
6. Select any regression measure sets in the Measure Sets pane and click Remove. Bayesian Classifier web pages do not use regression measure sets. For the sample datamart, remove the Revenue Generated measure set. A confirmation dialog box appears.
7. Click OK.
8. Click OK to complete configuring your Bayesian Classifier web page.

9. Open your topic and assign the new web page to the Bayes and Bayes – Add Dimensions nodes.
10. Add access permissions for your new web page.
11. Click OK to save the changes.
12. Refresh the Analytic Server. For more information, refer to "Refreshing the Analytic Server," on page 156.
13. Create a default saved report for this navigation node. For more information, refer to "Adding a Default Saved Report," on page 155.
14. View the web page.

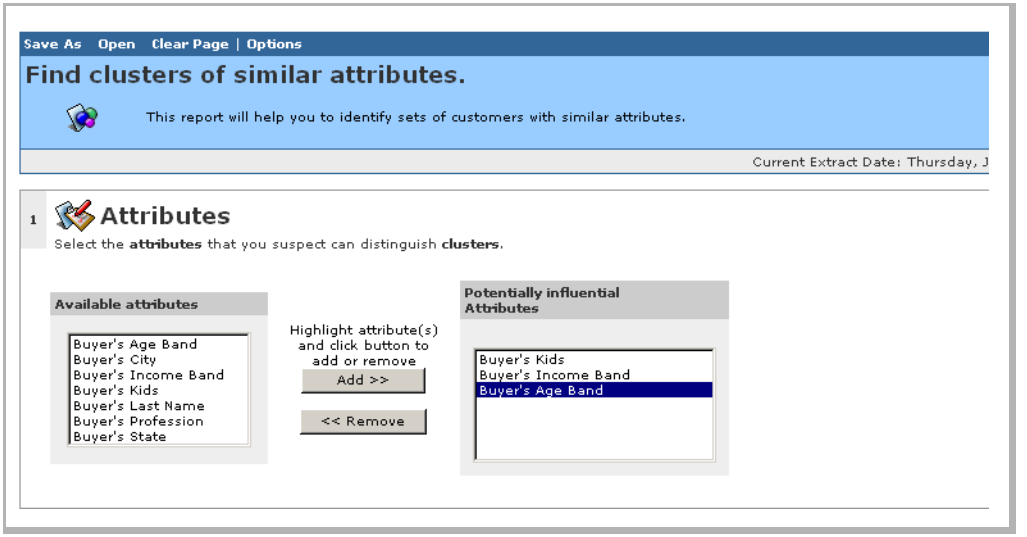
COMMUNITY CLUSTERS

The **Community Clusters** web page allows you to find correlations between various business parameters through a sophisticated data-mining algorithm.

Like an Influences web page, a Community Clusters web page has a primary dimension. Unlike an Influences web page, however, a Community Clusters web page does not have target attributes that it attempts to predict. Instead, Community Clusters reports identify groupings within the attributes that you select.

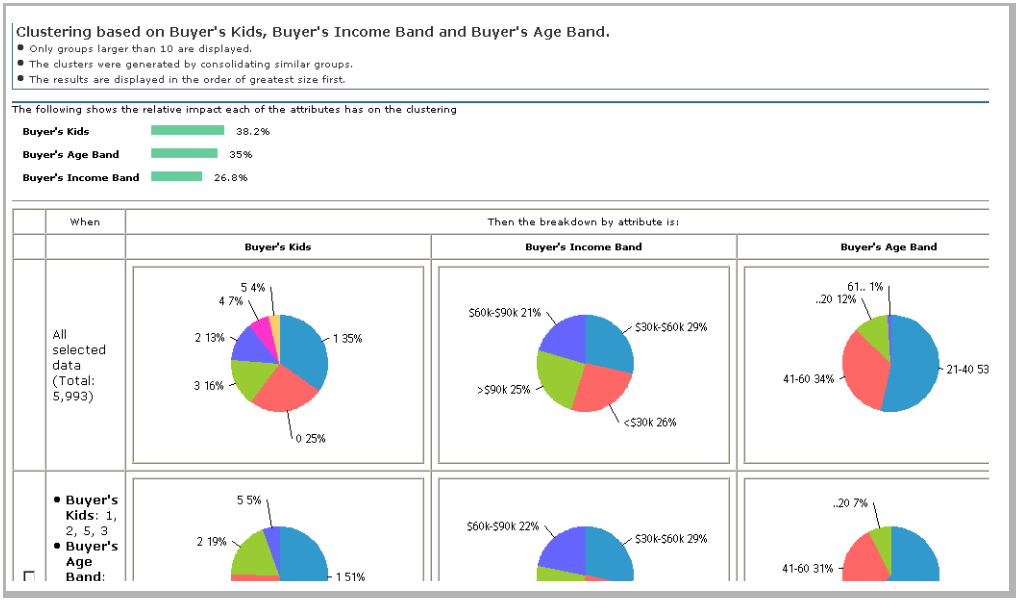
For the sample datamart, the Community Clusters web page appears as follows.

FIGURE 99: A COMMUNITY CLUSTERS WEB PAGE



The web page allows you to create the following report.

FIGURE 100: A COMMUNITY CLUSTERS REPORT



The Community Clusters web page makes use of a Clustering measure set to perform its analysis.

To create a Clustering measure set:

1. In Analytic Manager, navigate to the Measures folder and select the Measure Sets folder.
2. Right-click and select New Measure Set.The Measure Set dialog box opens.
3. Enter a name and description for the Measure Set. For the sample datamart, enter the name **`Individual Clustering`**.
4. Select Clustering from the Type drop-down list box.
5. Select the Count measure from the drop-down list box. For the sample datamart, select the Count of Customers measure that you created.
6. Click OK to define the measure set.

CREATING A COMMUNITY CLUSTERS WEB PAGE

Community Clusters web pages are configured in a similar way as Influences web pages. If you are using the same primary dimension as you did for your Influences web page, you can start with the completed Influences web page.

CREATING A COMMUNITY CLUSTERS WEB PAGE

1. Duplicate the Bayesian Classifier web page. For the sample datamart, duplicate the Sales Bayes web page and enter the name **`Sales Clusters`** for the new web page.
2. Open your new web page and change the Web Page Type to Community Clusters. A confirmation dialog box opens. Click Yes.
3. Verify that all attributes and filters have been copied correctly. For the sample datamart, use the same primary dimension and leave the attributes and filters unchanged.
4. Click the Measure Sets tab.
5. Remove the classification measure set from the Measure Sets pane by selecting it and clicking the Remove icon.
6. Add the clustering measure set by dragging it from the upper pane of the Object Gallery to the Measure Sets pane. For the sample datamart, add the Individual Clustering measure set.
7. Click OK to complete defining the web page.
8. Open your topic and assign the new web page to the Community Clusters and Community Clusters – Add Dimensions nodes.
9. Click OK to save the changes.
10. Refresh the Analytic Server. For more information, refer to "Refreshing the Analytic Server," on page 156.
11. Create a default saved report for this navigation node. For more information, refer to "Adding a Default Saved Report," on page 155.
12. View the web page.

COMPLETING TOPIC DEFINITIONS

You should now have web pages assigned to all nodes of the your topic except for the Dashboard and Alert nodes. Verify that all nodes are correctly configured and that you have granted access permission to all nodes.

After you have completed the Sales Analysis topic for the sample datamart, complete the definition of the Marketing Analysis topic. Add filters to the Marketing Reports web page and then create all of the additional web pages described in this chapter. You can create the Marketing Analysis web pages by modifying copies of the web pages that you created for the Sales Analysis topic. Make the following modifications to copies of the Sales Analysis web pages:

- From each web page, remove all attributes that refer to the Sales_Person dimension role, sales attribute layout, or all attributes that refer to the Sales_Person dimension role and the sales_order_no degenerate dimension.
- From each web page, remove the Sales Layout measure layout from each web page and add the Marketing Layout measure layout. This is not applicable to Influences or Community Clusters web pages.
- Create measures and measure sets for the Influences and Community Clusters web pages that use the Marketing fact table.
- Replace the measure sets in the Influences and Community Clusters web pages with the new Marketing measure sets.

When you complete these web pages, assign them to the appropriate nodes of the Marketing Analysis topic and grant node-access permissions to at least one user. Create default saved reports for the nodes in the topic. Refresh the Analytic Server and try the completed topic.

Creating Web Pages for Lists and Campaigns

This chapter describes how to create web pages for **List** and **Campaign Management**, update the datamart for list management, and complete a basic Campaign Management topic.

The Group and Individual Campaign web pages enable you to view the datamart from the perspective of demographic dimensions, that is, the dimensions that you assign to the Indiv and Group dimension roles. Depending on how you configure them, campaign web pages can provide either list-management or campaign-management capabilities. List management enables you to generate lists that contain members who have records in one of these demographic dimensions. Campaign management enables you to generate campaigns that are targeted to members of those demographic dimensions.

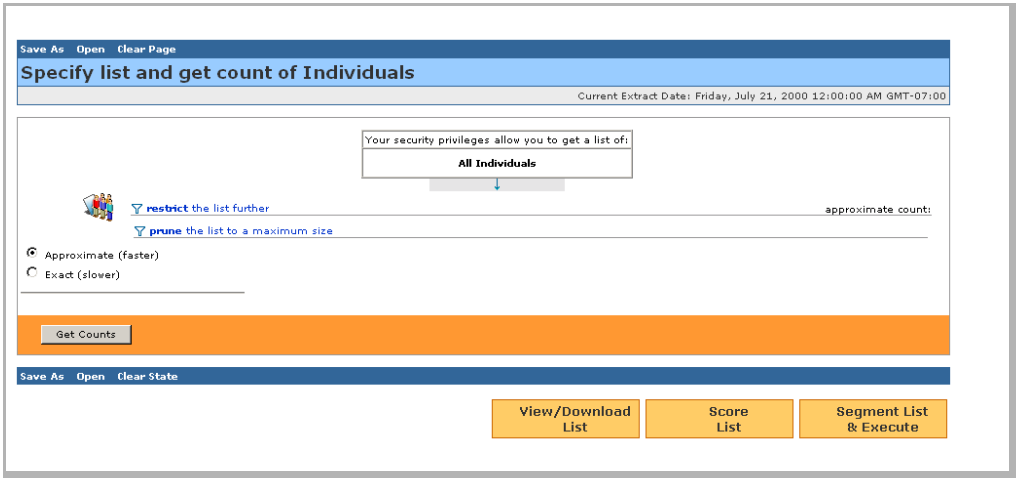
Creating Web Pages for List Management

When you create a web page for list management, users see a **List Manager** web page. Users can use this web page to generate lists by choosing filters and specifying selection criteria. You create a web page for list management by creating:

- A Group Campaigns web page for lists and assigning that web page to a Group List Manager navigation node.
- An Individual Campaigns web page for lists and then assigning that web page to an Individual List Manager navigation node.

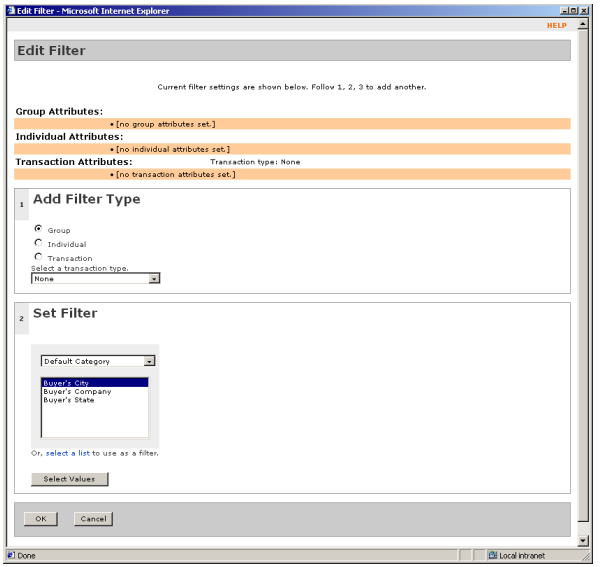
The preliminary List Manager web page appears as follows.

FIGURE 101: A LIST-MANAGEMENT WEB PAGE



The List Manager web page provides the means to configure filters to restrict the list to individuals who meet certain criteria. This configuration is done using the Filter Configuration screen.

FIGURE 102: THE FILTER-CONFIGURATION SCREEN

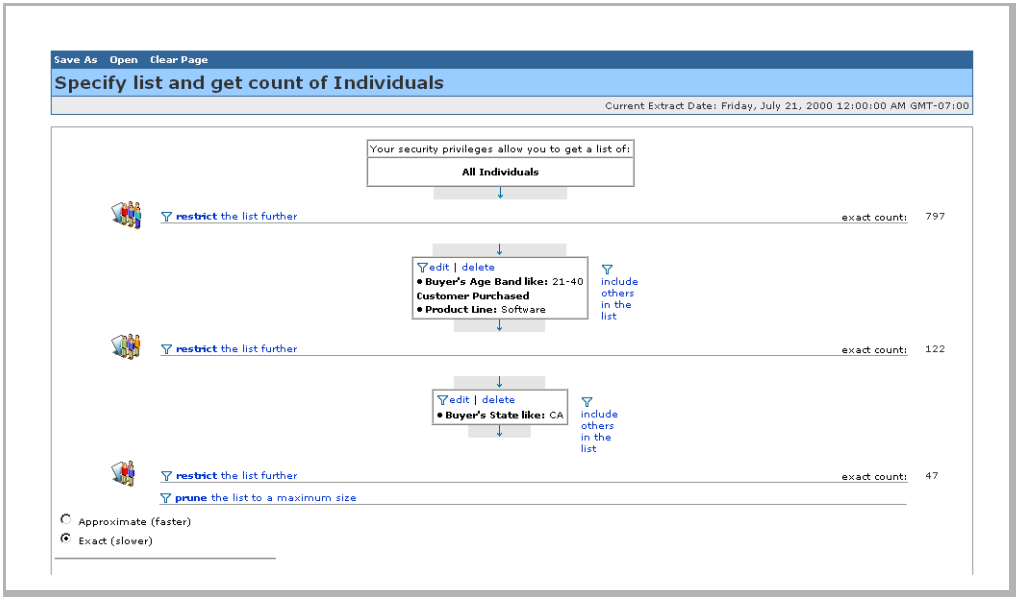


The Filter Configuration screen enables you to filter on:

- Individual attributes such as customer age or profession
- Group attributes such as company name or city
- Fact table measurements for individuals (such as purchases made or revenue generated; this requires a Transaction Filter).

Once the filters are configured, they are listed on the List Manager web page as follows.

FIGURE 103: A LIST-MANAGEMENT WEB PAGE WITH FILTERS DEFINED



You can then generate a list of filtered data which can be used in a campaign, exported, or used in other web pages.

FIGURE 104: A LIST GENERATED BY THE LIST MANAGER

List members as of July 16, 1999 12:00:00 AM PDT

#	Buyer's Last Name	Buyer's First Name	Buyer's Age	Buyer's Income	Buyer's Profession
1	Holte	Peter	41-60	$30k-$60k	Software Engineer
2	DiNucci	Heidi	41-60	$30k-$60k	Software Engineer
3	Giles	Ramon	..20	$30k-$60k	Software Engineer
4	Shoham	Doug	21-40	$30k-$60k	Software Engineer
5	Nilsson	Doug	21-40	<$30k	Finance Manager
6	Zhao	Karen	..20	<$30k	Finance Manager
7	Camplejohn	Tamara	..20	<$30k	Finance Manager
8	Zhao	Heidi	41-60	<$30k	Finance Manager
9	Perna	Steve	21-40	<$30k	Database Admin
10	Benson	Elliot	..20	$30k-$60k	Product Manager

You create web pages for list management by doing the following:

- Identifying Group and Indiv Dimension Roles
- Creating Transaction Filters
- Defining Transaction Filter Filters
- Creating a Web Page for Campaigns
- Creating a Web Page for Group Campaigns

IDENTIFYING GROUP AND INDIV DIMENSION ROLES

Before you use list management, you have to decide what dimensions to use for the group role. (If you are generating lists from a single dimension, then you have to give that dimension the group role.) If you plan to create lists of individuals, you must also decide what dimensions to use for the indiv dimension role.

Every record within the indiv dimension role needs to be associated with a unique record within the group role. For example, if every customer works for a single company, then you can use the customer dimension for the indiv role and the company dimension for the group role.

Candidates for individuals and groups are often clear from the structure of the data and the lists that need to be generated. For the sample datamart, Individual_base is a candidate for the **indiv** dimension role and Household_base a candidate for the group dimension role. (You assigned these base dimensions to the **indiv** and **group** dimension roles in Chapter 3, "Creating a Datamart.")

CREATING TRANSACTION FILTERS

Transaction filters enable you to create lists based on fact-table values. Transaction filters can be used to either filter by dimension attributes or fact-table measurements.

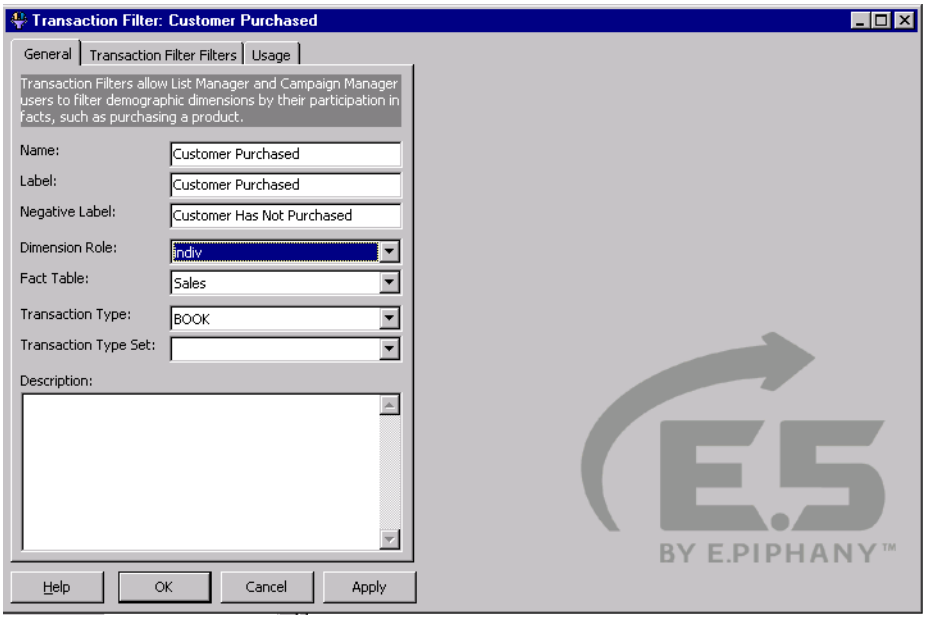
When you filter by dimension attributes, the members of the generated list have a fact table entry with the dimension attribute values that the user has chosen. For example, you can filter a list of customers by product purchased. The user of the web page specifies a product, and the transaction filter then restricts the list to those customers who have a fact table entry for a purchase of that product.

When you filter by fact table measurements, the members of the generated list have fact-table measure values within a range that you specify. For example, you can filter a list of customers by revenue generated by specifying a range of values, and the transaction filter then restricts the list to those customers who have generated revenue within that range.

CREATING A TRANSACTION FILTER

1. In Analytic Manager, expand the Presentation folder and select the Transaction Filters subfolder.
2. Right-click and select New Transaction Filter. The Transaction Filter dialog box appears.

FIGURE 105: THE TRANSACTION FILTER DIALOG BOX: GENERAL TAB



3. Enter a name and description for the transaction filter. For the sample datamart, enter the name Customer Purchased. Change the Negative Label to Customer has not Purchased.
4. Select a Dimension Role. Depending on whether you are filtering lists of individuals or groups, select either indiv or group. For the sample datamart, select indiv.

5. Select the Fact Table that contains the transactions that you are using for filtering. For the sample datamart, select the Sales fact table.
6. Select a Transaction Type or Transaction Type Set. Select the transaction types that you are using for filtering in the selected fact table. For the sample datamart, select a Transaction Type of BOOK.
7. Click OK to finish defining the transaction filter.

Defining Transaction Filter Filters

After you define a transaction filter, you add a **Transaction Filter Filter.** *A* Transaction Filter Filter enables you t*o* specify which of the transactions are to be used for filtering; that is, the transactions in the transaction filter (Booked Sales for the sample datamart) are themselves filtered. You can do this second level of filtering either on an attribute or a measure.

When filtering is based on an attribute, only transactions that involve the selected attribute are considered. For example, if Product Line is used for filtering, only transactions that involve the product lines selected by the user are considered.

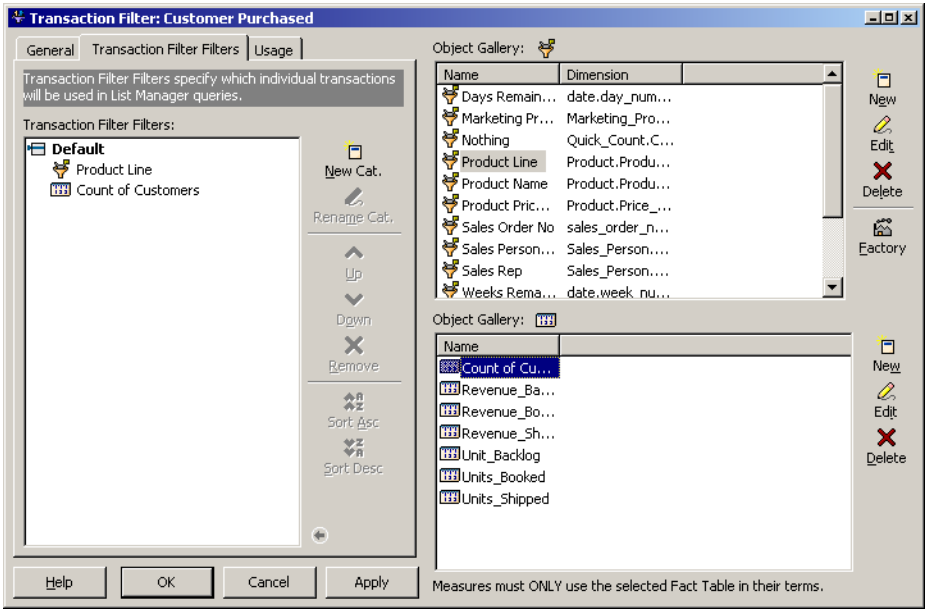
When filtering is based on a measure, only transactions that result in certain measure values are considered. For example, if Revenue Booked is used for filtering, then only transactions that involve individuals or groups who have booked revenue within a range selected by the user are considered.

Defining a Transaction Filter Filter

1. Open the Customer Purchased transaction filter that you created in the previous section. The Transaction Filter dialog box appears.
2. Click the Transaction Filter Filters tab.
3. Select an attribute to use for filtering by dragging it from the upper pane of the Object Gallery to the Transaction Filter Filters pane on the left. For the sample datamart, drag the Product Line attribute to the Transaction Filter Filters pane.

4. Select a measure to use for filtering by dragging it from the lower pane of the Object Gallery to the Transaction Filter Filters pane on the left. For the sample datamart, drag the Revenue Booked measure to the Transaction Filter Filters pane.

FIGURE 106: THE TRANSACTION FILTER DIALOG BOX: TRANSACTION FILTER FILTERS TAB



5. Click OK to complete defining the transaction filter.

CREATING WEB PAGES FOR CAMPAIGN MANAGEMENT

You can configure a web page for campaign management by creating either an Individual Campaigns web page for lists and assigning it to an Individual Campaigns navigation node or creating a Group Campaigns web page and assigning it to a Group Campaigns navigation node.

When you configure a web page for campaigns, users see a Campaign Manager web page. They can use this web page to generate campaigns by creating campaign segments and cells, selecting or creating output formats, saving the campaign information in the Report Gallery, and scheduling a date or set of recurrences on which the campaign is to run.

CREATING A WEB PAGE FOR CAMPAIGNS

The Group Campaigns and Individual Campaigns web pages enable users to view the datamart from the demographic dimensions perspective– the dimensions that you assign to the indiv and group dimension roles.

CREATING A WEB PAGE FOR INDIVIDUAL CAMPAIGNS

1. In Analytic Manager, expand the Presentation folder and select the web pages subfolder.
2. Right-click and select New Web Page. The Web Page dialog box appears.
3. Enter a name and description for the web page. For the sample datamart, enter **`Indiv Campaigns and Lists`**.
4. For Web Page Type, select Individual Campaigns.
5. Click the Attributes tab.
6. Click the Campaign tab and add the campaign and cell attributes from the Object Gallery.

NOTE: *These attributes are not used when the web page appears on a List Manager navigation node, but they are required for Campaign Manager nodes.*

When you use the web page on a Campaign Manager node, include the following attributes.

Table 5: Campaign Manager Node Attributes

Default Attribute Name	Dimension Role Column
Campaign Code	campaign.campaign_code
Campaign ID	campaign.campaign_ID
Campaign Label	campaign.campaign_label
Cell Label	cell.cell_label
Cell Size	cell.cell_size
Cell Treatment Code	cell.treatment_code

For the sample datamart, add the attributes described in the above table. You can also create an attribute layout named Campaigns, and add the layout. For more information, refer to "Creating Attribute Layouts," on page 111.

7. Click Preview and add demographic attributes from the Object Gallery.

   When you generate a list, you can select the attributes whose values are to be displayed in list previews. You can select any of the attributes that you add here. For the sample datamart, select the Indiv Buyer and Group Buyer attribute layouts, or all available attributes from the indiv and group dimension roles.

8. Click Filter and add demographic filter attributes. You can filter lists by any of the attributes that you add here. For the sample datamart, select all available attributes from the indiv and group dimension roles.
9. Click the Transaction Filters tab.
10. Add transaction filters by dragging them from the Object Gallery to the Transaction Filters pane on the left. For the sample datamart, add the Customer Purchased transaction filter.
11. Click OK to finish defining the web page.

CREATING A WEB PAGE FOR GROUP CAMPAIGNS

1. Create a transaction filter for groups. The configuration for this transaction filter is similar to a transaction filter for individuals, except you use the group dimension role instead of the indiv dimension role. For the sample datamart, name this filter **`Group Purchased`**.
2. Create a Group Campaigns web page. For the sample datamart, enter the name **`Group Campaigns and Lists.`**
3. Add Campaign, Preview, Filter and Preview Filter attributes to the web page. For a Group Campaigns web page, you can only add attributes from the group dimension role to the Preview attribute role.
4. Add transaction filters to the web page. You can add transaction filters for both group and indiv dimension roles to a Group Campaigns web page. For the sample datamart, add both the Customer Purchased and the Group Purchased transaction filters.

CREATING A BASIC CAMPAIGNS TOPIC

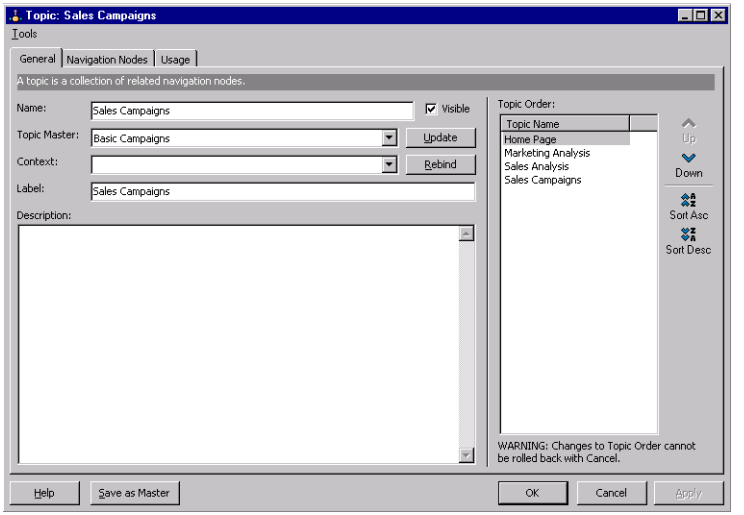
After you create a new web page, create a corresponding Basic Campaigns topic.

CREATING A BASIC CAMPAIGNS TOPIC

1. In Analytic Manager, expand the Presentation folder and select the Topics subfolder.

2. Right-click the Topics icon and select New Topic. The Topic dialog box appears.

FIGURE 107: THE TOPIC DIALOG BOX: GENERAL TAB



3. Enter a name for the topic. For the sample datamart, enter the name **`Sales Campaigns`**.
4. Select the Basic Campaigns topic master.
5. Click Apply.
6. Click the Navigation Nodes tab.
7. Assign your new Indiv Campaigns and Lists web page to the List – Specify – Indiv and List – Preview – Indiv nodes.
8. Assign your new Group Campaigns and Lists web page to the List – Specify – Group and List – Preview – Group nodes.
9. Select all of the navigation nodes and grant access to at least one user (see "Configuring a New Topic and Adding Security," on page 155).

10. Click OK to save your changes.

NOTE: *List Manager and Campaign Manager web pages are included in the Basic Campaigns topic master.*

UPDATING THE DATAMART FOR LIST MANAGEMENT

After you create a web page for list management, you add list management to the datamart. To update the datamart for list management, you do the following:

- Configuring Extraction for List Management
- Building the Sampling Table
- Verifying that Fact Table Columns are Queryable
- Adding Clusters to a Fact Table
- Enabling List and Campaign Management
- Running the Extraction Job

CONFIGURING EXTRACTION FOR LIST MANAGEMENT

List management requires a special fact table, called Ind_Group_Joiner, that specifies which individuals belong to each group. You populate this table during extraction. To populate this table, you create extraction commands and Semantic extraction steps.

CREATING THE SQL EXTRACTION GROUP FOR IND_GROUP_JOINER

1. Expand the Extraction folder in your EpiCenter, and double-click the Jobs icon. Open the job that you created in Chapter 4, "Extracting Data into the Datamart." For the sample datamart, the job is called Extractorama.
2. Click the Job dialog box Extraction Steps tab.
3. Right-click your fact extraction group (Get Fact Data for the sample datamart) in the Object Gallery and select Group from the New submenu. Create a new global extraction group called IndGroup Extract.
4. Right-click your new extraction group and select Truncation Step from the New submenu. Create a truncation step for the Ind_Group_Joiner fact table.
5. Right-click your new extraction group (IndGroup Extract) and select Extraction Command from the New submenu. Analytic Manager displays the Extraction Wizard.
6. You can use the Extraction Wizard to create the extraction command or you can use the standard dialog box interface. See the *Datamart Guide* for details on the Extraction Wizard. Click the Create Extraction Command with Editor button to use the dialog box interface.
7. Select Write SQL by Hand.

   The instructions that follow are tailored specifically to the dialog box interface with the Write SQL by hand radio button selected.
8. Enter a name. For the sample datamart, enter the name **`Get IndGroup Facts`**.
9. Click Populates Fact Table and select Ind_Group_Joiner from the drop-down list box.
10. Click the SQL Statement tab.

11. Click Template to display a template for this fact table. The template will be similar to the following:

```
SELECT
   <YOUR EXPRESSION>                              ss_key,
   $$TO_EPIDATE[ <YOUR EXPRESSION> ]              date_key,
   <YOUR EXPRESSION>                              transtype_key,
   $$NVL[ <YOUR EXPRESSION> ~,~ 'UNKNOWN' ]       group_sskey,
   $$NVL[ <YOUR EXPRESSION> ~,~ 'UNKNOWN' ]       indiv_sskey,
   $$NVL[ <YOUR EXPRESSION> ~,~ 0 ]               member
FROM
   <YOUR TABLE>
```

12. Fill in expressions that associate the sskey of each individual with the sskey of the group to which the individual belongs. An individual must not belong to more than one group.
13. Fill in the remaining SQL. For the sample datamart, use the following code:

```
SELECT
   C.Contact_Id                                   ss_key,
   $$TO_EPIDATE[$$DBNOW]                          date_key,
   1                                              transtype_key,
   $$NVL[ H.Household_ID ~,~ 'UNKNOWN' ]          group_sskey,
   $$NVL[ C.Contact_Id ~,~ 'UNKNOWN' ]            indiv_sskey,
   1                                              member
FROM
       Contact_table C, Household_table H
WHERE
       H.Household_ID = C.Household_Id
```

14. Click OK to define the extraction step.
15. To create the Semantic extraction step for Ind_Group_Joiner, open the Extractorama Job dialog box (if it is not still open).
16. Right-click the fact semantics group (Fact Semantics for the sample datamart) in the Object Gallery and from the New menu, select Semantic Instance. The Semantic Instance dialog box appears.
17. Under References, select the Fact Table radio button.
18. Select a fact table from the Fact drop-down list box. For the sample datamart, select Ind_Group_Joiner.
19. Select an appropriate semantic type from the Semantic Type list box. For the sample datamart, use Statelike/Error on Stage Duplicates.

20. Click OK to define the semantic instance.
21. Verify that the E.5.0.1 Builders group, which includes the AggBuild and MomBuild steps, has been added to the extraction job.
22. Click OK to finish setting up the job.

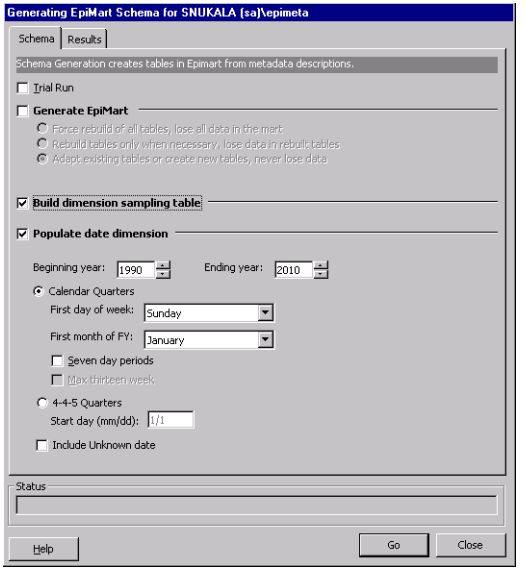
Building the Sampling Table

The List Manager and Campaign Manager use a special sampling table. You create the sampling table when you first generate the EpiMart schema. If you did not create the table, use the following procedure.

BUILDING THE SAMPLING TABLE

1. In Analytic Manager, from the EpiCenter menu, select Generate Schema. The Schema Generation dialog box appears..

FIGURE 108: SCHEMA GENERATION



2. Uncheck Trial Run and Generate EpiMart.
3. Check Build dimension sampling table.
4. Click Go to generate the sampling table.

VERIFYING THAT FACT TABLE COLUMNS ARE QUERYABLE

List management transaction filters use information from the main fact tables. Therefore, make sure that the appropriate fact table columns of your main constellation can be queried.

VERIFYING IF FACT TABLE COLUMNS ARE QUERYABLE

1. In Analytic Manager, navigate to the Schema folder and select the Facts subfolder.
2. Double-click to open the fact table you used in List Manager transaction filter. For the sample datamart, this is the Sales fact table.
3. Right-click and select Edit. The Fact Table dialog box appears.
4. Under Fact Columns, select a column that is used in a transaction filter. For the sample datamart, select the Revenue column.
5. Check the Allow List Filter check box if it is not checked for the column in the grid control.
6. Repeat steps 1-5 for other queried fact columns. For the sample datamart, no other fact columns are queried.

ADDING CLUSTERS TO A FACT TABLE

To run list management queries on your fact table, add at least one cluster to every fact table to be queried. In Chapter 3, "Creating a Datamart," you added a cluster to your fact tables.

ADDING CLUSTERS TO A FACT TABLE

1. Open the Sales fact table. The Fact Table dialog box appears.
2. Click the Clusters tab.
3. Click New. The Select Dimension Role dialog box appears.
4. Select dimension roles for clustering. Clusters speed up List Manager queries that involve those dimension roles, so add dimension roles that are likely to be used for filtering in List Manager queries.

For the sample datamart, only the Product dimension role is used in transaction filters. You do not have to add additional dimension roles.

5. Click OK. The Select Dimension Role dialog box closes and the Fact Table dialog box appears.
6. Click OK to complete setting up the table for the List Manager.

ENABLING LIST AND CAMPAIGN MANAGEMENT

In order to use list and campaign management features, you have to enable the MomentumEnabled flag.

ENABLING THE MOMENTUMENABLED FLAG

1. In Analytic Manager, expand the Configuration folder and navigate to the Settings subfolder.
2. Double-click Settings. Analytic Manager displays the Configuration dialog box.
3. Scroll down to the momentum_enabled key, click in the value column, and set the value to one.
4. Click OK.

RUNNING THE EXTRACTION JOB

To run the extraction job, refer to "Running the Extraction Job," on page 104. When the extraction job completes with no errors, you have successfully set up the List Manager.

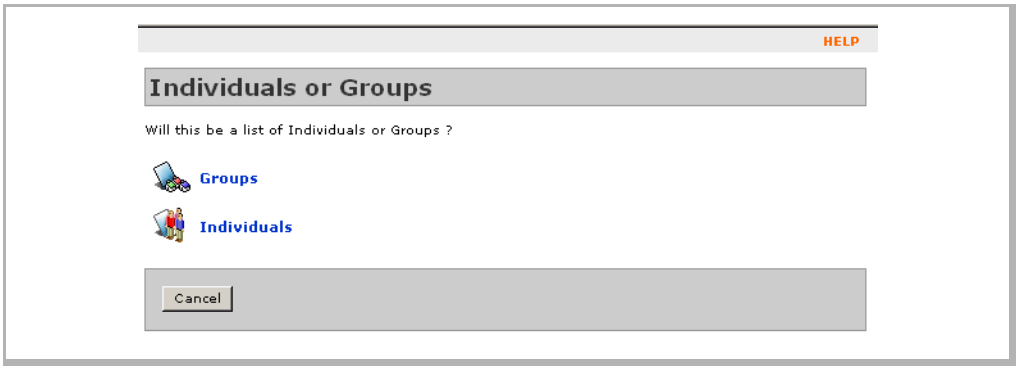
USING LIST-MANAGEMENT WEB PAGES

The Individual or Group Campaigns web page that you create for list management display List Manager web pages to end users. To view the list-management web pages, refresh the Analytic Server (see "Refreshing the Analytic Server," on page 156) and navigate to your List Manager web page. When the Analytic Server is refreshed, the Home Page displays the Basic Campaigns topic.

To reach the Individual List Manager web page:

1. On the home page, click the name of new Basic Campaigns topic (Sales Campaigns) to go to the Basic Campaigns start page.
2. Click Make a List to display the path selector popup window.

FIGURE 109: SELECTING AN INDIVIDUAL OR GROUP LIST



3. Click the Individuals link in the popup window to go to the Specify List and get Count of Individuals page.

For additional information on list filtering and generation, refer to the online help.

Adding List Membership Filters to Other Web Pages

After you have set up your list-management web pages, you can use those pages to generate lists. If you want yo reuse these lists as filters in other web pages, you can add list membership filters to your list management web pages. This is useful when users want to view a different slice of the current data. For example, you can use the List Manager to generate and save a list of promising customers, and then use the saved list as a filter on any web page in your Reporting and Analysis topic.

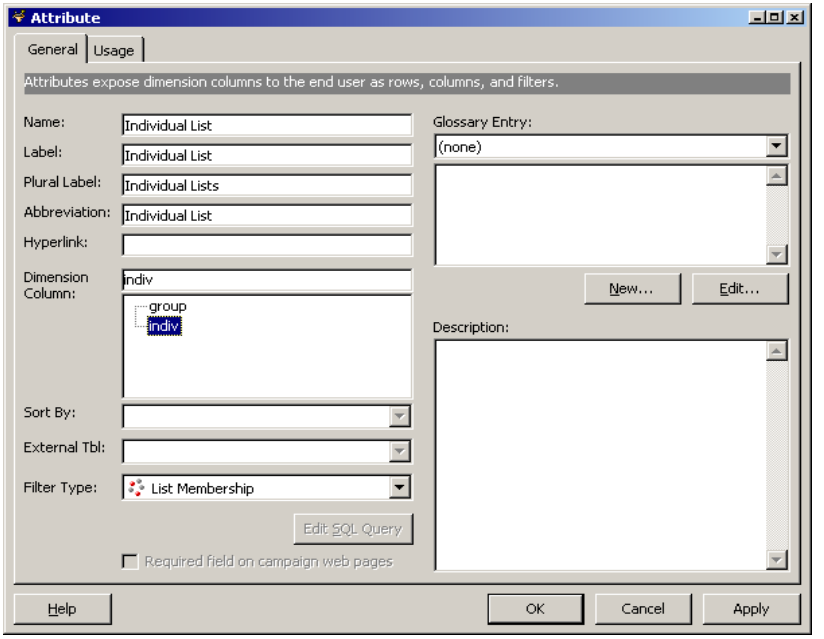
Adding List Membership Filters

1. In Analytic Manager, navigate to the Presentation icon and select the Attributes subfolder.
2. Right-click and select New Attribute. The Attribute dialog box appears.
3. Enter a name for the attribute. For the sample datamart, enter the name **`Individual List`**.
4. Select List Membership as the Filter Type. When you select List Membership, the choices for Dimension Column change to indiv and group (see Figure 110). Select one, depending on whether you will be generating lists of individuals or groups. For the sample datamart, start with the indiv dimension role.
5. Click OK to define the filter.
6. Define another list membership filter for the group dimension role. For the sample datamart, enter **`Group List`**.
7. Add both list membership filters to all of your analytic web pages.

   Add these filters to a web page by clicking Filters under the Attributes tab of the web page and then dragging the filters from the Object Gallery. You can also add these filters to attribute layouts. For the sample datamart, add the Individual List filter to the Indiv Buyer attribute layout and add the Group List filter to the Group Buyer attribute layout.

.

FIGURE 110: CREATING A LIST MEMBERSHIP FILTER



8. Refresh the Analytic Server (see "Refreshing the Analytic Server," on page 156) and view the new filters in your web pages.

COMPLETING THE BASIC CAMPAIGNS TOPIC

This section describes how to complete the Basic Campaigns topic by adding campaign-management features. You configure a topic for campaigns by doing the following:

- Creating Output Processors
- Configuring Campaign Nodes
- Configuring the Analytic Nodes
- Configuring External Links

- Configuring Campaign Extraction
- Enabling the Campaign Queue
- Starting the Scheduler Service

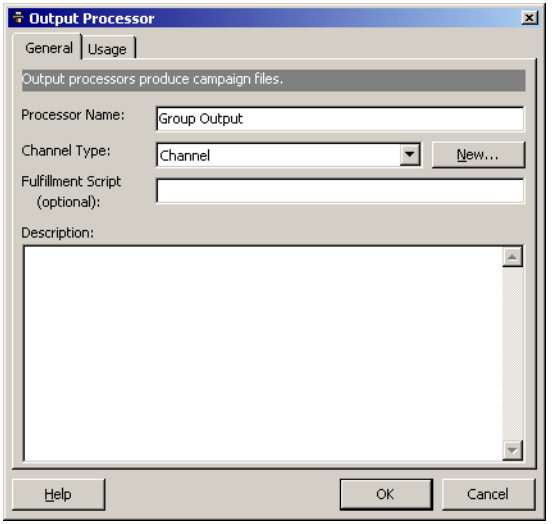
CREATING OUTPUT PROCESSORS

<<Introduction required>>

CREATING OUTPUT PROCESSORS

1. In Analytic Manager, navigate to the Security/Storage folder and select the Output Processors subfolder.
2. Right-click Output Processors and select New Output Processor. The Output Processor dialog box appears.

FIGURE 111: THE OUTPUT PROCESSOR DIALOG BOX



3. Enter a name for the output processor. For the sample datamart, enter the name **`Group Output`**.
4. Select a Channel Type. The channel is an identifier for the campaign fulfillment channel. For the sample datamart, accept the default Channel type.
5. Click OK to define the output processor.

CONFIGURING CAMPAIGN NODES

You can use the web pages that you created in the first part of this chapter (see "Creating Web Pages for List Management," on page 193) for many of the campaign-related nodes in your topic.

Open your Basic Campaigns topic (Sales Campaigns for the sample datamart) and assign the Individual Campaigns web page (Indiv Campaigns and Lists for the sample datamart) and the Group Campaigns web page (Group Campaigns and Lists for the sample datamart) to the nodes described in Table 6.

TABLE 6: BASIC CAMPAIGNS TOPIC NODES FOR CAMPAIGN WEB PAGES

NODES FOR INDIVIDUAL CAMPAIGNS WEB PAGE	NODES FOR GROUP CAMPAIGNS WEB PAGE
Campaign – Specify – Email – Indiv	Campaign – Specify – Email – Group
Campaign – Specify – Email – Sch – Indiv	Campaign – Specify – Email – Sch – Group
Campaign – Specify – Formatted – Indiv	Campaign – Specify – Formatted – Group
Campaign – Specify – Formatted – Sch – Indiv	Campaign – Specify – Formatted – Sch – Group
Campaign – Specify – Personalization – Indiv	Campaign – Specify – Personalization – Group
List – Preview – Indiv	List – Preview – Group
List – Specify – Indiv	List – Specify – Group

Configuring the Analytic Nodes

A **Basic Campaigns topic** allows you to make use of the results of High/Low Clusters and Basic Rows & Columns reports to select campaign targets. You can use the web pages that you created in Chapter 7, "Analyzing and Reporting Data," for these nodes.

Configuring Analytic Nodes

1. In Analytic Manager, navigate to the Presentation Folder and select the Topics subfolder.
2. Open the Sales Campaigns topic and click the Navigation Nodes tab.
3. Add a High/Low Clusters web page to the Insight – Highs & Lows node. For the sample datamart, use the Sales Highs and Lows web page.
4. Add a Rows & Columns web page to the Insight – Rows & Columns – Basic node. For the sample datamart, use the Sales Reports web page.
5. Click OK to close the Topic dialog box.

Configuring External Links

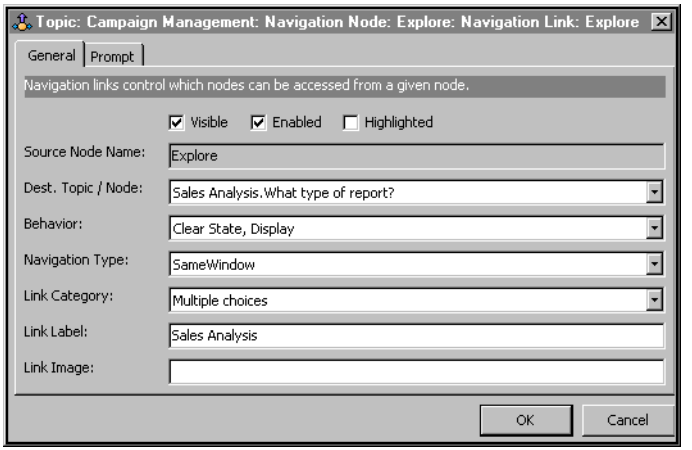
The Basic Campaigns topic includes links that allow you to go to other topics to do further data analysis.

Configuring External Links

1. In Analytic Manager, navigate to the Presentation Folder and select the Topics subfolder.
2. Open the Sales Campaigns topic and click the Navigation Nodes tab.
3. In the grid, select the YAH External Links – Both node and click the Edit icon. The Navigation Node dialog box appears.
4. Click the Links tab.

5. Add a link to another topic that you created by doing the following:
   a) Click Add Link. The Navigation Link dialog box appears.

FIGURE 112: THE NAVIGATION LINK DIALOG BOX



   b) Select a destination for the link from the Dest. Topic/Node drop-down list box.

      Since these links are intended to take a user to a different topic, select a node from one of the Reporting and Analysis topics. For the sample datamart, select the What type of Report node of the Sales Analysis topic.

   c) Select a Behavior for the link. The Carry State, Display allows you to perform further analysis on results from the current web page, and the Clear State, Display behavior clears any results from the current web page. For the sample datamart, select Clear State, Display.

   d) Select the Navigation Type. For the sample datamart, select SameWindow to display the new web page in the same window as the current web page.

   e) Change the Link Label field so that it describes the destination node. For the sample datamart, enter the name Sales Analysis.

   f) Click OK to finish editing the node.

   g) Add additional external links if required. For the sample datamart, add a link to the What Type of Report node of the Marketing Analysis topic.

6. Click OK. The Navigation Node dialog box appears.
7. Click OK. The Topic dialog box appears.
8. Verify that you have assigned a web page to every node.

NOTE: *You do not have to assign web pages to the nodes that have names beginning with* Email *if you have not installed E.piphany Campaign Management for E-Mail software. You also do not have to assign web pages to the nodes that have names beginning with* RP *if you have not installed E.piphany Personalization Platform software.*

9. Click OK to finish configuring the topic.

CONFIGURING CAMPAIGN EXTRACTION

When a campaign is run, campaign information is automatically saved in the EpiMart **backfeed tables.** To make this campaign data available to users, you have to configure your extraction job to copy the data from the backfeed tables to the campaign fact and dimension tables in your EpiCenter. You can do this configuration by adding the pre-defined E.5.0.1 Campaign Extraction group to your extraction job.

CONFIGURING CAMPAIGN EXTRACTION

1. Open your extraction job (Extractorama for the sample datamart). The Job dialog box appears.
2. Click the Extraction Steps tab.

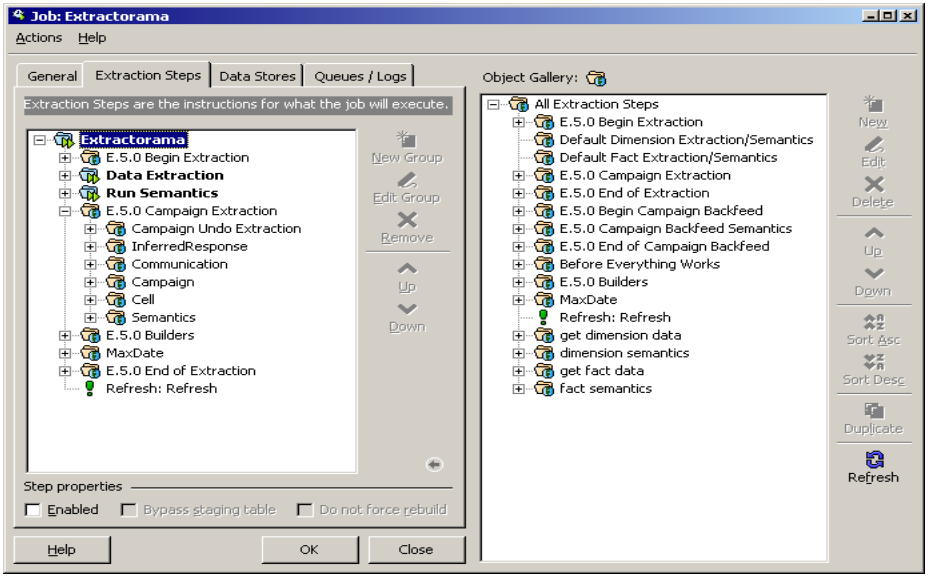
3. Find the Campaign Extraction group in the Object Gallery. This group contains the following steps:

TABLE 7: CAMPAIGN EXTRACTION STEPS

Campaign Undo Extraction steps	These extraction steps remove data for campaigns that have been cancelled by end users.
Extraction command steps	These extraction steps copy data from the backfeed tables into the Campaign, Cell, Communication, and InferredResponse staging tables. You do not have to change the SQL code in these steps unless you have made changes to the Campaign or Cell base tables.
Semantic extraction steps	These extraction steps merge data from the staging tables into the main Campaign, Cell, Communication, and InferredResponse datamart tables.

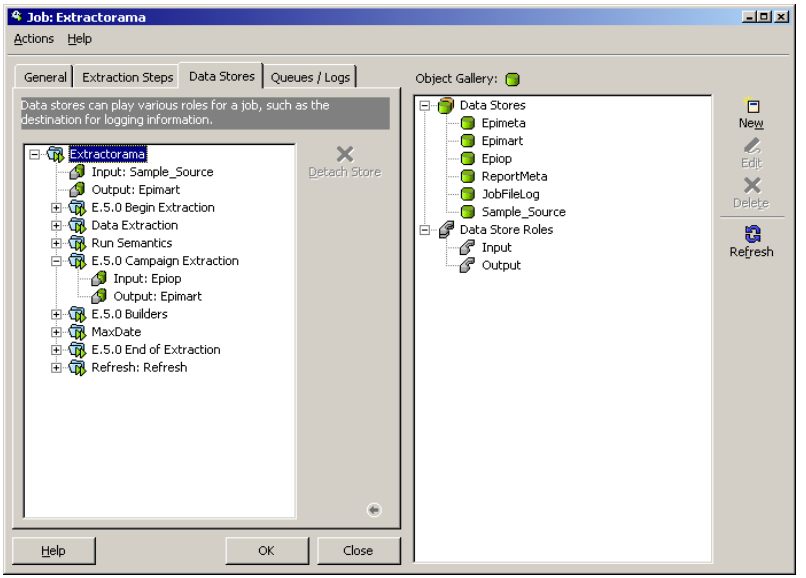
4. Drag the E.5.0.1 Campaign Extraction group from the Object Gallery to the root-level job step.

FIGURE 113: A JOB CONFIGURED FOR CAMPAIGN EXTRACTION



5. Use the Up button to move the Campaign Extraction group between the Run Semantics and E.5.0.1 Builders job steps.
6. Click the Job dialog box Data Stores tab.
7. Expand the Campaign Extraction group in the left pane.
8. The backfeed tables are located in the EpiOp database, so you have to specify that database as the input data store for the Campaign Extraction group. To set the proper input data store, drag the EpiOp data store from the Object Gallery to the Input data store role in the Campaign Extraction group.

FIGURE 114: DATA STORES FOR CAMPAIGN EXTRACTION



9. Click OK to finish configuring the job.

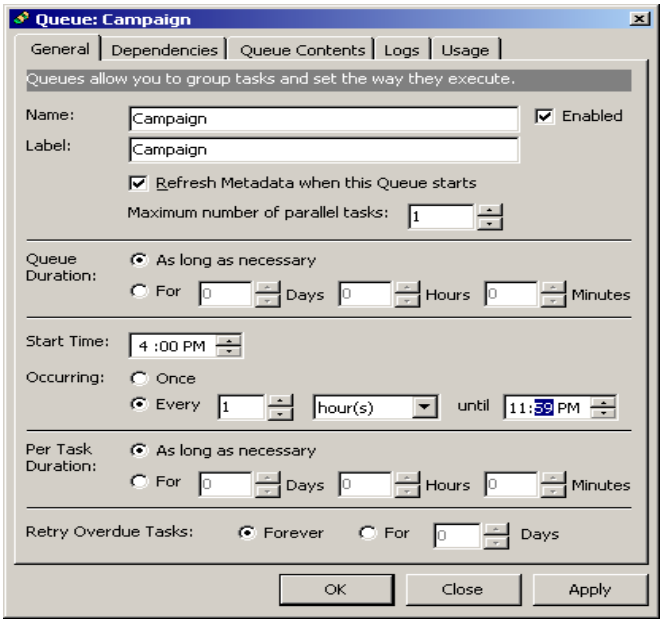
ENABLING THE CAMPAIGN QUEUE

When you schedule a campaign, EpiCenter scheduler runs the campaign based on the configuration of the appropriate Queue. By default, every EpiCenter contains a Campaign queue that is configured for basic campaign functionality.

ENABLING THE CAMPAIGN QUEUE

1. Expand your **Extraction** folder. Expand the **Queues** subfolder, then double-click **Campaign**. The Campaign: Queue dialog box appears.

FIGURE 115: THE QUEUE DIALOG BOX



2. Check **Enabled**.
3. Click **OK** to close the Queue dialog box. For more information on the Scheduler Queue configuration, refer to the *Datamart Guide*.

STARTING THE SCHEDULER SERVICE

The **Scheduler** service allows administrators to schedule individual and recurring extraction jobs and users to schedule long-running or recurring reports and campaigns.

STARTING THE SCHEDULER

1. Start the Scheduler:

   Start\Settings\Control Panel
2. Double-click Services. The Services dialog box appears.
3. Select the scheduler from the list of services. The scheduler service for an EpiCenter instance is named *InstanceName*_Scheduler, where *InstanceName* is the name of the EpiCenter instance.
4. Click the Start button to start the scheduler. Windows NT displays the Service Control panel with a message indicating that the service is attempting to start.
5. After a short wait, the Service Control dialog box disappears, and the Status column for your EpiCenter instance says "Started."

NOTE: *If there are errors in your datamart, you may get a dialog box saying that the service could not be started because of an internal Windows NT or Windows 2000 error. If you see this dialog box, review the logs to see what went wrong. The default installation places the Scheduler log files in the following directory:*

InstanceRootDir\Web\WWWroot\logfiles

6. Click Close to close the Services dialog box and close the Control Panel window.

After you successfully start the Scheduler service, refresh the Analytic Server (see "Refreshing the Analytic Server," on page 156) and try your new topic. You also have to create default saved reports for the nodes in the topic. (See "Adding a Default Saved Report," on page 155, for details.) Refer to the online help for instructions on creating and scheduling campaigns.

Building Aggregates

This chapter provides an overview of **aggregates** and describes how to define and build aggregates.

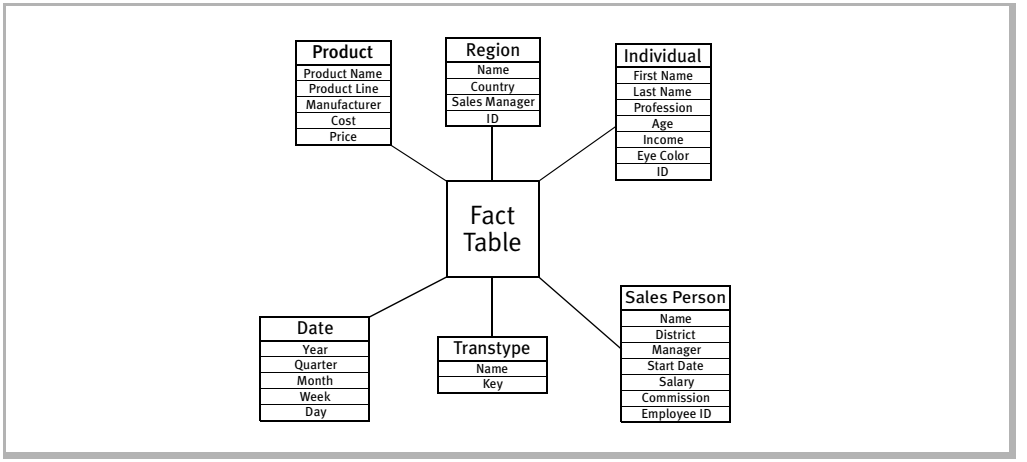
Aggregates are pre-computed summary data for frequently queried sets of dimension columns. Aggregates are built and updated during extraction. You can significantly improve the response time of your web pages by building aggregates in your EpiCenter. Any query that can be answered using summarized data is faster because much of the computation is already done. However, aggregate building slows down extraction time, so carefully select the aggregates that you want to build.

An aggregate is a data summary based on a collection of dimension columns that you select. For example, users can ask for monthly sales data based on product category and customer demographics. You can speed up response time for these queries by building an aggregate using these dimension columns.

DEFINING AGGREGATES

In Chapter 2, "Designing a Datamart," the star schema for the datamart was defined as follows.

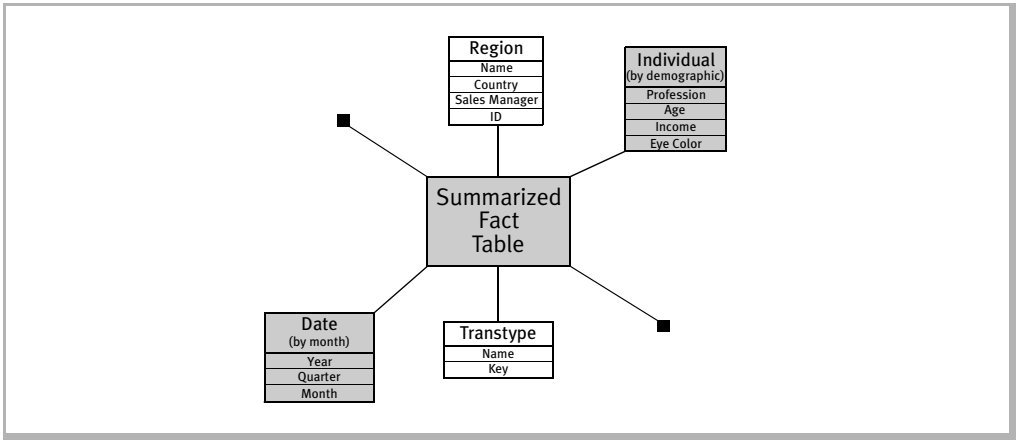
FIGURE 116: A TYPICAL STAR SCHEMA



You can answer many user queries without the level of detail that is found in the above star schema. For example, users can have queries that can be answered by considering only the demographic attributes of the Individual dimension, such as age and salary, and the month of a transaction. By building an aggregate, you can define a sub-star that contains just the data needed for these queries.

An **aggregate star** is an aggregated version of a star schema that is made up of fact aggregates and dimension aggregates arranged in the same way as the original star schema. You can define an aggregate star as follows.

Figure 117: An Aggregate Star



In the above aggregate star, two of the original dimensions have been replaced by sub-dimensions containing only some of the original columns, two dimensions have been included in their entirety, and two dimensions have been excluded. The fact table summarizes data from the original star schema based on these sub-dimensions.

Because this aggregate only considers the sub-dimensions shown in the aggregate star, this construction significantly speeds up query response time. The key to defining useful aggregates is to make them as small as possible (by defining sub-dimensions that have as few rows as possible), while ensuring that they are still useful for real-world queries.

Aggregate stars take up space. They must also be updated whenever new data is extracted. It can be tempting to build all possible aggregates, so that every query can be answered using an appropriate aggregate. However, a proliferation of aggregates can significantly increase extraction time. If aggregates are used indiscriminately, extraction can fail before completion. To keep extraction times within reasonable bounds, it is important to select aggregates carefully.

E.piphany recommends that you select aggregates based on actual usage. While you can build a few aggregates when you first set up your datamart, examining query logs for users' actual query patterns is the best way to find useful aggregation candidates. When you accumulate a significant query history, you can use the query history feature of the Fact and Dimension dialog boxes to generate aggregates that are optimized for the actual query patterns of users. For more information on the query history feature, refer to chapter 4, "Defining the EpiCenter Schema" of *Datamart Guide*.

Building Aggregates

You build aggregates by doing the following:

- Selecting Aggregate Dimensions
- Defining Dimension Aggregate Instructions
- Defining Fact Aggregate Groups
- Generating Fact Aggregate Instructions
- Running AggBuilder

Selecting Aggregate Dimensions

An **aggregate dimension** is an aggregated set of dimension columns that is available for use in an aggregate star. The columns in an aggregate dimension are a subset of the columns in the original dimension. For every combination of values for the columns in the aggregate dimension, if the original dimension has at least one row with those values in the corresponding columns, then the aggregate dimension has exactly one row with those values.

When you define aggregate dimensions, consider both column size and usage. E.piphany recommends that you select dimension columns that have a small number of possible values. A smaller aggregate dimension usually results in a smaller aggregate star, which gives improved query performance.

For example, if you aggregate your customer dimension based only on gender, then the resulting aggregate dimension is very small (most likely, it has rows for male, female, and unknown). On the other hand, if you include the street address column in your aggregate dimension, then the aggregate dimension may not be much smaller than the original dimension (since very few customers are likely to have the same street address).

To maximize the likelihood that a user query can be answered by an aggregate, make sure that frequently-queried dimension columns and combinations of dimension columns are included in aggregate dimensions.

For example, users may frequently look at customer data based on age and income, so it is useful to have an aggregate that includes these two columns. On the other hand, users are unlikely to look at customer data based on eye color, so an eye color column would not be very useful in an aggregate dimension (even though it probably has a small number of values).

The best way to determine usage is to analyze actual query statistics. There are no usage statistics for the sample datamart, so you have to estimate how often each column is likely to be queried.

To identify preliminary candidates for aggregate dimensions, you can perform some SQL queries on your datamart. What you are looking for is a collection of useful dimension attributes that produce a much smaller aggregate dimension.

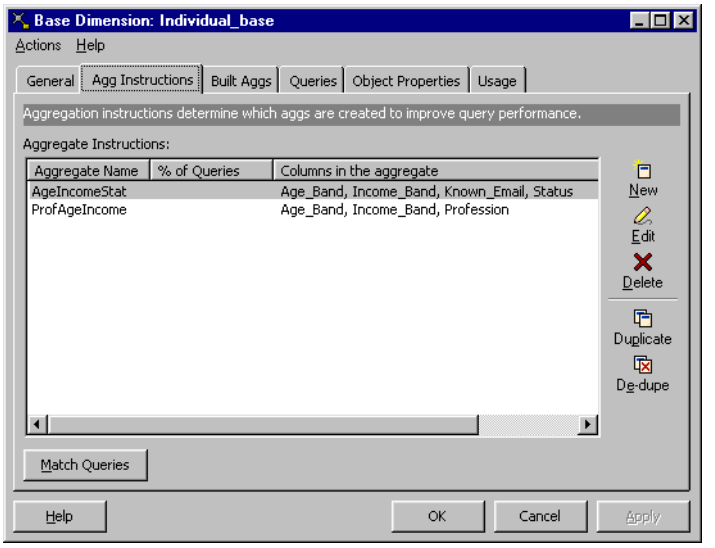
For example, if you count the rows in the Individual_0_A table in the sample datamart, you will find that there are 797 rows. Counting rows with distinct Income_Band values gives 5, since there are five income bands. You can find some other columns that you can include with Income_Band while still keeping the count low. If you count the rows with distinct Income_Band and Profession values you get a count of 29. If you include the Last_Name column as well, you get a count of 411. This is not much better than the original table. If you include Age_Band instead of Last_Name, you get a count of 80. This is better than the original table, but not by as much as you can like. This is the kind of aggregate dimension that you can build if you find that a lot of queries used just those attributes. E.piphany recommends that you build the dimensions only after analyzing the query logs.

For the sample datamart, build an aggregate Individual dimension with the Profession and Income_Band columns. This chapter describes how to define this aggregate dimension and then define two aggregate stars that include the dimension. Both of these aggregate stars also include the Quick_Count dimension and an aggregate date dimension for the calendar month. One of the stars includes the product dimension and one excludes it.

DEFINING DIMENSION AGGREGATE INSTRUCTIONS

1. In Analytic Manager, expand the Schema folder, and select the Base Dimensions subfolder.
2. Double-click the dimension that you want to work with. For the sample datamart, use the Individual_base dimension. The Base Dimension dialog box appears.

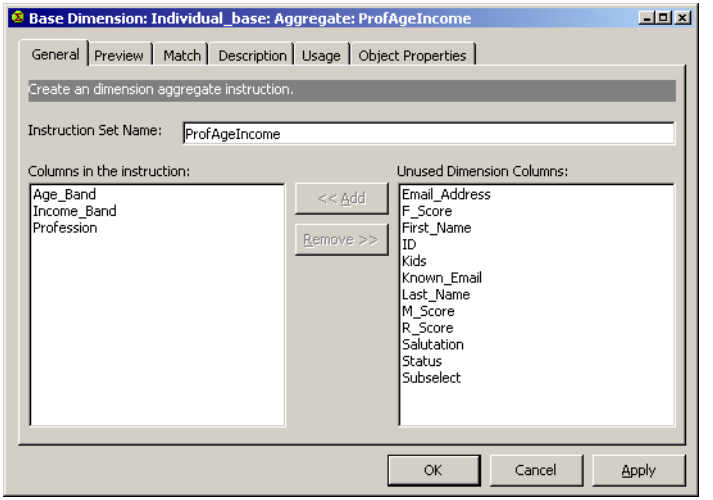
FIGURE 118: THE BASE DIMENSION DIALOG BOX: AGGREGATE INSTRUCTIONS TAB



3. Click the Agg Instructions tab.

4. Click New. The Base Dimension Aggregate dialog box appears.

FIGURE 119: THE BASE DIMENSION AGGREGATE DIALOG BOX



5. In the Instruction Set Name field, enter a name for the aggregate instruction. For the sample datamart, enter **`ProfAgeIncome`**.
6. Select the columns that you want to include in the dimension aggregate. For the sample datamart, select the Age_Band, Income_Band, and Profession columns.
7. Click Add to add the columns to the aggregate instruction.
8. Click OK. The new instruction appears in the base dimension dialog box.
9. Define additional aggregate instructions for this dimension (optional). For the sample datamart, create another aggregate instruction called **`AgeIncomeStat`** that includes the Status, Age_Band, Known_Email, and Income_Band columns.
10. Click OK to complete defining aggregate instructions.

NOTE: *The Date dimension is often used for aggregation and it comes with a large collection of pre-defined aggregate instructions.*

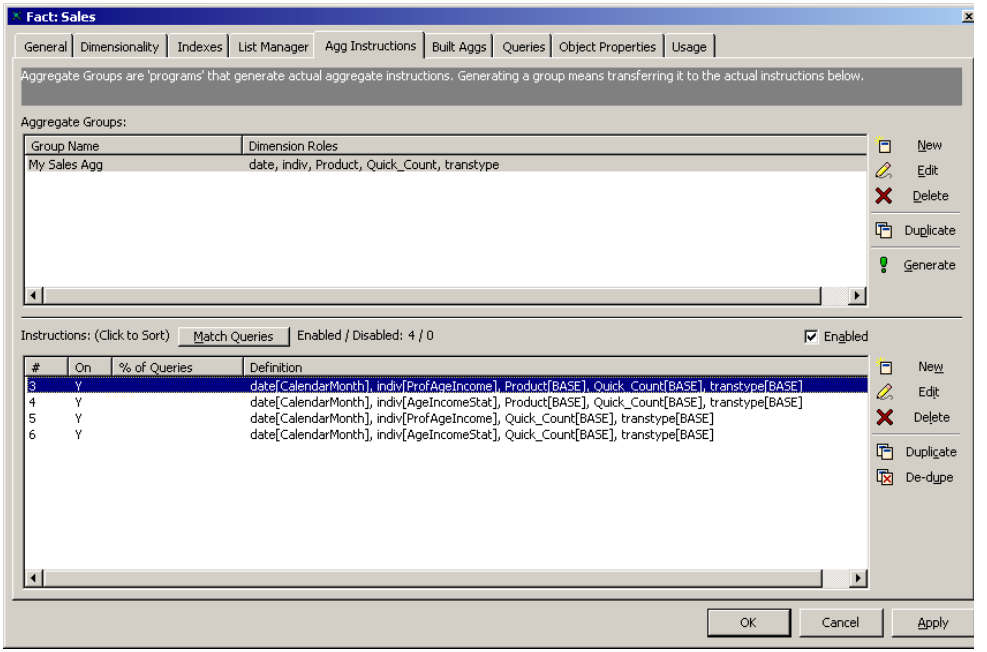
DEFINING FACT AGGREGATE GROUPS

Fact aggregate instructions are generated from fact aggregate groups.

DEFINING A FACT AGGREGATE GROUP

1. In Analytic Manager, expand the Schema folder, and select the Facts subfolder.
2. Double-click the fact for which you want to define an aggregate. For the sample datamart, select the Sales fact table. The Fact dialog box appears.

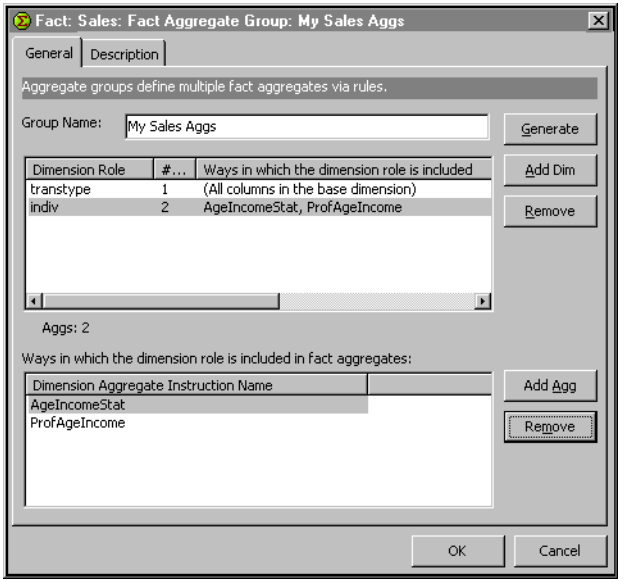
FIGURE 120: THE FACT DIALOG BOX: AGG INSTRUCTIONS TAB



3. Click the Agg Instructions tab.

4. Click New in the Aggregate Groups pane. The Fact Aggregate Group dialog box appears.

FIGURE 121: THE FACT AGGREGATE GROUP DIALOG BOX



5. In the Group Name field, enter a name. For the sample datamart, enter **`My Sales Aggs`**.
6. Click Add Dim. The Choose: Dimension Role dialog box appears.
7. Select a dimension role that points at the dimension for which you defined a column set. For the sample datamart, select the indiv dimension role, which points at the Individual_base dimension.
8. Click OK to add the dimension role. The Fact Aggregate Group dialog box shows the name of the dimension role in the top pane and the column group (All columns in the base dimension) in the bottom pane. The transtype dimension role is added automatically.
9. With the newly-added dimension role still highlighted in the top pane, click Add Agg. The Choose Dimension Aggregate Instruction dialog box appears.

10. Select the dimension aggregate instructions that you created. For the sample datamart, these instructions are called AgeIncomeStat and ProfAgeIncome.
11. Click OK to add the instructions. When an aggregate group has more than one aggregate instruction for a dimension role, a separate fact aggregate instruction is generated for each dimension aggregate instruction.
12. Select the (All columns in the base dimension) dimension-aggregate instruction, then click Remove.
13. Repeat the same process for any other dimension roles for which you want to add aggregate instructions. For the sample datamart, add the CalendarMonth column set of the Date dimension.

    To add a particular dimension in its entirety, click Add Dim, select the dimension role, and click OK. Any dimension roles that you add with only the (All columns in the base dimension) instruction are included in their entirety in every fact-aggregate instruction that is generated from this aggregate group. For the sample datamart, add the Quick_Count dimension role.

    To add a dimension that is included in some of the generated aggregates and excluded from others, use the above procedure but then add the instruction (Exclude the dimension role.

    For the sample datamart, add the Product_base dimension role with both (All columns in the base dimension) and (Exclude the dimension role) instructions.
14. Click OK to finish defining the aggregate group.

NOTE: *Adding a dimension role to an aggregate group with only the (Exclude the dimension role) instruction is equivalent to leaving that dimension role out of the aggregate group. If you add a dimension role to an aggregate group with the (Exclude the dimension role) instruction in addition to other dimension-aggregate instructions, then some of the fact-aggregate instructions that are generated from the aggregate group include the dimension role (using the other dimension-aggregate instructions) and the remaining fact-aggregate instructions exclude the dimension role.*

GENERATING FACT AGGREGATE INSTRUCTIONS

A fact aggregate group specifies potential aggregates that can be built. To generate actual aggregate instructions from an aggregate group:

1. In the Fact dialog box, click the Agg Instructions tab.
2. Select the desired aggregate group. For the sample datamart, select the My Sales Aggs group.
3. Click Generate.
4. Click OK to generate aggregate instructions.

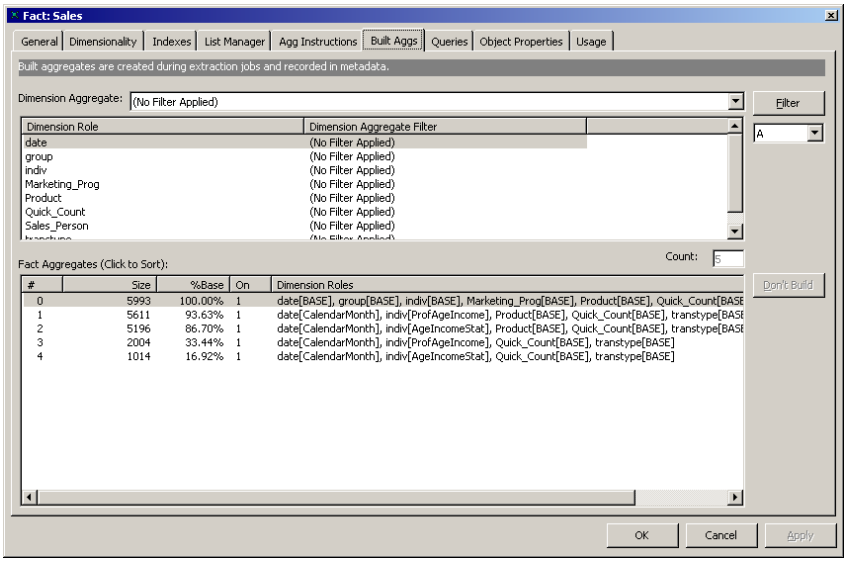
When you generate instructions, one fact aggregate instruction is generated for every combination of dimension aggregate instructions in the group. For the sample datamart, two of the dimension roles in the aggregate group have two dimension aggregate instructions and the remaining dimension roles have a single aggregate instruction, so a total of four fact aggregate instructions are generated.

RUNNING AGGBUILDER

After you define the aggregates, you run an extraction job with the AggBuilder enabled to build the aggregates. The Aggregate Builder program, which is invoked as a step in an extraction job, generates aggregates based on the dimension column sets that you select. In Chapter 4, "Extracting Data into the Datamart," you added the E.5.0.1 Builders group, which includes the AggBuild: Agg Builder step to your extraction job. You have to run the job again.

1. Run your extraction job (Extractorama in the sample datamart), as described in "Running the Extraction Job," on page 104.
2. After the job is completed, open the dialog box for the fact table for which you built aggregates (Sales in the sample datamart) and click the Built Aggs tab to view at the aggregates that were built.

FIGURE 122: THE FACT DIALOG BOX: BUILT AGGS TAB



Sample Datamart

This appendix shows the Analytic Manager screens that you use to configure specific details for the sample datamart.

THE BASE DIMENSIONS

FIGURE 123: THE HOUSEHOLD_BASE DIMENSION

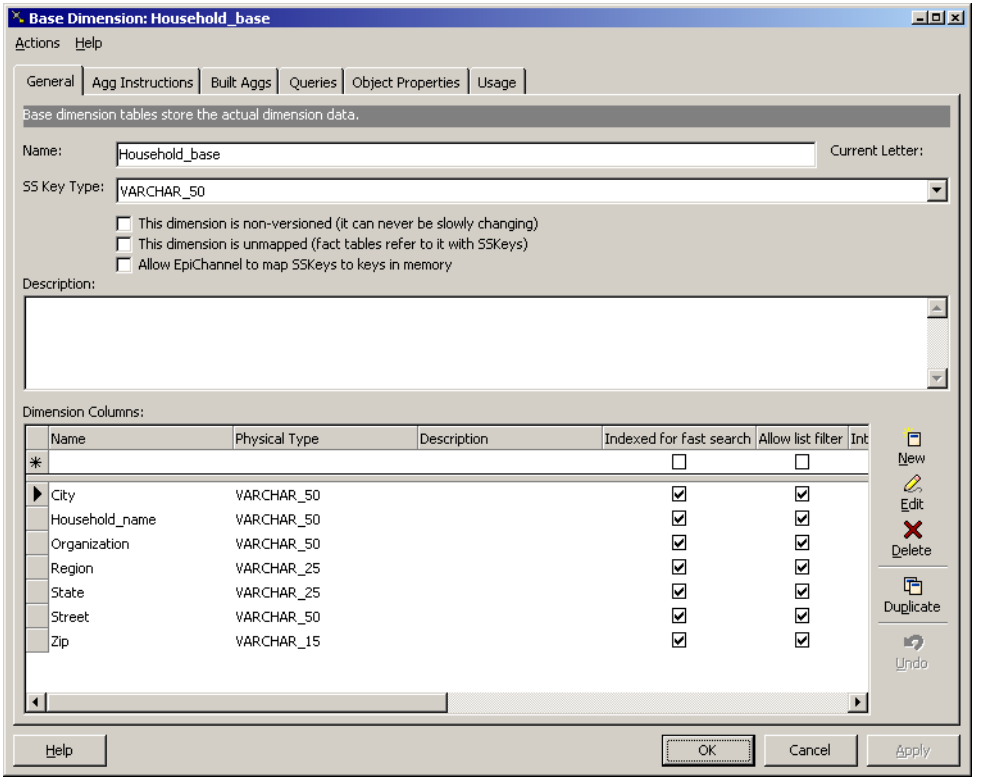


Figure 124: The Individual_base Dimension

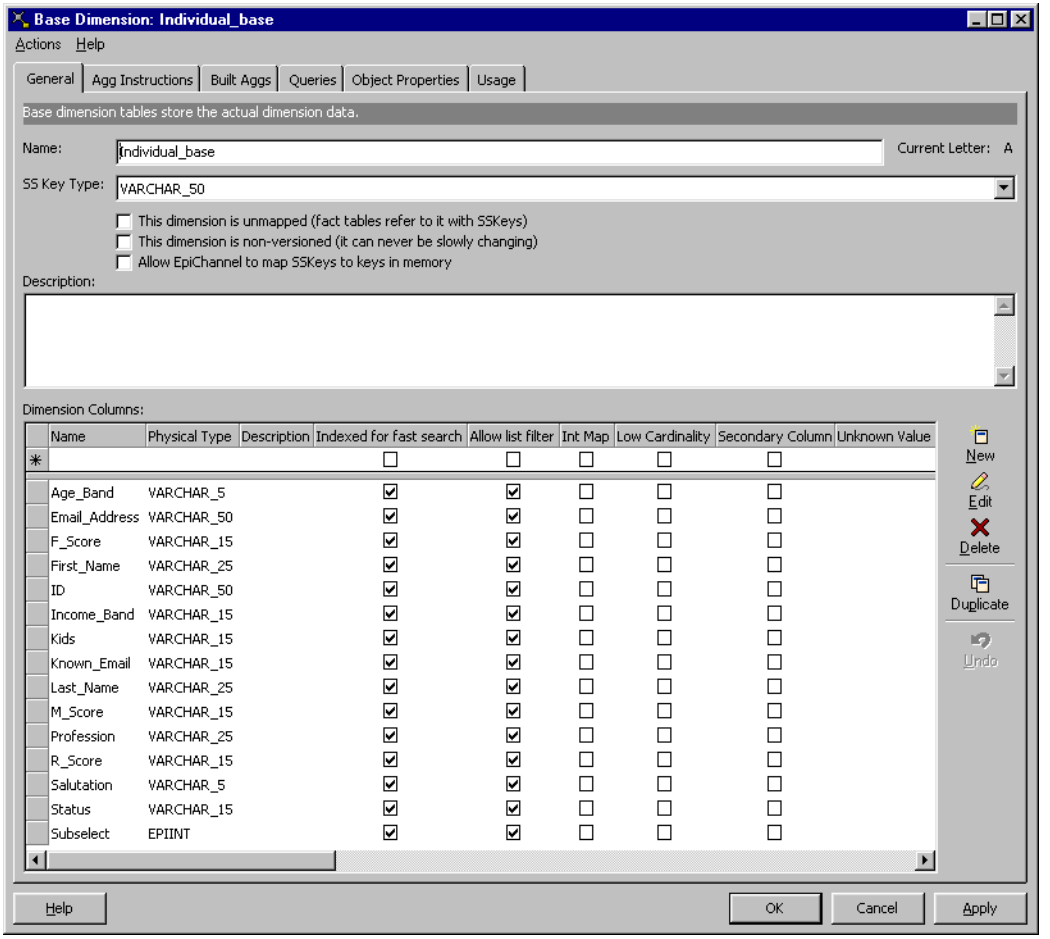


FIGURE 125: THE PRODUCT_BASE DIMENSION

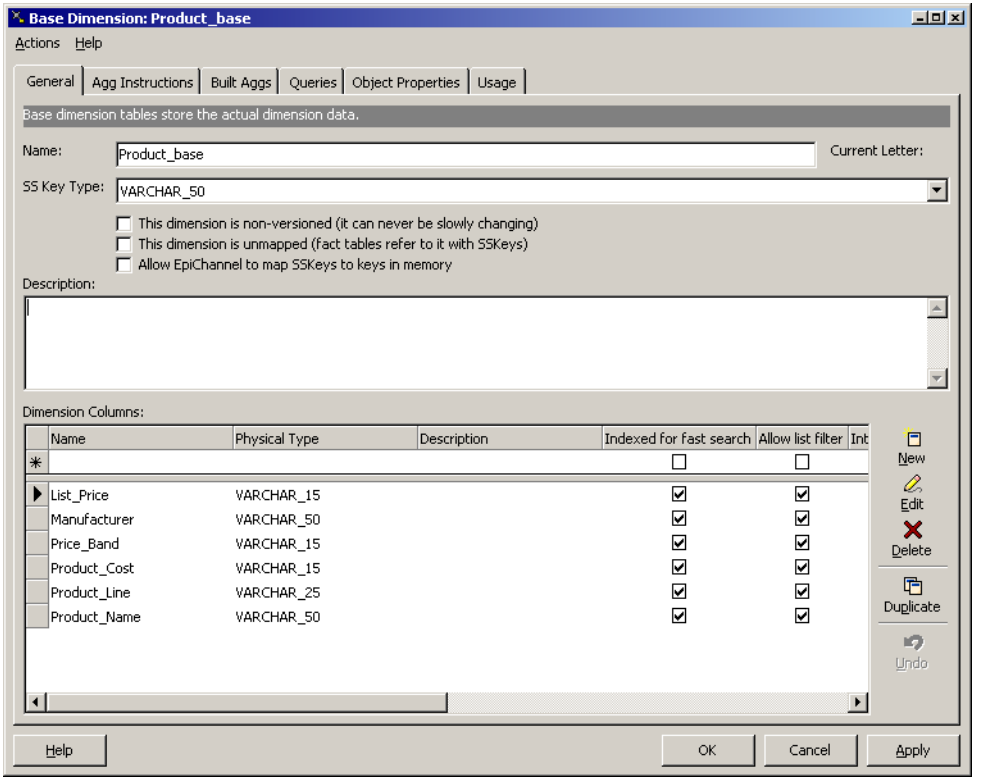


Figure 126: The Marketing_Prog_base Dimension

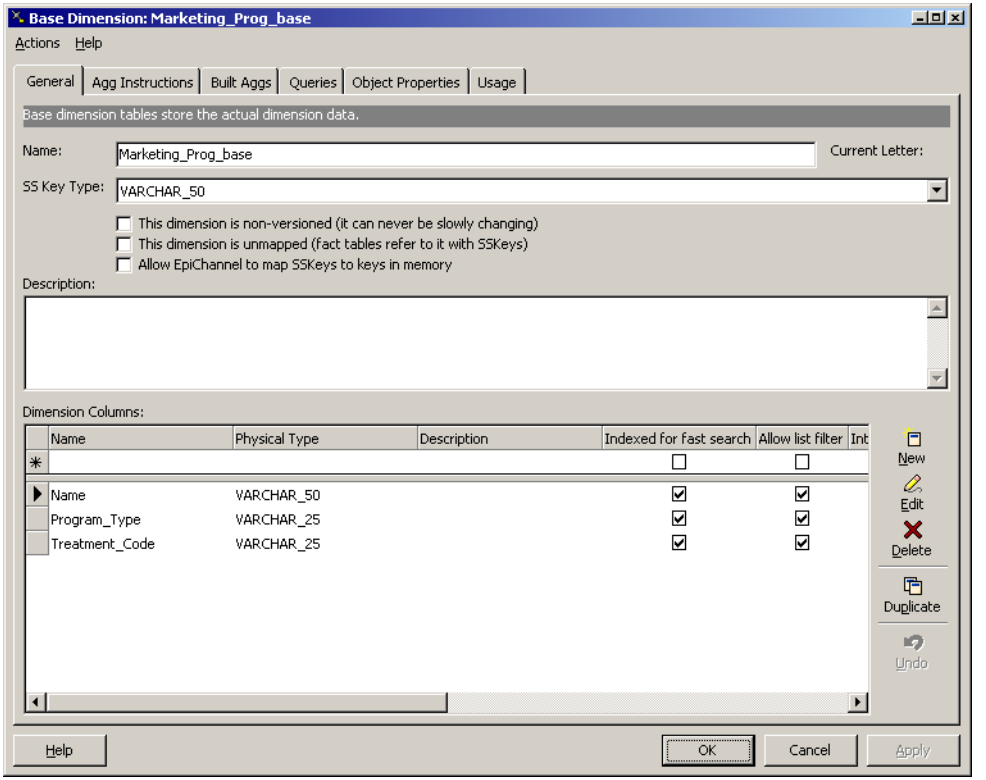


FIGURE 127: THE QUICK_COUNT_BASE DIMENSION

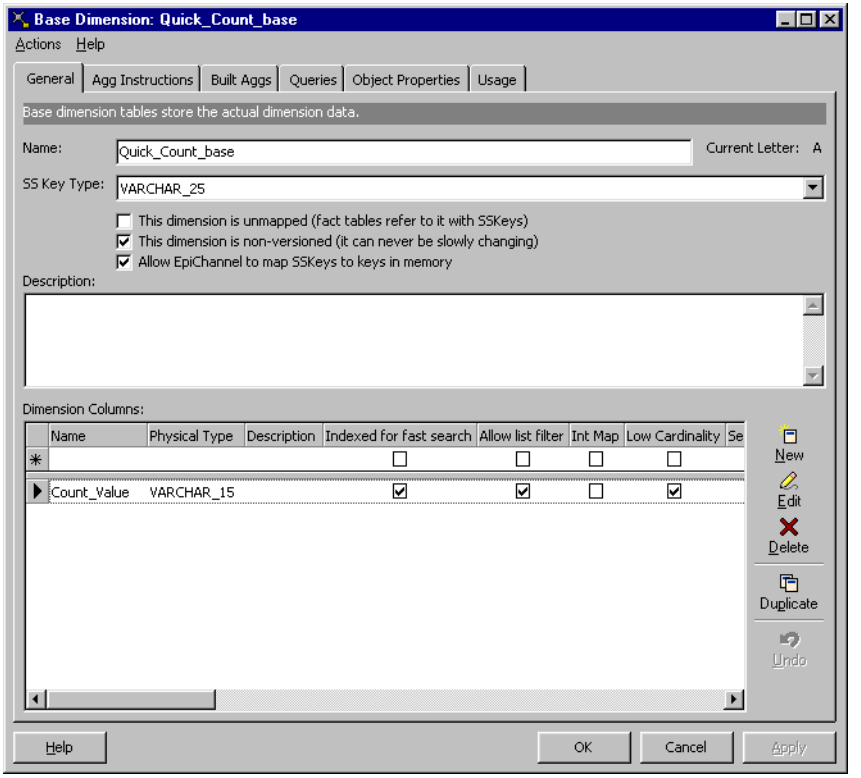
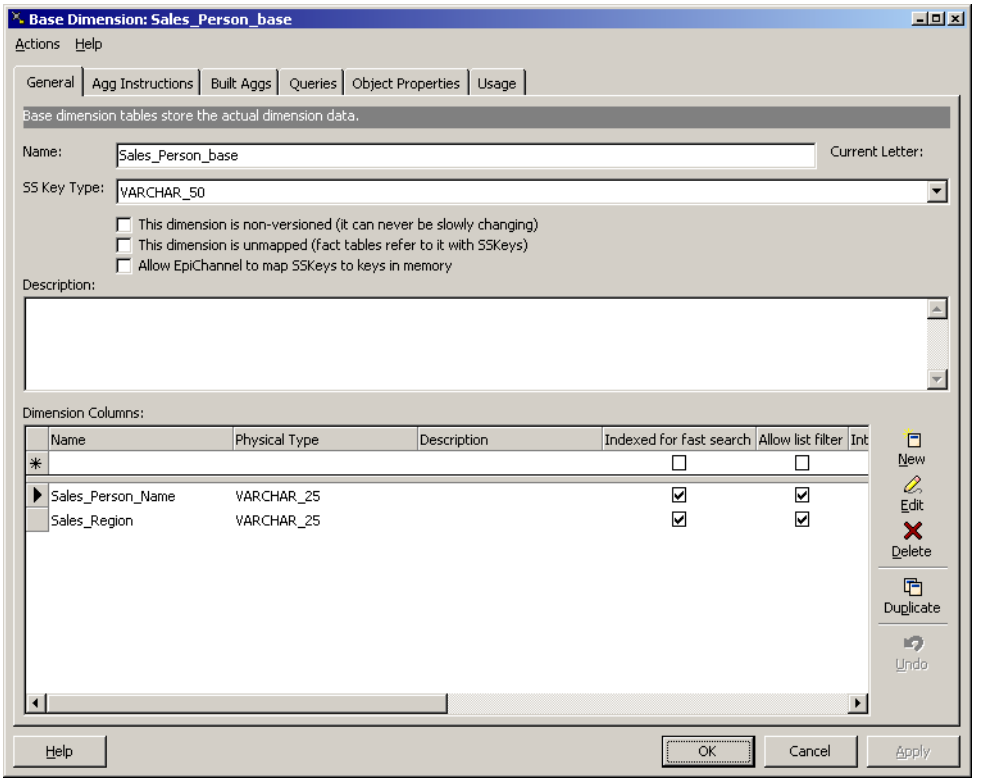


Figure 128: The Sales_Person_base Dimension



The Dimension Roles

Table 8: Dimension Roles

Dimension Role	Base Dimension
indiv	Individual_base
group	Household_base
Product	Product_base
Marketing_Program	Marketing_Prog_base
Quick_Count	Quick_Count_base
Sales_Person	Sales_Person_base

The Fact Tables

Figure 129: The Sales Fact Table: General Tab

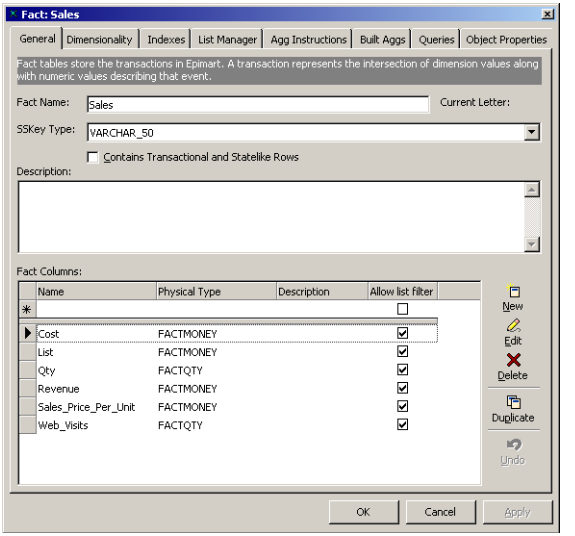


Figure 130: The Sales Fact Table: Dimensionality Tab

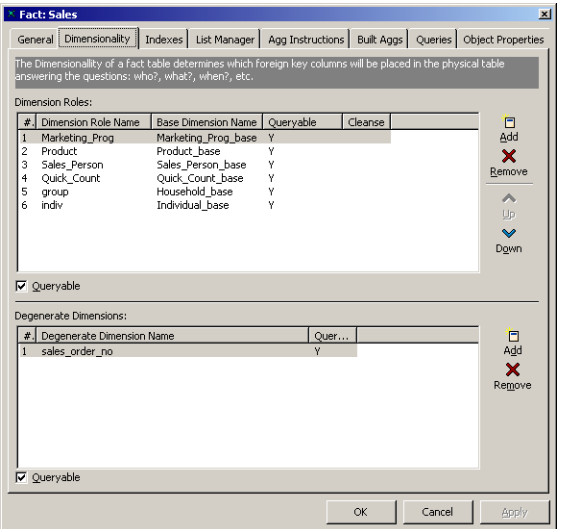


FIGURE 131: THE SALES FACT TABLE: LIST MANAGER TAB

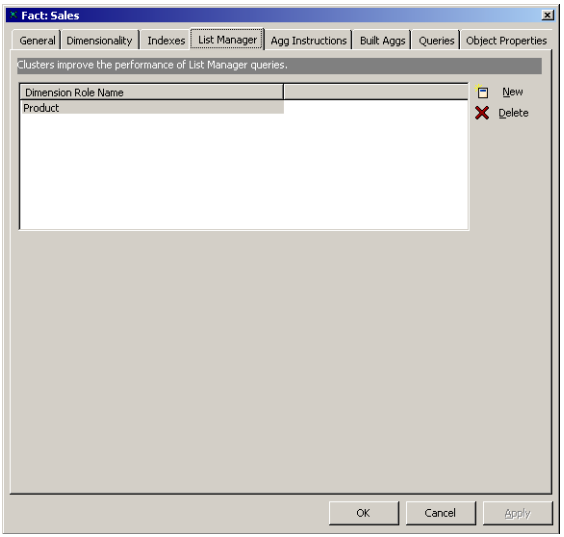


Figure 132: The Marketing Fact Table: General Tab

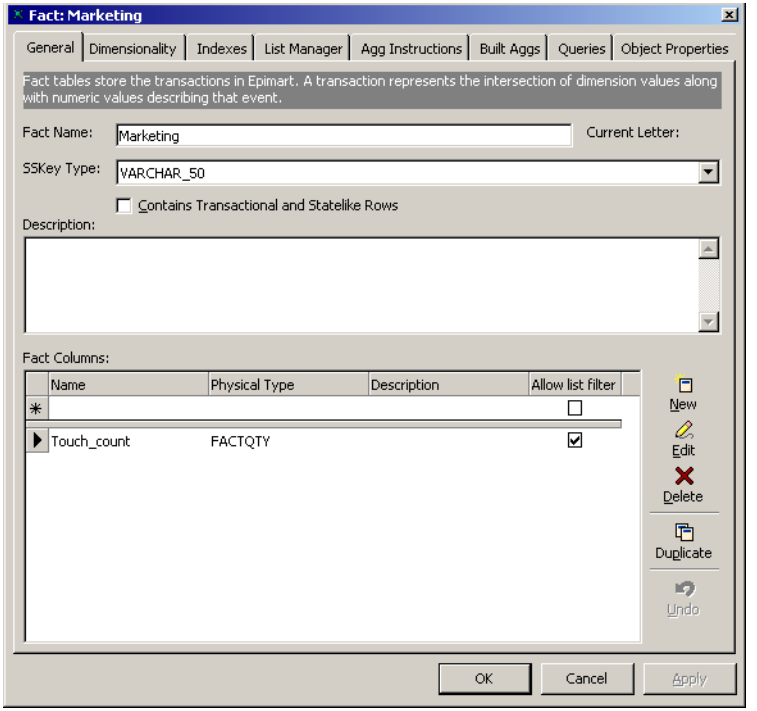


FIGURE 133: THE MARKETING FACT TABLE: DIMENSIONALITY TAB

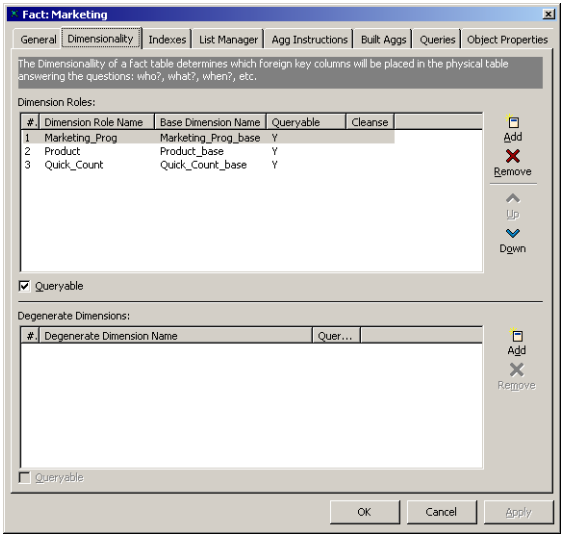
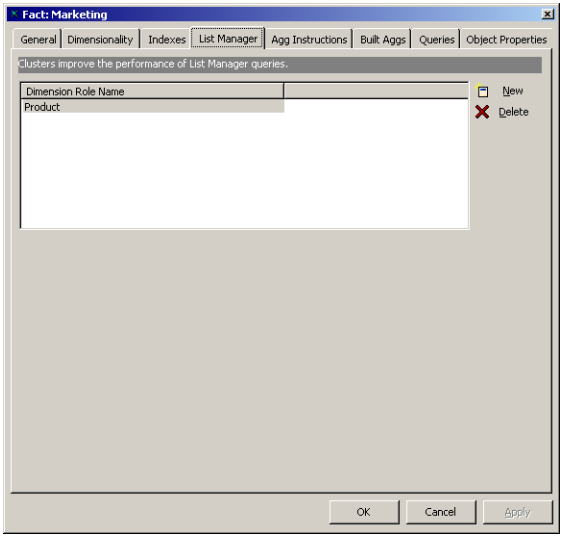


Figure 134: The Marketing Fact Table: List Manager Tab



ATTRIBUTE LAYOUTS

FIGURE 135: ATTRIBUTE LAYOUT: DATE

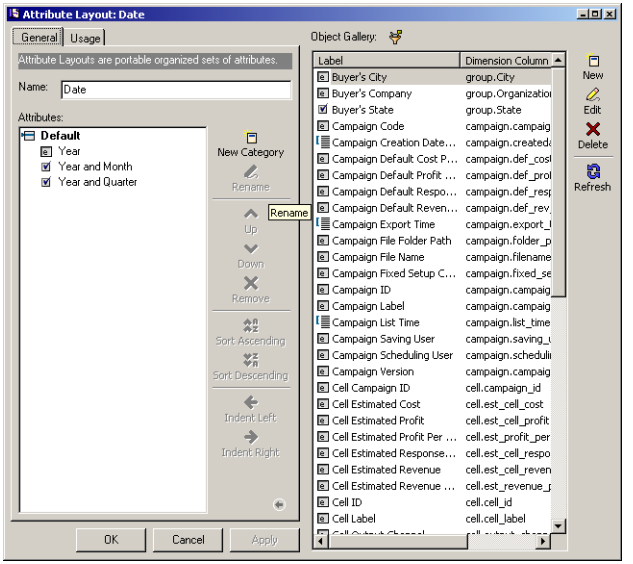


FIGURE 136: ATTRIBUTE LAYOUT: SALES

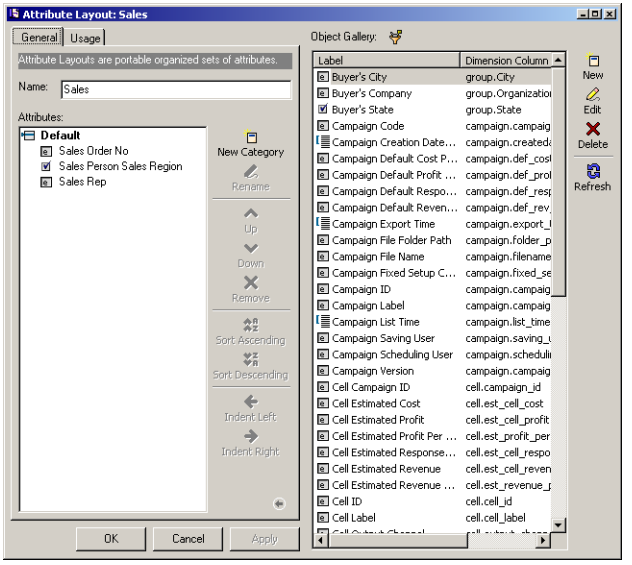


FIGURE 137: ATTRIBUTE LAYOUT: INDIV BUYER

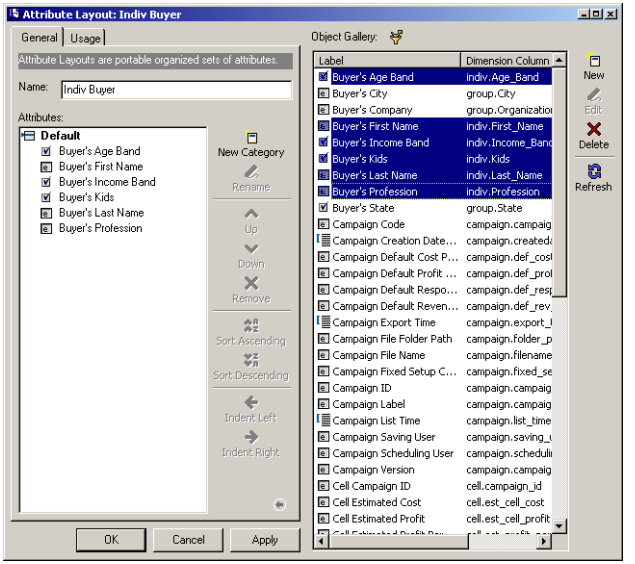
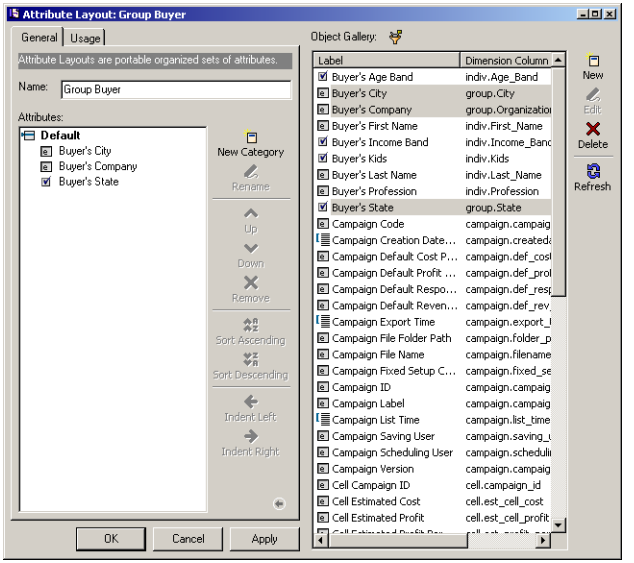


FIGURE 138: ATTRIBUTE LAYOUT: GROUP BUYER



MEASURES

FIGURE 139: THE NUMMAILINGS MEASURE: GENERAL TAB

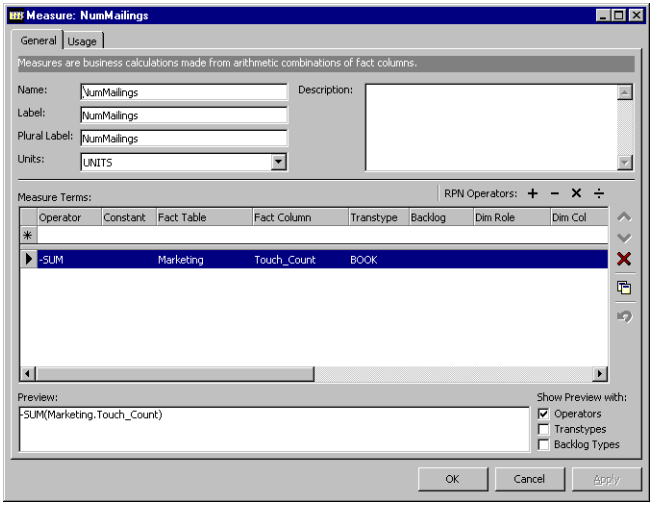


FIGURE 140: THE NUMRESPONSES MEASURE: GENERAL TAB

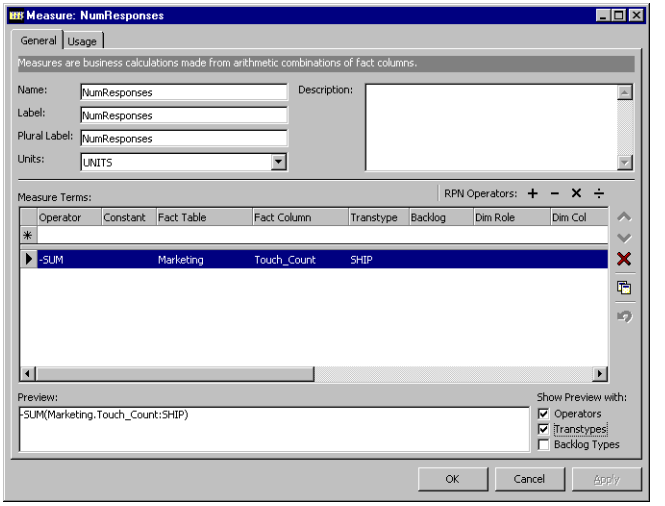


FIGURE 141: THE REVENUE BOOKED MEASURE

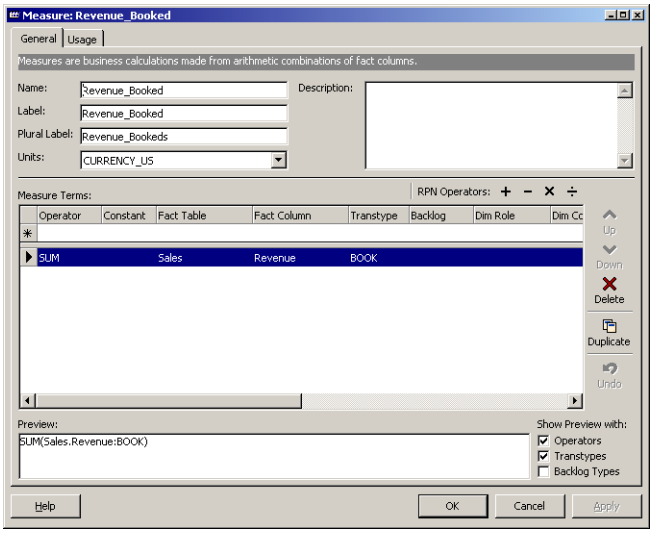


FIGURE 142: THE REVENUE SHIPPED MEASURE

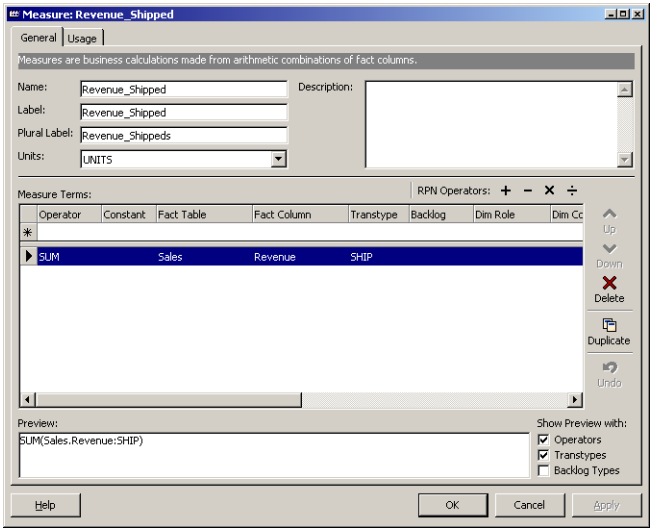


FIGURE 143: THE REVENUE BACKLOG MEASURE

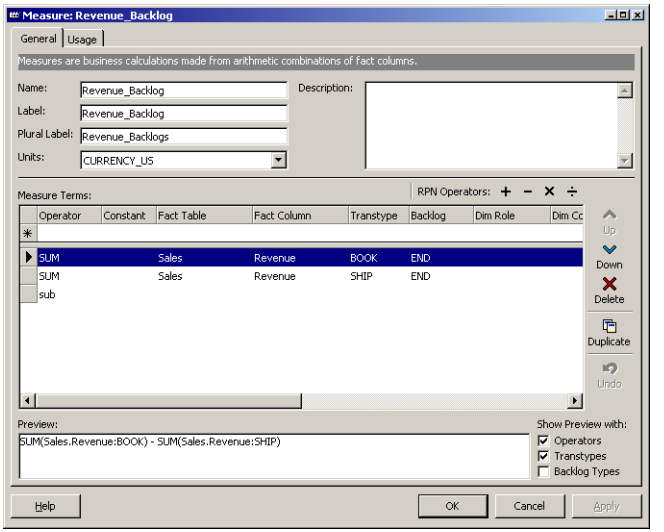


FIGURE 144: THE UNITS BOOKED MEASURE

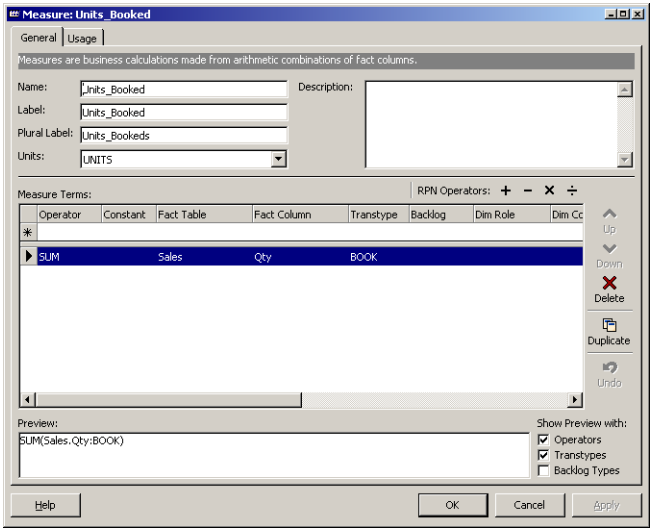


FIGURE 145: THE UNITS SHIPPED MEASURE

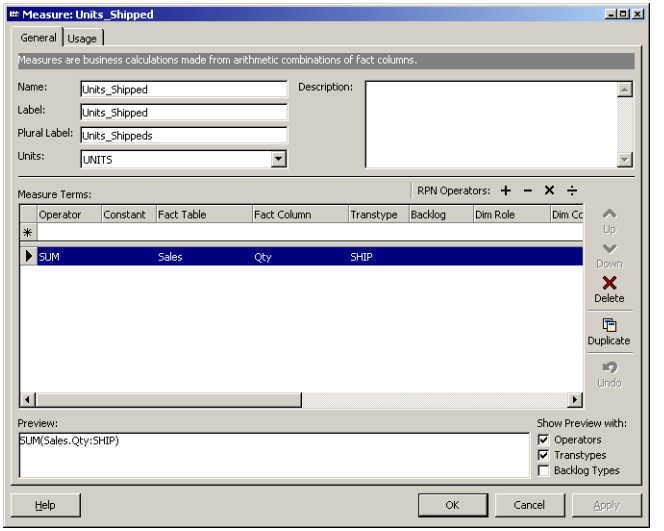


FIGURE 146: THE UNIT BACKLOG MEASURE

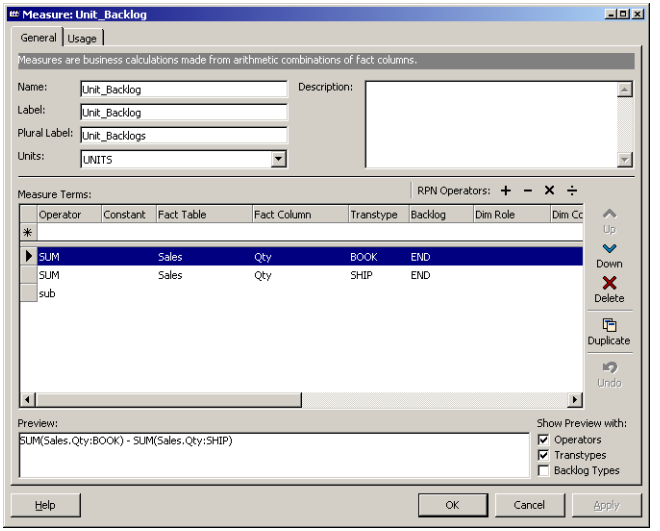


FIGURE 147: THE COUNT OF CUSTOMERS MEASURE

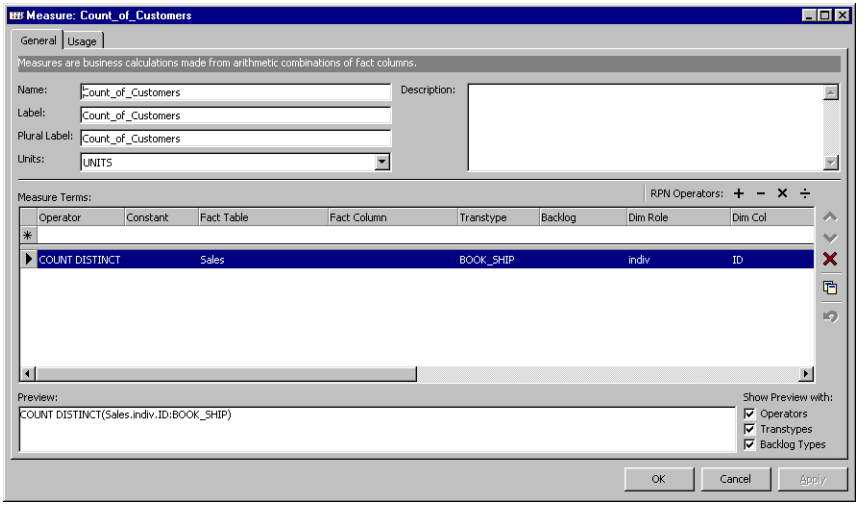
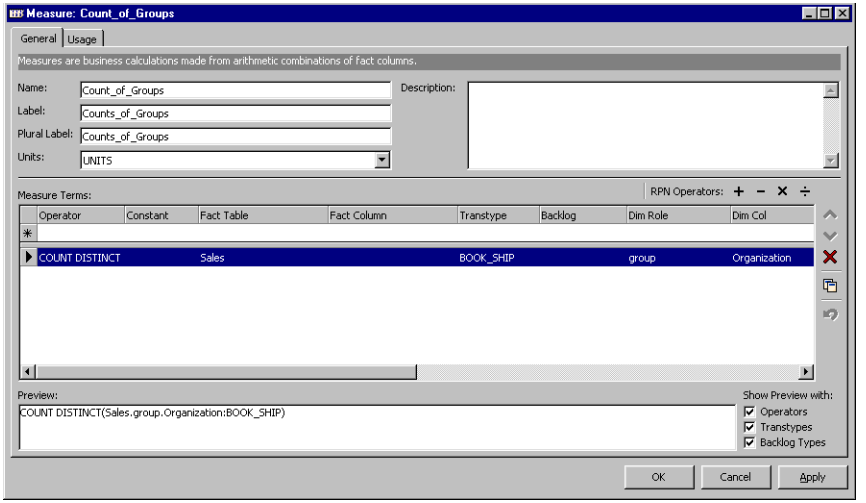


FIGURE 148: THE COUNT OF GROUPS MEASURE



ATTRIBUTES

Note: The sales order number is a degenerate dimension. When you select this dimension role, the name in the dimension column text box appears as sales_order_no.

TABLE 9: ATTRIBUTES AND FILTERS

ATTRIBUTE NAME	DIMENSION COLUMN	FILTER TYPE
Nothing	Quick_Count.Count_Value	Text Box
Year	Date.cy_name	Check Boxes
Year and Quarter	Date.cq_and_cy_name	Check Boxes
Year and Month	Date.month_and_cy_name	Check Boxes
Product Line	Product.Product_Line	Check Boxes

TABLE 9: ATTRIBUTES AND FILTERS (CONTINUED)

ATTRIBUTE NAME	DIMENSION COLUMN	FILTER TYPE
Product Name	Product.Product_Name	Check Boxes
Product Price Band	Product.Price_Band	Check Boxes
Sales Region	Sales_Person.Sales_Region	Check Boxes
Sales Rep	Sales_Person.Sales_Person_Name	List Box
Marketing Program	Marketing_Program.Name	List Box
Buyer's Income	indiv.Income_Band	Check Boxes
Buyer's Profession	indiv.Profession	List Box
Buyer's Age	indiv.Age_Band	Check Boxes
Buyer # of Children	indiv.Kids	Check Boxes
Buyer's Company	group.Organization	Dynamic List Box
Buyer's State	group.State	Check Boxes
Buyer's City	group.City	Text Box
Buyer's Last Name	indiv.Last_Name	Text Box
Buyer's First Name	indiv.First_Name	Text Box
Sales Order Number	sales_order_no	Text Box

Web Pages

Figure 149: Attributes for the Sales Reports Web Page

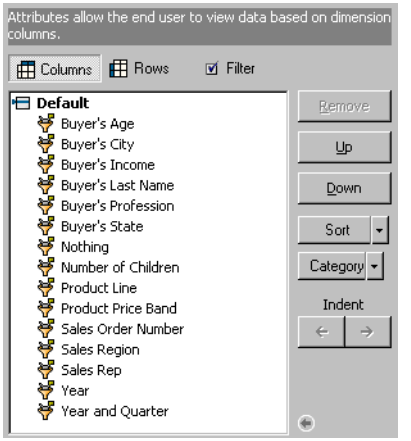
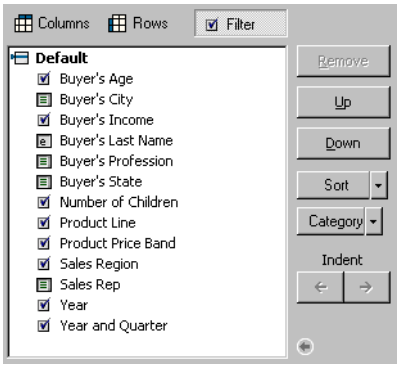


Figure 150: Filters for the Sales Reports Web Page



Tables in the Sample Database

Contact_table

```
Contact_Id
Household_Id
Salutation
Contact_first
Contact_last
Age
Income
Profession
Number_of_kids
Email
Status
Known_email
Known_MSN
R_score
F_score
M_score
```

Geographic_Region

```
Geographic_Region_Id
Region_Name
```

Household_table

```
Household_id
Company_name
Household_name
Street
City
State
Zip
Geographic_region_id
```

Manufacturer

```
Manufacturer_Id
Geographic_Region_Id
Manufacturer_Name
```

Order_items

```
Order_Item_Id
Sales_Order_Id
Product_Id
Number_of_Units
```

Product_Line

```
Product_Line_Id
Line_Name
Line_Name_PC
Line_Name_Engine
```

Product_table

```
Product_Id
Manufacturer_Id
Product_Line_Id
Product_Name
List_Price
Product_Cost
Product_Name_PC
List_Price_PC
Product_Cost_PC
Product_Name_Engine
List_Price_Engine
Product_Cost_Engine
```

program_table

```
program_number
program_name
program_name_PC
program_name_engine
product_code
Program_type
Treatment_code
segment
subsegment
cell
```

RESPONSE_TABLE

```
table_id
contact_id
date_mailed
date_responded
program_id
response_flag
```

SALES_ORDERS

```
Sales_Order_Id
Sales_Person_Id
Contact_Id
Discount
Web_visits
Booked_Date
Marketing_program_id
Shipped_date
Shipped_flag
```

SALES_PERSON

```
Sales_Person_Id
Sales_Region_Id
Sales_Person_Name
Quota
```

EXTRACTION SQL

GET BOOKINGS

```
SELECT
   SO.Sales_Order_Id $$CAT '-' $$CAT OI.Order_Item_Id ss_key,
   $$TO_EPIDATE[SO.Booked_Date]            date_key,
   1                                       transtype_key,
   1                                       process_key,
   $$NVL[ C.Household_Id ~,~ 'UNKNOWN' ]   group_sskey,
   $$NVL[ C.Contact_Id ~,~ 'UNKNOWN' ]     indiv_sskey,
   $$NVL[SO.Marketing_program_id~,~'UNKNOWN']
Marketing_Program_sskey,
   $$NVL[ OI.Product_Id ~,~ 'UNKNOWN' ]    Product_sskey,
   '1'                                     Quick_Count_sskey,
   $$NVL[ SO.Sales_Order_Id ~,~ 'UNKNOWN' ]sales_order_no_key,
   $$NVL[ SO.Sales_Person_Id ~,~ 'UNKNOWN' ]Sales_Person_sskey,
   $$NVL[ P.Product_Cost_PC ~,~ 0 ]        Cost,
   $$NVL[ P.List_Price_PC ~,~ 0 ]          List,
   $$NVL[ OI.Number_of_Units ~,~ 0 ]       Qty,
   $$NVL[P.List_Price_PC*(1-SO.Discount)*OI.Number_of_Units~,~0]
Revenue,
   $$NVL[ P.List_Price_PC * (1-SO.Discount) ~,~ 0 ]
Sales_Price_Per_Unit,
   $$NVL[ SO.web_visits ~,~ 0 ]            Web_Visits
FROM
   sales_orders     SO,
   Order_items      OI,
   contact_table    C,
   Product_table    P
WHERE
   SO.Sales_Order_Id = OI.Sales_Order_Id  AND
   SO.Contact_Id = C.Contact_Id  AND
   OI.Product_Id = P.Product_Id
```

NOTE: *The unique source-system key is generated by concatenating two ID numbers. The IDs are already in string form, and the* $$CAT *macro is used to concatenate the strings.*

GET EXTRACT DATE

```
SELECT
   $$TO_EPIDATE[$$DBNOW]      max_date,
   'UNKNOWN'                  tbl_name
$$NO_FROM_LIST
```

GET HOUSEHOLD

```
SELECT DISTINCT
   H.Household_id                            Household_base_sskey,
   $$TO_EPIDATE[$$DBNOW]                     date_modified,
   $$NVL[ H.Household_name ~,~ 'UNKNOWN' ] Household_Name,
   $$NVL[ H.Company_name ~,~ 'UNKNOWN' ]   Organization,
   $$NVL[ H.Street ~,~ 'UNKNOWN' ]         Street,
   $$NVL[ H.City ~,~ 'UNKNOWN' ]           City,
   $$NVL[ H.Zip ~,~ 'UNKNOWN' ]            Zip,
   $$NVL[ H.State ~,~ 'UNKNOWN' ]          State,
   $$NVL[ R.Region_Name ~,~ 'UNKNOWN' ]    Region
FROM
   household_table H,
   Geographic_Region R
WHERE
   H.Geographic_Region_id = R.Geographic_Region_Id
```

GET IND_GROUP_JOINER

```
SELECT
   C.Contact_Id                           ss_key,
   $$TO_EPIDATE['Dec 5 1997  2:03PM']   date_key,
   1                                      transtype_key,
   2                                      process_key,
   $$NVL[ H.Household_id ~,~ 'UNKNOWN' ] group_sskey,
   $$NVL[ C.Contact_Id ~,~ 'UNKNOWN' ]  indiv_sskey,
   1                                      member
FROM
   contact_table C,
   household_table H
WHERE
   H.Household_id = C.Household_Id
```

GET INDIVIDUAL

```
SELECT DISTINCT
   Contact_Id                                  Individual_base_sskey,
   $$TO_EPIDATE[$$DBNOW]                       date_modified,
   $$CASE_BEGIN[SIGN(21-Age)~,~1~,~' ..20']
      $$CASE_ELSEIF[SIGN(41-Age)~,~1~,~'21-40']
      $$CASE_ELSEIF[SIGN(61-Age)~,~1~,~'41-60']
      $$CASE_ELSEIF[SIGN(Age)~,~1~,~'61.. ']
      $$CASE_ELSE['UNKNOWN']
   $$CASE_END                                  Age_Band,
   $$NVL[ Contact_Id ~,~ 'UNKNOWN' ]           ID,
   $$NVL[ Email ~,~ 'UNKNOWN' ]                Email_Address,
   $$NVL[ F_Score ~,~ 'UNKNOWN' ]              F_Score,
   $$NVL[ Contact_first ~,~ 'UNKNOWN' ]        First_Name,
   $$CASE_BEGIN[SIGN(30001-Income)~,~1~,~'<$30k']
      $$CASE_ELSEIF[SIGN(60001-Income)~,~1~,~'$30k-$60k']
      $$CASE_ELSEIF[SIGN(90001-Income)~,~1~,~'$60k-$90k']
      $$CASE_ELSEIF[SIGN(Income)~,~1~,~'>$90k']
      $$CASE_ELSE['UNKNOWN']
   $$CASE_END                                  Income_Band,
   $$NVL[ $$TO_CHAR[Number_of_kids] ~,~ 'UNKNOWN' ] Kids,
   $$NVL[ Known_email ~,~ 'UNKNOWN' ]          Known_Email,
   $$NVL[ Contact_last ~,~ 'UNKNOWN' ]         Last_Name,
   $$NVL[ M_Score ~,~ 'UNKNOWN' ]              M_Score,
   $$NVL[ Profession ~,~ 'UNKNOWN' ]           Profession,
   $$NVL[ R_Score ~,~ 'UNKNOWN' ]              R_Score,
   $$NVL[ Salutation ~,~ 'UNKNOWN' ]           Salutation,
   $$NVL[ Status ~,~ 'UNKNOWN' ]               Status,
   $$NVL[ $$TO_INT[ Contact_Id ] % 20 ~,~ 0 ]Subselect
FROM
      contact_table
```

NOTE: *Since there is no platform-independent version of the* **CASE** *statement with inequalities, the same effect is achieved here using the* **$$CASE** *macros in combination with the* **SIGN** *function. You can also use binning macros for this purpose. See the Datamart Guide for a detailed description of macros.*

Since **Number_of_kids** *is in integer form in the source database, the* **$$TO_CHAR** *macro is used to convert it to a string.*

GET MAILINGS

```
SELECT
    R.table_id                                ss_key,
    $$TO_EPIDATE[R.date_mailed]               date_key,
    1                                         transtype_key,
    1                                         process_key,
    $$NVL[ C.Household_Id ~,~ 'UNKNOWN' ]group_sskey,
    $$NVL[ R.contact_id ~,~ 'UNKNOWN' ]  indiv_sskey,
    $$NVL[ R.program_id ~,~ 'UNKNOWN' ]  Marketing_Program_sskey,
    $$NVL[ P.product_code ~,~ 'UNKNOWN' ]Product_sskey,
    '1'                                       Quick_Count_sskey,
    1                                         Touch_Count
FROM
    response_table R,
    contact_table C,
    program_table P
WHERE
    R.contact_id = C.Contact_Id  AND
    R.program_id = P.program_number
```

NOTE: *Since* program_table *has null values in the* product_code *column, the* $$NVL *macro is used to assign the result values. This replaces all null values with* 'UNKNOWN'.

GET MARKETING PROGRAM

```
SELECT DISTINCT
    program_number                            Marketing_Prog_base_sskey,
    $$TO_EPIDATE[$$DBNOW]                     date_modified,
    $$NVL[ program_name_PC ~,~ 'UNKNOWN' ]Name,
    $$NVL[ Treatment_code ~,~ 'UNKNOWN' ]Treatment_Code,
    $$NVL[ Program_type ~,~ 'UNKNOWN' ]  Program_Type
FROM
    program_table
WHERE
    program_number is not null
```

GET PRODUCT

```
SELECT DISTINCT
   P.Product_Id                              Product_base_sskey,
   $$TO_EPIDATE[$$DBNOW]                     date_modified,
   $$NVL[ P.Product_Name_PC ~,~ 'UNKNOWN' ]Product_Name,
   $$NVL[ P.List_Price_PC ~,~ 'UNKNOWN' ]  List_Price,
   $$NVL[ PL.Line_Name_PC ~,~ 'UNKNOWN' ]  Product_Line,
   $$NVL[ P.Product_Cost_PC ~,~ 'UNKNOWN' ]Product_Cost,
   $$NVL[ M.Manufacturer_Name ~,~ 'UNKNOWN' ]Manufacturer,
   $$CASE_BEGIN[SIGN(501-P.List_Price_PC)~,~1~,~'<$500']
      $$CASE_ELSEIF[SIGN(1001-P.List_Price_PC)~,~1~,~'$500-
$1,000']
      $$CASE_ELSEIF[SIGN(3001-P.List_Price_PC)~,~1~,~'$1,000-
$3,000']
      $$CASE_ELSEIF[SIGN(P.List_Price_PC)~,~1~,~'>$3,000']
      $$CASE_ELSE['UNKNOWN']
   $$CASE_END                                Price_Band
FROM
   Product_table P,
   Product_Line PL,
   Manufacturer M
WHERE
   P.Product_Line_Id = PL.Product_Line_Id AND
   P.Manufacturer_Id = M.Manufacturer_Id
```

GET RESPONSES

```
SELECT
   R.table_id                              ss_key,
   $$TO_EPIDATE[R.date_responded]          date_key,
   101                                     transtype_key,
   1                                       process_key,
   $$NVL[ C.Household_Id ~,~ 'UNKNOWN' ]group_sskey,
   $$NVL[ R.contact_id ~,~ 'UNKNOWN' ]  indiv_sskey,
   $$NVL[ R.program_id ~,~ 'UNKNOWN' ]  Marketing_Program_sskey,
   $$NVL[ P.product_code ~,~ 'UNKNOWN' ]Product_sskey,
   '1'                                     Quick_Count_sskey,
   1                                       Touch_Count
FROM
   response_table R,
   contact_table C,
   program_table P
WHERE
   R.contact_id = C.Contact_Id  AND
   R.program_id = P.program_number AND
   R.response_flag LIKE '1'
```

GET SALES PERSON

```
SELECT DISTINCT
   S.Sales_Person_Id                       Sales_Person_base_sskey,
   $$TO_EPIDATE[$$DBNOW]                   date_modified,
   $$NVL[ G.Region_Name ~,~ 'UNKNOWN' ] Sales_Region,
   $$NVL[ S.Sales_Person_Name ~,~ 'UNKNOWN' ]Sales_Person_Name
FROM
   Sales_Person S,
   Geographic_Region G
WHERE
   S.Sales_Region_Id = G.Geographic_Region_Id
```

GET SHIPPINGS

```
SELECT
   SO.Sales_Order_Id $$CAT '-' $$CAT OI.Order_Item_Id  ss_key,
   $$TO_EPIDATE[SO.Shipped_date]              date_key,
   101                                        transtype_key,
   1                                          process_key,
   $$NVL[ C.Household_Id ~,~ 'UNKNOWN' ]      group_sskey,
   $$NVL[ C.Contact_Id ~,~ 'UNKNOWN' ]        indiv_sskey,
   $$NVL[ SO.Marketing_program_id ~,~ 'UNKNOWN' ]
Marketing_Program_sskey,
   $$NVL[ OI.Product_Id ~,~ 'UNKNOWN' ]       Product_sskey,
   '1'                                        Quick_Count_sskey,
   $$NVL[ SO.Sales_Order_Id ~,~ 'UNKNOWN' ]   sales_order_no_key,
   $$NVL[ SO.Sales_Person_Id ~,~ 'UNKNOWN' ]  Sales_Person_sskey,
   $$NVL[ P.Product_Cost_PC ~,~ 0 ]           Cost,
   $$NVL[ P.List_Price_PC ~,~ 0 ]             List,
   $$NVL[ OI.Number_of_Units ~,~ 0 ]          Qty,
   $$NVL[P.List_Price_PC*(1-SO.Discount)*OI.Number_of_Units~,~0]
Revenue,
   $$NVL[ P.List_Price_PC * (1-SO.Discount) ~,~ 0 ]
Sales_Price_Per_Unit,
   $$NVL[ SO.Discount ~,~ 0 ]                 Discount,
   $$NVL[ web_visits ~,~ 0 ]                  Web_Visits
FROM
   sales_orders SO,
   Order_items OI,
   Contact_table C,
   Product_table P
WHERE
   SO.Shipped_flag like '1' AND
   SO.Contact_Id = C.Contact_Id AND
   SO.Sales_Order_Id = OI.Sales_Order_Id AND
```

GET QUICK COUNT

```
SELECT
   '1'                        Quick_Count_base_sskey,
   $$TO_EPIDATE[$$DBNOW]      date_modified,
   'Results'                  Count_Value
$$NO_FROM_LIST
```

INDEX

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W