

## Informatica Power Center

Lesson 00: Course Overview

iGATE is now a part of Capgemini

People matter, results count.



©2016 Capgemini. All rights reserved.

The information contained in this document is proprietary and confidential. For Capgemini only.

## Course Audience

- This course is designed for:
  - Analysts, Designers and Developers involved in Design, Development, and Maintenance of Data Warehousing Applications



## Document History

Date	Course Version No.	Software Version No.	Developer / SME	Reviewer(s)	Approver	Change Record Remarks
15-Apr-2009	1.0	Informatica 8.5.1	Vandana Mistry			Content creation
24-Apr-2009	1.0	Informatica 8.5.1				Template application
1-Aug-2013	1.1	Informatica 9.5.1	Vandana Mistry			Align course as per new version
6-Jun-2015	1.2	Informatica 9.6	Vandana Mistry			Additional Topics
4-Jul-2016	1.3	Informatica 9.6	Swati Rao	Rajita Dhumal	Mahima Sharma	Material Revamp as per Integrated ToC for I & D LoT



Copyright © Capgemini 2015. All Rights Reserved. 3

## Course Goals and Non Goals

- Course Goals
  - To understand Informatica Power center as an ETL tool
- Course Non Goals
  - Power center Administration



## Pre-requisites

- Good knowledge of RDBMS
- Fair knowledge of programming
- Fair knowledge of Data Warehousing Concepts



Copyright © Capgemini 2015. All Rights Reserved.

## Intended Audience

- Professional Data warehousing experts, or anyone requiring a foundation knowledge in ETL process



## Day Wise Schedule

- Day 1
  - Lesson 1: Informatica PowerCenter
  - Lesson 2: PowerCenter Designer
- Day 2
  - Lesson 2: PowerCenter Designer
  - Lesson 3: Workflow Manager
  - Lesson 4: Transformations
- Day 3
  - Lesson 4: Transformations
  - Lesson 5: Mapplets
- Day 4
  - Lesson 6: Workflow and Session Log
  - Lesson 7: Additional Transformations



Copyright © Capgemini 2015. All Rights Reserved.

## Day Wise Schedule

- Day 5

- Lesson 7: Additional Transformations
- Lesson 8: Workflow Tasks

- Day 6

- Lesson 9 : Pre-SQL, Post-SQL, SQL Override, Update Override
- Lesson 10: Indirect Source Files
- Lesson 11: Mapping Parameter

- Day 7

- Lesson 11: Mapping variable
- Lesson 12: Using FTP Connection

- Day 8

- Lesson 13: Dynamic Lookup
- Lesson 14: Creation of Type 2 Dimension/Version Data Mapping
- Lesson 15: Metadata Tables and Views

Copyright © Capgemini 2015. All Rights Reserved.

## Day Wise Schedule

### ■ Day 9

- Lesson 16: Partitioning
- Lesson 17: Session Parameters
- Lesson 18 : Incremental Aggregation
- Lesson 19: Constraint Based Loading

### ■ Day 10

- Lesson 20: Row Error Logging
- Lesson 21: Error Handling Architecture
- Lesson 22: Metadata Tables and Views
- Lesson 23: Standards And Tips

## Table of Contents

- Lesson 1: Informatica PowerCenter
  - 1.1. What is a Data Warehouse ?
  - 1.2. Overview of Data Warehouse Model
  - 1.3. ETL Process
  - 1.4. Informatica PowerCenter Components
  - 1.5. PowerCenter Domain
  - 1.6. Integration Service
  - 1.7. Repository Manager
  - 1.8. User and Group Management
  - 1.9. Folder
- Lesson 2: PowerCenter Designer
  - 2.1. Concept of a Mapping
  - 2.2. PowerCenter Designer : Modes



Copyright © Capgemini 2015. All Rights Reserved. 10

## Table of Contents

- 2.3. Analyze Sources
- 2.4. Creating a Target Schema
- 2.5. Mapping Designer
- 2.6. Transformation Overview
- 2.7. Source Qualifier Transformation
- 2.8. Expression Transformation
- 2.9. Aggregator Transformation
- Lesson 3: Workflow Manager
  - 3.1. Workflow
  - 3.2. Workflow Monitor
  - 3.3. Database Connections
  - 3.4. Tasks
  - 3.5. Session Task
  - 3.6. Debugger



Copyright © Capgemini 2015. All Rights Reserved. 11

## Table of Contents

- Lesson 4: Transformations
  - 4.1. Lookup Transformation
  - 4.2. Filter Transformation
  - 4.3. Joiner Transformation
  - 4.4. Rank Transformation
  - 4.5. Working with Flat Files
- Lesson 5: Mapplets
  - 5.1. Concept of Mapplet
  - 5.2. Mapplet Input and Output
  - 5.3. Normalizer Transformation
- Lesson 6: Workflow and Session Log
  - 6.1. Workflow Schedule
  - 6.2. Scheduler Options



Copyright © Capgemini 2015. All Rights Reserved. 12

## Table of Contents

- 6.3. Editing a Workflow
- 6.4. Deleting a Workflow
- 6.5. Session Logs
- 6.6. Setting Tracing Levels
- 6.7. Update Strategy Transformation
  
- **Lesson 7: Additional Transformation**
  - 7.1. Router Transformation
  - 7.2. Sorter Transformation
  - 7.3. Stored Procedure Transformation
  - 7.4. Sequence Generator Transformation
  - 7.5. Union Transformation
  - 7.6. Reusable Transformations



Copyright © Capgemini 2015. All Rights Reserved. 13

## Table of Contents

- Lesson 8: Workflow Tasks
  - 8.1. Command Task
  - 8.2. Email Task
  - 8.3. Timer Task
  - 8.4. Decision Task
  - 8.5. Event Tasks
  - 8.6. Worklets
  - 8.7. Overview of Version Control
- Lesson 9: Pre-SQL, Post-SQL, SQL Override and Update Override
  - 9.1. Pre-Post SQL
  - 9.2. SQL Override
  - 9.3. Update Override



Copyright © Capgemini 2015. All Rights Reserved. 14

## Table of Contents

- Lesson 10: Indirect Source Files
  - 10.1. Using Multiple Source Files
  - 10.2. Indirect Files
- Lesson 11: Mapping Parameter and Mapping Variable
  - 11.1 Mapping Parameter and Variable Overview
  - 11.2. Mapping Parameters
  - 11.3. Mapping Variables
- Lesson 12: FTP connection
  - 12.1. Using an FTP connection



Copyright © Capgemini 2015. All Rights Reserved. 15

## Table of Contents

- Lesson 13: Dynamic Lookup
  - 13.1. Static Cache
  - 13.2. Dynamic Cache
- Lesson 14: Creating a Type 2 Dimension/Version Data Mapping
  - 14.1. Introduction
  - 14.2. Handling Keys
  - 14.3. Example
  - 14.4. Numbering Versions
  - 14.5. Understanding the Mapping
- Lesson 15: Commit Point
  - 15.1. Source based Commit Interval
  - 15.2. Target based Commit Interval
  - 15.3. User Defined Commit Interval

## Table of Contents

- Lesson 16: Partitioning
  - 16.1. Introduction to partition
  - 16.2. Types of partitions
- Lesson 17: Session Parameters
  - 17.1. Types Of Session Parameter
  - 17.2. User Of Parameter file
  - 17.3 Parameter file format
- Lesson 18: Incremental Aggregation
  - 18.1. Types Of Session Parameter
  - 18.2. User Of Parameter file
  - 18.3 Parameter file format



Copyright © Capgemini 2015. All Rights Reserved. 17

## Table of Contents

- Lesson 19 : Partitioning
  - 19.1. Introduction to partition
  - 19.2. Types of partitions
- Lesson 20: Row Error Logging
  - 20.1. Introduction to Row Error Logging
  - 20.2. Steps For Implementation
- Lesson 21: Error Handling Techniques
  - 21.1. Types Of Session Parameter
  - 21.2. User Of Parameter file
  - 21.3. Parameter file format



Copyright © Capgemini 2015. All Rights Reserved. 18

## Table of Contents

- Lesson 22: Metadata Tables and Views
  - 22.1. Types Of Session Parameter
  - 22.2. User Of Parameter file
  - 22.3. Parameter file format
  
- Lesson 23: Standards and Tips
  - 23.1. Naming Standards
  - 23.2. Tuning Tips



Copyright © Capgemini 2015. All Rights Reserved. 19

# Informatica PowerCenter

Lesson 1- Informatica  
PowerCenter

## Lesson Objectives

- In this lesson you will learn about:
  - What is a Data Warehouse ?
  - Overview of Data Warehouse Model
  - ETL Process
  - Informatica PowerCenter Components
  - PowerCenter Domain
  - Integration Service
  - Repository Manager
  - User and Group Management
  - Folder



Copyright © Capgemini 2014. All Rights Reserved.

1.1: Data Warehouse

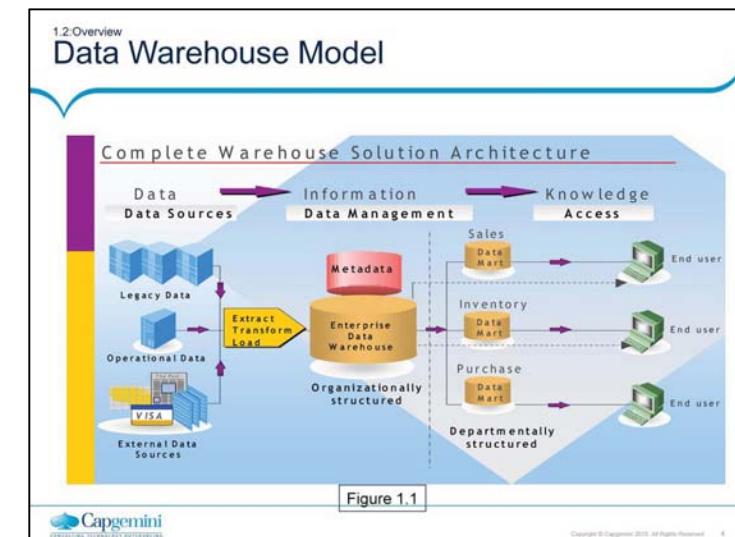
## What Is A Data Warehouse ?

- A Data Warehouse is an enterprise-scale database that has been derived from one or more sources
- It is intended for end-user decision support
- Allows us to move information from multiple sources into a common target area, that managers and decision makers are able to access through reporting tools
- This also serves to relieve the operational sources from query time

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2010. All Rights Reserved.

A data warehouse is a decision support database that is maintained separately from an organization's operational databases. It usually resides on a dedicated server. This database is designed based on what kind of information a company is seeking (e.g., sales, marketing, healthcare membership and providers, etc.). A data warehouse is "the place" for top executives, managers, analysts, and other end-users to mine a rich source of company information. They can ask compelling business questions and find answers in their data, and make key and timely business decisions from their desktops using GUI based On-line Analytical Processing (OLAP) tools.

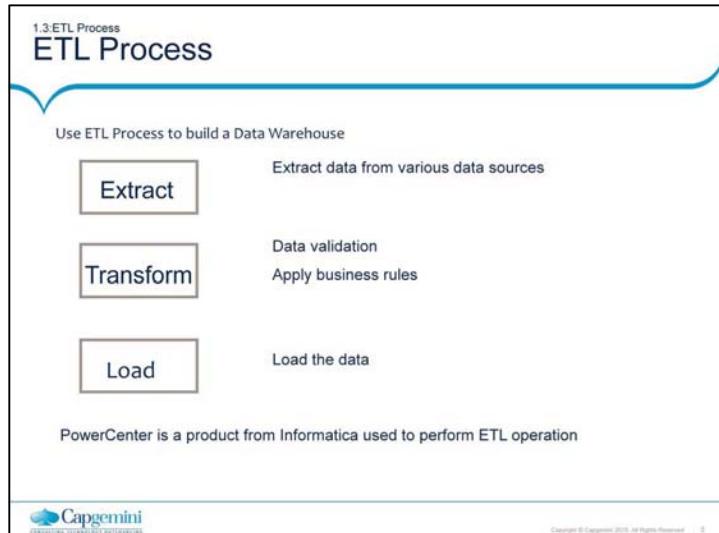


The above Figure 1.1. shows a simplified data warehouse model. The developer puts business rules for data transformation into the metadata repository. There is a transformation process that extracts data from the operational sources like database, legacy systems, and flat files. It transforms data according to the business rules, and loads the data into the data warehouse. From the data warehouse, atomic data flows to various departments for their customized usage. These departmental databases are called data marts.

The end user will use a query tool to look at the metadata in the repository and the transformed data in the warehouse. The data in the data warehouse will be used to cater to dynamic and complex reporting requirements.

When Informatica's PowerCenter is fit into the above model, the developer can use PowerCenter Designer tool to develop mappings and other metadata for the Informatica repository.

## Informatica PowerCenter



To populate the data in the data warehouse, the steps are:

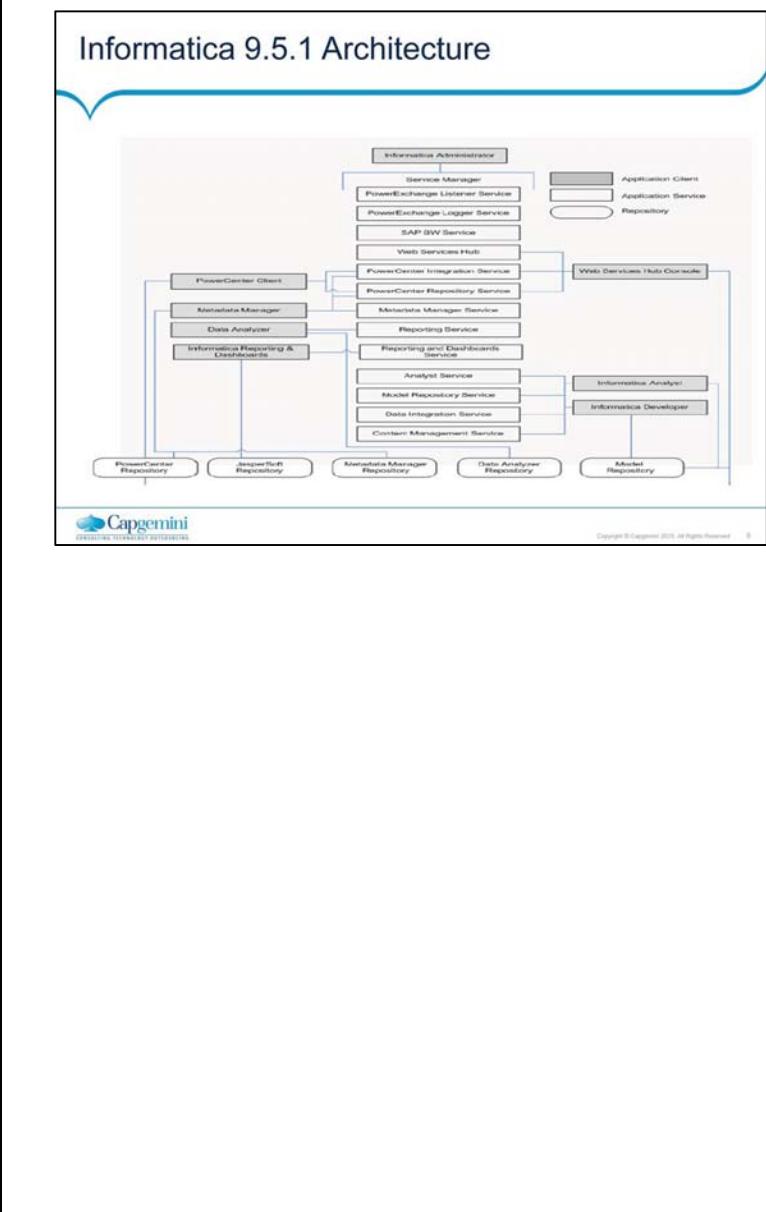
Extract - Extracting appropriate data from existing operational database(s), flat files etc

Transform - Cleansing or scrubbing the data, aggregating, denormalizing and filtering the data

Load - Loading the data into the database

This data population process is also known as the data transformation process. Moving data from the operational databases to the data warehouse needs to be done via extraction tools. Operational data needs to be mapped to the target data warehouse. As a part of the data movement, data transformation is performed as specified by the meta data rules developed during the data modeling stage. PowerCenter is a product from Informatica used to perform ETL operation.

## Informatica PowerCenter



1.4:Informatica Power Center

## Components

- Power Center includes the following components:
  - Power Center domain
  - Power Center repository
  - Administration Console
  - Power Center Client
  - Repository Service
  - Integration Service
  - Web Services Hub
  - SAP BW Service
  - Reporting Service
  - Data Analyzer
  - Metadata Manage
  - Power Center Repository Reports

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2010. All Rights Reserved.

PowerCenter provides an environment that allows to load data into a centralized location, such as a datamart, data warehouse, or operational data store (ODS). Data can be extracted from multiple sources, transformed according to business logic built in the client application, and load the transformed data into file and relational targets. Informatica provides the following integrated components:

**PowerCenter domain.** The Power Center domain is the primary unit for management and administration within PowerCenter. The Service Manager runs on a PowerCenter domain. The Service Manager supports the domain and the application services.

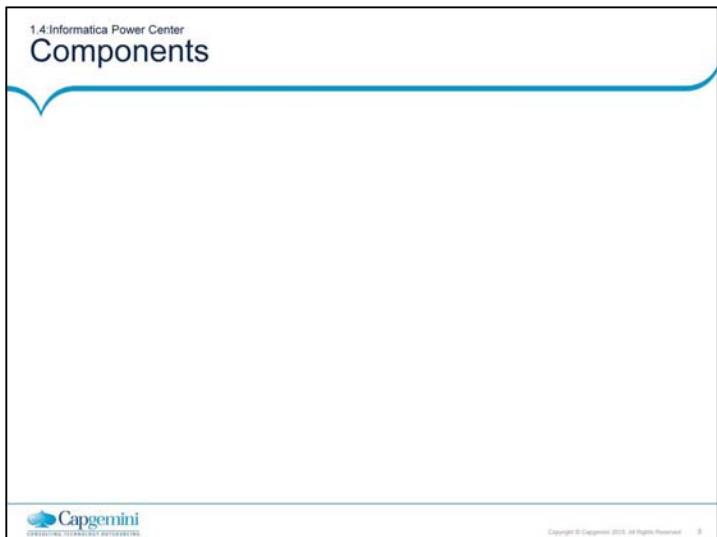
**PowerCenter repository.** The PowerCenter repository resides in a relational database. The repository database tables contain the instructions required to extract, transform, and load data

**Administration Console.** The Administration Console is a web application that you use to administer the PowerCenter domain and PowerCenter security

**PowerCenter Client.** The PowerCenter Client is an application used to define sources and targets, build mappings and mapplets with the transformation logic, and create workflows to run the mapping logic. The PowerCenter Client connects to the repository through the Repository Service to modify repository metadata.

**Repository Service.** The Repository Service accepts requests from the PowerCenter Client to create and modify repository metadata and accepts requests from the Integration Service for metadata when a workflow runs

**Integration Service.** The Integration Service extracts data from sources and loads data to targets



Web Services Hub. Web Services Hub is a gateway that exposes PowerCenter functionality to external clients through web services

SAP BW Service. The SAP BW Service extracts data from and loads data to SAP BW. If you use the PowerExchange for SAP NetWeaver BW Option, you must create and enable a SAP BW Service in the PowerCenter domain

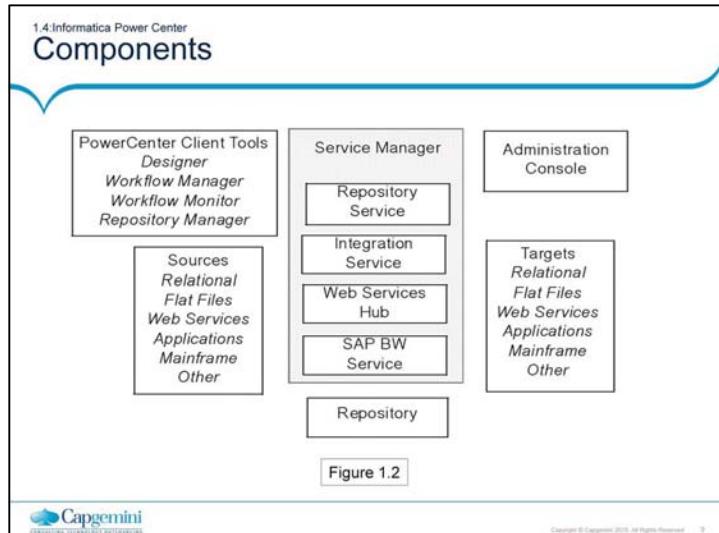
Reporting Service. The Reporting Service runs the Data Analyzer web application. Data Analyzer provides a framework for creating and running custom reports and dashboards. You can use Data Analyzer to run the metadata reports provided with PowerCenter, including the PowerCenter Repository Reports and Data Profiling Reports. Data Analyzer stores the data source schemas and report metadata in the Data Analyzer repository

Data Analyzer. Data Analyzer provides a framework to perform business analytics on corporate data. With Data Analyzer, you can extract, filter, format, and analyze corporate information from data stored in a data warehouse, operational data store, or other data storage models

Metadata Manager Service. The Metadata Manager Service runs the Metadata Manager web application. You can use Metadata Manager to browse and analyze metadata from disparate metadata repositories. Metadata Manager helps you understand and manage how information and processes are derived, how they are related, and how they are used. Metadata Manager stores information about the metadata to be analyzed in the Metadata Manager repository

PowerCenter Repository Reports. PowerCenter Repository Reports are a set of prepackaged Data Analyzer reports and dashboards to help you analyze and manage PowerCenter metadata

## Informatica PowerCenter



### Sources

PowerCenter accesses the following sources:

Relational: Oracle, Sybase ASE, Informix, IBM DB2, Microsoft SQL Server, and Teradata.

File: Fixed and delimited flat file, COBOL file, XML file, and web log.

Application: Business sources such as Hyperion Essbase, WebSphere MQ, IBM DB2 OLAP Server, JMS, Microsoft Message Queue, PeopleSoft, SAP NetWeaver, SAS, Siebel, TIBCO, and webMethods.

Mainframe: Mainframe databases such as Adabas, Datacom, IBM DB2 OS/390, IBM DB2 OS/400, IDMS, IDMS-X, IMS, and VSAM.

Other: Microsoft Excel, Microsoft Access, and external web services.

### Targets

PowerCenter can load data into the following targets:

Relational: Oracle, Sybase ASE, Sybase IQ, Informix, IBM DB2, Microsoft SQL Server, and Teradata.

File: Fixed and delimited flat file and XML.

Application: Business sources such as Hyperion Essbase, WebSphere MQ, IBM DB2 OLAP Server, JMS, Microsoft Message Queue, mySAP, PeopleSoft EPM, SAP BW, SAS, Siebel, TIBCO, and webMethods.

Mainframe: Mainframe databases such as IBM DB2 for z/OS, IMS, and VSAM.

Other: Microsoft Access and external web services.

## Informatica PowerCenter

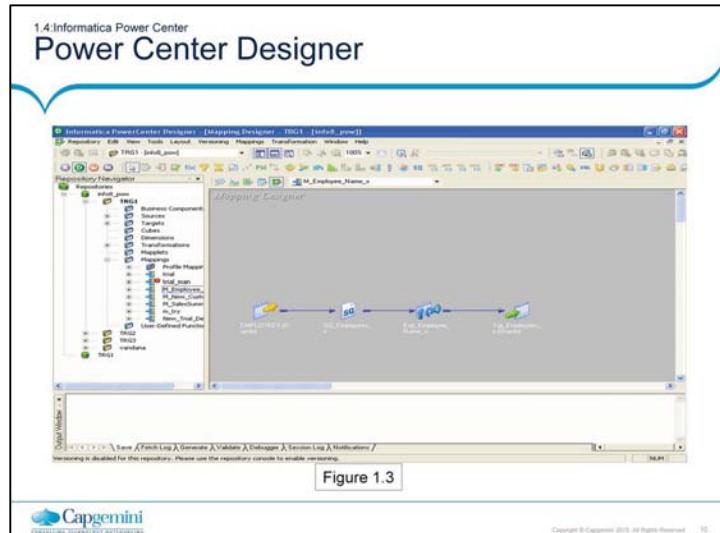


Figure 1.3

The PowerCenter Designer has five tools to build mappings and mapplets so as to specify how to move and transform data between sources and targets.

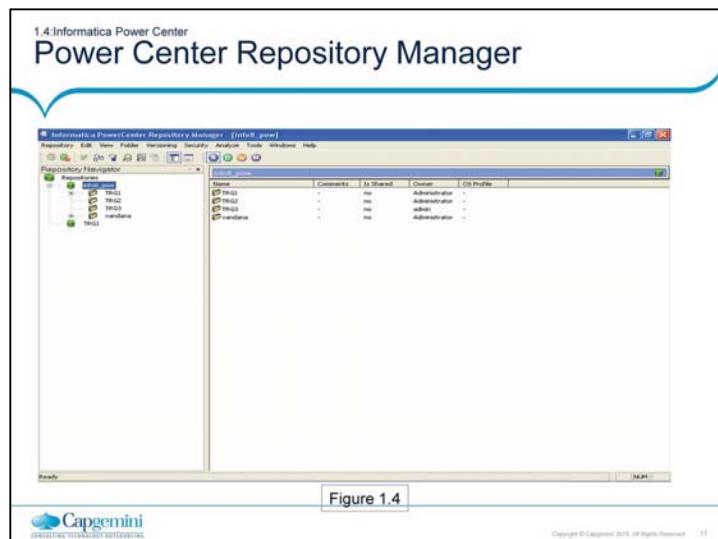
The five tools are as follows:

- Source Analyzer
- Target designer
- Transformation Developer
- Mapplet Designer
- Mapping Designer

The Designer helps create source definitions, target definitions, and transformations to build the mappings.

The Designer allows to work with multiple tools at one time and to work in multiple folders and repositories at the same time. It also includes windows so as to view folders, repository objects, and Tasks.

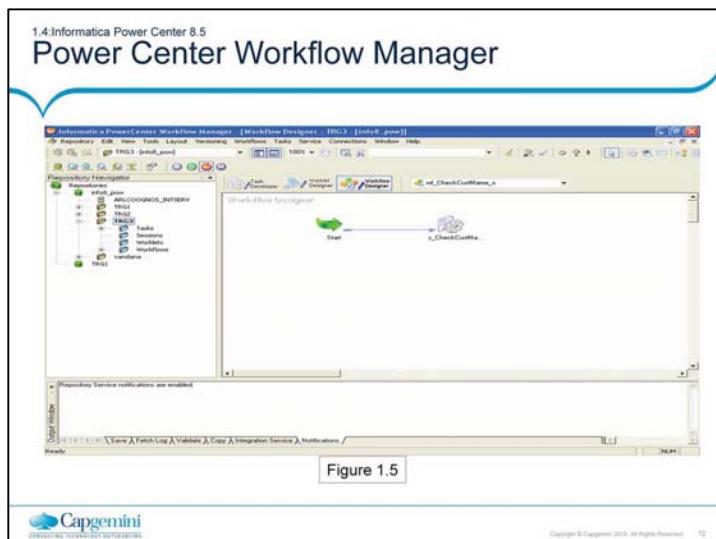
## Informatica PowerCenter



The Repository Manager client tool allows to navigate through multiple folders and repositories and perform basic repository Tasks. The Repository Manager can be used to:

- Add a repository
- Remove a repository
- Connect to a repository
- Search for repository objects, etc.

Informatica PowerCenter



The Workflow Manager is used to define a set of instructions called a Workflow to execute mappings built in the Designer. Generally, a Workflow contains a session and any other Task to be performed when a session is executed.

## Informatica PowerCenter

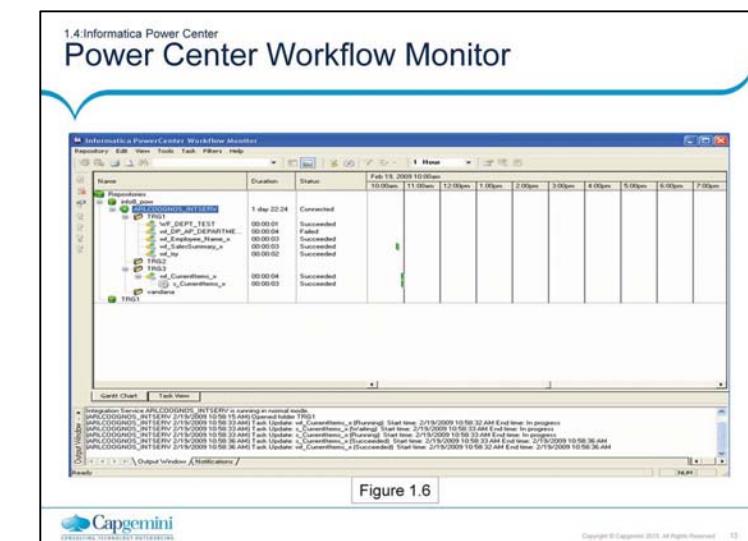


Figure 1.6

The Workflow Monitor is a tool that allows to monitor data loading process.

## Informatica PowerCenter

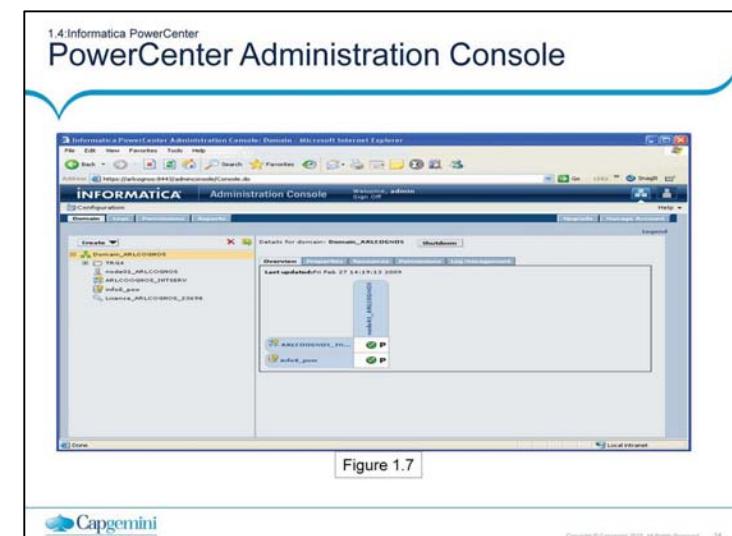


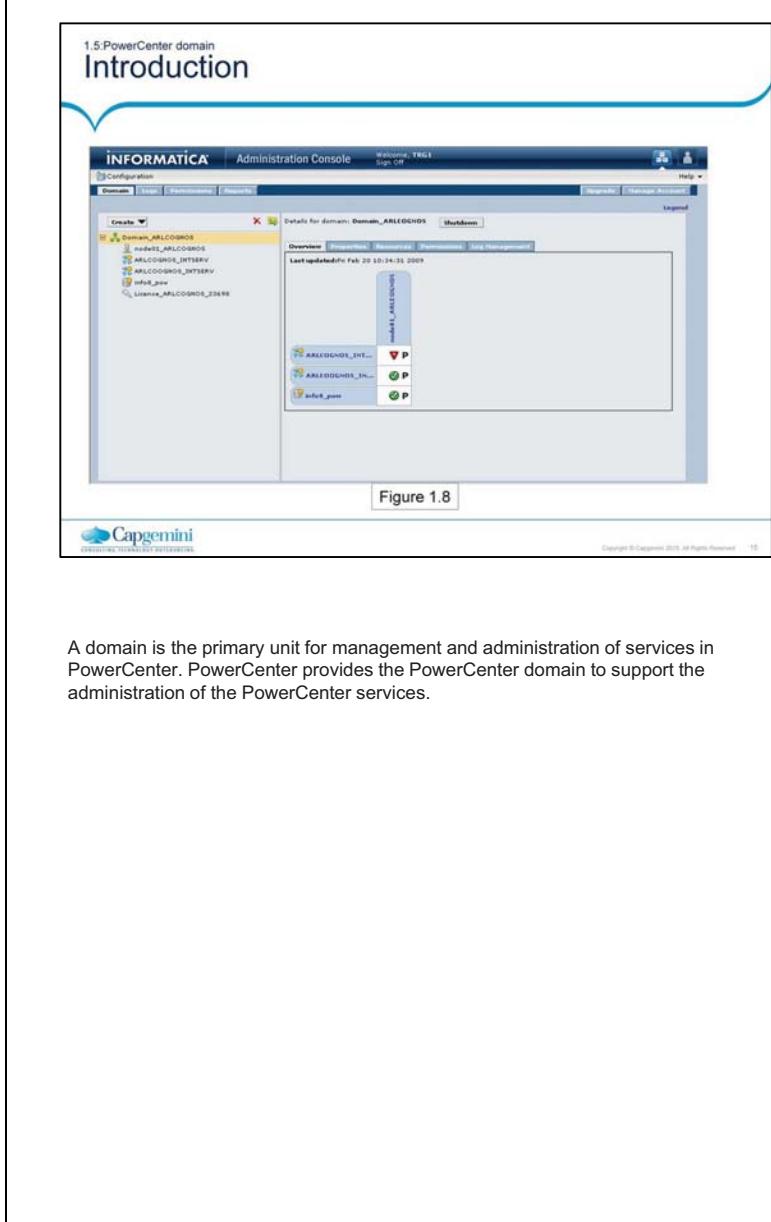
Figure 1.7

The PowerCenter Administration Console is the administration tool you use to administer the PowerCenter domain and PowerCenter security. Use the Administration Console to perform the following tasks:

Domain administrative tasks. Manage logs, domain objects, user permissions, and domain reports. Domain objects include services, nodes, grids, folders, and licenses.

Security administrative tasks. Manage users, groups, roles, and privileges.

## Informatica PowerCenter



1.6 Integration Service

## Introduction

- The Integration Service moves data from sources to targets based on workflow and mapping metadata stored in a repository.
- When a workflow starts, the Integration Service retrieves mapping, workflow, and session metadata from the repository.
- It extracts data from the mapping sources and stores the data in memory while it applies the transformation rules configured in the mapping.

 Capgemini  
Engineering Services for Business

Copyright © Capgemini 2010. All Rights Reserved. 10



Figure 1.10

Copyright © Capgemini 2010. All Rights Reserved. 11

#### Steps to Create Integration Service:

In the Administration Console, click Create > Integration Service.

The Create New Integration Service dialog box appears.

Enter values for the following Integration Service options Service Name, Location, License, Assign, Run the Service on Grid, Primary Node, Backup Nodes, Domain for Associated Repository Service, Associated Repository Service, Repository User Name, Repository Password, Security Domain, Data Movement Mode

Click OK.

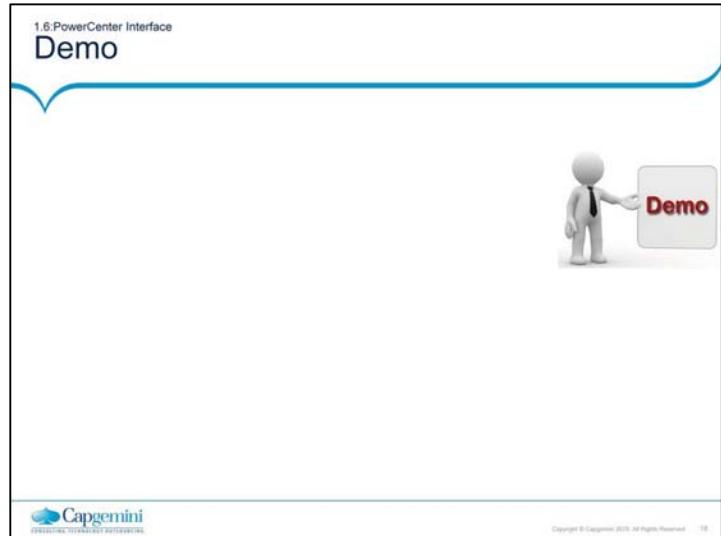
If you do not specify an associated repository, the following message appears:

No Repository Service is associated with this Integration Service. Select an associated Repository Service to view a list of available codepages.

You cannot enable the Integration Service until you assign the code page for each Integration Service process node.

Click OK.

## Informatica PowerCenter



PowerCenter Graphical User Interface

1.7 Repository Manager  
**Creating Repository Service**

- Open PowerCenter Administration Console and then click on Domain->Create from the drop down list select Repository Service

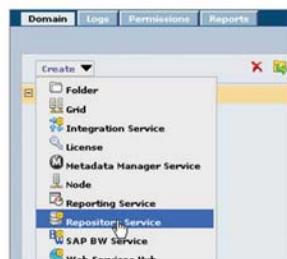
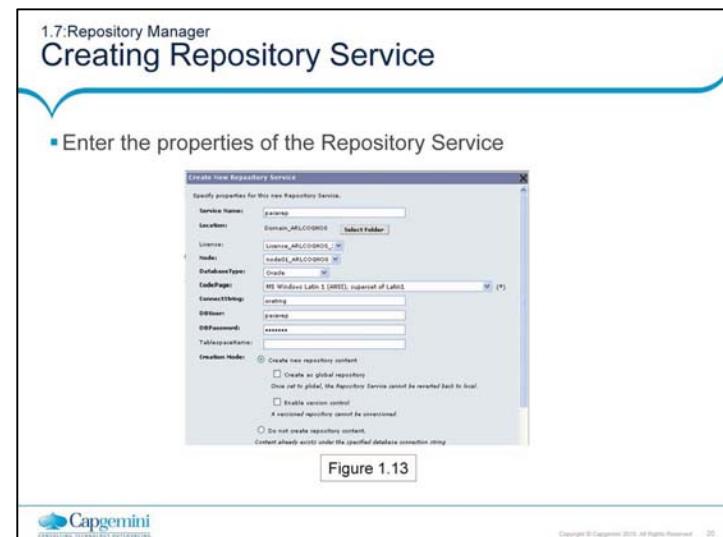
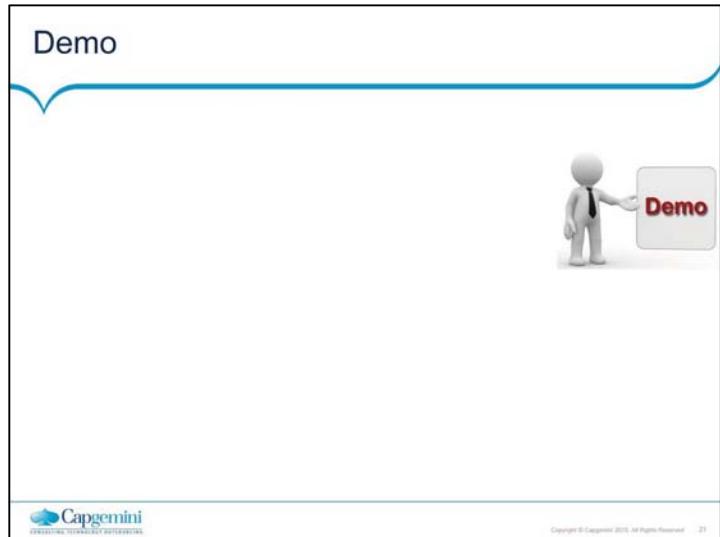


Figure 1.12

Capgemini



Copyright © Capgemini 2010. All Rights Reserved.



Following are the steps to create a repository:

In the Navigator of the Administration Console, select the folder where you want to create the Repository Service.

Note: If you do not select a folder, you can move the Repository Service into a folder after you create it.

2. Click Create > Repository Service.

The Create New Repository Service dialog box appears.

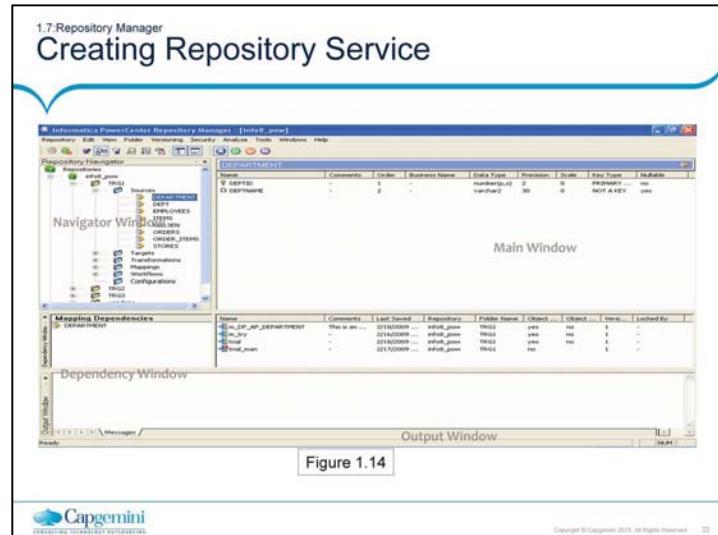
3. Enter values for the following Repository Service options:

Service Name, Location, License, Node, Database Type, CodePage, ConnectString, DBUser, DBPassword, TableSpaceName, Creation Mode

4. If you create a Repository Service for a repository with existing content and the repository existed in a different PowerCenter domain, verify that users and groups with privileges for the Repository Service exist in the current domain.

5. Click OK.

The created repository service will be seen in the folder. Then select the Repository Service in Navigator window and then the repository service can be enabled or disabled using Enable or Disable



The Repository Manager allows to perform basic repository tasks. It has the following four windows:

Navigator Window - Displays all the objects that are created in the Repository Manager, the Designer, and the Workflow Manager. It is organized first by repository, then by folder.

Main Window - Provides properties of the object selected in the Navigator window. The columns in this window will change depending upon the object selected in the Navigator window.

Dependency Window - Shows dependencies on sources, targets, mappings, and shortcuts for objects selected in either the Navigator or Main window.

Output Window - Provides the output of procedures executed within Repository Manager

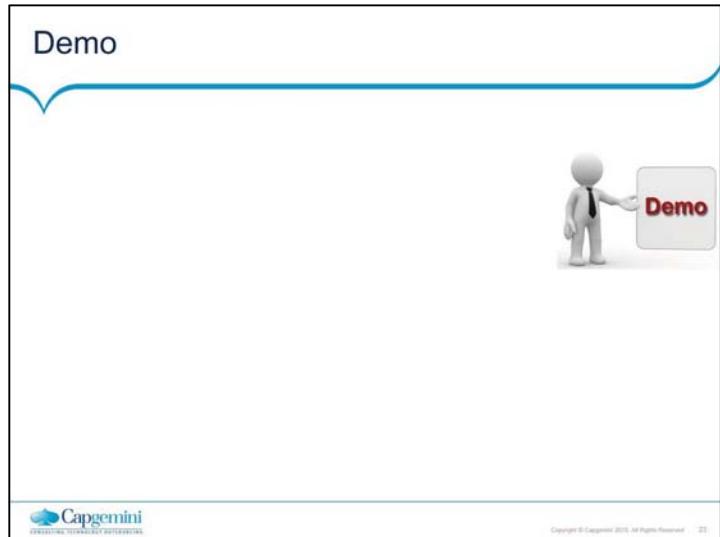
Other Tasks that can be performed within the Repository Manager include:

Create folders and edit folder properties

Establish users, groups, privileges, and edit folder permissions

Search metadata and perform dependency analyses

View locks and unlock objects



Steps to connect to the repository using Repository Manager for the first time:

Launch the Repository Manager tool, choose Start | Programs | Informatica PowerCenter | Client | Repository Manager.

Click on the Add Repository menu option.

Enter the following information:

Repository name

Username

5. Choose Repository | Connect or right click on the repository name and choose Connect.

6. The Connect to Repository dialog box appears.

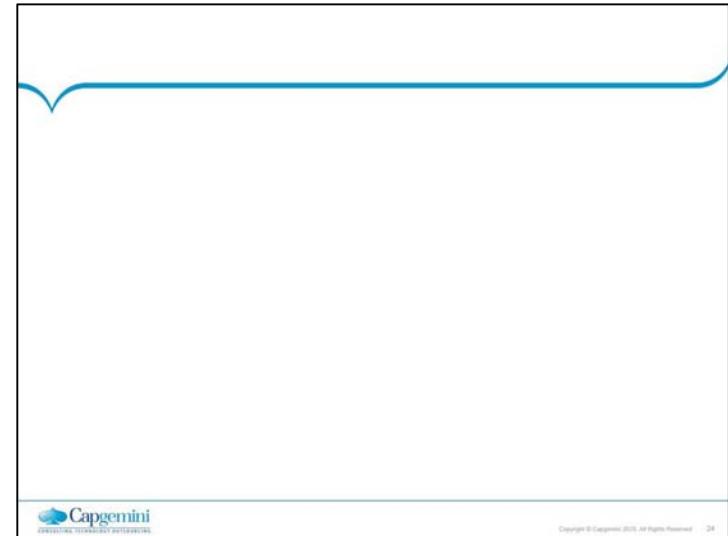
7. Enter the repository username and password.

8. Click More.

9. The Connect to Repository dialog box expands.

10. Enter the host name of the machine the Repository Server is running on and the port number the Repository Server uses for connections.

11. Click Connect.



Copyright © Capgemini 2010. All Rights Reserved. 30

The Repository Server now opens a connection to the database, and a new icon representing the repository appears in the Repository Manager. Folders within the repository now appear underneath the icon for that repository.

Steps to connect to a repository that has been accessed before:

- Launch the Repository Manager tool, choose Start | Programs | Informatica PowerCenter | Informatica PowerCenter Client | Repository Manager.
- Select the icon for the repository and click the Connect button on the toolbar.
- Alternatively, the repository icon can be selected, choose Repository | Connect.
- Enter the repository username and password.
- Click Connect.

Follow the same steps to connect to the repository using PowerCenter Designer, Workflow Manager, and Workflow Monitor.

1.8 User and Group Management

## Introduction

- To access the services and objects in the Power Center domain and to use the Power Center applications, you must have a user account.
- The tasks you can perform depend on the type of user account you have.
- User account management in Power Center involves the following key components:
  - Users. You can set up different types of user accounts in the Power Center domain.



Copyright © Capgemini 2014. All Rights Reserved. 30

1.8 User and Group Management

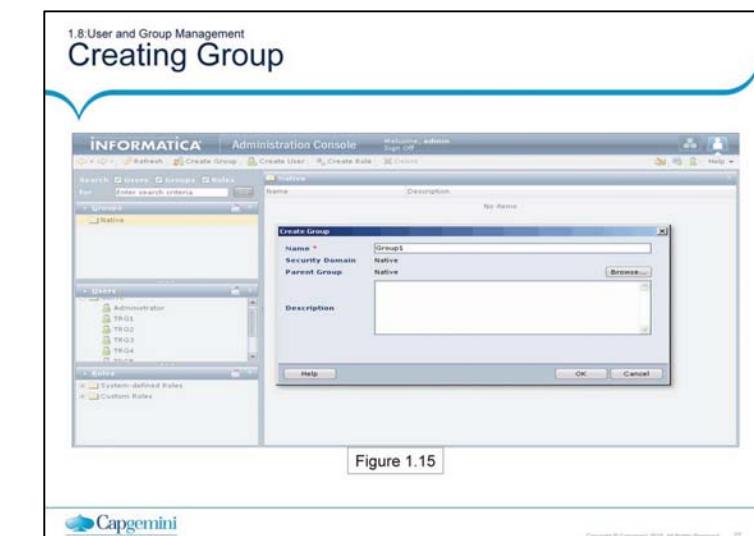
## Introduction

- Groups. You can set up groups of users and assign different roles, privileges, and permissions to each group.
- Privileges and Roles. Privileges determine the actions that users can perform in PowerCenter applications.
- A role is a collection of privileges that you can assign to users and groups.
- Authentication. When a user logs in to PowerCenter, the Service Manager authenticates the user account in the PowerCenter domain and verifies that the user can use the PowerCenter applications.



Copyright © Capgemini 2014. All Rights Reserved. 30

## Informatica PowerCenter



The steps for creating a group are as follows:

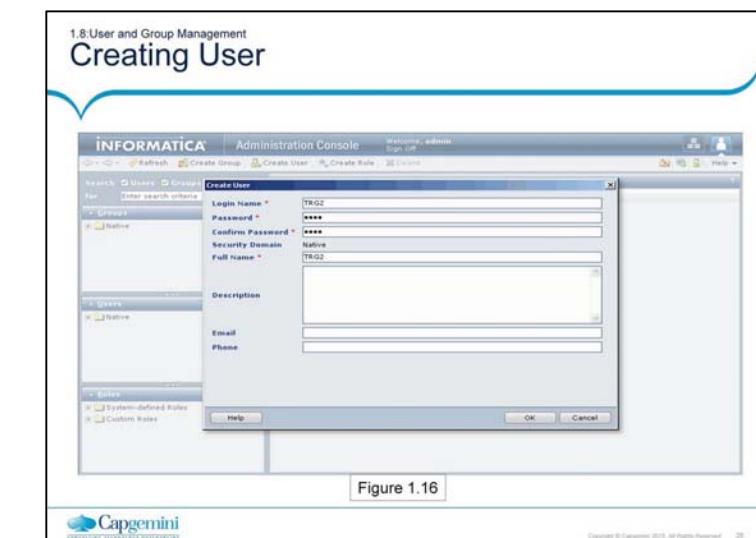
In the Administration Console Screen, Select Configure Security

Select Create Group

Enter Name and Description

Browse and select the Group in case of creating sub grouping

## Informatica PowerCenter



The steps for creating a user are as follows:

In the Administration Console Screen, Select Configure Security

Select Create User

Enter Name, Password, Full Name, Description, Email, Phone

Click OK

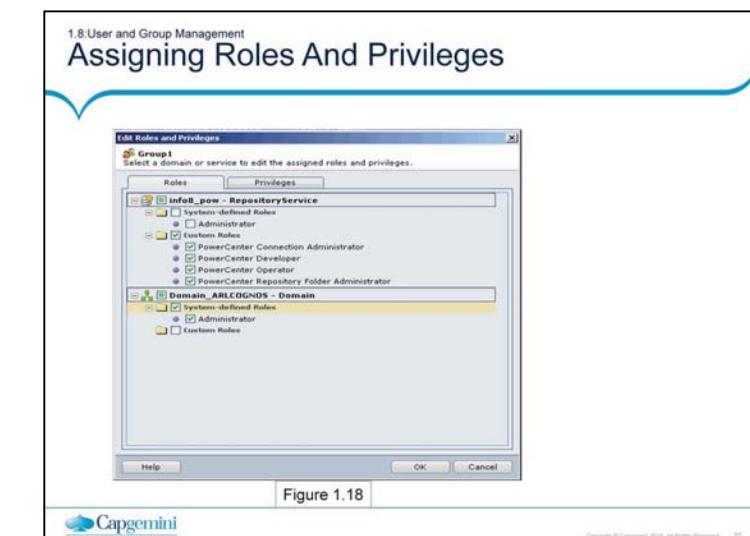


The steps for Assigning the Users to the Group are as follows:

Select the Group from Navigator Window

Overview|Edit |Users

Add the users to the group



The steps for Assigning the Roles to the Group are as follows:

Select Group from Navigator Window

Overview|Edit |Roles

Select the Appropriate Roles from:

System-Defined Roles

Custom Roles

Press OK

The steps for Assigning the Privileges to the Group are as follows:

Select Group from Navigator Window

Overview|Edit |Privileges

Select the appropriate privileges from:

Tools: Includes the privilege to log in to the Administration Console.

Security Administration: Includes privileges to manage users, groups, roles, and privileges.

Domain Administration: Includes privileges to manage the domain, folders, nodes, grids, licenses, and application services.

Press OK

1.9:Folder

## Introduction

- Folders provide a way to organize and store all metadata in the repository
- Create a folder in a repository before connecting to the repository using the Designer or Workflow Manager

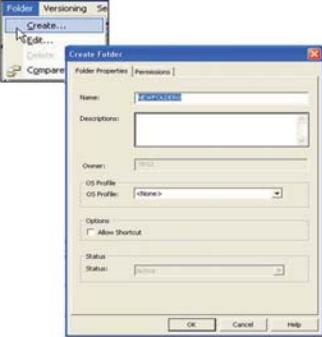


Figure 1.19

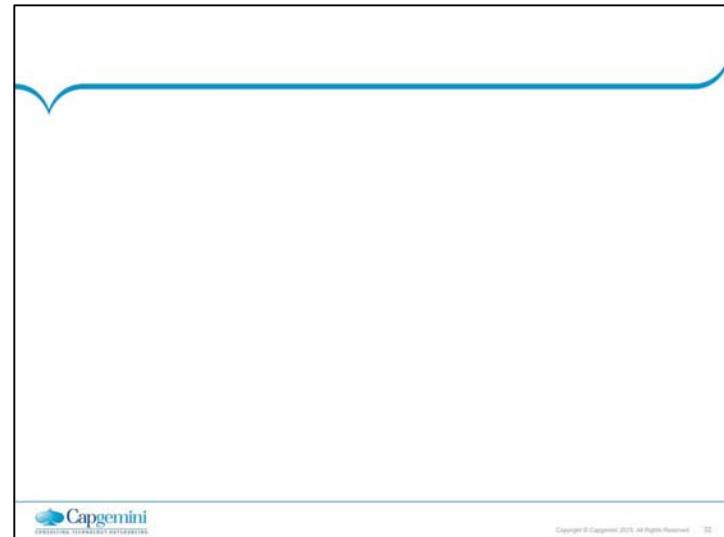
Copyright © Capgemini 2010. All Rights Reserved.

Folders provide a way to organize and store all metadata in the repository, including mappings, schemas, and sessions. Folders are designed to be flexible, to logically organize the repository. Each folder has a set of configurable properties that helps to define how users access the folder. For example, create a folder that allows all repository users to see objects within the folder, but not to edit them. Or, create a folder that allows users to share objects within the folder.

#### Folder Attributes

FOLDER OWNER	- User who serves as focal point to folder permissions
PERMISSIONS	- Rights to read, write, and/or execute objects in a folder
SHARED	- Property that allows users to make shortcuts in a folder
SHORTCUT	- A dynamic link to an object stored in a shared folder
VERSIONS	- Folder iterations that indicate development stages

A folder can be designated as shared by checking 'Allow Shortcuts' in the Folder Properties. When a folder is shared, users can create shortcuts to objects in the folder, thereby sharing the metadata in the folder. Once a folder is designated as shared, it cannot be reverted back to make it non-shared. If a folder is not shared, users will only be able to make copies of objects in the folder.



Copyright © Capgemini 2014. All Rights Reserved.

Folders can be versioned, allowing to keep copies at different stages of development. Versioning provides the flexibility to:

- Revert back to earlier copy of an object
- Re-create an object if one is accidentally deleted
- Save a functioning copy while developing and testing

Folders are created and maintained by the Repository Manager. In the Repository Manager, the users accessing folders, the objects and versions within each folder can be seen.

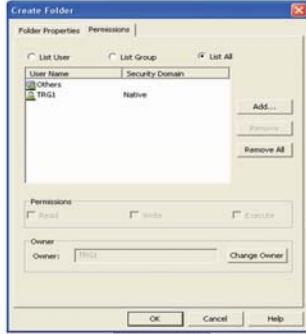
In the Designer, use folders to store sources, transformations, mapplets, targets, and mappings.

In the Workflow Manager, use folders to store Workflows, Tasks, and sessions. When creating a Workflow, include any session or Task in the folder.

Note: Create a folder in a repository before connecting to the repository using the Designer or Workflow Manager.

1.9:Folder  
**Folder Permissions**

- Permissions
  - Read permission
  - Write permission
  - Execute permission



The screenshot shows the 'Create Folder' dialog box. Under 'User Name', 'Others' and 'Native' are selected. Under 'Permissions', 'Read' is checked. Under 'Owner', 'Native' is listed. Buttons at the bottom include 'OK', 'Cancel', and 'Help'.

Figure 1.20

 Capgemini  
Engineering Services for Businesses

#### Permissions

Allows repository users to perform Tasks within a folder. To perform Tasks within a folder having the privilege to perform the Task in the repository as well as the applicable folder permission is a must. The available folder permissions are:

Read permission - Allows to view the folder as well as objects in the folder

Write permission - Allows to create or edit objects in the folder

Execute permission - Allows to execute or schedule a Workflow in the folder

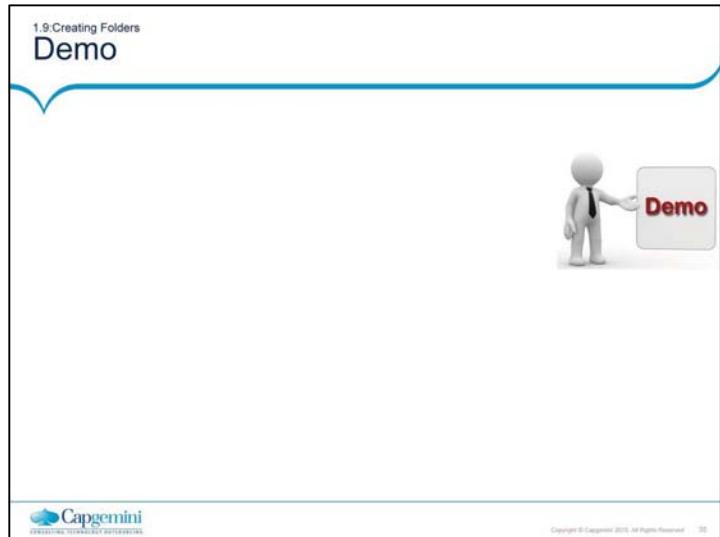
Note: Permissions work in conjunction with privileges. Privileges are actions that a user performs in PowerCenter applications. A user with the privilege to perform certain actions can require permissions to perform the action on a particular object.

## 1.9:Folder Folder Permissions

- Folder permissions can be granted on three levels of security.
  - List User: User of the folder.
  - List Group: Users of a particular group.
  - List All: All the user belonging to repository services.



Copyright © Capgemini 2014. All Rights Reserved 50



Steps to create a Folder:

In the Repository Manager, connect to the repository.

Choose Folder | Create.

Choose Folder Properties

Enter the following information:

Name – Folder name, Required.

Description - Description of the folder, Optional.

Owner - Owner of the folder. Any user in the repository can be the folder owner, Required.

Allow Shortcut - If selected, makes the folder shared, Optional

5. Choose Permissions Tab

Select the Radio Button from List User, List Group, List All

Then select Permission check boxes accordingly

## Summary

- After completing this lesson you now:
  - Understand the ETL Process
  - Know the role of PowerCenter ETL tool in a Data warehouse model
  - Know the various components of PowerCenter
  - Know your work environment in terms of a Repository, Users, and Folders



Summary



Copyright © Capgemini 2014. All Rights Reserved. 30

## Review Question

- Question 1: The three levels of folder permissions are \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_
- Question 2: Informatica Client, and Informatica Server connect to the repository through the \_\_\_\_\_
- Question 3: One can start working in PowerCenter without having an assigned folder.
  - True
  - False
- Question 4: The first step in setting up repository security is to establish groups and users.
  - True
  - False



Copyright © Capgemini 2014. All Rights Reserved.

# Informatica PowerCenter

Lesson 2 : PowerCenter  
Designer

## Lesson Objectives

- In this lesson you will learn about:
  - Concept of a Mapping
  - PowerCenter Designer : Modes
  - Analyze Sources
  - Creating a Target Schema
  - Mapping Designer
  - Transformation Overview
  - Source Qualifier Transformation
  - Expression Transformation
  - Aggregator Transformation

Copyright © Capgemini 2012. All Rights Reserved.

2.1: Mapping  
**Concept**

- Mappings depict the flow of data from source to target
- Integration service uses these instructions to read, transform, and write data
- Designer provides set of transformations to build mappings



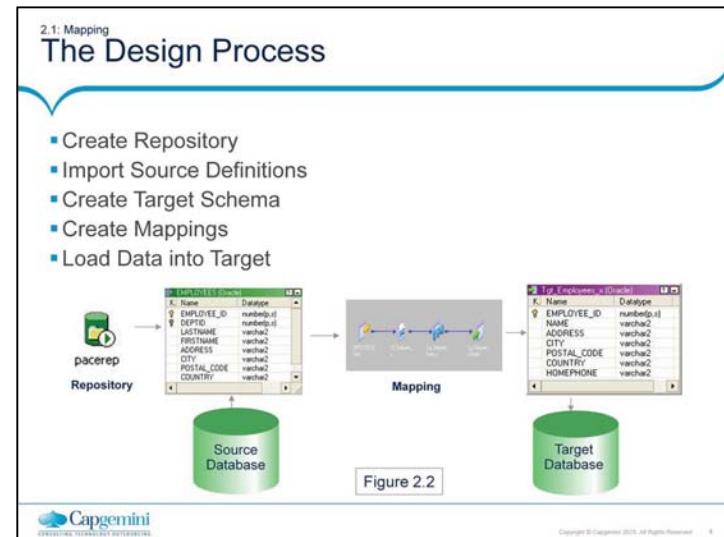
The diagram illustrates a data mapping process. It starts with a source icon representing 'EMPLOYEES (Oracle)' containing a yellow folder icon. An arrow points from this source to a transformation icon labeled 'SQL\_Employment\_1' containing a blue document icon with a question mark. Another arrow points from this transformation to a second transformation icon labeled 'Exp\_Employee (Oracle)' containing a green document icon with a pencil. A final arrow points from this second transformation to a target icon representing 'Tgt\_Employee (Oracle)' containing a green document icon with a checkmark.

Figure 2.1

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2010. All Rights Reserved.

When the Integration service loads data, it uses the instructions configured, to read, transform, and write data.



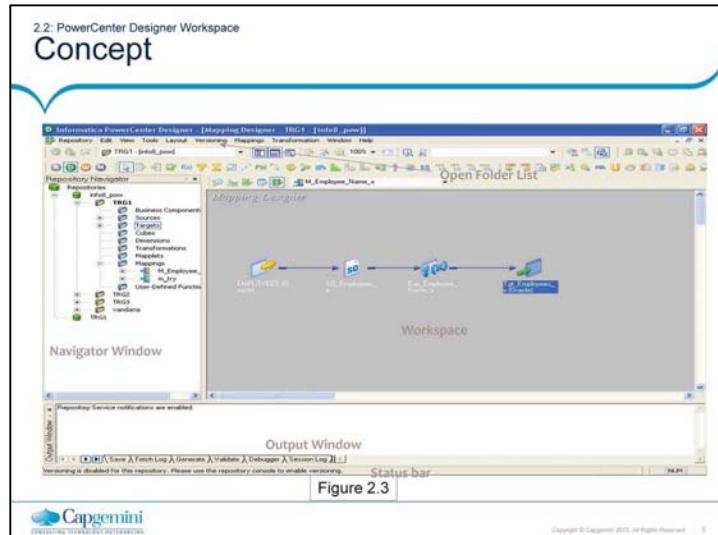
**Create Repository** - The first step is to create the Informatica repository from the Repository Server Administration Console. This repository will hold all related metadata and drive Informatica's extraction and transformation process. Once the repository and necessary folders have been created, the actual design process can begin. This is accomplished in the Designer client application, where developers will spend the majority of their time.

**Import Source Definitions** - The next step will be to put the source definitions into the repository. The Source Analyzer within the Designer tool is used for this process

**Create Target Schema** - After the source definitions are in the repository, the target schema should be created. This can be designed in the Target designer (also with the Designer), reverse engineered from the database, or imported through PowerPlugs

**Create Mappings** - Mappings can be created to link the sources with the targets – the translation of 'business rules'

**Load Data** - The final step is to load the data. The Workflow Manager is used to configure and schedule session Tasks and Workflows to run the mappings. Then, based upon all the information stored in the repository, the Integration service will extract, transform, and load the data



The PowerCenter Designer has five tools which helps to build mappings and mapplets so that one can specify how to move and transform data between sources and targets. The Designer helps to create source definitions, target definitions, and transformations to build the mappings.

The Designer consists of the following :

Navigator window - Use to connect to and work in multiple repositories and folders. Copy and delete objects options are available and shortcuts can be created using the Navigator

Workspace window- Use to view or edit sources, targets, mapplets, transformations, and mappings. Work with a single tool at a time in the workspace. One can use the workspace in default or workbook format

Status bar - Displays the status of the operation being performed

Output window- Provides details while performing certain Tasks, such as saving the work or validating a mapping. Right-click the Output window to access window options, such as printing output text, saving text to file, and changing the font size

2.2: PowerCenter Designer Workspace

## Modes

- The Informatica PowerCenter Designer has five modes:
- Source Analyzer
- Target designer
- Transformation Developer
- Mapplet Designer
- Mapping Designer



-  Source Analyzer
-  Target Designer
-  Transformation Developer
-  Mapplet Designer
-  Mapping Designer

Figure 2.4



The Informatica Designer has five modes:

The Source Analyzer is used to create or reverse-engineer source definitions

The Target designer is used to create or reverse-engineer target schema

The Transformation Developer is used to create reusable transformations

The Mapplet Designer is used to create and edit mapplets. Mapplets are reusable objects that represent a set of transformations

The Mapping Designer can be used to create the source to target mappings that contain the business rules for the server during the extract, transform, and load process

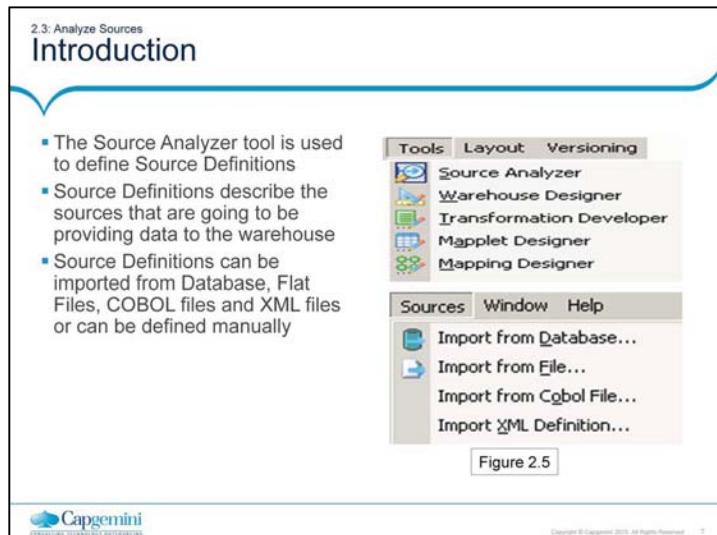


Figure 2.5

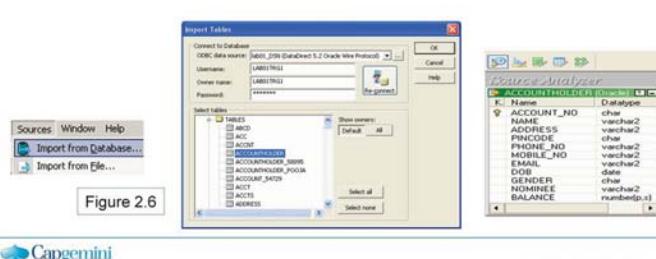
Copyright © Capgemini 2010. All Rights Reserved.

The first tool, the Source Analyzer, is used to identify the sources that will be used to build the data mart or warehouse and create repository definitions for those sources. Source definitions describe the sources that are going to be providing data to the warehouse. The different ways to create source definitions in the Repository.

- Import from Database
- Import from File
- Import from COBOL File
- Import from XML...
- Create manually
- Import from SAP
- Import from PeopleSoft
- Import from Siebel

2.3: Analyze Sources  
**Creating A Source Definition**

- To import the source definition from a database table the following details are required:
  - An ODBC connection using the Data Direct Driver for the database
  - Username and Password to connect to the appropriate database



The screenshot shows the 'Import Tables' dialog box and the 'Source Analyzer' window side-by-side.

**Import Tables Dialog:**

- Connect to Database: ODBC data source: ABCD\_123 (DataDirect 5.2 Oracle Wire Protocol)
- Username: LAB01PR02
- Owner name: LAB01PR02
- Password: \*\*\*\*\*
- Show owners: Default
- Tables list: ABCD, ACCT, ACNT, ADDRESS, ACCOUNTHOLDER\_88899, ACCOUNTHOLDER\_99034, ACCOUNT\_SALES, ACCT, ACCT, ADDRESS
- Buttons: OK, Cancel, Help

**Source Analyzer Window:**

- Title: SOURCE ANALYZER
- Table: ACCOUNTHOLDER (Oracle)
- Columns:
 

K. Name	Datatype
ACCOUNT_NO	char
NAME	varchar2
ADDRESS	char
PHONE_NO	varchar2
EMAIL_ID	varchar2
DOB	date
GENDER	char
NOMINEE	varchar2
BALANCE	number(dp,2)

Figure 2.6

Relational source definitions can be imported from database tables, views, and synonyms. When importing a source definition, import the following source metadata:

- Source name
- Database location
- Column names
- Datatypes
- Key constraints

To import a source definition, first connect to the source database from the client machine using a properly configured ODBC data source or gateway. Read permission is required on the database object.

After importing a relational source definition, optionally enter business names for the table and columns. Also manually define key relationships, which can be logical relationships created in the repository that do not exist in the database.

The screenshot shows a slide titled "2.4:Importing a Database Source Definition" with the word "Demo" in large bold letters. Below it is a bulleted list: "■ Demo Topic". To the right, there is a small 3D character holding a sign that also says "Demo". At the bottom left is the Capgemini logo, and at the bottom right is a copyright notice: "Copyright © Capgemini 2012. All Rights Reserved".

#### Steps to import a source definition:

In the Source Analyzer, choose Sources | Import from Database.

Select the ODBC data source used to connect to the source database. To create or modify an ODBC data source, click the Browse button to open the ODBC Administrator. Create the appropriate data source and click OK.

Select the new ODBC data source.

Enter a database username and password to connect to the database.

Note: The username must have the appropriate database permissions to view the object. The owner name for database objects will have to be specified in order to use them as sources.

Click Connect.

If no table names appear or if the table to be imported does not appear, click All.

Scroll down through the list of sources to find the source to be imported. Select the relational object or objects to be imported. Hold down the Shift key to select a block of sources within one folder, or hold down the Ctrl key to make non-consecutive selections within a folder.

Click OK.

2.4: Creating A Target Schema

## Introduction

- The Target designer provides a GUI interface for creating and customizing the logical target schema
- Target schema can be created in three ways:
  - Automatic Creation
  - Import from Database
  - Manual Creation



Figure 2.7

Copyright © Capgemini 2010. All Rights Reserved.

The next step in the design process is to create the target schema. This is accomplished in the second component of the Designer, the Target designer.

The Target designer provides a GUI interface for creating and customizing the logical target schema. Target definitions can be customized as needed.

When the customization is complete, one can create the physical targets in the database from SQL scripts executed from the Target designer.

The methods of creating target schema are:

- Automatic Creation
- Import from Database
- Manual Creation

2.4: Creating A Target Schema

## Definition

- Dragging and dropping a source definition from the Navigator window to the Target designer workspace automatically creates a duplicate of the source definition as a target definition

The screenshot shows the Informatica PowerCenter Designer interface. On the left, the 'Repository Navigator' window displays a tree structure under the 'info11\_pow' repository, specifically the 'TRG1' business component. It lists 'Sources', 'Targets', 'Cubes', 'Dimensions', 'Transformations', and 'Mapplets'. Under 'Targets', there are two entries: 'CLV\_CUSTOMER' and 'tgt\_Employee'. On the right, the 'Target Designer' window shows a workspace with a single target definition named 'tgt\_Employee\_x'. A tooltip 'Drag and drop' points to the target definition in the workspace. The Capgemini logo is visible at the bottom left, and a copyright notice 'Copyright © Capgemini 2010. All Rights Reserved.' is at the bottom right.

Figure 2.8

Target definitions are stored in a separate section of the repository.

Automatic target creation is useful for creating:

Staging tables, such as those needed in Dynamic Data Store

Target tables that will mirror much of the source definition

Target tables that will be used to migrate data from different databases, e.g., Sybase to Oracle

**2.4: Creating A Target Schema**

## Creating A Target Definition

- To import the target definition from a database table the following details are required:
- An ODBC connection using the Data Direct Driver for the database
- Username and Password to connect to the appropriate database

The screenshot shows two windows side-by-side. On the left is the 'Import Tables' dialog box, which has 'Import from Database...' selected. It shows an ODBC data source named 'Sales\_Prod (Established 5.2 Oracle View Protocol)', a 'Username' field with 'SalesUser', and a 'Owner name' field with 'SALES'. Below these are dropdown menus for 'Select Tables' and 'Select Columns'. On the right is the 'Target Designer' window titled 'Tgt\_Employees.v'. It displays a table structure with columns: K. Name, E.EMPLOYEE\_ID, E.EMPLOYEE\_NAME, E.ADDRESS, E.CITY, E.POSTAL\_CODE, E.COUNTRY, and E.HOMEPHONE. The 'E' column is marked with a key icon, indicating it is the primary key.

Figure 2.9

When a target definition is imported from a relational table, the Designer imports the following target details:

Target name - The name of the target

Database location - Specify the database location of a relational source is to be imported. Specify a different location while editing the target definition in the Target designer and while configuring a session

Column names - The names of the columns

Data types - The Designer imports the native datatype for each column

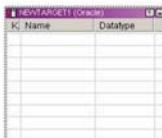
Key constraints - The constraints in the target definition can be critical, since they may prevent data from moving into the target if the Integration service violates a constraint during a Workflow. For example, if a column contains the NOT NULL constraint and one fail to map data to this column, the Integration service cannot insert new records into the target table

Key Relationships - The Target designer can be customized to automatically create primary-foreign key relationships. Choose Tools | Options and select the Format tab. Check Import Primary and Foreign Keys. Also logical relationships can be created in the repository. Key relationships do not have to exist in the database

**2.4: Creating A Target Schema**

## Creating A Manual Target

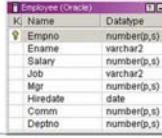
- Create empty definition



Add desired columns



- Finished target definition



**Figure 2.10**

Target Definitions can also be created manually via a menu option or the toolbar. An empty definition is created, you can enter the table name, columns, datatypes, precision, and key information.

Manual target creation is useful for creating tables whose columnar information cannot be copied from source definitions, such as time dimension tables or aggregate tables.

The screenshot shows a slide titled "2.4Creating Target Schema" with the main title "Demo". Below the title is a bullet point "■ Demo Topic". To the right of the text is a 3D white stick figure holding a grey rectangular sign with the word "Demo" in red. At the bottom left is the Capgemini logo, and at the bottom right is a small copyright notice: "Copyright © Capgemini 2012. All Rights Reserved." followed by a page number "10".

Steps to create a Target schema:

I Automatic Target Schema:

Drag a relational source definition into the Target designer workspace, the Designer will create a relational or flat file target definition that matches the source definition.

II From the Database:

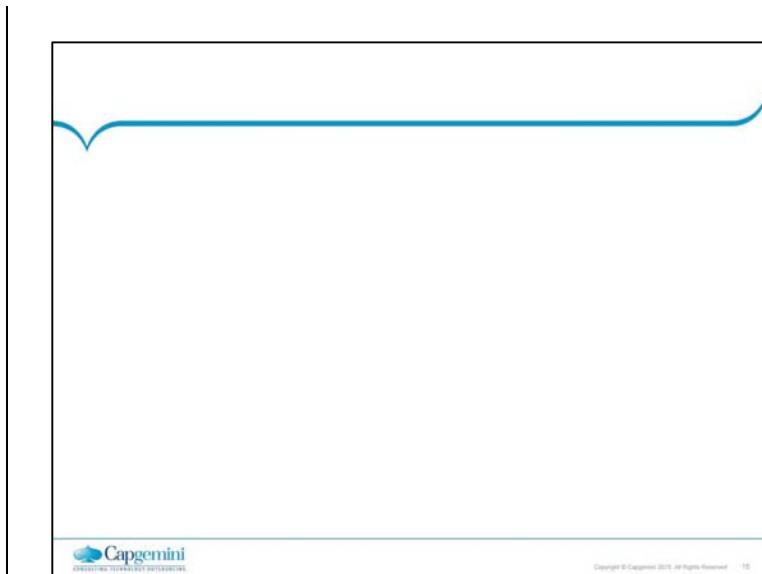
To import a relational target definition:

In the Target designer, choose Targets | Import from Database.

Select the ODBC data source used to connect to the target database. To create or modify an ODBC data source first, click the Browse button to open the ODBC Administrator. After creating or modifying the ODBC source, continue with the following steps.

Enter the username and password needed to open a connection to the database, and click Connect. If the user is not the owner of the table to be used as a target, specify the owner name.

Drill down through the list of database objects to view the available tables which can be used as targets.



Select the relational table or tables to import the definitions into the repository. Hold down the Shift key to select a block of tables, or hold down the Ctrl key to make non-contiguous selections. Also the Select All and Select None buttons can be used to select or clear all available targets.

Click OK. The selected target definitions now appear in the Navigator under the Targets icon.

Choose Repository | Save.

#### III Manual:

In the Target designer, choose Targets | Create.

Enter a name for the target and select the target type. One can create a flat file target definition by choosing Flat File for the target type.

The target name entered is the name of the new table in the database when a relational target definition is created. Follow any database-specific naming conventions.

4. Click Create.

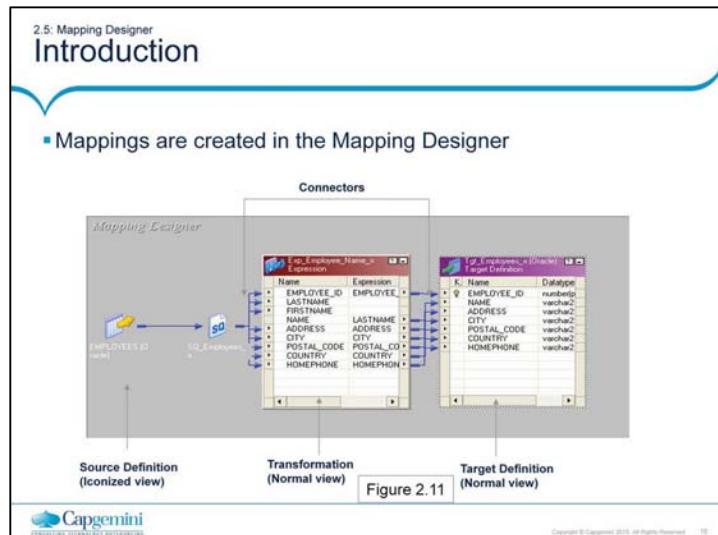
5. An empty table structure appears in the workspace. (It may be covered by the dialog box.) The new target definition also appears within the Navigator window.

6. If another target definition is to be created, enter a new target name and target type and click Create. Repeat this step for each target to be created.

7. Click Done when all target definitions are created.

8. Configure the target definition i.e., add the required columns.

9. Choose Repository | Save.



The above diagram shows how a mapping looks like. It has the various transformations which the source data has to go through. Every mapping must contain the following components:

- Source definition - Describes the characteristics of a source table or file
- Transformation - Modifies data before writing it to targets. Use different transformation objects to perform different functions
- Target Definition - Defines the target table or flat file
- Connectors - Connect sources, targets, and transformations so the Integration service can move the data as it transforms it

The Mapping Designer displays objects in three different views:

- Iconized - Shows an icon of the object with the object name
- Normal - Shows the columns in the ports tab and the input and output port indicators. One can connect objects that are in the normal view
- Edit - Shows the object properties. Switch between the different tabs and configure the object in this view

2.6: Transformation

## Overview

- A transformation is a repository object that generates, modifies, or passes data
- Transformations in a mapping represent the operations the Integration service performs on the data
- Data passes into and out of transformations through ports that are linked in a mapping or mapplet
- Ports can be one of the three types:
  - Input
    - data is received
  - Output
    - data is provided
  - Input / Output
    - data passes directly through transformation unchanged

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2010. All Rights Reserved. 13

The Designer provides a set of transformations that perform specific functions. For example, an Aggregator transformation performs calculations on groups of data. Perform the following Tasks to incorporate a transformation into a mapping:

Create the transformation - Create it in the Mapping Designer as part of a mapping, in the Mapplet Designer as part of a mapplet, or in the Transformation Developer as a reusable transformation

Configure the transformation - Each type of transformation has a unique set of options that can be configured

Link the transformation to other transformations and target definitions - Drag one port to another to link them in the mapping or mapplet

2.6: Transformation

## Types

- Active
  - Changes the number of rows passing through it
- Passive
  - Number of rows passing through it do not change
- Connected
  - Connected to the transformations in the data flow
- Unconnected
  - Not connected to any transformations in the data flow

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2010. All Rights Reserved. 10

Active - An active transformation can change the number of rows that pass through it, such as a Filter transformation that removes rows that do not meet the filter condition.

Passive - A passive transformation does not change the number of rows that pass through it, such as an Expression transformation that performs a calculation on data and passes all rows through the transformation.

Connected - Connected transformations are connected to the data flow.

Unconnected - An unconnected transformation is not connected to other transformations in the mapping. It is called within another transformation, and returns a value to that transformation.

**2.6: Transformation  
Tabs**

- Default tabs
  - Transformation
  - Ports
  - Properties
  - Metadata Extensions
- Additional tabs
  - Condition
  - Sources
  - Normalizer

**Transformation tabs**

Figure 2.12

Each transformation has a minimum of four tabs:

Transformation – allows to rename a transformation, switch between transformations, enter transformation comments, and make a transformation reusable

Ports – allows to specify level attributes such as port name, datatype, precision, scale, primary/foreign keys, nullability

Properties – allows to specify the amount of detail in the session log, and other properties specific to each transformation

Metadata Extensions – allows to extend the data stored in the repository by associating information with individual repository objects

The additional tabs are :

Condition – allows to enter conditions for e.g. Joiner or Filter Transformation

Sources – allows to specify additional source definitions e.g. Source Qualifier Transformation

Normalizer – allows to enter new ports for the Normalizer Transformation

2.6: Transformation Active and Passive Transformations	
Active	Passive
Source Qualifier	Expression
Normalizer	Lookup
Aggregator	Stored Procedures
Filter	Sequence Generator
Router	External Procedure
Rank	
Sorter	
Update Strategy	
Joiner	

Copyright © Capgemini 2014. All Rights Reserved.

#### Active Transformations:

- Source Qualifier - represents all data queries from the source
- Normalizer - normalizes records from VSAM or relational sources
- Aggregator - performs aggregate calculations
- Filter - serves as a conditional filter
- Router - serves as a group conditional filter
- Rank - limits records to top or bottom range
- Sorter - sorts the input rows in ascending or descending order
- Update Strategy - allows for logic to insert, update, delete, or reject data
- Joiner – allows extraction of data from heterogeneous sources
- Application Source Qualifier - Represents the rows that the Integration Service reads from an application, such as an ERP source, when it runs a session.
- Transaction Control - Defines commit and rollback transactions.
- XML Generator - Reads data from one or more input ports and outputs XML through a single output port.
- XML Parser - Reads XML from one input port and outputs data to one or more output ports.
- XML Source Qualifier - Represents the rows that the Integration Service reads from an XML source when it runs a session.

2.6: Transformation

## Active and Passive Transformations

Active	Passive
Application Source Qualifier	
Transaction Control	
XML Generator	
XML Parser	
XML Source Qualifier	

 Capgemini  
PROVIDING INTEGRATED BUSINESS SERVICES

Copyright © Capgemini 2010. All Rights Reserved. 21

### Passive Transformations:

- Expression - performs simple calculations
- Lookup - looks up values and passes to other objects
- Stored Procedures - calls a stored procedure and captures return values
- Sequence Generator - generates unique ID values
- External Procedure - Calls a procedure in a shared library

Some transformation can be both active or passive which are as follows:

- Java - Executes user logic coded in Java.
- Complex - Transforms data in unstructured and semi-structured formats.
- Custom - Calls a procedure in a shared library or DLL
- SQL - Executes SQL queries against a database.

These transformations will be discussed in the further lessons.

**2.7: Source Qualifier Transformation**

## Introduction

- Represents the source record set queried by the server
- Mandatory in mappings using relational and flat file sources
- Can be used to
  - Define custom query, Join data, Source Filter, Number of sorted ports, Select Distinct

Figure 2.13

The Source Qualifier displays the transformation datatypes. The transformation datatypes in the Source Qualifier determine how the source database binds data when the Integration service reads it. Do not alter the datatypes in the Source Qualifier. If the datatypes in the source definition and Source Qualifier do not match, the Designer marks the mapping invalid when saved. Source Qualifier can be used to perform following tasks:

Join data originating from the same source database - One can join two or more tables with primary-foreign key relationships by linking the sources to one Source Qualifier

Filter records when the Integration service reads source data - If a filter condition is included, the Integration service adds a WHERE clause to the default query

Specify an outer join rather than the default inner join - If a user-defined join is included, the Integration service replaces the join information specified by the metadata in the SQL query

Specify sorted ports - If a number for sorted ports are specified, the Integration service adds an ORDER BY clause to the default SQL query

Select only distinct values from the source - If Select Distinct is chosen, the Integration service adds a SELECT DISTINCT statement to the default SQL query

Create a custom query to issue a special SELECT statement for the Integration service to read source data

2.8: Expression Transformation  
**Introduction**

- Used to perform non-aggregate calculations
- Expression statement is entered in an output port
- Can enter only one expression for each output port

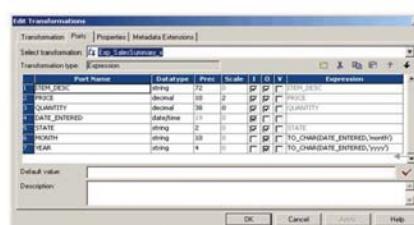
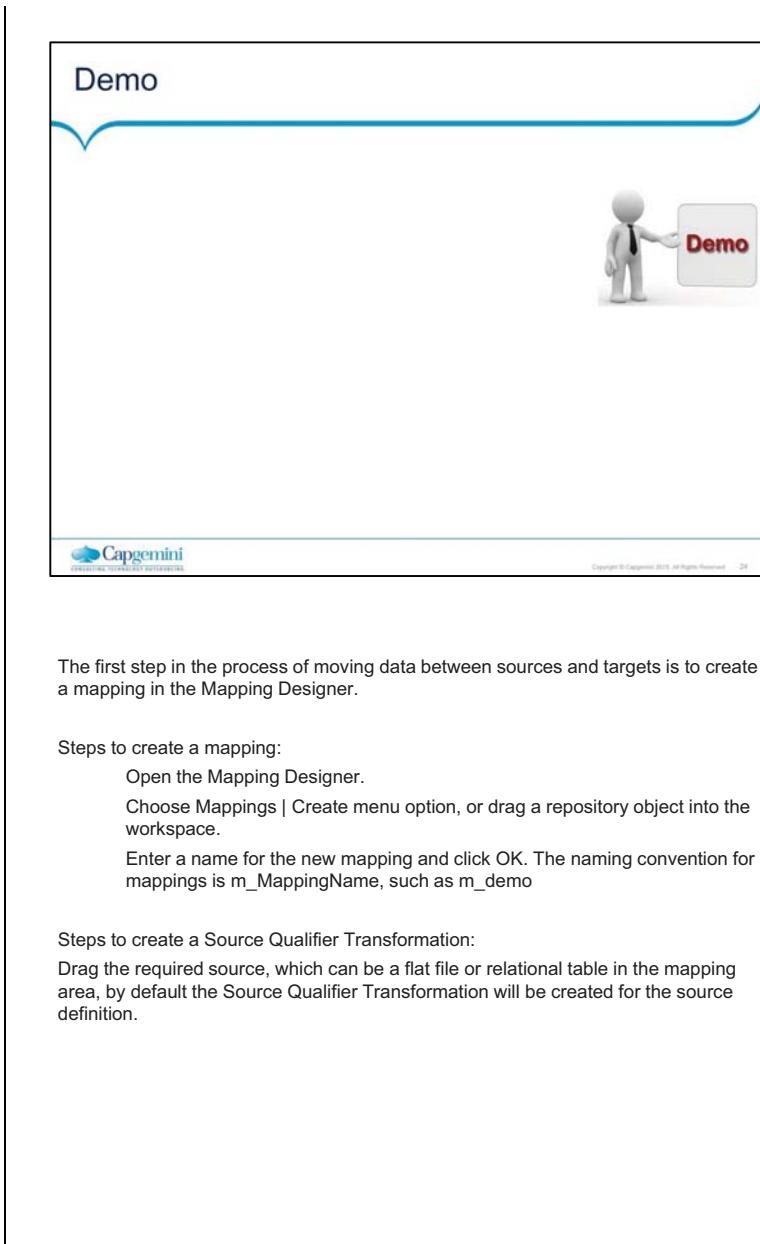


Figure 2.14

  
Copyright © Capgemini 2014. All Rights Reserved.

The Expression Transformation is used to calculate values in a single row. For example, in order to adjust employee salaries, concatenate first and last names, or convert strings to numbers. Use the Expression transformation to perform any non-aggregate calculations.



The first step in the process of moving data between sources and targets is to create a mapping in the Mapping Designer.

Steps to create a mapping:

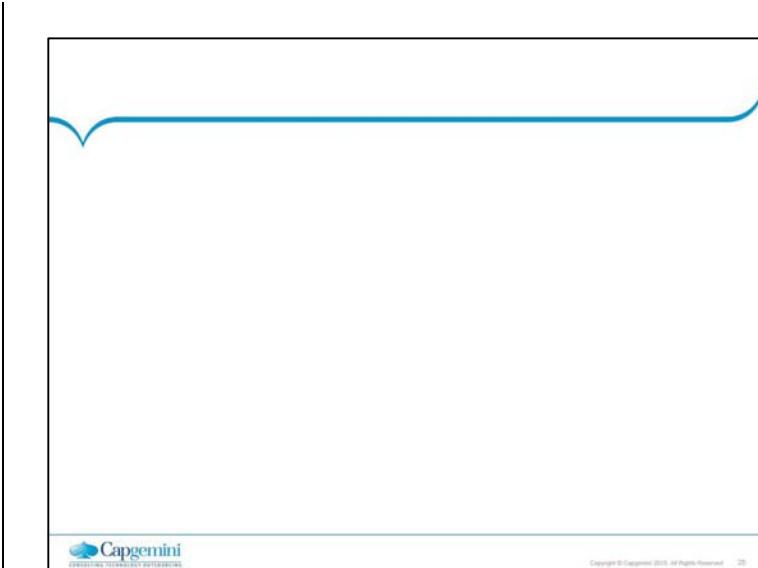
Open the Mapping Designer.

Choose Mappings | Create menu option, or drag a repository object into the workspace.

Enter a name for the new mapping and click OK. The naming convention for mappings is m\_MappingName, such as m\_demo

Steps to create a Source Qualifier Transformation:

Drag the required source, which can be a flat file or relational table in the mapping area, by default the Source Qualifier Transformation will be created for the source definition.



#### Steps to create an Expression Transformation:

In the Mapping Designer, choose Transformation | Create. Select the Expression transformation. Enter a name for it (the convention is EXP\_TransformationName) and click OK.

Create the input ports. If the input transformation is already available, select Link Columns from the Layout menu and then click and drag each port used in the calculation into the Expression transformation. With this method, the Designer copies the port into the new transformation and creates a connection between the two ports. Or, open the Edit dialog box and create each port manually.

Note: In order to make this transformation reusable, create each port manually within the transformation.

Repeat the previous step for each input port which has to be added to the expression.

Create the output ports (O) needed, making sure to assign a port datatype that matches the expression return value.

Click the small button that appears in the Expression section of the dialog box and enter the expression in the Expression Editor. To prevent typographic errors, where possible, use the listed port names and functions.

Port names used as part of an expression in an Expression transformation follow stricter rules than port names in other types of transformations:

- A port name must begin with a single- or double-byte letter or single- or double-byte underscore (\_)

- It can contain any of the following single- or double-byte characters: a letter, number, underscore (\_), \$, #, or @

Check the expression syntax by clicking Validate. If necessary, make corrections to the expression and check the syntax again. Then save the expression and exit the Expression Editor.

Connect the output ports to the next transformation or target.

Choose Repository | Save.

2.9: Aggregator Transformation

## Introduction

- Used to perform aggregate operations like sum, max etc.
- Active Transformation
- Have to group by values
- Aggregate expression is entered in an output port

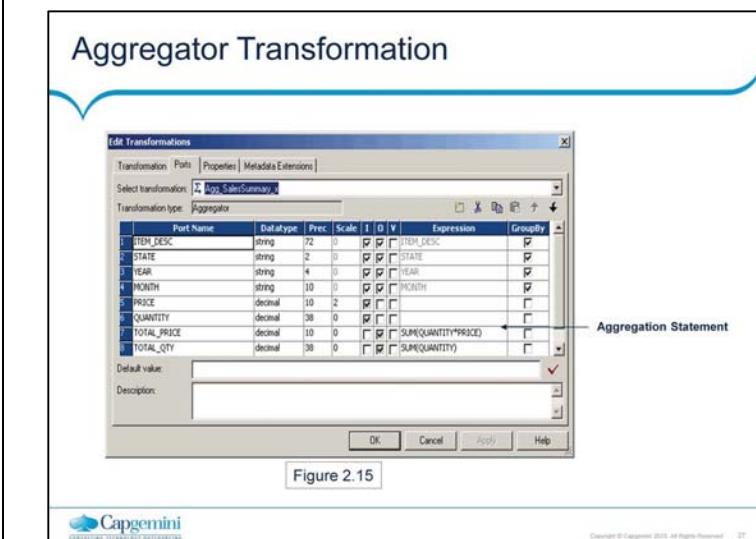
 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2010. All Rights Reserved. 30

The Aggregator Transformation allows to perform aggregate calculations, such as average and sum.

To configure ports in the Aggregator Transformation we can:

- Enter an aggregate expression in any output port, using conditional clauses or non-aggregate functions in the port
- Create multiple aggregate output ports
- Configure any input, input/output, output, or variable port as a group by port, and use non-aggregate expressions in the port
- Improve performance by connecting only the necessary input/output ports to subsequent transformations, reducing the size of the data cache
- Use variable ports for local variables



### Components of the Aggregator Transformation

The Aggregator is an active Transformation, changing the number of rows in the data flow. It must be connected to the data flow. The Aggregator Transformation has the following components and options:

Aggregate expression - Entered in an output port. Can include non-aggregate expressions and conditional clauses

Group by port - Indicates how to create groups.

Sorted input - Is used to improve session performance. To use this option, the data passed to the Aggregator Transformation sorted by Group by port in ascending or descending order

Aggregate Cache - The Integration service stores data in the aggregate cache until it completes aggregate calculations. It stores group values in an index cache and row data in the data cache.

## Sorted Input

- You can improve Aggregator transformation performance by using the sorted input option.
- When you use sorted input, the Integration Service assumes all data is sorted by group and it performs aggregate calculations as it reads rows for a group.
- When necessary, it stores group information in memory.
- To use the Sorted Input option, you must pass sorted data to the Aggregator transformation



Copyright © Capgemini 2012. All Rights Reserved. 30

## Sorted Input

- Used to decrease the use of aggregate caches since the Integration Service assumes all data is sorted by group.
- As the Integration Service reads rows for a group, it performs aggregate calculations and stores group information in memory.
- When using sorted inputs, pass sorted data through the Aggregator.
- The Sorted Input option reduces the amount of data cached during the session and improves performance.

Contd



Copyright © Capgemini 2010. All Rights Reserved. 30

Ex: for calculating sum of salaries department wise, if data comes in random order, Integration service needs to use cache for holding data until all the entries for a particular department is calculated (deptno 10,20....).

By using sorted input, Integration service does not use cache and just performs calculations assuming the data comes in sorted order thus improvement in performance.

2.9: Aggregator Transformation

## Aggregate Cache

- The aggregator stores data in the aggregate cache until it completes aggregate calculations.
- When you run a session that uses an aggregator transformation, the Integration Service creates index and data caches in memory to process the transformation.
- Index cache contains group values and data cache consists of row values.

Contd

 Capgemini  
CONSULTING SERVICES FOR BUSINESS

Copyright © Capgemini 2010. All Rights Reserved. 50

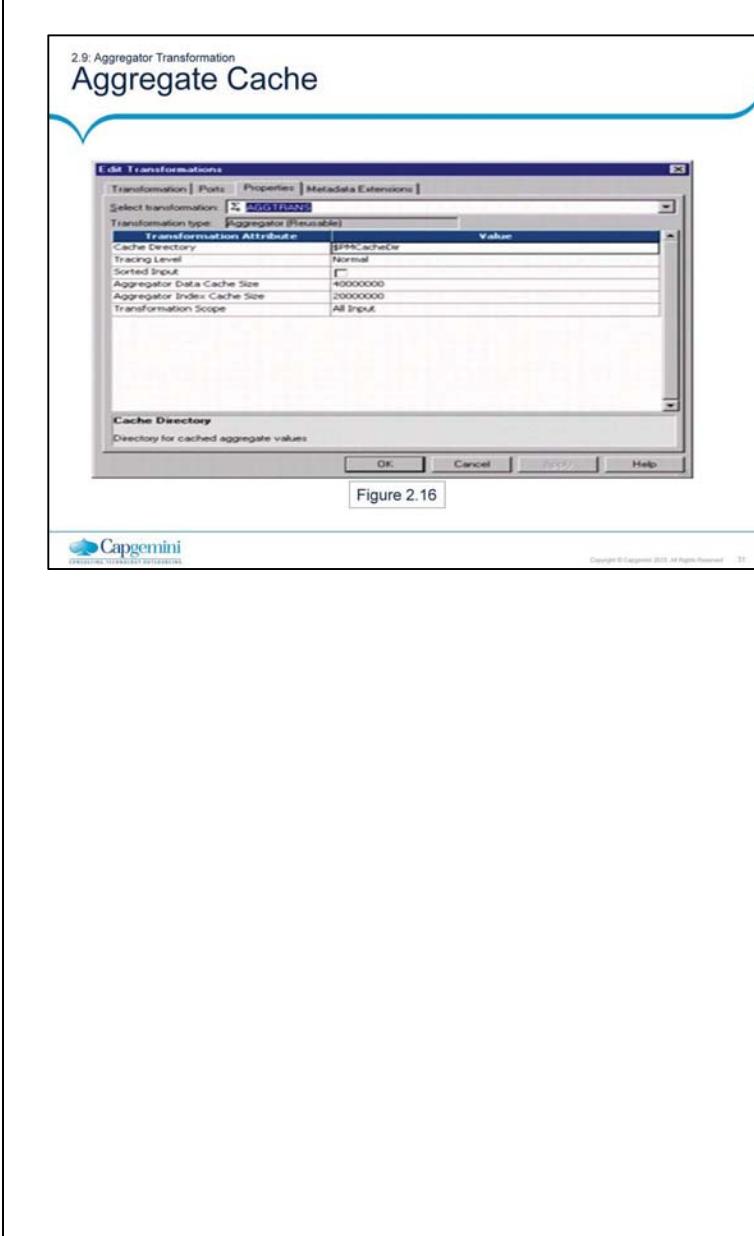
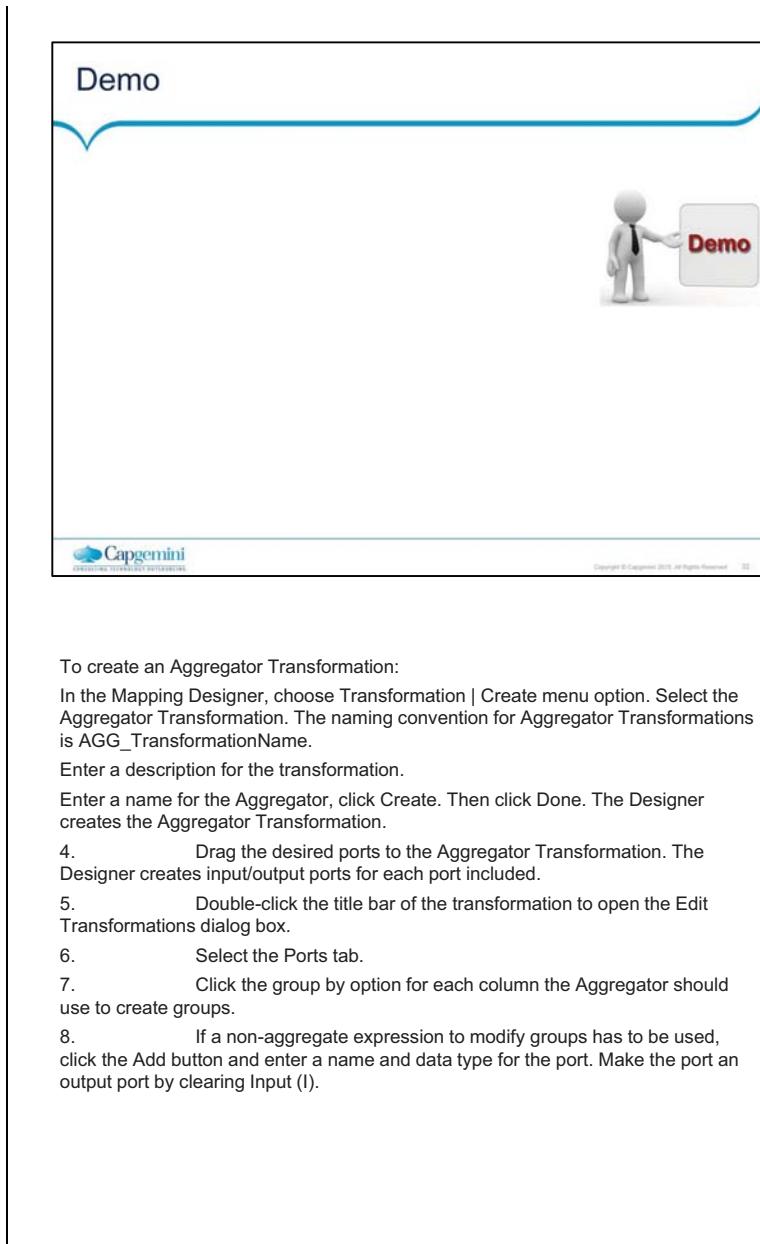


Figure 2.16

Copyright © Capgemini 2014. All Rights Reserved. 11



To create an Aggregator Transformation:

In the Mapping Designer, choose Transformation | Create menu option. Select the Aggregator Transformation. The naming convention for Aggregator Transformations is AGG\_TransformationName.

Enter a description for the transformation.

Enter a name for the Aggregator, click Create. Then click Done. The Designer creates the Aggregator Transformation.

4. Drag the desired ports to the Aggregator Transformation. The Designer creates input/output ports for each port included.
5. Double-click the title bar of the transformation to open the Edit Transformations dialog box.
6. Select the Ports tab.
7. Click the group by option for each column the Aggregator should use to create groups.
8. If a non-aggregate expression to modify groups has to be used, click the Add button and enter a name and data type for the port. Make the port an output port by clearing Input (!).

## Summary

- After completing this lesson you now:
  - Know about the PowerCenter Designer Component
  - Know how to create source definitions
  - Know how to create target definitions
  - What is a mapping
  - What is a transformation
  - Know how to create a mapping



Copyright © Capgemini 2014. All Rights Reserved. 10

## Review Question

- Question 1: An \_\_\_\_\_ transformation changes the number of rows passing through it
- Question 2: The Designer has five modes \_\_\_, \_\_\_, \_\_\_, \_\_\_, and \_\_\_
- Question 3: An expression statement can be entered in an input port
  - True
  - False
- Question 4: Connected transformations can be seen as a part of the mapping data flow
  - True
  - False

Copyright © Capgemini 2012. All Rights Reserved. 50

## **Informatica PowerCenter**

Lesson 3-Workflow Manager

## Lesson Objectives

- In this module you will learn about:
  - Workflow Manager components
  - Workflow Monitor
  - Database connections
  - Tasks
  - Session Task
  - Debugger



3.1: Workflow Manager components

## Workflow

- It is a set of instructions for the Integration service to perform the data transformation load
- The Integration service retrieves mapping, Workflow, and session metadata from the repository to:
  - Extract data from the source
  - Transform it
  - Load data into the target
- It combines the logic of Session Tasks, other types of Tasks and Worklets



Figure 3.1

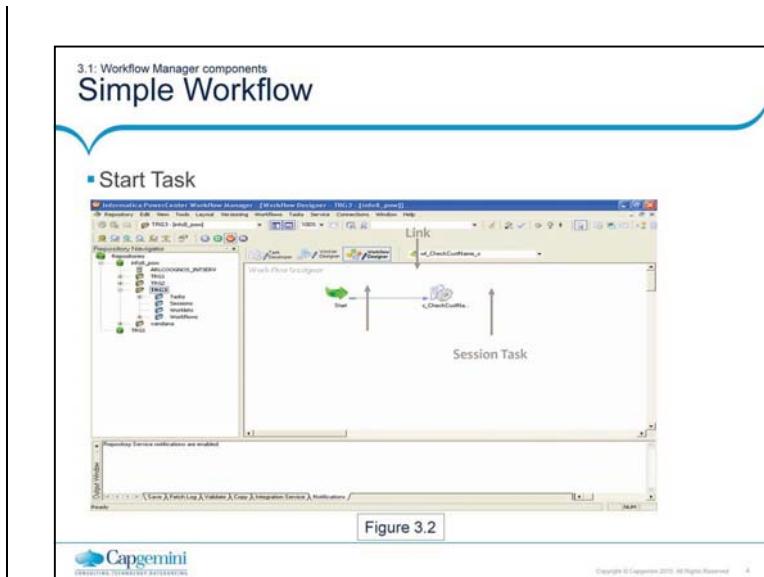
 Capgemini CONSULTING TECHNOLOGY INNOVATING

Copyright © Capgemini 2010. All Rights Reserved 3

Once a mapping is created in the Mapping Designer, the source to target dataflow specification is ready.

A *Workflow* is a set of instructions for the Integration Service to perform the data transformation and loading. When a Workflow starts, the Informatica Integration Service retrieves mapping, Workflow, and session metadata from the repository to extract data from the source, transform it, and load it into the target. It also runs the Tasks in the Workflow. The Integration Service uses Load Manager and Data Transformation Manager (DTM) processes to run the Workflow.

A *Workflow* combines the logic of Session Tasks, other types of Tasks and Worklets. A Session is a type of Task related to the movement of data through the Server. Tasks are the building blocks of a Workflow.



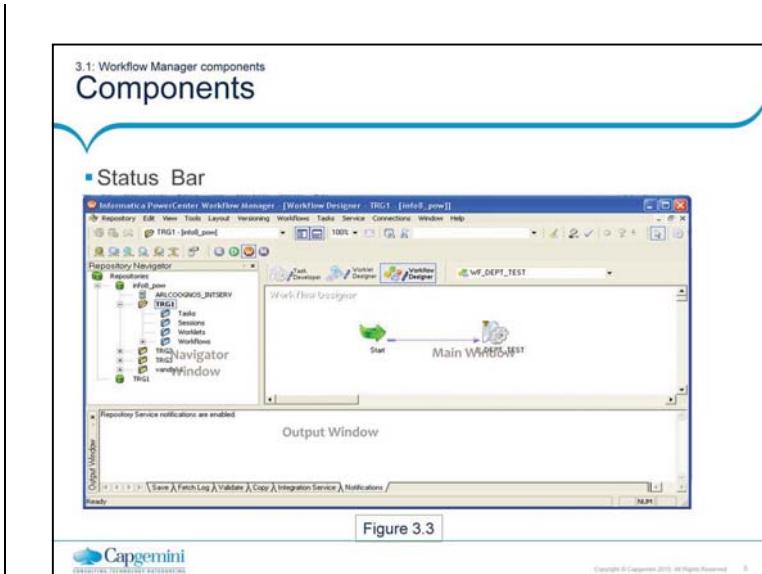
The simplest Workflow is composed of:

- Start Task
- A Link
- A Session Task (and any other Task as required for execution)

The first step to develop a Workflow is to create a new Workflow in the Workflow Designer. A Workflow must contain a Start Task. The Start Task represents the beginning of a Workflow.

When a Workflow is created, the Workflow Designer creates a Start Task and adds it to the Workflow. The Start Task cannot be deleted. The next step is to add Tasks to the Workflow. The Workflow Manager includes Tasks such as the Session Task, the Command Task, and the Email Task to design the Workflow.

Finally, the Workflow Tasks are connected with links to specify the order of execution in the Workflow.



The **Workflow Manager** tool is used to define the workflow to execute mappings built in the Designer. After a Workflow is created, it is run in the Workflow Manager and monitored in the Workflow Monitor.

The Workflow Manager provides several windows:

- The **Navigator Window** - Allows to view servers and all folders within a repository and the contents of those folders
- The **Main Window** – Allows to view all the sessions that have been created and linked to make a complete Workflow
- The **Output Window** - Contains messages from the server, such as success or failure to schedule or start sessions. The Output window contains the following tabs:
  - **Save** - Displays messages when a Workflow, a Worklet, or a Task is used. The Save tab displays a validation summary when a Workflow or a Worklet is saved
  - **Fetch Log** - Displays messages when the Workflow Manager fetches objects from the repository
  - **Validate** - Displays messages when a Workflow, a Worklet, or a Task is validated
  - **Copy** - Displays messages when repository objects are copied
  - **Server** - Displays messages from the Integration Service
  - **Notifications** - Displays messages from the Repository Service

3.1: Workflow Manager components

## Development Tools

- Task Developer
  - To develop various Tasks like Session, Command etc in a Workflow
- Worklet Designer
  - To develop reusable Workflows
- Workflow Designer
  - To develop Workflows

Figure 3-4

Capgemini  
CONSULTING TECHNOLOGY BUSINESS

Copyright © Capgemini 2008. All Rights Reserved.

The Workflow Manager has three development tools:

#### Task Developer

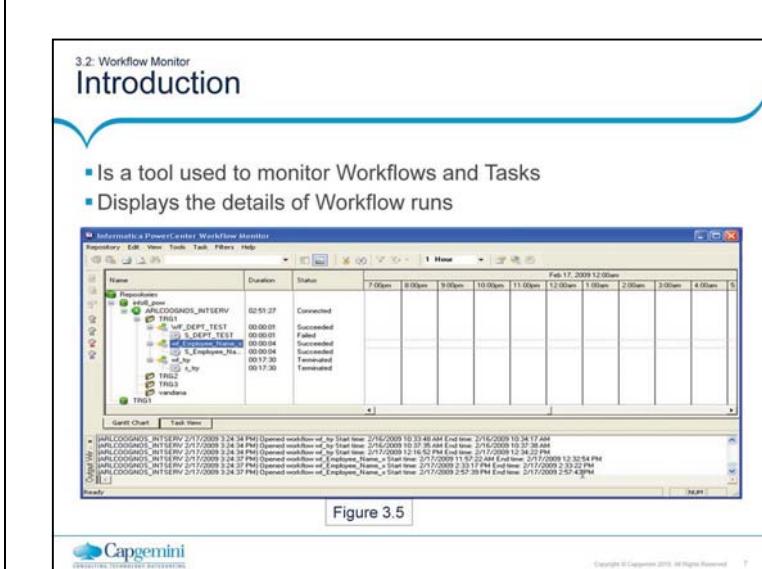
- Used to construct Session, Shell Command and Email Tasks
- Tasks created in Task Developer are reusable in Worklets or Workflows

#### Worklet Designer

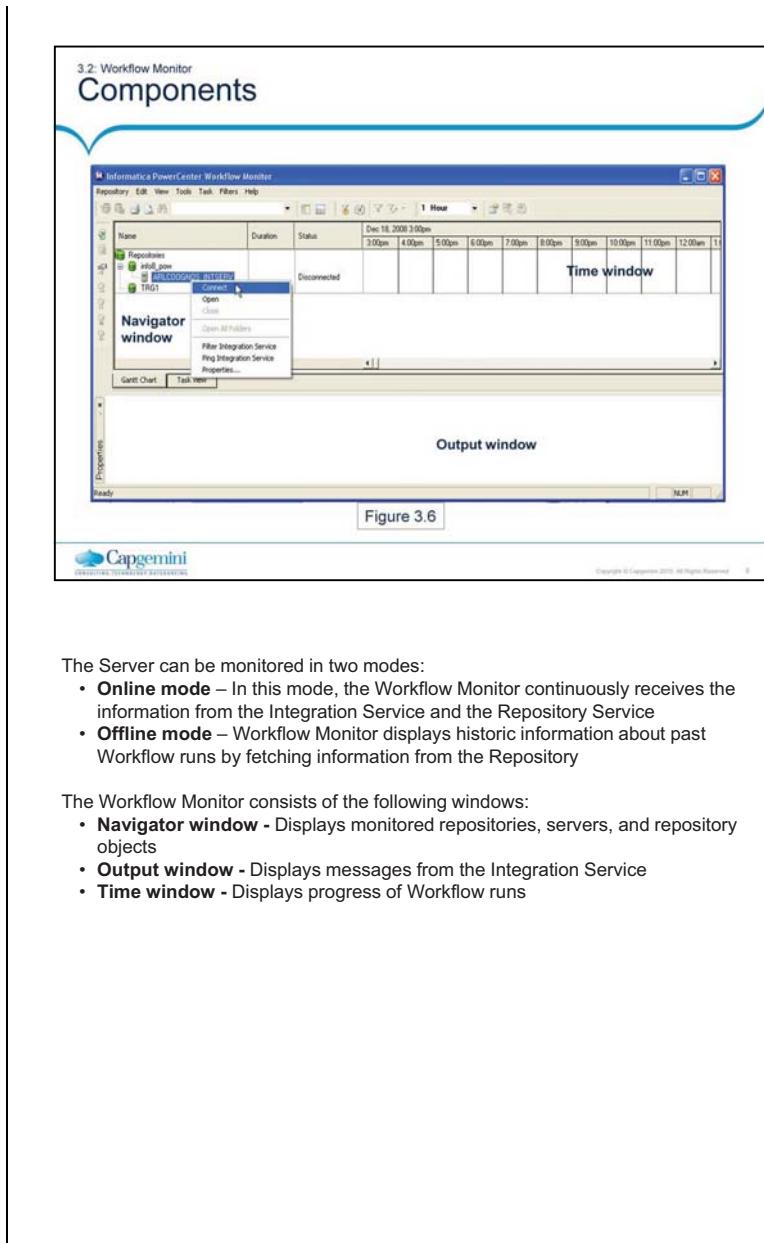
- Used to construct an object which represents a set of Tasks
- Worklets are reusable in multiple Workflows

#### Workflow Designer

- “Maps” the execution order of Sessions, Tasks and Worklets for the Integration Service



The *Workflow Monitor* is a tool used to monitor Workflows and Tasks. The details about a Workflow or a Task can be viewed in either Gantt Chart view or Task view. Workflows can be run, stopped, aborted, and resumed from the Workflow Monitor. The Workflow Monitor displays Workflows that have run at least once.



The Server can be monitored in two modes:

- **Online mode** – In this mode, the Workflow Monitor continuously receives the information from the Integration Service and the Repository Service
- **Offline mode** – Workflow Monitor displays historic information about past Workflow runs by fetching information from the Repository

The Workflow Monitor consists of the following windows:

- **Navigator window** - Displays monitored repositories, servers, and repository objects
- **Output window** - Displays messages from the Integration Service
- **Time window** - Displays progress of Workflow runs

3.2: Workflow Monitor

## Task View

- Displays details about Workflow runs in a report format
- The following details are displayed :
  - Workflow run
  - Task Name
  - Worker Server
  - Start time
  - Completion time
  - Status
  - Run Type
  - Task type
  - User information

Figure 3.7

Capgemini CONSULTING TECHNOLOGY INNOVATING

Copyright © Capgemini 2010. All Rights Reserved.

The Task view displays details about Workflow runs in a report format. It displays the following details:

- **Task name** - The name of the task will be displayed under the workflow name
- **Folder name** - The name of the folder containing the Task
- **Status** - The status of the Task or Workflow
- **Workflow name** - The name of the Workflow
- **Worklet name** - The name of the Worklet
- **Start time** - The time that the Integration Service starts executing the Task or Workflow
- **Completion time** - The time that the Integration service finishes executing the Task or Workflow
- **Status message** - Message from the Integration Service regarding the status of the Task or Workflow
- **User name** - Name of the user who owns the Workflow or Task
- **Run type** - The method used to start the Workflow. Workflow can be manually started or scheduled to start
- **Task type** - The type of the Task

3.2: Workflow Monitor

## Gantt Chart View

- Displays the details about Workflow runs in chronological format
- The following details are displayed:
  - Task name
  - Duration
  - Status
  - Connection between objects
  - Line 3.1.

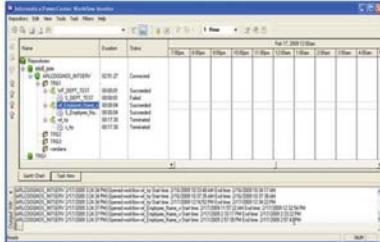


Figure 3.8

The Gantt Chart view displays details about Workflow runs in chronological format. It displays the following information:

- **Task name** - Name of the Task in the Workflow
- **Duration** - The length of time the Integration service spends executing the most recent Task or Workflow
- **Status** - The status of the most recent Task or Workflow
- **Connection between objects** - The Workflow Monitor shows links between objects in the Time window

3.3: Database Connections

## Introduction

- A database connection is configured in the Workflow Manager before the Server can access a source or target database in a session
- When a session that reads from or writes to a relational database is created or modified, only configured source and target databases can be selected
- Database connections are saved in the repository

 Capgemini  
CONSULTING TECHNOLOGY INTEGRATING

Copyright © Capgemini 2010. All Rights Reserved 11

Before running any Workflow, the first thing that needs to be done is to configure a database connection. A database connection specifies the source and target database connection information. When a session that reads from or writes to a relational database is created or modified, only configured source and target databases can be selected.

For every source and target database used in the session, a separate database connection has to be configured.

To create a database connection, one of the following privileges is required:

- Use Workflow Manager
- Super User

3.3: Database Connections

## Information

To create a connection, the following information should be available:

- Database name
- Database type
- Database username
- Password
- Connect string
- Database code page

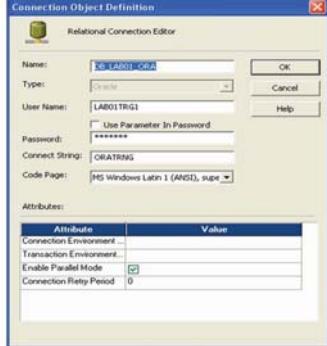
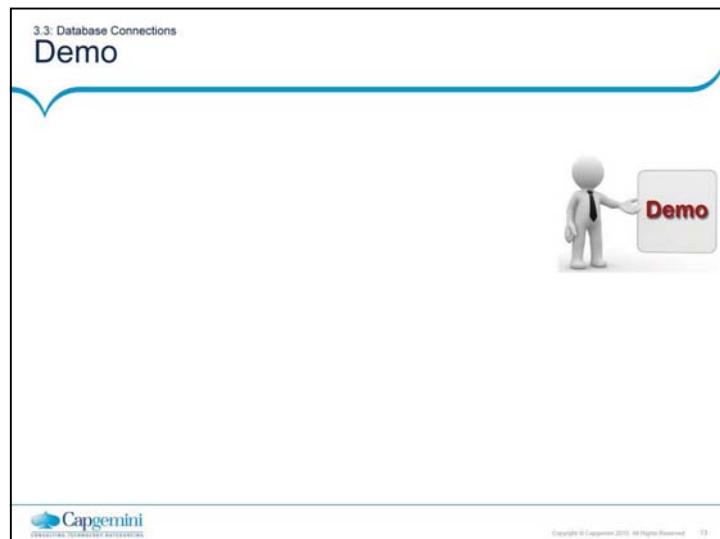


Figure 3.9

To create a connection, the following information should be available:

- **Database name** - Name for the connection
- **Database type** - Type of the source or target database
- **Database username** - Name of a user who has the appropriate database permissions to read from and write to the database
- **Password** - Database password
- **Connect string** - Connect string used to communicate with the repository
- **Database code page** - Code page associated with the database

**Steps to create a Relational Database Connection:**

1. In the Workflow Manager, connect to a repository.
2. Choose **Connections | Relational** menu option. A dialog box appears, listing all the registered source and target database connections.
3. Select the type of database connection to be created.
4. Click **Add**. The Connection Object Definition dialog box appears.
5. Enter the database connection name, database connection type, username, password, connect string and code page.
6. Click **OK**.

3.4: Tasks

## Introduction

- Tasks are used to build Workflows and Worklets
- Tasks can be created in:
  - Task Developer
  - Workflow Designer
  - Worklet Designer
- Tasks created in the Task Developer are reusable
- Tasks created in the Workflow Designer and Worklet Designer are non-reusable by default

 Capgemini  
CONSULTING TECHNOLOGY ENTERTAINMENT

Copyright © Capgemini 2011. All Rights Reserved 14

*Tasks* are building blocks of a Workflow. The Workflow Manager contains many types of Tasks to build Workflows and Worklets. Tasks created in the Task Developer are reusable and the ones created in the Workflow Designer and Worklet Designer are non-reusable.

The Workflow Manager validates Task attributes and links. If a Task is invalid, the Workflow becomes invalid. Workflows containing invalid sessions may still be valid.

3.4: Tasks

## Workflow Tasks

- Reusable Tasks

Task Name	Description
Session	Set of instructions to execute a mapping
Command	Specifies shell commands to run during the Workflow
Email	Sends email during the Workflow

 Capgemini  
CONSULTING TECHNOLOGY INTEGRATING

Copyright © Capgemini 2010. All Rights Reserved 18

Reusable Tasks can be used in multiple Workflows in the same folder. The reusable Tasks can be seen in the Tasks node in the Navigator window.

3.4: Tasks

## Workflow Tasks

- Non-reusable Tasks

Task Name	Description
Event-Raise	Used to trigger a user-defined event
Event-Wait	Waits for a user-defined or a pre-defined event to occur
Timer	Waits for a specified period of time to execute the next Task
Control	Stops or aborts the Workflow
Decision	Specifies a condition to evaluate in the Workflow. Use the Decision Task to create branches in a Workflow

 Capgemini  
CONSULTING TECHNOLOGY INTEGRATING

Copyright © Capgemini 2010. All Rights Reserved 16

Tasks created within the Workflow are non-reusable Tasks. The Non-reusable Tasks can be seen in the Sessions node in the Navigator window.

3.5: Session Task

## Introduction

- Created for each mapping to be run on the Integration Service
- Integration Service uses the instructions configured in the session to move data from sources to targets



S\_Employee\_Na...

Figure 3.10

 Capgemini CONSULTING TECHNOLOGY INNOVATING

Copyright © Capgemini 2010. All Rights Reserved 17

A **Session Task** is the basic object created in the Workflow Manager that assembles the information contained in a mapping together with database connection and other information. It is the vehicle for commanding the Integration Service to extract, transform, and load data and report session status and performance characteristics. A Workflow groups two or more sessions to run sequentially or concurrently or as specified in the Workflow.

A reusable Session Task is created in the Task Developer. We can also create Session Tasks in the Workflow Designer as we develop the Workflow. Once the Session is created, the session properties can be edited at any time.

3.5: Session Task

## Create Session Task

- Create a Task
- Select the type of Task and give a name to the Task
- Select the mapping to be associated with the Task

Figure 3.11

Figure 3.13

Capgemini CONSULTING TECHNOLOGY INNOVATING

To create a new Session Task, the following information is required:

- Mapping used for the Session Task
- Session name, which must be unique among all sessions in a given folder
- Source type
- Update strategy for writing to targets
- Target type

**Note:** Before creating a Session Task, the Workflow Manager has to be configured to communicate with databases and the Integration service. Appropriate permissions have to be assigned for any database, FTP, or external loader connections.

3.5: Session Task Properties

The Session Task properties has the following tabs

- General
- Properties
- Config Object
- Mapping
- Components

▪ Metadata Extension

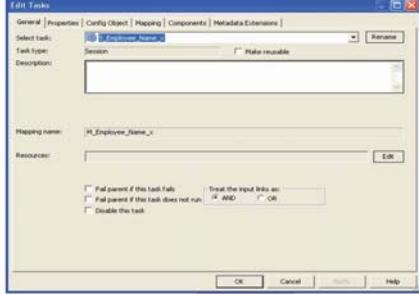


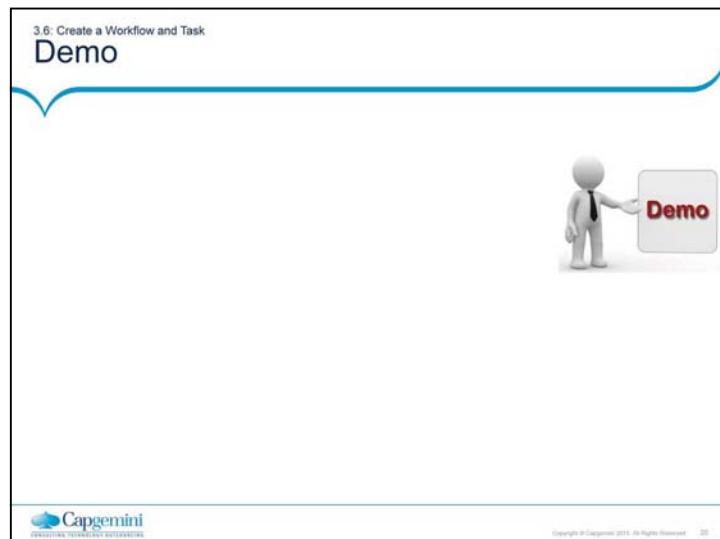
Figure 3.14

 Capgemini CONSULTING TECHNOLOGY INNOVATING

Copyright © Capgemini 2010. All Rights Reserved 18

The Session Task properties has the following tabs:

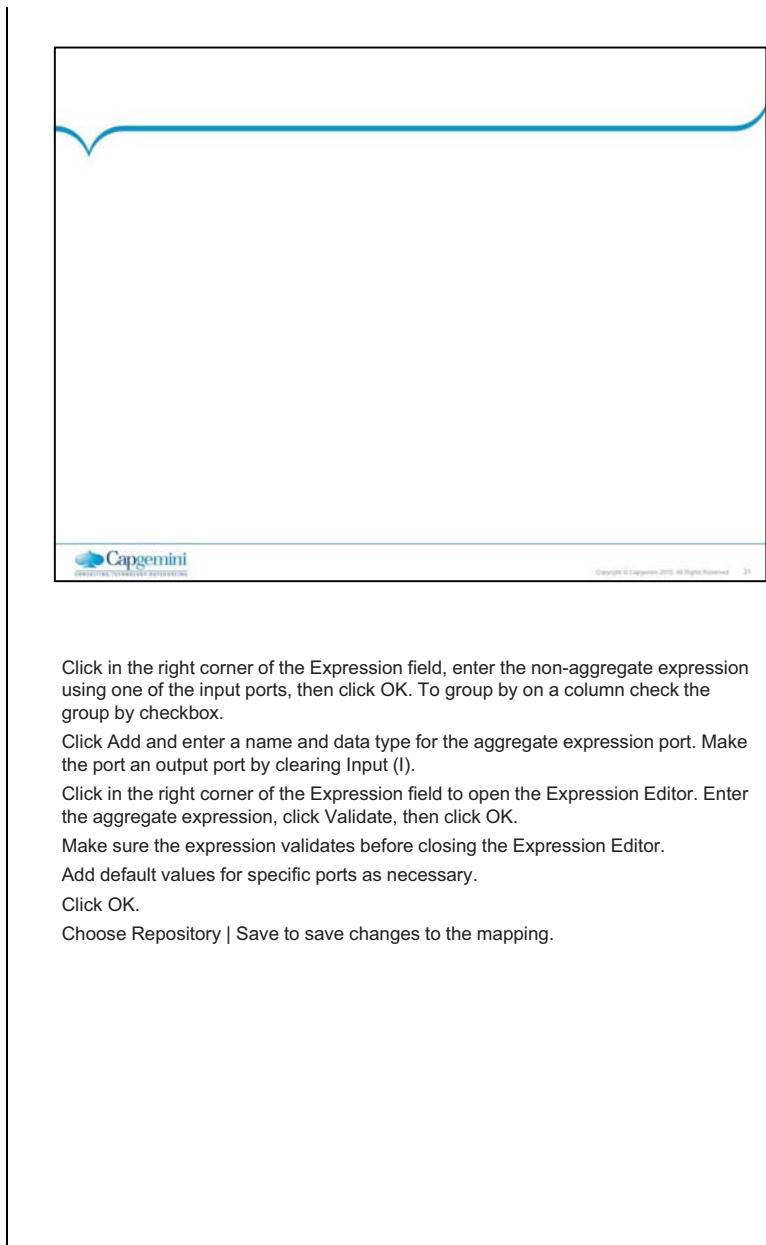
- **General** - Specifies session name, mapping name, and description for the Session Task
- **Properties** - Specifies Session log information, test load settings, and performance configuration
- **Config Object** - Specifies advanced settings, log options, and error handling configuration
- **Mapping** – Specifies source, target information and Override transformation properties
- **Components** - Configure pre- or post-session shell commands and emails
- **Metadata Extension** - Associate additional non-business information with repository objects

**To create a Workflow:**

1. Open the Workflow Designer.
2. Choose **Workflows** | **Create** menu option.
3. Enter a name for the new Workflow.
4. Click **OK**.

**To create a Task:**

1. In the Workflow Designer, click the **Session Task** icon on the Tasks toolbar.
2. Choose **Tasks** | **Create** menu option.
3. Select **Session** task for the Task type. Enter a name for the Session Task.
4. Click **Create**. The Mappings dialog box appears.
5. Select the mapping to be used in the Session Task and click **OK**.
6. Click **Done**. The Session Task appears in the workspace.



Copyright © Capgemini 2010. All Rights Reserved 21

Click in the right corner of the Expression field, enter the non-aggregate expression using one of the input ports, then click OK. To group by on a column check the group by checkbox.

Click Add and enter a name and data type for the aggregate expression port. Make the port an output port by clearing Input (I).

Click in the right corner of the Expression field to open the Expression Editor. Enter the aggregate expression, click Validate, then click OK.

Make sure the expression validates before closing the Expression Editor.

Add default values for specific ports as necessary.

Click OK.

Choose Repository | Save to save changes to the mapping.

3.7: Debugger

## Information

- The Debugger is used to debug a valid mapping to trace errors and gain more insight into the data
- The mapping can be debugged
  - Before a session is run
  - After a session is run
- The Debugger can run for the following session types:
  - Existing non-reusable session
  - Existing reusable
  - Debug session instance

 Capgemini  
CONSULTING TECHNOLOGY INNOVATING

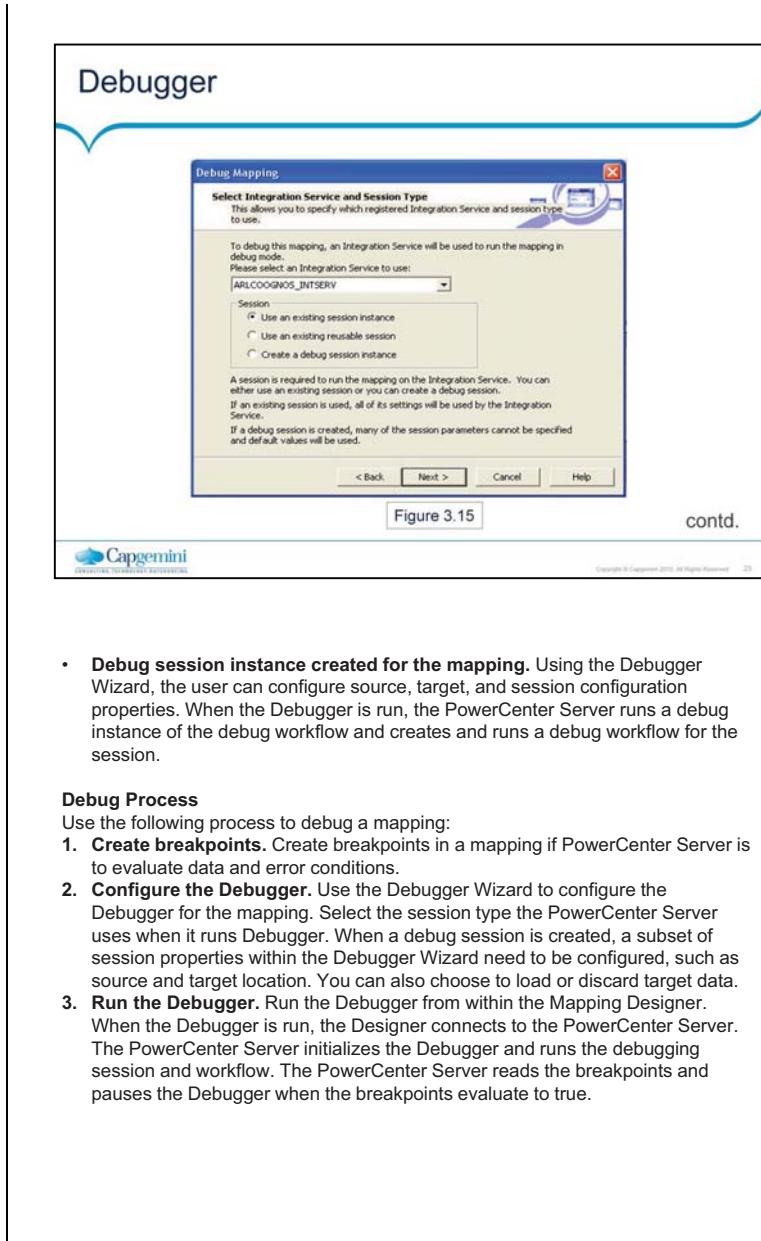
Copyright © Capgemini 2010. All Rights Reserved 32

You can debug a valid mapping to know about data and error conditions. To debug a mapping, the Debugger is configured and run from within the Mapping Designer.

When the Debugger is run, it pauses at breakpoints and allows viewing and editing of transformation output data.

The Debugger uses a session to run the mapping on the PowerCenter Server. When the Debugger is configured, three different debugger session types can be selected. The Debugger runs a workflow for each session type. The Debugger session types are :

- **Existing non-reusable session for the mapping.** The Debugger uses existing source, target, and session configuration properties. When the Debugger is run, the PowerCenter Server runs the non-reusable session and the existing workflow.
- **Existing reusable session for the mapping.** The Debugger uses existing source, target, and session configuration properties. When the Debugger is run, the PowerCenter Server runs a debug instance of the reusable session and creates and runs a debug workflow for the session.

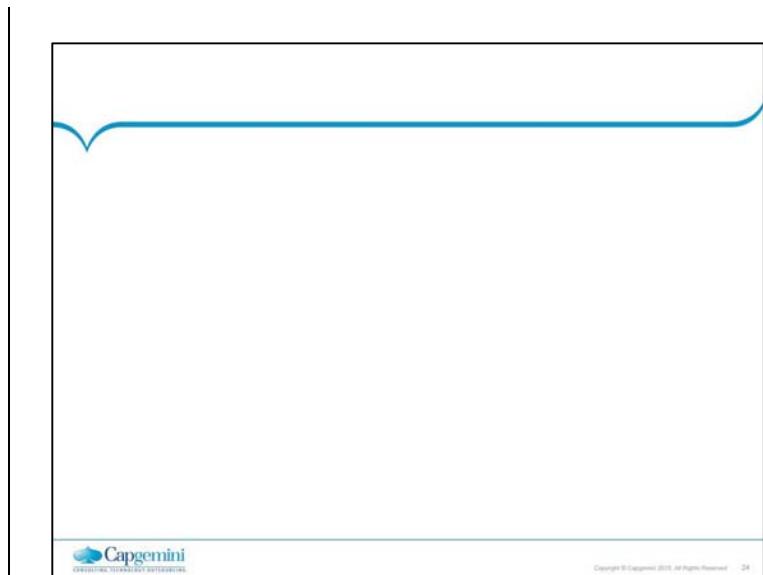


- **Debug session instance created for the mapping.** Using the Debugger Wizard, the user can configure source, target, and session configuration properties. When the Debugger is run, the PowerCenter Server runs a debug instance of the debug workflow and creates and runs a debug workflow for the session.

#### Debug Process

Use the following process to debug a mapping:

1. **Create breakpoints.** Create breakpoints in a mapping if PowerCenter Server is to evaluate data and error conditions.
2. **Configure the Debugger.** Use the Debugger Wizard to configure the Debugger for the mapping. Select the session type the PowerCenter Server uses when it runs Debugger. When a debug session is created, a subset of session properties within the Debugger Wizard need to be configured, such as source and target location. You can also choose to load or discard target data.
3. **Run the Debugger.** Run the Debugger from within the Mapping Designer. When the Debugger is run, the Designer connects to the PowerCenter Server. The PowerCenter Server initializes the Debugger and runs the debugging session and workflow. The PowerCenter Server reads the breakpoints and pauses the Debugger when the breakpoints evaluate to true.



Copyright © Capgemini 2010. All Rights Reserved. 30

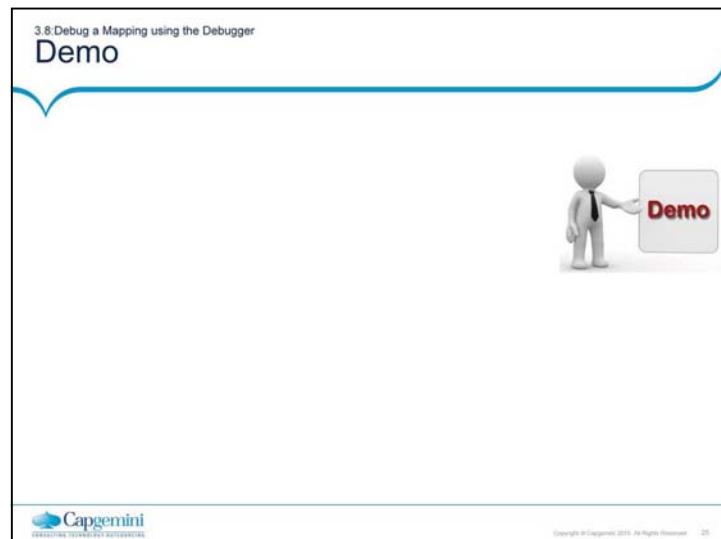
Monitor the Debugger. When the Debugger is running, target data, transformation and mapplet output data, the debug log, and the session log can be monitored. When the Debugger is running, the Designer displays the following windows:

Debug log. View messages from the Debugger.

Target window. View target data.

Instance window. View transformation data.

Modify data and breakpoints. When the Debugger pauses, data can be modified to see the effect on transformations, mapplets, and targets as the data moves through the pipeline. Breakpoint information can also be modified



### To use the Debugger

1. Select a valid mapping.
2. Select **Mappings – Debugger – Edit Breakpoints** to create breakpoints if any.
3. Select **Mappings – Debugger – Start Debugger** to invoke the Debugger.
4. Select the **server, session type, session instance and target table options** to configure the debugger.
5. Monitor the debugger. Observe transformation data in the instance window and target data in the target window.
6. Modify data if required.

## PowerCenter Integration Service Process

- Integration Service mainly consists of three process

- a. Integration Service Process
- b. Load Balancer
- c. Data Transformation Manager(DTM)



- Integration Service Process:

- Integration Service process is responsible to carry out below tasks:
- Workflow Schedule Management
- Locking Workflow Before Execution
- Reading Parameter File
- Creating Workflow Logs
- Running any Tasks in Workflow (e.g., Email)
- Creating DTM processes to run Workflow Sessions


Copyright © Capgemini 2010. All Rights Reserved 28

Whenever Integration Service receives a request to execute a workflow, it spawns an Integration Service Process. In admin community it is typically called as pmserver process.

Note that Integration Service can start one or multiple Integration Processes to run and monitor Workflows.

Whenever a workflow is scheduled or changed, Workflow Manager interacts with Integration Service Process. This interaction happens over TCP/IP. Integration Service process locks the Workflow and then knocks the Repository Service to get metadata about Workflow. Repository Service intern connects to Repository Database – fetches desired information – and gives it back to Integration Service process.

Once Integration Service process receives metadata information back from Repository Service, it creates connections to Source and Target databases using ODBC or Native drivers. Integration Service process also creates connections to any lookup or stored procedure databases which are used by the Workflow. Once the data loading process is done, Integration Service process releases and closes all open database connections.

Integration Service process is also responsible to start DTM process that will actually execute Workflow Sessions. Integration Service process passes Parameter File and Session information to DTM process, that helps it to retrieve required metadata from the repository. Please refer to article [Data Transformation Manager](#) for details on DTM Process.

## Load Balancer / Manager

- While running a Workflow, the PowerCenter Server uses the Load Manager process and the Data Transformation Manager Process (DTM) to run the workflow and carry out workflow tasks. When the PowerCenter Server runs a workflow, the Load Manager performs the following tasks:
  1. Locks the workflow and reads workflow properties.
  2. Reads the parameter file and expands workflow variables.
  3. Creates the workflow log file.
  4. Runs workflow tasks.
  5. Distributes sessions to worker servers.
  6. Starts the DTM to run sessions.
  7. Runs sessions from master servers.
  8. Sends post-session email if the DTM terminates abnormally.

Copyright © Capgemini 2010. All Rights Reserved. 27

## DTM Process

- When the PowerCenter Server runs a session, the DTM performs the following tasks:
  1. Fetches session and mapping metadata from the repository.
  2. Creates and expands session variables.
  3. Creates the session log file.
  4. Validates session code pages if data code page validation is enabled. Checks query conversions if data code page validation is disabled.
  5. Verifies connection object permissions.
  6. Runs pre-session shell commands.
  7. Runs pre-session stored procedures and SQL.
  8. Creates and runs mapping, reader, writer, and transformation threads to extract, transform, and load data.
  9. Runs post-session stored procedures and SQL.
  10. Runs post-session shell commands.
  11. Sends post-session email.



Copyright © Capgemini 2010. All Rights Reserved. 28

## Code Page And Data Movement Modes

- The data movement mode depends on whether Informatica Server should process single byte or multi-byte character data. This mode selection can affect the enforcement of code page relationships and code page validation in the Informatica Client and Server.
  - a) Unicode – IS allows 2 bytes for each character and uses additional byte for each non-ascii character (such as Japanese characters)
  - b) ASCII – IS holds all data in a single byte.
- The IS data movement mode can be changed in the Informatica Server configuration parameters. This comes into effect once you restart the Informatica Server.
- Code Page is used to identify characters that might be in different languages.
- Incorrect selection of Code Page can lead to Junk Data or rejection of data.
- ASCII, UTF-8, UTF-16, UTF-32 are few examples for encoding/ Code pages available in Informatica.



Copyright © Capgemini 2010. All Rights Reserved. 29

## Audit Trails Output Files

- While Informatica Load Manager and DTM tread together executes the code , it also creates multiple files for audit trails. These files help us in analysis the code for performance enhancement and data quality
- Informatica server log: Informatica server (on UNIX) creates a log for all status and error messages (default name: pm.server.log). It also creates an error log for error messages. These files will be created in Informatica home directory.
- Session log file: Informatica server creates session log file for each session. It writes information about session into log files such as initialization process, creation of sql commands for reader and writer threads, errors encountered and load summary. The amount of detail in session log file depends on the tracing level that you set.
- Session detail file: This file contains load statistics for each target in mapping. Session detail includes information such as table name, number of rows written or rejected. You can view this file by double clicking on the session in monitor window.



Copyright © Capgemini 2010. All Rights Reserved 33

## Audit Trails Output Files

- Performance detail file: This file contains information known as session performance details which helps you where performance can be improved. To generate this file select the performance detail option in the session property sheet.
- Reject file: This file contains the rows of data that the writer does not write to targets.
- Control file: Informatica server creates control file and a target file when you run a session that uses the external loader. The control file contains the information about the target flat file such as data format and loading instructions for the external loader.
- Post session email: Post session email allows you to automatically communicate information about a session run to designated recipients. You can create two different messages. One if the session completed successfully the other if the session fails.

## Audit Trails Output Files

- Indicator file: If you use the flat file as a target, you can configure the informatica server to create indicator file. For each target row, the indicator file contains a number to indicate whether the row was marked for insert, update, delete or reject.
- Output file: If session writes to a target file, the informatica server creates the target file based on file properties entered in the session property sheet.
- Cache files: When the informatica server creates memory cache it also creates cache files. For the following circumstances informatica server creates index and datacache files.
- Aggregator transformation
- Joiner transformation
- Rank transformation
- Lookup transformation



Copyright © Capgemini 2010. All Rights Reserved. 32

## Summary

- After completing this module you now know:
  - The Workflow Manager components
  - What is a Task
  - How to create Workflow
  - How to create a Task
  - Use the Debugger



## Review Question

- Question 1: The \_\_\_\_\_ is the connection between the server and the repository
- Question 2: Workflow monitor displays details about Workflow runs in two views: \_\_\_\_\_ and \_\_\_\_\_
- Question 3: \_\_\_\_\_ are reusable in multiple Workflows



## Review Question

- Question 4: Session Task is a reusable Task.
  - True/False
  
- Question 5: The Debugger can be used for an invalid mapping.
  - True/False



## Informatica PowerCenter

Lesson 4- Transformations

## Lesson Objectives

- In this lesson you will learn about:
  - Configuring the following transformations
    - Lookup
    - Filter
    - Joiner
    - Rank
  - Working with Flat files



4.1: Lookup Transformation

## Description

Used to look up data in a relational table, view, or synonym

- Passive Transformation
- Can be Connected/Unconnected
- Has a Lookup condition
- Can be used to:
  - Get a related value
  - Perform a calculation
  - Update slowly changing dimension tables

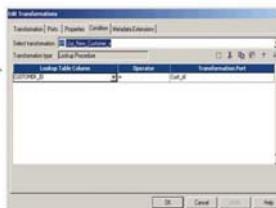


Figure 4.1

Copyright © Capgemini 2014. All Rights Reserved.

A Lookup Transformation is used to look up data in a relational table, view, or synonym. A lookup definition can be imported from any relational database to which both the Informatica Client and Server can connect. Multiple Lookup Transformations can be used in a mapping.

The Integration service queries the lookup table based on the lookup ports in the transformation. It compares Lookup Transformation port values to lookup table column values based on the lookup condition. The result of the lookup is passed to other transformations and the target.

The Lookup Transformation can be used to:

- **Get a related value** - For example, if the source table includes employee ID, but the target table requires employee name to make the summary data easier to read
- **Perform a calculation** - Many normalized tables include values used in a calculation, such as gross sales per invoice or sales tax, but not the calculated value (such as net sales)
- **Update slowly changing dimension tables** - Use a Lookup transformation to determine whether records already exist in the target

4.1: Lookup Transformation

## Description (Contd..)

Connected Lookup	Unconnected Lookup
Part of the mapping data flow	Separate from the mapping data flow
Returns multiple values (by linking output ports to another transformation)	Returns one value (by checking the Return (R) port option for the output port that provides the return value)
Executed for every record passing through the transformation	Only executed when the lookup function is called
More visible, shows where the lookup values are used	Less visible, as the lookup is called from an expression within another transformation
Default values are used	Default values are ignored

 Capgemini  
INFORMATION MANAGEMENT CONSULTING

Copyright © Capgemini 2014. All Rights Reserved.

The Lookup Transformation can be configured in two ways, **Connected** and **Unconnected**, to perform different types of lookups. These lookup transformations receive input and send output in different ways.

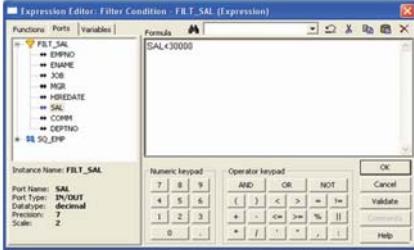
- **Connected** - A connected Lookup Transformation can be configured to receive input directly from the mapping pipeline
- **Unconnected** - An unconnected Lookup Transformation can be configured to receive input from the result of an expression in another transformation

A lookup condition has to be specified in the Condition tab. One input port is needed for each column used in the lookup condition. The same input port can be used more than once in a condition, and also multiple conditions can be specified.

4.1: Lookup Transformation  
Description (Contd..)

To configure a Lookup Transformation the following components are defined:

- Lookup table
- Ports
- Properties
- Condition



The screenshot shows the 'Expression Editor, Filter Condition - FILT\_SAL (Expression)' window. It displays a tree view of functions under 'Functions' and a formula entry field containing 'SAL<30000'. Below the formula field are numeric and operator keypads, and standard dialog buttons like OK, Cancel, Validate, and Command.

Figure 4.2

 Capgemini  
INFORMATION & INDUSTRIES

Copyright © Capgemini 2011. All Rights Reserved.

To configure a Lookup Transformation the following components are defined:

- **Lookup source** - Lookup source can be created from the following locations:
  - Relational or flat file source or target definition in repository
  - Source Qualifier definition in Mapping
  - Table or file that the Integration service and client can connect to
- **Lookup Ports** - The Ports tab contains options similar to other transformations, such as port name, data type and scale. In addition to input and output ports, the Lookup Transformation includes a lookup port type that represents columns of data in the lookup table. An Unconnected Lookup Transformation also includes a return port type that represents the return value
- **Lookup Properties** - The Properties tab is used to configure properties such as an SQL override for the lookup, the lookup table name, and tracing level for the transformation. Most of the options on this tab allow to configure caching properties
- **Lookup Condition** - The condition or conditions that the Integration service has to use is specified in the conditions tab

4.1 Lookup Transformation

## Connected Lookup Transformation

- Input values directly from another transformation in the pipeline
- For each input row, the Integration Service queries the lookup source or cache based on the lookup ports and the condition in the transformation
- If the transformation is uncached or uses a static cache, the Integration Service returns values from the lookup query
- If the transformation uses a dynamic cache
  - It inserts the row into the cache when the row is not found
  - It updates the row in the cache or leaves it unchanged
  - It flags the row as insert, update, or no change
- The Integration Service passes return values from the query to the next transformation. If the transformation uses a dynamic cache, you can pass rows to a Filter or Router transformation to filter new rows to the target.

 Capgemini  
INTEGRATION CONSULTING SERVICES

Copyright © Capgemini 2011. All Rights Reserved.

The following steps describe how the PowerCenter Server processes a connected Lookup transformation:

1. A Connected Lookup transformation receives input values directly from another transformation in the pipeline.
2. For each input row, the PowerCenter Server queries the lookup source or cache based on the lookup ports and the condition in the transformation.
3. If the transformation is uncached or uses a static cache, the PowerCenter Server returns values from the lookup query.  
If the transformation uses a dynamic cache, the PowerCenter Server inserts the row into the cache when it does not find the row in the cache. When the PowerCenter Server finds the row in the cache, it updates the row in the cache or leaves it unchanged. It flags the row as insert, update, or no change.
4. The PowerCenter Server passes return values from the query to the next transformation. If the transformation uses a dynamic cache, you can pass rows to a Filter or Router transformation to filter new rows to the target.

4.1: Lookup Transformation

## Unconnected Lookup Transformation

- The following steps describe the way the PowerCenter Server processes an unconnected Lookup transformation:
  - An unconnected Lookup transformation receives input values from the result of a :LKP expression in another transformation, such as an Update Strategy transformation
  - The Integration Service queries the lookup source or cache based on the lookup ports and condition in the transformation
  - The Integration Service returns one value into the return port of the Lookup transformation
  - The Lookup transformation passes the return value into the :LKP expression

 Capgemini  
INTEGRATION CONSULTING SERVICES

Copyright © Capgemini 2011. All Rights Reserved.

An unconnected Lookup transformation receives input values from the result of a :LKP expression in another transformation. You can call the Lookup transformation more than once in a mapping.  
A common use for unconnected Lookup transformations is to update slowly changing dimension tables.

## Lookup Caches

- When configuring a lookup cache, you can specify any of the following options:
  - Static cache:
  - Dynamic cache:
  - Persistent cache:
  - Shared cache:

Copyright © Capgemini 2010. All rights reserved.

## Non-Persistent Cache

- By default, the Integration Service uses a non-persistent cache when you enable caching in a Lookup transformation.
- The Integration Service deletes the cache files at the end of a session.
- The next time you run the session, the Integration Service builds the memory cache from the database



Copyright © Capgemini 2014. All Rights Reserved.

Lookups are cached by default in Informatica. Informatica, by default brings in the entire data of the lookup table from database server to Informatica Server as a part of lookup cache building activity during session run.

## Persistent Cache

If you want to save and reuse the cache files, the transformation can be configured to use a persistent cache.

- Persistent cache is used when the lookup table does not change between session runs.
- The first time the Integration Service runs a session using a persistent lookup cache, it saves the cache files to disk instead of deleting them.
- The next time the Integration Service runs the session, it builds the memory cache from the cache files.
- If the lookup table changes occasionally, you can override session properties to recache the lookup from the database.

Edit Transformations	
Transformation Attribute	Value
Lookup Id Override	
Lookup Cache	DEPT
Lookup Source Filter	<input checked="" type="checkbox"/> Any Value
Lookup caching enabled	<input checked="" type="checkbox"/>
Lookup policy on multiple match	<input type="radio"/> All
Connection Information	DEPTINFO > DEPTNO_IN
Source Type	Database
Transient	<input type="checkbox"/>
Lookup cache directory name	BPMCacheDir
Lookup cache persistent	<input checked="" type="checkbox"/>
Lookup cache size	Auto
Lookup Index Cache Size	<input type="radio"/>
Dynamic Lookup Cache	<input type="checkbox"/>
Temporary Dynamic Cache	<input type="checkbox"/>
Drop Old Values On Update	<input type="checkbox"/>
Update Dynamic Cache Condition	TRUE
Cache File Name Prefix	
Replace Existing Backup Source	<input type="checkbox"/>
Insert Else Update	<input checked="" type="checkbox"/>
Update Else Insert	<input type="checkbox"/>
Drop Old Values	<input type="checkbox"/>
Thousands Separator	,
Decimal Separator	.
Case Sensitive String Comparison	<input checked="" type="checkbox"/>
Null Value Comparison	Equal To Highest Value
<b>Lookup Cache persistent</b>	Indicates if a cache is persistent or non-persistent

 Copyright © Capgemini 2010. All Rights Reserved.

If the lookup table is too huge, the process of bringing data from database server to Informatica server will take time. It might be possible that you are looking up to the same table different times using different lookups in different mappings. It would take a long time building the lookup cache again and again for each lookup. In such scenario persistent cache can be used.

## Other Types Of Cache

- Static cache
  - Static Cache is same as a Cached Lookup in which once a Cache is created the Integration Service always queries the Cache instead of the Lookup Table.
  - In Static Cache when the Lookup condition is true it return value from lookup table else returns Null or Default value. In Static Cache the important thing is that you cannot insert or update the cache.
- Dynamic cache
  - In Dynamic Cache we can insert or update rows in the cache when we pass the rows. The Integration Service dynamically inserts or updates data in the lookup cache and passes the data to the target. The dynamic cache is synchronized with the target
- Shared cache
  - When we use shared Cache Informatica server creates the cache memory for multiple lookup transformations in the mapping and once the lookup is done for the first lookup then memory is released and use that memory used by the other look up transformation.
  - We can share the lookup cache between multiple transformations.
    - Unnamed cache is shared between transformations in the same mapping and
    - Named cache between transformations in the same or different mappings.



Copyright © Capgemini 2014. All Rights Reserved. 11

If the lookup table is too huge, the process of bringing data from database server to Informatica server will take time. It might be possible that you are looking up to the same table different times using different lookups in different mappings. It would take a long time building the lookup cache again and again for each lookup. In such scenario persistent cache can be used.

## Key Points to Remember

- Where applicable, try to use small tables as Lookup table and minimize the cache – We should consider using the bigger table as a Source
  - If feasible, try to replace lookup transformation with Joiner transformation.
- Non-equi join is possible using Lookup transformation.
- If a Lookup transformation specifies several conditions, you can improve lookup performance by placing all the conditions that use the equality operator first in the list of conditions that appear under the Condition tab.
- When Looking upon a table, uncheck Lookup port(s) for unwanted columns to minimize the cache – never delete those ports from the transformation.

Copyright © Capgemini 2018. All Rights Reserved.

Demo

- Creating a Lookup Transformation

A 3D character wearing a white shirt and black tie is holding a rectangular sign with the word "Demo" written in red capital letters.

Capgemini

Copyright © Capgemini 2014. All Rights Reserved.

**Steps to create a Lookup transformation:**

1. In the Mapping Designer, choose **Transformation | Create** menu option. Select the Lookup Transformation. Enter a name for the lookup. The naming convention for Lookup Transformations is LKP\_TransformationName. Click **OK**.
2. In the *Select Lookup Table* dialog box choose the lookup table. Click the **Import** button if the lookup table is not in the source or target database.
3. To manually define the Lookup Transformation, click the **Skip** button.
4. Define input ports for each Lookup condition.
5. For an Unconnected Lookup Transformation, create a return port for the value you want to return from the lookup.
6. Define output ports for the values you want to pass to another transformation.
7. Add the lookup conditions.
8. On the **Properties** tab, set the properties for the lookup.
9. Click **OK**.
10. For Unconnected Lookup Transformations, write an expression in another transformation using :LKP to call the Unconnected Lookup Transformation.

**4.2. Filter Transformation**

## Description

- Used to filter rows in a Mapping.
- Active transformation
- Specify a filter condition which returns TRUE or FALSE
- Records that return TRUE are allowed to pass
- Filtered records are lost

Figure 4.3

The Filter Transformation limits the rows sent to the target or other transformations. All the rows from a source transformation can be passed through the Filter Transformation based on a filter condition. All ports in a Filter Transformation are input/output, and only rows that meet the condition pass through the Filter Transformation. In some cases, you need to filter data based on one or more conditions before writing it to targets. For example, if you have a human resources target containing information about current employees, you might want to filter out employees who have resigned.

## Filter Condition

▪ Filter Condition is entered in the Expression Editor of the Filter transformation

Figure 4.4

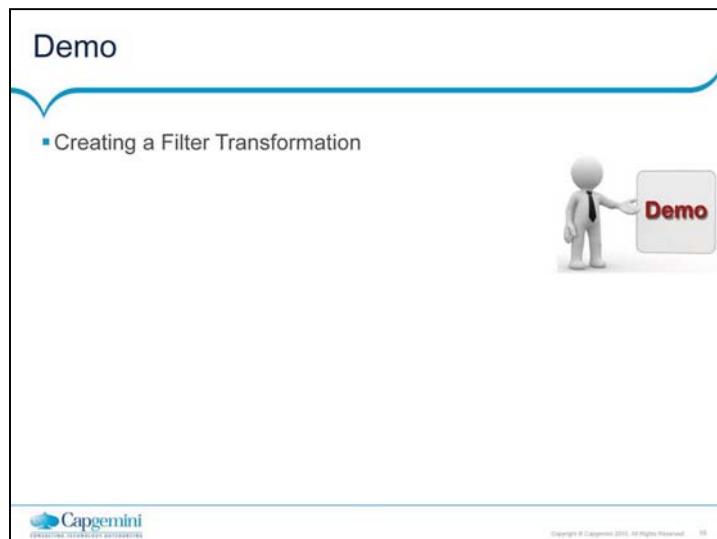
▪ The transformation language is used to enter the filter condition expression

The Filter Condition can be entered in the expression editor. The expression editor is invoked by clicking on down arrow next to Filter Condition option under the Properties tab of Edit Transformation window.

For example, if you want to filter out rows for employees whose salary is less than \$30,000, you enter the following condition:

SALARY < 30000

Multiple conditions can be defined using the AND and OR logical operators. TRUE and FALSE values are implicit return values from any condition and hence need not be specified in the condition expression. If the filter condition evaluates to NULL, the row is assumed to be FALSE.

**Steps to create a Filter transformation:**

1. In the Designer, switch to the Mapping Designer and open a mapping.
2. Choose **Transformation** | **Create** menu option. Select Filter Transformation, and enter the name of the new transformation. Click **Create**, and then click **Done**.
3. Select and drag all the desired ports from a source qualifier or other transformation to add them to the Filter Transformation. After the ports are selected and dragged, copies of these ports appear in the Filter Transformation.
4. Double-click the title bar of the new transformation.
5. Click the **Properties** tab. A default condition appears. The default condition is TRUE (constant with a numeric value of 1).
6. Click the **Value** section of the condition, and then click the **Open** button. The Expression Editor appears.
7. Enter the filter condition to be applied. Use values from one of the input ports in the transformation as part of this condition. However, values from output ports in other transformations can also be used.
8. Click **Validate** to check the syntax of the conditions entered. Fix syntax errors before continuing.
9. Click **OK**.
10. Choose **Repository** | **Save** menu option to save the mapping.

4.3. Joiner Transformation

## Description

Used to join two related heterogeneous sources

- Active transformation
- Heterogeneous
  - Relational tables in separate databases
  - Files in separate file systems
  - A relational and a flat file
- Settings for joiner
  - Master and detail source
  - Type of join
  - Condition of

Figure 4.5

While a Source Qualifier Transformation can join data originating from a common source database, the Joiner transformation joins two related heterogeneous sources residing in different locations or file systems.

The combination of sources can be varied. The following sources can be used:

- Two relational tables existing in separate databases
- Two flat files in potentially different file systems
- Two different ODBC sources
- Two instances of the same XML source
- A relational table and a flat file source
- A relational table and an XML source

The Joiner Transformation is used to join two sources with at least one matching port. The Joiner Transformation uses a condition that matches one or more pairs of ports between the two sources. For example, you can join a flat file with in-house customer IDs and a relational database table that contains user-defined customer IDs. If two relational sources contain keys, then a Source Qualifier Transformation can easily join the sources on those keys. Joiner Transformations typically combine information from two different sources that do not have matching keys, such as flat file sources.

One of the sources is specified as the master source, and the other as the detail source. This is specified in the Ports tab of the transformation by clicking the **M** (Master/Detail checkbox) column. When ports of a transformation are added to a Joiner transformation, the ports from the first source are automatically set as detail sources. Adding the ports from the second transformation automatically sets them as master sources. The **master/detail relation** determines how the join treats data from those sources based on the type of join.

The Joiner Transformation supports the following **join types**, which are set in the Properties tab:

- Normal (Default)
- Master Outer
- Detail Outer
- Full Outer

The **condition** of the join is a mandatory condition defining at least one field from each data source that the transformation uses to perform the join. These fields must be declared to be of the same data type. For example, the following condition joins data from two sources based on an item ID:

ITEM\_NO = ITEM\_NO1

**Note:** A Sequence Generator or Update Strategy transformation cannot be used as a source of a Joiner Transformation.

**Steps to create a Joiner Transformation:**

1. In the Mapping Designer, choose **Transformation | Create** menu option. Select the **Joiner Transformation**. Enter a name, click **OK**. The naming convention for Joiner transformations is JNR\_TransformationName.
2. Enter a description for the transformation. This description appears in the Repository Manager, making it easier for all to understand or remember what the transformation does.
3. Drag all the desired input/output ports from the first source into the Joiner Transformation. The Designer creates input/output ports for the source fields in the Joiner, as detail port by default. This property can be edited later.
4. Select and drag all the desired input/output ports from the second source into the Joiner Transformation. The Designer configures the second set of source fields and master fields by default.
5. Double-click the title bar of the Joiner Transformation to open the Edit Transformations dialog box.
6. Select the **Ports** tab.

7. Click any box in the **M** column to switch the master/detail relationship for the sources. Change the master/detail relationship if necessary by selecting the master source in the M column.  
**Note:** Designating the source with fewer unique records as master increases performance during a join.
8. Add default values for specific ports as necessary. Certain ports are likely to contain NULL values, since the fields in one of the sources may be empty. A default value can be specified if the target database does not handle NULL values.
9. Select the **Condition** tab and set the condition.
10. Click the **Add** button to add a condition. Multiple conditions can be added. The master and detail ports must have matching data types. The Joiner Transformation only supports equivalent (=) joins:
11. Select the Properties tab and enter any additional settings for the transformations.
12. Click **OK**.
13. Choose **Repository | Save** menu option to save changes done to the mapping.

4.4. Rank Transformation

## Description

- Used to select only the top or bottom rank of data.
- Active transformation
- Generates the Rank Index
- Grouping can be done

Figure 4.6

The Rank Transformation is used to select only the top or bottom rank of data. It can be used to return the largest or smallest numeric value in a port or group.

The Rank Transformation differs from the transformation functions MAX and MIN, in that it allows to select a group of top or bottom values, not just one value. For example, use Rank to select the top 10 salespersons in a given territory. Or, to generate a financial report, can also be used to identify the three departments with the lowest expenses in salaries and overhead. While the SQL language provides many functions designed to handle groups of data, identifying top or bottom strata within a set of rows is not possible using standard SQL functions.

All ports representing the same row set can be connected to the transformation. Only the rows that fall within that rank, based on some measure set during configuration, pass through the Rank Transformation. Expression to transform data or perform calculations can also be written.

The screenshot shows a slide titled "Demo" with a sub-section titled "Creating a Rank Transformation". At the bottom right, there is a small 3D character holding a rectangular sign that also says "Demo". The slide footer includes the Capgemini logo and copyright information.

**Steps to create a Rank Transformation:**

1. In the Mapping Designer, choose **Transformation | Create** menu option. Select the Rank Transformation. Enter a name for the Rank. The naming convention for Rank Transformations is RNK\_TransformationName.
2. Enter a description for the transformation.
3. Click **OK**, and then click **Done**. The Designer creates the Rank Transformation.
4. Link columns from an input transformation to the Rank Transformation.
5. Click the **Ports** tab, and then select the Rank (**R**) option for the port used to measure ranks.
6. To create groups for ranked rows, select **Group By** for the port that defines the group.
7. Click the **Properties** tab and select either the top or bottom rank. For the Number of Ranks option, enter the number of rows to select for the rank.
8. Click **OK** to return to the Designer.
9. Choose **Repository | Save** menu option.

4.5 Working with Flat Files

## Description

- Flat files can be used as sources and targets in a mapping.
- Flat file source definitions can be imported or created in the Source Analyzer.
- Flat file target definitions can be imported or created in the Target designer.
- Fixed-width and delimited flat file source and target definitions can be imported.
- Flat File Wizard is used to import flat files.

Sources Window Help

Import from Database...  
Import from File...

Figure 4.7

Capgemini

Copyright © Capgemini 2011. All Rights Reserved.

Flat Files can be used as sources and targets in a mapping. To do so, flat file sources and targets must be defined in the repository. Flat file source definitions can be imported or created in the Source Analyzer. Flat file target definitions can be imported or created in the Target designer.

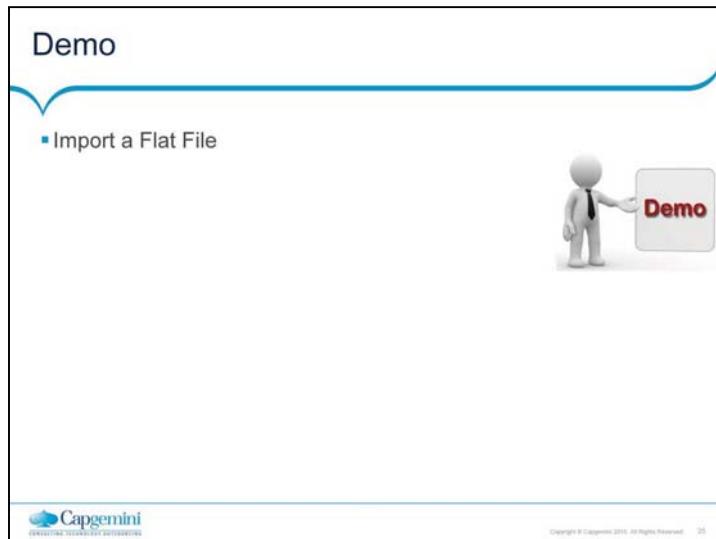
After the file source and target definitions are created, they can be used in mappings.

**Note:** Because source definitions must exactly match the source, Informatica recommends importing file source definitions instead of creating them manually.

Fixed-width and delimited flat file source and target definitions can be imported that do not contain binary data. When importing the definition, the file must be in a directory local to the client machine. In addition, the Integration service must be able to access all source files during the Workflow. When a file source or target definition is created, the properties of the file must be defined.

The Source Analyzer and the Target designer provides a Flat File Wizard, which prompts for the following file properties:

- File name and location
- File code page
- File type
- Column names and data types
- Number of header rows in the file
- Column size and null characters for fixed-width files
- Delimiter type, quote character, and escape character for delimited files

**Steps to import a Flat file definition:**

1. Open the Source Analyzer and choose **File | Sources | Import** menu option. -or- Open the Target designer and choose **File | Targets | Import** menu option. The Open Flat File dialog box appears.
2. Click **OK**. The contents of the file appear in the window at the bottom of the Flat File Wizard.
3. Edit the following settings as necessary:
  - Flat File Type
  - A name for this source or target
  - Start Import At Row
  - Import Field Names From First Line
4. Click **Next**.
5. Follow the instructions given in the wizard to manipulate the column breaks in the file preview window. Move existing column breaks by dragging them. Double-click a column break to delete it. In case of a de-limited flat file, column breaks are automatically created based on the delimiter.
6. Click **Next**.
7. Enter column information for each column in the file.
9. Click **Finish**.
10. Choose **Repository | Save** menu option to save.

## Summary

- After completing this lesson you now:
  - Know how to configure the following transformations
    - Lookup
    - Filter
    - Joiner
    - Rank
  - Know how to work with flat files as source or target definitions



## Review Question

- Question 1: \_\_\_\_\_ lookup transformation is part of the mapping data flow.
- Question 2: To rank values \_\_\_\_\_ is created in the rank transformation.
- Question 3: A Joiner transformation can only be used for relational sources.
  - True/False
- Question 4: Flat files can be used as sources and targets in a mapping.
  - True/False



## **Informatica PowerCenter**

Lesson 5: Mapplets

## Lesson Objectives

- In this lesson you will learn about:
  - Concept of Mapplet
  - Mapplet Input and Output Transformations
  - Configuring a Normalizer Transformation



5.1. Concept of Maplet  
**Description**

- A maplet is a reusable object that represents a set of transformations
- It allows to reuse transformation logic and can contain as many transformations as needed

*Maplet Designer*

The screenshot shows the Informatica Maplet Designer interface. It consists of four main windows:

- Source Qualifier:** A table with columns: Name, Datatype. Data includes: ITEM\_ID (decimal), ITEM\_NAME (string), ITEM\_DESC (string), PRIME\_COST (decimal), WHOLESALE (decimal), DISCONTINU (decimal), QUANTITY (decimal), PRICE (decimal), and DISTRIBUTOR (decimal). A note indicates: 'Name: NAME, Description: NAME'.
- Aggregator:** A table with columns: Name, Expressions. Data includes: ITEM\_ID (ITEM\_ID), ITEM\_NAME (ITEM\_NAME), DATE\_ENTERED (YEAR), QUANTITY (TO\_CHAR(QUANTITY)), PRICE (PRICE), DISCOUNT (DISCOUNT), Q1Sales (Q1Sales), and SUMQTY (SUM(QTY)). A note indicates: 'Name: Sales, Description: Sales'.
- Output Transform 1:** A table with columns: Name. Data includes: ITEM\_ID, ITEM\_NAME, YEAR, Q1Sales, SUMQTY, and Q4Sales.
- Output Transform 2:** A table with columns: Name. Data includes: ITEM\_ID, ITEM\_NAME, YEAR, Q1Sales, SUMQTY, and Q4Sales.

Figure 5.1

Copyright © Capgemini 2010. All Rights Reserved. 3

A mapplet is a reusable object that is created in the Maplet Designer. It contains a set of transformations. It allows to reuse the transformation logic in multiple mappings.

5.1. Concept of Mapplet

## What can Mapplets do?

- Mapplets help simplify mappings in the following ways:
  - Include source definitions
  - Accept data from sources in a mapping
  - Include multiple transformations
  - Pass data to multiple transformations
  - Contain unused ports

 Capgemini  
CONSULTING TECHNOLOGY SERVICES

Copyright © Capgemini 2013. All Rights Reserved. 4

Mapplets help simplify mappings in the following ways:

- **Include source definitions** - Multiple source definitions and source qualifiers can be used to provide source data for a mapping
- **Accept data from sources in a mapping** - The mapplet can receive source data from the mapping by using an Input transformation to receive source data
- **Include multiple transformations** - A mapplet can contain as many transformations as needed
- **Pass data to multiple transformations** - A mapplet can feed data to multiple transformations. Each Output transformation in a mapplet represents one output group in a mapplet
- **Contain unused ports** - Like a reusable transformation, a mapplet can have input and output ports that are not used in a mapping. This helps to design a mapplet for a range of uses

5.2. Mapplet Input and Output

## Description

- To use a mapplet in a mapping it has to be configured for input and output
- In addition to transformation logic a mapplet has the following components:
  - Mapplet input
  - Mapplet output
  - Mapplet ports
- Targets cannot be placed in a mapplet
- It has two additional transformations
  - Mapplet Input (Optional)
  - Mapplet Output

 Capgemini  
PERFORMING TECHNOLOGY FOR BUSINESS

Copyright © Capgemini 2010. All Rights Reserved.

In addition to the transformation logic, a mapplet has the following components.

- **Mapplet input** - Data can be passed into a mapplet using source definitions and/or Input transformations. An Input transformation is used to connect a mapplet to the source pipeline in the mapping
- **Mapplet output** - Each mapplet must contain one or more output transformations to pass data from the mapplet into the mapping
- **Mapplet ports** - Mapplet ports display only in the Mapping Designer. Mapplet ports consist of input ports from Input transformations and output ports from Output transformations. Other ports within the mapplet are referred to as transformation ports. If a mapplet uses source definitions rather than Input transformations for input, it does not contain any input ports in the mapping

5.2. Mapplet Input and Output

## Mapplet Input Transformation

- A Mapplet Input Transformation is used when input has to be received from a source in a mapping.(Source definitions are in Mapping)
- It provides input ports to pass data through the mapplet.

Figure 5.2  
Maplet Input Transformation

Capgemini  
CONSULTING TECHNOLOGY SERVICES

Copyright © Capgemini 2010. All Rights Reserved.

An Input Transformation is used to receive input from a source in a mapping. When the mapplet is used in a mapping, the Input transformation provides input ports so data can be passed through the mapplet. Each port in the Input transformation connected to another transformation in the mapplet becomes a mapplet input port. Input transformations can receive data from a single active source.

**5.2. Mapplet Input and Output**

## Mapplet Output Transformation

- A Mapplet Output Transformation is used to pass data through the mapplet into a mapping
- A mapplet must contain at least one Output transformation with at least one connected port in the mapplet

The screenshot shows the Mapplet Designer interface. On the left, there's a 'Maplet Explorer' tree view. In the center, a flow starts with an 'SQL Source Qualifier' (labeled 'ITEMS') connected to an 'Aggregator' (labeled 'Agg\_SellPrice'). The aggregator has several output ports, one of which is connected to a 'Data Select Output Trans.' (labeled 'DSOT'). A callout box labeled 'Maplet Output Transformation' points to the DSOT component. Below the interface, the text 'Figure 5.3' is centered.

Figure 5.3

Copyright © Capgemini 2010. All Rights Reserved

Every mapplet ends with a Mapplet Output Transformation. The data from the mapplet passes to the mapping through the Mapplet Output Transformation. A mapplet must contain at least one Output transformation with at least one connected port in the mapplet. Each connected port in an Output transformation displays as a mapplet output port in a mapping. Each Output transformation in a mapplet displays as an output group in a mapping. An output group can pass data to multiple pipelines in a mapping.

A mapplet can be expanded in the Mapping Designer by selecting it and choosing **Mappings-Expand** from the menu. This expands the mapplet within the mapping for view.

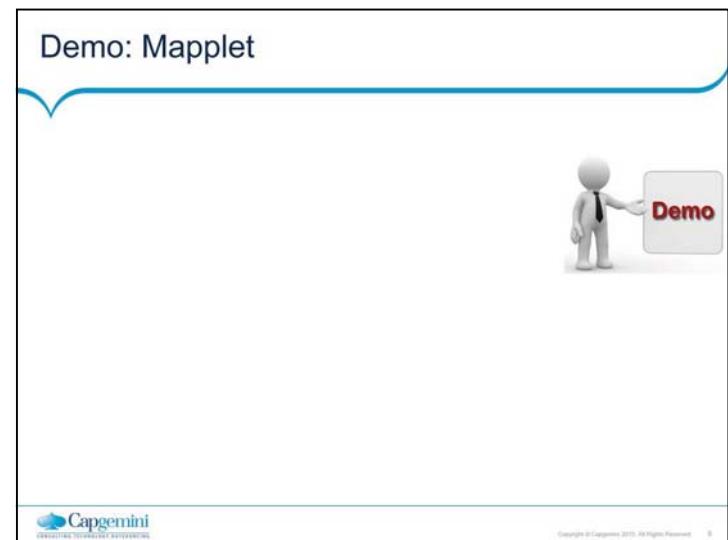
5.2. Mapplet Input and Output

## Creating the Data Profile

- After you create the mapplet to filter and aggregate the data, you can profile the mapplet output data. From the Profile Manager, create a custom profile.



Copyright © Capgemini 2015. All Rights Reserved.

**Steps to create and configure a mapplet:**

1. Create a mapplet. Choose **Mapplets | Create** from the menu in the Mapplet Designer. The recommended naming convention for mapplets is `mplt_MappletName`.
2. Create Mapplet Transformation logic. Create and Link Transformations in the same manner as in a mapping.
3. Create mapplet ports.

5.3. Normalizer Transformation

## Description

- Normalization is the process of organizing data
- The Normalizer Transformation normalizes records from COBOL and relational sources
- It is primarily used to create multiple rows from a single row of data
- It is used instead of the Source Qualifier Transformation for a COBOL source

 Capgemini  
CONSULTING TECHNOLOGY SERVICES

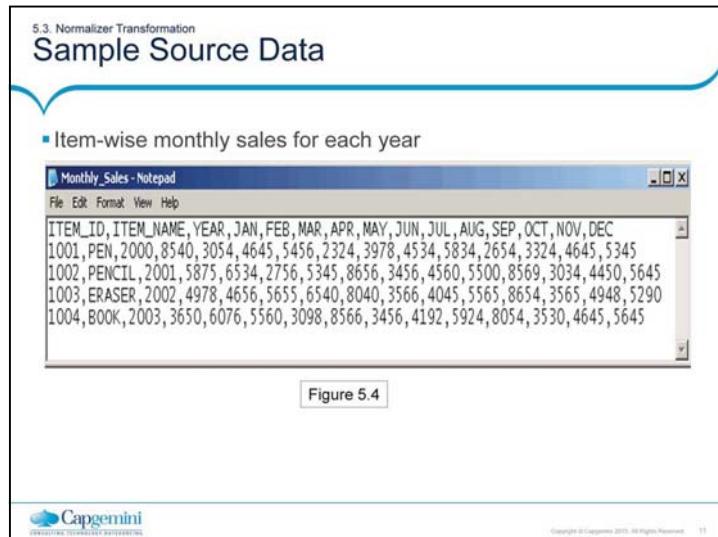
Copyright © Capgemini 2013. All Rights Reserved 10

**Normalization** is the process of organizing data. In database terms, this includes creating normalized tables and establishing relationships between those tables according to rules designed to both protect the data and make the database more flexible by eliminating redundancy and inconsistent dependencies.

The Normalizer Transformation normalizes records from COBOL and relational sources. It allows to organize the data according to the requirements. A Normalizer transformation can appear anywhere in a data flow when a relational source is normalized.

A Normalizer Transformation is used instead of the Source Qualifier Transformation when a COBOL source has to be normalized. When a COBOL source is dragged into the Mapping Designer workspace, the Normalizer Transformation automatically appears, creating input and output ports for every column in the source. COBOL sources are often stored in a denormalized format. The OCCURS statement in a COBOL file nests multiple records of information in a single record.

The Normalizer Transformation is used to break out repeated data within a record into separate records. For each new record it creates, the Normalizer Transformation generates a unique identifier. This key value is used to join the normalized records. The Normalizer Transformation can also be used with relational sources to create multiple rows from a single row of data.



In the above example, the source flat file contains the year-wise, monthly sales information in one row. The objective is to split this one row of data into multiple rows to get sales information by month.

5.3. Normalizer Transformation

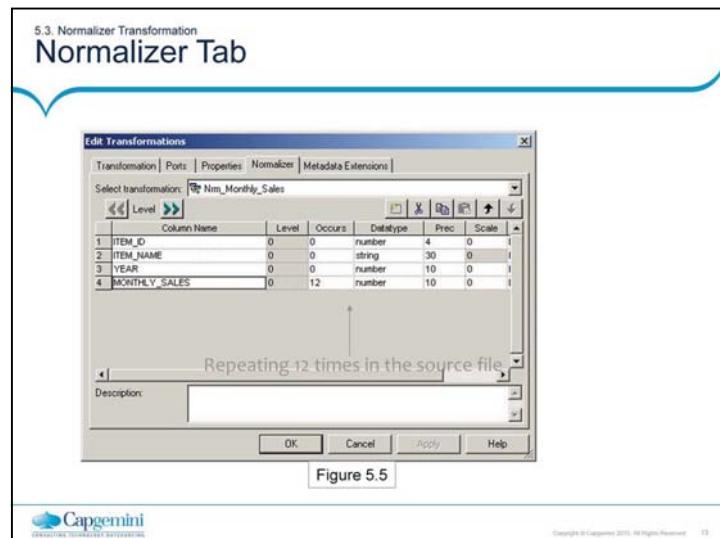
## Operations Performed

- A unique key called GKID is created for each new record.
- The Designer generates one column (port) for each OCCURS clause specified in the Normalizer tab to specify the positional index within an OCCURS clause.
- A column id by the name GCID\_occuring\_field\_name is generated for each OCCURS port.

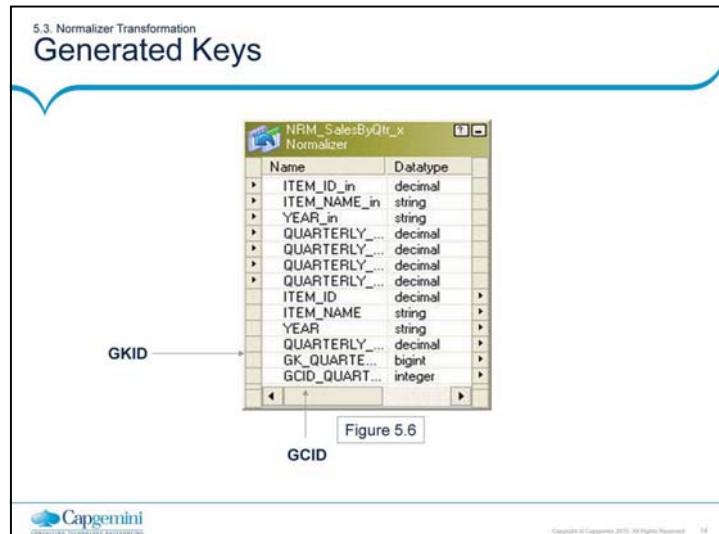
 Capgemini  
CONSULTING TECHNOLOGY SERVICES

Copyright © Capgemini 2010. All Rights Reserved 12

The first step, after bringing in the source definition, would be to create a Normalizer Transformation. Ports must be manually added under the **Normalizer** tab – they cannot be dragged from other transformations. Defining the output records structure here automatically creates the input ports that are required to create the desired output. The value in the **Occurs** column of the **Normalizer** tab designates the number of repeated ports created in the Ports tab.



In the above example, the monthly sales will have the occurs value as 12.



For each new record it creates, the Normalizer transformation generates a unique identifier called `GKID_occuring_field_name`. This key value can be used to join the normalized records.

The Designer also generates one column (port) for each `OCCURS` clause in the source file to specify the positional index within an `OCCURS` clause. The naming convention for the Normalizer column ID is: `GCID_occuring_field_name`.

5.3. Normalizer Transformation Output			
	<b>ITEM_MASTER - Notepad</b>	File	Edit Format View Help
	ITEM_MASTER_ID ITEM_ID ITEM_NAME		YEAR
	1 1001 PEN		2000
	2 1002 PENCIL		2001
x 3 1003 ERASER			2002
4 1004 BOOK			2003
	<b>SALES_DETAIL_BY_MONTH - Notepad</b>	File	Edit Format View Help
	ITEM_MASTER_ID MONTH MONTH_SALES		
GKID_MONTHLY_SALES	1 1 8540		
	1 2 3054		
	1 3 4645		
	1 4 5456		
	1 5 2324		
	1 6 3536		
	1 7 4534		
	1 8 5834		
	1 9 2654		
	1 10 3324		
	1 11 4645		
	1 12 5345		
GCID_MONTHLY_SALES			
Figure 5.8			

As shown in the above Figure 5.7, this generated unique key is GKID\_MONTHLY\_SALES and the generated column id is GCID\_MONTHLY\_SALES for each OCCURS in the source. The Normalizer ID columns informs about the order of records in an OCCURS clause. For example, if a record occurs two times, when the Workflow is run, the Integration service numbers the first record 1 and the second record 2. The Normalizer column ID is also useful to pivot input columns into rows.

## Demo: Normalizer Transformation



The slide features a large, empty central area for content. At the bottom left is the Capgemini logo, which includes a blue cloud icon and the text "Capgemini CONSULTING TECHNOLOGY INNOVATION". At the bottom right is a small copyright notice: "Copyright © Capgemini 2010. All Rights Reserved 10".

**Steps to create a Normalizer Transformation:**

1. In the Designer, create a new mapping or open an existing one.
2. Click and drag an imported source definition into the mapping. This source definition can be a COBOL file, a simple data file or even a relational source.
3. For a COBOL source, the Designer creates a Normalizer Transformation by default. For others, manually create the Normalizer Transformation.
4. For the COBOL source, Designer connects the Normalizer Transformation to the COBOL source definition. Open the new Normalizer Transformation.
5. For a COBOL source, select the **Ports** tab and review the ports in the Normalizer Transformation. For others, the port details will have to be entered in the Normalizer tab.
6. Select the **Normalizer** tab and add new output ports. Add a port corresponding to each column in the source record that contains denormalized data. The new ports only allow the number or string datatypes. Only new ports can be created in the **Normalizer** tab.
7. Set the appropriate OCCURS value for the specific source data column.

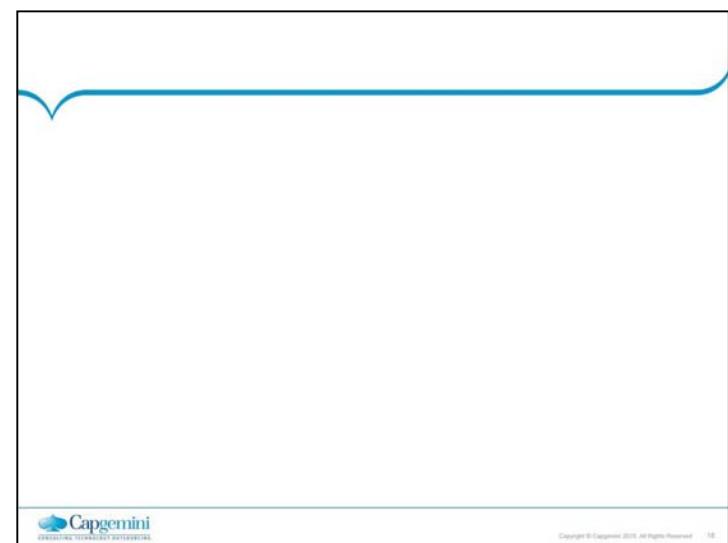
## Summary

- After completing this lesson you now:
  - Know how to configure a Mapplet using:
    - Mapplet Input Transformation
    - Mapplet Output Transformation
  - Know how to configure a Normalizer Transformation

Copyright © Capgemini 2010. All Rights Reserved. 17

Informatica PowerCenter

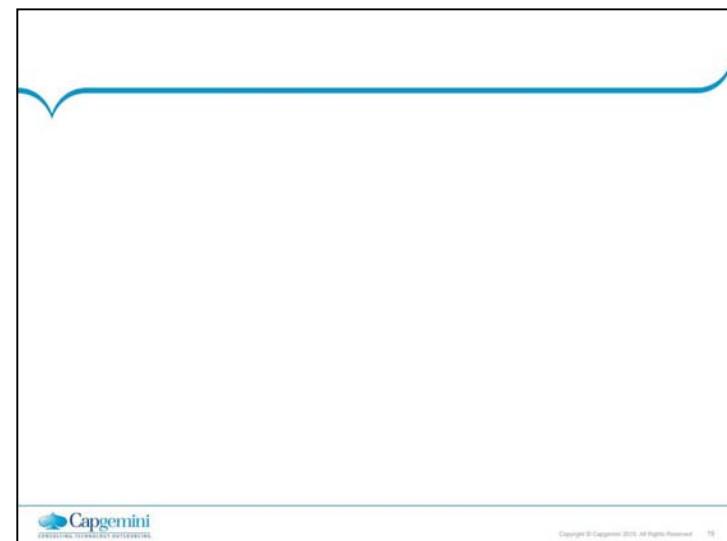
Mapplets



Copyright © Capgemini 2014. All Rights Reserved. 10

Informatica PowerCenter

Mapplets



Copyright © Capgemini 2014. All Rights Reserved. 10

## Review Question

- Question 1: Mapplets are configured using the \_\_\_\_\_.
- Question 2: A mapplet cannot have \_\_\_\_\_ definitions.
- Question 3: A Normalizer Transformation can be used for COBOL, VSAM and relational sources.
- True/False
- Question 4: GKID is a unique key generated for each column with the OCCURS clause.
- True/False



## **Informatica PowerCenter**

Lesson 6- Workflow and  
Session Log

## Lesson Objectives

- In this lesson you will learn about:
  - Workflow Variables
  - Workflow Schedule
  - Scheduler Options
  - Editing a Workflow
  - Deleting a Workflow
  - Session Logs
  - Setting Tracing Levels
  - Update Strategy Transformation

Copyright © Capgemini 2010. All Rights Reserved.

6.1. Workflow Variable

## Description

- Workflow variables are used to reference and record information.
  - For Example, we can use variable in a decision task to determine whether the previous task executed/ran properly.
  - If it did, you can run the next task and if not, you can stop the workflow.
- Types of Workflow variables
  - Predefined workflow variables
    - Task-specific
    - Built-in
  - User-defined workflow variables

 Capgemini  
Engineering Services for Business

Copyright © Capgemini 2014. All Rights Reserved. 5

## 6.1. Workflow Variable

## Predefined workflow variables

- The Workflow Manager provides predefined workflow variables for tasks within a workflow.
- Each workflow contains a set of predefined variables that you use to evaluate workflow and task conditions.

Copyright © Capgemini 2010. All Rights Reserved.

## Types of predefined variables

- Task-specific variables: The Workflow Manager provides a set of task-specific variables for each task in the workflow.
  - Use task-specific variables in a link condition to control the path the Integration Service takes when running the workflow.
  - The Workflow Manager lists task-specific variables under the task name in the Expression Editor. Eg: Condition, End Time, ErrorCode etc

Copyright © Capgemini 2012. All Rights Reserved.

Use workflow variables when you configure the following types of tasks:

Decision tasks. Decision tasks determine how the Integration Service runs a workflow. For example, use the Status variable to run a second session only if the first session completes successfully.

Links. Links connect each workflow task. Use workflow variables in links to create branches in the workflow. For example, after a Decision task, you can create one link to follow when the decision condition evaluates to true, and another link to follow when the decision condition evaluates to false.

Timer tasks. Timer tasks specify when the Integration Service begins to run the next task in the workflow. Use a user-defined date/time variable to specify the time the Integration Service starts to run the next task.

Use the following keywords to write expressions for user-defined and predefined workflow variables:

AND

OR

NOT

TRUE

FALSE

NULL

SYSDATE

## Types of predefined variables - Task Specific

Specific Variables	Description	Task Types	Datatype
<b>Condition</b>	Evaluation result of decision condition expression. If the task fails, the Workflow Manager keeps the condition set to null. Sample syntax: <code>#s_task_status.Condition = &lt;TRUE   FALSE   NULL   any integer&gt;</code>	Decision	Integer
<b>EndTime</b>	Date and time the associated task ended. Precision is to the second. Sample syntax: <code>#s_task_status.EndTime &gt; TO_DATE('11/10/2004 08:13:25')</code>	All tasks	Date/Time
<b>ErrorCode</b>	Last error code for the associated task. If there is no error, the Integration Service sets ErrorCode to 0 when the task completes. Sample syntax: <code>#s_item_summary.ErrorCode = 24013</code>	All tasks	Integer
<b>PrevTaskStatus</b>	Status of the previous task in the workflow that the Integration Service ran. Statuses include: ABORTED FAILED SUCCEEDED STOPPED Use the <code>any</code> words when writing expressions to evaluate the status of the previous task. Sample syntax: <code>#sro_TaskStatus.PrevTaskStatus = FAILED</code>	All tasks	Integer



Copyright © Capgemini 2010. All Rights Reserved.

Note: For a complete of Task Specific Predefined workflow variables, please refer to the documentation.

## Types of predefined variables

- Built-in variables: The Workflow Manager lists built-in variables under the Built-in node in the Expression Editor
- Use built-in variables in a workflow to return run-time or system information such as folder name, Integration Service Name, system date, or workflow start time.
- Eg. \$PMSessionName : Returns the name of session as a string value

Figure 6.1



Copyright © Capgemini 2010. All Rights Reserved.

\$PMSessionName : This variable can be used in mappings or mapplets.

\$PMWorkflowName: This variable can be used in a mapping, a mapplet, or in workflow tasks such as Decision tasks, and links. You can also use \$PMWorkflowName in input fields that accept mapping or workflow variables.

Note: For complete details of Built-in variables, please refer to the documentation.

## User-defined Workflow variables

- You create user-defined workflow variables when you create/edit a workflow
- These variables are used in tasks within that workflow
- You can edit and delete user-defined workflow variables
- Use user-defined variables when you need to make a workflow decision based on criteria you specify.

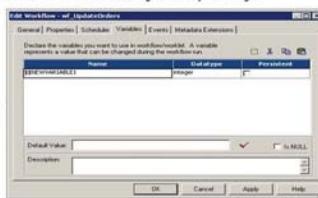


Figure 6.2



Copyright © Capgemini 2014. All Rights Reserved.

While creating a user defined work flow variable verify that the name of a user defined workflow variable does not have a \$ since single \$ is reserved for system variables. Use \$\$ instead.

6.1. Workflow Variable  
**Example**

For example, you create a workflow to load data to an orders database nightly. You also need to load a subset of this data to headquarters periodically, every tenth time you update the local orders database. Create separate sessions to update the local database and the one at headquarters. The workflow looks like as the picture below:

```
graph TD; StartWorkflow([StartWorkflow]) --> tUpdateLocal[t_UpdateLocal]; tUpdateLocal --> dUpdateHQ{$d_UpdateHQ.C...}; dUpdateHQ --> aIncrWorkflowCount[a_IncrWorkflowCount]; dUpdateHQ --> tUpdateHQ[t_UpdateHQ]; tUpdateHQ --> aIncrWorkflowCount;
```

Figure 6.3

Copyright © Capgemini 2010. All Rights Reserved.

Use a user-defined variable to determine when to run the session that updates the orders database at headquarters.

To configure user-defined workflow variables, set up the workflow as follows:

1. Create a persistent workflow variable, \$\$WorkflowCount, to represent the number of times the workflow has run.
2. Add a Start task and both sessions to the workflow.
3. Place a Decision task after the session that updates the local orders database. Set up the decision condition to check to see if the number of workflow runs is evenly divisible by 10. Use the modulus (MOD) function to do this.
4. Create an Assignment task to increment the \$\$WorkflowCount variable by one.
5. Link the Decision task to the session that updates the database at headquarters when the decision condition evaluates to true. Link it to the Assignment task when the decision condition evaluates to false. When you configure workflow variables using conditions, the session that updates the local database runs every time the workflow runs. The session that updates the database at headquarters runs every 10th time the workflow runs.

## Demo

- Create a user defined Workflow variable.



Copyright © Capgemini 2010. All Rights Reserved. 10

6.2. Workflow Schedule

## Description

- A Workflow schedule defines when the Integration service should start the data transformation and load process
- By default a Workflow runs on demand
- A Workflow schedule is defined using a Workflow Scheduler
- A Workflow schedule can be reusable or non-reusable

General | Properties | Scheduler | Variables | Events | Metadata Extensions |

A workflow can have a reusable or a non-reusable scheduler attached to it. Choose "Reusable" to browse through a list of reusable schedulers and select or edit from the list. Select "Non-Reusable" to create a new scheduler that cannot be reused.

Non Reusable    Reusable

Scheduler: Scheduler 

Figure 6.4

 Capgemini  
PROVIDING INTEGRATED BUSINESS SERVICES

Every Workflow created in the Workflow Manager, runs on a schedule. The Integration service runs a scheduled Workflow unless the prior run of the Workflow fails. When a Workflow fails, the Integration service removes the Workflow from the schedule. A Workflow must be rescheduled manually if the Workflow if a scheduled Workflow fails. By default, a Workflow runs on demand.

## Workflow Scheduler

- A scheduler is a repository object that contains a set of schedule settings
- Each Workflow has an associated scheduler
- Schedule settings defined in a Workflow are non-reusable
- A reusable Workflow scheduler can be created from the Workflows - Schedulers menu option

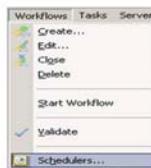


Figure 6.5

The schedule settings defined for a Workflow can be edited in the scheduler if a reusable scheduler is created. The Integration service reschedules the Workflow according to the new settings.

Non-reusable scheduler settings can only be used in the Workflow that they are defined in.

In case of a reusable scheduler, the same set of schedule settings can be used for any Workflow in a folder.

The Workflow Manager marks a Workflow as “invalid” if the scheduler associated with a Workflow is deleted.

The screenshot shows a section titled '6.3. Scheduler Options' under the heading 'Brief'. It lists several categories of options:

- Run Options
  - Run On Server Initialization
  - Run On Demand
  - Run Continuously
- Schedule Options
  - Run Once
  - Run Every
  - Customized Repeat
- Start Options
  - Start Date
  - Start Time
- End Options
  - End On
  - End After
  - Forever

At the bottom of the page, there is a Capgemini logo and a copyright notice: "Copyright © Capgemini 2010. All Rights Reserved." followed by a small icon.

A Workflow can be scheduled to run continuously, repeat at a given time or interval, or start manually.

The various options for the schedule settings are:

#### Run Options

Run On Server Initialization - The Integration service runs the Workflow as soon as the service is initialized. The Integration service then starts the next run of the Workflow according to the settings in Schedule Options

Run On Demand - The Integration service runs the Workflow when the Workflow is started manually

Run Continuously - The Integration service runs the Workflow as soon as the service initializes. The Integration service then starts the next run of the Workflow as soon as it finishes the previous run

#### Schedule Options

Run Once - The Integration service runs the Workflow once, as scheduled in the scheduler

Run Every - The Integration service runs the Workflow at regular intervals, as configured

Customized Repeat - The Integration service runs the Workflow on the dates and times specified in the Repeat dialog box

**Demo**

- Create a Scheduler and Schedule a Workflow



Copyright © Capgemini 2010. All Rights Reserved.

Steps to create a reusable scheduler:

Select Schedulers from Workflows menu.

Click New to add a new scheduler.

In the General tab, enter a name for the scheduler.

Configure the scheduler settings in the Scheduler tab.

Click on OK.

Steps to schedule a Workflow:

In the Workflow Designer, open the Workflow.

Choose Workflows | Edit.

In the Scheduler tab, choose Non-reusable if you want to create a non-reusable set of schedule settings for the Workflow. Choose Reusable if an existing reusable scheduler for the Workflow has to be selected.

Note: If there is no reusable scheduler in the folder, it has to be created before one chooses Reusable. The Workflow Manager displays a warning message if there is no existing reusable scheduler.

Click the right side of the Scheduler field to select the reusable scheduler.

Click on OK.

6.4. Editing a Workflow

### Brief

- A Workflow can be edited to modify various properties
- The repository updates the Workflow information when the Workflow is saved

Figure 6.6

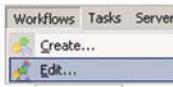
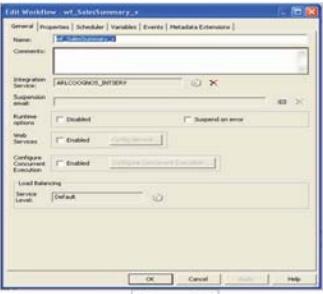


Figure 6.7



Copyright © Capgemini 2010. All Rights Reserved.

To edit a Workflow the menu option Workflows | Edit is used. When a Workflow is edited, the repository updates the Workflow information when the Workflow is saved. If a running Workflow is edited, the Integration service uses the updated information the next time the Workflow runs.

6.5. Deleting a Workflow

## Explanation

- A Workflow that is no longer in use can be deleted
- If a Workflow that is running is deleted, then the Integration service aborts the Workflow
- A Workflow can be deleted from
  - The Navigator window
  - Workflow Designer workspace

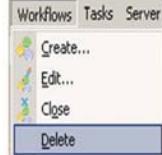


Figure 6.8

Copyright © Capgemini 2010. All Rights Reserved. 10

When a Workflow is deleted, all non-reusable Tasks and reusable Task instances associated with the Workflow are deleted. Reusable Tasks used in the Workflow remain in the folder when the Workflow is deleted.

If a Workflow that is running is deleted, then the Integration service aborts the Workflow. If a Workflow that is scheduled to run is deleted, the Integration service removes the Workflow from the schedule.

A Workflow can be deleted in the Navigator window, or the Workflow currently displayed in the Workflow Designer workspace can be deleted.

To delete a Workflow from the Navigator window, open the folder, select the Workflow and press the Delete key

To delete a Workflow currently displayed in the Workflow Designer workspace, choose Workflows-Delete

6.6. pmcmd command

## Explanation

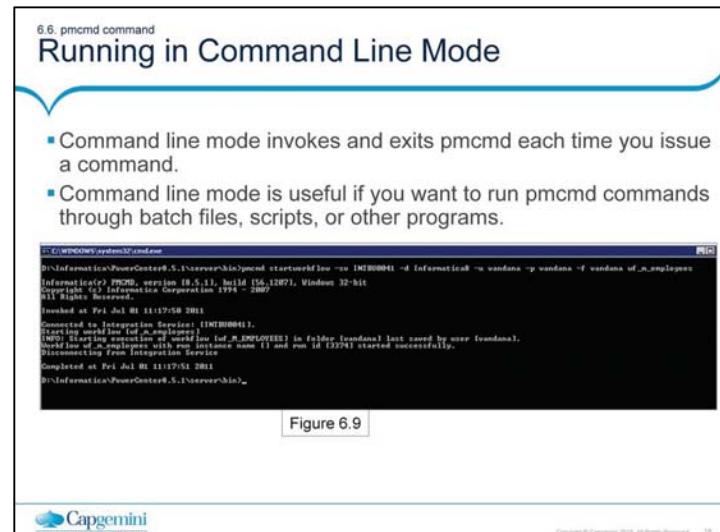
- pmcmd is a program you use to communicate with the Integration Service.
- It is used to perform some of the tasks that can also be performed in the Workflow Manager, such as starting and stopping workflows and sessions.
- It can be used in the following modes
  - Command line mode
  - Interactive mode

 Capgemini  
PROVIDING INNOVATION FOR BUSINESS

Copyright © Capgemini 2010. All Rights Reserved. 17

Command line mode: You invoke and exit pmcmd each time you issue a command. You can write scripts to schedule workflows with the command line syntax. Each command you write in command line mode must include connection information to the Integration Service.

Interactive Mode: You establish and maintain an active connection to the Integration Service. This lets you issue a series of commands.



By default, the PowerCenter installer installs pmcmd in the \server\bin directory.

When you run pmcmd in command line mode, you enter connection information such as domain name, Integration

To run pmcmd commands in command line mode:

At the command prompt, switch to the directory where the pmcmd executable is located.

Enter pmcmd followed by the command name and its required options and arguments:

```
pmcmd command_name [-option1] argument_1 [-option2]
argument_2...
```

Service name, user name and password in each command. For example, to start the workflow "wf\_SalesAvg" in

folder "SalesEast," use the following syntax:

```
D:\Informatica\powercenter8.5.1\server\bin>pmcmd startworkflow -sv INTIM0045 -d
Informatica8 -u INF1 -p INF1 -f INF1FLD wf_SalesAvg
```

**Running in Interactive Mode**

- pmcmd is used in interactive mode to start and stop workflows and sessions without writing a script.
- In this mode, the connection information such as domain name, Integration Service name, username, and password is entered.
- The subsequent commands are run without entering the connection information for each command.

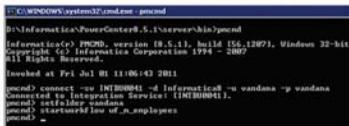


Figure 6.10



To run pmcmd commands in interactive mode:

1. At the command prompt, switch to the directory where the pmcmd executable is located.

By default, the PowerCenter installer installs pmcmd in the \server\bin directory.

2. At the command prompt, type pmcmd.

This starts pmcmd in interactive mode and displays the pmcmd> prompt. You do not have to type pmcmd before each command in interactive mode.

3. Enter connection information for the domain and Integration Service. For example:

```
connect -sv MyIntService -d MyDomain -u seller3 -p jackson
```

4. Type a command and its options and arguments in the following format:

```
command_name [-option1] argument_1 [-option2] argument_2...
```

pmcmd runs the command and displays the prompt again.

5. Type exit to end an interactive session.

For example, the following commands invoke the interactive mode, establish a connection to Integration Service

"MyIntService," and start workflows "wf\_SalesAvg" and "wf\_SalesTotal" in folder "SalesEast":

```
pmcmd
```

```
pmcmd> connect -sv INTIM0045 -d Informatica8 -u INF1 -p INF1
```

```
pmcmd> setfolder INF1
```

```
pmcmd> startworkflow wf_SalesAvg
```

```
pmcmd> startworkflow wf_SalesTotal
```

## 6.6. pmcmd command Return Codes

- In command line mode, pmcmd indicates the success or failure of a command with a return code.
- Return code "0" indicates that the command succeeded.
- Any other return code indicates that the command failed.



Copyright © Capgemini 2014. All Rights Reserved. 30

6.6. pmcmd command

## Session Log Messages

- The Integration service precedes each message in the log file with a thread identification and then a code and number
- The code defines a group of messages for a specific process
- The number defines a specific message
- The message can provide general information or it can be an error message
- The level of detail in a session log depends on the tracing level set

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2010. All Rights Reserved. 21

The Session Log contains informational, warning, and error messages from the session processes.

The session log includes a load summary that reports the number of rows inserted, updated, deleted, and rejected for each target as of the last commit point. The Integration service reports the load summary for each session by default.

6.6. pmcmd command

## Configuring Session Logs

- Properties that can be configured for a Session Log are:
  - Location
    - \$PMSSessionLogDir server variable is mapped to a local directory on the Integration service
    - The default name for the session log is mapping name.log
    - It can be changed
  - Archive
    - Used to specify the number of runs of the Session Log to save
    - Default is one
    - Tracing levels
      - Define the amount of detail in the Session Log

 Capgemini  
PROVIDING INNOVATION FOR BUSINESS

Copyright © Capgemini 2010. All Rights Reserved. 10

Session Logs options are configured in the session properties. The following information can be configured for a session log:

Location - The directory where the session log should be created. By default, the Integration service creates the session log in the directory configured for the \$PMSSessionLogDir server variable. A different directory can be entered, but if the directory does not exist or is not local to the Integration service that runs the session, the session fails.

Name - The session log can be named or the default name can be accepted. The default name for the session log is s\_mapping name.log

Archive - By default, the Integration service does not archive session logs. It creates one session log for each session and overwrites the existing log with the latest session log. We can specify the number of log files to be created by altering the properties- Save session log for these runs, in Task properties-Config objects-log options

Tracing levels - Setting a tracing level for the session can control the type of information the Integration service includes in the session log. By default, the Integration service uses tracing levels configured in the mapping

The name and location of the session log can be configured on the Properties tab of the session properties in the Workflow Manager.

6.7. Setting Tracing Levels

## Description

Tracing Level	Description
None	Integration service uses the tracing level set in the mapping.
Terse	Integration service logs initialization information as well as error messages and notification of rejected data.
Normal	<ul style="list-style-type: none"><li>▪ Integration service logs initialization and status information</li><li>▪ Errors encountered</li><li>▪ Skipped rows due to transformation row errors</li><li>▪ Summarizes session results, but not at the level of individual rows</li></ul>

 Capgemini  
PROVIDING INNOVATION FOR SUSTAINABILITY

Copyright © Capgemini 2014. All Rights Reserved. 33

The amount of detail in the session log depends on the tracing level set. Tracing levels can be defined for each transformation or for the entire session. By default, the Integration service uses tracing levels configured in the mapping.

Setting a tracing level for the session overrides the tracing levels configured for each transformation in the mapping. If a normal tracing level or higher is selected, the Integration service writes row errors into the session log, including the transformation in which the error occurred and complete row data.

6.7. Setting Tracing Levels

## Description (contd..)

Tracing Level	Description
<b>Verbose Initialization</b>	<ul style="list-style-type: none"><li>▪ Additional initialization details</li><li>▪ Name of index and data files used</li><li>▪ Detailed transformation statistics</li></ul>
<b>Verbose Data</b>	In addition to verbose initialization tracing, Integration service logs each row that passes into the mapping

 Capgemini  
PROVIDING INNOVATION FOR BUSINESS

Copyright © Capgemini 2010. All Rights Reserved. 30

Generally the tracing level is kept at Normal. For debugging purpose, the tracing level is set to Verbose Data to get a row by row data loading status.

**6.7. Setting Tracing Levels**

## Viewing Session Logs

- **Session Logs**
- Can be viewed through the Workflow monitor
- Are temporary text files that can be opened with any text editor



Figure 6.11

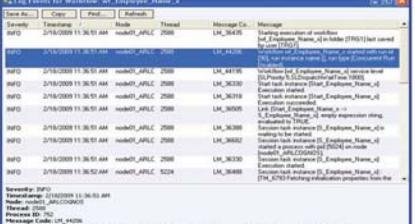


Figure 6.12

Session logs are text files that can be opened with any text editor. The Integration service saves session logs in the directory specified in the Session Log File Directory field in the session properties.

Session logs can be viewed through the Workflow Monitor. The Workflow Monitor creates a temporary file that stores the session log. A session log file can be viewed even if the session fails.

The Integration service generates the session log based on the Integration service code page. The language of the session log can be specified based on the locale of the machine hosting the Integration service .

6.8. Update Strategy Transformation

## Description

- Used to specify how each individual row will be used to update target tables (insert, update, delete, reject)
- Active transformation
- Commonly used in combination with Lookup transformation
- Update strategy can be used at two levels
  - Within a session
  - Within a mapping

 Capgemini  
PROVIDING INTEGRATED BUSINESS SERVICES

Copyright © Capgemini 2010. All Rights Reserved.

While designing the data warehouse, often there might be a requirement to decide what type of information to store in targets. As part of the target table design, we need to determine whether to maintain all the historic data or just the most recent changes.

For example, there might be a target table, T\_CUSTOMERS, that contains customer data. When a customer address changes, there might be a need to save the original address in the table, instead of updating that portion of the customer row. In this case, a new row needs to be created containing the updated address. The original row may be required to be preserved with the old customer address. This illustrates how historical information might be stored in a target table. However, if the T\_CUSTOMERS table has to be a snapshot of current customer data, the existing customer row can be updated. The original address might be lost in this case. The model chosen constitutes the update strategy, how to handle changes to existing rows.

Update strategy can be used at two different levels:

Within a session - When a session is configured, the Integration service can be instructed to either treat all rows in the same way (for example, treat all rows as inserts), or use instructions coded into the session mapping to flag rows for different database operations.

Within a mapping - Within a mapping, the Update Strategy Transformation is used to flag rows for insert, delete, update, or reject.

6.8. Update Strategy Transformation

## Setting the Update Strategy

- One of the following methods can be used to define the update strategy
  - Add an Update Strategy Transformation to the mapping to control how rows are flagged for insert, update, delete, or reject within a mapping
  - Define an Update Strategy when a session is configured.

 Capgemini  
PROVIDING INTEGRATED BUSINESS SERVICES

Copyright © Capgemini 2010. All Rights Reserved. 31

To control how rows are flagged for insert, update, delete, or reject within a mapping, add an Update Strategy transformation to the mapping. The update strategy can also be defined when a session is configured. All rows can be flagged for insert, delete, or update. Alternatively the data driven option can be selected, where the Integration service follows instructions coded into Update Strategy Transformations within the session mapping.

6.8. Update Strategy Transformation

## Constants for Update Strategy Transformation

The following constants can be used in the Update Strategy expression while creating an Update Strategy Transformation in the mapping

Operation	Constant	Numeric Value
Insert	DD_INSERT	0
Update	DD_UPDATE	1
Delete	DD_DELETE	2
Reject	DD_REJECT	3

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2014. All Rights Reserved. 30

DD\_INSERT - Flags records for insertion in an update strategy expression.  
DD\_INSERT is equivalent to the integer literal 0

DD\_UPDATE - Flags records for update in an update strategy expression.  
DD\_UPDATE is equivalent to the integer literal 1

DD\_DELETE - Flags records for deletion in an update strategy expression.  
DD\_DELETE is equivalent to the integer literal 2

DD\_REJECT - Flags records for rejection in an update strategy expression.  
DD\_REJECT is equivalent to the integer literal 3. DD\_REJECT is generally used to filter or validate data. If you flag a record as reject, the Integration service skips the record and writes it to the session reject file

**Example:**

The following expression marks items with an ID number of 1001 for deletion, and all

other items for insertion:

`IIF( ITEM_ID = 1001, DD_DELETE, DD_INSERT )`

## Demo

- Create an Update Strategy Transformation



Copyright © Capgemini 2010. All Rights Reserved.

Steps to create an Update Strategy Transformation:

In the Mapping Designer, add an Update Strategy Transformation to a mapping.

Choose Layout | Link Columns.

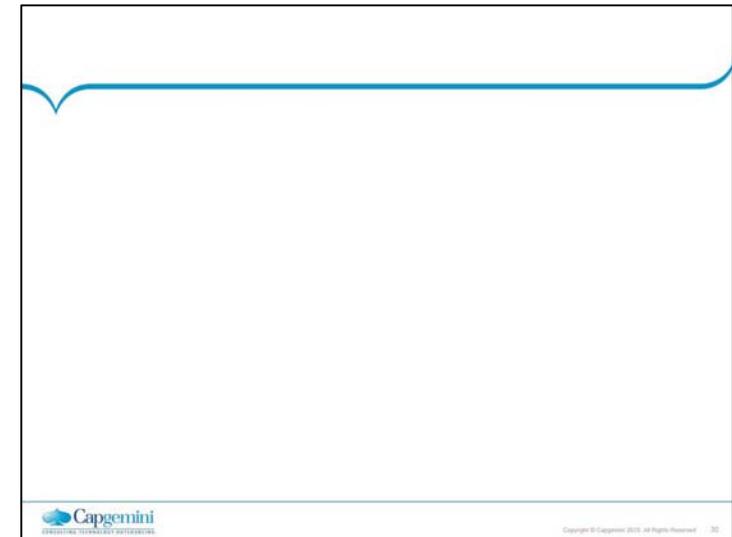
Click and drag all ports from another transformation representing data to be passed through the Update Strategy Transformation. In the Update Strategy Transformation, the Designer creates a copy of each port which is clicked and dragged. The Designer also connects the new port to the original port. Each port in the Update Strategy Transformation is a combination input/output port.

Normally, all of the columns destined for a particular target would be selected. After they pass through the Update Strategy Transformation, this information is flagged for update, insert, delete, or reject.

Open the Update Strategy Transformation and rename it. The naming convention for Update Strategy Transformations is UPD\_TransformationName.

Click the Properties tab.

Click the button in the Update Strategy Expression field. The Expression Editor appears.

Copyright © Capgemini 2014. All Rights Reserved. 50

Enter an update strategy expression to flag rows as inserts, deletes, updates, or rejects.

Validate the expression and click OK.

Click OK to save the changes.

Connect the ports in the Update Strategy Transformation to another transformation or a target instance.

Choose Repository | Save.

## Summary

- After completing this lesson you now know:
  - To schedule a Workflow
  - To edit and deleted a Workflow
  - View and configure session logs
  - Set tracing levels
  - Configure an Update Strategy Transformation

Copyright © Capgemini 2014. All Rights Reserved. 31

## Review Question

- Question 1: The default name for the session log is \_\_\_\_\_
  
- Question 2: In case of a \_\_\_\_\_ scheduler, the same set of schedule settings can be used for Workflows in the folder
  
- Question 3: The \_\_\_\_\_ tracing level summarizes session results, but not at the level of individual rows
  
- Question 4: Update Strategy Transformation is a passive transformation.
- True/False

Copyright © Capgemini 2014. All Rights Reserved. 10

## Informatica PowerCenter

Lesson 7- Additional  
Transformations

## Lesson Objectives

- In this lesson you will learn about:
  - Router Transformation
  - Sorter Transformation
  - Stored Procedure Transformation
  - Sequence Generator Transformation
  - Union Transformation
  - Reusable Transformations
  - Shared Folder



7.1. Router Transformation

## Description

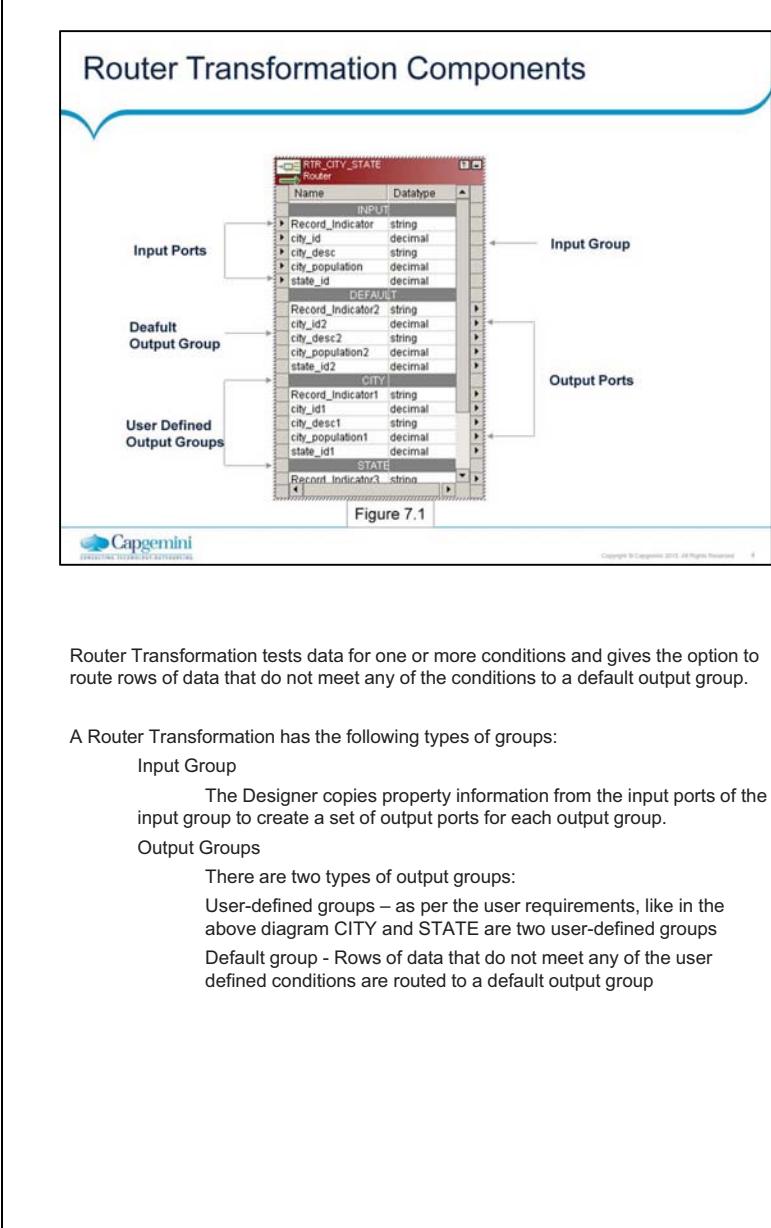
- A Router Transformation is used to test input data for multiple conditions
- It is an Active Transformation
- A Router Transformation:
  - Tests data for one or more conditions
  - Gives the option to route rows of data that do not meet any of the conditions to a default output group
- A Router Transformation is more efficient

 Capgemini  
INTEGRATING BUSINESS & TECHNOLOGY

Copyright © Capgemini 2010. All Rights Reserved.

A Router Transformation is similar to a Filter Transformation because both transformations allow to use a condition to test data. A Filter Transformation tests data for one condition and drops the rows of data that do not meet the condition. However, a Router Transformation tests data for one or more conditions and gives the option to route rows of data that do not meet any of the conditions to a default output group.

The Router Transformation is more efficient. For example, to test data based on three conditions, only one Router Transformation needs to be configured instead of three Filter Transformations to perform this Task. Likewise, when a Router Transformation is used in a mapping, the Integration service processes the incoming data only once. When multiple Filter Transformations are used in a mapping, the Integration service processes the incoming data for each transformation.



Router Transformation tests data for one or more conditions and gives the option to route rows of data that do not meet any of the conditions to a default output group.

A Router Transformation has the following types of groups:

#### Input Group

The Designer copies property information from the input ports of the input group to create a set of output ports for each output group.

#### Output Groups

There are two types of output groups:

User-defined groups – as per the user requirements, like in the above diagram CITY and STATE are two user-defined groups

Default group - Rows of data that do not meet any of the user defined conditions are routed to a default output group

7.2. Sorter Transformation

## Description

- The Sorter Transformation allows to sort data in ascending or descending order
- It is an Active Transformation
- Sorting can be case-sensitive
- Distinct output rows can be specified

Figure 7.2

Copyright © Capgemini 2010. All Rights Reserved.

The Sorter Transformation can sort data from relational tables or flat files. Sort takes place on the Integration service machine. Multiple sort keys are supported. The Sorter Transformation is often more efficient than a sort performed on a database with an ORDER BY clause.

The Sorter Transformation can also be used to sort data passing through an Aggregator Transformation configured to use sorted input.

When a Sorter Transformation is created in a mapping, one or more ports can be specified as a sort key and configure each sort key port to sort in ascending or descending order.

## Demo

- Configure a Router and Sorter Transformation



Capgemini  
Enterprise Software & Services

Copyright © Capgemini 2010. All Rights Reserved.

Steps to create a Router Transformation:

In the Mapping Designer, open a mapping.

Choose Transformation | Create menu option. Select Router Transformation, and enter the name of the new transformation. The naming convention for the Router Transformation is RTR\_TransformationName. Click Create, and then click Done.

Select and drag all the desired ports from a transformation to add them to the Router Transformation, or manually create input ports on the Ports tab.

Double-click the title bar of the Router Transformation to edit transformation properties.

Click the Transformation tab and configure transformation properties as desired.

Click the Groups tab, and then click the Add button to create a user-defined group. The Designer creates the default group when the first user-defined group is created.

Click the Group Filter Condition field to open the Expression Editor.

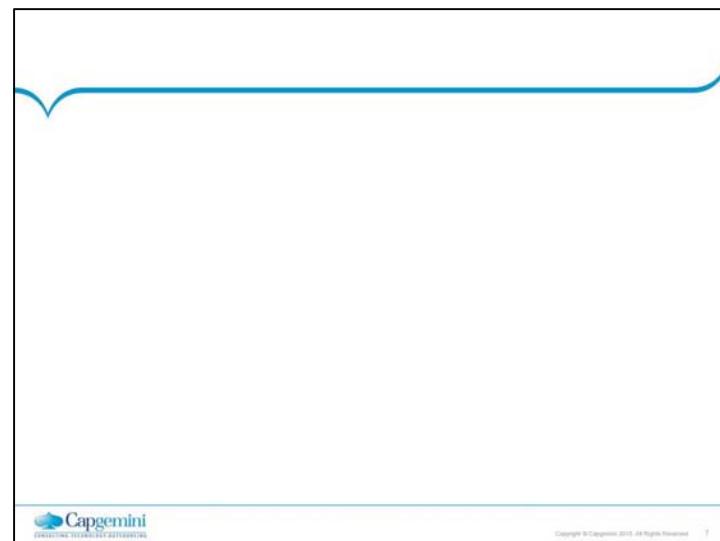
Enter a group filter condition.

Click Validate to check the syntax of the conditions entered.

Click OK.

Connect group output ports to transformations or targets.

Choose Repository | Save menu option.



Copyright © Capgemini 2010. All Rights Reserved.

#### Steps to create a Sorter Transformation:

In the Mapping Designer, choose Transformation | Create. Select the Sorter Transformation. The naming convention for Sorter Transformations is SRT\_TransformationName. Enter a description for the transformation. This description appears in the Repository Manager, making it easier to understand what the transformation does.

Enter a name for the Sorter and click Create. The Designer creates the Sorter Transformation.

Click Done.

Drag the ports which are to be sorted into the Sorter Transformation. The Designer creates the input/output ports for each port included.

Double-click the title bar of the transformation to open the Edit Transformations dialog box.

Select the Ports tab.

Select the ports to be used as the sort key.

For each port selected as part of the sort key, specify whether the Integration service has to sort data in ascending or descending order.

Select the Properties tab. Modify the Sorter Transformation properties as needed.

Click OK.

Choose Repository | Save to save changes to the mapping.

7.3. Stored Procedure Transformation

## Description

- A stored procedure is a precompiled collection of SQL statements and optional flow control statements, similar to an executable script stored and run within the database
- A Stored Procedure Transformation is used to call a database store procedure to perform a query or calculation that would otherwise be configured in a mapping
- It is a Passive Transformation

 Capgemini  
Enterprise Software & Services

Copyright © Capgemini 2010. All Rights Reserved.

Database developers and programmers use stored procedures for various Tasks within databases, since stored procedures allow greater flexibility than SQL statements. Stored procedures also provide error handling and logging necessary for critical Tasks. Developers create stored procedures in the database using the client tools provided with the database.

A Stored Procedure Transformation is used to call a store procedure created in the database to perform a query or calculation that would otherwise be configured in a mapping.

For example, if there is a well-tested stored procedure for calculating sales tax created in the database, a Stored Procedure Transformation calling the database stored procedure can be configured, instead of recreating the same calculation in an Expression Transformation.

The stored procedure must exist in the database before creating a Stored Procedure Transformation, and the stored procedure can exist in a source, target, or any database with a valid connection to the Integration service.

7.3. Stored Procedure Transformation  
**Connected/Unconnected**

Connected	Unconnected
Part of the mapping data flow	Separate from the mapping data flow
The mapping data passes through this transformation	It either runs before or after the session, or is called by an expression in another transformation in the mapping.
Returns multiple values (by linking output ports to another transformation)	Returns one value. PROC_RESULT is used in the parameter list of the stored procedure call to designate the return value

 Capgemini  
Engineering Services

Copyright © Capgemini 2012. All Rights Reserved.

#### Connected Stored Procedure Transformation

The flow of data through a mapping in connected mode also passes through the Stored Procedure Transformation. All data entering the transformation through the input ports affects the stored procedure. A connected Stored Procedure Transformation should be used when needed data from an input port sent as an input parameter to the stored procedure, or the results of a stored procedure sent as an output parameter to another transformation.

#### Unconnected Stored Procedure Transformation

The Unconnected Stored Procedure Transformation is not connected directly to the flow of the mapping. It either runs before or after the session, or is called by an expression in another transformation in the mapping.

Page 07-9

## Demo

- Creating a Stored Procedure Transformation



 Capgemini  
Engineering Services for Enterprises

Copyright © Capgemini 2010. All Rights Reserved.

### Steps to create a Stored Procedure Transformation:

Create the stored procedure in the database and test it through the provided database client tools.

Import or create the Stored Procedure Transformation, by providing ports for any necessary input/output and return values.

Determine whether to use the transformation as connected or unconnected.

Determine how the stored procedure relates to the mapping before configuring the transformation.

If connected, map the appropriate input and output ports. Click and drag the appropriate input flow ports to the transformation, and create mappings from output ports to other transformations.

If unconnected, configure it to run from an expression in another transformation. The expression can contain variables, and may or may not include a return value.

Configure the session. The session properties in the Workflow Manager includes options for error handling when running stored procedures and several SQL override options.

**7.4. Sequence Generator Transformation**

## Description

- The Sequence Generator Transformation is used to generate numeric values
- It is a Passive Transformation
- It has two predefined output ports
  - CURRVAL
  - NEXTVAL
- The output ports cannot be changed
- No ports can be added



Figure 7.3

A Sequence Generator Transformation can be used to generate numeric values for the following:

- Create keys
- Replace missing values
- Cycle through a sequential range of numbers

#### NEXTVAL

The NEXTVAL port is used to generate a sequence of numbers by connecting it to a transformation or target.

It is used to connect to a downstream transformation to generate the sequence based on the Current Value and Increment By properties. It can be connected to multiple transformations to generate unique values for each row in each transformation.

For example, NEXTVAL can be connected to two target tables in a mapping to generate unique primary key values. The Integration service creates a column of unique primary key values for each target table. The figure above illustrates connecting NEXTVAL to two target tables in a mapping.

Approximately two billion primary or foreign key values can be created with the Sequence Generator by connecting the NEXTVAL port to the desired transformation or target and using the widest range of values (1 to 2147483647) with the smallest interval (1).

#### CURRVAL

CURRVAL is the NEXTVAL value plus one or NEXTVAL plus the Increment By value. When a row enters the transformation connected to the CURRVAL port, the Integration service passes the last-created NEXTVAL value plus one.

## Demo

- Creating a Sequence Generator Transformation



Copyright © Capgemini 2010. All Rights Reserved. 13

Steps to create a Sequence Generator Transformation:

In the Mapping Designer, select Transformation | Create. Select the Sequence Generator transformation. The naming convention for Sequence Generator Transformations is SEQ\_TransformationName.

Enter a name for the Sequence Generator, and click Create. Click Done.

Double-click the title bar of the transformation to open the Edit Transformations dialog box.

Enter a description for the transformation. This description appears in the Repository Manager, making it easier to understand what the transformation does.

Select the Properties tab. Enter settings as necessary.

Note: Unlike other transformations, the Sequence Generator Transformation properties cannot be overridden at the session level. This protects the integrity of the sequence values generated.

Click OK.

To generate new sequences during a session, connect the NEXTVAL port to at least one transformation in the mapping. The NEXTVAL or CURRVAL ports can be used in an expression in other transformations.

Choose Repository | Save.

**7.5. Union Transformation**

## Description

- Union transformation is used to merge data from multiple input sources into one output target.
- The Union transformation is an active transformation
- Reverse of a Router transformation

Figure 7.4

Copyright © Capgemini 2010. All Rights Reserved. 13

The Union transformation is a multiple input group transformation that you can use to merge data from multiple pipelines or pipeline branches into one pipeline branch. Using the Union transformation to merge data from multiple sources is similar to using the UNION ALL SQL statement to combine the results from two or more SQL statements.

You can connect heterogeneous sources to a Union transformation. The Union transformation merges sources with matching ports and outputs the data from one output group with the same ports as the input groups.

#### Rules and Guidelines

Consider the following rules and guidelines when you work with a Union transformation:

You can create multiple input groups, but only one output group.

All input groups and the output group must have matching ports. The precision, datatype, and scale must be identical across all groups.

The Union transformation does not remove duplicate rows. To remove duplicate rows, you must add another transformation such as a Router or Filter transformation.

You cannot use a Sequence Generator or Update Strategy transformation upstream from a Union transformation.

The Union transformation does not generate transactions.

## Demo

- Creating a Union Transformation



Capgemini  
Consulting Services for Businesses

Steps to create a Union Transformation:

In the Mapping Designer, select Transformation | Create. Select the Union transformation. The naming convention for Union Transformations is UN\_TransformationName.

Enter a name for the Union Transformations, and click Create. Click Done. The Designer creates the Union Transformation.

Double-click the title bar of the transformation to open the Edit Transformations dialog box.

Enter a description for the transformation. This description appears in the Repository Manager, making it easier to understand what the transformation does.

Click the Groups tab.

Add an input group for each pipeline or pipeline branch you want to merge. The Designer assigns a default name for each group. The groups can be renamed.

Click the Group Ports tab.

Add a new port for each row of data you want to merge. Enter port properties such as name and datatype.

Click the Properties tab to configure the tracing level.

Click OK.

Choose Repository | Save to save changes.

7.6. Reusable Transformations

## Description

- Reusable Transformations are transformation objects which are defined once and used multiple times in any number of mappings
- A reusable transformation is created in two ways:
  - Using the Transformation Developer
  - Promoting a standard Mapping transformation as reusable
- All the transformation objects can be reusable except for:
  - Source Qualifier Transformation
  - ERP Source Qualifier Transformation

 Capgemini  
Engineering Services for Enterprises

Copyright © Capgemini 2010. All Rights Reserved. 10

Mappings contain two classes of transformations, standard and reusable. Transformations created in a mapping are standard, they cannot be used in any other mapping. If a transformation is common in multiple mappings then, it can be made reusable.

Reusable Transformations are transformation objects which are defined once and can be used multiple times in any number of mappings.

Reusable Transformations have an additional advantage in that every instance of the transformation that appears in the mapping inherits any changes made to the original transformation. The Designer stores each reusable transformation as metadata separate from any mappings that use the transformation.

Except for the Source Qualifier and the ERP Source Qualifier, all of the transformation objects can be reusable.

They can be created in two ways:

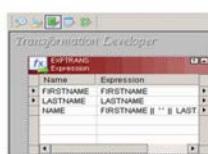
By promoting a standard transformation in the Mapping Designer

By creating the transformation in the Transformation Developer

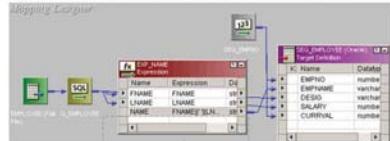
Once a reusable transformation has been created, it can only be edited in the Transformation Developer.

## 7.6. Reusable Transformations Instances

- When a Reusable transformation is added to a mapping, an instance of the Reusable transformation gets added
- If the Reusable transformation is changed, all instances of the Reusable transformation automatically inherit the changes



Reusable Expression Transformation



Instance of the Reusable Transformation in a mapping

Figure 7.6



Copyright © Capgemini 2010. All Rights Reserved.

When a Reusable Transformation is added to a mapping, a copy (or instance) of the transformation gets added. The definition of the transformation still exists outside the mapping.

The instance of a Reusable Transformation is a pointer to that transformation. When the transformation in the Transformation Developer is changed, its instances automatically reflect the changes. This feature saves a great deal of work. Instead of updating the same transformation in every mapping, the Reusable transformation can be updated once. All instances of the transformation will automatically inherit the change.

Note: Instances do not inherit changes to property settings, only modifications to ports, expressions, and the name of the transformation.

As seen in the above example, we can configure an Expression Transformation to concatenate 'First name' and 'Last name'. This expression can now be used in any mapping where the concatenation of names is required.

## Demo

- Creating a Reusable Transformation



Capgemini  
Engineering Services for Enterprises

### Steps to create a Reusable Transformation:

In the Designer, switch to the Transformation Developer.

Click the button on the Transformation toolbar corresponding to the type of transformation to be created.

Click and drag within the workbook to create the transformation.

Double-click the transformation title bar to open the dialog displaying its properties.

Click the Rename button and enter a descriptive name for the transformation, and click OK. The naming convention for Reusable Transformations is `reuTransformation_Name`.

Click the Ports tab, then add any input and output ports needed for this transformation.

Set the other properties of the transformation, and click OK. These properties vary according to the transformation created. For example, if an Expression Transformation is created, an expression for one or more of the transformation output ports is to be entered. If a Stored Procedure Transformation is created, identify the stored procedure to be called.

Choose Repository | Save.

## Summary

- After completing this lesson you now know:
  - Router Transformation
  - Sorter Transformation
  - Stored Procedure Transformation
  - Sequence Generator Transformation
  - Union Transformation
  - Reusable Transformations



## Shared Folder

- You can designate a folder to be shared.
- In the Designer, shared folders allow users to create shortcuts to objects in the folder.
- If you have an object that you want to use in several mappings or across multiple folders, you can place the object in a shared folder.
- You can access the object from other folders by creating shortcuts to the object.
- Shortcuts inherit changes to the original object.
- When you create a shared folder, the folder icon in the Navigator displays an open hand icon.

Copyright © Capgemini 2010. All Rights Reserved. 10

## Demo

- Create and use a shared folder



## XML Transformation

- Different types of XML Transformation are
  - XML Source Qualifier Transformation
  - XML Parser Transformation
  - XML Generator Transformation Overview
  - XML Source Qualifier Transformation
  - It's an Active Transformation, as well as Connected.
  - Just like the normal SourceQualifier Transformation we can use the XML Source Qualifier Transformation by dragging an XML source definition to the Mapping Designer workspace or by manually creating one.
  - The source definition needs to be connected to the target via XML Source Qualifier Transformation. This Source qualifier defines the data elements that the Integration Service reads when it executes a session.
  - XML Source Qualifier has one input or output port for every column in the source.
  - If you remove an XML source definition from a mapping, the Designer also removes the corresponding XML Source Qualifier transformation. But we can link one XML source definition to one XML Source Qualifier transformation.
  - Also we can link ports of XML Source Qualifier Transformation to ports of different transformations

Copyright © Capgemini 2010. All Rights Reserved.

## XML Transformation

Continued .....

- XML Parser Transformation
  - It's an Active Transformation, as well as Connected.
  - We use an XML Parser transformation to extract XML inside a pipeline and then pass this to the target.
  - The XML is extracted from the source systems such as files or databases. The XML Parser transformation reads XML data from a single input port and writes data to one or more output ports.
- XML Generator Transformation Overview
  - It's an Active Transformation, as well as Connected.
  - We use XML Generator Transformation Overview to create XML inside a pipeline. It reads data from source such as files and databases and generates XML in the pipeline. The XML Generator transformation accepts data from multiple ports and writes XML through a single output port.



Copyright © Capgemini 2010. All Rights Reserved. 33

## Demo

- Creating a XML Transformation



Capgemini  
Enterprise Services for Information

### Steps to create a Reusable Transformation:

In the Designer, switch to the Transformation Developer.

Click the button on the Transformation toolbar corresponding to the type of transformation to be created.

Click and drag within the workbook to create the transformation.

Double-click the transformation title bar to open the dialog displaying its properties.

Click the Rename button and enter a descriptive name for the transformation, and click OK. The naming convention for Reusable Transformations is `reuTransformation_Name`.

Click the Ports tab, then add any input and output ports needed for this transformation.

Set the other properties of the transformation, and click OK. These properties vary according to the transformation created. For example, if an Expression Transformation is created, an expression for one or more of the transformation output ports is to be entered. If a Stored Procedure Transformation is created, identify the stored procedure to be called.

Choose Repository | Save.

## Review Question

- Question 1: The \_\_\_\_\_ transformation tests input data for multiple conditions
- Question 2: The \_\_\_\_\_ port generates numeric values in a Sequence Generator Transformation
- Question 3: A Stored Procedure Transformation is an Active transformation.
  - True/False
- Question 4: \_\_\_\_\_ is the reverse of a router transformation
- Question 5: Reusable transformation can be configured in the Mapping Designer.
  - True/False

Copyright © Capgemini 2010. All Rights Reserved 30

## **Informatica PowerCenter**

Lesson 8: Workflow Tasks

## Lesson Objectives

- In this lesson you will learn about:
  - Command Task
  - Email Task
  - Timer Task
  - Assignment Task
  - Control Task
  - Event Tasks
  - Decision Task
  - Worklets
  - Version Control Overview

Copyright © Capgemini 2010. All Rights Reserved.

8.1. Command Task

## Description

- A Command Task is used to specify commands to run during the Workflow
- It can be used in two ways:
  - Standalone Command Task
  - Pre- or post-session command
- It can be used to:
  - Delete reject files
  - Copy files
  - Archive target files
- Command Tasks created in the Task Developer are reusable



Figure 8.1

Copyright © Capgemini 2010. All Rights Reserved.

The Command Task can be used in two ways:

Standalone Command Task – Command Task can be used anywhere in the Workflow or Worklet to run shell commands

Pre- and post-session shell command – Command Task can be called as the pre- or post- session shell command for a Session Task

Any valid UNIX command or shell script can be used for UNIX servers, or any valid DOS command or batch file for Windows servers.

The status of the command (success or failure) is stored in the pre-defined variable @command\_Task\_name.STATUS.

For example, a command Task may be used to copy a file from one directory to another. For a Windows Server the following DOS command will be used to copy a file SALES.DAT from the source directory, C:\SALES to the target directory E:\MARKETING

COPY C:\SALES\SALES.DAT E:\MARKETING

For a UNIX server, the following command will be used to perform a similar operation:

Cp sales/sales.dat /marketing

Each command runs in the same environment (UNIX or Windows) as the Integration service.

## Demo

- Configure a Command Task



Copyright © Capgemini 2010. All Rights Reserved.

### Steps to create a Command Task:

In the Workflow Designer or the Task Developer, click the Command Task icon on the Tasks toolbar.

-or-

Choose Task | Create. Select Command Task for the Task type.

Enter a name for the Command Task. Click Create. Then click Done.

Double-click the Command Task in the workspace to open the Edit Tasks dialog box.

In the Commands tab, click the Add button to add a command.

In the Name field, enter a name for the new command.

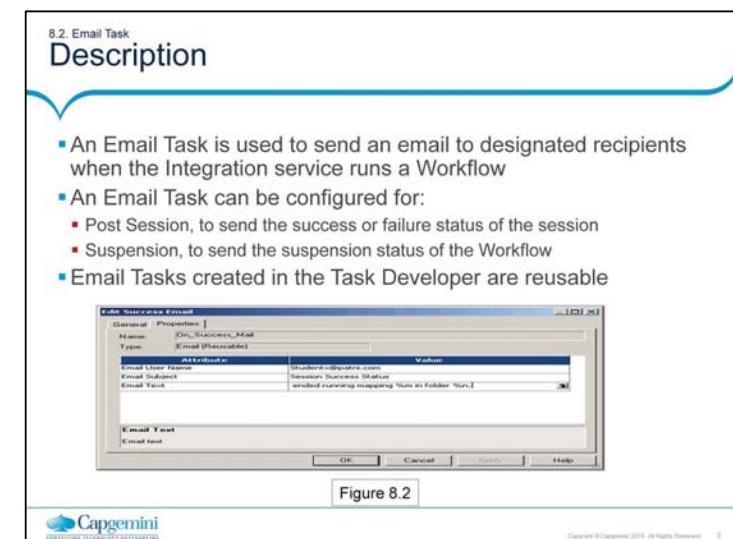
In the Command field, click the Edit button to open the Command Editor.

Enter the command which has to be performed. Enter only one command in the Command Editor.

Click OK to close the Command Editor.

Repeat steps 3-8 to add more Commands in the Task.

Click OK.



The Workflow Manager provides an Email Task to send an email during a Workflow. Re-usable Email Tasks can be created in the Task Developer for any type of email. Email Tasks created in the Workflow and Worklet Designer are non-reusable.

When a Workflow or Worklet is created the following types of email can be included:

- Post-session email. The session can be configured to send an email when the session completes or fails.
- Suspension email. The Workflow can be configured to send an email when the Workflow suspends.

For example, if the time taken for a session to complete has to be tracked, the session can be configured to send an email containing the time and date the session starts and completes.

An Email Task can also be used anywhere in a Workflow or Worklet. For example, it can be included in a Workflow after a Command Task that executes a shell script. The Workflow links can be configured for the Integration service to send an email if the Command Task fails.

8.2. Email Task

## Email Variables

Email Variable	Description
%a>	Name of file attached
%g	Session log file attached
%n	Folder name
%m	Mapping name
%d	Repository name
%t	Target table details
%s	Session name
%e	Session status
%b	Session start time
%c	Session completion time
%l	Session elapsed time
%i	Total rows loaded
%r	Total rows rejected

 Capgemini  
INNOVATING BUSINESS & TECHNOLOGY

Copyright © Capgemini 2010. All Rights Reserved.

Email variables and format tags can be used in an email message for post-session emails.

Using email variables, important session information, such as:

- Number of rows loaded
- Session completion time
- Read and write statistics, etc.

Can be included in the email. The session log or other relevant files can also be attached to the email. Format tags for a tab (lt) and newline(ln) can be used in the body of the message to make the message easier to read.

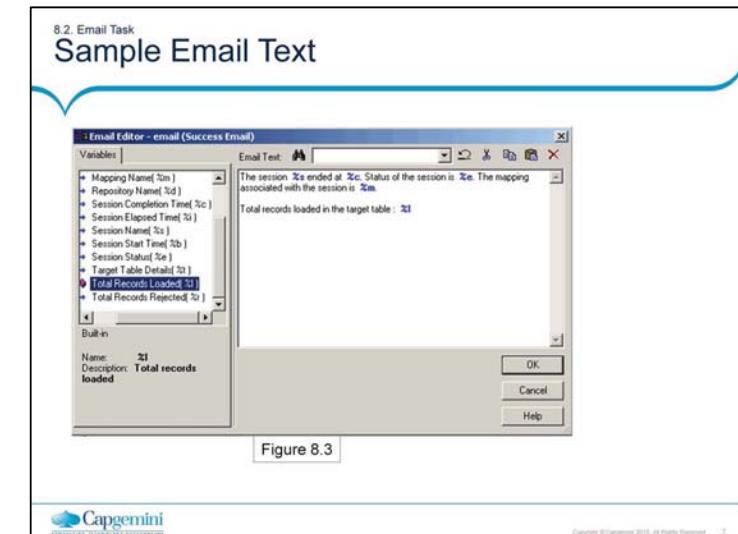


Figure 8.3

In the above figure you see a sample email text, which gives the following details  
Name of the session and mapping  
Session completion time  
Number of records loaded

## Demo

- Configure an Email Task



Capgemini  
Engineering Services for Enterprises

Steps to create an Email Task in the Task Developer:

In the Task Developer, choose Tasks | Create.

Select an Email Task and enter a name for the Task. Click Create.

Click Done.

Double-click the Email Task in the workspace. The Edit Tasks dialog box appears.

Click Rename to enter a name for the Task.

Optionally enter a description for the Task in the Description field.

Click the Properties tab.

Enter the fully qualified email address of the mail recipient in the Email User Name field.

Enter the subject of the email in the Email Subject field. Or, leave this field blank.

Click the Open button in the Email Text field to open the Email Editor.

Enter the text of the email message in the Email Editor.

When the Email Task is used for post-session email, incorporate variables and format tags in the message. Leave the Email Text field blank.

Click OK twice to save changes.

8.3. Timer Task

## Description

- A Timer Task is used to specify the period of time to wait before the Integration service executes the next Task in a Workflow
- There are two types of settings in the timer
  - Absolute time
  - Relative time



Figure 8.4

Copyright © Capgemini 2010. All Rights Reserved.

When a Timer Task is used in a Workflow, the next Task in the Workflow (after the timer Task) can be started at an exact time and date. After the start time of another Task, a wait period can be set in the Workflow, or Worklet before starting the next Task.

The Timer Task has two types of settings:

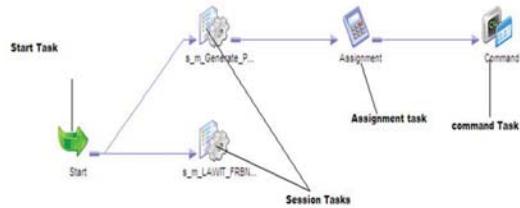
Absolute time - The exact time that the Integration service starts executing the next Task in the Workflow is specified

Relative time - The Integration service will wait for a specified period of time after the Timer Task, the parent Workflow, or the top-level Workflow starts

For example, if there are two sessions in a Workflow and the second session should start 1 minute after the first session completes; A Timer Task can be used after the first session.

## Assignment Task

- Assigns a value to a Workflow variable
- Variables are defined in the Workflow object

Copyright © Capgemini 2010. All Rights Reserved 10

## Control Task

- Used to stop, abort, or fail the top-level workflow or the parent workflow based on an input link condition.
- A parent workflow or worklet is the workflow or worklet that contains the Control task.

Copyright © Capgemini 2018. All Rights Reserved. 11

8.4. Event Tasks

## Description

- An Event is defined in a Workflow to specify the sequence of Task execution
- The event is triggered based on the completion of the sequence of Tasks
- To use events in the Workflow the Tasks are:
  - Event-Raise Task
  - Event-Wait Task
- Events that can be defined are:
  - Pre-defined event
  - User-defined event

 Capgemini  
INTEGRATING INNOVATION

Copyright © Capgemini 2010. All Rights Reserved. 13

To coordinate the execution of the Workflow, the following types of events are specified for the Event-Wait and Event-Raise Tasks:

Pre-defined event - A pre-defined event is a file-watch event. For pre-defined events, an Event-Wait Task is used to instruct the Integration service to wait for the specified indicator file to appear before continuing with the rest of the Workflow. When the Integration service locates the indicator file, it starts the next Task in the Workflow.

User-defined event - A user-defined event is a sequence of Tasks in the Workflow. An Event-Raise Task is used to specify the location of the user-defined event in the Workflow. A user-defined event is sequence of Tasks in the branch from the Start Task leading to the Event-Raise Task.

When all the Tasks in the branch from the Start Task to the Event-Raise Task complete, the Event-Raise Task triggers the event. The Event-Wait Task waits for the Event-Raise Task to trigger the event before continuing with the rest of the Tasks in its branch.



#### Event Raise Task

The Event-Raise Task represents the location of a user-defined event. A user-defined event is the sequence of Tasks in the branch from the Start Task to the Event-Raise Task. When the Integration service executes the Event-Raise Task, the Event-Raise Task triggers the user-defined event.

To use an Event Raise Task, a user-defined event is declared first. Then, an Event-Raise Task is declared in the Workflow to represent the location of the user-defined event. In the Event-Raise Task properties, the name of a user-defined event is specified.

#### Event Wait Task

The Event-Wait Task waits for an event to occur. The event can be a user-defined or a pre-defined event. Once the event triggers, the Integration service continues executing the rest of the Workflow.

If the Event Wait Task has to wait for a user-defined event then, it has to be triggered by the Event-Raise Task. The user-defined event has to be specified in this Task.

8.5. Decision Task

## Description

- A Decision Task is used to enter a condition that determines the execution of a Workflow
- A pre-defined variable, \$Decision\_Task\_name.Condition represents the result of the decision condition
- The Integration service evaluates the condition in the Decision Task and sets the pre-defined condition variable to True (1) or False (0).

```
graph LR; Start((Start)) --> dt[dt_test]; dt --> C1[cmd_1]; dt --> C2[cmd_2];
```

Figure 8.5

Copyright © Capgemini 2010. All rights reserved.

One decision condition can be specified per Decision Task. After the Integration service evaluates the Decision Task, the pre-defined condition variable can be used in other expressions in the Workflow to develop the Workflow.

You can use the Decision task instead of multiple link conditions in a workflow. Instead of specifying multiple link conditions, use the pre-defined Condition variable in a Decision task to simplify link conditions.

The Decision Task simplifies the Workflow. If a condition is not specified in the Decision Task, the Integration service evaluates the Decision Task to True.

8.6. Worklets

## Description

- A Worklet is a set of reusable Workflow logic
- The Worklet Designer is used to create and edit Worklets
- It can contain any Task available in the Workflow Manager
- Worklets can be run inside a Workflow
- The Workflow that contains the Worklet is called the parent Workflow
- A Worklet can be nested in another Worklet

```
graph LR; Start((Start)) --> Node1["wl_employee_n..."]; Node1 --> cmd["cmd"];
```

Figure 8.6

Copyright © Capgemini 2010. All Rights Reserved.

A Worklet is an object that represents a set of Tasks. When the Integration service executes a Worklet, it expands the Worklet. The Integration service then runs the Worklet as it would run any other Workflow, executing Tasks and evaluating links in the Worklet.

The Worklet does not contain any scheduling or server information. To execute a Worklet, it has to be included in a Workflow. The Workflow Manager does not provide a parameter file or log file for Worklets. The Integration service writes information about Worklet execution in the Workflow log.

8.8. Worklets

## Version Control Overview

- Version Control is PAID tool in Informatica
- Version Control is used to
  - Prevent multiple users working on same code
  - Stores older versions of code which can be retrieved if required.
  - Can compare the changes in different version of code
- How to do version control ?
  - Code version control is gets operated by check-in and check-out options of informatica.
  - Check out: Check out the object if you want to modify code. Informatica will place write-intent lock to it. No other user can edit the object after check-out.
  - Check in: Save the object after editing. After check-in your changes will be visible to others. Now object can be edited by other user. write-intent lock will be released.
  - Undo Check-out: This option will rollback all the changes. Informatica will restore latest checked-in version to repository.
- Other Version Control Tools
  - SVN
  - TFS

Figure 8.6

 Capgemini  
Engineering Services - INFORMATICA

Demo

- Creating a Worklet



Capgemini  
Engineering Services

Copyright © Capgemini 2010. All Rights Reserved.

Steps to create a Worklet:

In the Worklet Designer, choose Worklets | Create. The Create Worklets dialog box appears.

Enter a name for the Worklet.

Click OK.

The Worklet Designer creates a Start Task in the Worklet.

Now, drag the required Tasks in the designer space as per the Worklet logic.

## Summary

- After completing this lesson you now:
  - Command Task
  - Email Task
  - Assignment Task
  - Control Task
  - Timer Task
  - Decision Task
  - Event Tasks
  - Worklet
  - Version control

Copyright © Capgemini 2018. All Rights Reserved. 10

## Review Question

- Question 1: In a Workflow, an email is used to send \_\_\_\_\_ information
- Question 2: There are two kinds of events \_\_\_\_\_ and \_\_\_\_\_
- Question 3: A \_\_\_\_\_ Task is used to enter a condition that determines the execution of a Workflow
- Question 4: Worklets can be run independently.
  - True/False

Copyright © Capgemini 2010. All Rights Reserved. 10

## **Informatica PowerCenter**

Lesson 09: Pre-Post SQL ,  
SQL Override, Update  
override

## Lesson Objectives

- In this Lesson you will learn about:
  - Pre-Post SQL
  - SQL Override and Update Override



## 9.1. Pre-Post SQL

- The Integration Service runs pre-session SQL commands before it reads the source.
- It runs post-session SQL commands after it writes to the target.
- You can specify pre- and post-session SQL in the Source Qualifier transformation and the target instance when you create a mapping.



Copyright © Capgemini 2015. All Rights Reserved 3

9.1. Adding Pre- and Post -SQL in Source

## Qualifier transformation

- The Integration Service runs pre-session SQL commands against the source database before it reads the source
- It runs post-session SQL commands against the source database after it writes to the target
- Example: You might want to use pre-session SQL to write a timestamp row to the source table when a session begins.

 Capgemini  
INNOVATION THROUGH INTEGRATION

Copyright © Capgemini 2014. All Rights Reserved.

Use the following guidelines when you enter pre- and post-session SQL commands in the Source Qualifier transformation or target instance:

- You can use any command that is valid for the database type. However, the Integration Service does not allow nested comments, even though the database might.
- You can use mapping parameters and variables in the source pre- and post-session SQL commands.
- Use a semi-colon (;) to separate multiple statements.
- The Designer does not validate the SQL.

Pre- and post-session SQL commands can be entered on the Properties tab of the target instance in a mapping.

## 9.1. Adding Pre- and Post-Session SQL Commands in target

- The Integration Service runs pre-session SQL commands against the target database before it reads the source
- It runs post-session SQL commands against the target database after it writes to the target.
- Example: A Pre-session SQL can be run on the target to drop indexes before a session runs and a Post-session SQL can be run to rebuild index when session completes.

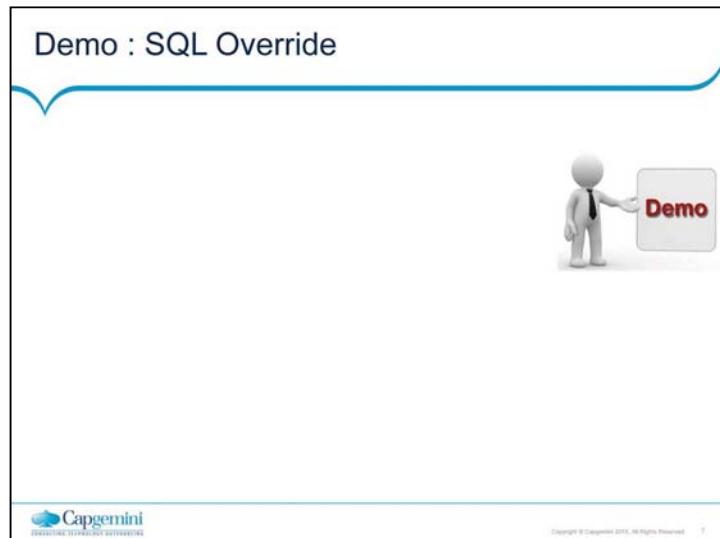
## 9.2. SQL Override

- The Source Qualifier transformation provides the SQL Query option to override the default query
- When you create an SQL Query, you can either:
  - Generate and edit the default query
  - Manually enter the entire query
- The fields of the Source Qualifier transformation have to be mapped from the source by a matching (data type and width) port

Copyright © Capgemini 2010. All Rights Reserved.

When you create an SQL Query, you can either:

- **Generate and edit the default query:** If you want to use the existing transformation options in the extract override, generate and edit the default query. When the Designer generates the default query, it incorporates all other configured options, such as a filter or number of sorted ports. The resulting query overrides all other options you might subsequently configure in the transformation
- **Manually enter the entire query:** The resulting query overrides all other options configured in the transformation

**Steps:**

1. Open the **Source Qualifier transformation**, and click the Properties tab.
2. Click the Open button in the SQL Query field. The SQL Editor dialog box appears.
3. Click **Generate SQL**.

The Designer displays the default query it generates when querying rows from all sources included in the Source Qualifier transformation.

4. Enter your own query in the space where the default query appears.  
Every column name must be qualified by the name of the table, view, or synonym in which it appears. For example, if you want to include the ORDER\_ID column from the ORDERS table, enter ORDERS.ORDER\_ID. You can double-click column names appearing in the Ports window to avoid typing the name of every column.
5. Select the ODBC data source containing the sources included in the query.
6. Enter the user name and password to connect to this database.

**7. Click *Validate*.**

The Designer runs the query and reports whether its syntax was correct.

9. Choose **Repository-Save**.

### 9.3. Target Update Override

- By default, the Integration Service updates target tables based on key values
- However, you can override the default UPDATE statement for each target in a mapping
- You might want to update the target based on non-key columns.
- Because the target ports must match the target column names, the update statement includes the keyword :TU to specify the ports in the target transformation.

Contd..



Copyright © Capgemini 2014. All Rights Reserved.

## Target Update Override(Example)

- If you modify the UPDATE portion of the statement, be sure to use :TU to specify ports
- You can override the WHERE clause to include non-key columns.
- For example, you might want to update records for employees named Mike Smith only.
- To do this, you edit the WHERE clause as follows:

```
UPDATE T_SALES SET DATE_SHIPPED = :TU.DATE_SHIPPED,  
TOTAL_SALES = :TU.TOTAL_SALES WHERE :TU.EMP_NAME = EMP_NAME and  
EMP_NAME = 'MIKE SMITH'
```



Copyright © Capgemini 2008. All Rights Reserved.

### General Rules:

Use the following rules and guidelines when you enter target update queries:

- When you save a mapping, the Designer verifies that you have referenced valid port names. It does not verify the accuracy of the SQL.
- A WHERE clause that does not contain any column references updates all rows in the target table, or no rows in the target table, depending on the WHERE clause and the data from the mapping.

For example, the following query sets the EMP\_NAME to 'MIKE SMITH' for all rows in the target table if any row of the transformation has EMP\_ID > 100.

UPDATE T\_SALES set EMP\_NAME = 'MIKE SMITH' WHERE :TU.EMP\_ID > 100

- If the WHERE clause contains no port references, the mapping updates the same set of rows for each row of the mapping.

For example, the following query updates all employees with EMP\_ID > 100 to have the EMP\_NAME from the last row in the mapping.

- UPDATE T\_SALES set EMP\_NAME = :TU.EMP\_NAME WHERE EMP\_ID > 100

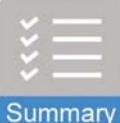
## Demo : Update Override



Copyright © Capgemini 2010. All Rights Reserved.

## Summary

- After this Lesson you now have knowledge of
  - Pre-Post SQL
  - SQL Override and Update Override



Summary



Copyright © Capgemini 2010. All Rights Reserved. 11

## Review Question

- Question 1: The Integration Service runs pre-session SQL commands against the \_\_\_\_\_ database before it reads the source
- Question 2: \_\_\_\_\_ is used to update the target based on non-key columns.



## **Informatica PowerCenter**

Lesson 10: Indirect Files

## Lesson Objectives

- In this Lesson you will learn about:
  - Using File List
  - Indirect files



## 10.1. Using File List

- You can create a session to run multiple source files for one source instance in the mapping
- You might use this feature if, for example, your company collects data at several locations which you then want to move through the same session
- When you create a mapping to use multiple source files for one source instance, the properties of all files must exactly match the source definition

Copyright © Capgemini 2010. All Rights Reserved.

To use multiple source files, you create a file containing the names and directories of each source file you want the Integration service to use. This file is referred to as a file list.

When you configure the session properties, enter the file name of the file list in the Source Filename field and enter the location of the file list in the Source File Directory field. When the session starts, the Integration service reads the file list, then locates and reads the first file source in the list. After the Integration service reads the first file, it locates and reads the next file in the list.

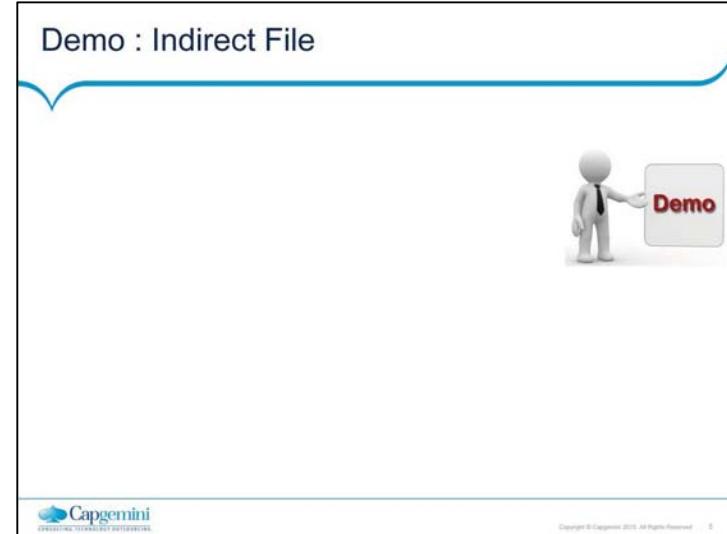
The Integration service writes the path and name of the file list to the session log. If the Integration service encounters an error while accessing a source file, it logs the error in the session log and stops the session.

## 10.2. Indirect Files

- Indirect file is used when the source file contains a list of filenames and not the data
- It enables you to configure multiple file sources by using a file list
- When you select Indirect, the Integration service finds the file list and reads each listed file when it runs the session

Copyright © Capgemini 2012. All Rights Reserved.

The source file type option in the Session properties enables you to select whether the source file is direct or Indirect. The Direct or Indirect option indicates whether the source file contains the source data, or whether it contains a list of files with the same file properties. Choose Direct if the source file contains the source data. Choose Indirect if the source file contains a list of files.



In the Workflow Manager, open the session properties.  
Click the Mapping tab and open the Transformations view.  
Click the Properties settings in the Sources node.  
Select the flat file source instance in the Instances field  
In the Source File type field, choose Indirect.  
In the Source Filename field, replace the file name with the name of the file list.  
Click OK.

## Summary

- In this Lesson we have learnt about
  - Using Multiple Source File
  - Indirect Files



Summary



Copyright © Capgemini 2014. All Rights Reserved.

## Review Question

- Question 1: \_\_\_\_\_ is used when the source file contains a list of filenames and not the data.
- Question 2: A file containing the names and directories of each source file is termed as \_\_\_\_\_



## **Informatica PowerCenter**

Lesson 11: Mapping  
Parameter And Mapping  
Variable

## Lesson Objectives

- In this Lesson you will learn about:
  - Mapping Parameter and Variable Overview
  - Mapping Parameters
  - Mapping Variables



11.1:Mapping Parameters and Variables

## Overview

- In the Designer, you can use mapping parameters and variables to make mappings more flexible
- Mapping parameters and variables represent values in mappings and mapplets
- If you declare mapping parameters and variables in a mapping, you can reuse a mapping by altering the parameter and variable values of the mapping in the session
- This can reduce the overhead of creating multiple mappings when only certain attributes of a mapping need to be changed

 Capgemini  
CONSULTING SERVICES FOR BUSINESS

Copyright © Capgemini 2012. All Rights Reserved.

11.1: Mapping Parameters and Variables

## Overview

- You can create mapping parameters and variables in the Mapping Designer or Maplet Designer.
- Once created, mapping parameters and variables appear on the Variables tab of the Expression Editor.
- You can use them in any expression in the mapplet or mapping.
- The Designer validates mapping parameters and variables in the Expression Editor of mapplets and mappings.

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2010. All Rights Reserved. 4

You can use mapping parameters and variables in a source qualifier in a mapplet or mapping. When you use mapping parameters and variables in a Source Qualifier transformation, the Designer expands them before passing the query to the source database for validation. This allows the source database to validate the query.

You cannot use mapping parameters and variables interchangeably between a mapplet and a mapping. Mapping parameters and variables declared for a mapping cannot be used within a mapplet. Similarly, you cannot use a mapping parameter or variable declared for a mapplet in a mapping.

11.1: Mapping Parameters and Variables

## Initial And Default Values

- When you declare a mapping parameter or variable in a mapping or a mapplet, you can enter an initial value.
- The Integration Service uses the configured initial value for a mapping parameter when the parameter value is not defined in the parameter file.
- Similarly, the Integration Service uses the configured initial value for a mapping variable when the variable value is not defined in the parameter file, and there is no saved variable value in the repository

 Capgemini  
PROVIDING INTEGRATED SERVICES

Copyright © Capgemini 2010. All Rights Reserved. 5

When the Integration Service needs an initial value, and you did not declare an initial value for the parameter or variable, the Integration Service uses a default value based on the data type of the parameter or variable.

The following table lists default Values for Mapping Parameters and Variables Based on Data type

Data	Default Value
String	Empty string
Numeric	0
Datetime	1/1/1753 A.D. or 1/1/1 * when Integration Service is configured for compatibility with 4.0.

11.2 Mapping Parameter  
**Information**

- A mapping parameter represents a constant value that you can define before running a session.
- A mapping parameter retains the same value throughout the entire session.
- When you use a mapping parameter, you declare and use the parameter in a mapping or mapplet.
- Then define the value of the parameter in a parameter file.
- During the session, the Integration Service evaluates all references to the parameter to that value.

 Capgemini  
CONSULTING SERVICES & SOFTWARE ENGINEERING

Copyright © Capgemini 2010. All Rights Reserved. 5

For example, you want to use the same session to extract transaction records for each of your customers individually. Instead of creating a separate mapping for each customer account, you can create a mapping parameter to represent a single customer account. Then you can use the parameter in a source filter to extract only data for that customer account. Before running the session, you enter the value of the parameter in the parameter file.

To reuse the same mapping to extract records for other customer accounts, you can enter a new value for the parameter in the parameter file and run the session.

11.2 Mapping Parameter  
**Information**

- In the Designer, you can create a mapping parameter in a mapplet or mapping.
- After you create a parameter, it appears in the Expression Editor.
- You can then use the parameter in any expression in the mapplet or mapping.
- You can also use parameters in a source qualifier filter, user-defined join, or extract override, and in the Expression Editor of reusable transformations.

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2012. All Rights Reserved.

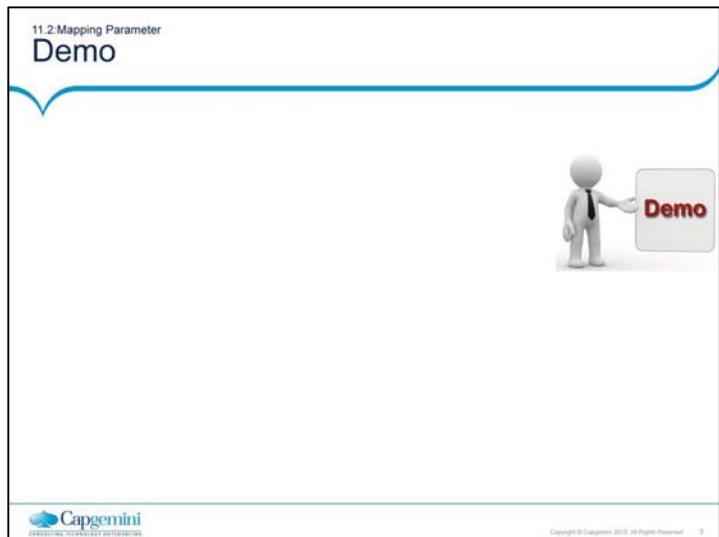
Before you run a session, define the mapping parameter value in a parameter file for the session. You can use any constant value. During the session, the Integration Service evaluates all references to the parameter to the specified value. If the parameter is not defined in the parameter file, the Integration Service uses the user-defined initial value for the parameter.

11.2 Mapping Parameter  
**Information**

- When you want to use the same value for a mapping parameter each time you run the session, use the same parameter file for each session run.
- When you want to change the value of a mapping
  - Update the parameter file between sessions.
  - Create a different parameter file and configure the session to use the new file.
  - Remove the parameter file from the session properties so that the session uses the configured initial value of the parameter in the mapping.

 Capgemini  
CONSULTING SERVICES FOR BUSINESS

Copyright © Capgemini 2014. All Rights Reserved. 5



To use a mapping parameter, perform the following steps:

1. Create a mapping parameter
2. Use the parameter
3. Define the parameter val

#### Step 1. Create a Mapping Parameter

In the Mapping Designer, choose Mappings-Parameters and Variables. Or to create parameters for a mapplet, in the Mapplet Designer, choose Mapplet-Parameters and Variables.

Click the Add button

Enter the Name of the Mapping parameter, select the type as Parameter, select the appropriate datatype and precision for the Mapping parameter.

#### Step 2. Use a Mapping Parameter

After you create a parameter, you can use it in the Expression Editor of any transformation in a mapping or mapplet. You can also use it in Source Qualifier transformations and reusable transformations.

#### Step 3. Define a Parameter Value

Before you run a session, define values for mapping parameters in the parameter file. When you do not define a parameter value, the Integration Service uses the initial value for the parameter. If the initial value is not defined, the Integration Service uses the default value for the parameter data type.

Lab



Copyright © Capgemini 2012. All Rights Reserved. 10

11.3 Mapping Variable

## Information

- Unlike a mapping parameter, a mapping variable represents a value that can change through the session.
- The Integration Service saves the value of a mapping variable to the repository at the end of each successful session run and uses that value the next time you run the session.
- When you use a mapping variable, you declare the variable in the mapping or mapplet, and then use a variable function in the mapping to change the value of the variable.

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2012. All Rights Reserved. 11

Mapping variables can be used to perform incremental reads of a source. For example, suppose the customer accounts in the mapping parameter example above are numbered from 001 to 065, incremented by one. Instead of creating a mapping parameter, you can create a mapping variable with an initial value of 001. In the mapping, use a variable function to increase the variable value by one. The first time the Integration Service runs the session, it extracts the records for customer account 001. At the end of the session, it increments the variable by one and saves that value to the repository. The next time the Integration Service runs the session, it extracts the data for the next customer account, 002. It also increments the variable value so the next session extracts and looks up data for customer account 003.

11.3 Mapping Variable

## Information

- At the beginning of a session, the Integration Service evaluates references to a variable to its start value
- At the end of a successful session, the Integration Service saves the final value of the variable to the repository
- The next time you run the session, the Integration Service evaluates references to the variable to the saved value

 Capgemini

Copyright © Capgemini 2014. All Rights Reserved. 10

11.3 Mapping Variable

## Information

- In the Designer, you can create mapping variables in a mapping or mapplet
- After you create a mapping variable, it appears in the Expression Editor
- You can then use it in any expression in the mapping or mapplet
- You can also use mapping variables in a source qualifier filter, user-defined join, or extract override, and in the Expression Editor of reusable transformations

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2010. All Rights Reserved.

You might use a mapping variable to perform an incremental read of the source. For example, you have a source table containing timestamped transactions and you want to evaluate the transactions on a daily basis. Instead of manually entering a session override to filter source data each time you run the session, you can create a mapping variable, \$\$IncludeDateTime. In the source qualifier, create a filter to read only rows whose transaction date equals \$\$IncludeDateTime, such as:

`TIMESTAMP = $$IncludeDateTime`

In the mapping, you can use a variable function to set the variable value to increment one day each time the session runs. If you set the initial value of \$\$IncludeDateTime to 9/1/2000, the first time the Integration Service runs the session, it reads only rows dated 9/1/2000. During the session, the Integration Service sets \$\$IncludeDateTime to 9/2/2000. It saves 9/2/2000 to the repository at the end of the session. The next time it runs the session, it reads only rows from September 2, 2000.

11.3 Mapping Variable

## Variable Values

- The Integration Service holds two different values for a mapping variable during a session run:
  - Start value of a mapping variable
  - Current value of a mapping variable

 Capgemini  
CONSULTING SERVICES & SOFTWARE ENGINEERING

Copyright © Capgemini 2012. All Rights Reserved. 10

### Start Value

The start value is the value of the variable at the start of the session. The Integration Service looks for the start value in the following order:

Value in parameter file

Value in pre-session variable assignment

Value saved in the repository

Initial value

Default value

For example, you create a mapping variable in a mapping or mapplet and enter an initial value, but you do not define a value for the variable in a parameter file. The first time the Integration Service runs the session, it evaluates the start value of the variable to the configured initial value. The next time the session runs, the Integration Service evaluates the start value of the variable to the value saved in the repository.

### Current Value

The current value is the value of the variable as the session progresses. When a session starts, the current value of a variable is the same as the start value. As the session progresses, the Integration Service calculates the current value using a variable function that you set for the variable. The current value can change as the Integration Service evaluates the current value of a variable as each row passes through the mapping. The final current value for a variable is saved to the repository at the end of a successful session. When a session fails to complete, the Integration Service does not update the value of the variable in the repository.

11.3 Mapping Variable

## Variable Datatype And Aggregation Type

- When you declare a mapping variable in a mapping, you need to configure the datatype and aggregation type for the variable.
- You can create a variable with the following aggregation types:
  - Count
  - Max
  - Min

 Capgemini  
PROVIDING SERVICES FOR BUSINESS

Copyright © Capgemini 2010. All Rights Reserved. 10

You can configure a mapping variable for a Count aggregation type when it is an Integer or Small Integer. You can configure mapping variables of any datatype for Max or Min aggregation types.

To keep the variable value consistent throughout the session run, the Designer limits the variable functions you can use with a variable based on aggregation type. For example, you can use the SetMaxVariable function for a variable with a Max aggregation type, but not with a variable with a Min aggregation type.

11.3 Mapping Variable

## Variable Functions

- Variable functions determine how the Integration Service calculates the current value of a mapping variable in a pipeline.
- You can use variable functions in an expression to set the value of a mapping variable for the next session run.
- The transformation language provides the following variable functions to use in a mapping:
  - SetMaxVariable
  - SetMinVariable
  - SetCountVariable

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2010. All Rights Reserved. 10

**SetMaxVariable.** Sets the variable to the maximum value of a group of values. To use the SetMaxVariable with a mapping variable, the aggregation type of the mapping variable must be set to Max.

**SetMinVariable.** Sets the variable to the minimum value of a group of values. To use the SetMinVariable with a mapping variable, the aggregation type of the mapping variable must be set to Min.

**SetCountVariable.** Increments the variable value by one. In other words, it adds one to the variable value when a row is marked for insertion, and subtracts one when the row is marked for deletion. To use the SetCountVariable with a mapping variable, the aggregation type of the mapping variable must be set to Count.

## Summary

- This Lesson gives knowledge of:
  - Mapping Parameter and Mapping Variable Overview
  - Mapping Parameter
  - Mapping Variable

Copyright © Capgemini 2014. All Rights Reserved. 11

## Review Question

- Question 1: A \_\_\_\_\_ represents a constant value that you can define before running a session.
- Question 2: A \_\_\_\_\_ represents a value that can change through the session.



Informatica PowerCenter

FTP connection

## **Informatica PowerCenter**

Lesson 12: FTP connection

## Lesson Objectives

- In this Lesson you will learn about
  - Using an FTP connection



12.1: Using FTP

## Overview

- The Integration Service can use File Transfer Protocol (FTP) to access source and target files.
- With both source and target files, you can use FTP to transfer the files directly to the Integration Service.
- When using FTP file sources and targets in a session, you should know the following information:
  - FTP connection name
  - Remote file name and exact path

 Capgemini  
CONSULTING SERVICES FOR BUSINESS

Copyright © Capgemini 2014. All Rights Reserved. 5

12.1: Using FTP

## Creating An FTP Connection

- The Integration Service can access source and target files on remote machines using FTP.
- Before you create a session using FTP, you must configure the FTP connection in the Workflow Manager.
- You must know the following information when you create an FTP connection:
  - Connection name
  - Host name
  - Default remote directory

 Capgemini  
Engineering Services for Businesses

Copyright © Capgemini 2010. All Rights Reserved. 4

Connection name: The connection name used by the Workflow Manager.

Host name: The name or IP address of the remote machine. Optionally, you can specify a port number between 1 and 65535 inclusive. If you do not specify a port number, the Integration Service uses the port number 21 by default. Use the following syntax for specifying a host name: hostname:port-number

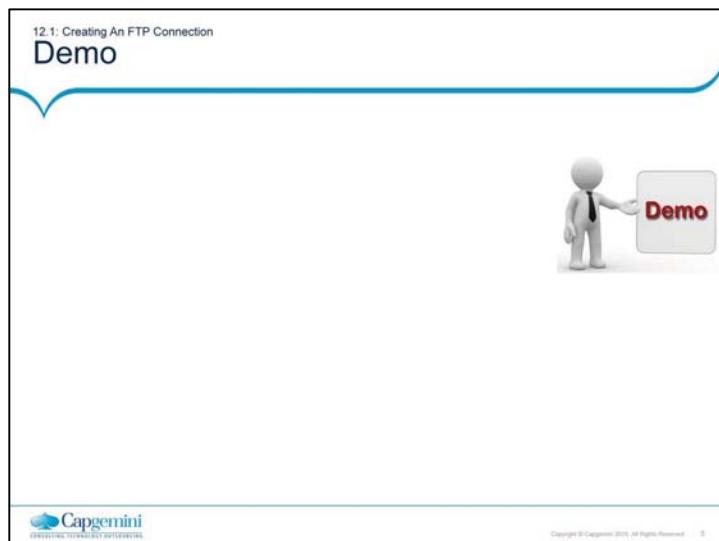
or

IP address:port-number

When you specify a port number, enable that port number for FTP on the host machine

Default remote directory: The directory you want the Integration Service to use by default.

In the session, when you enter a file name without a directory, the Integration Service appends the file name to this directory. Therefore, this path must be exact and contain the appropriate trailing delimiters. For example, if you enter c:\data\ and in the session specify the file FILENAME, the Integration Service reads the path and file name as c:\data\FILENAME



To create an FTP connection:

In the Workflow Manager, connect to a repository.

Choose Connections-FTP. The FTP Object Browser appears

Click New.

Enter the FTP Connection name, username, password, hostname and Default Remote Directory.

Click OK.

12.1: Using FTP

## Creating An FTP Connection

- After defining FTP connections in the Workflow Manager, you can create sessions using FTP file sources and targets
- You can use any mapping with the flat file sources or targets
- The steps to create FTP sessions vary for source and target files
- You can use FTP to access both source and target files in a session

 Capgemini  
CONSULTING SERVICES FOR BUSINESS

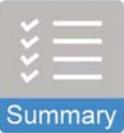
Copyright © Capgemini 2012. All Rights Reserved.

## Summary

- After completing this lesson you now:
- Know Using FTP



Copyright © Capgemini 2014. All Rights Reserved.



## **Informatica PowerCenter**

Lesson 13: Dynamic Lookup

## Lesson Objectives

- In this Lesson you will learn about:
  - Static Cache
  - Dynamic Cache



13.1: Static Cache

## Uncached Lookup Or Static Cache

- By default, the Integration Service creates a static lookup cache when you configure a Lookup transformation for caching.
- The Integration Service builds the cache when it processes the first lookup request.
- It queries the cache based on the lookup condition for each row that passes into the transformation.
- The Integration Service processes an uncached lookup the same way it processes a cached lookup except that it queries the lookup source instead of building and querying the cache.

 Capgemini  
CONSULTING SERVICES FOR BUSINESS

Copyright © Capgemini 2012. All Rights Reserved. 3

13.1: Static Cache

## Uncached Lookup Or Static Cache

- When the lookup condition is TRUE
  - The Integration Service returns the values from the lookup source or cache
  - For connected Lookup transformations, the Integration Service returns the values represented by the lookup/output ports
  - For unconnected Lookup transformations, the Integration Service returns the value represented by the return port
- When the look up condition is NOT TRUE
  - The Integration Service returns either NULL or default values
  - For connected Lookup transformations, the Integration Service returns the default value of the output port when the condition is not met
  - For unconnected Lookup transformations, the Integration Service returns NULL when the condition is not met

 Capgemini  
CONSULTING SERVICES & TECHNOLOGIES

Copyright © Capgemini 2012. All Rights Reserved.

13.2: Dynamic Cache

## Dynamic Lookup Cache

- For relational lookups, you might want to configure the transformation to use a dynamic cache when the target table is also the lookup table.
- When you use a dynamic cache, the Integration Service updates the lookup cache as it passes rows to the target.
- The Integration Service builds the cache when it processes the first lookup request.
- It queries the cache based on the lookup condition for each row that passes into the transformation.

 Capgemini  
CONSULTING SERVICES FOR BUSINESS

Copyright © Capgemini 2012. All Rights Reserved. 5

13.2: Dynamic Cache

## Dynamic Lookup Cache

- When the Integration Service reads a row from the source, it updates the lookup cache by performing one of the following actions:
  - Inserts the row into the cache
  - Updates the row in the cache
  - Makes no change to the cache
- The Integration Service either inserts or updates the cache or makes no change to the cache, based on the results of the lookup query, the row type, and the Lookup transformation properties you define.

 Capgemini  
PROVIDING INNOVATION FOR BUSINESS

Copyright © Capgemini 2010. All Rights Reserved. 5

When the Integration Service reads a row from the source, it updates the lookup cache by performing one of the following actions:

Inserts the row into the cache. The row is not in the cache and you specified to insert rows into the cache. You can configure the transformation to insert rows into the cache based on input ports or generated sequence IDs. The Integration Service flags the row as insert.

Updates the row in the cache. The row exists in the cache and you specified to update rows in the cache. The Integration Service flags the row as update. The Integration Service updates the row in the cache based on the input ports.

Makes no change to the cache. The row exists in the cache and you specified to insert new rows only. Or, the row is not in the cache and you specified to update existing rows only. Or, the row is in the cache, but based on the lookup condition, nothing changes. The Integration Service flags the row as unchanged.

13.2: Dynamic Cache

## Dynamic Lookup Cache

- A Lookup transformation using a dynamic cache has the following properties:

Port Name	Datatype	Prec	Scale	I	O	L	R	Associated Port	L... Ignore in Co...
NewLookupPort	Integer	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Associated Sequence ID	<input type="checkbox"/>
ACC_KEY	Integer	10	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Acc_No1	<input type="checkbox"/>
ACC_NO	decimal	10	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Current_Balance	<input type="checkbox"/>
BALANCE	decimal	10	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Acc_No1	<input type="checkbox"/>
Acc_No1	decimal	10	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NewLookupPort	<input type="checkbox"/>
Current_Balance	decimal	10	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NewLookupPort	<input type="checkbox"/>

Figure 13.1

Copyright © Capgemini 2010. All Rights Reserved.

NewLookupRow. The Designer adds this port to a Lookup transformation configured to use a dynamic cache. Indicates with a numeric value whether the Integration Service inserts or updates the row in the cache, or makes no change to the cache. To keep the lookup cache and the target table synchronized, you pass rows to the target when the NewLookupRow value is equal to 1 or 2.

Associated Port. Associate lookup ports with either an input/output port or a sequence ID. The Integration Service uses the data in the associated ports to insert or update rows in the lookup cache. If you associate a sequence ID, the Integration Service generates a primary key for inserted rows in the lookup cache.

Ignore Null Inputs for Updates. The Designer activates this port property for lookup/output ports when you configure the Lookup transformation to use a dynamic cache. Select this property when you do not want the Integration Service to update the column in the cache when the data in this column contains a null value.

Ignore in Comparison. The Designer activates this port property for lookup/output ports not used in the lookup condition when you configure the Lookup transformation to use a dynamic cache. The Integration Service compares the values in all lookup ports with the values in their associated input ports by default. Select this property if you want the Integration Service to ignore the port when it compares values before updating a row.

13.2: Dynamic Cache

## Using New Lookup Row Port

- When you define a Lookup transformation to use a dynamic cache, the Designer adds the New Lookup Row port to the transformation
- The Integration Service assigns a value to the port, depending on the action it performs to the lookup cache
- New Look UP Row values

NewLookupRow Value	Description
0	The PowerCenter Server does not update or insert the row in the cache.
1	The PowerCenter Server inserts the row into the cache.
2	The PowerCenter Server updates the row in the cache.

 Capgemini  
PROVIDING INNOVATION FOR BUSINESS

Copyright © Capgemini 2010. All Rights Reserved. 5

When the Integration Service reads a row, it changes the lookup cache depending on the results of the lookup query and the Lookup transformation properties you define. It assigns the value 0, 1, or 2 to the NewLookupRow port to indicate if it inserts or updates the row in the cache, or makes no change.

## Summary

- This lesson has taught basically two parts:
  - Static Cache
  - Dynamic Cache



Summary



Copyright © Capgemini 2012. All Rights Reserved. 9

## Review Question

- Question 1: If a Lookup transformation is configured to use a dynamic cache, the Designer adds the \_\_\_\_\_ port to the transformation.
  
- Question 2: The Integration Service uses the data in the \_\_\_\_\_ ports to insert or update rows in the lookup cache.



## **Informatica PowerCenter**

Lesson 14: Creating a Type  
2 Dimension/Version Data  
Mapping

## Lesson Objectives

- In this Lesson you will learn about:
  - Type 1 & Type 2 Dimension
  - Using a Mapping Wizard to create a Type 2 Dimension mapping

Copyright © Capgemini 2012. All Rights Reserved.

## 14.1. Introduction

- Dimensions contains Textual attributes.
- The Type 1 Dimension mapping - Loads a slowly changing dimension table by inserting new dimensions and overwriting existing dimensions.
- Ie. Rows containing changes to existing dimensions are updated in the target by overwriting the existing dimension.
- In the Type 1 Dimension mapping, all rows contain current dimension data and historical data is lost.



Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

### 14.1. Type 1 - Example

- For example, we have a fact table, FACT\_TRANS, containing the sales data for various snack and beverage products and a dimension table, LU\_PRODUCT, containing the product descriptions. Over time, however, one of these product descriptions as recorded in the transaction log, TRANS\_LOG.TXT, changed, resulting in an SCD that must be processed.

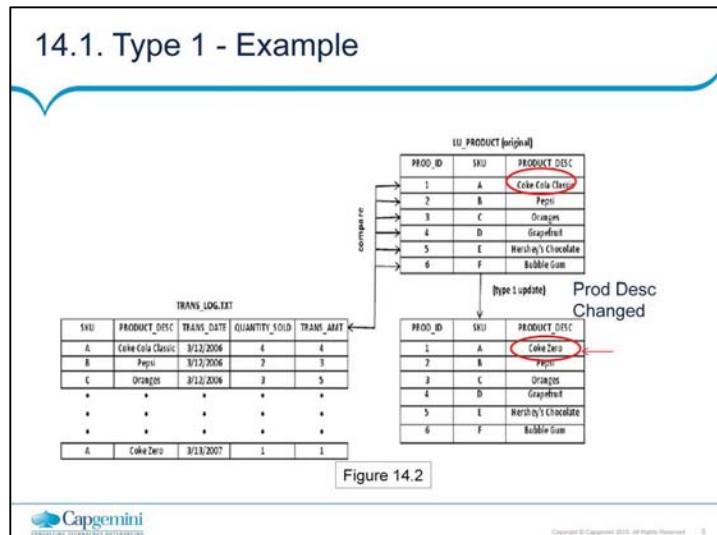
TRANS_LOG.TXT				
SKU	PRODUCT_DESC	TRANS_DATE	QUANTITY SOLD	TRANS_AMT
A	Coke Cola Classic	3/12/2006	4	4
B	Pepsi	3/12/2006	2	3
C	Oranges	3/12/2006	3	5
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
A	Coke Zero	3/13/2007	1	1

Figure 14.1



contd.

Copyright © Capgemini 2010. All Rights Reserved.



## 14.1. Introduction

- The Type 2 Dimension/Version Data mapping filters source rows based on user-defined comparisons and inserts both new and changed dimensions into the target.
- Changes are tracked in the target table by versioning the primary key and creating a version number for each dimension in the table. When you use this option, the Designer creates two additional fields in the target:
  - PM\_PRIMARYKEY. The Integration Service generates a primary key for each row written to the target.
  - PM\_VERSION\_NUMBER. The Integration Service generates a version number for each row written to the target

Copyright © Capgemini 2010. All Rights Reserved.

In the Type 2 Dimension/Version Data target, the current version of a dimension has the highest version number and the highest incremented primary key of the dimension.

Use the Type 2 Dimension/Version Data mapping to update a slowly changing dimension table when you want to keep a full history of dimension data in the table. Version numbers and versioned primary keys track the order of changes to each dimension.

## 14.2. Handling Keys

- In a Type 2 Dimension/Version Data mapping, the Integration Service generates a new primary key value for each new dimension it inserts into the target.
- An Expression transformation increments key values by 1,000 for new dimensions.
- When updating an existing dimension, the Integration Service increments the existing primary key by 1.



Copyright © Capgemini 2010. All Rights Reserved.

### 14.3. Example

- For example, the Integration Service inserts the following new row with a key value of 65,000 since this is the sixty-fifth dimension in the table.

PM_PRIMARYKEY	ITEM	STYLES
65000	Sandal	5

Contd...



Copyright © Capgemini 2010. All Rights Reserved.

- The next time you run the workflow containing the session, the same item has a different number of styles.

- The Integration Service creates a new row with updated style information and increases the existing key by 1 to create a new key of 65,001.

- Both rows exist in the target, but the row with the higher key version contains current dimension data.

▪ PM_PRIMARYKEY	ITEM	STYLES
65000	Sandal	5
65001	Sandal	14

Contd...



Copyright © Capgemini 2014. All Rights Reserved.

- When you run the workflow again, the Integration Service again increments the key.
- The highest key version contains current dimension data.
- The target keeps a full history of the item as well as the order in which the versions occurred.

PM_PRIMARYKEY	ITEM	STYLES
65000	Sandal	5
65001	Sandal	14
65002	Sandal	17

Copyright © Capgemini 2010. All Rights Reserved.

## 14.4. Numbering Versions

- In addition to versioning the primary key, the Integration Service generates a matching version number for each row inserted into the target.
- Version numbers correspond to the final digit in the primary key.
- New dimensions have a version number of 0.

Copyright © Capgemini 2014. All Rights Reserved. E1

For example, in the data below, the versions are 0, 1, and 2. The highest version number contains the current dimension data.

PM_PRIMARYKEY	ITEM	STYLES	PM_VERSION_NUMBER
65000	Sandal	5	0
65001	Sandal	14	1
65002	Sandal	17	2

## 14.5. Understanding the Mapping

- Selects all rows
- Caches the existing target as a lookup table
- Compares logical key columns in the source against corresponding columns in the target lookup table
- Compares source columns against corresponding target columns if key columns match
- Flags new rows and changed rows
- Creates two data flows: one for new rows, one for changed rows
- Generates a primary key and version number for new rows
- Inserts new rows to the target
- Increments the primary key and version number for changed rows
- Inserts changed rows in the target



Copyright © Capgemini 2010. All Rights Reserved.

The Type 2 Dimension/Version Data mapping uses a Lookup and an Expression transformation to compare source data against existing target data. When you step through the Slowly Changing Dimensions Wizard, you enter the lookup conditions (source key columns) and source columns that you want the Integration Service to compare against the existing target.

For each source row :

without a matching primary key in the target, the Expression transformation marks the row new.

with a matching primary key in the target, the Expression compares user-defined source and target columns. If those columns do not match, the Expression marks the row changed.

The mapping then splits into two separate data flows.

The first data flow uses the Filter transformation, FIL\_InsertNewRecord, to filter out existing rows. The Filter transformation passes only new rows to the UPD\_ForceInserts Update Strategy transformation. UPD\_ForceInserts inserts new rows to the target. A Sequence Generator creates a primary key for each row. The Expression transformation, EXP\_KeyProcessing\_InsertNew, increases the increment between keys by 1,000 and creates a version number of 0 for each new row.

In the second data flow, the FIL\_InsertChangedRecord Filter transformation allows only changed rows to pass to the Update Strategy transformation, UPD\_ChangedInserts. UPD\_ChangedInserts inserts changed rows to the target. The Expression transformation, EXP\_KeyProcessing\_InsertChanged, increments both the existing key and the existing version number by one.

## Summary

- This Lesson gives knowledge about Creating a Type 2 Dimension/Version Data Mapping



Summary



Copyright © Capgemini 2014. All Rights Reserved. 13

## Review Question

- Question 1: In a Type 2 Dimension/Version Data Mapping, new dimensions have a version number \_\_\_\_\_.
  
- Question 2: The Type 2 Dimension/Version Data \_\_\_\_\_ both new and changed dimensions into the target.
  - Option 1: Inserts
  - Option 2: Deletes
  - Option 3: Updates

Copyright © Capgemini 2014. All Rights Reserved. 14

## **Informatica PowerCenter**

Lesson 15: Commit Point

## Lesson Objectives

- In this Lesson you will learn about:

- Commit Interval
  - Source based Commit Interval
  - Target based Commit Interval



Copyright © Capgemini 2012. All Rights Reserved.

## 15.1. Introduction To Commit Point

- A commit point is the interval at which the server commits data to relational targets during a session.
- There are 3 types of commits
  - Source based Commit interval
  - Target based Commit Interval
  - User-Defined Commit Interval



Copyright © Capgemini 2010. All Rights Reserved. 5

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

## 15.2. Source based Commit Point

- Server commits data based on the number of source rows. The commit point is the commit interval you configure in the session properties.
- During a session, the server commits data to the target based on the number of rows from an active source in a single pipeline. The rows are referred to as source rows.
- A pipeline consists of a source qualifier and all the transformations and targets that receive data from source qualifier.
- Although the Filter, Router and Update Strategy transformations are active transformations, the server does not use them as active sources in a source based commit session.
- When a server runs a session, it identifies the active source for each pipeline in the mapping. The server generates a commit row from the active source at every commit interval.
- When each target in the pipeline receives the commit rows the server performs the commit.

Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

### 15.3. Target based Commit Point

- Server commits data based on the no of target rows and the key constraints on the target table. The commit point also depends on the buffer block size and the commit interval.
- During a session, the server continues to fill the writer buffer, after it reaches the commit interval. When the buffer block is full, the Informatica server issues a commit command. As a result, the amount of data committed at the commit point generally exceeds the commit interval.
- The server commits data to each target based on primary –foreign key constraints.



Copyright © Capgemini 2014. All Rights Reserved. 5

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

#### 14.4. User Defined Commit Point

- The Integration Service commits data based on transactions defined in the mapping properties .
- We can also configure some commit and rollback options in the session properties.
- Note :Source-based and user-defined commit sessions have partitioning restrictions.
- If we configure a session with multiple partitions to use source-based or user-defined commit, we have to choose pass-through partitioning at certain partition points in a pipeline.



Copyright © Capgemini 2014. All Rights Reserved. 5

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

## Summary

- This Lesson gives knowledge about commit types in Informatica



Summary



Copyright © Capgemini 2014. All Rights Reserved.

## Review Question

- Question 1: Name type of commit Interval .
- Question 2: Which is default commit interval
  - Option 1: Target Based commit interval
  - Option 2: Source Based commit interval
  - Option 3: None

Copyright © Capgemini 2014. All Rights Reserved.

## **Informatica PowerCenter**

**Lesson 16: Partitioning**

## Lesson Objectives

- In this Lesson you will learn about:
  - Introduction to Informatica Partitioning
  - Different Types Of Partitioning
    - Database partitioning
    - Hash auto-keys
    - Hash user keys
    - Key range
    - Pass-through
    - Round-robin

Copyright © Capgemini 2012. All Rights Reserved.

## 16.1. Introduction To Partitioning

- The partition type controls how the Integration Service distributes data among partitions at partition points.
- The Integration Service creates a default partition type at each partition point.
- After tuning all the performance bottlenecks we can further improve the performance by addition partitions.
- Partitions are widely divided into –
  - Dynamic Partitioning ( No. of partitions passed as parameters)
  - Non-Dynamic Partitioning ( No. of partitions fixed while coding)
- The partition attributes include setting the partition point, the number of partitions, and the partition types.
- Informatica partition are useful in situations where
  - we need to load huge volume of data or
  - when using Informatica source which already has partitions defined, and using those partitions will allow to improve the session performance.

Copyright © Capgemini 2014. All Rights Reserved.

## 16.1. Introduction To Partitioning

- Partition Point: There can be one or more pipelines inside a mapping.
- Adding a partition point will divide this pipeline into many pipeline stages.
- Informatica will create one partition by default for every pipeline stage.
- As we increase the partition points it increases the number of threads.
  - Informatica has mainly three types of threads –Reader, Writer and Transformation Thread.
- Maximum 64 partition point can be define.
  - However, creating large number of partitions or partition points in a session that processes large amounts of data, can overload the system
- Partition points cannot be created for the following transformations:
  - Source definition
  - Sequence Generator
  - XML Parser
  - XML target
  - Unconnected transformations



Copyright © Capgemini 2012. All Rights Reserved.

## 16.2. Types Of Partition

- Type of partitions are
  - Database partitioning
  - Pass-through
  - Hash auto-keys
  - Hash user keys
  - Key range
  - Round-robin
- Database Partitioning:
  - The PowerCenter Integration Service queries the IBM DB2 or Oracle system for table partition information.
  - It reads partitioned data from the corresponding nodes in the database.
  - Use database partitioning with Oracle or IBM DB2 source instances on a multi-node table space.
  - Use database partitioning with DB2 targets

Copyright © Capgemini 2014. All Rights Reserved.

## 16.2. Types Of Partition

- Pass Through Partitioning
  - This is default configuration for session.
  - The PowerCenter Integration Service passes all rows at one partition point to the next partition point without redistributing them
  - Choose pass-through partitioning where you want to create an additional pipeline stage to improve performance, but do not want to change the distribution of data across partitions
- Hash auto-keys Partitioning
  - The PowerCenter Integration Service uses all grouped or sorted ports as a compound partition key
  - You may need to use hash autokeys partitioning at Rank, Sorter, and unsorted Aggregator transformations.
- Hash user keys.
  - The PowerCenter Integration Service uses a hash function to group rows of data among partitions.
  - You define the number of ports to generate the partition key

Copyright © Capgemini 2014. All Rights Reserved.

## 16.2. Types Of Partition

- Key Range

- You specify one or more ports to form a compound partition key.
- The PowerCenter Integration Service passes data to each partition depending on the ranges you specify for each port.
- Use key range partitioning where the sources or targets in the pipeline are partitioned by key range.

- Round-robin

- The PowerCenter Integration Service distributes blocks of data to one or more partitions.
- Use round-robin partitioning so that each partition processes rows based on the number and size of the blocks



Copyright © Capgemini 2014. All Rights Reserved.

## Summary

- This Lesson gives knowledge about partitions in Informatica

Copyright © Capgemini 2014. All Rights Reserved.

## Review Question

- Question 1: Partitions are used for
  - Option 1: Performance Improvement
  - Option 2: Remove performance bottleneck
  - Option 3: None of above
  
- Question 2: Is default partition
  - Option 1: Round Robin
  - Option 2: Pass-trough
  - Option 3: Hash key

Copyright © Capgemini 2014. All Rights Reserved.

## **Informatica PowerCenter**

Lesson 17: Session  
Parameters

## Lesson Objectives

- In this Lesson you will learn about:
  - Types of Session Parameter
  - Use Of Parameter file for setting session parameters
  - Parameter file format



Copyright © Capgemini 2012. All Rights Reserved.

## 17.1. Introduction To Session Parameters

- Session parameters, represent values you might want to change between sessions, such as a database connection or source file.
- Use session parameters in the session properties, and then define the parameters in a parameter file.
- You can specify the parameter file for the session to use in the session properties
- The Workflow Manager provides one built-in session parameter, \$PMSessionLogFile. With \$PMSessionLogFile, you can change the name of the session log generated for the session.

Naming Conventions for User-Defined Session Parameters

Parameter Type	Naming Convention
Database Connection	\$DBConnectionString
Source File	\$InputFileName
Target File	\$OutputFileName
Lookup File	\$LookupFileName
Reject File	\$BadFileName

- Use parameters file to make sessions more flexible and reusable.



Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

## 17.2 Notations Of Parameter file

```
[<folder_name>.WF:<workflow_name>.ST:<session_name>]  
$DBConnectionSource=  
$DBConnectionTarget=  
$PMSourceFileDir=  
$PMTargetFileDir=  
$PMSSessLogDir=  
$PMBadFileDir=  
$PMSSessionLogFile=  
$$SRC_SCHEMA= < These are VARIABLES AND PARAMETERS>
```

Copyright © Capgemini 2014. All Rights Reserved.

### 17.3 Configuring Parameter File

- We can configure parameter file at Workflow or session level  
GO To Workflow /Session -> Edit -> Properties -> Parameter File

Copyright © Capgemini 2014. All Rights Reserved.

## Summary

- This Lesson gives knowledge session parameters and configuring it using parameter file



Summary



Copyright © Capgemini 2012. All Rights Reserved.

## Review Question

- 1. Can we use the mapping parameters or variables , created in one mapping into another mapping ?

A.no  
B.Yes

Option A



- 2. what will happen if informatica server doesn't find session parameter and variable ?

A.Work flow will fail  
B.Automatically it will created  
C.Give a message to user to put it  
D.none of these

Option A



Copyright © Capgemini 2012. All Rights Reserved.

## **Informatica PowerCenter**

Lesson 18: Incremental  
Aggregation

## Lesson Objectives

- In this Lesson you will learn about:
  - What is incremental aggregation ?
  - Understanding Incremental Aggregation Process
  - Implementation – Sample mapping



## 18.1. Introduction To Incremental Aggregation

- Using incremental aggregation, we apply captured changes in the source data (CDC part) to aggregate calculations in a session.
- If the source changes incrementally and we can capture the changes, then we can configure the session to process those changes
- This allows the Integration Service to update the target incrementally, rather than forcing it to delete previous loads data, process the entire source data and recalculate the same data each time you run the session.
- INCREMENTAL AGGREGATION, may be helpful in cases when we need to load data in monthly facts in a weekly basis.
- Note : Applicable if CDC ( Change data capture) is implemented
  - Use incremental aggregation when the changes do not significantly change the target



Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

## 18.2. Understanding Incremental Aggregation Process

- The first time we run an incremental aggregation session, the Integration Service processes the entire source.
- At the end of the session, the Integration Service stores aggregate data for that session run in two files, the index file and the data file in the cache directory specified in the Aggregator transformation properties.
- For next run Integration Service check if the CDC data from the source is existing in the index file and does the incremental aggregation and updates the index and data file accordingly.
- Each subsequent time we run a session with incremental aggregation, the Integration Service creates a backup of the incremental aggregation files.
- The cache directory for the Aggregator transformation must contain enough disk space for two sets of the files.
- The Integration Service creates new aggregate data, instead of using historical data, when we configure the session to reinitialize the aggregate cache, Delete cache files etc.
- When the Integration Service rebuilds incremental aggregation files, the data in the previous files is lost.
- Note: To protect the incremental aggregation files from file corruption or disk failure, periodically back up the files.



Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

### 18.3. Implementation of Incremental Aggregation

- Sample Mapping



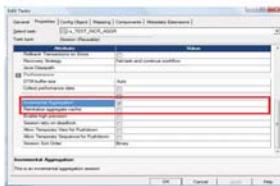
- CDC Source : Source query should fetch only the changed record .  
In below case , Batch Load Control table is used



Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

### 18.3. Implementation of Incremental Aggregation

- Expression and Aggregate transform can have the desired logic
- Session Configuration : Click check box of Incremental aggregate



- If we want to reinitialize the aggregate cache suppose during first week of every month we will configure the same session in a new workflow at workflow level with the Reinitialize aggregate cache property checked in.



Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

### 18.3. Implementation of Incremental Aggregation

- Example with Data :
- Source data table

CUSTOMER_KEY	INVOICE_KEY	AMOUNT	LOAD_DATE
1111	5001	100	01/01/2010
2222	5002	250	01/01/2010
3333	5003	300	01/01/2010
1111	6007	200	07/01/2010
1111	6008	150	07/01/2010
2222	6009	250	07/01/2010
4444	1234	350	07/01/2010
5555	6157	500	07/01/2010

CUSTOMER_KEY	INVOICE_KEY	MON_KEY	AMOUNT
1111	5001	201001	100
2222	5002	201001	250
3333	5003	201001	300

- Target Table After 1st Weeks Load

Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

### 18.3. Implementation of Incremental Aggregation

- Now during the 2nd week load it will process only the incremental data in the source i.e those records having load date greater than the last session run date. After the 2nd weeks load after incremental aggregation of the incremental source data with the aggregate cache file data will update the target table with the following dataset:

CUSTOMER_KEY	INVOICE_KEY	MON_KEY	AMOUNT	Remarks/Operation
1111	6008	201001	450	The cache file updated after aggregation
2222	6009	201001	500	The cache file remains the same as before
3333	5003	201001	300	New group row inserted in cache file
4444	1234	201001	350	New group row inserted in cache file
5555	6157	201001	500	New group row inserted in cache file

Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

## Summary

- This Lesson gives knowledge incremental aggregation



Summary



Copyright © Capgemini 2014. All Rights Reserved. 5

## Review Question

- Is a pre-requisite for incremental aggregation

- A. SCD
- B. CDC

Option B

- Cache file is recreated when incremental aggregation

- check box is checked

- A. True
- B. False

Option B



Copyright © Capgemini 2014. All Rights Reserved.

## **Informatica PowerCenter**

Lesson 19: Constraint  
Based Loading

## Lesson Objectives

- In this Lesson you will learn about:
  - Introduction to Constraint Based Loading
  - Steps For Implementation



## 19.1. Introduction To Constraint Based Loading

- Constraint based loading is used when we need to populate the tables, which have parent child relationship, in the same mapping
- Constrained base load order Informatica load the data on the basis of constraints to target tables, basically Primary key and Foreign key
- When target tables have no key relationships, the Informatica Server does not perform constraint-based loading
- When target tables have circular key relationships, the Informatica Server reverts to a normal load.
- The constraint based load ordering option applies for only insert operations.
- The target tables must be in the same Target connection group.
- Use parameters file to make sessions more flexible and reusable.

Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

## 19.2. Steps For Implementing Constraint Based Loading

- Steps for creating mapping following Constraint based loading
  - First import the source and target tables (Take for example Product and Customer table)
  - Make sure that the foreign key relationship exists between the Product and Customer table else create the relationship in the target analyzer
  - Create the mapping with the source pointing to both the target Product and Customer table.
  - Once mapping is done create workflow and session and check the option of constraint based load ordering.
  - Save the mapping and run the workflow



Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

## Summary

- This Lesson gives knowledge constraint based loading



Summary



Copyright © Capgemini 2012. All Rights Reserved.

## Review Question

- 1. Constraint based loading can be used in insert /update strategy of target load ?

- A. no
- B. yes

Option A



- 2. Constraint based loading required target tables to be in same flow and have PK-FK relations defined in DB or in Informatica

- A. True
- B. False

Option A

## **Informatica PowerCenter**

Lesson 20: Row Error  
Logging

## Lesson Objectives

- In this Lesson you will learn about:
  - Introduction to Row Error Logging
  - Steps For Implementation



## 20.1. Introduction To Row Error Logging

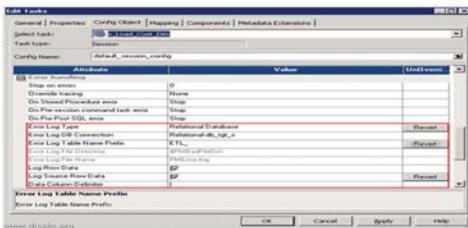
- During execution we can encounter two types of errors
  - Critical Error/Fatal Errors – Which breaks the flow
  - Non-Critical Error – Row level data errors
- For Non-critical Row level data issues we can either Configure session property
  1. Stop On Error =1 , which will break the flow and fail the session.
  2. Error logging into flat file/DB tables



Copyright © Capgemini 2014. All Rights Reserved.

## 20.2. Steps For Implementing Row Error Logging

- Steps for creating mapping following Constraint based loading
- Configure below session properties.



- With this configuration, we are done with the setting required to capture any error during the session execution.

Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

## 20.2. Steps For Implementing Row Error Logging

- Here is the details on what each property means.  
Error Log Type :- Specifies the type of error log to create. It can be relational database or flat file.
- Error Log DB Connection :- Database connection for a relational log.
- Error Log Table Name Prefix :- Specifies the table name prefix for relational logs.
- Log Row Data :- Specifies whether or not to log transformation row data.
- Log Source Row Data :- Specifies whether or not to log source row data.
- Data Column Delimiter :- Data will be delimited by the specified character in DB column.
- With this configuration we specified, Informatica PowerCenter will create four different tables for error logging and the table details as below  
`ETL_PMERR_DATA` :- Stores data about a transformation row error and its corresponding source row.
- `ETL_PMERR_MSG` :- Stores metadata about an error and the error message.
- `ETL_PMERR_SESS` :- Stores metadata about the session.
- `ETL_PMERR_TRANS` :- Stores metadata about the source and transformation ports, when error occurs.

Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

## 20.2. Steps For Retrieving Row Error Report

- For Retrieving Error Records there are two ways
  - Informatica Functionality –
- Download the workflow `wflw_To_Create_Files_For_Rejected_Records`.  
Download the UNIX script `splitfile.sh`.
  - Query the reject tables –
- Attached is the query for your reference
- Advantages
  - Helps create error file with respect to each transformation within Informatica session.
  - Helps you monitor any data quality issues.
  - Helps in debugging any transformation errors in Informatica mappings.
  - Helps you monitor the error records and provide corrections to existing ETL mapping to avoid these rejections on daily basis. This in turn will provide ongoing process improvements.
  - Provides visibility to Business and IT Analysts about the bad data or incorrect business logic in ETL mappings.



Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

### 20.3. Advantage of Row Error Report

- **Advantages**

- Helps create error file with respect to each transformation within Informatica session.
- Helps you monitor any data quality issues.
- Helps in debugging any transformation errors in Informatica mappings.
- Helps you monitor the error records and provide corrections to existing ETL mapping to avoid these rejections on daily basis. This in turn will provide ongoing process improvements.
- Provides visibility to Business and IT Analysts about the bad data or incorrect business logic in ETL mappings.

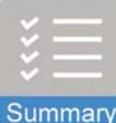


Copyright © Capgemini 2014. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

## Summary

- This Lesson gives knowledge Row Error Logging , its implementation and advantages



Summary



Copyright © Capgemini 2014. All Rights Reserved.

## Review Question

- 1. We have to create Error tables for Row Error logging ?

- A. no
- B. Yes

Option A

- 2. Error logging is done for

- A. Performance Gain
- B. Data Quality

Option B



## **Informatica PowerCenter**

Lesson 21: Error Handling  
Techniques

## Lesson Objectives

- In this Lesson you will learn about:
  - Introduction to Row Error Logging
  - Steps For Implementation



## 20.1. Introduction To Error Handling Techniques

- Data Quality is very critical to the success of every data warehouse projects. So ETL Architects and Data Architects spent a lot of time defining the error handling approach. Lets see how do we leverage the PowerCenter options to handle your exceptions.
- Error Classification
  - Fatal Errors – Which breaks the flow/stops the workflow
  - Non-Fatal Error –
    - Unexpected Conversion Error or other error due to which record is skipped
    - User defined /Logic failure Exception
- User Defined Exception
  - Business users define the user defined user defined exception, which is critical to the data quality. We can setup the user defined error handling using:
    - Error Handling Functions.
    - User Defined Error Tables.

Copyright © Capgemini 2014. All Rights Reserved.

## 20.1. Error Handling Functions

- **ERROR()** :

- This function Causes the PowerCenter Integration Service to skip a row and issue an error message, which you define.
- The error message displays in the session log or written to the error log tables based on the error logging type configuration in the session.
- You can use ERROR in Expression transformations to validate data

- Eg IIF(TRANS\_DATA > SYSDATE,ERROR('Invalid Transaction Date'))

- **ABORT()** :

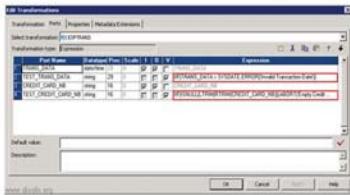
- Stops the session, and issues a specified error message to the session log file or written to the error log tables based on the error logging type configuration in the session.
- When the PowerCenter Integration Service encounters an ABORT function, it stops transforming data at that row.
- It processes any rows read before the session aborts
- E.g. : IIF(ISNULL(LTRIM(RTRIM(CREDIT\_CARD\_NB))),ABORT('Empty Credit Card Number'))



Copyright © Capgemini 2014. All Rights Reserved.

## 21.2. Error Handling Function

- Example Of Using Error Handling Functions



- Note :- You need to use these two functions in a mapping along with a session configuration for row error logging to capture the error data from the source system. Depending on the session configuration, source data will be collected into Informatica predefined PMERR error tables or files.

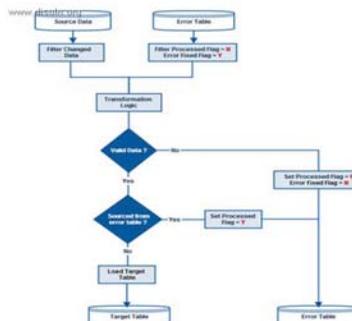


Copyright © Capgemini 2010. All Rights Reserved.

Use this mapping when you do not need to keep any previous versions / history of dimensions in the table.

## 20.2. User Defined Error Handling

- High Level Approach of Error Handlin



## 20.2. User Defined Error Handling

- Non-Fatal Exception Can be handled by
  - Default Port Value Setting.
  - Row Error Logging.
  - Error Handling Settings.
- Default Port Value Setting : Good way to handle Null , invalid dates , etc.
  - Row Error Logging – Configuring the session property
  - Error Handling Settings : Configuring session -> Configure Objects -> Error Handling following options are available
  - Stop – will stop the session on failure of the task.



Capgemini  
PROVIDING INTEGRATION SINCE 1975

Copyright © Capgemini 2014. All Rights Reserved

### 20.3. Handling Fatal Errors

- For Handling Fatal Error either we need to design
  - Workflow Recovery
  - Restartable ETL Design

Copyright © Capgemini 2014. All Rights Reserved.

## Workflow Recovery Overview

- Workflow recovery allows you to continue processing the workflow and workflow tasks from the point of interruption.
- You can recover a workflow if the Integration Service can access the workflow state of operation.
- The workflow state of operation includes the status of tasks in the workflow and workflow variable values.
- The Integration Service stores the state in memory or on disk, based on how you configure the workflow:
  - Enable recovery
  - Suspend
- The Integration Service recovers tasks in the workflow based on the recovery strategy of the task.
- By default, the recovery strategy for Session and Command tasks is to fail the task and continue running the workflow. You can configure the recovery strategy for Session and Command tasks.
- The strategy for all other tasks is to restart the task.

Copyright © Capgemini 2014. All Rights Reserved.

## Suspending the Workflow

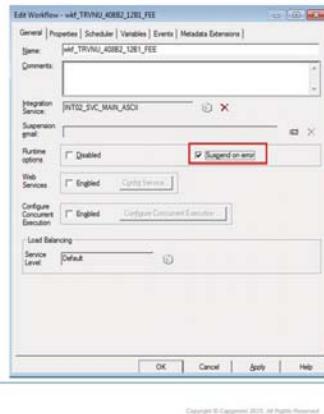
- When a task in the workflow fails, you might want to suspend the workflow, fix the error, and recover the workflow.
- The Integration Service suspends the workflow when you enable the Suspend on Error option in the workflow properties.
- Optionally, you can set a suspension email so the Integration Service sends an email when it suspends a workflow.
- When you enable the workflow to suspend on error, the Integration Service suspends the workflow when one of the following tasks fail:
  - Session
  - Command
  - Worklet
  - Email
- When the status of the workflow is "Suspended" or "Suspending," you can fix the error, such as a target database error, and recover the workflow in the Workflow Monitor.
- When you recover a workflow, the Integration Service restarts the failed tasks and continues evaluating the rest of the tasks in the workflow. The Integration Service does not run any task that already completed successfully.

Copyright © Capgemini 2010. All Rights Reserved. 10

## Suspending the Workflow

- To suspend a workflow:

1. In the Workflow Designer, open the workflow.
2. Click Workflows > Edit.
3. In the General tab, enable Suspend on Error.
4. Click OK.



Copyright © Capgemini 2014. All Rights Reserved. 11

## Workflow Recovery

- To recover the workflow, you must enable the workflow for recovery or configure the workflow to suspend on task error. When the workflow is configured for recovery, you can recover it if it stops, aborts, terminates, or suspends.
- There are 4 Recoverable workflow status
  - [1] Aborted :
  - You abort the workflow in the Workflow Monitor or through pmcmd.
  - You can also choose to abort all running workflows when you disable the service process in the Administration Console.
  - You can recover an aborted workflow if you enable the workflow for recovery.
  - You can recover an aborted workflow in the Workflow Monitor or by using pmcmd.
- [2] Stopped:
  - You stop the workflow in the Workflow Monitor or through pmcmd.
  - You can also choose to stop all running workflows when you disable the service or service process in the Administration Console.
  - You can recover a stopped workflow if you enable the workflow for recovery.
  - You can recover a stopped workflow in the Workflow Monitor or by using pmcmd.



Copyright © Capgemini 2010. All Rights Reserved. 10

## Workflow Recovery

[3] Suspended:

- A task fails and the workflow is configured to suspend on a task error.
- If multiple tasks are running, the Integration Service suspends the workflow when all running tasks either succeed or fail.
- You can fix the errors that caused the task or tasks to fail before you run recovery.
- By default, a workflow continues after a task fails. To suspend the workflow when a task fails, configure the workflow to suspend on task error.

[4] Terminated:

- The service process running the workflow shuts down unexpectedly. Tasks terminate on all nodes running the workflow.
- A workflow can terminate when a task in the workflow terminates and you do not have the high availability option.
- You can recover a terminated workflow if you enable the workflow for recovery.
- When you have high availability, the service process fails over to another node and workflow recovery starts.

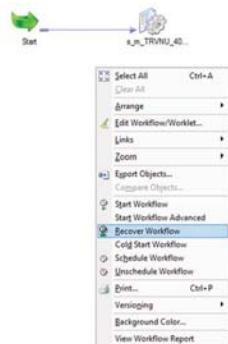


Copyright © Capgemini 2014. All Rights Reserved. 10

## Recovering a Workflow

- To recover a workflow using the Workflow Manager:

- Select the workflow in the Navigator or open the workflow in the Workflow Designer workspace.
  - Right-click the workflow and choose Recover Workflow.
- The Integration Service recovers the interrupted tasks and runs the rest of the workflow.



## Task Recovery

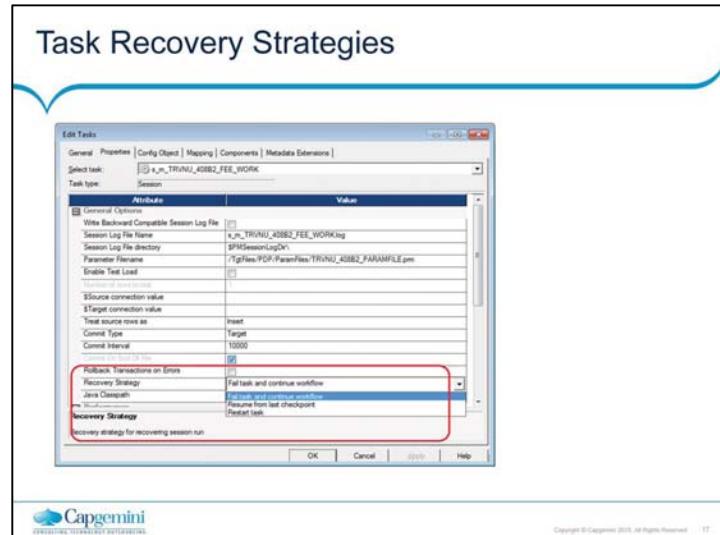
- When you recover a workflow, the Integration Service recovers the tasks based on the recovery strategy for each task.
- Depending on the task, the recovery strategy can be fail task and continue workflow, resume from the last checkpoint, or restart task.
- When you enable workflow recovery, you can recover a task that you abort or stop.
- You can recover a task that terminates due to network or service process failures.
- When you configure a workflow to suspend on error, you can recover a failed task when you recover the workflow.
- There are 3 Task Recovery Strategies
  - [1] Restart task:
    - When the Integration Service recovers a workflow, it restarts each recoverable task that is configured with a restart strategy.
    - You can configure Session and Command tasks with a restart recovery strategy.
    - All other tasks have a restart recovery strategy by default.

Copyright © Capgemini 2014. All Rights Reserved. 10

## Task Recovery

- [2] Fail task and continue workflow:
  - When the Integration Service recovers a workflow, it does not recover the task.
  - The task status becomes failed, and the Integration Service continues running the workflow.
  - Configure a fail recovery strategy if you want to complete the workflow, but you do not want to recover the task.
  - You can configure Session and Command tasks with the fail task and continue workflow recovery strategy.
- [3] Resume from the last checkpoint:
  - The Integration Service recovers a stopped, aborted, or terminated session from the last checkpoint.
  - You can configure a Session task with a resume strategy.
  - Notes: You can use one of the following methods to recover a workflow or task:
    - Recover a workflow. Continue processing the workflow from the point of interruption.
    - Recover a session. Recover a session but not the rest of the workflow.
    - Recover a workflow from a session. Recover a session and continue processing a workflow.

Copyright © Capgemini 2010. All Rights Reserved. 10



## Review Question

■ Which is NOT Recoverable workflow status ?

- A. Aborted
- B. Stopped
- C. Suspended
- D. Terminated
- E. Declined

Option E



■ Which is NOT Task Recovery Strategies ?

- A. Recover a workflow
- B. Restart task
- C. Fail task and continue workflow
- D. Resume from the last checkpoint

Option A



Copyright © Capgemini 2014. All Rights Reserved. 10

## Summary

- This Lesson gives knowledge Row Error Logging , its implementation and advantages



Summary



Copyright © Capgemini 2014. All Rights Reserved. 10

## Review Question

- We have to create Error tables for Row Error logging ?

A.no  
B.yes

Option A

- Error logging is done for

A.Performance Gain  
B.Data Quality

Option B



Copyright © Capgemini 2014. All Rights Reserved. 30

## Informatica PowerCenter

Lesson 22: Metadata Tables  
and Views

## Lesson Objectives

- In this Lesson you will learn about:
  - Metadata tables and views



## What are Repository tables?

- All objects that we create in Informatica Power Center (sources, targets, transformations, mappings, sessions, workflows, command tasks etc) get stored in a set of database tables.
- These database tables are known as either Repository tables or metadata tables or OPB tables.



Copyright © Capgemini 2015. All Rights Reserved. 2

## Repository Tables

- OPB\_TASK\_ATTR - Task attributes tables
  - This is the table that stores the attribute values (like Session log name etc) for tasks.
- OPB\_WIDGET - Transformations table
  - This table stores the names and IDs of all the transformations with their folder details
- OPB\_WIDGET\_FIELD - Transformation ports table
  - This table stores the names and IDs of all the transformation fields for each of the transformations.
- OPB\_WIDGET\_ATTR - Transformation properties table
  - This table stores all the properties details about each of the transformations
- OPB\_EXPRESSION - Expressions table
  - This table stores the details of the expressions used anywhere in PowerCenter.



Copyright © Capgemini 2008. All Rights Reserved.

### OPB\_TASK\_ATTR - Task attributes tables

Usage: Use the ATTR\_ID of this table to that of the ATTR\_ID of OPB\_ATTR table to find what each attribute in this table means. You can know more about OPB\_ATTR table in the next paragraphs.

### OPB\_WIDGET - Transformations table

Usage: Use WIDGET\_ID from this table to that of the WIDGET\_ID of any of the tables to know the transformation name and the folder details. Use this table in conjunction with OPB\_WIDGET\_ATTR or OPB\_WIDGET\_EXPR to know more about each transformation etc.

### OPB\_WIDGET\_FIELD - Transformation ports table

Usage: Take the FIELD\_ID from this table and match it against the FIELD\_ID of any of the tables like OPB\_WIDGET\_DEP and you can get the corresponding information.

### OPB\_WIDGET\_ATTR - Transformation properties table

Usage: Use the ATTR\_ID of this table to that of the ATTR\_ID of OPB\_ATTR table to find what each attribute in this transformation means.

### OPB\_EXPRESSION - Expressions table

Usage: Use this table in conjunction with OPB\_WIDGET/OPB\_WIDGET\_INST and OPB\_WIDGET\_EXPR to get the expressions in the Expression transformation for a particular, mapping or a set.

## Repository Tables

- PB\_ATTR - Attributes
  - This table has a list of attributes and their default values if any.
- OPB\_COMPONENT - Session Component
  - This table stores the component details like Post-Session-Success-Email, commands in Post-Session/pre-Session etc.
- OPB\_CFG\_ATTR - Session Configuration Attributes
  - This table stores the attribute values for Session Object configuration like "Save Session log by", Session log path.



Copyright © Capgemini 2016. All Rights Reserved.

### OPB\_ATTR - Attributes

You can get the ATTR\_ID from this table and look it up against any of the tables where you can get the attribute value. You should also make a note of the ATTR\_TYPE, OBJECT\_TYPE\_ID before you pick up the ATTR\_ID. You can find the same ATTR\_ID in the table, but with different ATTR\_TYPE or OBJECT\_TYPE\_ID.

### OPB\_COMPONENT - Session Component

Usage: Match the TASK\_ID with that of the SESSION\_ID in OPB\_SESSION table to get the SESSION\_NAME and to get the shell command or batch command that is there for the session, join this table with OPB\_TASK\_VAL\_LIST table on TASK\_ID.

### OPB\_CFG\_ATTR - Session Configuration Attributes

This table stores the attribute values for Session Object configuration like "Save Session log by", Session log path etc.

## Mapping and Maplet Views

- Mapping and maplet views allow you to see the sources, targets, and transformations used in a mapping or a maplet by folder in a PowerCenter repository.
- These views also display properties of mappings and maplets such as description, version and creation date, validity of the mapping or maplet, and whether the mapping or maplet is a shortcut.

Copyright © Capgemini 2008. All Rights Reserved.

## Mapping and Maplet Views

View	Description
REP_ALL_MAPPINGS	This view provides a list of the latest version of all mappings defined in each folder of a repository.
REP_ALL_MAPPLETS	This view provides a list of the latest version of all mapplets defined in each folder of a repository.
REP_TARG_MAPPING	This view provides access to the compound table-level transformation expressions for each target table.
REP_TARG_FLD_MAP	This view shows compound field-level transformation expressions associated with a target.
REP_FLD_MAPPING	This view shows the source fields used by the target fields in a mapping. This is the companion view for the REP_TBL_MAPPING view.
REP_SRC_MAPPING	This view shows all sources used in a mapping.
REP_SRC_FLD_MAP	This view shows all of the source fields used in a mapping.
REP_TBL_MAPPING	This view shows all of the target tables used in a mapping and provides source to target mapping information.

Copyright © Capgemini 2008. All Rights Reserved

## Mapping and Maplet Views

REP_TARG_TBL_JOINS	This view contains join information between target tables.
REP_MAPPING_CONN_PORTS	This view displays the port-level connections between the objects of a mapping.
REP_MAPPING_UNCONN_PORTS	This view displays the unconnected ports in sources, targets, and transformations in a mapping.

Copyright © Capgemini 2008. All Rights Reserved.

Demo

- Show Metadata Tables, Views, execute queries



Capgemini  
INNOVATING INTELLIGENCE EXPERIENCE

Copyright © Capgemini 2008. All Rights Reserved.

#### Steps to create a Lookup transformation:

In the Mapping Designer, choose Transformation | Create menu option. Select the Lookup Transformation. Enter a name for the lookup. The naming convention for Lookup Transformations is LKP\_TransformationName. Click OK.

In the Select Lookup Table dialog box choose the lookup table. Click the Import button if the lookup table is not in the source or target database.

To manually define the Lookup Transformation, click the Skip button.

Define input ports for each Lookup condition.

For an Unconnected Lookup Transformation, create a return port for the value you want to return from the lookup.

Define output ports for the values you want to pass to another transformation.

Add the lookup conditions.

On the Properties tab, set the properties for the lookup.

Click OK.

For Unconnected Lookup Transformations, write an expression in another transformation using :LKP to call the Unconnected Lookup Transformation.

## Summary

- This Lesson gives knowledge about Metadata tables and Views



Summary



Copyright © Capgemini 2016. All Rights Reserved.

## Informatica PowerCenter

Lesson 23: Standards And Tips

## Lesson Objectives

- In this Lesson you will learn about:
  - Naming Convention
  - Tuning Tips



## Naming Conventions

- Good Practice to Follow Naming Conventions
- Can be project specific:-
  - Workflow: wfl\_ followed by workflow functionality
  - Session: s\_ followed by mapping name
  - Mapping: m\_ followed by mapping functionality
  - Source: Table/File name
  - Target: Table/File name
  - Ports:
    - Input & Output :- Column Names
    - Variable:- v\_ followed by functionality

Copyright © Capgemini 2014. All Rights Reserved.

## Naming Conventions - Transformations:

Source Qualifier:	sql_(followed by Source Name)
Stored Procedure:	sp_(followed by purpose of transformation)
Sequence Generator:	seq_
Expression:	exp_
Joiner:	jnr_
Lookup:	lkp_
Filter:	fil_
Rank:	rnk_
Router:	rtr_
Update Strategy:	upd_
Aggregator:	agg_
Normalizer:	nrm_

Copyright © Capgemini 2012. All Rights Reserved.

## Golden Rule No. 1 – Set Out Standards

- In other words..... Create a great team!
  - Naming standards
    - Everyone knows what it does
    - Easy to pick up someone else's work
    - Don't end up with 100 connection objects to the same database
    - Don't end up with 100 lookups to the same table
  - Development Standards
    - Annotate Objects clearly
    - Audit trail
  - Shared Object Policy
    - Object Stewardship
    - Single version of the truth
    - Use your shared folder!

Copyright © Capgemini 2014. All Rights Reserved.

## Golden Rule No. 2 – Know Your Data!

- In other words.....talk to the data owners AND see for yourself!
  - Analyse, analyse and analyse again
    - Don't always accept the statement "there are no errors in this data"
  - Verify actual values against permitted values
    - Devise an error handling strategy
    - Implement in all mappings
    - Don't rely on the .bad files
  - Verify the business rules (and get sign off!)
  - Design and unit test with real data – or at least as realistic as possible
  - Will design of source and target help or hinder performance?
    - Indexes
    - Constraints

Copyright © Capgemini 2012. All Rights Reserved.

## Golden Rule No. 3 – Plan Your Flows

- In other words..... Know where you're going!
  - Plan for reuse
    - Common lookups
    - Common expressions (e.g. date conversions)
    - Maple's
  - Use appropriate Transformations
    - Routers vs multiple filters
    - SQ Override vs Joiner/Union
    - Expression vs aggregator
  - Make the most of the resources
    - Push processing back to the database
    - Why not stage files?
  - Design for rerun

Copyright © Capgemini 2012. All Rights Reserved.

## Golden Rule No. 4 – Reduce Data ASAP

- In other words..... Trash what you don't need!
  - Only connect ports that you need
    - Reduces data passing through transformations
  - Remove bad data early
    - Route to your error handling
    - Avoids excessive processing on bad data
    - Complex Transforms only on verified data
  - Filter / Aggregate ASAP
    - If possible filter data in source qualifier
    - Consider aggregation in SQL override
  - Minimise Transformations
    - Avoid using one expression for one calculation
    - You'll need more DTM if you have excessive transformations

Copyright © Capgemini 2012. All Rights Reserved.

## Golden Rule No. 5 – Avoid Big Caches

- In other words..... Don't be a waste of space!
  - Pre Sort Joiners & Aggregators
    - Use the order by in the source qualifier
  - Only Use necessary ports
    - Avoid caching large text ports if the data isn't needed later
  - Filter ASAP
    - Don't aggregate / join data you'll trash later
  - Utilise the Master / Detail properties
    - Make the master the set with the fewest records
  - Avoid the sorter - order by instead
  - Need to sort a text file?
    - Consider staging it instead!
  - Size your caches for production

Copyright © Capgemini 2012. All Rights Reserved.

## Golden Rule No. 6 – Remember It's Shared

- In other words..... Be a good neighbour!
- Use Fewer concurrent sessions
  - Running more may cause other teams to go into wait state (and in production this means missed deadlines)
- Use performance stats only when tuning
  - Session requires twice the memory which may cause other sessions to fail

Copyright © Capgemini 2014. All Rights Reserved.

## Don'ts - avoid where possible:

- SQL Override
  - Use source qualifier properties where possible
- Multiple filters
  - Try a router instead!
- Sorters – Very inefficient
- Field level stored procedures
  - Get called for every row
- Extracting more data than you need
  - Filter in the source qualifier
  - Only map required ports
- Aggregator
  - Try a SQL override
- Use sorted data where possible
- Create running sums in an expression instead
- Use complex rules in filters and routers
- Hardcode Values
- Give connections environment specific names

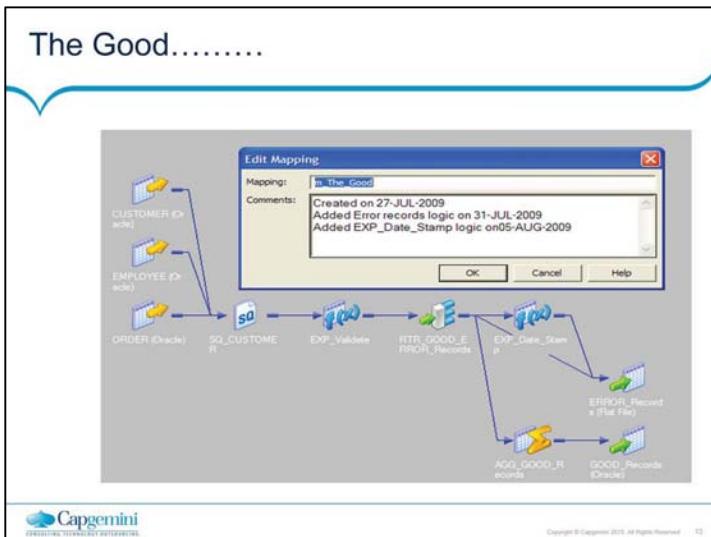
Copyright © Capgemini 2014. All Rights Reserved. 11

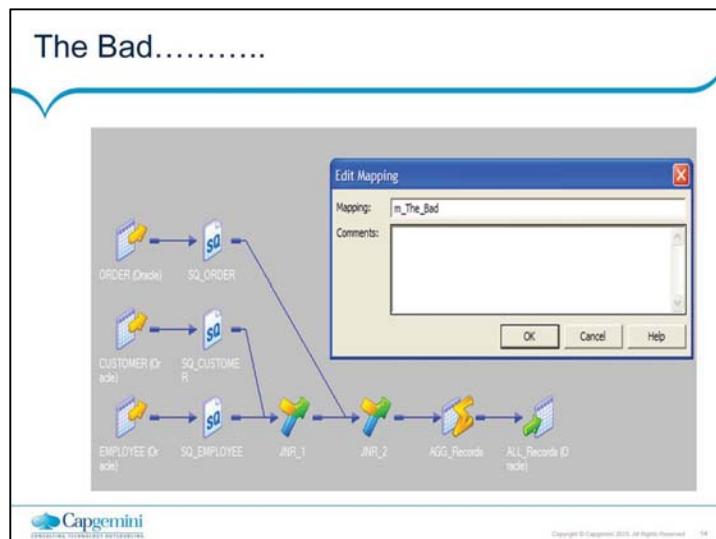
## The Good And the Bad

- Informatica Design Examples



Copyright © Capgemini 2014. All Rights Reserved. 10





## Summary

- This Lesson gives knowledge about Standard Naming convention and designing tips.



Summary



Copyright © Capgemini 2014. All Rights Reserved. 10

# Informatica Powercenter Lab Book

## Table of Contents

Table of Contents .....	2
Lab 1-1 Create a Folder .....	3
Lab 2-1 Analyze Source Data.....	7
Lab 2-2 Design a Target Schema.....	17
Lab 2-3 Create a Mapping.....	24
Lab 3-1 Creating Workflow .....	39
Lab 3-2 Start and Monitor Workflows .....	48
Lab 4-1 Sales Summary .....	54
Lab 5-1 New Customer.....	61
Lab 6-1 Flat File Join .....	69
Lab 7-1 Create a Mapplet.....	75
Lab 7-2 Quarterly-Sales Mapping.....	79
Lab 7-3 Annual Sales Mapping .....	88
Lab 8-1 Updating Current Items .....	92
Lab 9-1 Listing Order Details.....	96
Lab 10-1 Router Transformation .....	100
Lab 11-1 Sequence Generator Transformation.....	105
Lab 12-1 Stored Procedure Transformation.....	110
Lab 13-1 Configure an Email Task.....	116
Lab 14-1 Configure a Command Task .....	120
Lab 15-1 Event-Wait task .....	122
Lab 16-1 Pre-SQL and Post-SQL .....	123
Lab 17-1 Multiple Source Files(Indirect File).....	124
Lab 18-1 Mapping Parameter.....	125
Lab 19-1 Mapping Variable .....	126
Lab 20-1 FTP Connection .....	127
Lab 21-1 Dynamic Lookup.....	128
Lab 22-1 Using Mapping Wizard for SCD Type1 .....	130
Lab 23-1 Using Mapping Wizard for SCD Type2 .....	135
Appendix A – Sources and Targets used in the Mappings .....	139

## Lab 1.1. Lab 1-1 Create a Folder

<b>Goals</b>	<ul style="list-style-type: none"><li>• Learn to navigate through PowerCenter Repository Manager</li><li>• Connect to the repository</li><li>• Create a folder, subject to appropriate privileges</li></ul>
<b>Time</b>	15 Minutes
<b>Lab Setup</b>	<ul style="list-style-type: none"><li>• Informatica Client</li><li>• Login details for connecting to the repository</li></ul>

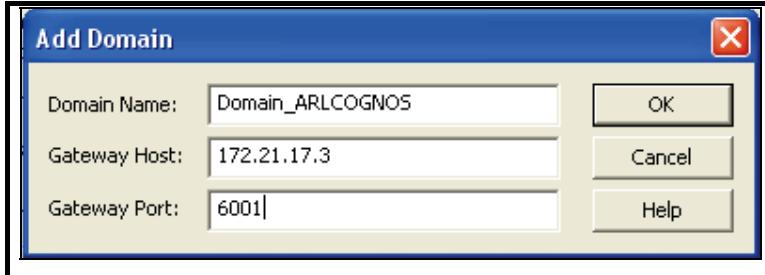
### Start the Repository Manager

1. START | PROGRAMS | INFORMATICA POWERCENTER | CLIENT | POWERCENTER REPOSITORY MANAGER.

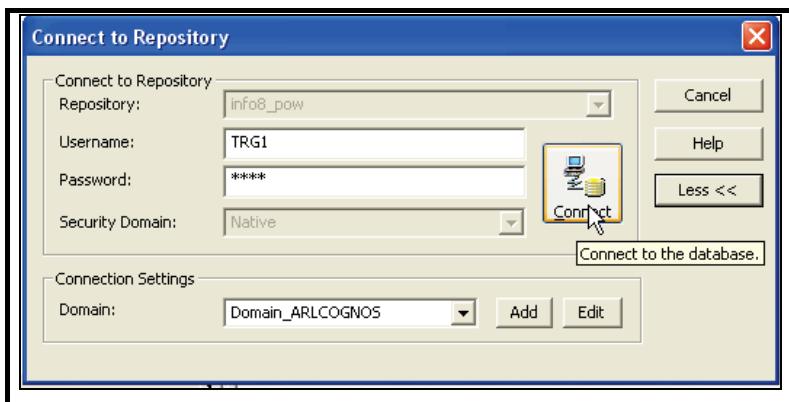
### Connect to the repository

1. In the Repository Manager's Navigator Window, use any one of the methods given below:
  - i. Double click on the repository (The repository name will be given by your instructor), or
  - ii. Select the repository and select the menu option REPOSITORY | CONNECT, or
  - iii. Select the repository and click the  connect icon in the toolbar.
2. In the Connect to Repository dialog box, enter the following details:
  - iv. Assigned Username.
  - v. Assigned Password.
3. Add Domain Name, Gateway Host, Gateway Port .

**Note:** Your faculty will provide the above details.



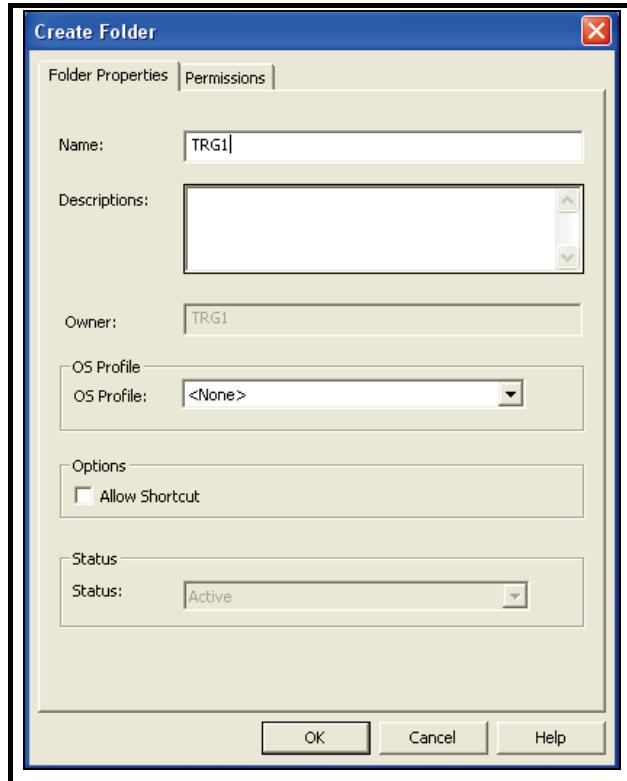
4. Click on the Connect button to connect to the repository.

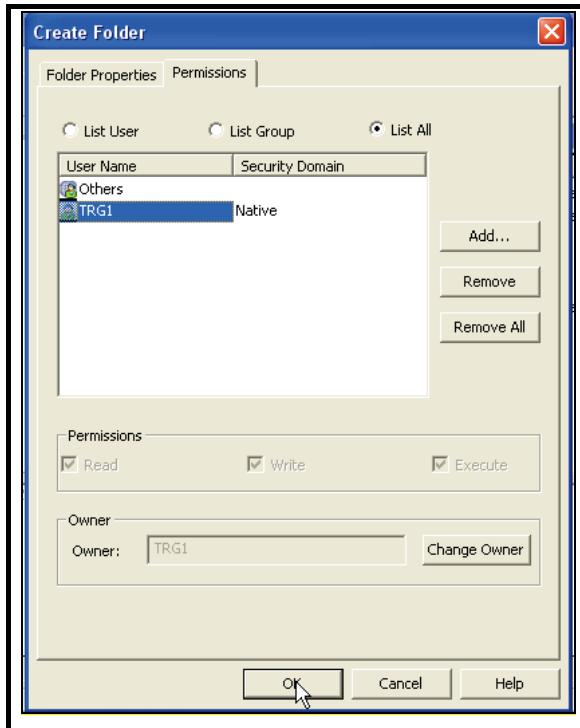


**Note:** The Add Domain Name, Gateway Host, Gateway Port are required when you connect to the repository for the first time. Subsequent connections to the repository require only the Username and Password.

#### Create a Folder

1. Click on the repository and select FOLDER | CREATE.
2. Enter the following information:
  - Name : **TRGx** (where x is your student number)
  - Description: Comments about the folder
  - Click on the permission tab and select TRGx from the **user name**  
Select TRGx as owner
3. Click OK.





## Lab 2.1. Lab 2-1 Analyze Source Data

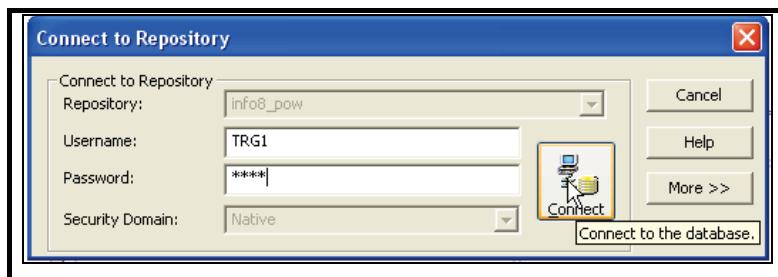
<b>Goals</b>	<ul style="list-style-type: none"><li>• Getting connected to PowerCenter Designer</li><li>• Understand the Source Analyzer tool</li><li>• Import source definitions into your folder</li><li>• Edit source definitions</li></ul>
<b>Time</b>	15 Minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Designer

### Getting connected to PowerCenter Designer

1. START PROGRAMS | INFORMATICA POWERCENTER CLIENT | DESIGNER.

You can connect to the repository in the following ways:

- i. Double click on the repository, or
- ii. Select the repository and select the menu option REPOSITORY | CONNECT, or
- iii. Select the repository and click on the  icon.



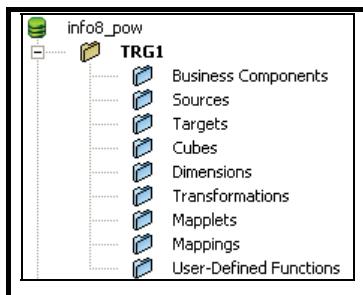
5. Enter the Username and Password as follows

Username : **TRGx**

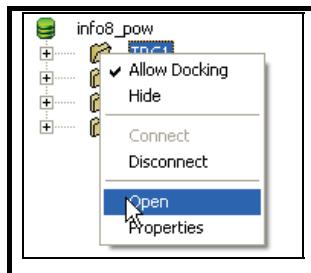
Password : **TRGx**

X is the student number assigned to you by your instructor.

6. Once you are connected you can see your folder name (TRGx) appearing in the Navigator Window.
7. To view all objects in the folder, click on the + sign.



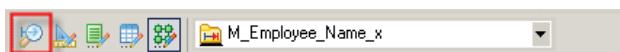
8. To open the folder, right click on the folder and select Open.



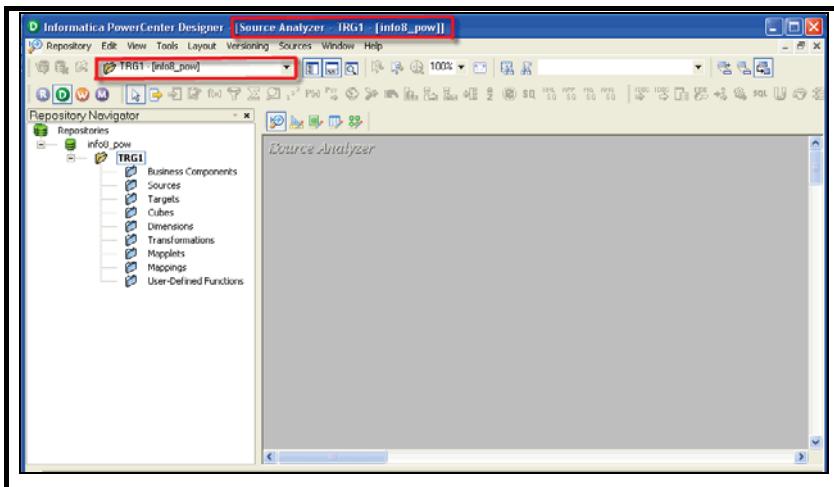
**Note :** An open folder is required in order to add, delete or modify objects. All of the work is performed in the Workspace Window, to the right of the Navigator Window (i.e. - Where the tools such as the Source Analyzer, Warehouse Designer, Mapplet Designer, etc., are active). The objects created in the Workspace, will appear in the Navigator Window.

### Working with Source Analyzer

1. In the Designer's Navigator Window, highlight your TRGx folder and select TOOLS | SOURCE ANALYZER or open the folder. This opens the Source Analyzer window. You can also click on the Source Analyzer button shown below to open Source Analyzer :

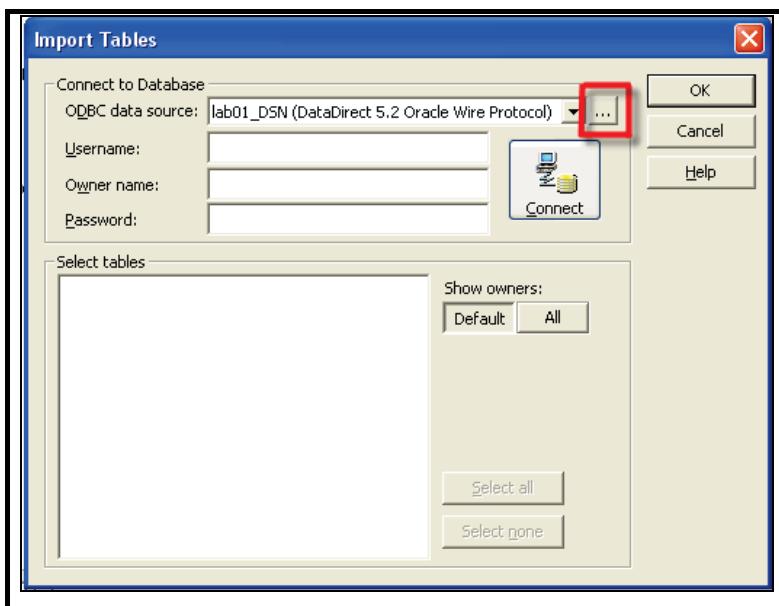


2. Notice the folder name and the repository names are displayed in the application title bar and Open Folders drop-down list.

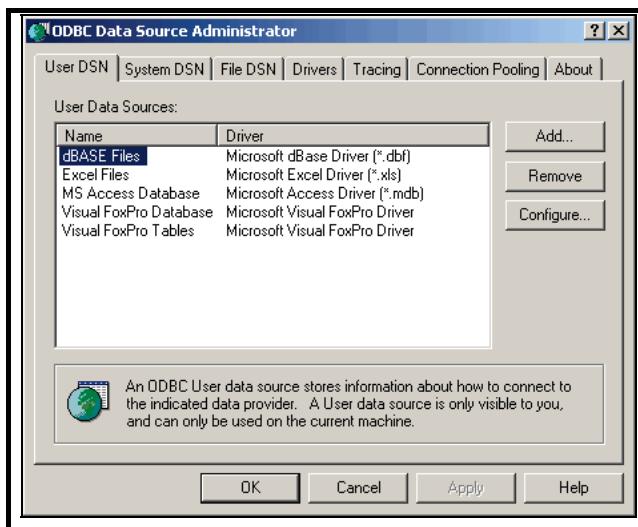


#### Importing Source from Database

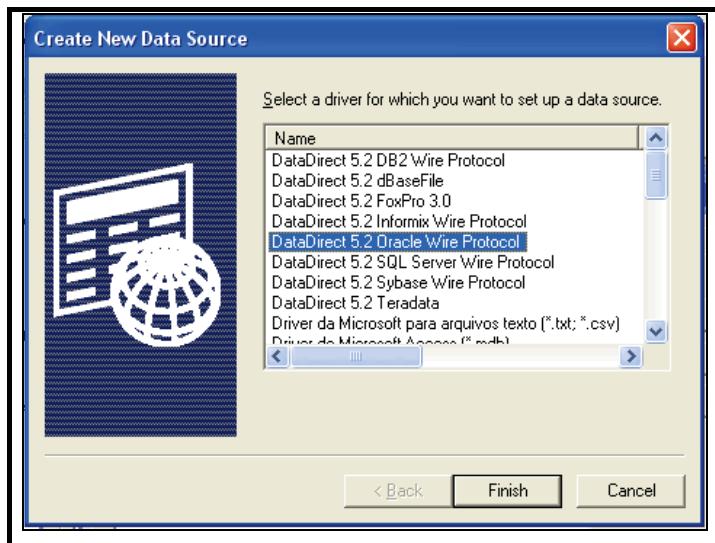
1. Select SOURCES | IMPORT FROM DATABASE.
  - i. The Import tables dialog box appears.



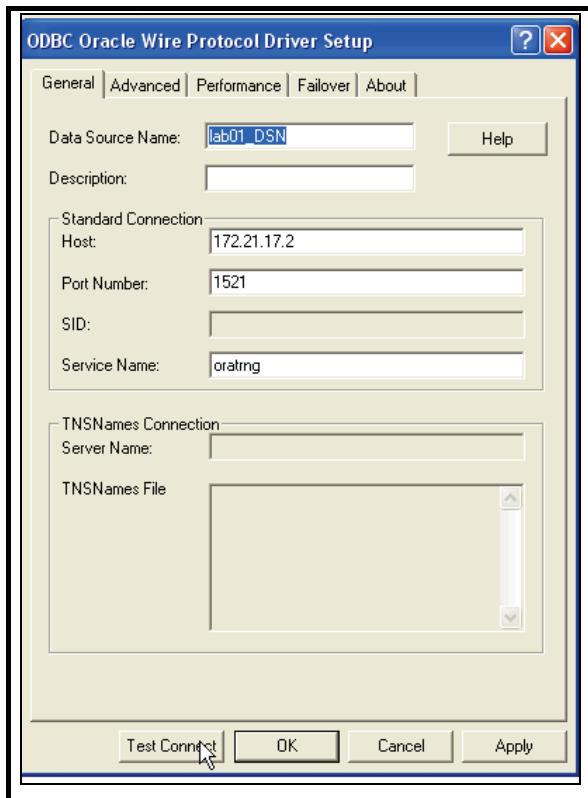
ii. Click on the [...] icon, to create the data source.



- iii. Click on Add in the ODBC Data Source Administrator box



- iv. Select the DataDirect 5.2 Oracle Wire Protocol and enter the Data Source Name, Description and Server Name.



- v. Click on Test Connect to test connectivity.

- vi. Enter the following information:

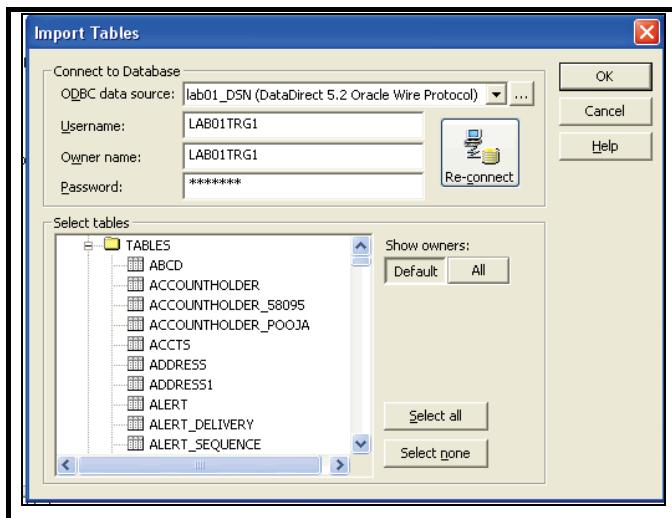
Enter appropriate oracle username and password

Username: **LAB01TRGx**(X is the student number assigned to you)

Password: **oraclex**



- vii. If the connection to the database is successful, click on Apply or OK.
- viii. Select the ODBC data source from the pull down list, which corresponds to the location of the source tables (Oracle database).
9. Click on Connect after providing all the details.  
**Note:** Your Instructor will provide the username and password.
10. In the Select Tables box, expand the owner name until you see a TABLES listing. Select the EMPLOYEES table.  
**HINT:** To select multiple tables, press the **Ctrl** key while selecting each table with a single mouse click.

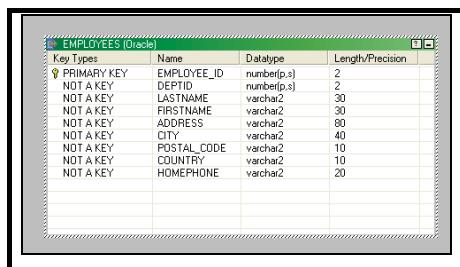


11. Click OK. The source table definition now appear in the Source Analyzer workbook.

**Note:** Using Designer's Navigator Window, notice the source definition has also been added to the Source section, or node, in your folder

#### Verify the source definitions

1. In the Source Analyzer workbook, for the source definition, expand the Key Types column.



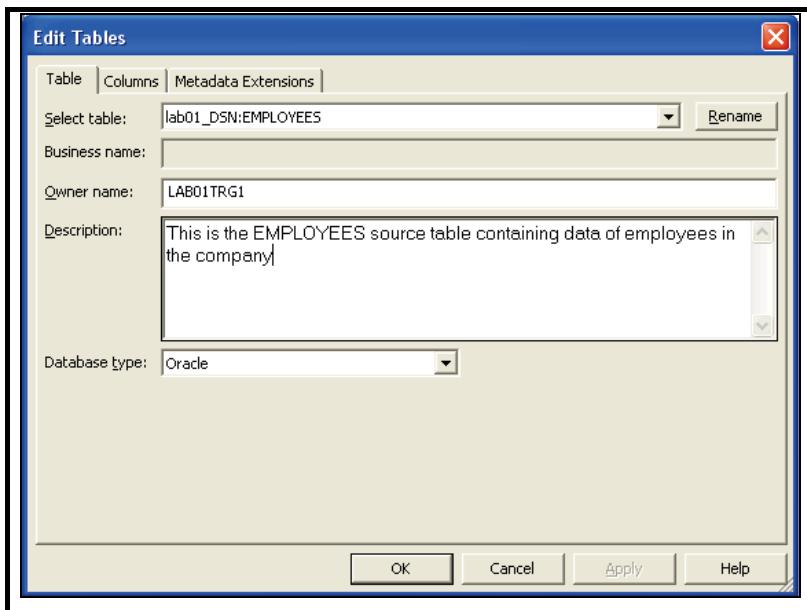
Key Types	Name	Datatype	Length/Precision
PRIMARY KEY	EMPLOYEE_ID	number(p,s)	2
NOT A KEY	DEPTID	number(p,s)	2
NOT A KEY	LAST_NAME	varchar2	30
NOT A KEY	FIRSTNAME	varchar2	30
NOT A KEY	ADDRESS	varchar2	80
NOT A KEY	CITY	varchar2	40
NOT A KEY	POSTAL_CODE	varchar2	10
NOT A KEY	COUNTRY	varchar2	10
NOT A KEY	HOMEPHONE	varchar2	20

**HINT:** When the source definitions are in **normal** mode, hold the mouse over the separator between the Key Types column and the Name column. When the

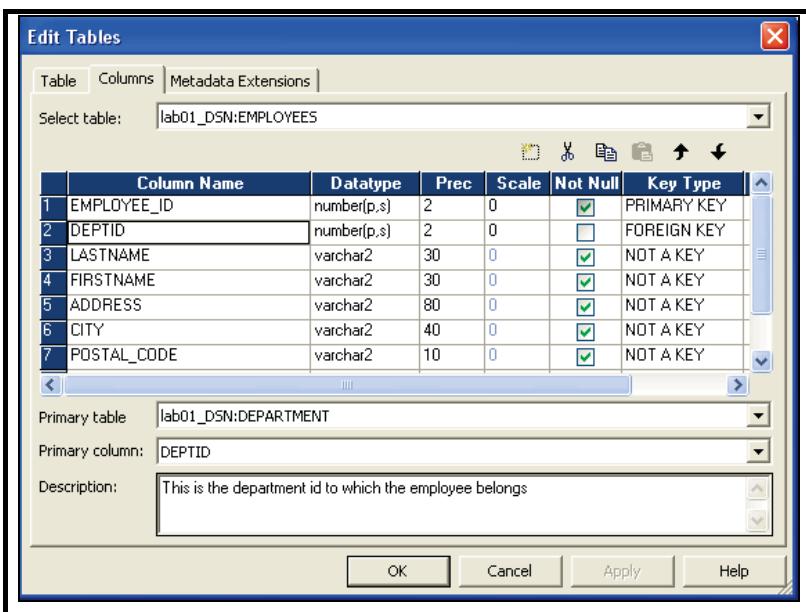
mouse turns into a bold double-arrow, click and drag the mouse to the right to expand the column.

### Edit Source Definitions

1. In the Source Analyzer workbook, double-click on the EMPLOYEES table. The Edit Tables dialog box appears.
2. Select the Table tab.
3. In the Description Window, enter:  
**"This is the EMPLOYEES source table containing data of employees in the company"**



4. Select the Columns tab.
5. Select the DEPT\_ID column name.
6. In the Description window, type:  
**"This is the department id to which the employee belongs"**



7. If you want to enter comments for additional columns, repeat the above steps.
8. Click on OK to save the comments and close the Edit Tables dialog box.
9. Select REPOSITORY | SAVE to save your folder in the repository.

## Lab 3.1. Lab 2-2 Design a Target Schema

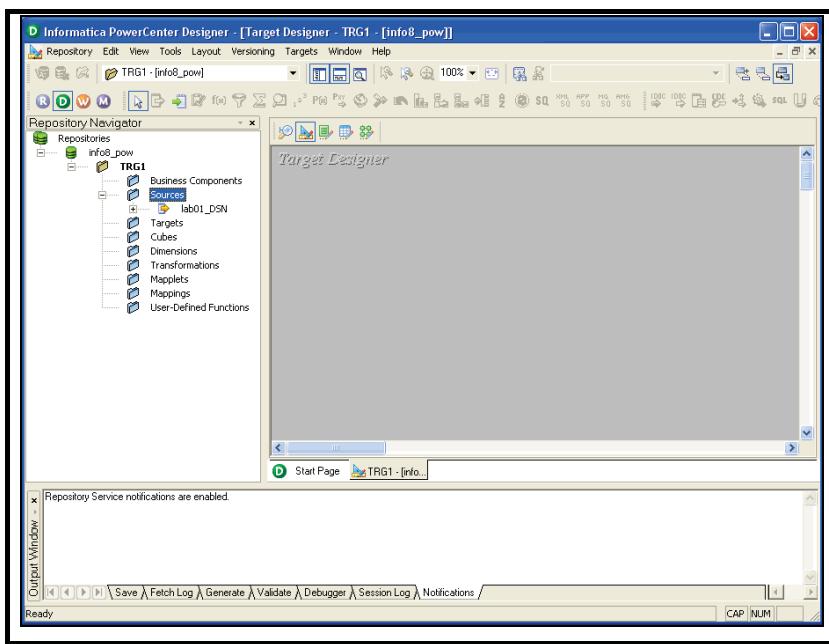
<b>Goals</b>	<ul style="list-style-type: none"><li>• Understand the TARGET Designer tool</li><li>• Create logical target definition</li><li>• Add new columns into the target</li><li>• Create physical targets</li></ul>
<b>Time</b>	10 Minutes
<b>Lab Setup</b>	A successful connection to the repository using PowerCenter Designer

### Understand TARGET Designer

1. Select TOOLS | TARGET DESIGNER to open the TARGET Designer or click on the TARGET Designer button as shown below.



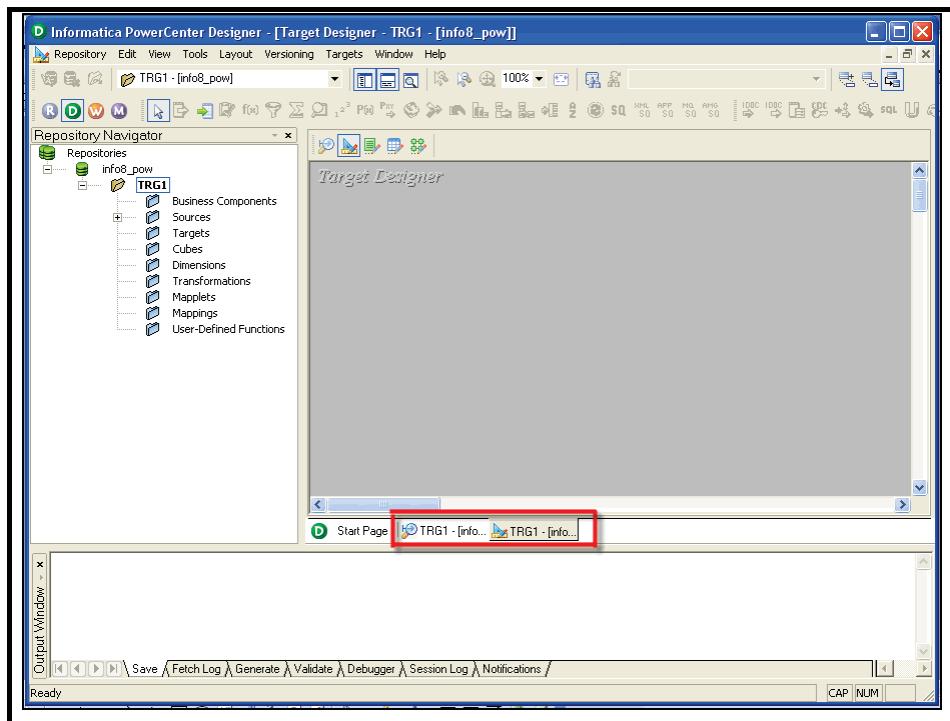
2. Make sure you are in the Workbook mode, by selecting VIEW | WORKBOOK. You should see the workbook tab.



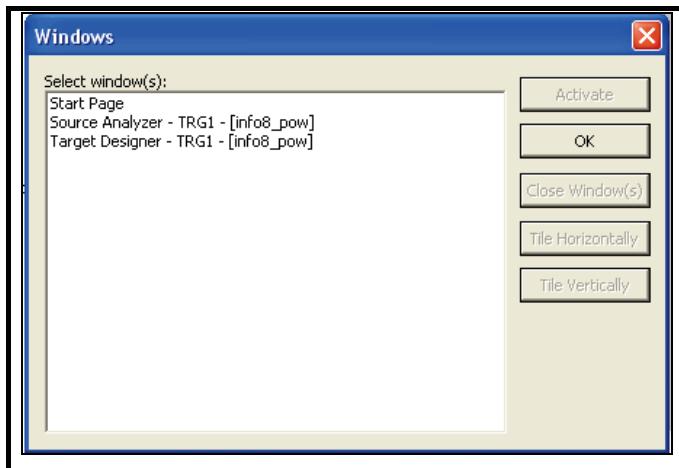
3. Click on the Source Analyzer button. The tab now displays the Source Analyzer icon as shown below.



4. Select WINDOW | NEW WINDOW. A new TARGET Designer Window is created.
5. In the Workbook Tab section, notice that now there are 3 tabs.



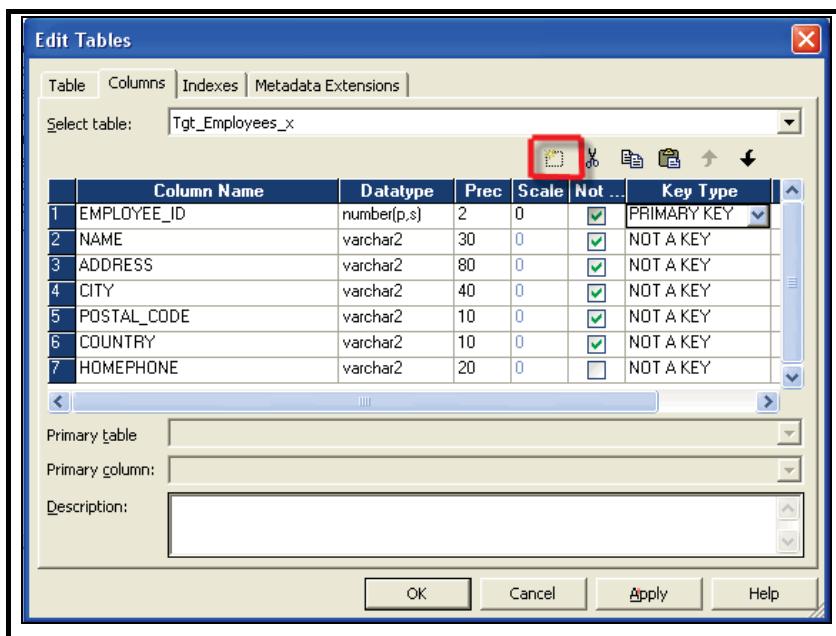
6. If you need to switch back and forth between the tools, you can click on the appropriate workbook tab, or select WINDOW | WINDOWS and select the window.

**Design the target schema (Physical target not existing in database)**

1. Select TARGETS | CREATE. The Create Target Table dialog box appears.
2. Enter the name of the target, Tgt\_Employees\_x.
3. Select the database type for the target table, i.e. Oracle.
4. Click on Create and Done.
5. The new table definition appears in the Target Designer workbook.
6. The target schema definition has also been added to your folder's Targets section, or node.
7. Double-click on the Tgt\_Employees\_x table. The Edit Tables dialog box appears.

8. Click on the Columns tab.

- i. Add the following columns using  icon.

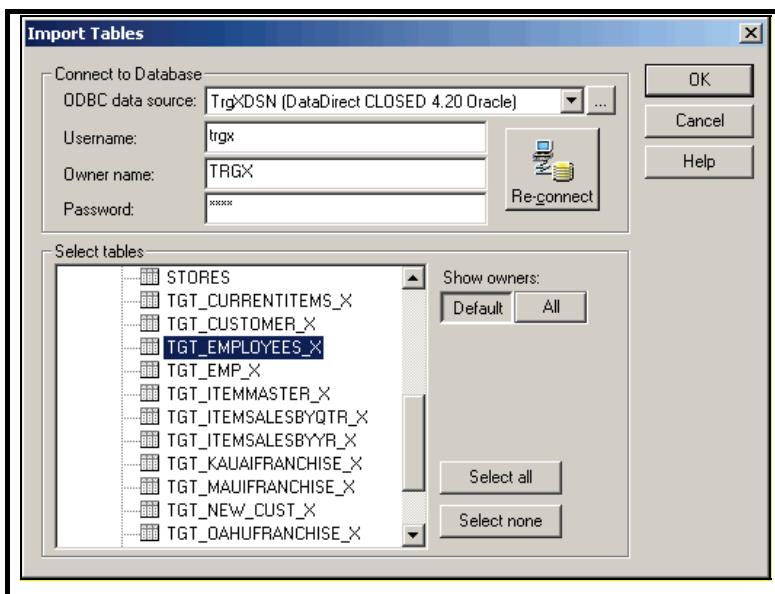


- ii. Once the table has been defined, close the Edit Tables dialog box

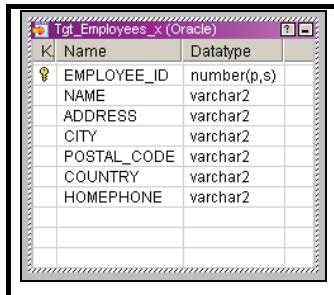
- iii. Save the newly designed schema to the repository.

#### Design the target schema (Physical target existing in database)

1. Select TARGETS | IMPORT FROM DATABASE.
2. Enter login details as shown below or as specified by your Instructor.
3. Click on Connect, expand TRGX and Tables and select the target table.
4. Click on OK.



5. The new table definition appears in the Target Designer workbook.
6. The target schema definition is added to your folder's Targets section, or in the node in the Designer's Navigator Window.



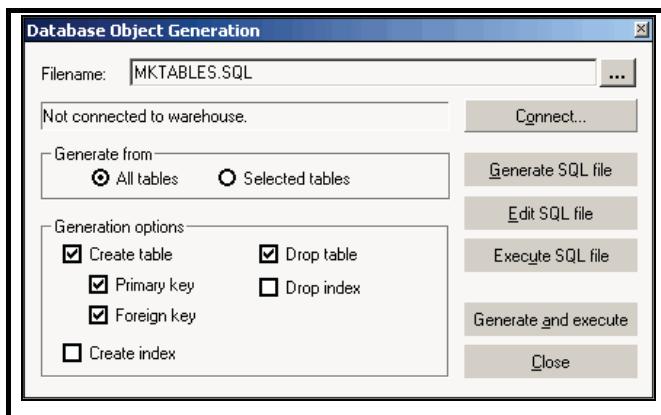
The screenshot shows the 'Tgt\_Employees\_X (Oracle)' table in the Target Designer. The columns and their data types are:

Name	Datatype
EMPLOYEE_ID	number(p,s)
NAME	varchar2
ADDRESS	varchar2
CITY	varchar2
POSTAL_CODE	varchar2
COUNTRY	varchar2
HOMEPHONE	varchar2

7. Double-click on the Tgt\_Employees\_x table. The Edit Tables dialog box appears. Click on Rename under the Table tab.
8. Click on the Columns tab and note the columns in the table.
9. Save the newly designed schema to the repository.

#### Create the physical target in the database

1. Click on the Tgt\_Employees\_x table to select it.
2. Select TARGETS | GENERATE/EXECUTE SQL from the menu. The Database Object Generation dialog box appears:



3. In the Filename entry box, accept the default script file name, MKTABLES.SQL.
4. Select the Selected tables radio button.

**Note:** Selecting the **All tables** radio button will write the code for all tables, which are in the Warehouse Designer workspace window.

5. Under Generation options, make sure the Create Table, Primary Key, Foreign Key and Drop Table options are checked.
6. Click on the Connect button.
7. Log in to the target database, using the proper ODBC connect string.
8. Click the Generate and Execute button.
9. If you receive a prompt asking if you want to overwrite the contents of the MKTABLES.SQL file, click on OK.
10. The MKTABLES.SQL containing the DDL script to create the Tgt\_Employees\_x table is created and runs against the target database.
11. Scroll through the contents of the Output Window to verify that the table was successfully created.
12. Click on the Close button.

## Lab 4.1. Lab 2-3 Create a Mapping

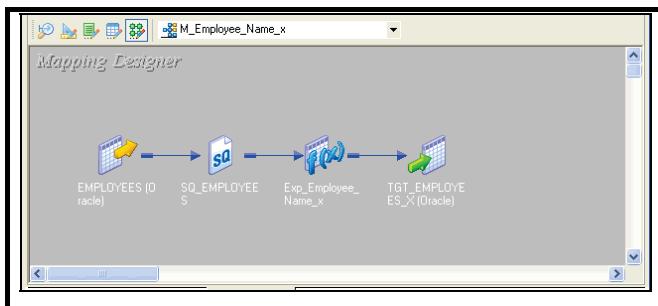
<b>Goals</b>	<ul style="list-style-type: none"> <li>• Understand the Designer's Mapping Designer tool</li> <li>• Create a mapping using:           <ul style="list-style-type: none"> <li>➢ A source definition</li> <li>➢ Source Qualifier and Expression Transformation</li> <li>➢ A target definition</li> </ul> </li> <li>• Edit transformations</li> <li>• Add new ports in transformations</li> <li>• Add formulas in the Expression transformation</li> <li>• Validate a mapping</li> <li>• Understand Designer's Output window</li> </ul>
<b>Time</b>	30 Minutes
<b>Lab Setup</b>	<ul style="list-style-type: none"> <li>• A successful connection to the repository using PowerCenter Designer</li> <li>• A source definition created</li> <li>• A target definition created</li> </ul>

### Solution

- Create a relational target table that contains the **Name** as a concatenation of First Name and Last Name

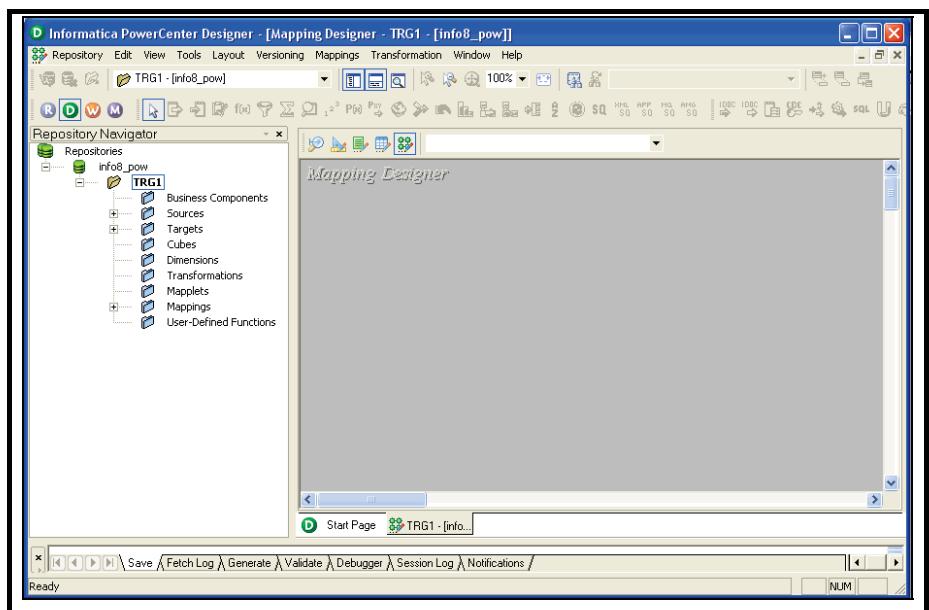
TRANSFORMATION NAME	TYPE	DESCRIPTION
EMPLOYEES	Relational Source Definition	Source Definition
SQ_EMPLOYEES_X	Source Qualifier	Data source qualifier for source table
EXP_EMPLOYEE_NAME_X	Expression	Link all ports from Source Qualifier to Expression transformation. Contains the expression for concatenation of First Name and Last Name
TGT_EMPLOYEES_X	Relational Target Table	Target definition which contains the concatenated name

## Mapping Layout

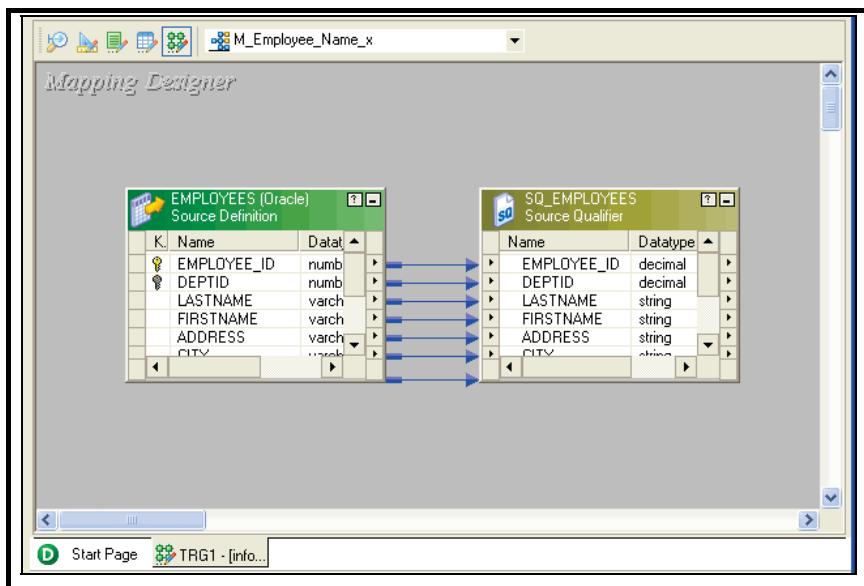


## Source Qualifier Transformation

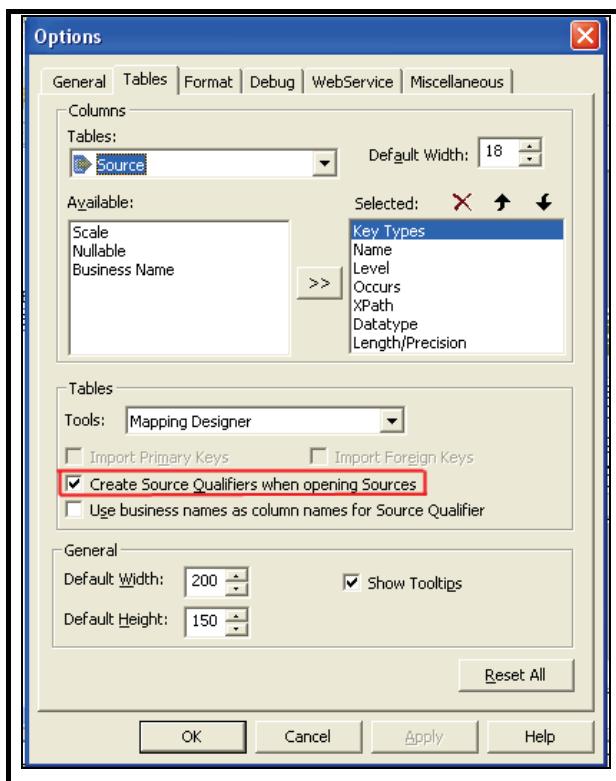
1. Select TOOLS | MAPPING DESIGNER, or click on the  icon. The workbook changes to the Mapping Designer.



2. Select MAPPINGS | CREATE, or click the Create Mapping  icon from the Mapping toolbar.
3. Enter M\_Employee\_Name\_x for the new mapping name
4. Click on OK.
5. Drag the source from the navigator to the Mapping Designer. Designer creates the Source Qualifier by default, and connects it to the source as shown below:



6. If the Source Qualifier transformation does not appear by default, Select TOOLS | OPTIONS and click on the TABLES tab. Ensure automatic creation by checking the Create Source Qualifiers When Opening Sources checkbox.

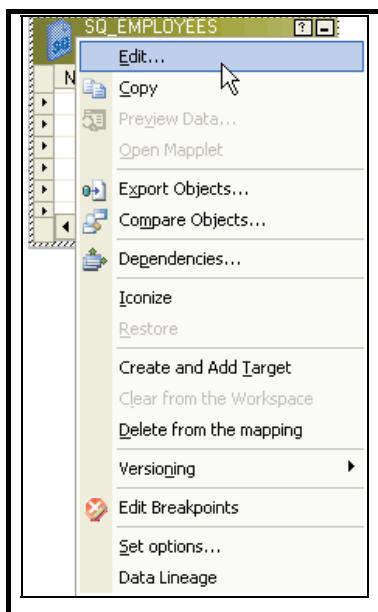


7. Drag the source again into the workspace. The Source Qualifier transformation automatically appears and is connected to the source. Delete the previously dragged source from the workspace by pressing the delete key.

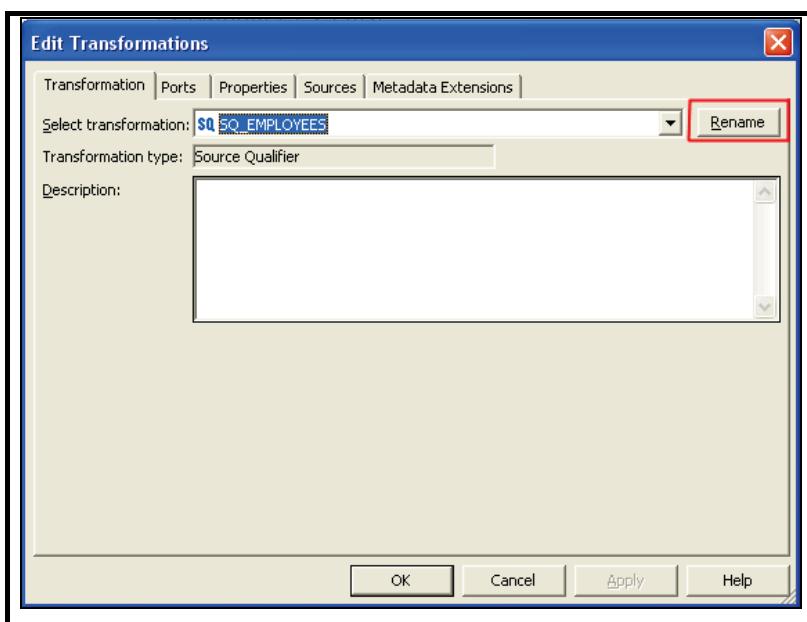
**Note:** Use the automatic Source Qualifier creation when you want to create one Source Qualifier for each source in your mapping and disable the automatic creation when you want to join data from different sources.

#### Edit the Source Qualifier Transformation

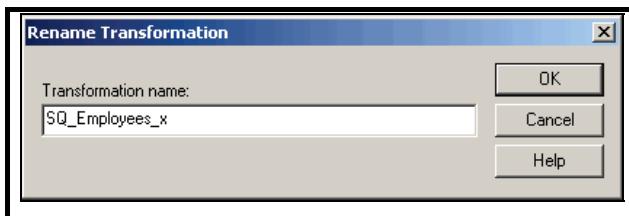
1. Use either of the methods to invoke the Edit Transformations box.
2. Right click on the Source Qualifier and select Edit...



- i. Double click on the Source Qualifier transformation.
- ii. Click on the Transformation tab and click on Rename to rename the transformation.



3. Enter the name and click on OK.



#### Create the Expression Transformation and link to the Source Qualifier

1. To create the Expression Transformation, use either of following methods:
2. Click on the expression transformation  icon in the transformation toolbar. Drag the pointer, which now appears as crosshairs , into the Workspace window to the right of the Source Qualifier transformation
3. Select TRANSFORMATION | CREATE.

- i. In the Create Transformation Dialog box, select the type of transformation from the pull-down list. Enter a name for the transformation.



- ii. Click on the Create button.
- iii. Click on the Done button.

**Note:** The expression transformation appears in **normal** mode.

4. Link the following ports from the **SQ\_Employees\_x** to the new expression transformation using any of the methods:

- i. Select LAYOUT | LINK COLUMNS.
- ii. Select the Link Columns icon in the toolbar .

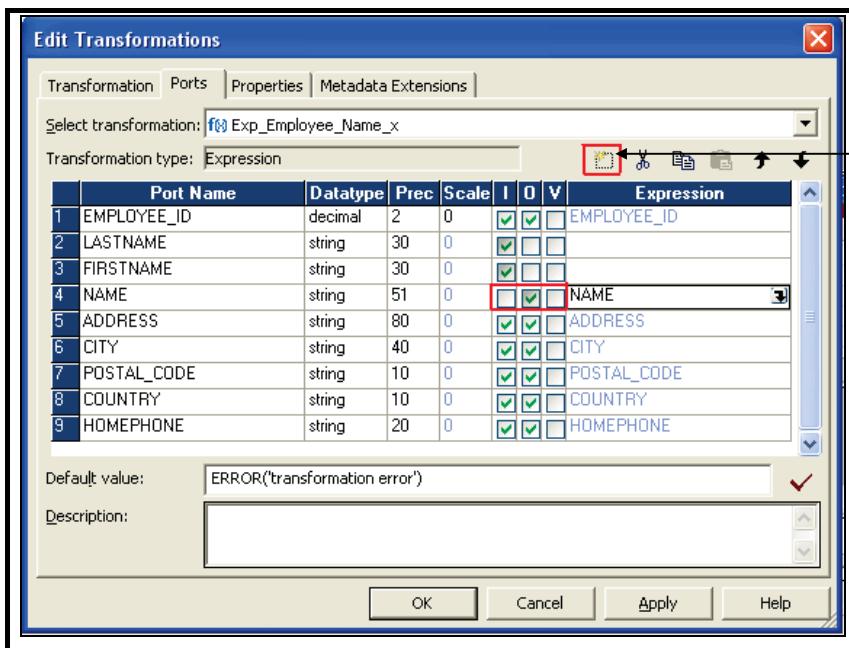
**HINT :** Hold down the Ctrl or Shift key and select the ports in the **SQ\_Employees\_x**. Drag them to an empty line on the expression transformation. When the mouse is released, not only will the port names (including data types, precision and scale) will be copied from **SQ\_Employees\_x**, but links connecting the ports between the two transformations will also be created.

### Edit Expression Transformation

1. Double click on the header of the Expression transformation to enter Edit mode.
2. Click the Rename button under the Transformation tab.
3. Rename the transformation as Exp\_Employee\_Name\_x.

**Note:** If the transformation was created via the **TRANSFORMATION | CREATE** menu instead of the transformation toolbar, the object will have a name.

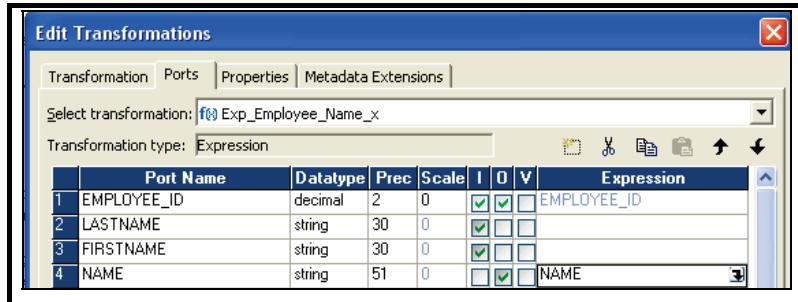
4. Click on the Ports tab.
5. Disable the output ports for FIRSTNAME and LASTNAME by removing the checkmark in the 'O' (output) column – this will define the port as input only.
6. Select the FIRSTNAME port and click on the  icon to add a new port and rename it to Name. This will cause the new port to be positioned immediately after FIRSTNAME.
7. Verify whether the data type is string and increase the precision to 51.
8. Disable the input port for NAME by removing the checkmark in the 'I' (input) column- this will define the port as output only.
9. The Edit Transformations box should look as shown below:



### Create Expression Formula

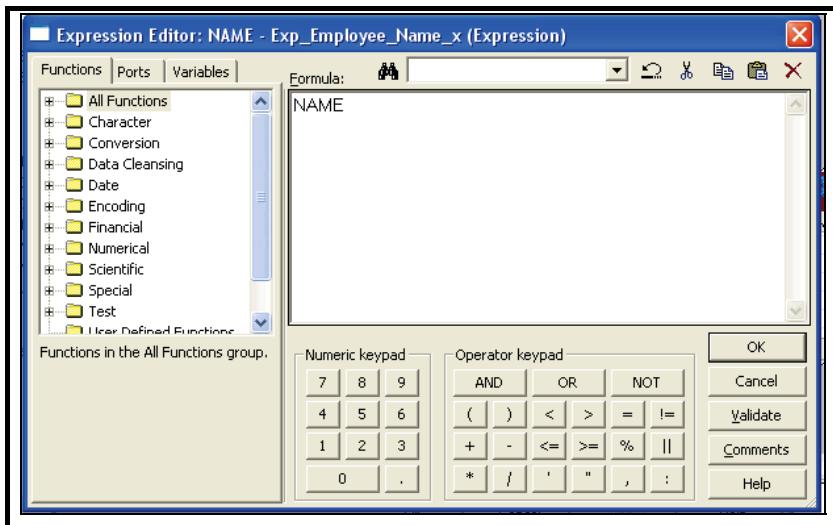
- Click the down arrow  in the expression column of the NAME port.

Add Port Icon



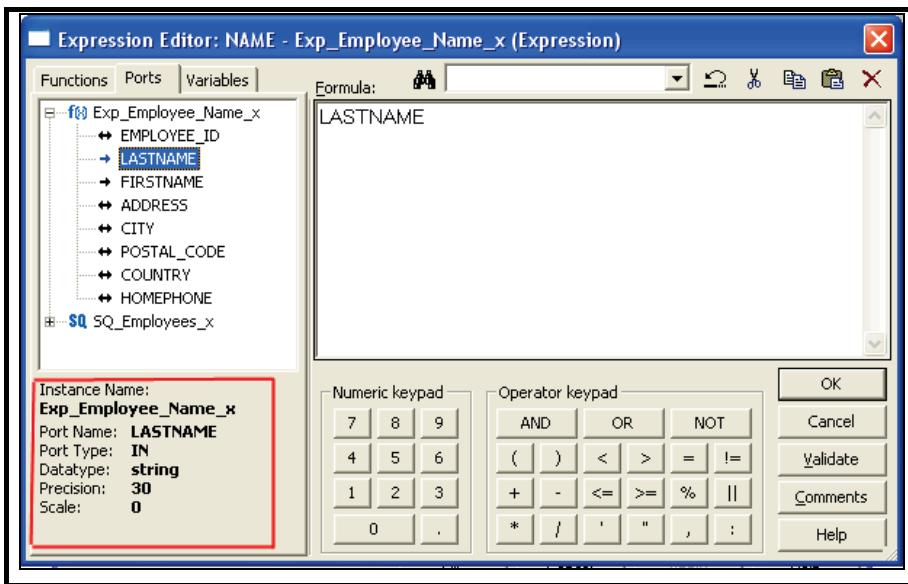
	Port Name	Datatype	Prec	Scale	I	O	V	Expression
1	EMPLOYEE_ID	decimal	2	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	EMPLOYEE_ID
2	LASTNAME	string	30	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
3	FIRSTNAME	string	30	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
4	NAME	string	51	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NAME

- Delete the text NAME in the Formula field in the Expression Editor.

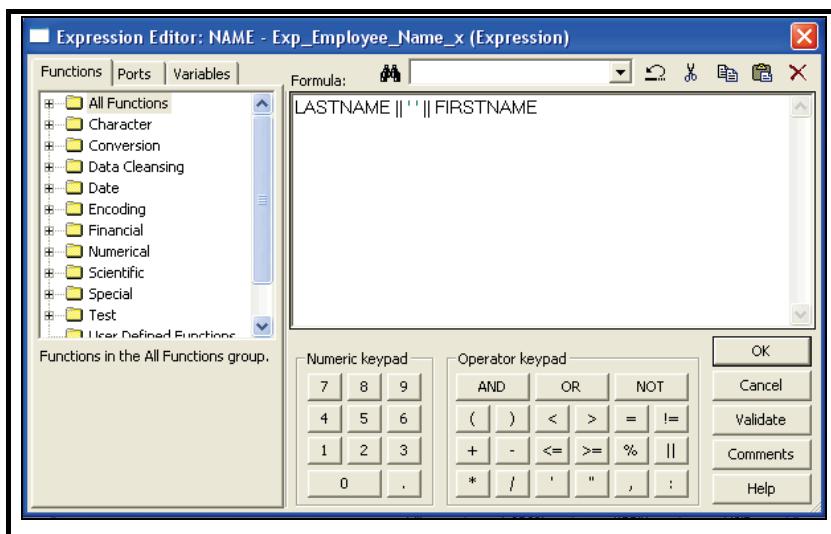


- Select the Ports tab as shown in the figure below.

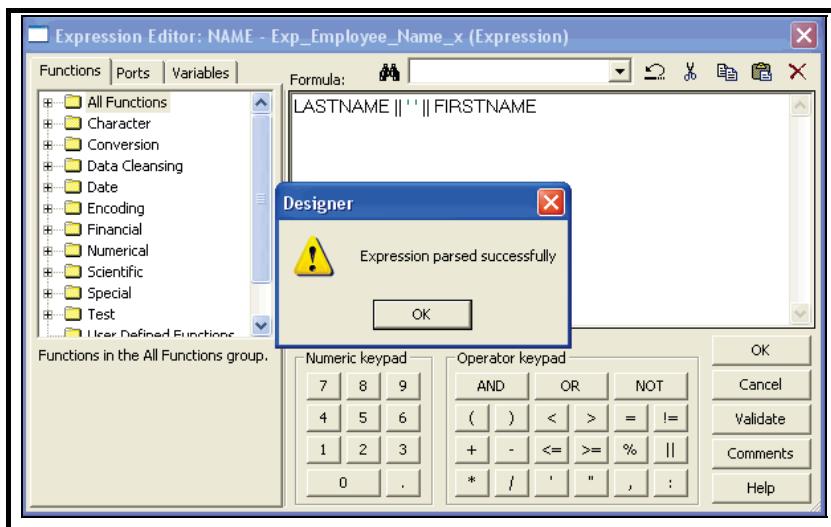
4. Double-click on the port LASTNAME and note its presence in the Formula field.  
Based on which port is selected, the port details appear under the 'Instance Name:' section of the ports tab.



5. Create the expression as shown below:

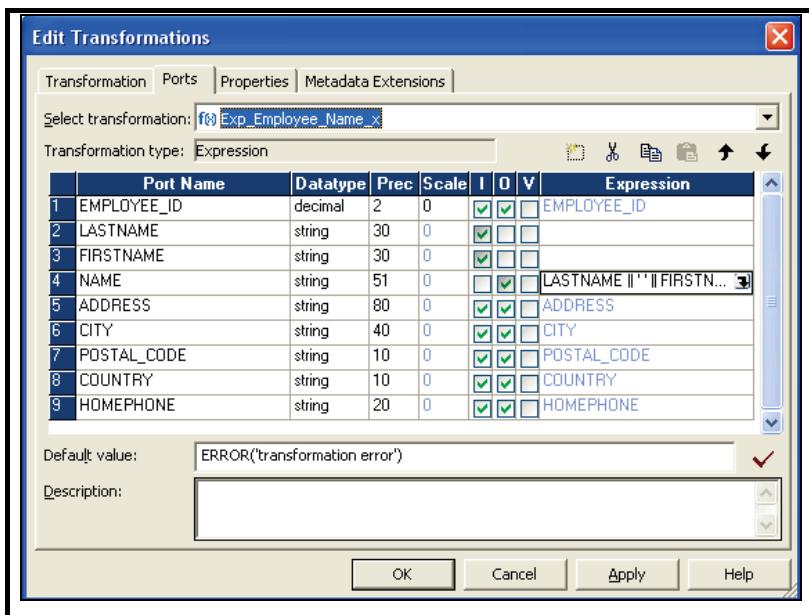


6. Click on Validate. This will parse the expression.



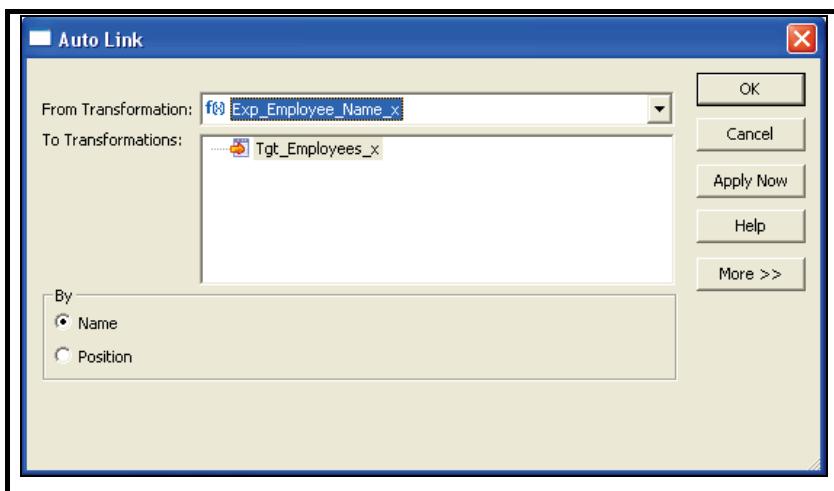
7. Click on OK if the expression is successfully parsed.

8. Click OK to close the Edit Transformation dialog box.
9. To save the changes to the repository, select REPOSITORY | SAVE or Ctrl-S.
10. The finished transformation will look like the following:



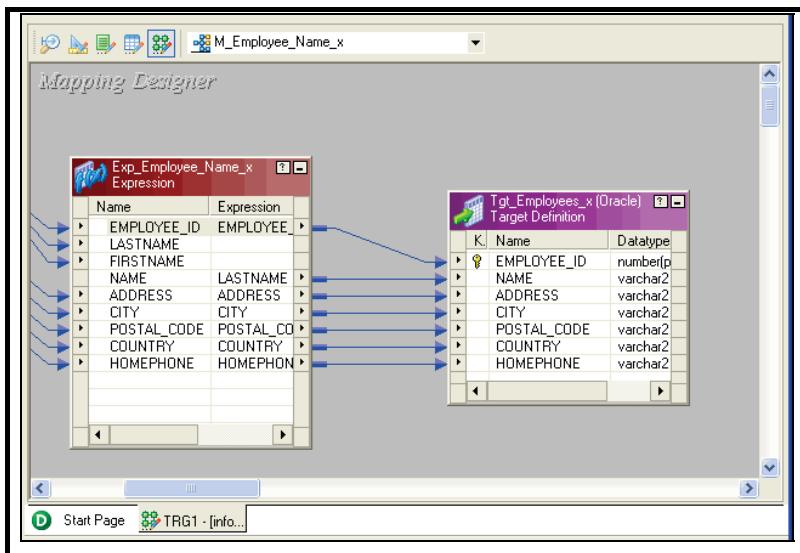
### Link Expression Transformation and Target

1. Drag the target definition from Navigator Window into the workspace.
2. Link columns of the Expression Transformation to the target definition:
  - i. Select LAYOUT | AUTOLINK.
  - ii. In the Autolink dialog box, select the From Transformation and To Transformation.

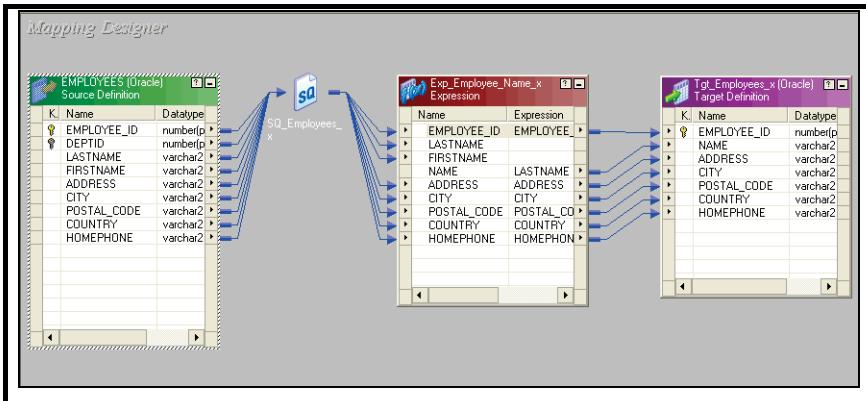


iii. Click on Apply Now and OK.

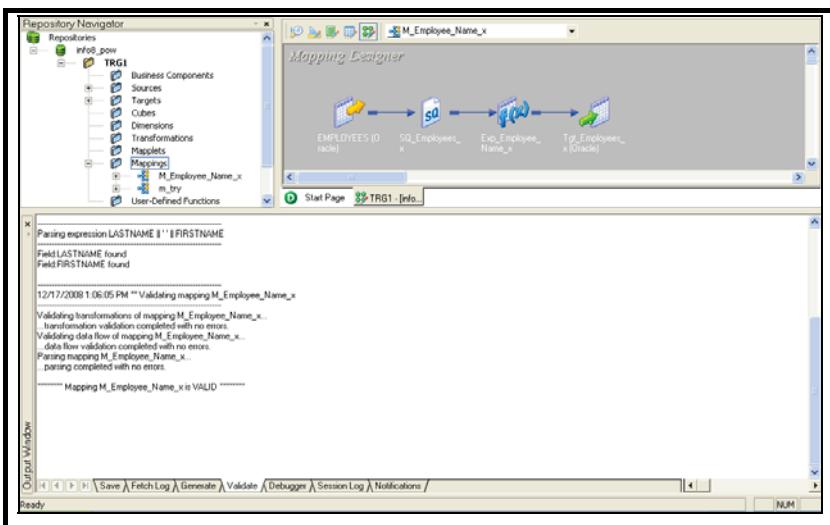
3. Notice the links between expression transformation and target table.



4. The mapping is now complete and should look like the figure shown below:



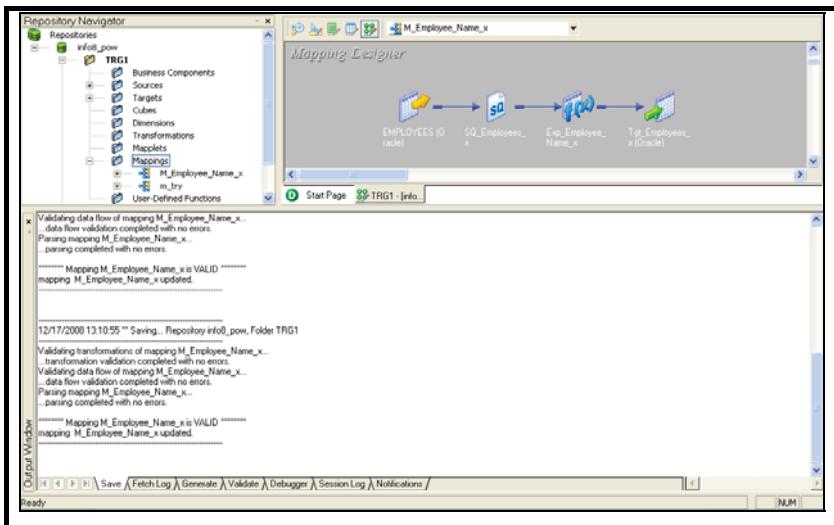
#### 5. Select MAPPINGS | VALIDATE to validate the mapping.



#### 6. Select REPOSITORY | SAVE or press Ctrl-S to save the changes to the repository.

**Note :** Every time a repository save is executed, a series of validation checks are performed on what has been changed.

#### 7. View the results of the Validation by locating Save tab of the Output window, at the bottom of Designer.



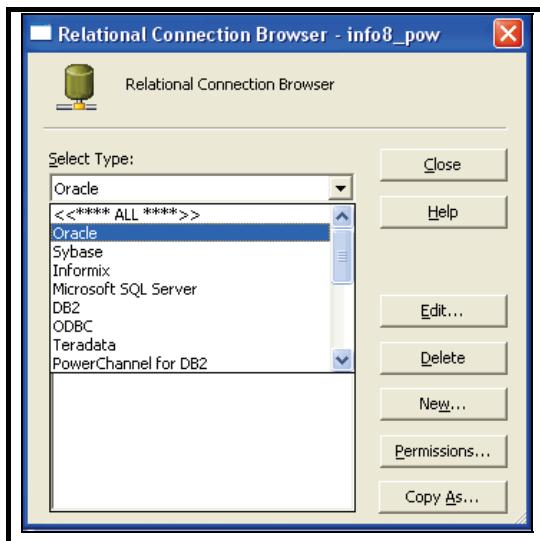
**Note:** If the Validation results show 'INVALID', locate the last time stamp when the save repository was executed and scan for the first error. The series of validation checks will display all of the errors. Rectify the errors and validate the mappings again, until the mapping is valid.

## Lab 5.1. Lab 3-1 Creating Workflow

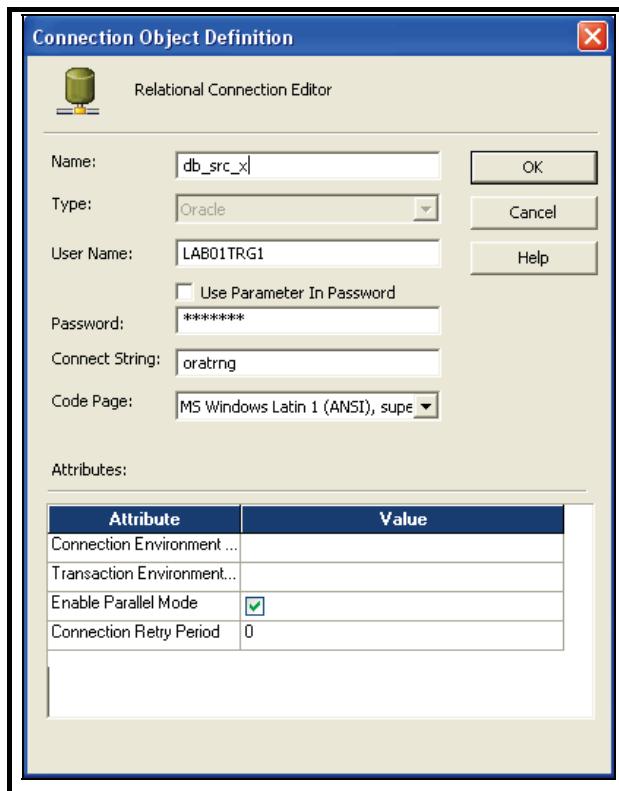
<b>Goals</b>	<ul style="list-style-type: none"><li>• Create database connections for sources and targets</li><li>• Learn how to use Workflow Manager.</li><li>• Create a simple Workflow</li><li>• Create a session task and start task</li><li>• Link tasks</li></ul>
<b>Time</b>	30 Minutes
<b>Lab Setup</b>	<ul style="list-style-type: none"><li>• Successful connection to the repository using Workflow Manager</li><li>• A valid mapping created</li></ul>

### Create Relational Connection

1. Select START | PROGRAMS | INFORMATICA POWERCENTER | CLIENT | WORKFLOW MANAGER.
2. Connect to the repository given by your Instructor.
3. Locate the assigned Studentx folder and open it.
4. Select CONNECTIONS | RELATIONAL.



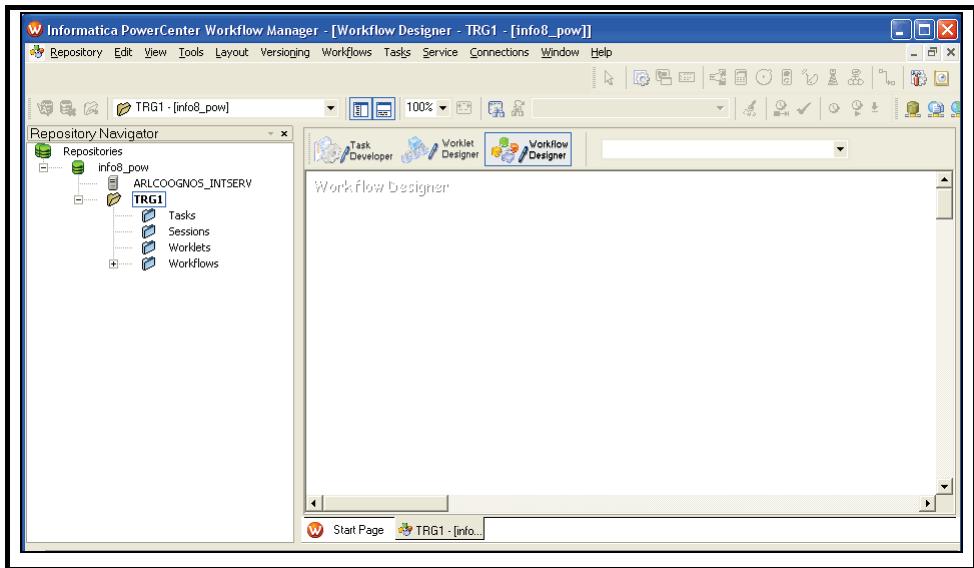
5. Select the database type from the dropdown and click on New.
6. In the Connection Object Definition box, enter the name db\_src\_x, username, password and connection string to connect to the source database.  
**Note :** Use the same login information used for creating the source definition in Designer.



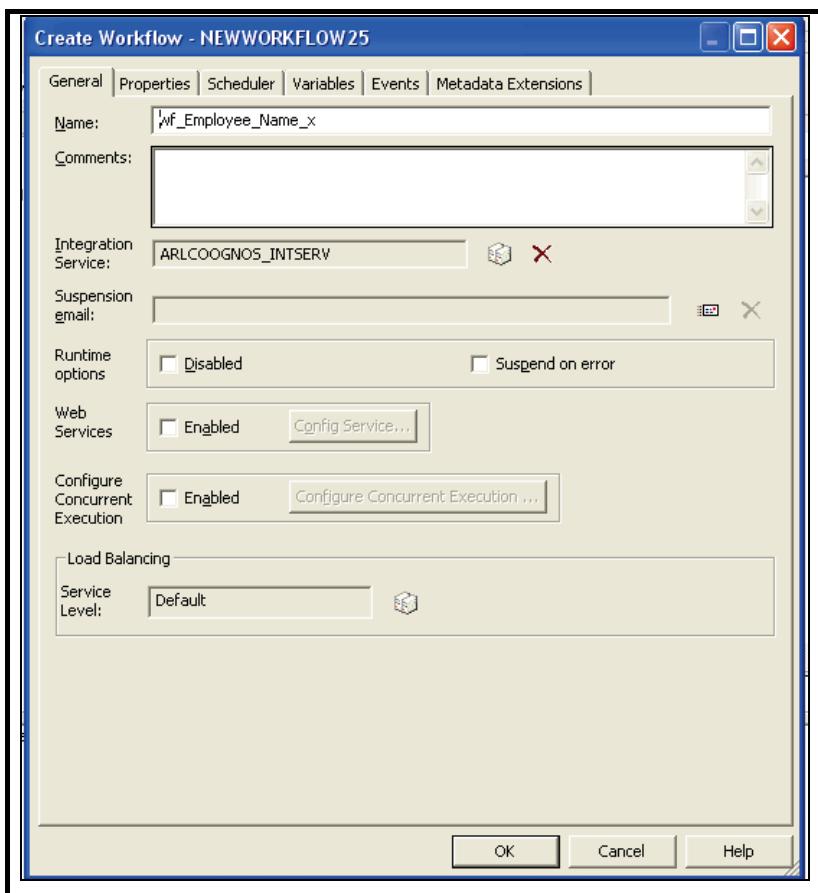
7. To create a target database connection, repeat steps 1 and 2 and in the Connection Object Definition box, enter the name db\_tgt\_x, username, password and connection string to connect to the target database db\_tgt\_x.  
**Note:** Use the same login information used for creating the target definition in Designer.
8. Click **CLOSE** to close the Relational connection Browser.

**Create a Workflow**

1. Switch to Workflow Designer.
  - i. Select TOOLS | WORKFLOW DESIGNER  
(OR)
  - ii. Click on the button as shown below



2. Select WORKFLOWS | CREATE.
3. Enter the name of the Workflow as wf\_Employee\_Name\_x in the Name box under the General tab, (x represents the assigned student number).



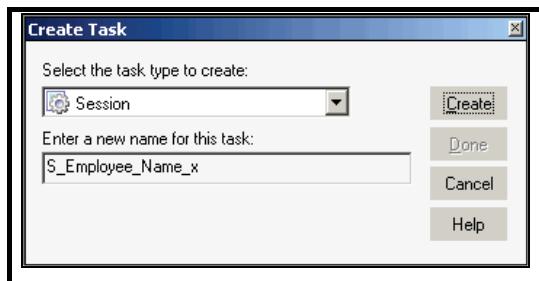
4. Click on the OK button. The Start Task is added to each new workflow by default
5. Select Repository | Save or Ctrl-S to save.

#### Edit the Start Task

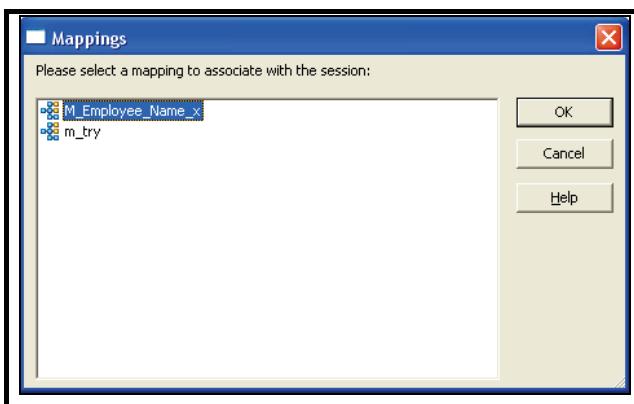
1. Double Click on the Start task
2. Click the Rename button
3. Type Start\_Employee\_Name\_x in the Rename Task dialog box.

**Create a Session Task**

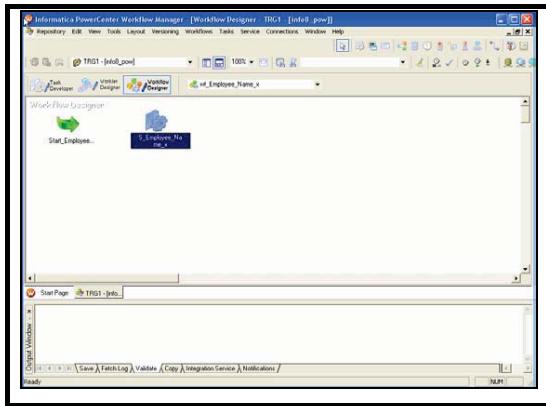
1. Click on TASKS | CREATE and In the Create task dialog box enter a name for the task as S\_Employee\_Name\_x



2. Associate a mapping with the task. The mapping is M\_Employee\_Name\_x, which you just created.

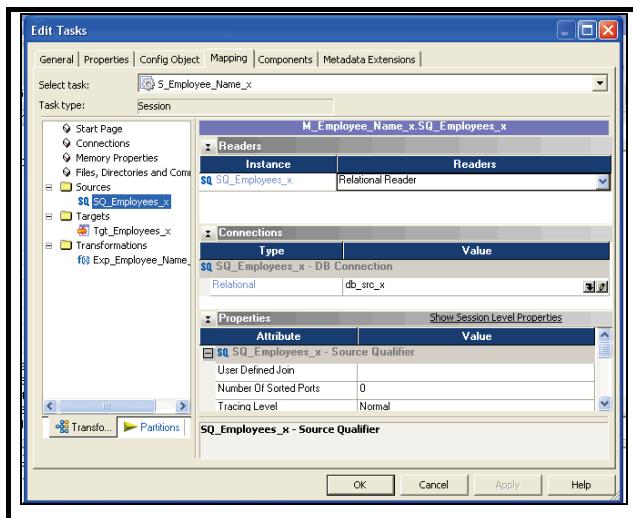


3. Click on OK in Mappings dialog box and Done in Create Task Dialog box.
4. A Session Task appears in the Workspace. Select REPOSITORY |SAVE.

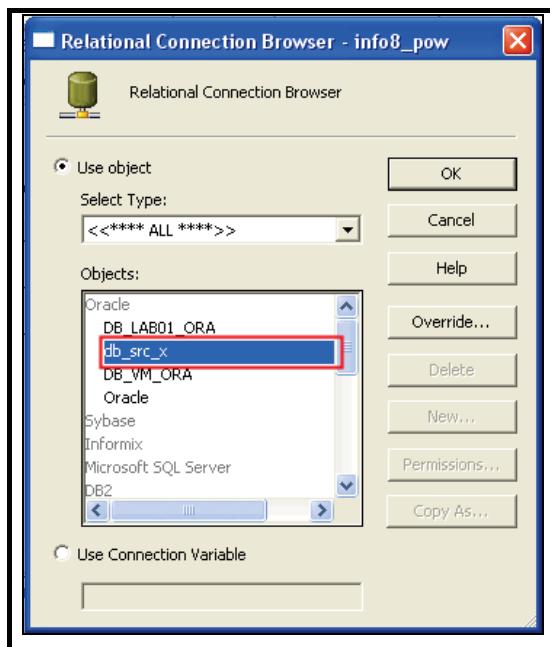


### Edit Session Task

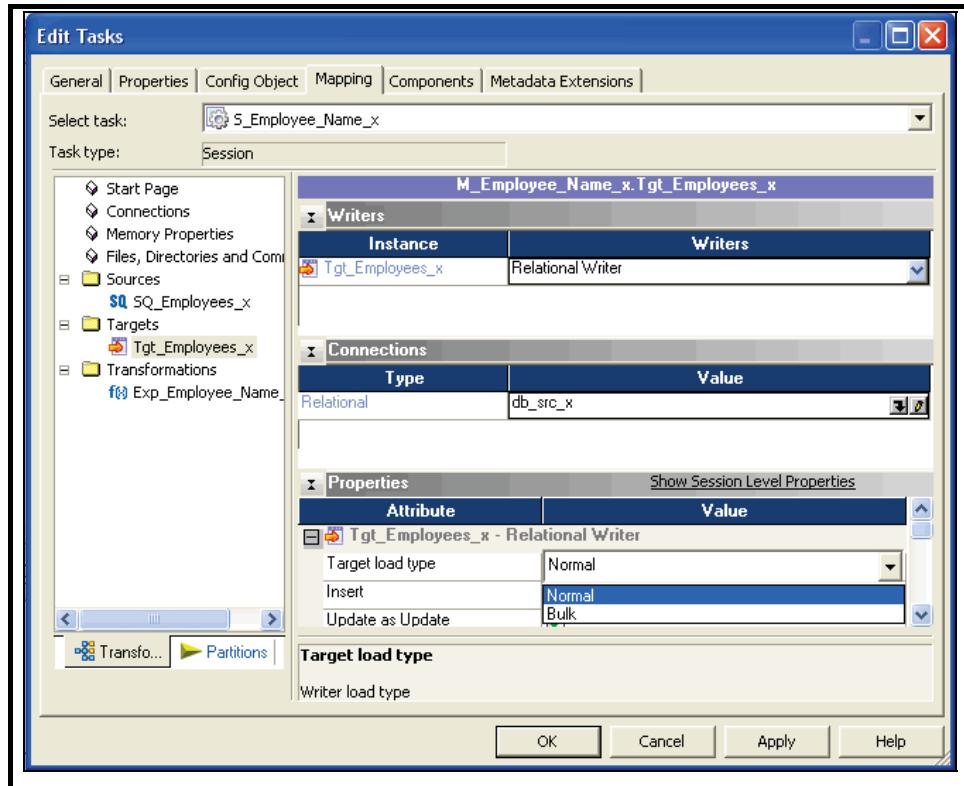
1. Double-click on the S\_Employee\_Name\_x Session Task in the Workspace and click on the Mapping tab.
2. Under the Sources folder, click on the down arrow to open the Relational Connection Browser as shown in the figure below.



3. In the Relational Connection Browser, select the database type from the dropdown and select the source database connection just created, from the Objects box.

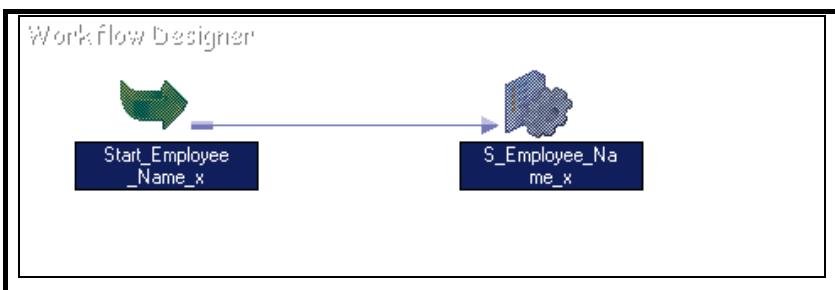


4. Under the Target folder, click on the down arrow to open the Relational Connection Browser and select the target database connection.
5. Under Properties, Select **Normal** for the **Target load type**.

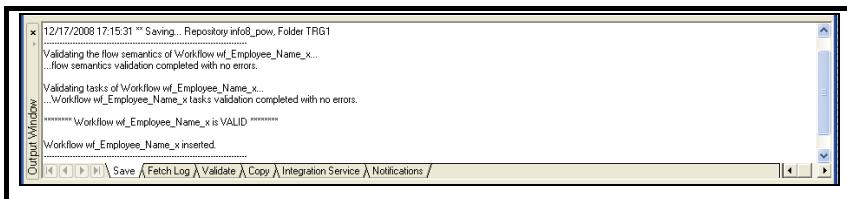


#### Link Workflow Tasks

1. Locate the Link icon on the right side of the Task Toolbar
2. Link the Start\_Employee\_Name and the S\_Employee\_Name\_x tasks.



3. Toggle off the 'link mode' by clicking again on the Link icon or single click on one of the objects.
4. Save changes to the repository.
5. Select WORKFLOW | VALIDATE to validate the workflow.
6. Locate the save tab in the Output Window at the bottom of the Workflow Manager and view the results of the validation checks.



7. If there are any errors fix them and perform the WORKFLOW | VALIDATE command from the main menu. The results will be located in the Output Window's Validate tab.
8. Repeat the process until the Workflow is valid.

## Lab 6.1. Lab 3-2 Start and Monitor Workflows

<b>Goals</b>	<ul style="list-style-type: none"> <li>• Different methods to start a Workflow</li> <li>• Monitor a Workflow</li> <li>• Create a simple Workflow</li> <li>• Create a session task and start task</li> <li>• Link tasks</li> </ul>
<b>Time</b>	15 Minutes
<b>Lab Setup</b>	<p>Successful connection to the repository using Workflow Manager</p> <p>A valid Workflow created</p>

### Start and Monitor The Workflow

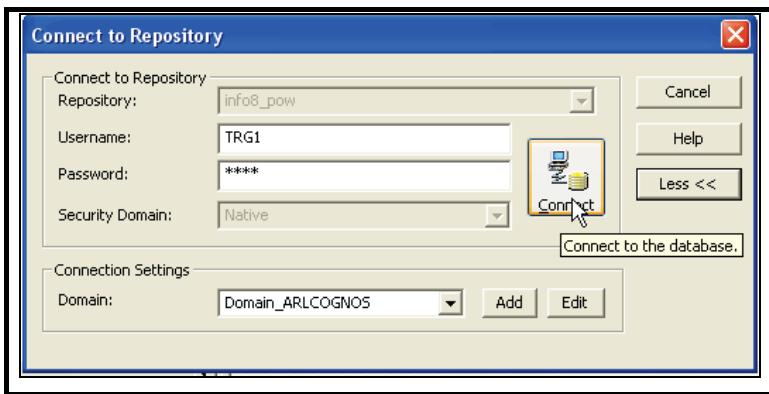
1. If the Workflow is valid, it is ready for execution. In the Workflow Designer, use one of the following methods to start the wf\_Employee\_Name\_x Workflow:
  - i. Select WORKFLOWS | START WORKFLOW.
  - ii. Right-click in the Workspace and select Start Workflow or
  - iii. Right-click on the wf\_Employee\_Name\_x Workflow in the Navigator Window and select Start Workflow.
  
2. To monitor a Workflow the Workflow Monitor must be opened. This is opened automatically when a workflow is executed. If this does not happen perform the following steps to start the Workflow Monitor
  - i. Select Start | Programs | Informatica PowerCenter Client | Workflow Monitor.
  - ii. To connect to Repository use one of the following methods:

In the Workflow Monitor Select REPOSITORY | CONNECT; or

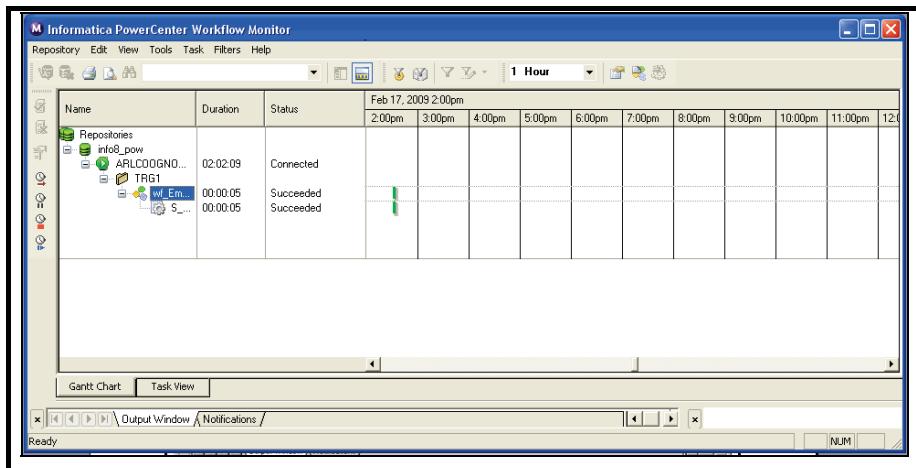
Click on the  icon in the toolbar; or double-click on Repository in the Navigator Window. The **Connect To** Repository box appears.

**Note:** Enter repository, server and login details as given by your Instructor.

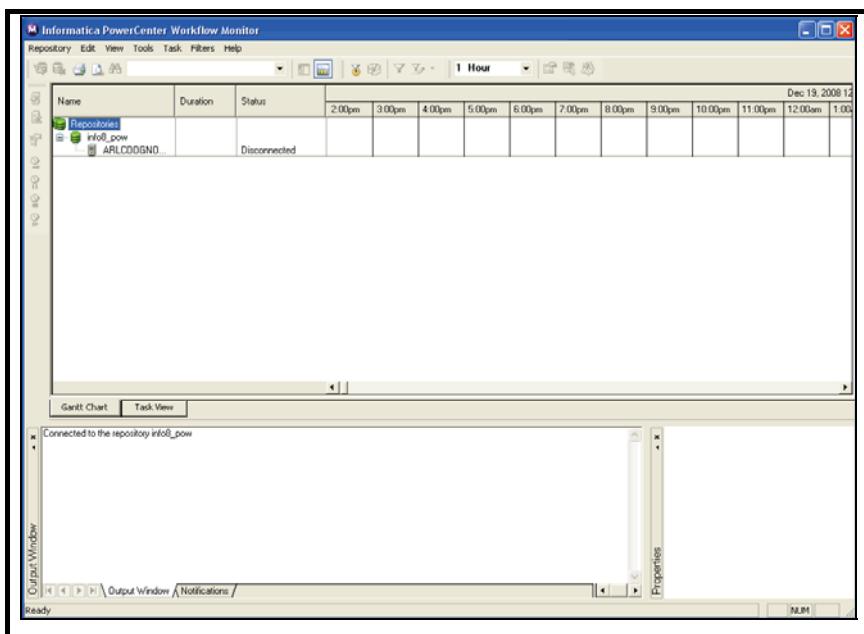
- iii. Select the repository from the Repository pull-down list.
- iv. Enter the Username in the Username box.
- v. Enter the Password in the Password box.
- vi. Select the Domain from the domain pull-down list and connect to database.



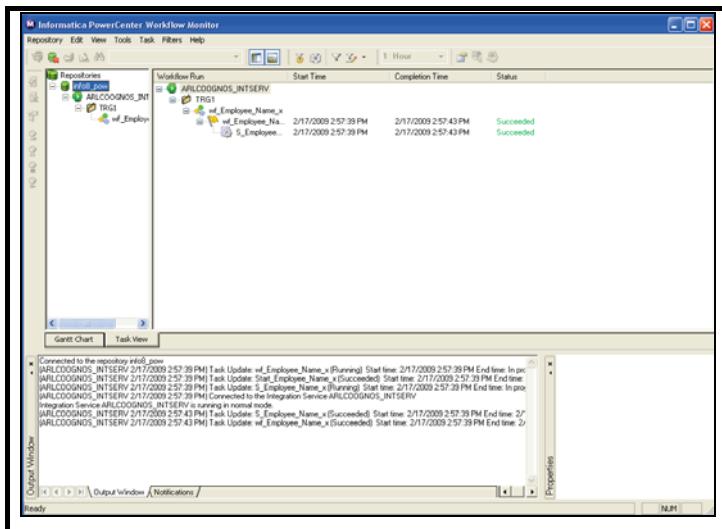
3. Double click on the folder to view Workflow sessions.
4. You get two views, the Gantt Chart View and the Task View.



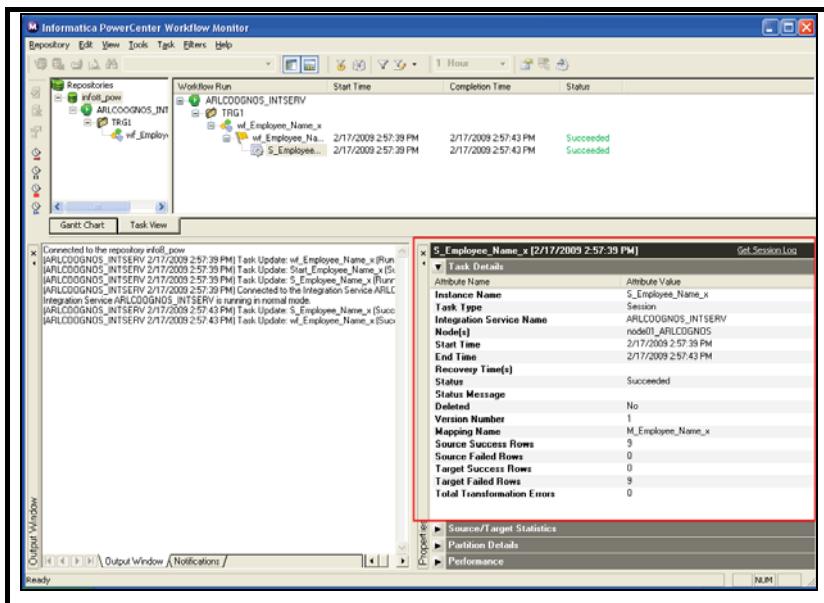
5. Select the Gantt View tab. This view displays details about workflow runs in chronological format. It displays the following information as shown below:



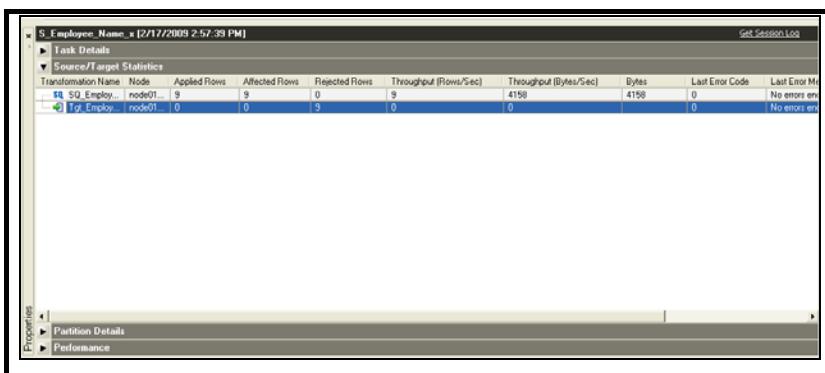
6. Select the Task View. This view displays details about workflow runs in a report format. The Status column gives the following information:



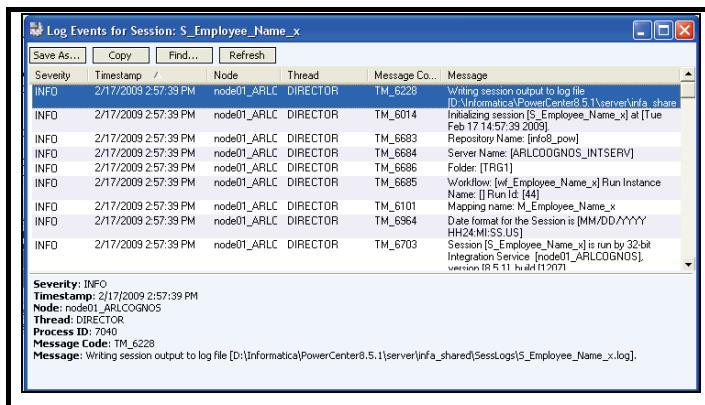
- i. A Succeeded status if the PowerCenter Server was able to successfully complete a Workflow or Task.
  - ii. A Failed status may occur if the PowerCenter Server failed the Workflow or Task due to fatal processing errors.
  - iii. A Running status if the PowerCenter Server is still processing is still processing Workflow or Task.
7. View the Session properties by doing one of the following:
- i. Right click on the Session selected and select **Get Run Properties**; or
  - ii. Click on the Session Properties icon; or simply,
  - iii. In Properties window select **Task Details**.
8. The Properties tab of the **S\_Employee\_Name\_x** dialog box opens. The Session should display the number of Target Success Rows as shown below:



9. Click on the Source/Target Statistics tab in Properties window. More detail on the number of rows handled by the Server are shown here:



- i. Applied rows are rows the Informatica Server successfully produced and applied to the target without errors.
- ii. Affected rows are generated by the Server and 'affected to' (or accepted by) the target.
- iii. Rejected rows are either those read rows that the Server dropped during the transformation process, or, the rows that were rejected when writing to the target.
  
10. View Session Log to determine what occurred during the system run. To view detailed Session information, do one of the following:
  - i. Right-click on the Session in the Name column and select Get Session Log.
  - ii. Select the Session name and click on the  icon.
  - iii. Select Get Session Log in Properties window



**Final Output**

1. View the final output using the following SQL statement:

```
SQL> SELECT * FROM tgt_Employees_x;
```

```
SQL> /
```

EMPLOYEE_ID	NAME	ADDRESS
1	Davolio Nancy	507 - 20th Ave. E.Apt.2A
2	Fuller Andrew	908 W. Capital Way
3	Leverling Janet	722 Moss Bay Blvd.
4	Peacock Margaret	4110 Old Redmond d.
5	Buchanan Steven	14 Garrett Hill
6	Suyana Michael	Coventry House Miner Rd.
7	King Robert	Edgeham Hollow Winchester Way
8	Callahan Laura	4726 - 11th Ave. N.E.
9	Dodsworth Anne	7 Houndstooth Rd.

```
9 rows selected.
```

## Lab 7.1. Lab 4-1 Sales Summary

<b>Goals</b>	<ul style="list-style-type: none"> <li>• Create a mapping that gives the summary of all sales by item description, state, and month           <ul style="list-style-type: none"> <li>➢ Use multiple sources</li> <li>➢ Create and use Expression and Aggregator transformations in the mapping</li> <li>➢ Use functions in the Expression transformation</li> <li>➢ Group columns in the Aggregator transformation</li> </ul> </li> </ul>
<b>Time</b>	180 Minutes
<b>Lab Setup</b>	Informatica PowerCenter Client and login details for connecting to Model repository

### Background

The Inventory system maintains details of items, stock available, orders placed and customer related information. There are various requirements related to sales of an item. The company requires sales summary information.

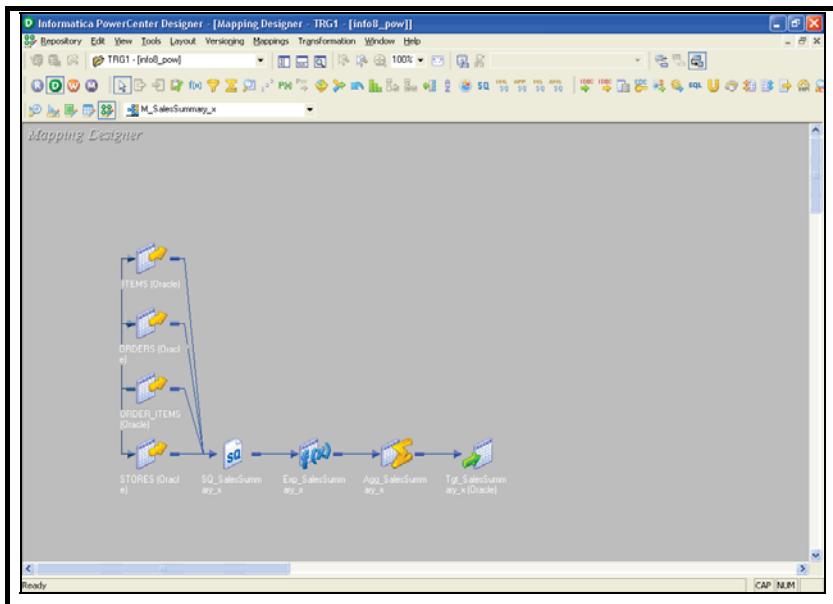
### Solution

- To get a summary of sales by item description, state and month,
- Collect data from various relational tables sources to consolidate the information.
- Create a relational target containing the summary wise details are created.

TRANSFORMATION NAME	TYPE	DESCRIPTION
ITEMS ORDER_ITEMS ORDERS STORES	Relational Source Definition	Source definitions
SQ_SALES_SUMMARY_x	Source Qualifier	Data source qualifier for all source tables
EXP_SALES_SUMMARY_x	Expression	Link ITEM_DESC, PRICE, QUANTITY, DATE_ENTERED, and STATE from the Source Qualifier. Create a MONTH and YEAR port, and extract the month and year from the DATE_ENTERED.

AGG_SALES_SUMMARY_x	Aggregator	Link all ports except the DATE_ENTERED into the Aggregator. Create ports to hold the TOTAL SOLD and TOTAL PRICE. Create expressions in those ports to calculate the total quantity sold and the total price. You will want to select Group By for ITEM_DESC, STATE, MONTH, AND YEAR.
TGT_SALES_SUMMARY_x	Relational Target Table	Target definition

### Mapping Layout



## Final Output

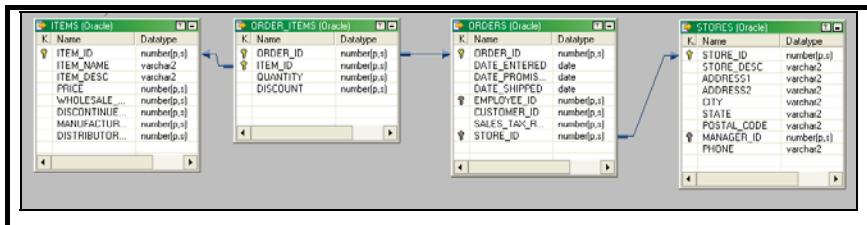
```
SQL> select * from tgt_salessummary_x;
```

DESCRIPTION	TOTAL SOLD	TOTAL PRICE	MONTH	YEAR	ST
Air Regulators	11	3135	April	1998	HI
Air Regulators	10	2535	December	1998	HI
Air Regulators	8	1910	January	1998	HI
Air Regulators	4	920	June	1998	HI
Air Regulators	6	1880	November	1998	HI
Air Tank	7	1525	February	1998	HI
Air Tank	3	829	March	1998	HI
Air Tank	2	358	May	1998	HI
Air Tank	6	1550	November	1998	HI
Buoyancy Compensation	6	1910	April	1998	HI
Buoyancy Compensation	4	1420	August	1998	HI
DESCRIPTION	TOTAL SOLD	TOTAL PRICE	MONTH	YEAR	ST
Buoyancy Compensation	4	1720	November	1998	HI
Buoyancy Compensation	7	2305	October	1998	HI
Small Instruments	13	736.8	April	1998	HI
Small Instruments	2	36	August	1998	HI
Small Instruments	3	968	December	1998	HI
Small Instruments	10	1475	January	1998	HI
Small Instruments	9	2381	June	1998	HI
Small Instruments	3	87	May	1998	HI
Small Instruments	2	376	November	1998	HI
Small Instruments	9	462.7	October	1998	HI
Small Instruments	2	36	September	1998	HI
DESCRIPTION	TOTAL SOLD	TOTAL PRICE	MONTH	YEAR	ST
Tools	2	82	April	1998	HI
Tools	9	519.8	August	1998	HI
Tools	7	350.8	December	1998	HI
Tools	5	316	January	1998	HI
Tools	11	1299.5	March	1998	HI
Tools	2	119.9	May	1998	HI
Tools	3	154.9	October	1998	HI

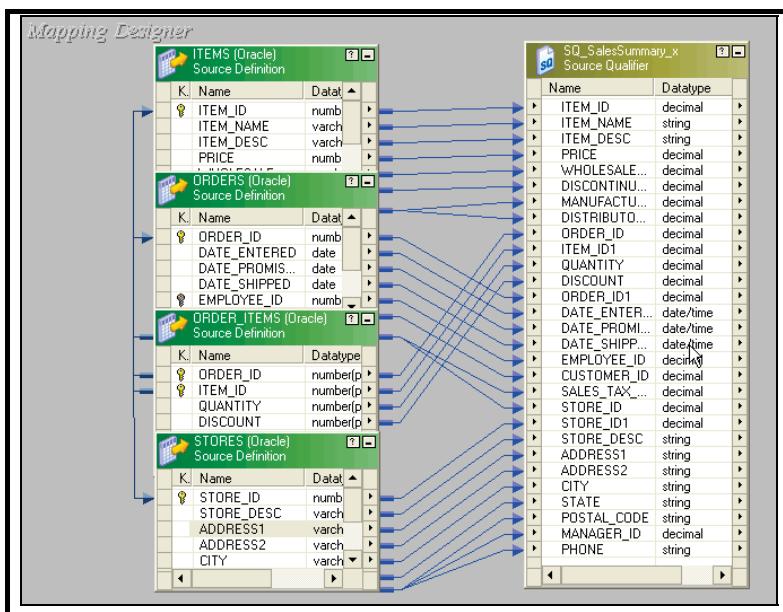
29 rows selected.

### Problem Solution

1. Import the tables ITEMS, ORDERS, ORDER\_ITEMS, and STORES tables .
2. Observing the key relationships indicated by link lines between the tables, shown below.



3. Create a target schema with name Tgt\_SalesSummary\_x having following columns:  
DESCRIPTION,TOTAL SOLD,TOTAL PRICE,MONTH,YEAR,STATE .
4. Create a new mapping with name M\_SalesSummary\_x.
5. Create Source Qualifier Transformation using  with name SQ\_SalesSummary\_x.



6. Create the Expression transformation with name Exp\_SalesSummary\_x and Link the following ports from the SQ\_SalesSummary\_x to the Expression Transformation:

**ITEM\_DESC, PRICE, QUANTITY, DATE\_ENTERED, and STATE**

**HINT:** Select the Link Columns icon in the toolbar for Auto-link.



7. Make DATE\_ENTERED an input-only port .
8. Add new ports MONTH,YEAR as output port only.
9. Configure the MONTH port by entering the expression:

**TO\_CHAR(DATE\_ENTERED, 'Month')**

10. Configure the YEAR port by entering the expression :

**TO\_CHAR(DATE\_ENTERED, 'YYYY')**

11. Create an Aggregator transformation with name Agg\_SalesSummary\_x.
12. Link the following columns from Exp\_SalesSummary\_x to the Aggregator transformation:

**ITEM\_DESC, PRICE, QUANTITY, STATE, MONTH, and YEAR**

13. Make PRICE and QUANTITY input-only ports.
14. Add new ports for the TOTAL\_QTY and TOTAL\_PRICE.They will be output-only ports with expressions.

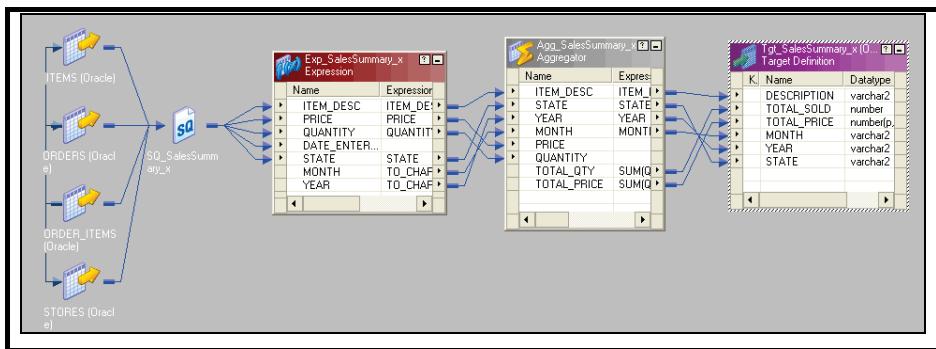
**TOTAL\_QTY : SUM(QUANTITY)**

**TOTAL\_PRICE: SUM(QUANTITY \* PRICE)**

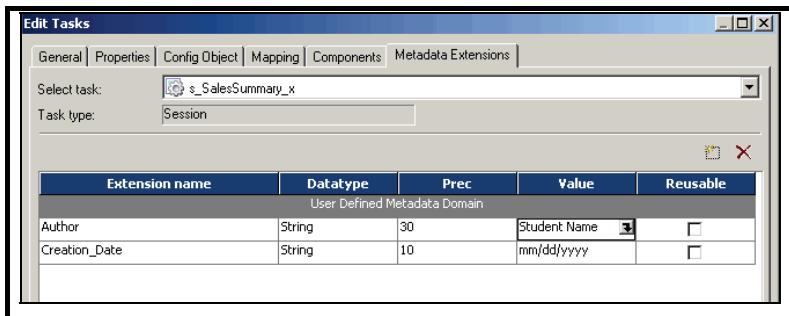
15. Check the GroupBy boxes on the lines for ITEM\_DESC, STATE, MONTH, and YEAR These are the columns by which we want to summarize.

**Note :** The order of **GroupBy** ports should be in the sequence as given above.

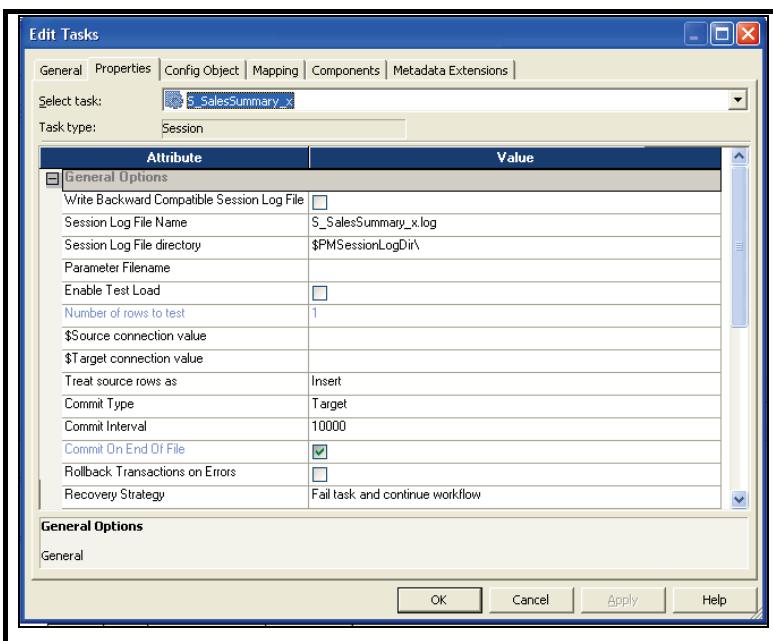
16. Link the appropriate ports from Agg\_SalesSummary\_x to Tgt\_SalesSummary\_x.
17. If the mapping is invalid, make changes and validate the mapping again till valid.
18. The Final mapping will look like the one given below.



19. Create a Workflow by name wf\_SalesSummary\_x in Workflow Manager.
20. Under the Properties tab, note name of the **Workflow Log File Name**: wf\_SalesSummary\_x.log
21. To create non-reusable, local Metadata Extensions, click on the Metadata Extensions tab
  - i. Enter Creation\_Date and Author into the Extension Name.
  - ii. Do not select Reusable.



- iii. Enter appropriate value for Creation\_Date [using the syntax = mm/dd/yyyy] and Author.
22. Create a Session task by name **s\_SalesSummary\_x** task and Select the **M\_SalesSummary\_x** mapping from the list of valid mappings.
23. Enter the appropriate description for the session in the General tab.
24. Under the Properties tab, you can enter session log file name, session log file directory, and other general session settings.



25. Select the appropriate Source Database Connection and Target Database Connection
26. Run and monitor the Workflow.
27. Verify the results.

## Lab 8.1. Lab 5-1 New Customer

<b>Goals</b>	<ul style="list-style-type: none"><li>• Create a mapping which reads from a flat file and creates a relational table consisting of new customers</li><li>• Analyze a fixed width flat file</li><li>• Configure a Connected Lookup transformation</li><li>• Use a Filter transformation to exclude records from the lookup transformation pipeline</li></ul>
<b>Time</b>	120 Minutes
<b>Lab Setup</b>	Successful Connection to the repository using PowerCenter Client

### Background

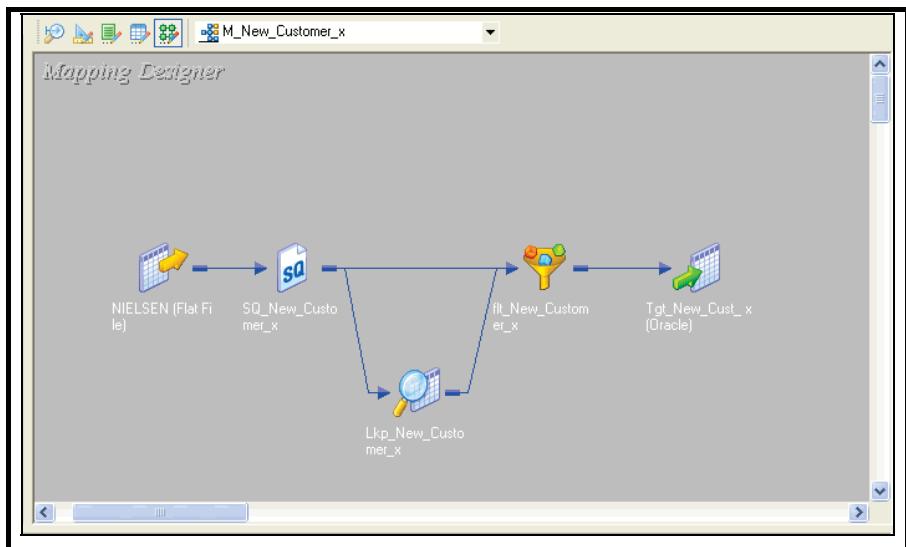
The Marketing department is holding a special promotion for potential customers. The company has purchased an industry listing from the Nielsen Research Company of their target market, which includes potential new customers as well as many of their current customers.

Because this promotion is extended to new customers only, the company must first exclude existing customers from this listing before the promotional mailing is sent.

### Solution

- Build a new target table that will contain data only for the new customers
- Use the Informatica tools to import and analyze the source files and create a target database table
- Use a Lookup Transformation to compare the Customer\_ID from the flat file and the relational table
- Use a Filter transformation to test the result of the lookup and filter out matches. When no match is found for a given CUSTOMER\_ID, the filter allows the potential customer record into the relational table
- The relational table will contain the list of potential customers, which can now be used for the promotional mailing

### Mapping Layout



TRANSFORMATION	TYPE	DESCRIPTION
NIELSEN	Source	Flat file source definition
SQ_NIELSEN_X	Source Qualifier	Data source qualifier for flat file
LKP_NEW_CUSTOMER_X	Lookup	Check the CUSTOMERS table in the source database for occurrences of companies that are listed in the flat file. The condition will check NIELSEN.CUST_ID against CUSTOMERS.CUSTOMER_ID
FIL_NEW_CUST_X	Filter	Pass through all records from NIELSEN that do not match up with the CUSTOMER table (CUST_ID has no corresponding CUSTOMER_ID)
TGT_NEW_CUST_X	Target	Target definition (Relational Table)

### Final Output

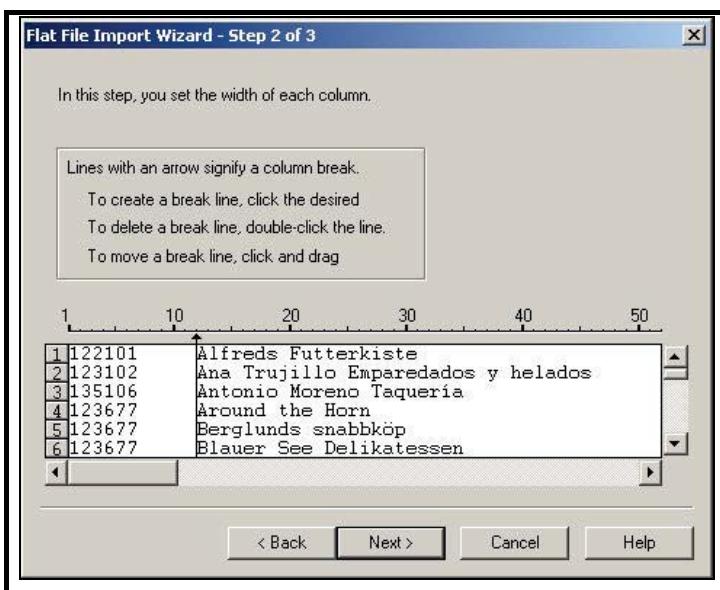
SQL> select * from tgt_new_cust_x;		
CUST_ID	COMPANY_NAME	ADDRESS1
93677	The Cracker Box	55 Grizzly Peak Rd.
103677	Suprêmes délices	Boulevard Tirou, 255
113671	Magazzini Alimentari Riuniti	Via Ludovico il Moro 22
113677	The Big Cheese	89 Jefferson Way
121077	Wellington Importadora	Rua do Mercado, 12
121177	White Clover Markets	305 - 14th Ave. S.
121277	Wilman Kala	Keskuskatu 45
122377	Tradição Hipermercados	Av. Inês de Castro, 414
123077	Great Lakes Food Market	2732 Baker Blvd.
123177	Folk och fä HB	Åkersgatan 24
123277	Folies gourmandes	184, chaussée de Tournai
123377	Godos Cocina Típica	C/ Romero, 33
123456	Galería del gastrónomo	Rambla de Cataluña, 23
123477	Wolski Zajazd	ul. Filtrowa 68
123479	Furia Bacalhau e Frutos do Mar	Jardim das rosas n. 32
123574	Trail's Head Gourmet Provisioners	722 DaVinci Blvd.
123577	Chop-suey Chinese	Hauptstr. 29
123579	GROSELLA-Restaurante	5ª Ave. Los Palos Grandes
123612	Uaffeljernet	Snagslæget 45
123613	Laughing Bacchus Wine Cellars	1900 Oak St.
123614	Lazy K Kountry Store	12 Orchestra Terrace
-----		
-----		
-----		
723672	Split Rail Beer & Ale	P.O. Box 555
723677	Hungry Owl All-Night Grocers	8 Johnstown Road
823672	Ernst Handel	Kirchgasse 6
84 rows selected.		

**Problem Solution**

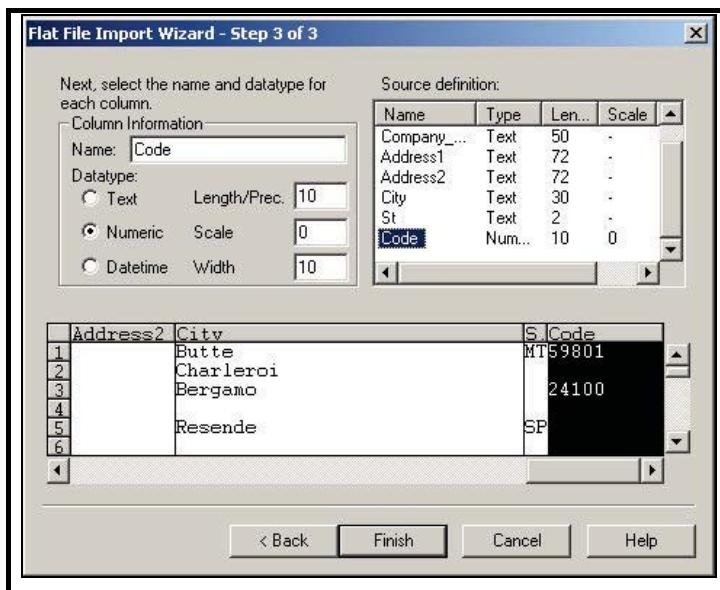
1. Import the Nielsen.dat flat file definition into the repository.

**HINT:** Be sure to set the Files of type: to All files (\*.\*) from the pull-down list, before clicking on **OK**.

- i. Set the following options in the Flat File Wizard:
  - ii. Select Fixed Width and check the Import field names from first line box.
  - iii. This option will extract the field names from the first record in the file.
  - iv. Create a break line or separator between the fields.



v. Structure of NIELSEN.DAT flat file is as shown below:



- Change field name St to State and Code to Postal\_Code and click finish.

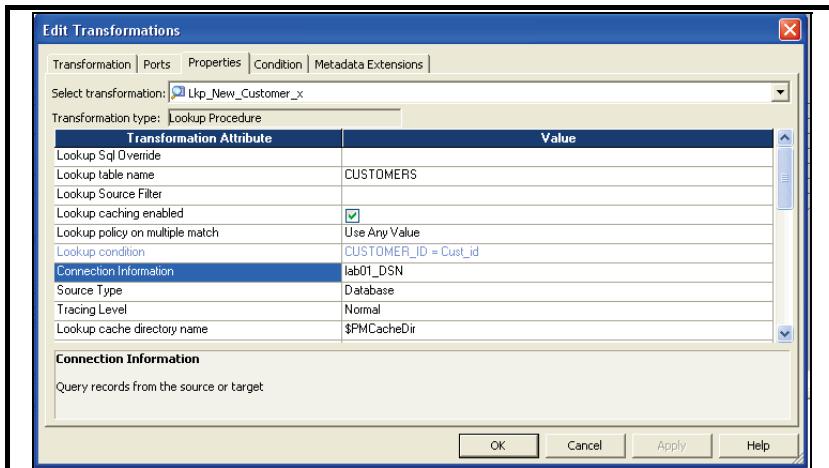
**NOTE:** The physical data file will be present on the Server. At runtime, when the Server is ready to process the data (which is now defined by this new source definition called NIELSEN.dat) it will look for the flat file that contains the data in Nielsen.dat.

- Name the new source definition NIELSEN. This is the name that will appear as metadata in the repository, for the source definition.
  - Create a target definition based on the structure of the source file definition with name Tgt\_New\_Cust\_x.
- Hint:** From the Edit table properties in Target designer, change the database type to oracle.
- Create a new mapping with the name M\_New\_Customer\_x
  - Create Lookup transformation with name Lkp\_New\_Customer\_x.
  - Identify the Lookup table in the Lookup transformation. Use the CUSTOMERS table from the source database to serve as the Lookup table and import it from the database.

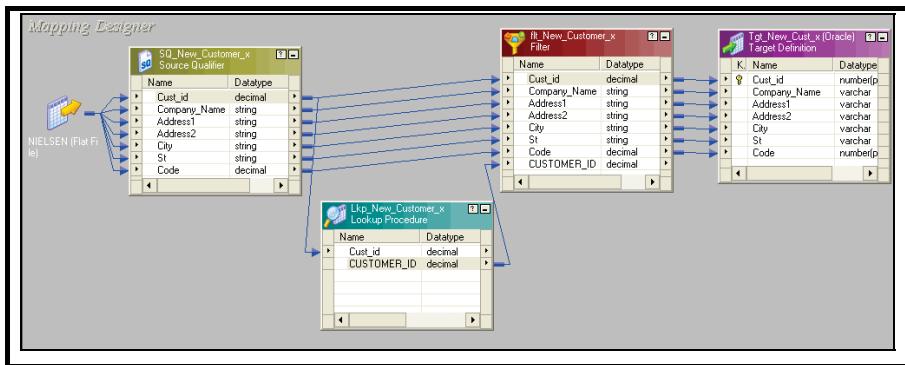


8. Create an input-only port in Lkp\_New\_Customer\_x to hold the Customer\_Id value, coming from SQ\_NIELSEN\_x .
9. Drag/Drop Cust\_Id column from the SQ\_NIELSEN\_x to Lkp\_New\_Customer\_x .
  - i. Make Cust\_Id an input-only port and Make CUSTOMER\_Id a lookup and output port.
  - ii. Add the lookup condition: CUSTOMER\_ID = Cust\_Id.

10. Click the Properties tab and note Connection Information .



11. Create a Filter transformation that will filter through those records that do not match the lookup condition and name it Flt\_New\_Cust\_x.
12. Drag / drop all the ports from Source Qualifier to the new Filter and CUSTOMER\_ID port from Lkp\_New\_Customer\_x .
13. Enter the filter condition: ISNULL(CUSTOMER\_ID). This condition will allow only those records whose value for CUSTOMER\_ID is = null, to pass through the filter.
14. Link all ports except CUSTOMER\_ID from the Filter to the Target table.
15. Given below is the final mapping.



16. Create a Workflow by name wf\_New\_Customer\_x and a Session Task by name s\_New\_Customer\_x.
17. Select the **Mapping** tab.
18. Select the **Source** folder and verify the attribute settings are set to the following:  
Source Directory path = \$PMSourceFileDir  
File Name = Nielsen.dat (Use the same case as that present on the server)  
Source Type: Direct  
**Note :** For the session you are creating, the Server needs the exact path, file name and extension for the file as it resides on the Server, to use at run time
19. Under Set File Properties button, Click on Advanced and Check the Line sequential file format check box.
20. Set appropriate Database Connection for the Lkp\_New\_Customer transformation.
21. Run and monitor the Workflow.
22. Verify the results.

```
SQL> select * from tgt_new_cust_x;
```

CUST_ID	COMPANY_NAME	ADDRESS1
93677	The Cracker Box	55 Grizzly Peak Rd.
103677	Suprêmes délices	Boulevard Tirou, 255
113671	Magazzini Alimentari Riuniti	Via Ludovico il Moro 22
113677	The Big Cheese	89 Jefferson Way
121077	Wellington Importadora	Rua do Mercado, 12
121177	White Clover Markets	305 - 14th Ave. S.
121277	Wilman Kala	Keskuskatu 45
122377	Tradição Hipermercados	Av. Inês de Castro, 414
123077	Great Lakes Food Market	2732 Baker Blvd.
123177	Folk och fä HB	Åkersgatan 24
123277	Folies gourmandes	184, chaussée de Tournai
123377	Godos Cocina Típica	C/ Romero, 33
123456	Galería del gastrónomo	Rambla de Cataluña, 23
123477	Wolski Zajazd	ul. Filtrowa 68
123479	Furia Bacalhau e Frutos do Mar	Jardim das rosas n. 32
123574	Trail's Head Gourmet Provisioners	722 DaVinci Blvd.
123577	Chop-suey Chinese	Hauptstr. 29
123579	GROSELLA-Restaurante	5 <sup>a</sup> Ave. Los Palos Grandes
123612	Vaffeljernet	Smagsløget 45
123613	Laughing Bacchus Wine Cellars	1988 Oak St.
123614	Lazy K Kountry Store	12 Orchestra Terrace
.....		
.....		
.....		
723672	Split Rail Beer & Ale	P.O. Box 555
723677	Hungry Owl All-Night Grocers	8 Johnstown Road
823672	Ernst Handel	Kirchgasse 6

84 rows selected.

## Lab 9.1. Lab 6-1 Flat File Join

<b>Goals</b>	<ul style="list-style-type: none"><li>• Analyze delimited files</li><li>• Join heterogeneous sources</li></ul> <p>Use Aggregator and Rank transformations to provide top ten revenue producing items</p>
<b>Time</b>	60 minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Client

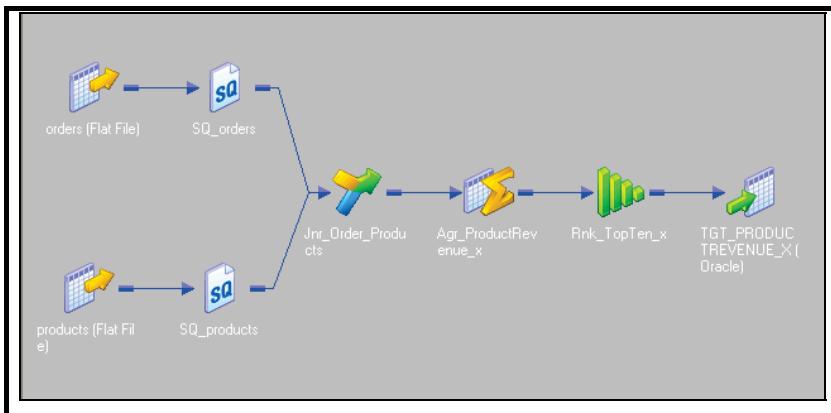
### Background

The Sales Department wants to use data contained in flat files to build a table summarizing the revenue on each product by Product Code, Product Name, and Product Category. The weekly orders from each store are consolidated into one orders file, and the IT organization has downloaded from the Mainframe a flat file listing of each product sold by the company.

### Solution

- Build a new target table that will contain only top ten selling items ranked by sales revenue
- Use PowerCenter tools to import and analyze the source files and create a target database table
- Use an Aggregator transformation to group
- Use a Rank transformation object to identify only top ten items to be sent to the target tables

### Mapping Layout



TRANSFORMATION	TYPE	DESCRIPTION
ORDERS, PRODUCTS	Sources	Flat file source definitions
SQ_ORDERS, SQ_PRODUCTS	Source Qualifier	Data source qualifiers for flat file sources
JNR_ORDERS_PRODUCTS_X	Joiner	Join the heterogeneous sources on the ITEM_NO field. The products file will be your master source.
AGG_PRODUCT_REVENUE_X	Aggregator	Calculate total price and quantity for items grouped by ITEM_NO, ITEM_NAME, and PRODUCT_CATEGORY
RNK_TOPTEN_X	Rank	Rank the top ten revenue-producing items
TGT_PRODUCTREVENUE_X	Target	Target definition (Relational)

### Final Output

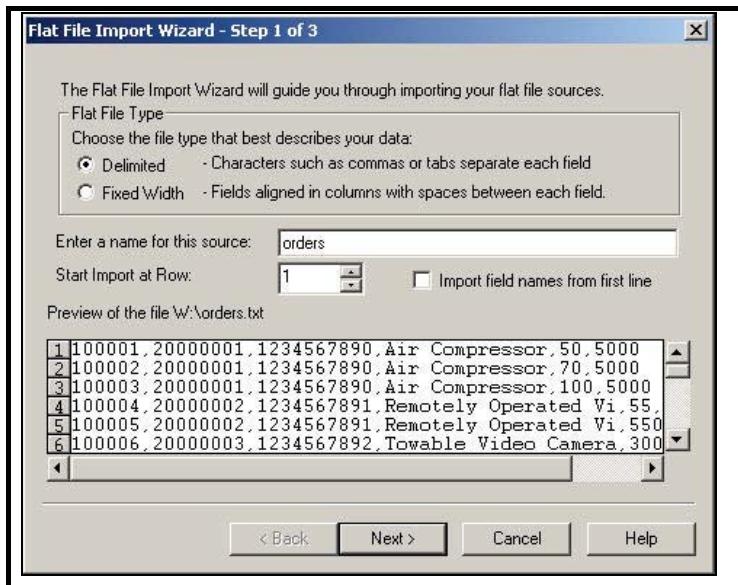
```
SQL> select * from tgt_ProductRevenue_x;
```

ITEM_NO	ITEM_NAME	PRODUCT_CATEGORY	PRICE	TOTAL_QUANTITY	TOTAL_REVENUE
12347	Towable Video Camera	Photo Equipment	7800	43000000	3.0101E+11
12349	Dive computer	Small Instruments	8880	19050	169164000
12343	Under water Metal Det	Misc Equipment	7500	13900	104250000
12348	Marine super vhs vi	Photo Equipment	8800	8900	71200000
12344	Stabilizing vest	Misc Equipment	6800	10900	65400000
12340	Under Water Dive vehicel	Vehicle	7500	3600	27000000
12346	Remotely Operated Vi	Photo Equipment	6000	605	3630000
12345	Air Compressor	Misc Equipment	5800	220	1100000

8 rows selected.

### Problem Solution

1. Import the ORDERS.TXT and PRODUCTS.TXT flat files definition into the repository.
2. Set the following options in the Flat File Wizard:
  - i. Select the Delimited radio button.



- ii. Name the source definition ORDERS,PRODUCTS.
- iii. Enter the column names and specify data types and field widths as shown below.

*Source Analyzer*

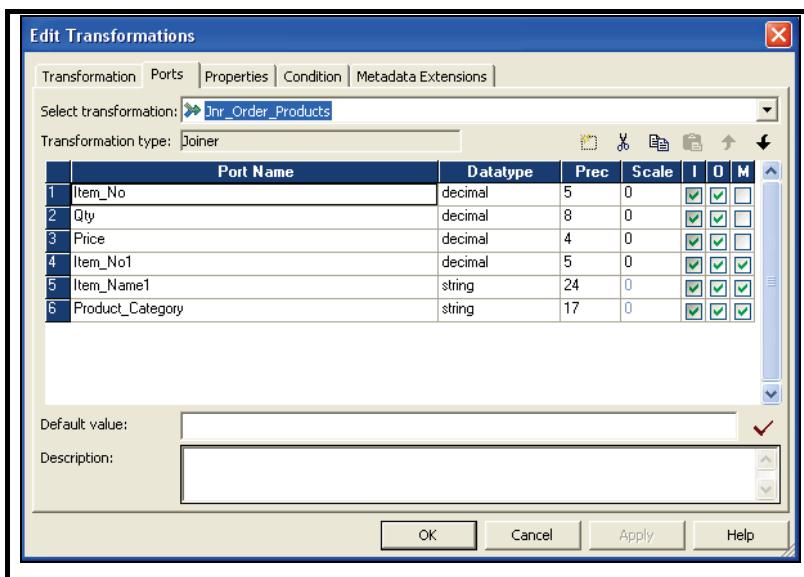
orders (Flat File)		
K.	Name	Datatype
	Order_No	number
	Line_No	number
	Item_No	number
	Item_Name	string
	Qty	number
	Price	number

products (Flat File)		
K.	Name	Datatype
	Item_No	number
	Item_Name	string
	Cat	string
	Cust_Price	number
	Vendor_Price	number
	Product_Category	string

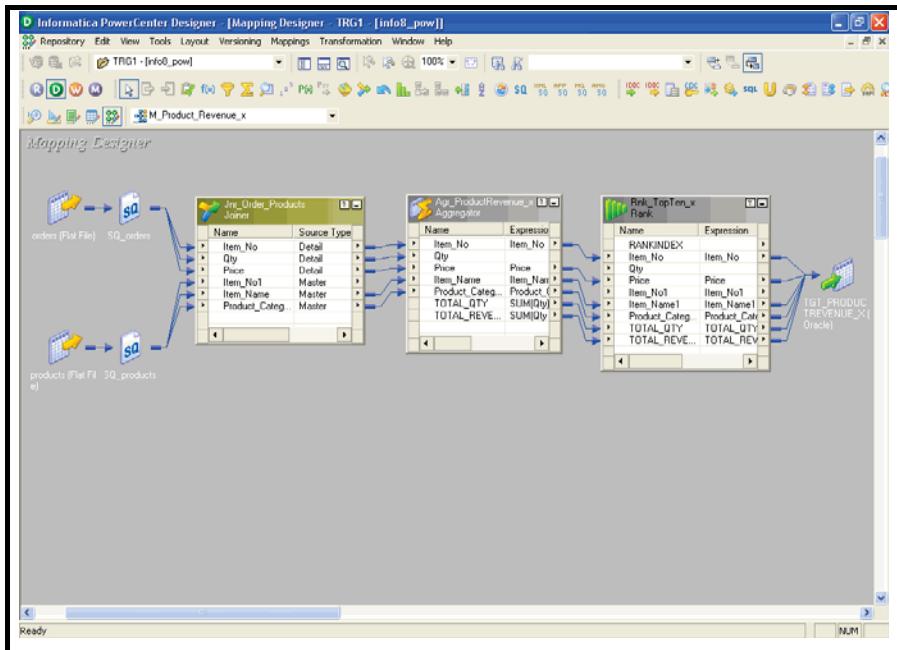
3. Create the target table with name Tgt\_ProductRevenue\_x .It should have following columns:  
ITEM\_NO,ITEM\_NAME,PRODUCT\_CATEGORY,PRICE,TOTAL\_QUANTITY,TOT AL\_REVENUE.
4. Create mapping with the name M\_Product\_Revenue\_x.
5. DRAG the ORDERS and PRODUCTS source definitions.
6. Drag the Tgt\_ProductRevenue\_x target definition
7. Create the Joiner transformation Jnr\_Orders\_Products\_x.
8. Link the ITEM\_NO, ITEM\_NAME, and PRODUCT\_CATEGORY ports from SQ\_Products and ITEM\_NO, QTY, and PRICE ports: from SQ\_ORDERS (Source Qualifier) into Jnr\_Orders\_Products\_x (Joiner).
9. Identify all the ports from PRODUCTS as Master ports.

**HINT:** Check the M column checkbox for any one of the ports, which flow originally from the PRODUCTS source definition.



10. Add a new condition: ITEM\_NO = ITEM\_NO1.
11. Create an aggregator transformation with name Agg\_ProductRevenue\_x link it to JNR\_ORDERS\_PRODUCTS.
12. Add TOTAL\_QTY and TOTAL\_REVENUE ports from Tgt\_ProductRevenue\_x (target definition) into the Agg\_ProductRevenue\_x(Aggregator).
13. Group by ITEM\_NO, ITEM\_NAME, and PRODUCT\_CATEGORY.
14. Enter aggregate expressions for the TOTAL\_QUANTITY and TOTAL\_REVENUE ports:  
**TOTAL\_QUANTITY: SUM(QTY)**  
**TOTAL\_REVENUE : SUM(PRICE \* QTY)**
15. Create Rank Transformation with name Rnk\_TopTen\_x.
16. Link Agg\_ProductRevenue to Rnk\_TopTen .
17. Identify the TOTAL\_REVENUE port as the one to rank.
18. Deselect the GroupBy options on the ITEM\_NAME and PRODUCT\_CATEGORY ports.
19. Select Top/Bottom = Top, and Number of Ranks = 10.  
Connect the Rank transformation to the target table.

20. Given below is the final mapping.



21. Create a Workflow by name wf\_ProductRevenue\_x and a session task by the name s\_ProductRevenue\_x
22. Run and monitor the Workflow.
23. Verify the results for target table Tgt\_ProductRevenue\_x.

## Lab 10.1. Lab 7-1 Create a Maplet

<b>Goals</b>	<ul style="list-style-type: none"><li>• Learn to create a maplet</li><li>• Understand how to use variables</li></ul>
<b>Time</b>	60 minutes
<b>Lab Setup</b>	PowerCenter Client connectivity

### Background

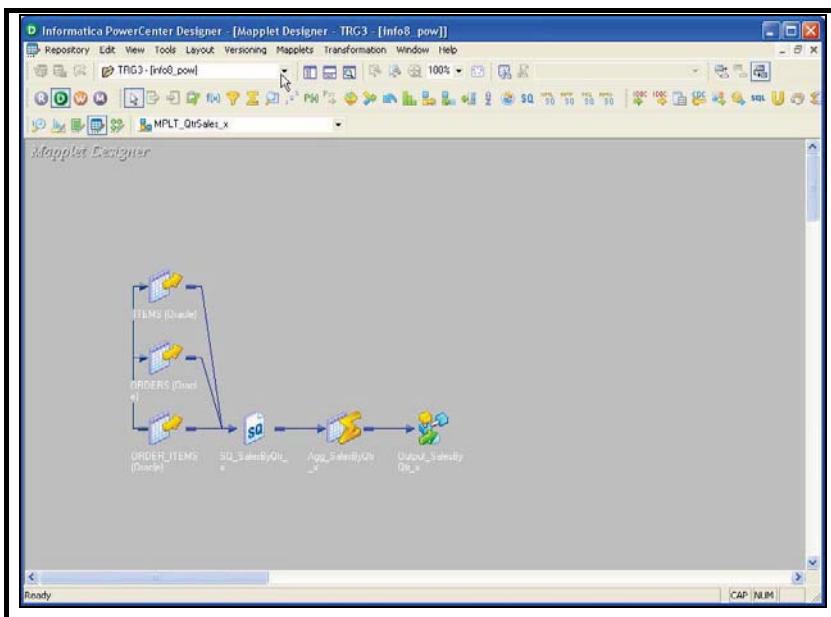
The Sales Department is interested in getting both the quarterly and yearly sales. The Data Mart server has a high CPU usage, so you would like to pre-calculate these amounts and put them in summary tables.

### Solution

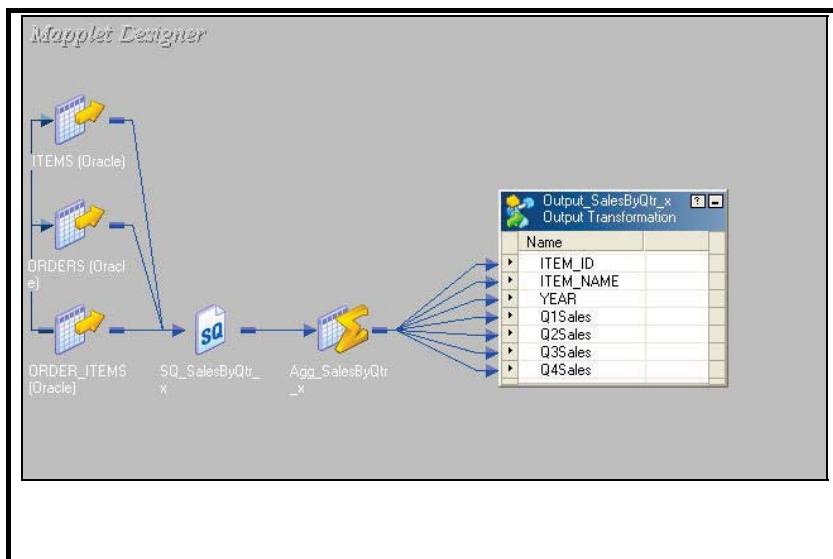
- Build a maplet that uses multiple sources and aggregate functions
- Create a variable within the maplet for use in the aggregate functions
- In the next lab, use this maplet to give the quarterly sales

TRANSFORMATIONS	TYPE	DESCRIPTION
ITEMS ORDER_ITEMS ORDERS	Source	Relational source definitions
SQ_SALESBYQTR_X	Source Qualifier	Data source qualifier to join the source tables and sort the data
AGG_SALESBYQTR_X	Aggregator	Extract the month from the Date_Entered into a variable port. Call to that variable port when aggregating the quarterly sales
OUTPUT_SALESBYQTR_X	Target	Output transformation for the maplet

## Maplet Layout



### Maplet output



### Problem Solution

1. Create a Maplet with the name MPLT\_QtrSales\_x. Import the tables ITEMS, ORDER\_ITEMS, and ORDERS.
2. Manually create a Source Qualifier SQ\_SalesByQtr\_x to pull in data from the above three source definitions.
3. Create an Aggregator transformation and name it Agg\_SalesByQtr\_x. Copy and link ITEM\_ID and ITEM\_NAME from the Source Qualifier (SQ\_SalesByQtr\_x) into the aggregator (Agg\_SalesByQtr\_x). On the Columns tab, add the following ports:

YEAR	string(4)	Output
MONTH	integer	Variable
Q1Sales	decimal(19,2)	Output
Q2Sales	decimal(19,2)	Output
Q3Sales	decimal(19,2)	Output

Q4Sales	decimal(19,2)	Output
---------	---------------	--------

4. Build the expressions for the Variable and Output ports as follows:

PORT	EXPRESSION BEFORE VALIDATION	EXPRESSION AFTER VALIDATION
YEAR	TO_CHAR(SQ_SalesByQtr_x.DATE_ENTERED, 'YYYY')	TO_CHAR(DATE_ENTERED, 'YYYY')
MONTH	GET_DATE_PART(DATE_ENTERED, 'MM')  NOTE: DATE_ENTERED is from Agg_SalesByQtr_x not from SQ_SalesByQtr_x >	<no change>
Q1Sales	SUM(SQ_SalesByQtr_x.QUANTITY * SQ_SalesByQtr_x.PRICE – SQ_SalesByQtr_x.DISCOUNT, MONTH = 1 OR MONTH = 2 OR MONTH = 3)	SUM(QUANTITY * PRICE – DISCOUNT, MONTH = 1 OR MONTH = 2 OR MONTH = 3)
Q2Sales	SUM(QUANTITY * PRICE – DISCOUNT, MONTH = 4 OR MONTH = 5 OR MONTH = 6)	<no change>
Q3Sales	SUM(QUANTITY * PRICE – DISCOUNT, MONTH = 7 OR MONTH = 8 OR MONTH = 9)	<no change>
Q4Sales	SUM(QUANTITY * PRICE – DISCOUNT, MONTH = 10 OR MONTH = 11 OR MONTH = 12)	<no change>

5. Aggregate records by ITEM\_ID, ITEM\_NAME and YEAR.
6. Select the Properties tab. Check the Sorted Input box. Exit the Edit Transformation dialog box.
7. Edit the Source Qualifier. The ports in the Source Qualifier must be in the same order as the ports in the Aggregator, in order to facilitate the correct summarization by the groupings you have specified, above.
8. Open the SQL query window. Click the Generate SQL button. Append the following text to the end of the default SQL statement:
9. ORDER BY, ITEMS.ITEM\_ID, ITEMS.ITEM\_NAME,  
ORDERS.DATE\_ENTERED.
10. Verify that there are no errors in the SQL. Exit the SQL editor and the Source Qualifier Transformation.
11. Create a mapplet output transformation by the name Output\_SalesByQtr\_x. Connect all the output ports of the Aggregator to the mapplet Output transformation.
12. Verify the results of the mapplet validation in the Output Window.

## Lab 11.1. Lab 7-2 Quarterly-Sales Mapping

<b>Goals</b>	<ul style="list-style-type: none"><li>• Use a mapplet in a mapping</li><li>• Configure a Normalizer transformation</li></ul>
<b>Time</b>	60 minutes
<b>Lab Setup</b>	Quarterly sales mapplet created in the Lab 7-1

### Background

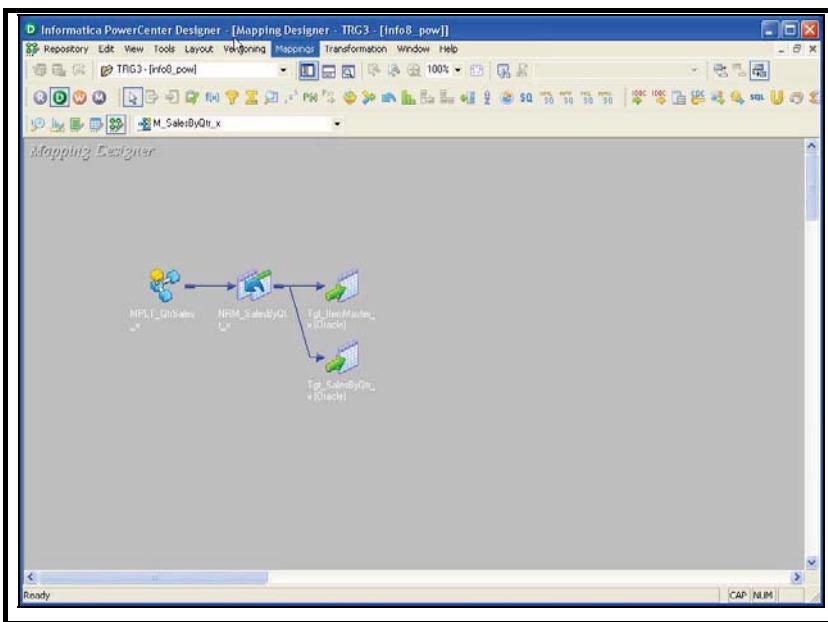
The Sales Department wants to run queries on sales by quarter. The Data Mart server has a high CPU usage, so you would like to pre-calculate these amounts and put them in summary tables.

### Solution

- Use the quarterly sales mapplet as source for the mapping
- Use a Normalizer transformation to split the data into master and detail targets
  - The master target will contain descriptive information about each item
  - The detail target would contain a separate record for each quarterly sales figure and will have a foreign key to the master table

TRANSFORMATION	TYPE	DESCRIPTION
MPLT_QTRSALES_X	Mapplet	Partially transformed data(Source)
NRM_SALESBYQTR_X	Normalizer	Split the data into a master/detail relationship, in which the detail table contains a separate record for each quarterly sales figure.
TGT_ITEMMASTER_X	Target	Descriptive information about each item
TGT_SALESBYQTR_X	Target	Quarterly sales figures with a foreign key to ITEM_MASTER

### Mapping Layout



## Final Output

SQL> select * from tgt_itemmaster_x;			
ITEM_MASTER_ID	ITEM_ID	ITEM_NAME	YEAR
1	1813	Regulator System	1998
2	1814	Second Stage Regulator	1998
3	1880	Alternate Inflation Unit	1998
4	1890	First Stage Regulator	1998
5	1986	Depth/Pressure Gauge	1998
6	2314	Electronic Console	1998
7	2341	Depth/Pressure Gauge	1998
8	2343	Personal Dive Sonar	1998
9	2350	Compass Console Mount	1998
10	2367	Compass (meter only)	1998
11	2612	Direct Sighting Computer	1998
12	2613	Dive Computer	1998
13	2619	Navigation Compass	1998
14	2630	Wrist Band Thermometer	1998
15	3316	Stabilizing Vest	1998
16	3326	Front Clip Stabilizing Vest	1998
17	3340	Trim Fit Stabilizing Vest	1998
18	3386	Welded Seam Stabilizer	1998
19	5313	Safety Knife	1998
20	5318	Medium Titanium Knife	1998
21	5324	Chisel Point Knife	1998
ITEM_MASTER_ID	ITEM_ID	ITEM_NAME	YEAR
22	5349	Flashlight	1998
23	5356	Medium Stainless Steel Knife	1998
24	5378	Divers Knife and Sheath	1998
25	7612	Krypton Flashlight	1998
26	7619	Flashlight (Recharge)	1998
27	7654	Halogen Flashlight	1998
28	9312	60.6 cu ft Tank	1998
29	9316	95.1 cu ft Tank	1998
30	9318	71.4 cu ft Tank	1998
31	9354	75.8 cu ft Tank	1998

31 rows selected.

```
SQL> select * from tgt_salesbyqtr_x;
```

ITEM_MASTER_ID	QUARTER	QUARTERLY_SALES
1	1	489.5
1	2	1719.41
1	3	
1	4	729
2	1	719.5
2	2	1084.48
2	3	
2	4	2524
3	1	
3	2	1534.47
3	3	
3	4	249.5
4	1	669.5
4	2	159.69
4	3	
4	4	840
5	1	177.5
5	2	372.01
5	3	
5	4	543
6	1	759.5

ITEM_MASTER_ID	QUARTER	QUARTERLY_SALES
6	2	1549.5
6	3	
6	4	769.5
7	1	409.5
7	2	494
7	3	
7	4	
8	1	
8	2	224.5
8	3	
8	4	
9	1	76.5
9	2	82

9	3	
9	4	
10	1	
10	2	281
10	3	
10	4	83.5
11	1	
11	2	64.9

**ITEM\_MASTER\_ID QUARTER QUARTERLY\_SALES**

---

11	3	
11	4	129.3
12	1	
12	2	
12	3	
12	4	168.5
13	1	
13	2	29.4
13	3	
13	4	29.4
14	1	
14	2	
14	3	51
14	4	
15	1	
15	2	
15	3	849.5
15	4	1699.5
16	1	
16	2	835
16	3	539.5

**ITEM\_MASTER\_ID QUARTER QUARTERLY\_SALES**

---

16	4	
17	1	
17	2	779.5
17	3	
17	4	1174.5
18	1	
18	2	269.5
18	3	
18	4	1115

19	1	
19	2	71.5
19	3	
19	4	30.5
20	1	
20	2	
20	3	
20	4	310.7
21	1	30.5
21	2	
21	3	
21	4	112.5

**ITEM\_MASTER\_ID QUARTER QUARTERLY\_SALES**

22	1	54.5
22	2	
22	3	125
22	4	
23	1	205
23	2	
23	3	59.5
23	4	
24	1	49.5
24	2	
24	3	129.5
24	4	
25	1	79.4
25	2	
25	3	169.3
25	4	
26	1	988.7
26	2	
26	3	
26	4	
27	1	109.4

**ITEM\_MASTER\_ID QUARTER QUARTERLY\_SALES**

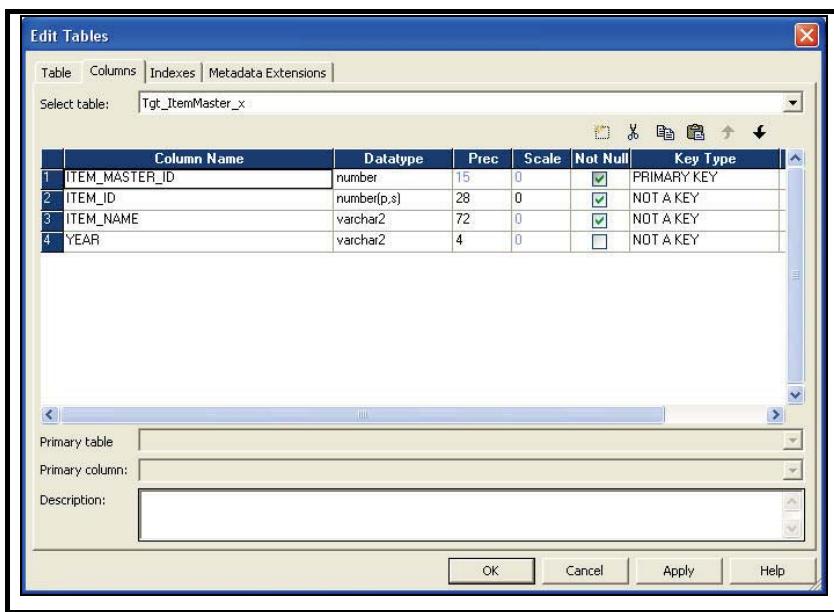
27	2	109.4
27	3	
27	4	
28	1	174

28	2	347.5
28	3	
28	4	
29	1	639.5
29	2	
29	3	
29	4	645
30	1	574.5
30	2	
30	3	
30	4	184.5
31	1	919.5
31	2	
31	3	
31	4	684.5

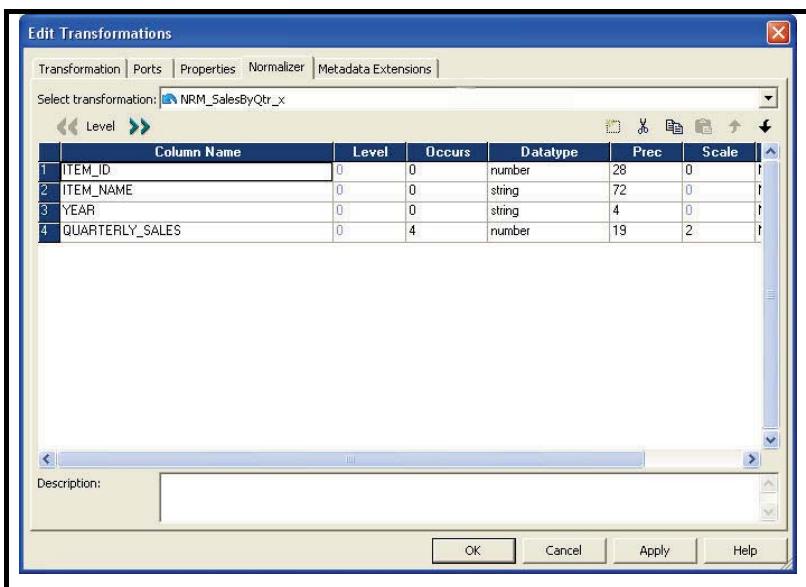
124 rows selected.

### Problem Solution

1. Create a new mapping by the name M\_SalesByQtr\_x.
2. Drag the MPLT\_QtrSales\_x Maplet from the Navigator Window into your workspace.
3. Create the target table definition Tgt\_ItemMaster\_x table. The structure should be as shown below:

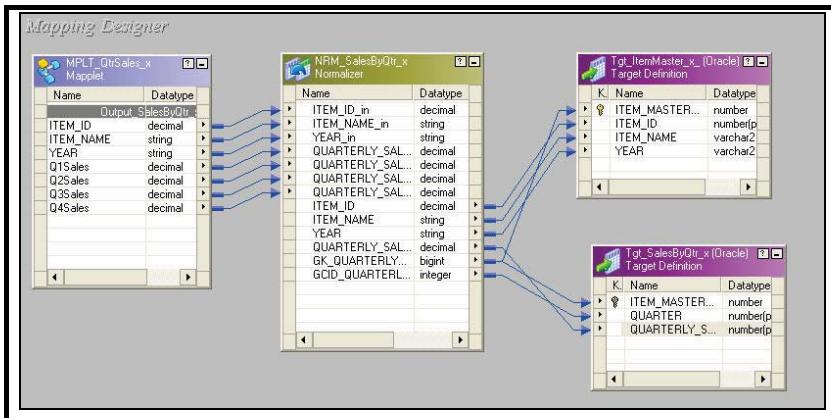


4. Create the target table definition for the Tgt\_SalesByQtr\_x table.
5. Make ITEM\_MASTER\_ID a FOREIGN\_KEY.
6. Choose Tgt\_Item\_Master\_x as the primary table.
7. Choose ITEM\_MASTER\_ID as the primary column.
8. Create a Normalizer Transformation with the name NRM\_SalesByQtr\_x and create ports with the data types as shown below:



9. Validate the Mapping and save it to the repository.

10. Given below is the final mapping.



11. Create a workflow by name wf\_SalesByQtr\_x and a session task by name s\_SalesByQtr\_x.

12. Select the Source and Target Database Connection.

13. Run and monitor the Workflow.

14. Verify the results.

## Lab 12.1. Lab 7-3 Annual Sales Mapping

<b>Goals</b>	<ul style="list-style-type: none"><li>• Use a mapplet</li><li>• Create a Rank Transformation</li></ul>
<b>Time</b>	60 minutes
<b>Lab Setup</b>	Quarterly sales mapplet created in the Lab 7-1

### Background

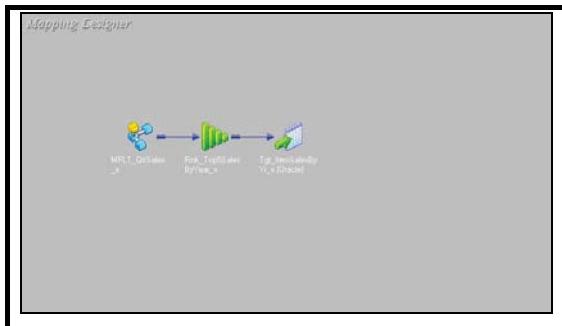
The Sales Department wants to run queries on sales by quarter. The Data Mart server has a high CPU usage, so you would like to pre-calculate these amounts and put them in summary tables.

### Solution

- Use the quarterly sales mapplet as source for the mapping
- Use a Rank transformation that will rank the data coming out of the mapplet and compute an annual sales figure for each item. The end result should be a table containing the top five selling items for each year

TRANSFORMATION	TYPE	DESCRIPTION
MPLT_QTRSALES_X	Mapplet	Partially transformed data (Source)
RNK_TOP5SALESBYYEAR_X	Rank	Add the quarterly sales figure to compute annual sales figures. Rank the top 5 selling items for each year
TGT_SALESBYYR_X	Target	Top 5 selling items for each year

## Mapping Layout



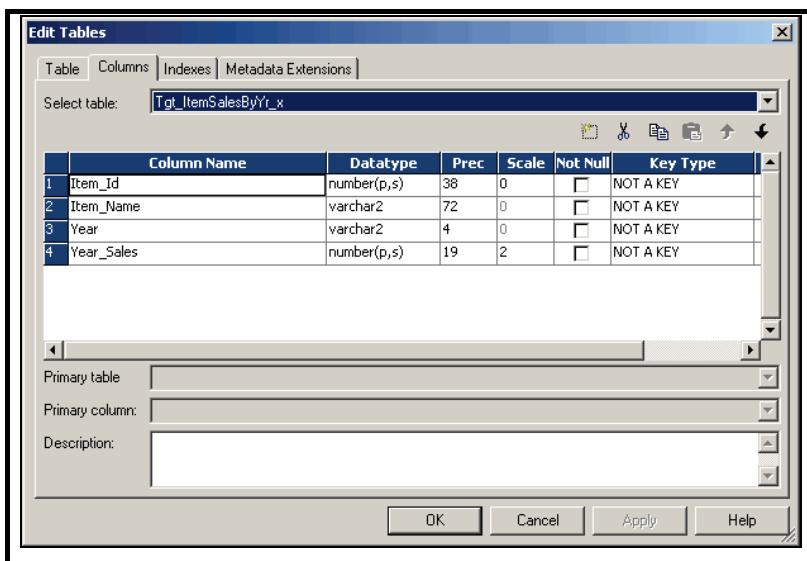
## Final Output

```
SQL> select * from tgt_itemsalesbyyr_x;
```

ITEM_ID	ITEM_NAME	YEAR	YEAR_SALES
1314	Second Stage Regulator	1998	4327.98
2314	Electronic Console	1998	3878.5
1313	Regulator System	1998	2937.91
3316	Stabilizing Vest	1998	2549
3340	Trim Fit Stabilizing Vest	1998	1954

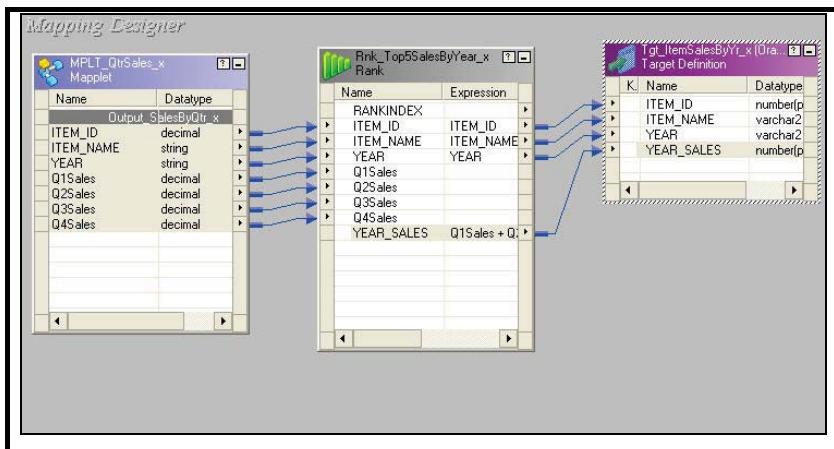
## Problem Solution

1. Create a new mapping by the name M\_Top5SalesByYear\_x.
2. Drag the MPLT\_QtrSales\_x Maplet from the Navigator Window into your workspace.
3. Create the target table definition Tgt\_ItemsalesByYr\_x and create the physical table in the database. The structure should be as shown below:



4. Create a Rank Transformation and name it Rnk\_Top5SalesByYear\_x.
5. Connect the ports from MPLT\_QtrSales\_x (maplet) to the new Rank transformation.
6. Click on the Ports tab of the Rnk\_Top5SalesByYear\_x Rank transformation. Make Q1Sales, Q2Sales, Q3Sales and Q4Sales input-only ports and give a default value Zero to each port.
7. Group by YEAR.
8. Add a new output port for rank column and name it as Year\_Sales with the expression:  
Q1Sales + Q2Sales + Q3Sales + Q4Sales.
9. Click on the Properties tab.
  - i. Set the Top/Bottom attribute value to Top.
  - ii. Set the Number of Ranks attribute to 5.
10. Connect the ports in Rnk\_Top5SalesByYear\_x (Rank transformation) to the Tgt\_SalesByYear\_x (target definition).
11. Validate the Mapping and save it to the repository.

12. Given below is the final mapping:



13. Create a workflow by name wf\_Top5SalesByYear\_x.

14. Run and monitor the Workflow.

15. Verify the result.

## Lab 13.1. Lab 8-1 Updating Current Items

<b>Goals</b>	<ul style="list-style-type: none"> <li>Configure an Unconnected Lookup Transformation</li> <li>Configure an Update Strategy transformation</li> <li>Understand Tracing levels in session logs</li> </ul>
<b>Time</b>	120 Minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Designer and Workflow Manager

### Background

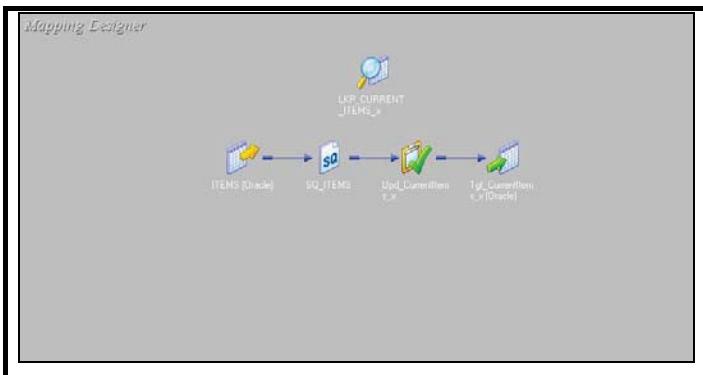
The operational source system that supplies data to your data mart tracks all items that your company has ever sold, even if they have since been discontinued. Your Sales Department wants to run queries against a Data Mart table that contains only currently selling items. They don't want to use views or SQL, and they want this table updated on a regular basis.

### Solution

- Use the operational source table ITEMS to build a new Data Mart table, CURRENT\_ITEMS, which will contain only current selling items
- Create an Unconnected Lookup transformation object to match source items against current items in the Data Mart.
- Create an Update Strategy transformation to test the result of the lookup and determine the appropriate row action to take on the first and subsequent runs of the session.
- New current items will be inserted, discontinued items will be rejected, current items already in the target will be updated and current items already in the target but discontinued since the last session run will be deleted.

TRANSFORMATIN	TYPE	DESCRIPTION
ITEMS	Source	Operational source table (Relational)
SQ_ITEMS	Source Qualifier	Data source qualifier – no special overrides necessary
LKP_NEW_CUSTOMER_X	Lookup	Match ITEMS.ITEM_ID against Tgt_CurrentItems_x.ITEM_ID
UPD_CURRENTITEMS_X	Update Strategy	Use the Update Strategy expression to populate the data in the target table
TGT_CURRENTITEMS_X	Target	Target definition (Relational)

### Mapping Layout



### Final Output

ITEM_ID	ITEM_NAME	ITEM_DESC	PRICE	WHOLESALE_COST	DISCONTINUED_FLAG	MANUFACTURER_ID	DISTRIBUTOR_ID
1313	Regulator System	Air Regulators	250	150	0	100	2012
1314	Second Stage Regulator	Air Regulators	365	265	0	100	2012
1330	Alternate Inflation Unit	Air Regulators	260	160	0	100	2001
1398	First Stage Regulator	Air Regulators	170	70	0	101	2000
1986	Depth/Pressure Gauge	Small Instruments	188	88	0	101	2002
2314	Electronic Console	Small Instruments	399	299	0	102	2009
2341	Depth/Pressure Gauge	Small Instruments	105	5	0	102	2004
2343	Personal Dive Sonar	Small Instruments	235	135	0	102	2006
2350	Compass Console Mount	Small Instruments	29	17	0	103	2004
2367	Compass (meter only)	Small Instruments	52	48	0	104	2008
2612	Direct Sighting Computer	Small Instruments	34.95	15.05	0	105	2008
2613	Dive Computer	Small Instruments	179	79	0	106	2009
2619	Navigation Compass	Small Instruments	19.95	8.05	0	107	2006
2630	Wrist Band Thermometer	Small Instruments	18	8	0	108	2007
3316	Stabilizing Vest	Buoyancy Compensation	430	330	0	109	2009
3326	Front Clip Stabilizing Vest	Buoyancy Compensation	280	180	0	110	2003
3386	Welded Seam Stabilizer	Buoyancy Compensation	280	180	0	108	2011
5313	Safety Knife	Tools	41	29	0	107	2003
5324	Chisel Point Knife	Tools	41	19	0	105	2006
5349	Flashlight	Tools	65	35	0	104	2007
5356	Medium Stainless Steel Knife	Tools	70	30	0	103	2007
5378	Divers Knife and Sheath	Tools	70	30	0	102	2001
7612	Krypton Flashlight	Tools	44.95	25.05	0	101	2004
7619	Flashlight (Recharge)	Tools	169.95	69.95	0	100	2012
7654	Halogen Flashlight	Tools	59.95	40.05	0	103	2011
9312	60.6 cu ft Tank	Air Tank	179	79	0	104	2018
9318	71.4 cu ft Tank	Air Tank	195	95	0	106	2009
9354	75.8 cu ft Tank	Air Tank	235	135	0	107	2007

28 rows selected.

**Problem Solution**

1. Import source as ITEMS table.
2. Create a target definition having same ports as in source table and name it as Tgt\_CurrentItems\_x.
3. Create a mapping called M\_CurrentItems\_x.
4. Create an Unconnected Lookup transformation to match ITEMS.ITEM\_ID against Tgt\_CurrentItems\_x.ITEM\_ID and name it as LKP\_CURRENT\_ITEMS\_x.
5. Add a new input port, ITEM\_ID\_IN, with the same data type as ITEM\_ID.
6. Make ITEM\_ID the R port.
7. Click the Properties tab.
8. Verify that the database connection is set to the correct target database string provided by your Instructor.
9. Create an Update Strategy transformation.
10. The pseudocode for the logic is as follows:

```
if (the record doesn't exist in the target table) then
    if (the discontinued flag is not set) then
        INSERT
    else
        REJECT
    else if (record exists)
        if (the discontinued flag is not set) then
            UPDATE the record
        else
            DELETE the record
```
11. Create an expression for the above pseudo code and enter it in the Update Strategy expression editor. The expression will call the Unconnected Lookup transformation.
12. Complete the mapping.
13. Create a Workflow by name wf\_CurrentItems\_x and Session Task by name s\_CurrentItems\_x.
14. Run and monitor the Workflow.
15. Verify the results.

## Lab 14.1. Lab 9-1 Listing Order Details

<b>Goals</b>	<ul style="list-style-type: none"><li>• Create a mapping that uses a Sorter Transformation</li></ul>
<b>Time</b>	30 minutes
<b>Lab Setup</b>	A connection to the repository using PowerCenter Designer and Workflow Manager

### Background

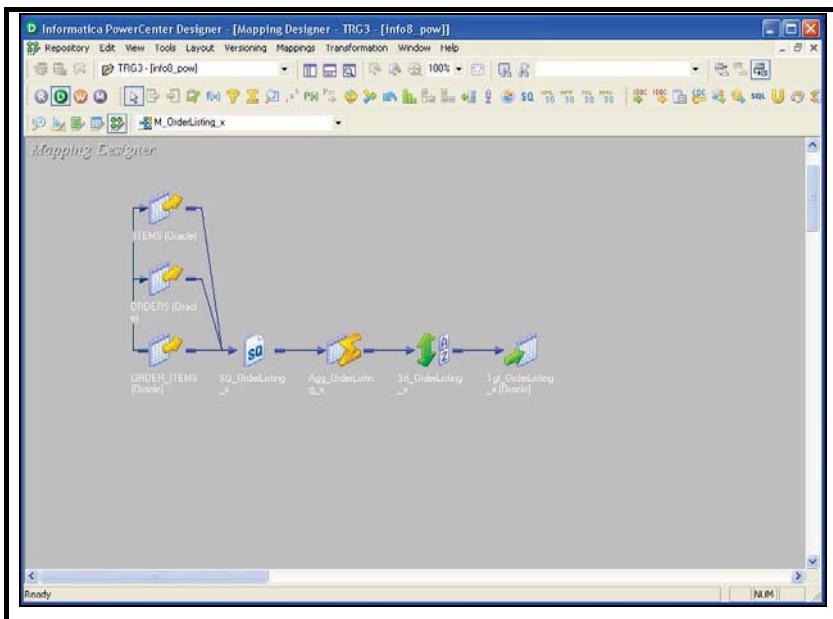
Company requires a report, which will show all order details in descending order of order amount.

### Solution

- Import Order, Items, Order\_Items tables from the database
- Calculate the total Order Amount for each Order
- Create a target, which will show the total order amount in descending order.

TRANSFORMATION NAME	TYPE	DESCRIPTION
ITEMS ORDER_ITEMS ORDERS	Relational Source Definition	Source definitions
SQ_ORDERLISTING_X	Source Qualifier	Data source qualifier for all source tables
AGG_ORDERLISTING_X	Aggregator	Link ports ORDER_ID, DATE_ENTERED, CUSTOMER_ID, QUANTITY, PRICE, DISCOUNT into the Aggregator. Create an output port to hold the ORDER_AMOUNT
SRT_ORDERLISTING_X	Sorter	Sorts Order Amount in descending order
TGT_ORDERDETAILS_X	Relational Target Table	Target definition

### Mapping Layout



## Final Output

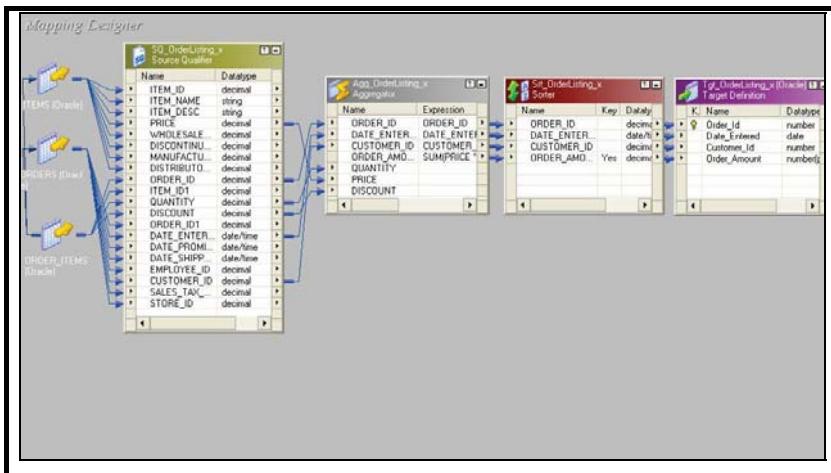
```
SQL> select * from tgt_orderlisting_x;

ORDER_ID DATE_ENTERED CUSTOMER_ID ORDER_AMOUNT
----- ---------
65 17-NOV-98    156003      3734
66 02-DEC-98    156003      3436
45 01-JUN-98    122101      2811
44 01-APR-98    122101      2599
61 15-OCT-98    156304      2423
58 01-JAN-98    156304      2093
51 01-APR-98    135608      1956
50 11-NOV-98    135608      1700
55 18-FEB-98    135106      1494
60 02-AUG-98    156003      1415
54 02-MAR-98    138401      1242
56 01-JAN-98    651605      1289
67 03-APR-98    156304      873
63 01-MAR-98    651605      798
53 15-AUG-98    123102      483
64 11-MAY-98    156304      457
46 09-JUN-98    123102      424
48 16-OCT-98    123102      411
52 31-DEC-98    138401      320
59 02-APR-98    156003      292
62 01-JAN-98    651605      290
47 04-MAY-98    123102      82
49 07-SEP-98    135106      26

23 rows selected.
```

### Problem Solution

1. Import all the sources from the database (Orders, Items, Order\_Items).
2. Create target table with ports:  
Order\_Id(Primary key, Not Null), Date\_Entered(Not Null), Customer\_Id(Not Null) and Order\_Amount(Not Null).
3. Create a Source Qualifier transformation and name it SQ\_OrderListing\_x.
4. Create an Aggregator transformation and group on the Order\_id column. Link ports ORDER\_ID, DATE\_ENTERED, CUSTOMER\_ID, QUANTITY, PRICE, DISCOUNT into the Aggregator. Make QUANTITY, PRICE, DISCOUNT only input ports.
5. Add a new output port Order\_Amount with the expression:  
 $SUM(PRICE * QTY - DISCOUNT)$ .
6. Create the Sorter Transformation and name it Srt\_OrderListing\_x.
7. Check the Key column of the Order\_Amount port and select Descending.
8. Connect the Sorter Transformation to target table.
9. Your mapping should look like the one as given below:



10. Create a workflow by name wf\_OrderListing\_x and a session task by name s\_OrderListing\_x for.
11. Run and monitor the Workflow.
12. Verify the results.

## Lab 15.1. Lab 10-1 Router Transformation

<b>Goals</b>	<ul style="list-style-type: none"> <li>• Create a mapping that uses a Router Transformation</li> </ul>
<b>Time</b>	60 mins
<b>Lab Setup</b>	A connection to the repository using PowerCenter Designer and Workflow Manager

### Background

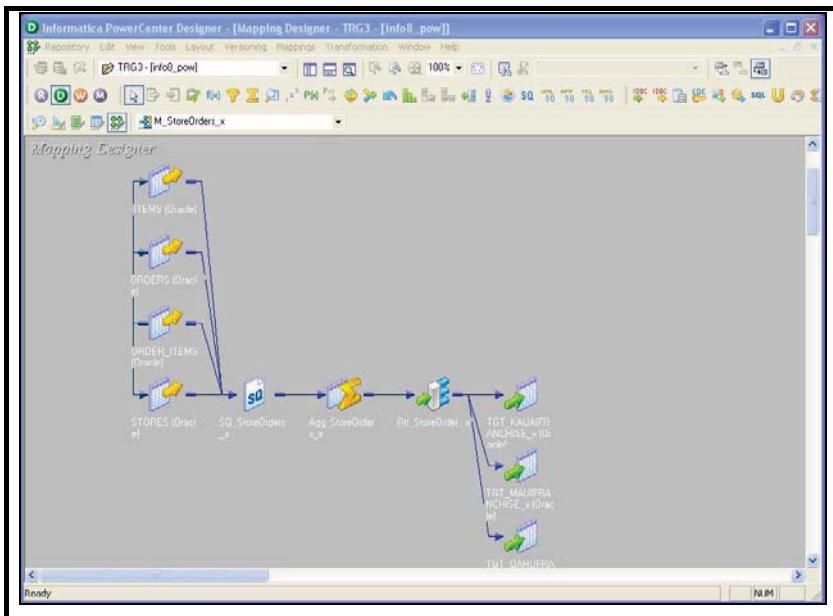
The company requires Store wise order details.

### Solution

- Calculate order amount for each order for each store
- Route the output based on store\_id and load the data in different tables created for each store
- Retrieve store wise order details

TRANSFORMATION NAME	TYPE	DESCRIPTION
ITEMS ORDER_ITEMS ORDERS STORES	Relational Source Definition	Source definitions
SQ_STOREORDERS_X	Source Qualifier	Data source qualifier for all source tables
AGG_STOREORDERS_X	Aggregator	Link ports STORE_ID, ORDER_ID, DATE_ENTERED, CUSTOMER_ID, QUALNTITY, PRICE, DISCOUNT, STORE_DESC, ORDER into the Aggregator. Create an output port to hold the ORDER_AMOUNT
RTR_STOREORDER_X	Router	Routes order details to different targets based on the Store_id, which is the group filter conditions.
TGT_KAUAIFRANCHISE_X TGT_MAUIFRANCHISE_X TGT_OAHUFRANCHISE_X	Relational Target Tables	Three target tables that will contain the order amount details for 3 different stores

## Mapping Layout



## Final Output

```
SQL> select * from tgt_kauaifranchise_x;

ORDER_ID DATE_ENTE CUSTOMER_ID ORDER_AMOUNT
-----
44 01-APR-98    122101      2599
46 09-JUN-98    123102      424
48 16-OCT-98    123102      411
50 11-NOV-98    135608      1700
52 31-DEC-98    138401      320
54 02-MAR-98    138401      1242
45 01-JUN-98    122101      2811
47 04-MAY-98    123102      82
49 07-SEP-98    135106      26
51 01-APR-98    135608      1956
53 15-AUG-98    123102      483
55 18-FEB-98    135106      1494

12 rows selected.
```

```
SQL> select * from tgt_mauifranchise_x;

ORDER_ID DATE_ENTE CUSTOMER_ID ORDER_AMOUNT
-----
56 01-JAN-98    651605      1209
58 01-JAN-98    156304      2093
60 02-AUG-98    156003      1415
62 01-JAN-98    651605      298
64 11-MAY-98    156304      457
65 17-NOV-98    156003      3734
59 02-APR-98    156003      292
61 15-OCT-98    156304      2423
63 01-MAR-98    651605      798
66 02-DEC-98    156003      3436
67 03-APR-98    156304      873

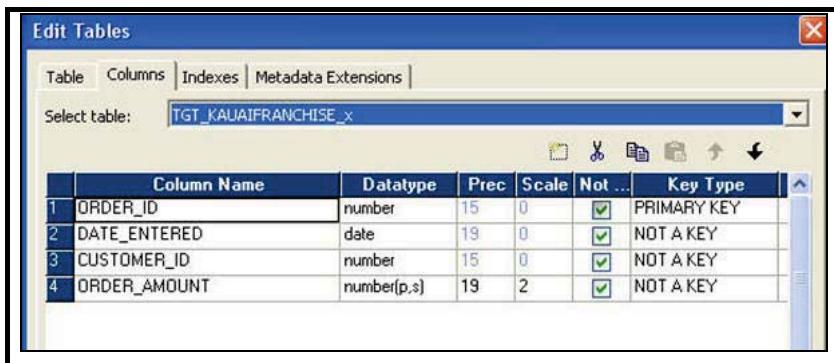
11 rows selected.
```

```
SQL> select * from tgt_ohaufranchise_x;

no rows selected
```

### Problem Solution

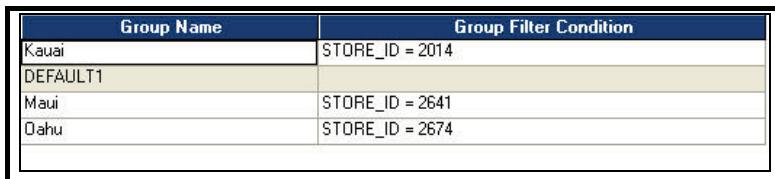
1. Import source tables from the database (Items, Orders, Order-Items and Stores).
2. Create three target tables as shown below and name them as follows:  
Tgt\_KAUAIFRANCHISE\_x , Tgt\_MAUIFRANCHISE\_x and  
Tgt\_OAHUFRANCHISE\_x.
3. The ports in all three target tables are as shown below:



The screenshot shows the 'Edit Tables' dialog box with the 'Columns' tab selected. The 'Select table:' dropdown is set to 'TGT\_KAUAIFRANCHISE\_x'. The table contains four columns with the following details:

	Column Name	Datatype	Prec	Scale	Not ...	Key Type
1	ORDER_ID	number	15	0	<input checked="" type="checkbox"/>	PRIMARY KEY
2	DATE_ENTERED	date	19	0	<input checked="" type="checkbox"/>	NOT A KEY
3	CUSTOMER_ID	number	15	0	<input checked="" type="checkbox"/>	NOT A KEY
4	ORDER_AMOUNT	number(p,s)	19	2	<input checked="" type="checkbox"/>	NOT A KEY

4. Drag all columns from Source qualifier into the Aggregator transformation and group on Store\_id and Order\_id.
5. Create an output port ORDER\_AMOUNT with the expression:  
 $SUM(PRICE * QUANTITY - DISCOUNT)$ .
6. Change PRICE, QUANTITY and DISCOUNT to input ports only.
7. Create a Router transformation with the name Rtr\_StoreOrder\_x. Link all the output ports from Aggregator to Router.
8. Select the Groups tab and enter the values under Group Name and Group Filter Condition as shown in the figure below:



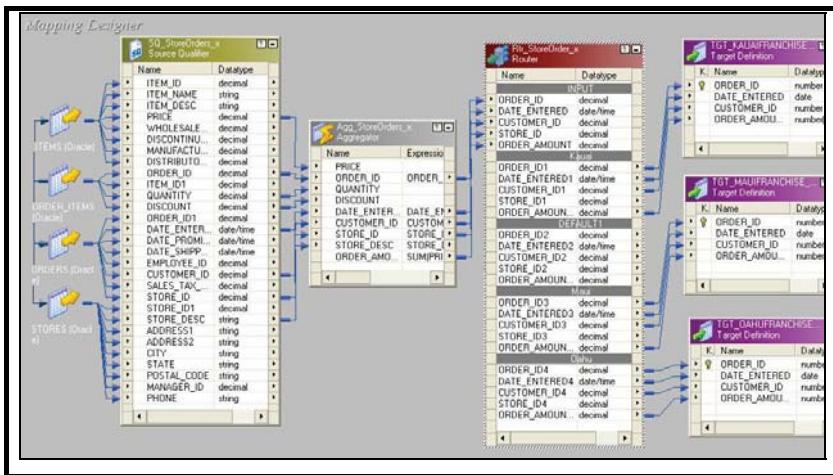
The screenshot shows the 'Groups' tab of the Router transformation configuration. It lists four groups with their corresponding group filter conditions:

Group Name	Group Filter Condition
Kauai	STORE_ID = 2014
DEFAULT1	
Maui	STORE_ID = 2641
Oahu	STORE_ID = 2674

9. The router transformation will generate three groups : Kauai, Maui, Oahu and a default group.

10. Link columns from each group to the respective targets. For example, the ports under the Kauai group are linked to the Tgt\_KAUAIFRANCHISE\_x target. This target table contains the order details for the store where store id = 2014.

11. The final mapping will look like one given below:



12. Create a workflow by name wf\_StoresOrders\_x and a session task by name s\_StoresOrders\_x.  
13. Run and monitor the Workflow.  
14. Verify the results.

## Lab 16.1. Lab 11-1 Sequence Generator Transformation

<b>Goals</b>	<ul style="list-style-type: none"><li>• Create a mapping that uses a Sequence Generator Transformation</li></ul>
<b>Time</b>	60 minutes
<b>Lab Setup</b>	A connection to the repository using PowerCenter Designer and Workflow Manager

### Background

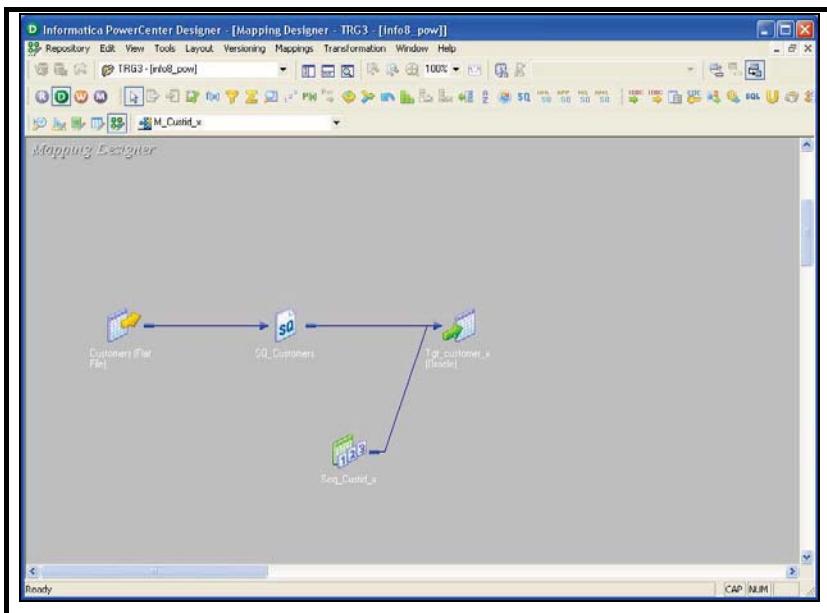
Customer source data arrives at each store in a flat file. Each file contains the customer name and other customer details. However, there is no unique id to identify each customer. The unique id for each customer will be generated through the mapping.

### Solution

- Use a Sequence Generator transformation to generate a unique id for each customer
- Use this generated Customer id as the primary key in the target table

TRANSFORMATION NAME	TYPE	DESCRIPTION
CUSTOMERS	Flat file Definition	Source definition
SQ_CUSTOMERS	Source Qualifier	Data source qualifier
SEQ_CUSTID_X	Sequence Generator	Used for generating a sequence id for each customer
TGT_CUSTOMER_X	Relational	Contains the unique customer id for each customer.

## Mapping Layout



**Final Output**

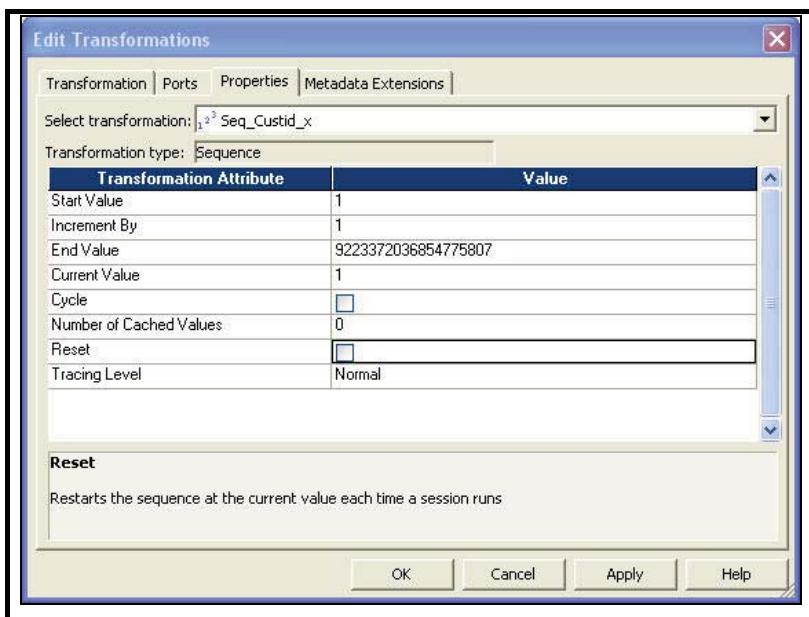
```
SQL> select * from tgt_customer_x;

CUSTOMER_ID COMPANY                               FIRST_NAME
-----  -----
1 Alfreds Futterkiste                           Maria
2 Ana Trujillo Emparedados y helados           Ana
3 Antonio Moreno Taquería                      Antonio
4 Around the Horn                             Thomas
5 Berglunds snabbköp                          Christina
6 Blauer See Delikatessen                     Hanna
7 Blondel père et fils                       Frederique
8 Bólido Comidas preparadas                  Martin
9 Bon app'                                 Laurence
10 Bottom-Dollar Markets                     Elizabeth
11 B's Beverages                            Victoria

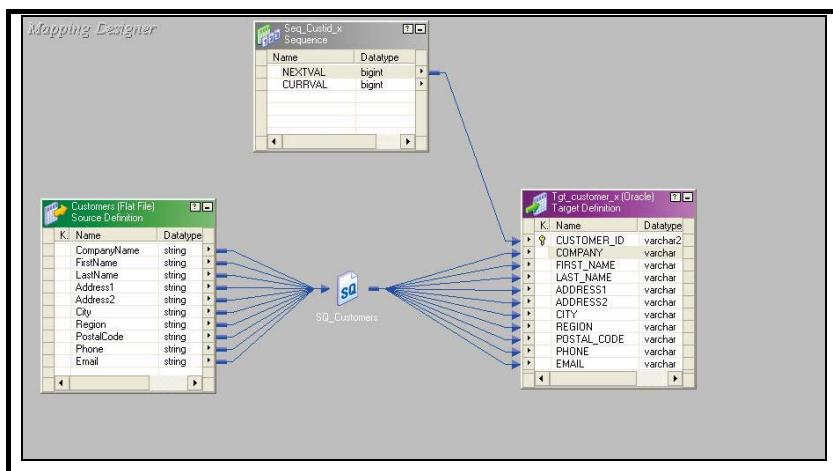
11 rows selected.
```

**Problem Solution**

1. Import source definition for Customers, which is a flat file uploaded on the server.
2. Create a target table and name it as Tgt\_Customer\_x, which is similar to the source. Add the CUSTOMER\_ID port as a Primary Key.
3. Create a mapping and name it as M\_Custid\_x.
4. Create the Sequence Generator transformation and name it as Seq\_Custid\_x.
5. Set the Start Value, End Value, Increment Value and other attributes as shown below. Check the Reset box.



6. Link NEXTVAL column of Sequence Generator to target table.
7. Link remaining columns from Source qualifier to target table.
8. Your mapping should look like the one shown below:



9. Create a workflow by name wf\_Custid\_x and a session task by name s\_Custid\_x.
10. Run and monitor the Workflow.
11. Verify the results.

## Lab 17.1. Lab 12-1 Stored Procedure Transformation

<b>Goals</b>	<ul style="list-style-type: none"><li>• Create a mapping that uses a Connected and Unconnected Stored Procedure Transformation</li></ul>
<b>Time</b>	60 mins
<b>Lab Setup</b>	A connection to the repository using PowerCenter Designer and Workflow Manager and mapping created in Lab 11-1

### Background

Customer source data arrives in a flat file from each store. At times, the customer names may contain some invalid data. All customer names should be validated to check for spaces, digits, special characters, etc. so that there is valid customer data in the Data Mart.

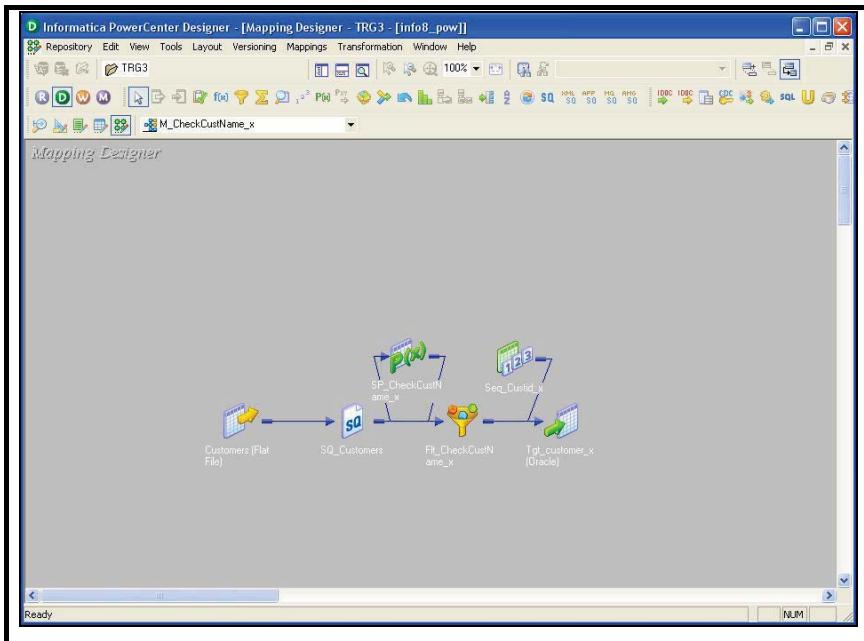
### Solution

- Use a Connected Stored Procedure transformation to validate the customer name
- The customer name is passed as a parameter to the Stored Procedure
- The Stored Procedure returns a 'V' value for valid names and 'I' for invalid names

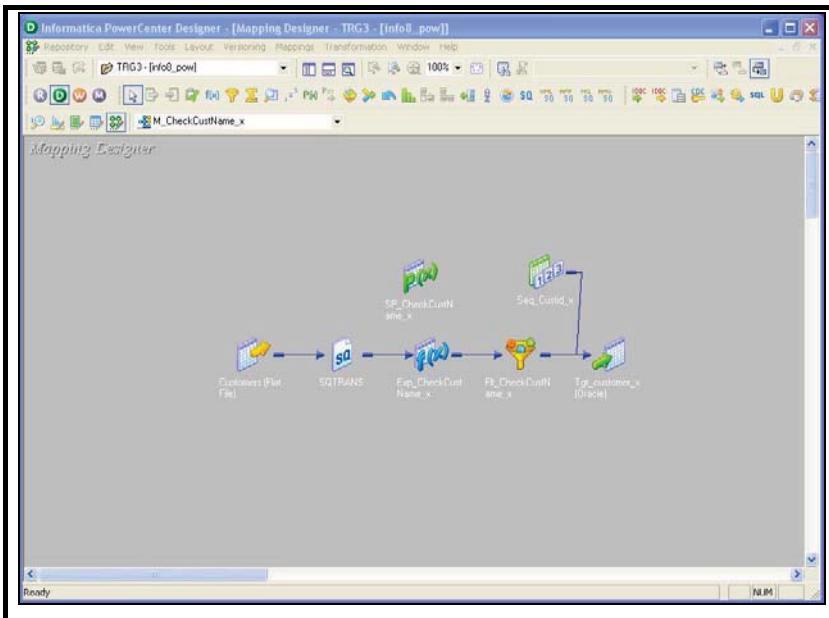
TRANSFORMATION NAME	TYPE	DESCRIPTION
CUSTOMERS	Flat file Definition	Source definition
SQ_CUSTOMERS	Source Qualifier	Data source qualifier
FLT_CHECKCUSTNAME_X	Filter	Allows rows with valid names to pass to the target
SEQ_CUSTOMERID_X	Sequence Generator	Used for generating a sequence id for each customer
SP_CHECKCUSTNAME_X	Stored Procedure	Receives the Customer Name as a parameter, validates it, i.e. checks for unwanted characters, and returns a flag 'V' if the name is valid and 'I' if the name is invalid
TGT_CUSTOMER_X	Relational	Contains customers whose names are valid

## Mapping Layout

### Connected Stored Procedure:



**Unconnected Stored Procedure:**



**Final Output**

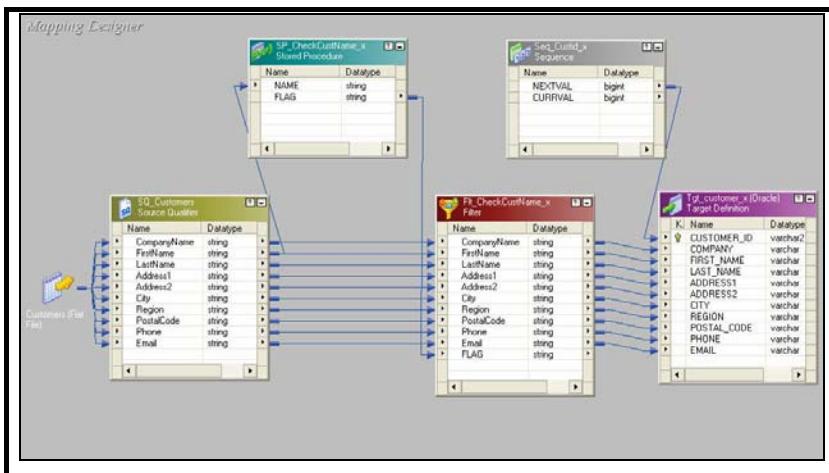
```
SQL> select * from tgt_customer_x;
```

CUSTOMER_ID	COMPANY	FIRST_NAME
1	Ana Trujillo Emparedados y helados	Ana
2	Antonio Moreno Taquería	Antonio
3	Around the Horn	Thomas
4	Berglunds snabbköp	Christina
5	Blauer See Delikatessen	Hanna
6	Blondel père et fils	Frederique
7	Bólido Comidas preparadas	Martin
8	Bon app'	Laurence
9	Bottom-Dollar Markets	Elizabeth
10	B's Beverages	Victoria

```
10 rows selected.
```

**Problem Solution****Connected Procedure**

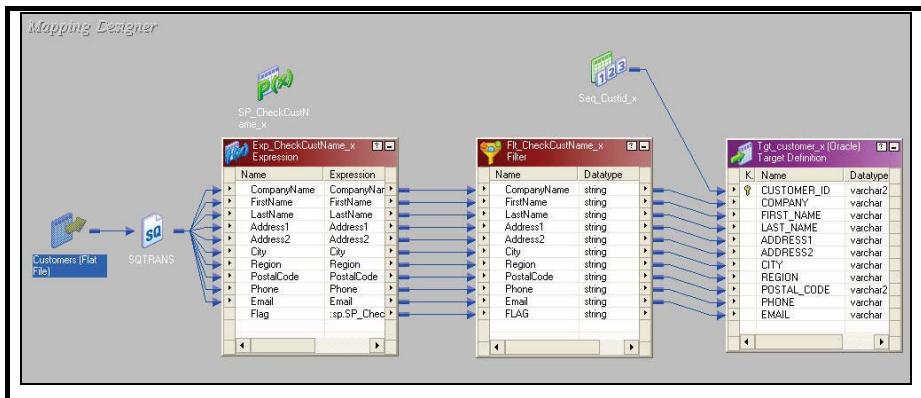
1. Copy the mapping M\_Custid\_x and rename it as M\_CheckCustName\_x.
2. Create a connected Stored Procedure transformation and name it as SP\_CheckCustName\_x.
3. Select the procedure name from the PROCEDURES folder.
4. The Stored Procedure transformation appears with two ports: Name and Flag.
5. Double click on the stored procedure transformation. Click on the Properties tab, select the connection information as db\_src\_x.
6. Note : Your Instructor will provide login details to import the procedure and name of the procedure. The procedure contains two parameters, Name which is an IN parameter and FLAG, which is an OUT parameter.
7. Delete the existing links between the Source Qualifier and Tgt\_Customer\_x.
8. Link Firstname port from Source Qualifier into the Name port of the Stored Procedure transformation.
9. Create a Filter transformation and link all ports from Source Qualifier into Filter transformation. Link the FLAG port from Stored Procedure into the Filter.
10. Create the filter condition : FLAG = 'V'. Link all ports except FLAG into the target.
11. The Sequence Generator transformation will generate the Customer\_id in the target. Only rows with valid customer names will pass to the target.
12. The final mapping should look as given below:



13. Create a Workflow by name wf\_CheckCustName\_x.
14. Run and monitor the Workflow.
15. Verify the Results.

### Unconnected Stored Procedure

1. Using the same mapping, remove the existing Stored Procedure transformation.
2. Create the Stored Procedure transformation again. Do not link it to any other transformation.
3. In the same mapping, create an Expression transformation before the Filter transformation. Link relevant Ports.
4. To call the Stored Procedure from the Expression transformation, enter the expression for the FLAG column, the newly added output port as:
5. :SP.SP\_CheckCustName\_x(FirstName, Proc\_result).
6. FirstName is passed as a parameter to the Stored Procedure and the value returned by the Stored Procedure will be available in the PROC\_RESULT variable.
7. The final mapping is shown as below:



8. Create the Workflow by name wf\_CheckCustName\_Unconnected\_x.
9. Run and monitor the Workflow.
10. Verify the results.

## Lab 18.1. Lab 13-1 Configure an Email Task

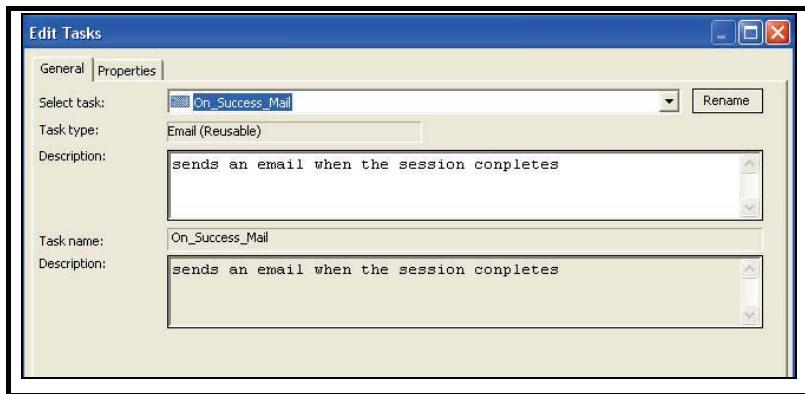
<b>Goals</b>	<ul style="list-style-type: none"><li>Configure a Workflow to send an email to designated recipients when the Integration service runs the workflow</li></ul>
<b>Time</b>	20 minutes
<b>Lab Setup</b>	Configured Mail server and a valid Workflow

### Background

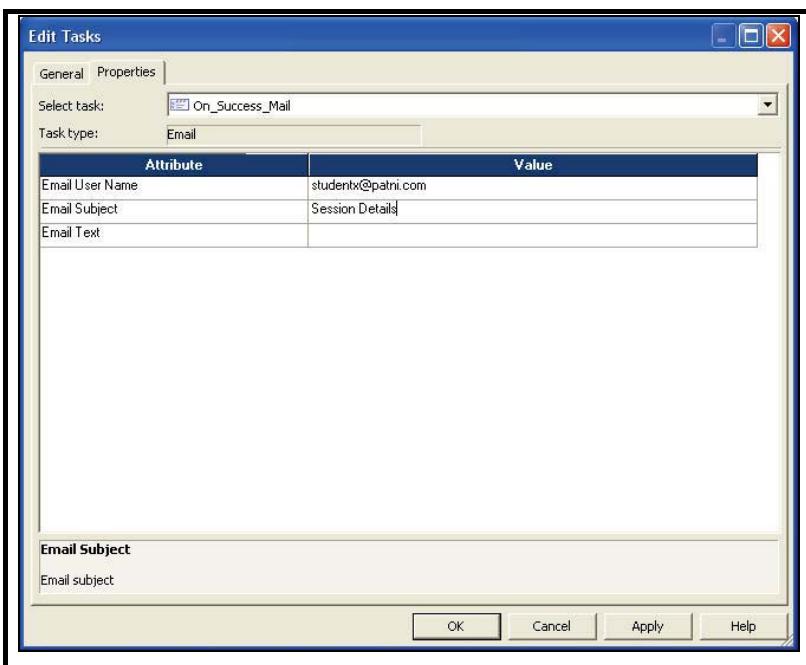
People, who are authorized to receive the session status, get an email, once the session has completed. The email gives details of number of rows loaded, rejected, time taken to complete, etc. An email task can be configured for a session. The Email Task gives information about the start time, completion time of the session, notification about the workflow status.

### Solution

- Create an Email task and place it in a Workflow
  1. Create an Email Task and name it as On\_Success\_Mail in Task Developer.
  2. Double-click on the email task. Click on the General tab, enter the description for the task as: "sends an email when the session completes"

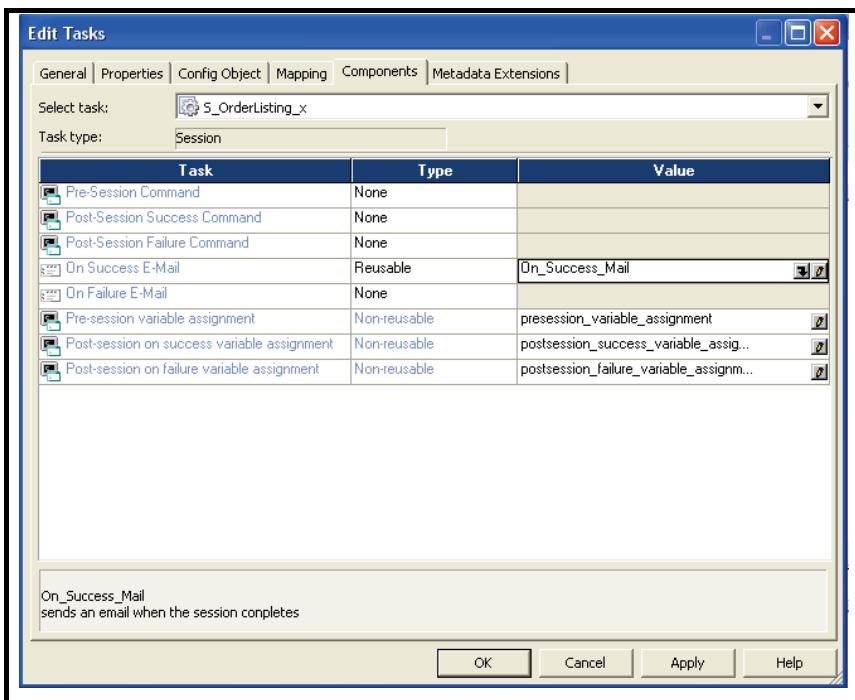


3. Select the Properties tab and enter the Email User Name and Email Subject details.

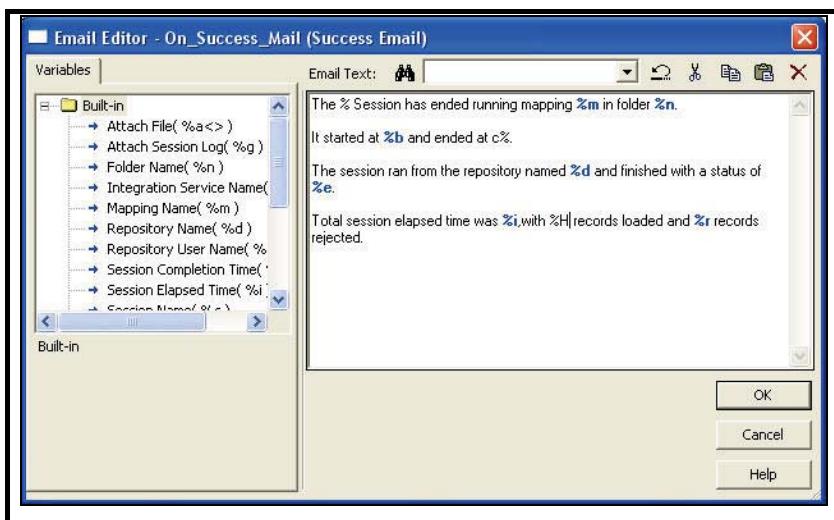


4. Create one more Email task, give the name as On\_Failure\_Mail and set its properties.
5. Switch to the Workflow Designer and drag the wf\_OrderListing\_x Workflow created in Lab 9-1.
6. Double-click on the Session Task s\_OrderListing\_x.
7. Click on the Components tab.
8. Click On Success E-Mail option select Reusable from the drop down list.
9. Click on the  icon and select On\_Success\_Mail from the drop down list.

10. Click on the icon (shown highlighted in the figure).



11. Enter the email text. Here you can select any post-session built-in Email variables, useful for including important session information.



12. Select the reusable Email task for On Failure E-Mail. Enter the details required.
13. Run the Workflow.
14. Verify the results.

**Note:** The concerned people will receive an email regarding the status of the Workflow, subject to mail server configuration.

## Lab 19.1. Lab 14-1 Configure a Command Task

<b>Goal</b>	<ul style="list-style-type: none"><li>Configure a command task to delete reject files after successful completion of a Workflow</li></ul>
<b>Time</b>	10 minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Workflow Manager

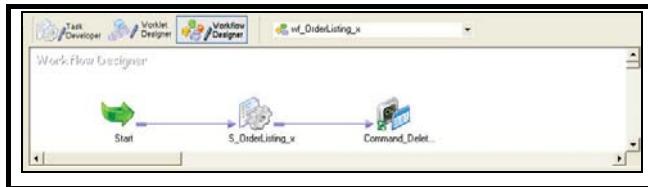
### Background

Some reject files created during a Workflow run need to be deleted.

### Solution

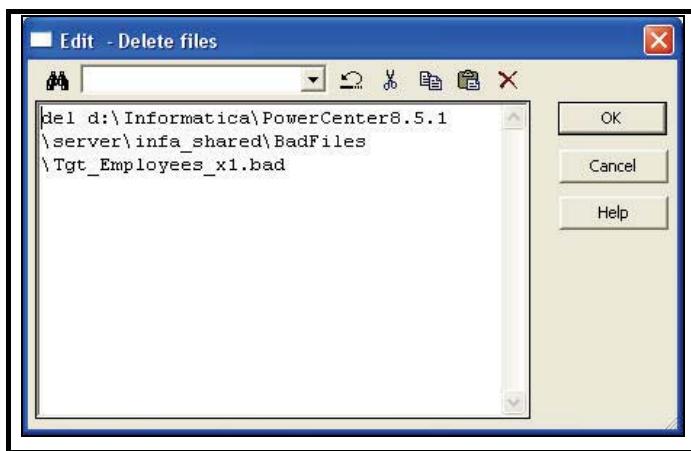
- A Command Task can be configured to specify shell or DOS commands, to delete reject files, copy a file, or archive target files.
- Use the Command Task to delete reject files.

### Workflow Layout



1. Create a Command task in Task Developer and name it as Command\_Delete\_x.
2. In the Commands tab, add a new command and name it as DeleteFiles.
3. Enter the command as shown below:

**Note:** The command can be any valid UNIX command or shell script for UNIX servers, or, any valid DOS or batch file for Windows servers.



**Note:** Check out the actual path for the reject files with the instructor.

4. Open the Workflow wf\_Employee\_Name\_x create in Lab 3-1.
5. Run the Workflow.
6. Verify the results.

**Note:** The commands specified in the Command Task are executed on the Informatica Server. To verify the execution of the commands given in the Command Task you need to have privileges to login to the Informatica Server and view the BadFiles directory that has all the reject files.

## Lab 20.1. Lab 15-1 Event-Wait task

<b>Goal</b>	<ul style="list-style-type: none"><li>• Create Event Wait Task and Event Raise task</li></ul>
<b>Time</b>	20 minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Workflow Manager

### I. Create Workflow and Task

1. Create a New Workflow.
2. Create an Event Wait task.
3. Go to the Events tab of the Event wait task and select the Radio Button PreDefined.
4. Enter the following in the name of file to watch.  
iii. D:\Informatica\PowerCenter9.5.1\server\infa\_shared\SrcFiles\dept.txt  
(Any path on the Powercenter server)
5. Connect the Event wait task to a reusable Command task.

**Note:** The Command task may contain any command

e.g del D:\Informatica\PowerCenter9.5.1\server\infa\_shared\SrcFiles\dept.txt

6. Execute the workflow.

**Note:** The Event Wait Task will wait for the file to be present on the above path(on the Powercenter server). After the file arrives on that path then only the Reusable command task begins execution.

## Lab 21.1. Lab 16-1 Pre-SQL and Post-SQL

<b>Goal</b>	<ul style="list-style-type: none"><li>• Create Pre-SQL and Pre-SQL</li></ul>
<b>Time</b>	30 minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Workflow Manager

### I. Create Mapping and Workflow

1. Create Mapping with the Source as Dept table and the target as TGT\_DEPT table.
2. The Structure of the TGT\_DEPT table is same as that of the Dept table.
3. The mapping does not have any transformations.
4. Create an index PKTGTDNO on DEPTNO column of TGT\_DEPT table.
5. Create a Workflow with a session task. In the Session, Mapping tab ensure that the Target Load type is BULK.
6. Execute the workflow.
7. The execution fails since BULK loading does not permit Index defined on it.

### II. Pre-SQL and Post-SQL in target

1. In the Session, Mapping tab perform the following
2. In the Pre-SQL Drop index PKTGTDNO.
3. In the Post-SQL Create an index PKTGTDNO on DEPTNO column of TGT\_DEPT.
4. Execute the work flow. The execution succeeds and the Target table gets populated. The index gets created on the target table.

### III. Pre-SQL and Post-SQL in source

1. In the Session, Mapping tab do the following
2. In the Post-SQL Drop index PKSRCRCDNO.
3. In the Pre-SQL Create an index PKSRCRCDNO on DEPTNO column of SRC\_DEPT.
4. In the Session, Mapping tab ensure that the Target Load type is NORMAL.
5. Execute the work flow. The execution succeeds and the Target table gets populated. The index gets created on the target table.

**Note:** Index may be required to be created on the source table before reading data, to increase access speed, but it may be dropped in Post-SQL since it might reduce performance.

## Lab 22.1. Lab 17-1 Multiple Source Files(Indirect File)

<b>Goal</b>	<ul style="list-style-type: none"><li>• Use Multiple source files</li></ul>
<b>Time</b>	20 minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Workflow Manager

### I. Create a File List

1. Create a file (indfile.txt) in a directory local to the PowerCenter Server. The file will contain the names and directories of each source file to be used in the session.

D:\Informatica\PowerCenter9.5.1\server\infa\_shared\SrcFiles\dept1.txt

D:\Informatica\PowerCenter9.5.1\server\infa\_shared\SrcFiles\dept2.txt

D:\Informatica\PowerCenter9.5.1\server\infa\_shared\SrcFiles\dept3.txt

D:\Informatica\PowerCenter9.5.1\server\infa\_shared\SrcFiles\dept4.txt

**Note:** The paths mentioned above must be local to the PowerCenter Server machine. The data present in all the above files is similar to the dept table in oracle.

### II. Create a Mapping and Workflow

1. Create a mapping which uses one of the source flat files as the source destination and TGT\_DEPT as the target table.
2. The mapping does not have any transformations other than the source qualifier transformation. Drag the fields from Source Qualifier transformation to the target.
3. Create a workflow and make the following changes to the session task :
4. Set the Source filename to indfile.txt.
5. Set the Source filetype to Indirect.
6. Execute the workflow.
7. The target table will be populated with rows from all the source files.

## Lab 23.1. Lab 18-1 Mapping Parameter

<b>Goal</b>	<ul style="list-style-type: none"><li>• Create a Mapping Parameter</li></ul>
<b>Time</b>	45 minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Workflow Manager

### I. Create Parameter File

1. Create a parameter file by the name Param\_file.txt  
and put it in srcfiles folder  
(E.g D:\Informatica\PowerCenter9.5.1\server\infa\_shared\SrcFiles\)

The contents of the file are

```
[TRG2.S_MappingParameter]  
$$PAY=2000  
$$JO=MANAGER
```

### II. Create Mapping

1. Create a New Mapping.
2. In the Mapping create parameters \$\$PAY and \$\$JO.  
**Hint:** (Mappings->Parameters and Variables).  
**Note:** Type is parameter.
3. Drag the Emp table from the sources to the Mapping (the source qualifier is added automatically).
4. Create a Filter Transformation and in the Filter Condition enter:  
`SAL>$$PAY AND JOB=$$JO`

### III. Create Workflow

1. Create a workflow and a session task.
2. Make appropriate changes in the relational connection for the session task.
3. In the Properties tab of the Session put the parameter file name as:  
`$PMSourceFileDir\Param_file.txt`
4. Execute the workflow.

## Lab 24.1. Lab 19-1 Mapping Variable

<b>Goal</b>	<ul style="list-style-type: none"><li>• Create a Mapping Variable</li></ul>
<b>Time</b>	45 minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Workflow Manager

### I. Create source definition and target.

1. Import source definition for Customers, which is a flat file uploaded on the server.
2. Create a target table, which is similar to the source. Add the CUSTOMER\_ID port as a Primary Key. Name the target as Tgt\_Customer\_x.

### II. Create Mapping and Workflow.

1. Create a mapping with Source as Customers(flat file) and Target as Tgt\_Customer\_x.
2. Create a mapping variable as \$\$rownum. Set its aggregation to Count.
3. Create an Expression transformation and drag all fields from Source Qualifier to the expression transformation.
4. Create an output port in expression transformation and name it as o\_rownum. Let the data type be integer.
5. In expression editor for this output port enter the following:

```
SETCOUNTVARIABLE($$rownum)
```

6. Connect o\_rownum in expression transformation to CUSTOMER\_ID in target.
7. Connect the remaining fields from expression to the Target.
8. Save the mapping and create a session for the mapping.
9. Truncate Tgt\_Customer\_x table using following script:

```
Truncate table Tgt_Customer_x;
```

10. Execute the session and verify the result in Tgt\_customers\_x.
11. Verify the result in Tgt\_customers\_x table.

**Note:** The target table contains customer\_id whose value ranges from 1 to 7. If the same session is run again the values generated will be from 8 onwards.

Right click on the session and click on **view persistent values**. It shows the value of \$\$rownum which is 7. We can click on **Reset values** to reset the value.

## Lab 25.1. Lab 20-1 FTP Connection

<b>Goal</b>	<ul style="list-style-type: none"><li>• Use an FTP connection</li></ul>
<b>Time</b>	20 minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Workflow Manager

### I. Create a Mapping

1. Import a flat file(stored on your local machine) in the source analyzer. Create a target table in oracle with the same definition as that of source.
2. Drag the ports of the Source Qualifier transformation to the Target table.

### II. Create a workflow

1. Goto the Workflow Manager.
2. Create FTP connection.
3. Give the username,password,hostname and default home directory.  
**e.g.** If you are connecting to a linux server give the appropriate username and password. Give the host name as 172.21.20.10(IP address of the linux server) and the default home directory as /home/testuser2(path where the source file resides).
4. Create a Session task based on the mapping created above.
5. Select Edit Tasks->Mapping. Select Sources. Click on FTP. Select the FTP connection created by you.
6. Click on override in the FTP connection browser. The host name and the default directory are given. Enter the remote file name(E.g Sales.txt)
7. Execute the workflow.
8. Observe that the target table(Relational) contains the rows from the source(FTP – Linux server).

**Note:** Ensure the the FTP service is started on the FTP server(Linux server in this case).

## Lab 26.1. Lab 21-1 Dynamic Lookup

<b>Goal</b>	<ul style="list-style-type: none"><li>• Create a Dynamic Lookup</li></ul>
<b>Time</b>	45 minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Workflow Manager

### I. Create source definition and target

1. Import source definition for Daily Transaction, which is a flat file uploaded on the server(Daily\_transaction.txt).
2. Create a target table using the script below:

```
CREATE TABLE Tgt_Accounts_x
(
    ACC_KEY number NOT NULL,
    ACC_NO number(12),
    BALANCE number(10)
);
```

```
ALTER TABLE TGT_ACCOUNTS_X ADD PRIMARY KEY (ACC_KEY);
```

### II. Create Mapping and Workflow

1. Create a mapping with Source as Daily\_transaction(flat file) and Target as Tgt\_Accounts\_x.
2. Create a Lookup Transformation. Use target table(Tgt\_Accounts\_x) as lookup table.
3. Drag Account\_Number and Current\_Balance columns from source to Lookup Transformation.
4. Double click on Lookup Transformation.
5. Go to condition tab. Put the condition as follows:  
ACC\_NO= Account\_number.
6. Go to properties tab. Check the options Dynamic Lookup Cache and Insert Else Update.
7. Go to ports tab. Change the datatype of ACC\_KEY port to integer. Set Associated port for ACC\_KEY as Sequence-ID, ACC\_NO as Account\_Number,BALANCE as Current\_Balance.
8. Create a filter transformation. Drag NewLookupRow,ACC\_KEY, ACC\_NO and BALANCE from lookup transformation to filter transformation.
9. Open filter transformation. Go to properties tab and in filter condition put following expression:

```
NewLookupRow = 1 OR NewLookupRow = 2
```

10. Create an update strategy transformation. Drag all the ports from filter transformation to update strategy transformation.
11. Open update strategy transformation. Go to properties tab and in Update Strategy Expression put following expression:  
`IIF(NewLookupRow=1, DD_INSERT, DD_UPDATE)`
12. Connect ACC\_KEY, ACC\_NO and BALANCE columns to target.
13. Create workflow and session.
14. Ensure that in the session properties Data Driven is selected.
15. In session go to mapping tab. Click on target instance.
16. Check Insert and Update as update. Uncheck delete.
17. Save the workflow and execute it.
18. Verify the result in Tgt\_Accounts\_x table.

**Note:**

1. When you check Dynamic lookup cache option in lookup property NewLookupRow port is automatically created.
2. When you change datatype of ACC\_KEY from decimal to integer, Sequence-Id automatically comes in Associated port pulldown list.
3. New sequence number is not generated for rows where matching account number is found. New sequence number is generated only for rows that are inserted.
4. New sequence number always starts from '*maximum value for key in table*'+1.

## Lab 27.1. Lab 22-1 Using Mapping Wizard for SCD Type1

<b>Goal</b>	<ul style="list-style-type: none"> <li>Create a Type 1 mapping using Wizard</li> </ul>
<b>Time</b>	45 minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Workflow Manager

### I. Create source definition

1. Create tables Customers using following scripts:-

```
CREATE TABLE CUSTOMERS
```

```
(
```

```
CUSTOMERID number(3) NOT NULL,
```

```
COMPANYNAME varchar2(34),
```

```
FIRSTNAME varchar2(10),
```

```
LASTNAME varchar2(8),
```

```
ADDRESS varchar2(29),
```

```
CITY varchar2(10)
```

```
);
```

```
ALTER TABLE CUSTOMERS ADD PRIMARY KEY (CUSTOMERID);
```

2. Use the flat file customers from the **Srcfiles** folder to populate data in the Customers relational table.
3. Create a Mapping to perform the above mentioned task.
4. Create a Workflow, task to execute the table.
5. The Customers table contains the following data:

CUSTOMERID	COMPANYNAME	FIRSTNAME	LASTNAME	ADDRESS	CITY
101	Alfreds Futterkiste	María	Anders	Obere Str. 57	Berlin
102	Ana Trujillo Emparedados y helados	Ana	Trujillo	Auda, de la Constitucion 2222	Mexico
103	Antonio Moreno Taqueria	Antonio	Moreno	Mataderos 2312	Mexico
104	Around the Horn	Thomas	Hardy	128 Hanover Sq.	London
105	Berglunds snabbköp	Christina	Berglund	Berguvsvagen 8	Luleå
106	Blauer See Delikatessen	Hanna	Noos	Görsterstr. 57	Mannheim
107	Blondel père et fils	Frederique	Citeaux	24 Place Kieber	Strasbourg
108	Bolido Comidas preparadas	Martin	Sommer	C/ Aragóil 67	Madrid
109	Bon app	Laurence	Lebihan	12 Rue Des Bouchers	Marseille
110	Bottom-Dollar Markets	Elizabeth	Lincoln	23 Tsawassen Blvd.	Tsawassen
111	B's Beverages	Victoria	Ashworth	Fauntleroy Circus	London

6. Import the Customers table in the **Source Analyzer**.

### II. Create Mapping

- In the Mapping Designer, choose **Mappings-Wizards-Slowly Changing Dimension**.
- Enter a mapping name and select **Type 1 Dimension**, and click **Next**.

3. Select the **source** as **Customers**.
4. In the New Target Table Enter the target name as **T\_Customers**.  
**Note:** This target table will just be added to the Repository. Drag the definition to the **Target Designer** and **Generate/Execute SQL** to create the target table.
5. Select the column Customer ID to use as a lookup condition from the Target Table Fields list and click **Add**.  
**Note:** The wizard adds selected columns to the Logical Key Fields list.
6. Select the columns **City** and **Address**. Click on **Add** to add in the list of **compare for changes**.  
**Note:** The wizard adds selected columns to the Fields to Compare for Changes list. When you run the workflow containing the session, the PowerCenter Server compares the columns in the Fields to Compare for Changes list between source rows and the corresponding target (lookup) rows. If the PowerCenter Server detects a change, it marks the row changed
7. Click **Finish**.
8. To save the mapping, choose **Repository-Save**.

### III. Create Workflow

1. Create a workflow and session task to execute the mapping.
2. The target table gets populated with all the rows from source.
3. Update the Source table with the following statement:

```
update CUSTOMERS set companyname = 'PATNI'  
WHERE CUSTOMERID=101;
```
4. Commit the transaction.
5. Execute the session task.  
**Note:** The change made in the source is NOT reflected in the target since the field companyname was not chosen in the list of **Compare for changes**.
6. Update the Source table with the following statement:

```
update CUSTOMERS set city = 'Mumbai' WHERE CUSTOMERID=101;
```
7. Commit the transaction.
8. Execute the session task.  
**Note:** The change made in the source is reflected in the target since the field city was chosen in the list of **Compare for changes**.

### IV. Explanation of the Mapping:

The Type 1 Dimension mapping performs the following tasks:

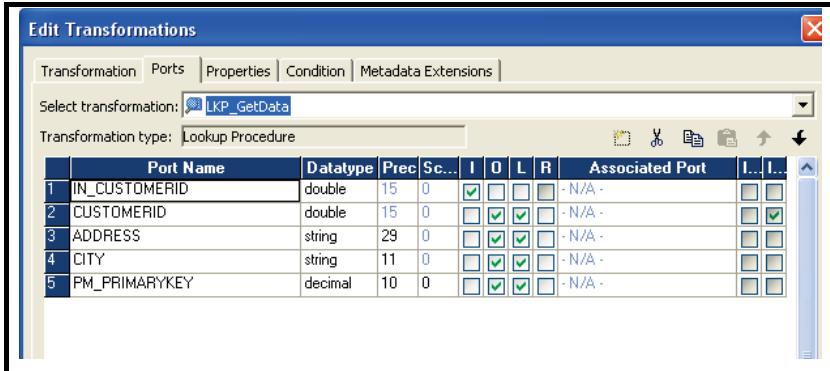
1. Selects all rows.
2. Caches the existing target as a lookup table.
3. Compares logical key columns in the source against corresponding columns in the target lookup table.

4. Compares source columns against corresponding target columns if key columns match.
5. Flags new rows and changed rows.
6. Creates two data flows: one for new rows, one for changed rows.
7. Generates a primary key for new rows.
8. Inserts new rows to the target.
9. Updates changed rows in the target, overwriting existing rows

**Note:** The Type 1 Dimension mapping uses a Lookup and an Expression transformation to compare source data against existing target data. The LookUp table is the target table T\_customers.

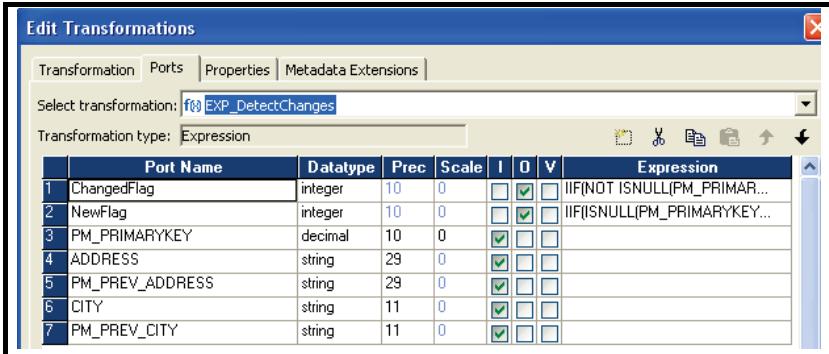
## V. LookUp Transformation

1. The IN\_CUSTOMERID field is the field from the source table which is taken from the SQ\_CUSTOMERS.



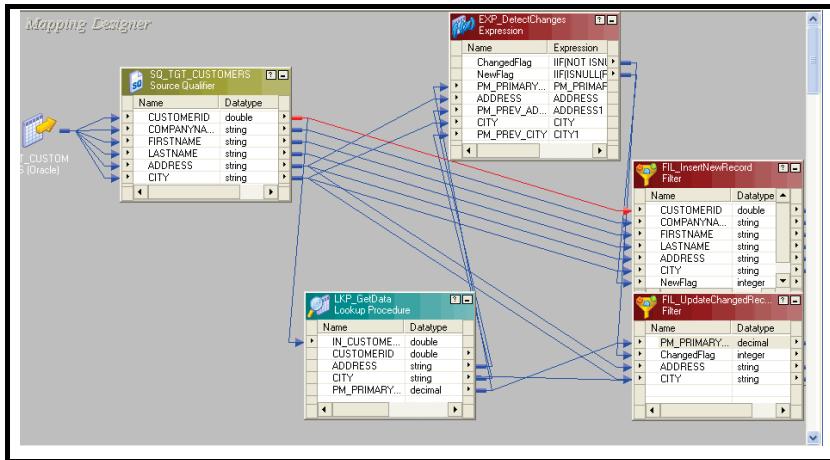
Port Name	Datatype	Prec	Sc...	I	O	L	R	Associated Port	I...	I...
1 IN_CUSTOMERID	double	15	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input type="checkbox"/>
2 CUSTOMERID	double	15	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3 ADDRESS	string	29	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input type="checkbox"/>
4 CITY	string	11	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input type="checkbox"/>
5 PM_PRIMARYKEY	decimal	10	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input type="checkbox"/>

## VI. Expression transformation



Port Name	Datatype	Prec	Scale	I	O	V	Expression
1 ChangedFlag	integer	10	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IIF(NOT ISNULL(PM_PRIMARYKEY...)
2 NewFlag	integer	10	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IIF(ISNULL(PM_PRIMARYKEY...)
3 PM_PRIMARYKEY	decimal	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
4 ADDRESS	string	29	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5 PM_PREV_ADDRESS	string	29	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6 CITY	string	11	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7 PM_PREV_CITY	string	11	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

- The ADDRESS and CITY fields are connected from the SQ\_Customers, the PM\_PREV\_ADDRESS,PM\_PREV\_CITY,PM\_PRIMARYKEY are connected from the Look Up transformation.



- The value of ChangedFlag is computed as :

```

IIF(NOT ISNULL(PM_PRIMARYKEY))
AND
(
  DECODE(ADDRESS,PM_PREV_ADDRESS,1,0) = 0
OR
  DECODE(CITY,PM_PREV_CITY,1,0) = 0
),
TRUE,FALSE)

```

- The value of NewFlag is computed as :

```
IIF(ISNULL(PM_PRIMARYKEY),TRUE,FALSE)
```

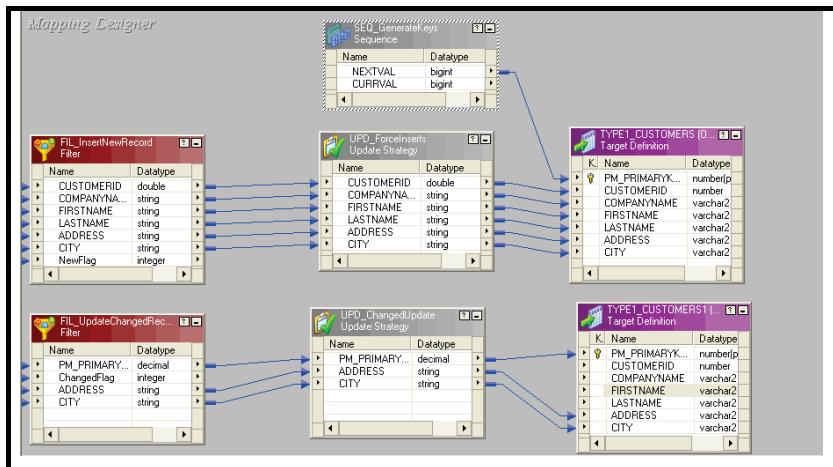
## VII. Filter Transformation

- The Filter Transformation FIL\_InsertNewRecord allows only those records to pass through the filter where the Filter Condition of NewFlag i.e only those records which are not present in the target.
- All the fields of this transformation are connected through the Source Qualifier transformation. The field New flag is connected through Expression transformation.

3. The Filter Transformation FIL\_UpdateChangedRecord allows only those records to pass through the filter where the Filter Condition of Changed i.e only those records which have been changed.
4. The field ADDRESS and CITY of this transformation are connected through the Source Qualifier transformation. The field Changed flag is connected through Expression transformation. The field PM\_PRIMARYKEY is connected through the LookUp transformation.

### VIII. Update Strategy Transformation

1. The UPD\_ForceInserts has the Update Strategy Expression DD\_INSERT and inserts row in the target table.
2. The UPD\_ChangedUpdate has the Update Strategy Expression DD\_UPDATE and updates rows in the target table.



## Lab 28.1. Lab 23-1 Using Mapping Wizard for SCD Type2

<b>Goal</b>	• Create a Type 2 mapping using Wizard
<b>Time</b>	45 minutes
<b>Lab Setup</b>	Successful connection to the repository using PowerCenter Workflow Manager

### I. Create source definition.

1. Create tables Customers using following scripts:

```
CREATE TABLE CUSTOMERS
(
  CUSTOMERID number(3) NOT NULL,
  COMPANYNAME varchar2(34),
  FIRSTNAME varchar2(10),
  LASTNAME varchar2(8),
  ADDRESS varchar2(29),
  CITY varchar2(10)
);
```

```
ALTER TABLE CUSTOMERS ADD PRIMARY KEY (CUSTOMERID);
```

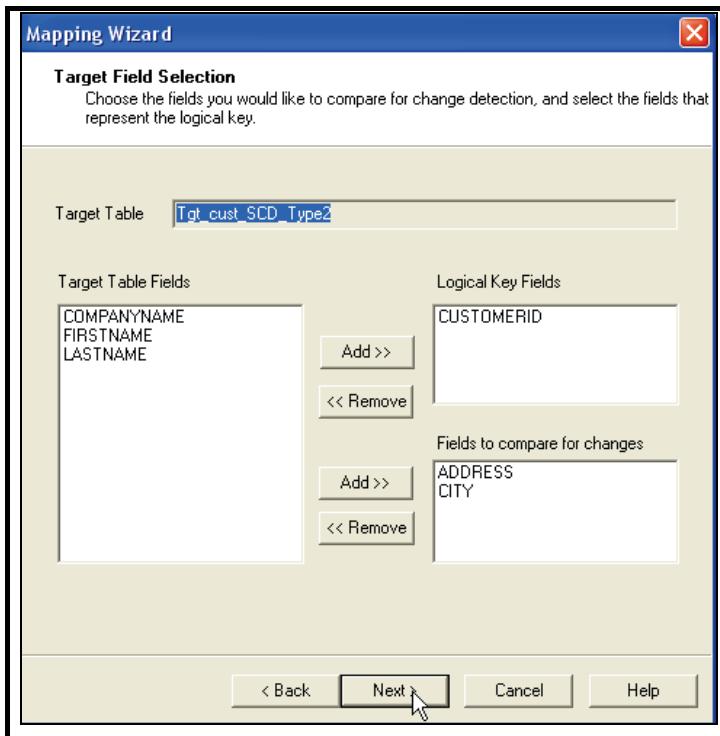
2. Use the flat file **customers** from the **Srcfiles** folder to populate data in the Customers relational table.
3. Create a Mapping to perform the above mentioned task.
4. Create a Workflow, task to execute the mapping.
5. The Customers table contains the following data.

CUSTOMERID	COMPANYNAME	FIRSTNAME	LASTNAME	ADDRESS	CITY
101	Alfreds Futterkiste	Maria	Anders	Obere Str. 57	Berlin
102	Ana Trujillo Emparedados y helados	Ana	Trujillo	Avda. de la Constitucion 2222	Mexico
103	Antonio Moreno Taqueria	Antonio	Moreno	Mataderos 2312	Mexico
104	Around the Horn	Thomas	Hardy	120 Hanover Sq.	London
105	Berglunds snabbköp	Christina	Berglund	Berguvsvagen 8	Lulea
106	Blauer See Delikatessen	Hanna	Moos	Forsterstr. 57	Mannheim
107	Blondel pere et fils	Frederique	Citeaux	24 Place Kleber	Strasbourg
108	Bolido Comidas preparadas	Martin	Sommer	C/ Araquil 67	Madrid
109	Bon app	Laurence	Lebihan	12 Rue Des Bouchers	Marseille
110	Bottom-Dollar Markets	Elizabeth	Lincoln	23 Tsawassen Blvd.	Tsawassen
111	B's Beverages	Victoria	Ashworth	Fauntleroy Circus	London

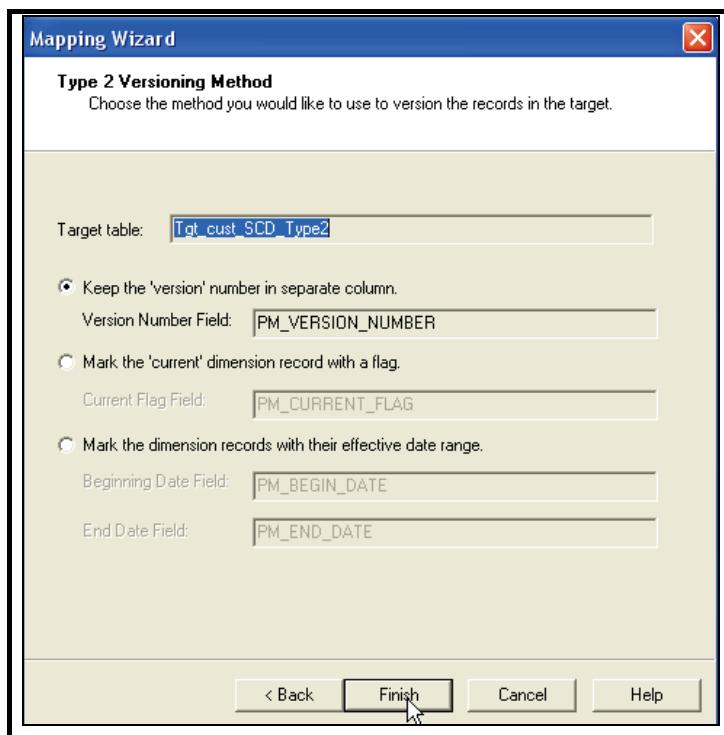
6. Import the Customers table in the **Source Analyzer**.

## II. Create Mapping

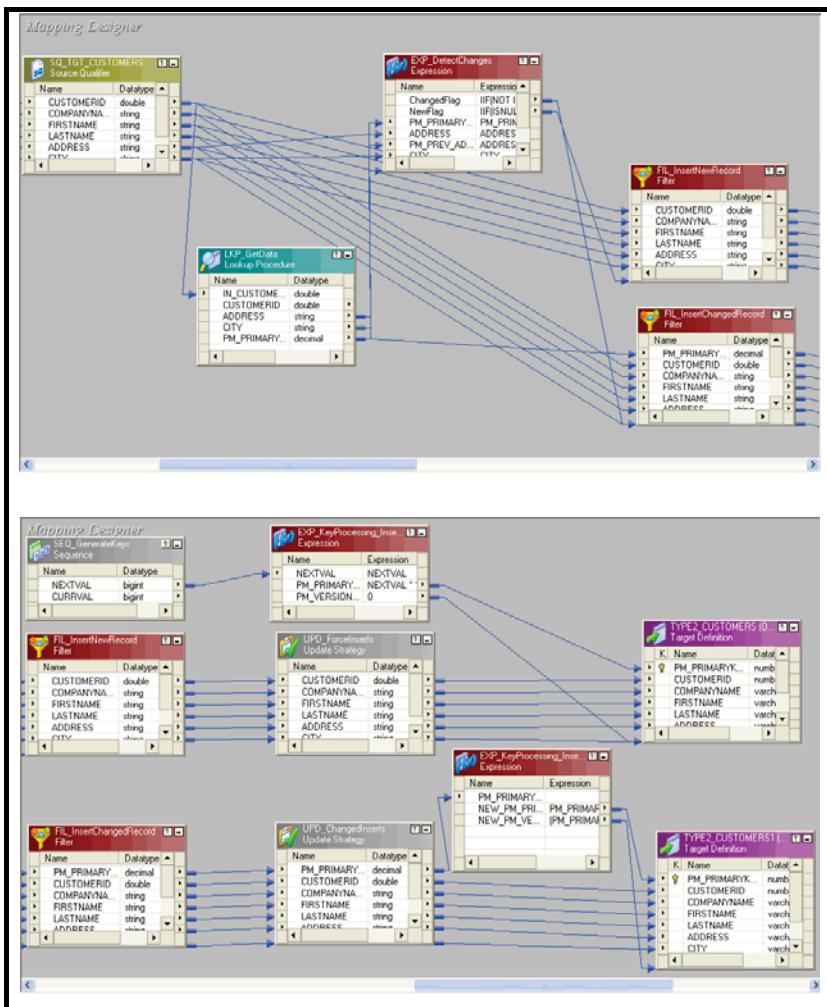
1. In the Mapping Designer, choose Mappings-Wizards-Slowly Changing Dimensions.
2. Enter a mapping name and select Type 2 Dimension. Click Next.
3. Select the **Logical Key Fields** and the **Fields to compare for changes**.



4. Select the Versioning Method. Click **Finish**.



5. The Final Mapping looks like this:



**Note:** In the Expression Transformation EXP\_KeyProcessing\_InsertChanged, use the following expression to increment the existing version number by one:

$(PM\_PRIMARYKEY + 1) \% 1,000.$

## Lab 29.1. Appendix A – Sources and Targets used in the Mappings

### Relational Tables

#### STORES

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
STORE_ID	Number(28)	Decimal(28)	Decimal(28)	PK/NOT NULL	ORDERS
STORE_DESC	Varchar2(72)	Varchar(72)	Varchar(72)	NOT NULL	
ADDRESS1	Varchar2(72)	Varchar(72)	Varchar(72)	NOT NULL	
ADRESS2	Varchar2(72)	Varchar(72)	Varchar(72)		
CITY	Varchar2(32)	Varchar(32)	Varchar(32)	NOT NULL	
STATE	Varchar2(2)	Varchar(2)	Varchar(2)	NOT NULL	
POSTAL_CODE	Varchar2(10)	Varchar(10)	Varchar(10)	NOT NULL	
MANAGER_ID	Number(28)	Decimal(28)	Decimal(28)	NOT NULL	
PHONE	Varchar2(32)	Varchar(32)	Varchar(32)	NOT NULL	

#### ITEMS

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
ITEM_ID	Number(28)	Decimal(28)	Decimal(28)	PK/NOT NULL	ORDER_ITEMS
ITEM_NAME	Varchar2(72)	Varchar(72)	Varchar(72)	NOT NULL	
ITEM_DESC	Varchar2(40)	Varchar(40)	Varchar(40)		
PRICE	Number(10,2)	Money(19,4)	Decimal(10,2)	NOT NULL	
WHOLESALE_COST	Number(10,2)	Money(19,4)	Decimal(10,2)	NOT NULL	
DISCONTINUED_FLAG	Number(1)	Decimal(1)	Decimal(1)	NOT NULL	

MANUFACTURER_ID	Number(28)	Decimal(28)	Decimal(28)	FK/NOT NULL	MANUFACTURERS
DISTRIBUTOR_ID	Number(28)	Decimal(28)	Decimal(28)	FK/NOT NULL	DISTRIBUTORS

**ORDER\_ITEMS**

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
ORDER_ID	Number(28)	Decimal(28)	Decimal(28)	FK/NOT NULL	ORDERS
ITEM_ID	Number(28)	Decimal(28)	Decimal(28)	FK/NOT NULL	ITEMS
QUANTITY	Number(28)	Decimal(28)	Decimal(28)	NOT NULL	
DISCOUNT	Number(10,2)	Money(19,4)	Decimal(10,2)	NOT NULL	

## ORDERS

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
ORDER_ID	Number(28)	Decimal(28)	Decimal(28)	PK/NOT NULL	ORDER_ITE MS
DATE_ENTERED	Date	Datetime	Datetime	NOT NULL	
DATE_PROMISE_D	Date	Datetime	Datetime	NOT NULL	
DATE_SHIPPED	Date	Datetime	Datetime	NOT NULL	
EMPLOYEE_ID	Number(28)	Decimal(28)	Decimal(28)	FK/NOT NULL	EMPLOYEES
CUSTOMER_ID	Number(28)	Decimal(28)	Decimal(28)	FK/NOT NULL	CUSTOMERS
SALES_TAX_RATE	Number(5,4)	Money(19,4)	Decimal(5,4)	NOT NULL	
STORE_ID	Number(28)	Decimal(28)	Decimal(28)	FK/NOT NULL	STORES

## MANUFACTURERS

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
MANUFACTURER_ID	Number(28)	Decimal(28)	Decimal(28)	PK/NOT NULL	ITEMS
MANUFACTURER_NAME	Varchar2(48)	Varchar(48)	Varchar(48)	NOT NULL	

## DISTRIBUTORS

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
DISTRIBUTOR_ID	Number(28)	Decimal(28)	Decimal(28)	PK/NOT NULL	ITEMS
DISTRIBUTOR_NAME	Varchar2(48)	Varchar(48)	Varchar(48)	NOT NULL	
CONTACT_PERSON	Varchar2(48)	Varchar(48)	Varchar(48)		

CONTACT_PHONE	Varchar2(32)	Varchar(32)	Varchar(32)		
CONTACT_EMAIL	Varchar2(72)	Varchar(72)	Varchar(72)		

## EMPLOYEES

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
EMPLOYEE_ID	Number(28)	Decimal(28)	Decimal(28)	PK/NOT NULL	ORDERS
DEPTID	NUMBER(2)	Decimal(28)	Decimal(28)	FK/NOT NULL	DEPARTMENT
LAST_NAME	Varchar2(30)	Varchar(30)	Varchar(30)	NOT NULL	
FIRST_NAME	Varchar2(30)	Varchar(30)	Varchar(30)	NOT NULL	
ADDRESS	Varchar2(72)	Varchar(72)	Varchar(72)	NOT NULL	
CITY	Varchar2(32)	Varchar(32)	Varchar(32)	NOT NULL	
STATE	Varchar2(2)	Varchar(2)	Varchar(2)	NOT NULL	
POSTAL_CODE	Varchar2(10)	Varchar(10)	Varchar(10)	NOT NULL	
HOME_PHONE	Varchar2(30)	Varchar(30)	Varchar(30)	NOT NULL	

Tgt\_Customer\_x

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
CUSTOMER_ID	Number(28)	Decimal(28)	Decimal(28)	PK/NOT NULL	ORDERS
COMPANY	Varchar2(50)	Varchar(50)	Varchar(50)		
FIRST_NAME	Varchar2(30)	Varchar(30)	Varchar(30)	NOT NULL	
LAST_NAME	Varchar2(30)	Varchar(30)	Varchar(30)	NOT NULL	
ADDRESS1	Varchar2(72)	Varchar(72)	Varchar(72)	NOT NULL	
ADRESS2	Varchar2(72)	Varchar(72)	Varchar(72)		
CITY	Varchar2(30)	Varchar(30)	Varchar(30)	NOT NULL	
STATE	Varchar2(2)	Varchar(2)	Varchar(2)	NOT NULL	
POSTAL_CODE	Varchar2(10)	Varchar(10)	Varchar(10)	NOT NULL	
PHONE	Varchar2(30)	Varchar(30)	Varchar(30)	NOT NULL	
EMAIL	Varchar2(30)	Varchar(30)	Varchar(30)		

Tgt\_CURRENTITEM\_x

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
ITEM_ID	Number(28)	Decimal(28)	Decimal(28)	PK/NOT NULL	
ITEM_NAME	Varchar2(72)	Varchar(72)	Varchar(72)	NOT NULL	
ITEM_DESC	Varchar2(40)	Varchar(40)	Varchar(40)		
PRICE	Number(10,2)	Money(19,4)	Decimal(10,2 )	NOT NULL	
WHOLESALE_COST	Number(10,2)	Money(19,4)	Decimal(10,2 )	NOT NULL	
DISCONTINUED_FLAG	Number(1)	Decimal(1)	Decimal(1)		
MANUFACTURER_ID	Number(28)	Decimal(28)	Decimal(28)		

DISTRIBUTOR_ID	Number(28)	Decimal(28)	Decimal(28)		
----------------	------------	-------------	-------------	--	--

## Tgt\_ItemMaster\_x

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
ITEM_MASTER_ID	Number(15)	Decimal(15)	Decimal(15)	PK/NOT NULL	
ITEM_ID	Number(28,0)	Decimal(28)	Decimal(28)	NOT NULL	
ITEM_NAME	Varchar2(72)	Varchar(72)	Varchar(72)	NOT NULL	
YEAR	Varchar2(4)	Varchar(4)	Varchar(4)		

## Tgt\_SalesByQtr\_x

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
ITEM_MASTER_ID	Number(15)	Decimal(15)	Decimal(15)	FK	Tgt_ItemMaster_x
QUARTER	Number(19,2)	Decimal(19,2)	Decimal(19,2)		
QUARTERLY_SALES	Number(19,2)	Decimal(19,2)	Decimal(19,2)		

## Tgt\_ItemSalesByYr\_x

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
ITEM_ID	Number(38,0)	Decimal(38)	Decimal(38)		
ITEM_NAME	Varchar2(72)	Varchar(72)	Varchar(72)		
YEAR	Varchar2(4)	Varchar(4)	Varchar(4)		
YEAR_SALES	Number(19,2)	Decimal(19,2)	Decimal(19,2)		

## Tgt\_OrderListing\_x

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
ORDER_ID	Number(15)	Decimal(15)	Decimal(15)	PK/NOT NULL	
DATE_ENTERED	Date	Datetime	Datetime	NOT NULL	
CUSTOMER_ID	Number(15)	Decimal(15)	Decimal(15)	NOT NULL	
ORDER_AMOUNT	Number(19,0)	Decimal(19,0)	Decimal(19,0)	NOT NULL	

## Tgt\_KAUAIFRANCHISE\_x

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
ORDER_ID	Number(15)	Decimal(15)	Decimal(15)	PK/NOT NULL	
DATE_ENTERED	Date	Datetime	Datetime	NOT NULL	
CUSTOMER_ID	Number(15)	Decimal(15)	Decimal(15)	NOT NULL	
ORDER_AMOUNT	Number(19,0)	Decimal(19,0)	Decimal(19,0)	NOT NULL	

## Tgt\_MAUIFRANCHISE\_x

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
ORDER_ID	Number(15)	Decimal(15)	Decimal(15)	PK/NOT NULL	
DATE_ENTERED	Date	Datetime	Datetime	NOT NULL	
CUSTOMER_ID	Number(15)	Decimal(15)	Decimal(15)	NOT NULL	
ORDER_AMOUNT	Number(19,0)	Decimal(19,0)	Decimal(19,0)	NOT NULL	

Tgt\_OAHUFRANCHISE\_x

Field Name	Oracle	MS SQL Server	Sybase	Key Type / Constraint	Reference
ORDER_ID	Number(15)	Decimal(15)	Decimal(15)	PK/NOT NULL	
DATE_ENTERED	Date	Datetime	Datetime	NOT NULL	
CUSTOMER_ID	Number(15)	Decimal(15)	Decimal(15)	NOT NULL	
ORDER_AMOUNT	Number(19,0)	Decimal(19,0)	Decimal(19,0)	NOT NULL	

## Flat Files

### Nielson : Structure

NIELSEN		
Column Name	Datatype	Length, Scale
CUST_ID	Numeric	18
COMPANY_NAME	Text	50
ADDRESS1	Text	72
ADDRESS2	Text	72
CITY	Text	30
ST	Text	2
CODE	Text	10

### Orders : Structure

ORDERS		
Column Name	Datatype	Length, Scale
ORDER_NO	Numeric	6
LINE_NO	Numeric	8
ITEM_NO	Numeric	10
ITEM_NAME	Text	23
QTY	Numeric	8
PRICE	Numeric	14, 2

### Orders

Order no	Line No	Item no	Item Name	Qty	Price
100001	20000001	12345	Air Compressor	50	5000
100002	20000001	12345	Air Compressor	70	5000
100003	20000001	12345	Air Compressor	100	5000

100004	20000002	12346	Remotely Operated Via	55	6000
100005	20000002	12346	Remotely Operated Via	550	6000
100006	20000003	12347	Towable Video Camera	300	7000
100007	20000003	12347	Towable Video Camera	500	7000
100008	20000003	12347	Towable Video Camera	43000000	7000
100009	20000004	12348	Marine super vhs vi	4100	8000
100010	20000004	12348	Marine super vhs vi	1400	8000
100011	20000004	12348	Marine super vhs vi	3400	8000
100012	20000005	12349	Dive computer	9100	8880
100013	20000005	12349	Dive computer	9050	8880
100014	20000005	12349	Dive computer	900	8880
100015	20000006	12340	Under Water Dive veh	2300	7500
100016	20000006	12340	Under Water Dive veh	1300	7500
100017	20000007	12344	Stabilizing vest	3800	6000
100018	20000007	12344	Stabilizing vest	7100	6000
100019	20000008	12343	Under water Metal Det	7600	7500
100020	20000008	12343	Under water Metal Det	6300	7500

### Products Structure

PRODUCTS		
Column Name	Datatype	Length, Scale
ITEM_NO	Numeric	10
ITEM_NAME	Text	23
CAT	Text	5
CUST_PRICE	Numeric	9, 2
VENDOR_PRICE	Numeric	9, 2
PRODUCT_CATEGORY	Text	21

### Products

Item_No	Item_Name	Cat	Cust_Price	Vendor_Price	Product_Category
12340	Under Water Dive	cat6	7000	7500	Vehicle
12343	Under water Metal	cat1	5000	5500	Misc Equipment
12344	Stabilizing vest	cat1	5000	5500	Misc Equipment
12345	Air Compressor	cat1	5000	5500	Misc Equipment

12346	Remotely Operated	cat2	2000	6500	Photo Equipment
12347	Towable Video	cat3	6000	7500	Photo Equipment
12348	Marine super vhs vi	cat4	3000	8500	Photo Equipment
12349	Dive computer	cat5	8000	9000	Small Instruments

### Customers

CompanyName	FirstName	LastName	Address1	Address2	City
Alfreds Futterkiste	Maria	Anders	Obere Str.		Berlin
Ana Trujillo	Ana	Trujillo	Avda. de la		México D.F.
Antonio Moreno	Antonio	Moreno	Mataderos		México D.F.
Around the Horn	Thomas	Hardy	120 Hanover		London
Berglunds snabbköp	Christina	Berglund	Berguvsväge		Lulea
Blauer See	Hanna	Moos	Forsterstr.		Mannheim
Blondel père et fils	Frederique	Citeaux	24, place		Strasbourg
Bólido Comidas	Martin	Sommer	C/ Araquil,		Madrid
Bon app'	Laurence	Lebihan	12, rue des		Marseille
Bottom-Dollar	Elizabeth	Lincoln	23		Tsawassen
B's Beverages	Victoria	Ashworth	Fauntleroy		London

Region	PostalCode	Phone	Email
DE	12209	030-0074321	Maria.Anders@Aol.com
TO	05021	(5) 555-4729	
TO	05023	(5) 555-3932	
UK	WA1 1DP	(171) 555-7788	
SE	S-958 22	0921-12 34 65	
US	68306	0621-08460	
FR	67000	88.60.15.31	
ES	28023	(91) 555 22 82	
FR	13008	91.24.45.40	
BC	T2F 8M4	(604) 555-4729	
UK	EC2 5NT	(171) 555-1212	