

Informatica Powercenter

Lab Book

Copyright © 2011 IGATE Corporation. All rights reserved. No part of this publication shall be reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior written permission of IGATE Corporation.

IGATE Corporation considers information included in this document to be Confidential and Proprietary.

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..... | 18 |
| Lab 2-3 Create a Mapping | 25 |
| Lab 3-1 Creating Workflow | 40 |
| Lab 3-2 Start and Monitor Workflows..... | 51 |
| Lab 4-1 Sales Summary | 59 |
| Lab 5-1 New Customer | 66 |
| Lab 6-1 Flat File Join | 74 |
| Lab 7-1 Create a Mapplet..... | 80 |
| Lab 7-2 Quarterly-Sales Mapping | 85 |
| Lab 7-3 Annual Sales Mapping..... | 95 |
| Lab 8-1 Updating Current Items | 99 |
| Lab 9-1 Listing Order Details | 103 |
| Lab 10-1 Router Transformation | 107 |
| Lab 11-1 Sequence Generator Transformation | 113 |
| Lab 12-1 Stored Procedure Transformation | 118 |
| Lab 13-1 Configure an Email Task | 124 |
| Lab 14-1 Configure a Command Task | 128 |
| Lab 15-1 Event-Wait task..... | 130 |
| Lab 16-1 Pre-SQL and Post-SQL | 131 |
| Lab 17-1 Multiple Source Files(Indirect File) | 132 |
| Lab 18-1 Mapping Parameter | 133 |
| Lab 19-1 Mapping Variable | 134 |
| Lab 20-1 FTP Connection | 135 |
| Lab 21-1 Dynamic Lookup | 136 |
| Lab 22-1 Using Mapping Wizard for SCD Type1 | 138 |
| Lab 23-1 Using Mapping Wizard for SCD Type2..... | 144 |
| Appendix A – Sources and Targets used in the Mappings | 148 |

Lab 1-1 Create a Folder

Goals

- Learn to navigate through PowerCenter Repository Manager
- Connect to the repository
- Create a folder, subject to appropriate privileges

Time

15 Minutes


Lab Setup

- Informatica Client
- Login details for connecting to the repository

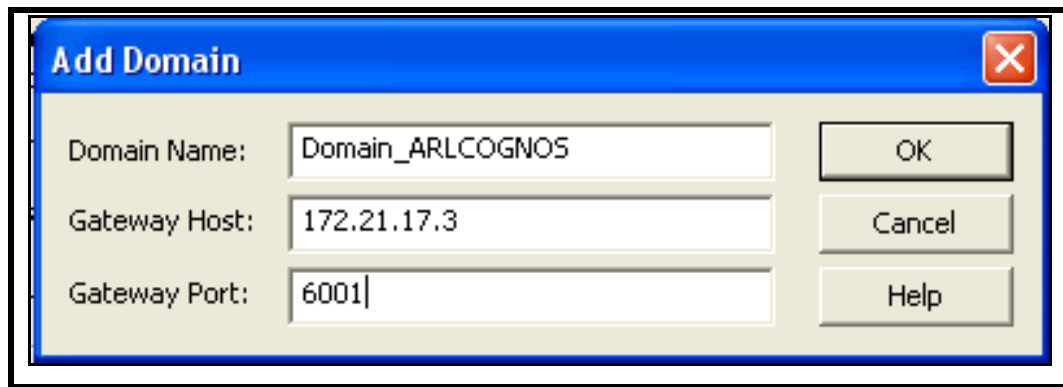
Start the Repository Manager

1. START | PROGRAMS | INFORMatica POWERCENTER | CLIENT | POWERCENTER REPOSITORY MANAGER.

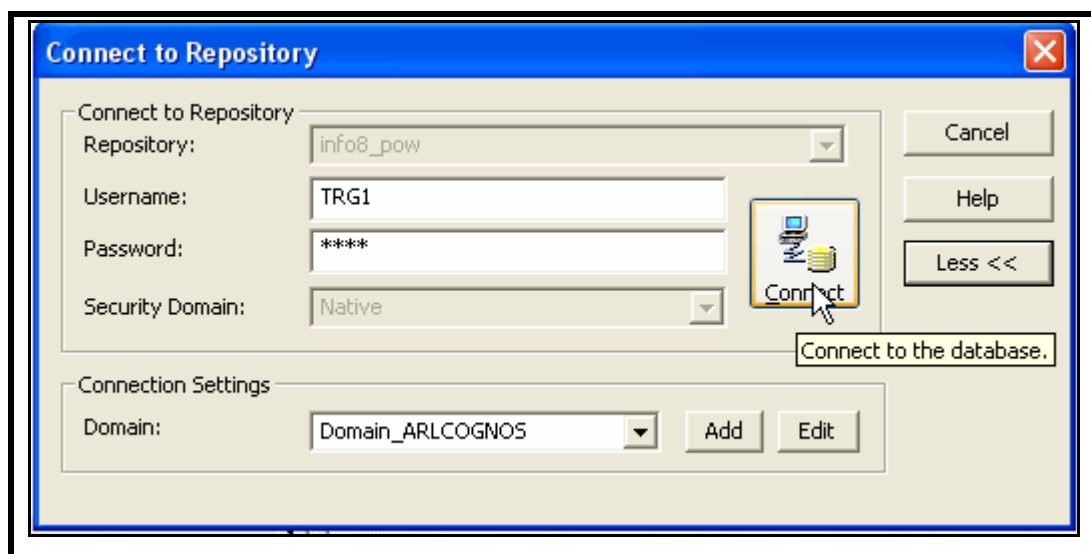
Connect to the repository

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.



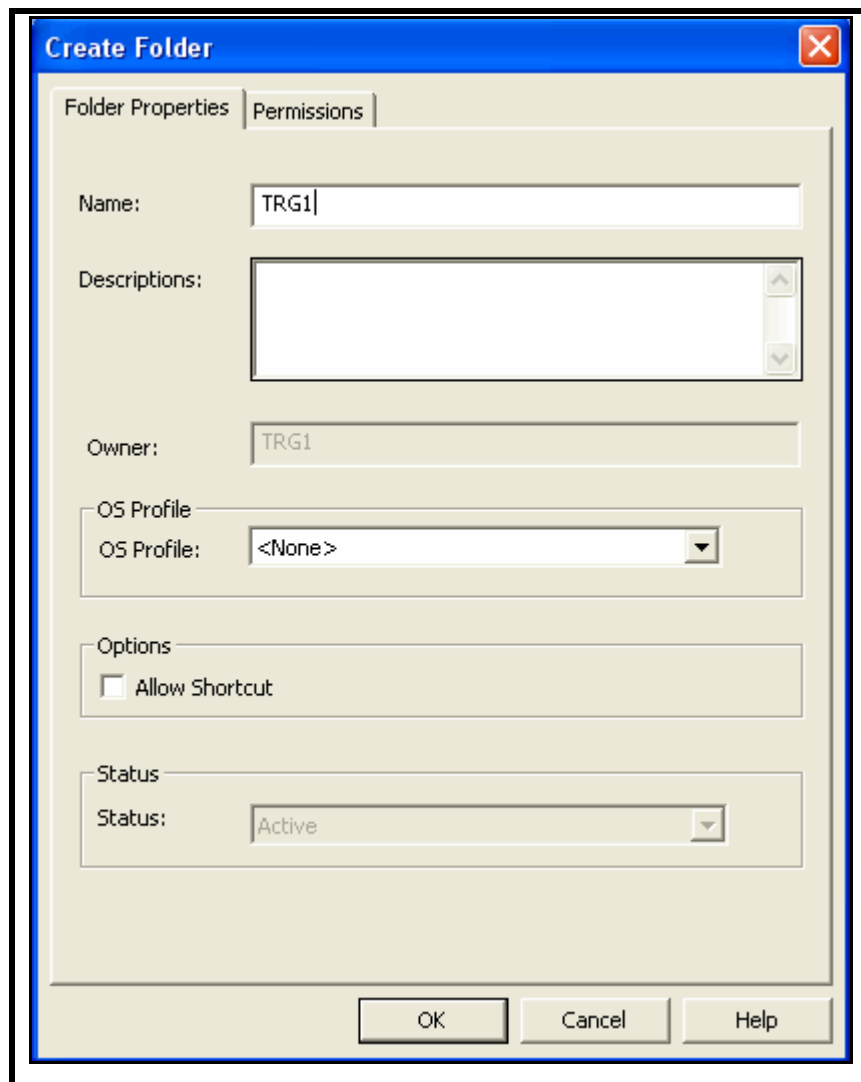
4. Click on the Connect button to connect to the repository.



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)
 - Descriptions: Comments about the folder
 - Click on the permission tab and select TRGx from the **user name**
Select TRGx as owner
3. Click OK.



The image shows a 'Create Folder' dialog box with a blue title bar and a close button. It has two tabs: 'Folder Properties' (selected) and 'Permissions'. The 'Folder Properties' tab contains several fields: 'Name' with the text 'TRG1', 'Descriptions' with an empty text area, 'Owner' with the text 'TRG1', 'OS Profile' with a dropdown menu showing '<None>', 'Options' with an unchecked checkbox for 'Allow Shortcut', and 'Status' with a dropdown menu showing 'Active'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Create Folder

Folder Properties | Permissions

Name: TRG1

Descriptions:

Owner: TRG1

OS Profile

OS Profile: <None>

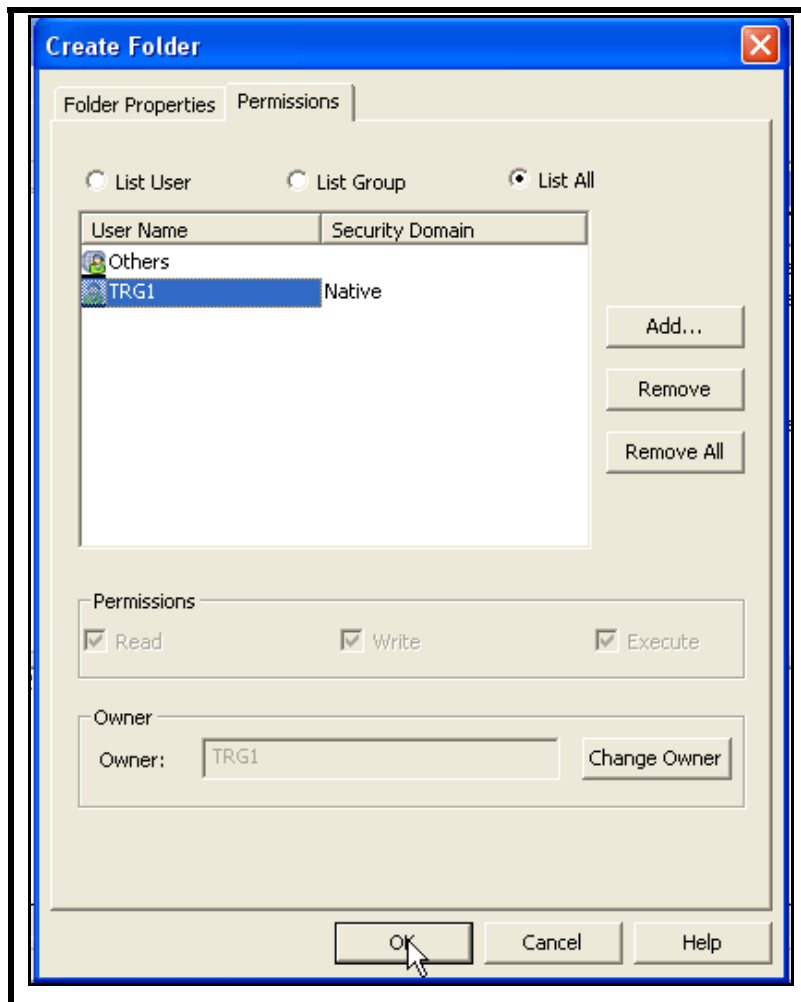
Options

☐ Allow Shortcut

Status

Status: Active

OK Cancel Help




Lab 2-1 Analyze Source Data

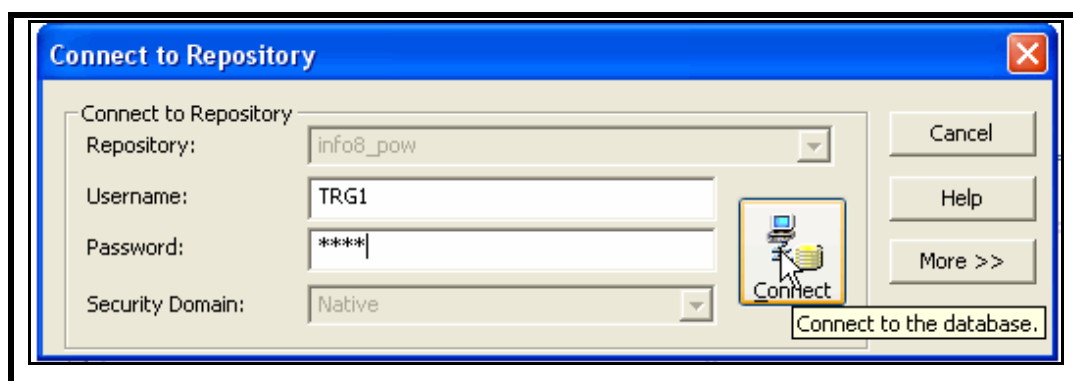
| | |
|------------------|---|
| Goals | <ul style="list-style-type: none"> Getting connected to PowerCenter Designer Understand the Source Analyzer tool Import source definitions into your folder Edit source definitions |
| Time | 15 Minutes |
| Lab Setup | Successful connection to the repository using PowerCenter Designer |

Getting connected to PowerCenter Designer

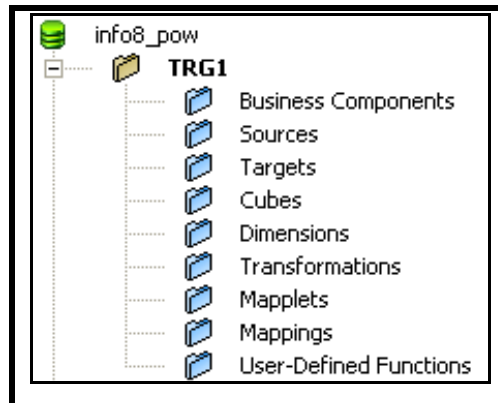
1. START PROGRAMS | INFORMATICA POWERCENTER CLIENT | DESIGNER.

You can connect to the repository in the following ways:

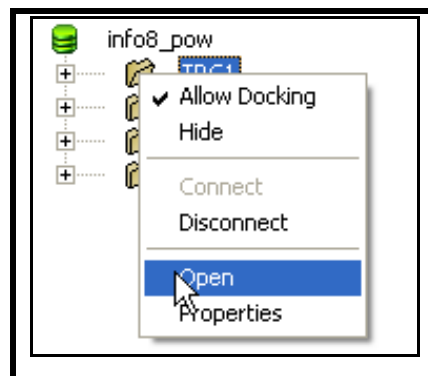
- i. Double click on the repository, or
- ii. Select the repository and select the menu option REPOSITORY | CONNECT, or
- iii. Select the repository and click on the  icon.



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.



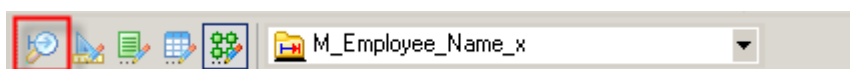
8. To open the folder, right click on the folder and select Open.



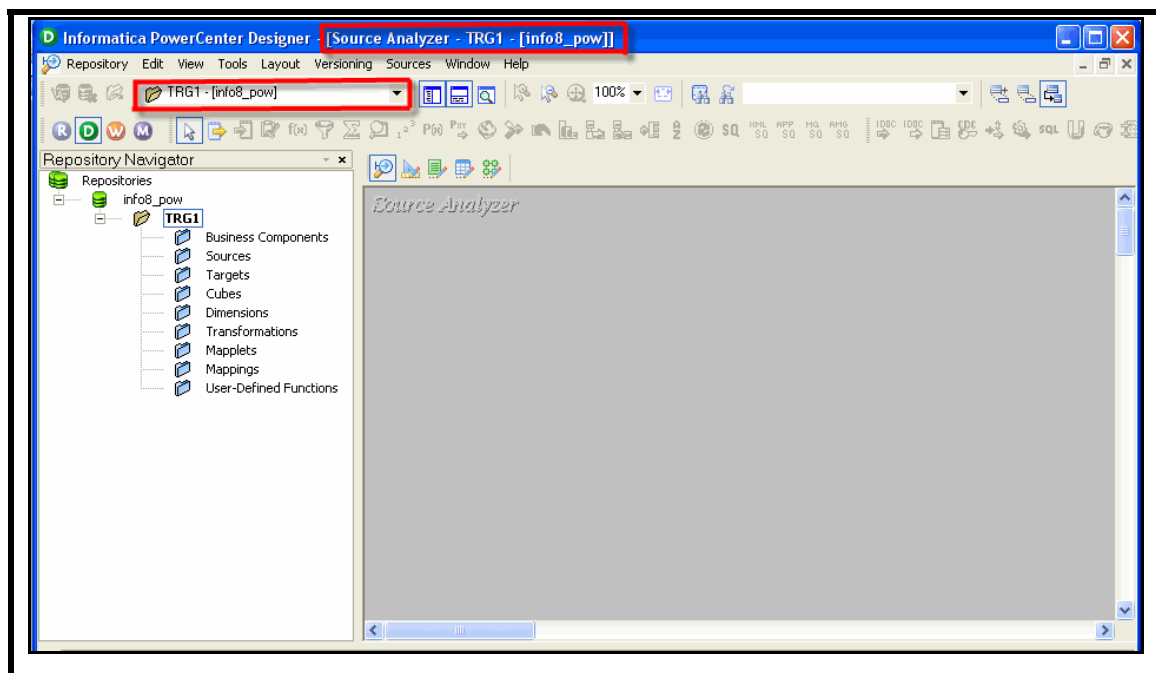
Note : An open folder is required in order to add, delete or modify objects. All of the work is performed in the Workspace Window, to the right of the Navigator Window (i.e. - Where the tools such as the Source Analyzer, Warehouse Designer, Mapplet Designer, etc., are active). The objects created in 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 :

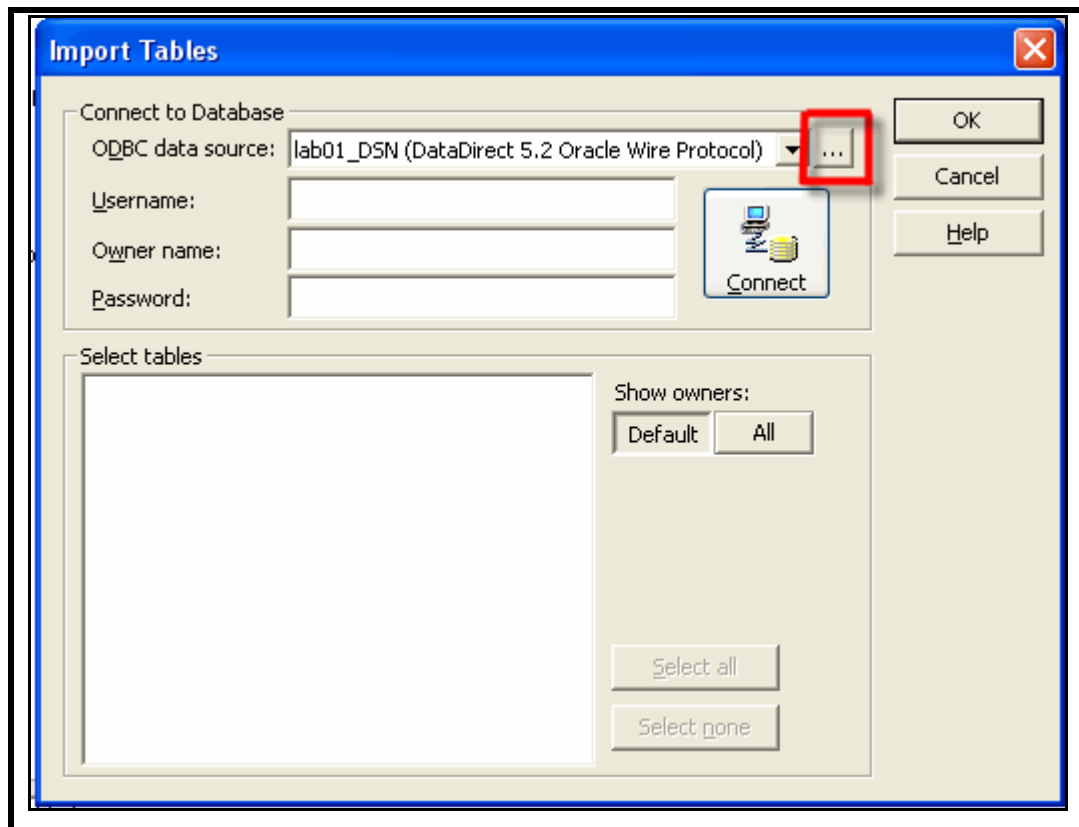



2. Notice the folder name and the repository names are displayed in the application title bar and Open Folders drop-down list.

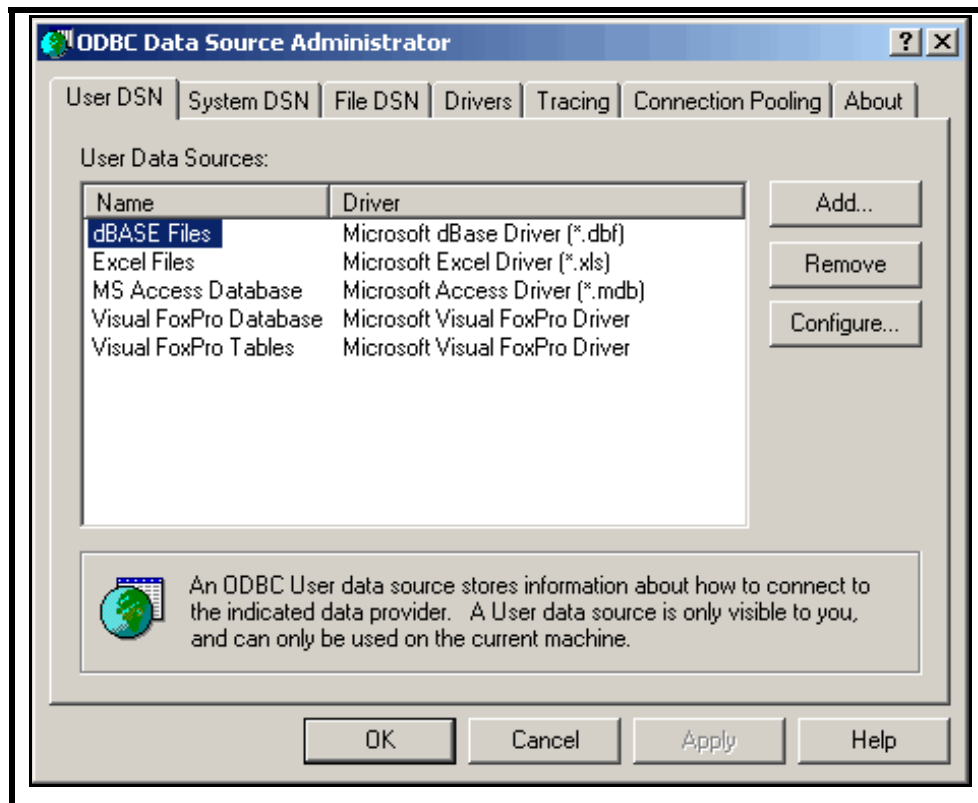


Importing Source from Database

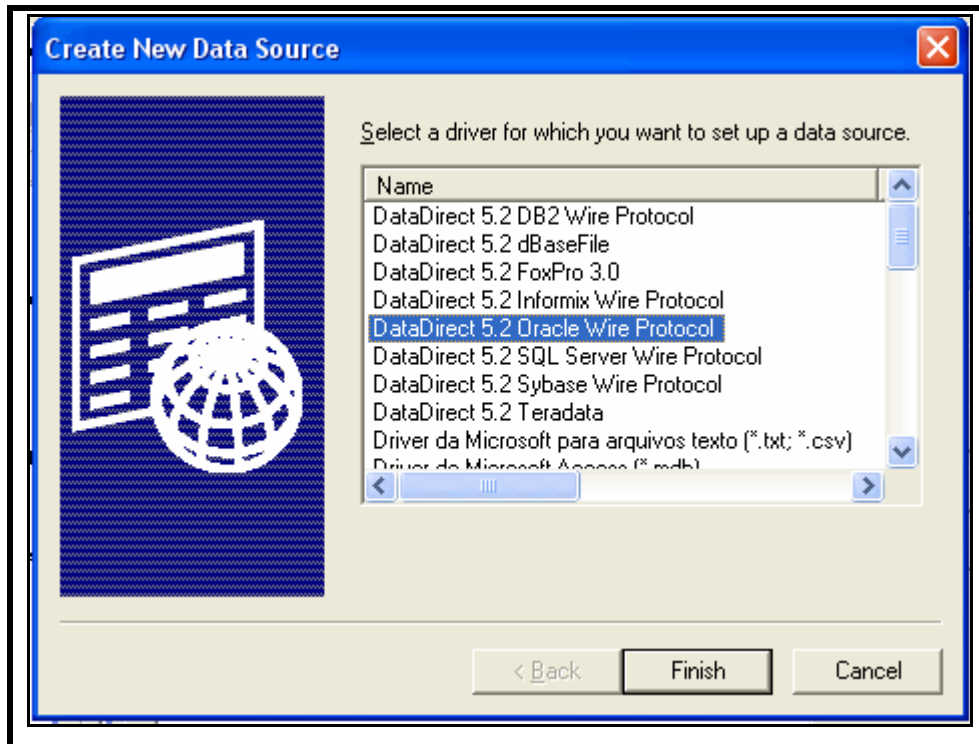
1. Select SOURCES | IMPORT FROM DATABASE.
 - i. The Import tables dialog box appears.



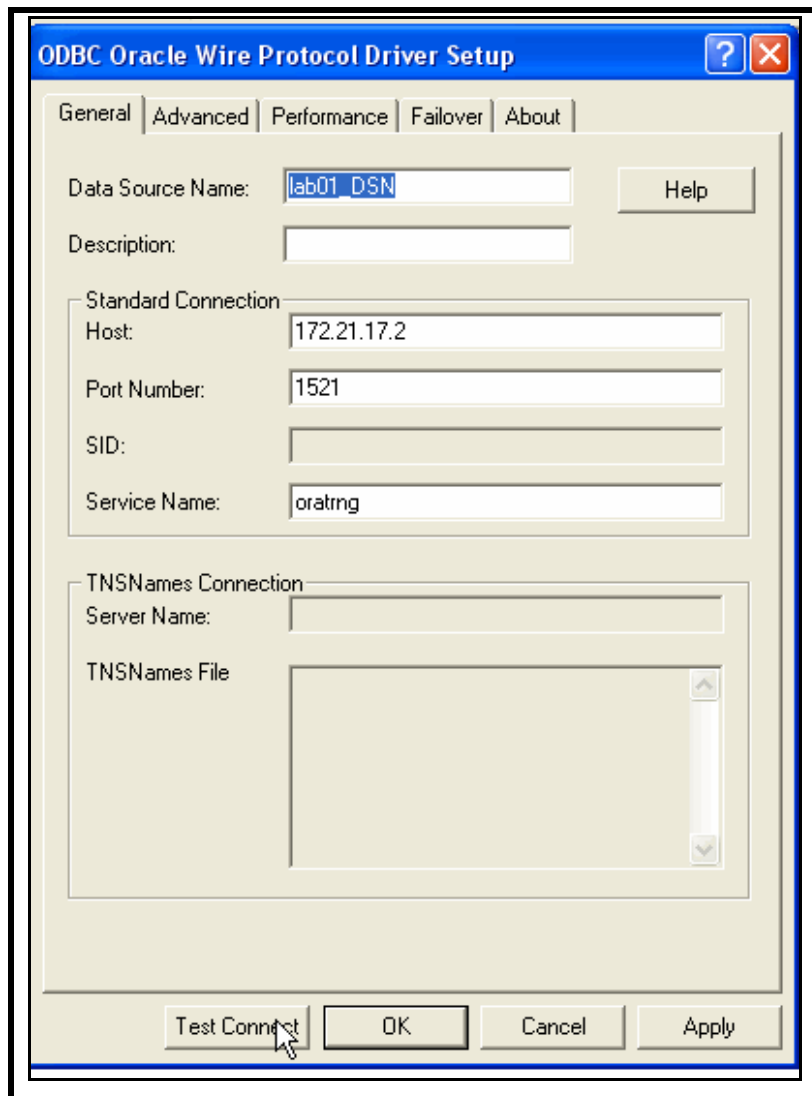
- ii. Click on the  icon, to create the data source.



- iii. Click on Add in the ODBC Data Source Administrator box

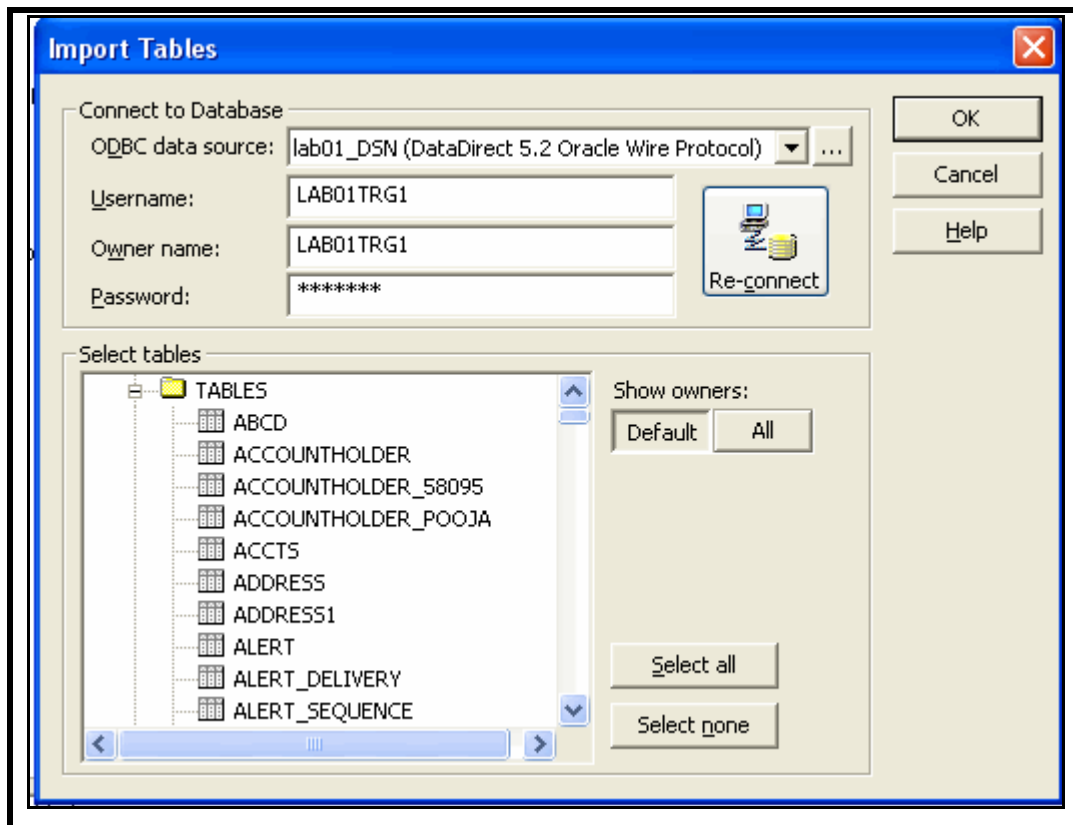


- iv. Select the DataDirect 5.2 Oracle Wire Protocol and enter the Data Source Name, Description and Server Name.



- 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

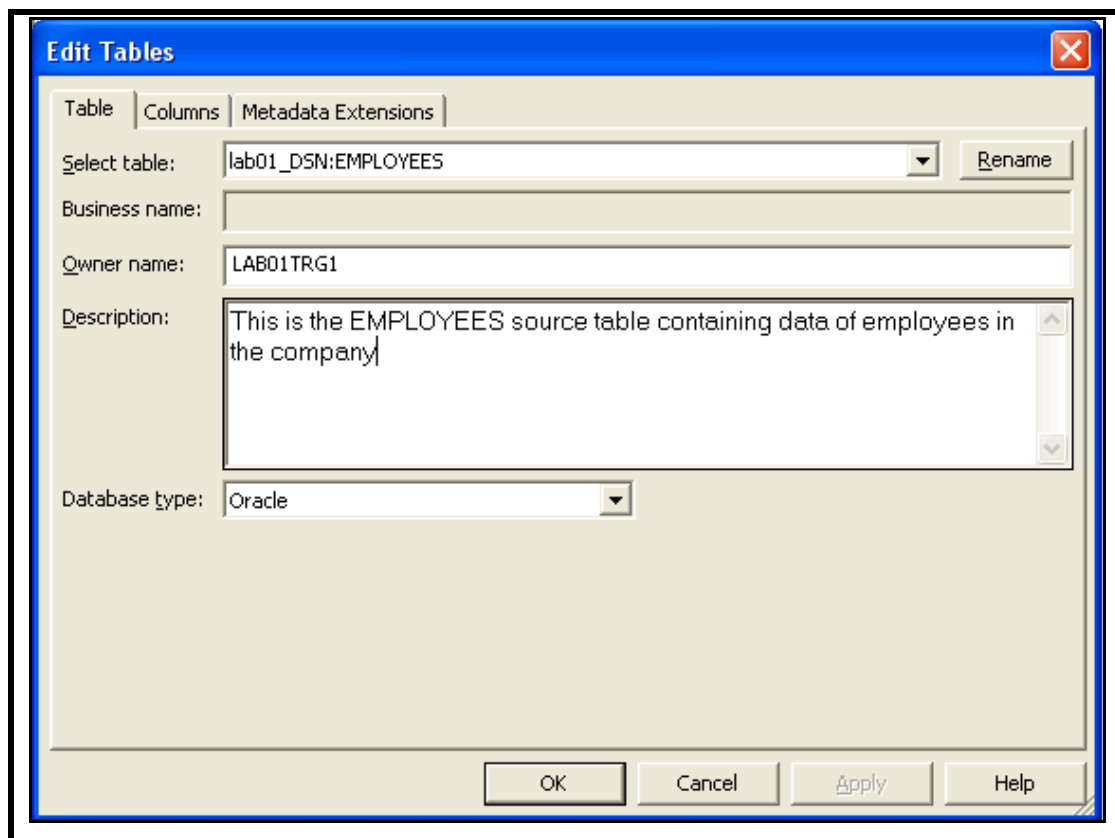
1. In the Source Analyzer workbook, for the source definition, expand the Key Types column.

| EMPLOYEES (Oracle) | | | |
|--------------------|-------------|-------------|------------------|
| 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 | LASTNAME | 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”

Edit Tables

Table Columns Metadata Extensions

Select table: lab01_DSN:EMPLOYEES

| | Column Name | Datatype | Prec | Scale | Not Null | Key Type |
|---|-------------|-------------|------|-------|-------------------------------------|-------------|
| 1 | EMPLOYEE_ID | number(p,s) | 2 | 0 | <input checked="" type="checkbox"/> | PRIMARY KEY |
| 2 | DEPTID | number(p,s) | 2 | 0 | <input type="checkbox"/> | FOREIGN KEY |
| 3 | LASTNAME | varchar2 | 30 | 0 | <input checked="" type="checkbox"/> | NOT A KEY |
| 4 | FIRSTNAME | varchar2 | 30 | 0 | <input checked="" type="checkbox"/> | NOT A KEY |
| 5 | ADDRESS | varchar2 | 80 | 0 | <input checked="" type="checkbox"/> | NOT A KEY |
| 6 | CITY | varchar2 | 40 | 0 | <input checked="" type="checkbox"/> | NOT A KEY |
| 7 | POSTAL_CODE | varchar2 | 10 | 0 | <input checked="" type="checkbox"/> | NOT A KEY |

Primary table: lab01_DSN:DEPARTMENT

Primary column: DEPTID

Description: This is the department id to which the employee belongs

OK Cancel Apply Help

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 2-2 Design a Target Schema

Goals

- Understand the TARGET Designer tool
- Create logical target definition
- Add new columns into the target
- Create physical targets

Time

10 Minutes

Lab Setup

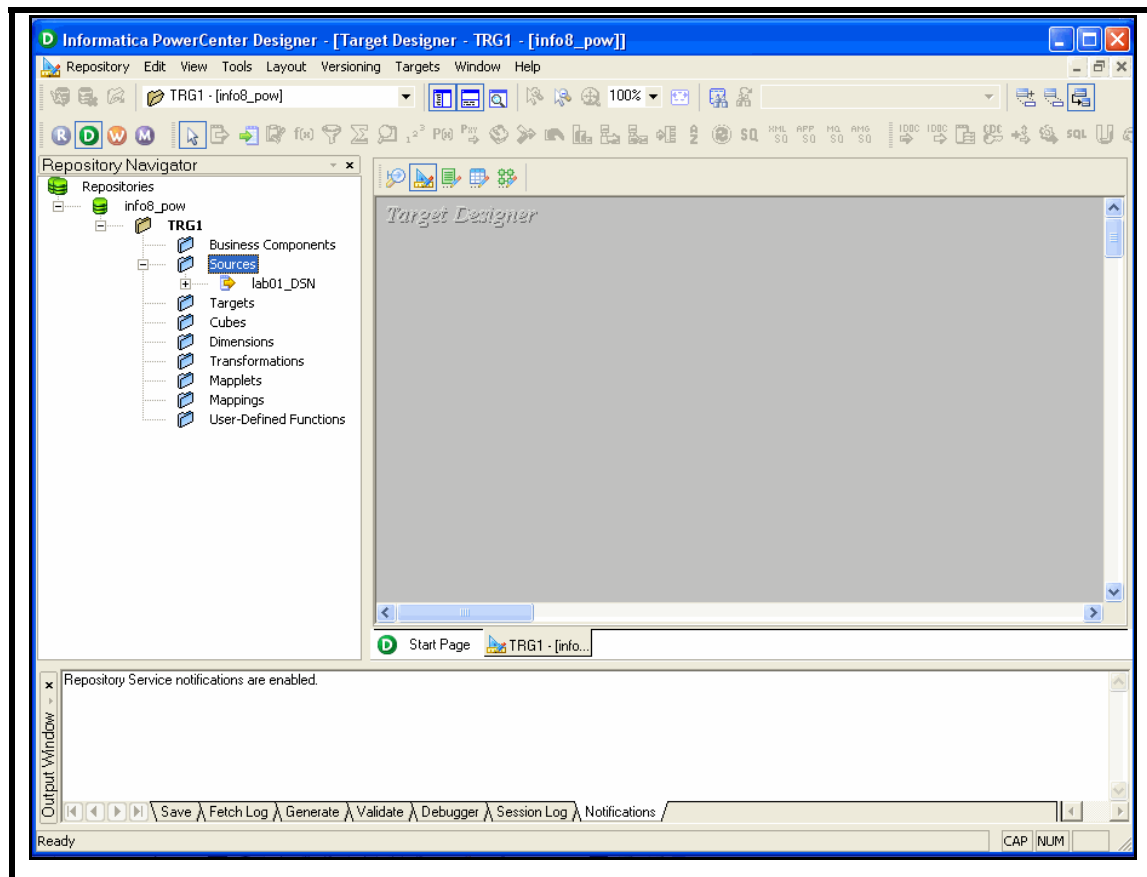
A successful connection to the repository using PowerCenter Designer

Understand TARGET Designer

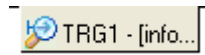
1. Select TOOLS | TARGET DESIGNER to open the TARGET Designer or click on the TARGET Designer button as shown below.



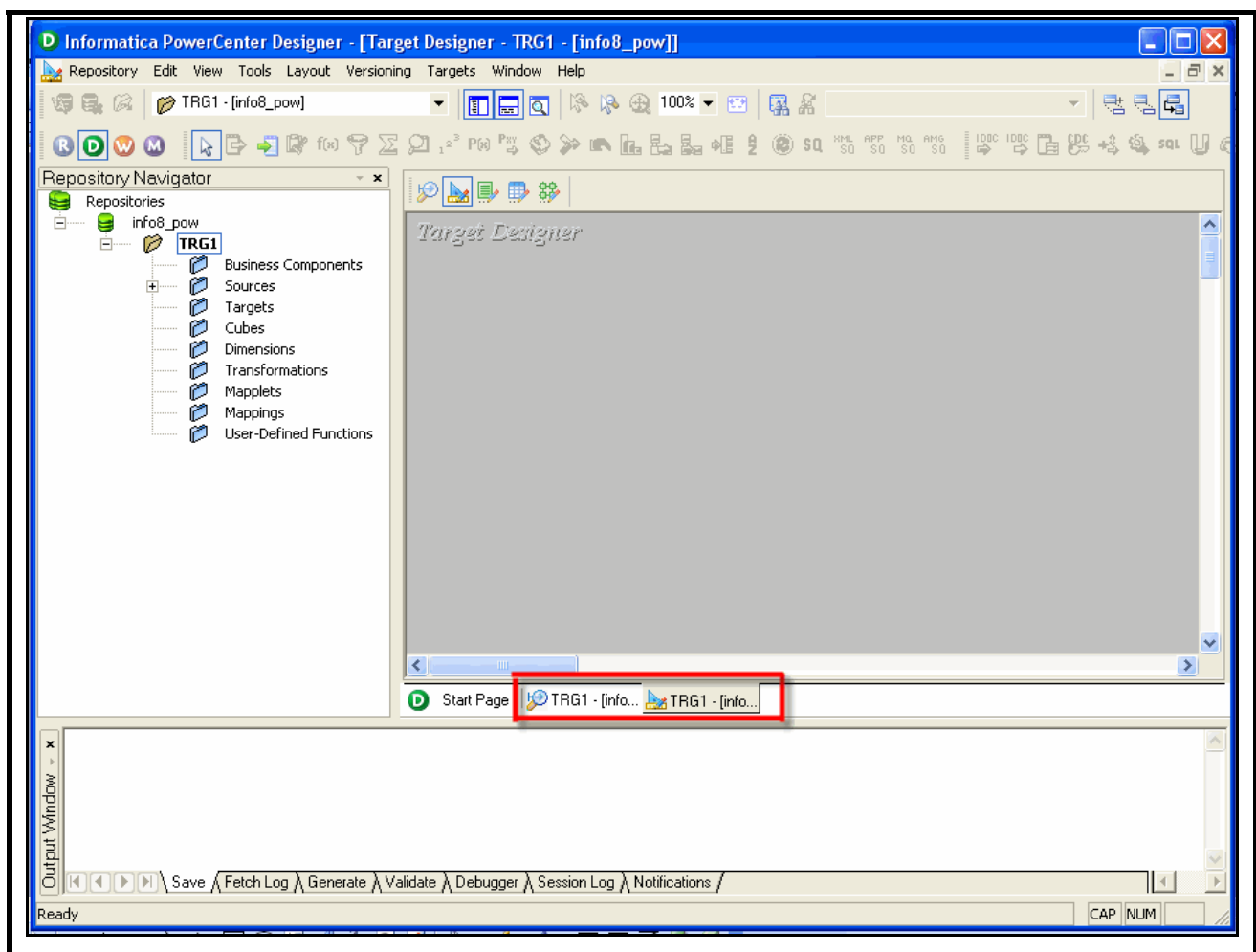
2. Make sure you are in the Workbook mode, by selecting VIEW | WORKBOOK. You should see the workbook tab.



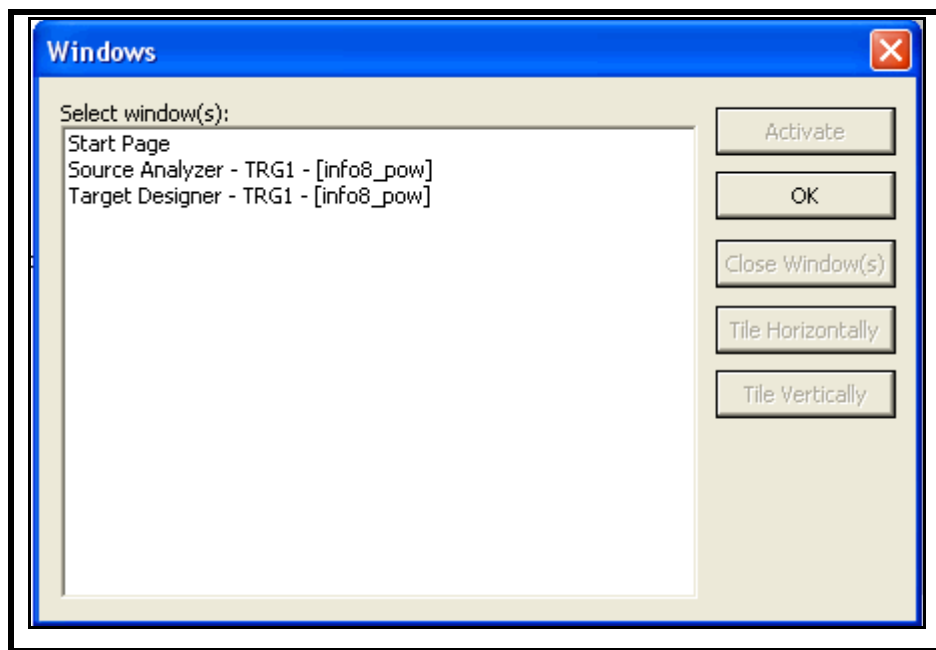
- Click on the Source Analyzer button. The tab now displays the Source Analyzer icon as shown below.



- Select WINDOW | NEW WINDOW. A new TARGET Designer Window is created.
- 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.

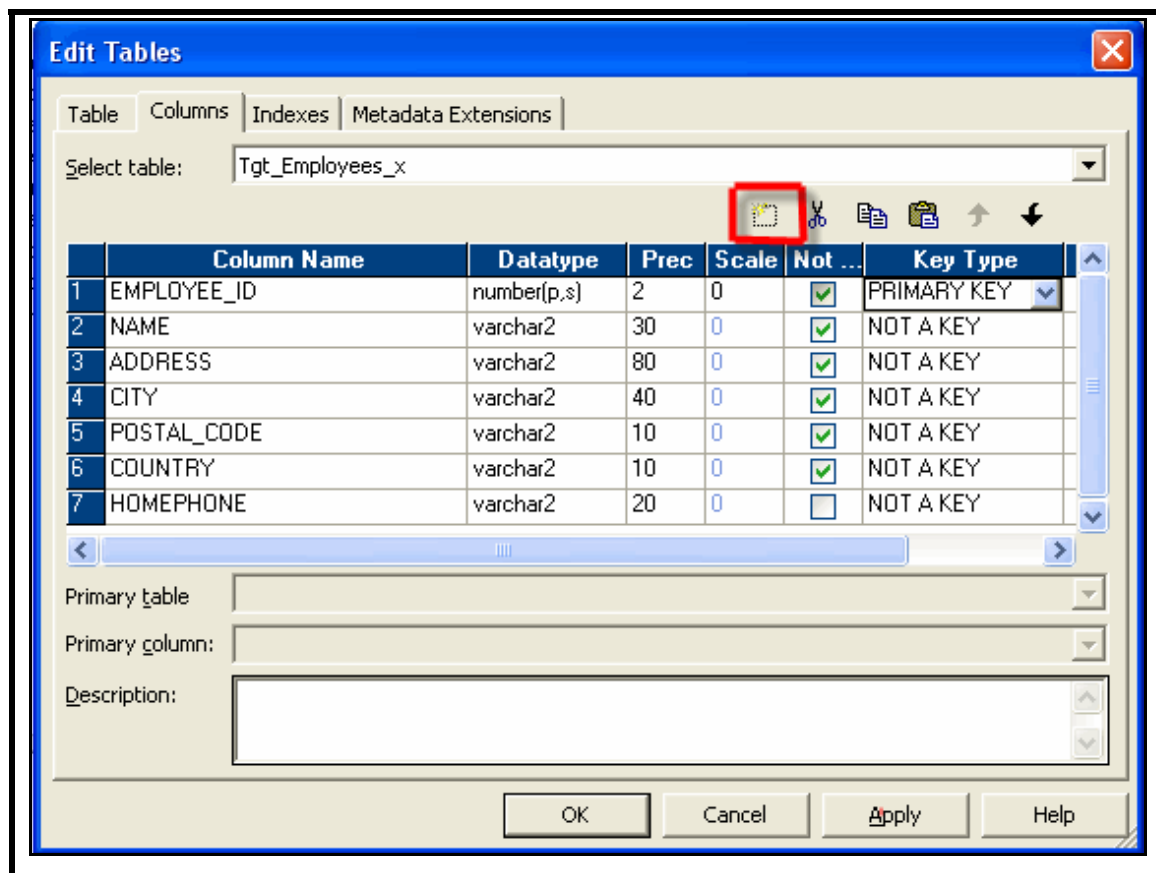


Design the target schema (Physical target not existing in database)

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.

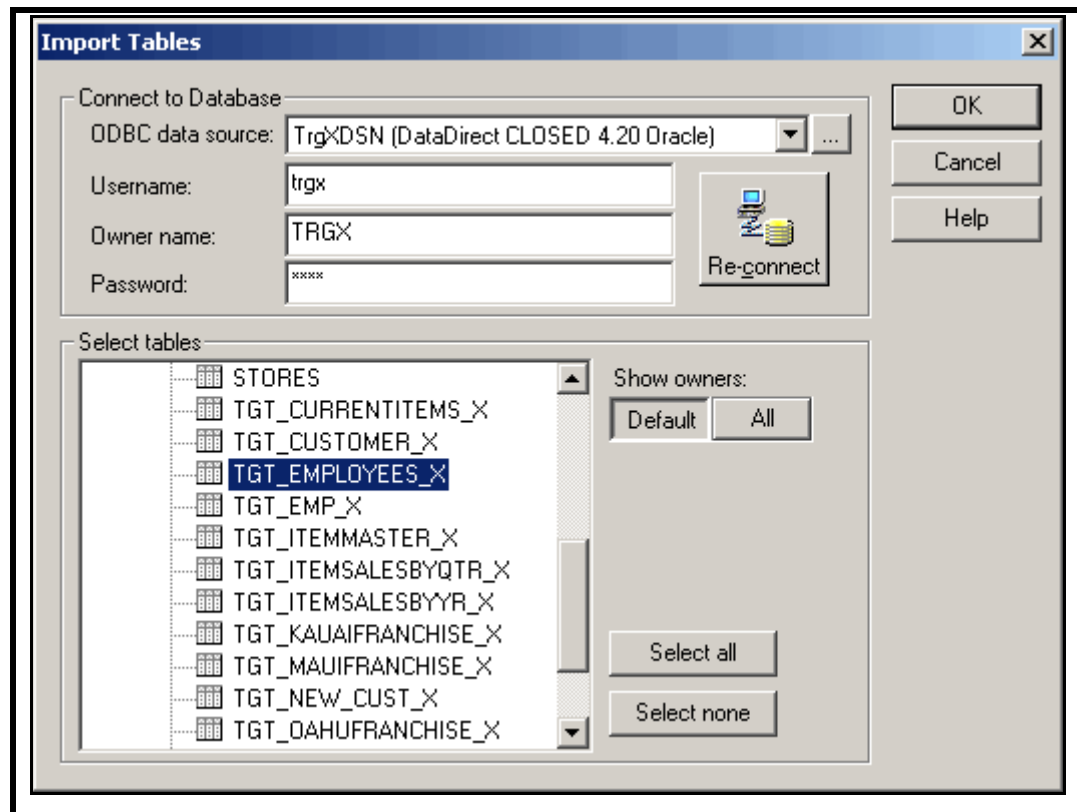


ii. Once the table has been defined, close the Edit Tables dialog box

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.



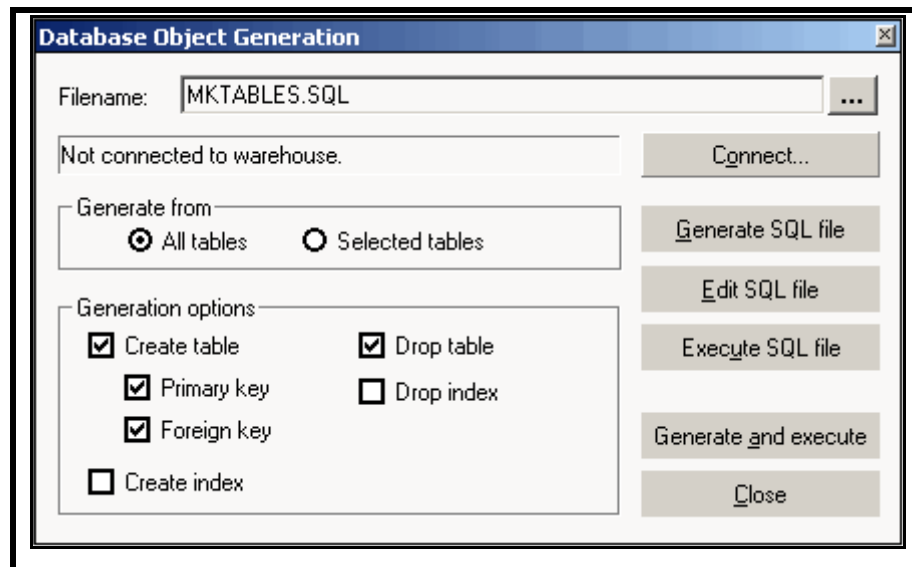
5. The new table definition appears in the Target Designer workbook.
6. The target schema definition is added to your folder's Targets section, or in the node in the Designer's Navigator Window.

| Tgt_Employees_x (Oracle) | | |
|--------------------------|-------------|-------------|
| K | 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 2-3 Create a Mapping

Goals

- Understand the Designer's Mapping Designer tool
- Create a mapping using:
 - A source definition
 - Source Qualifier and Expression Transformation
 - A target definition
- Edit transformations
- Add new ports in transformations
- Add formulas in the Expression transformation
- Validate a mapping
- Understand Designer's Output window

Time

30 Minutes

Lab Setup

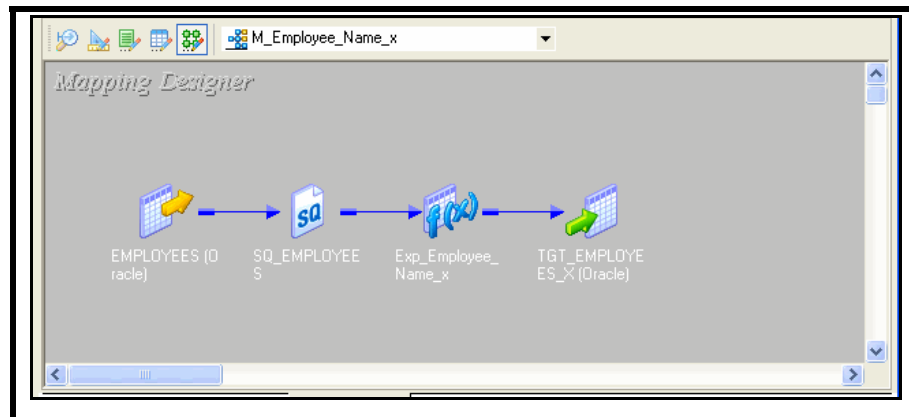
- A successful connection to the repository using PowerCenter Designer
- A source definition created
- A target definition created

Solution


- Create a relational target table that contains the **Name** as a concatenation of First Name and Last Name

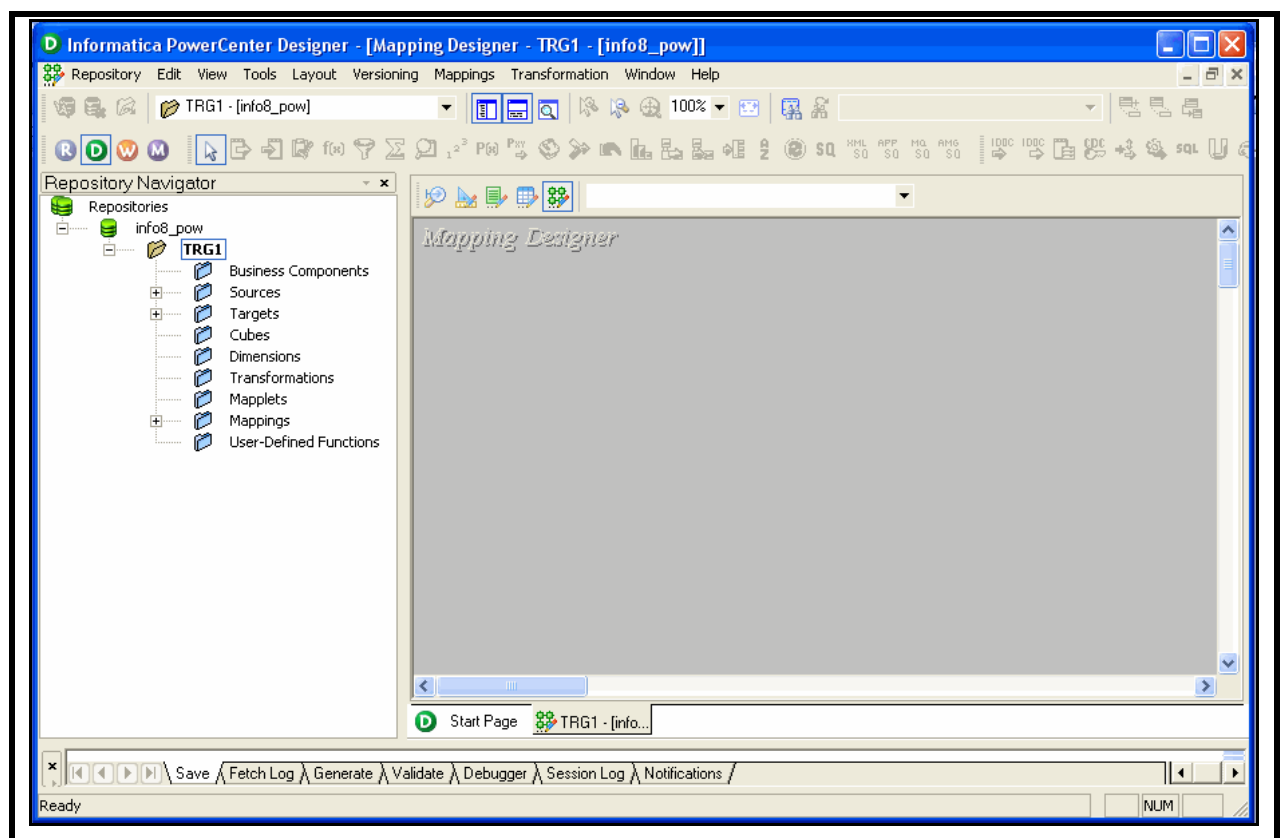
| TRANSFORMATION NAME | TYPE | DESCRIPTION |
|---------------------|------------------------------|--|
| EMPLOYEES | Relational Source Definition | Source Definition |
| SQ_EMPLOYEES_X | Source Qualifier | Data source qualifier for source table |
| EXP_EMPLOYEE_NAME_X | Expression | Link all ports from 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

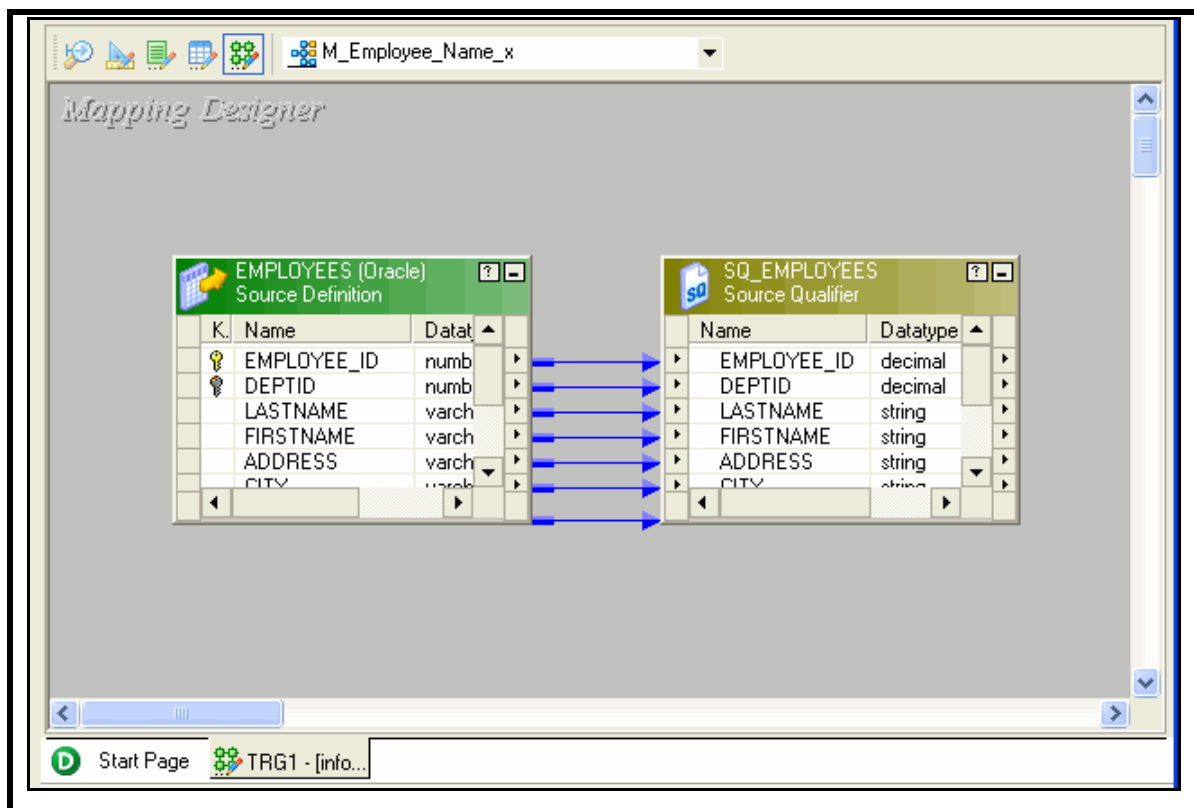


Source Qualifier Transformation

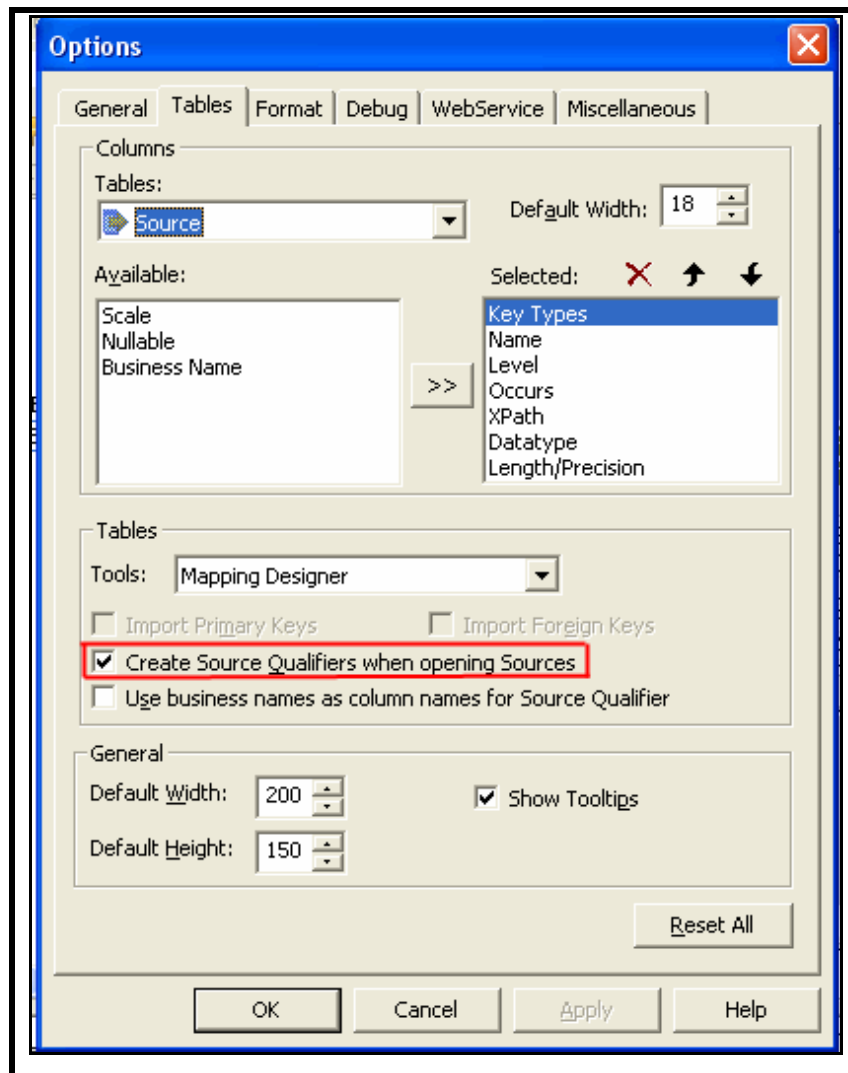
1. Select TOOLS | MAPPING DESIGNER, or click on the  icon. The workbook changes to the Mapping Designer.



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.

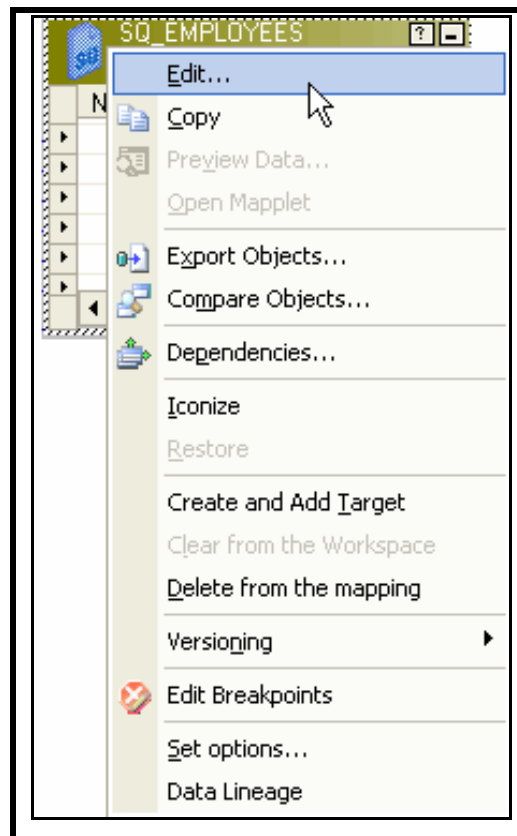


7. Drag the source again into the workspace. The Source Qualifier transformation automatically appears and is connected to the source. Delete the previously dragged source from the workspace by pressing the delete key.

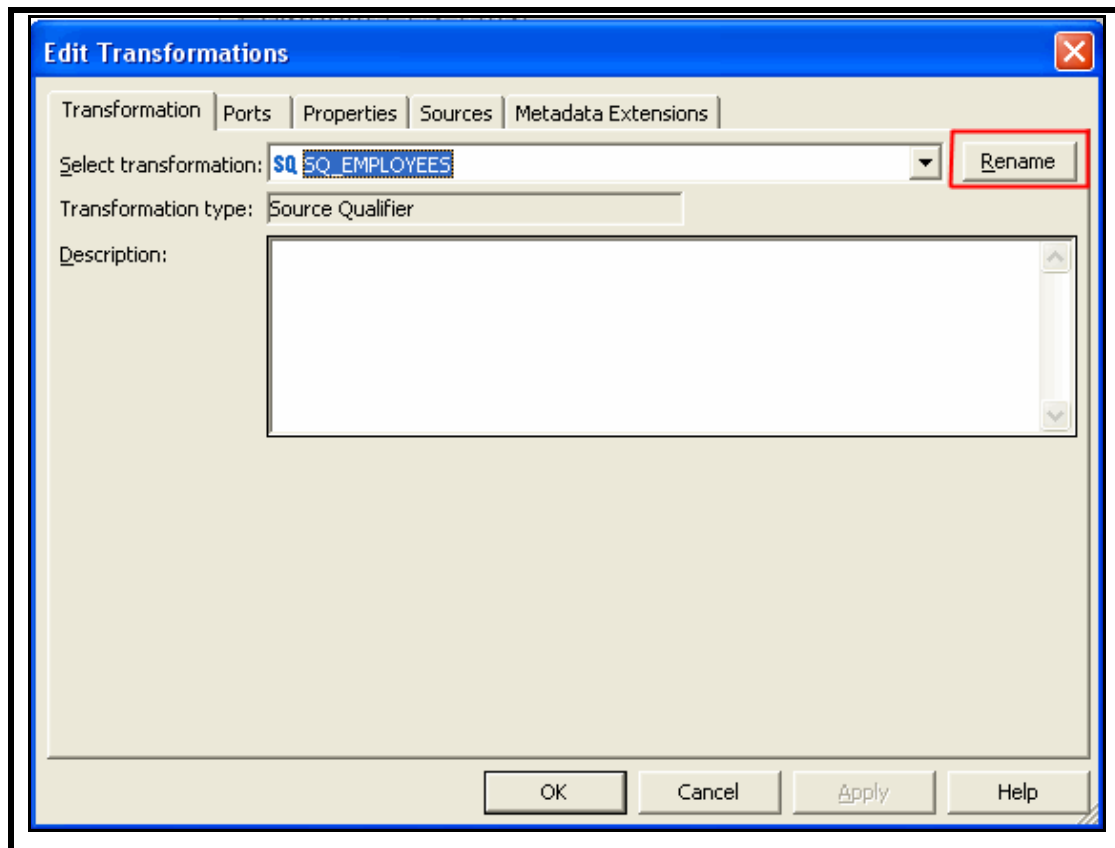
Note: Use the automatic Source Qualifier creation when you want to create one Source Qualifier for each source in your mapping and disable the automatic creation when you want to join data from different sources.

Edit the Source Qualifier Transformation

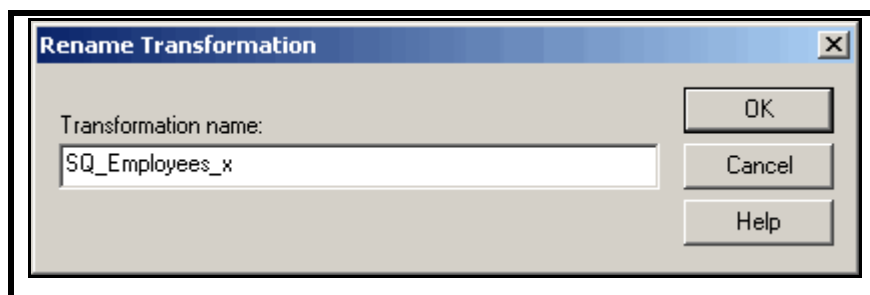
1. Use either of the methods to invoke the Edit Transformations box.
2. Right click on the Source Qualifier and select Edit...





- i. Double click on the Source Qualifier transformation.
- ii. Click on the Transformation tab and click on Rename to rename the transformation.



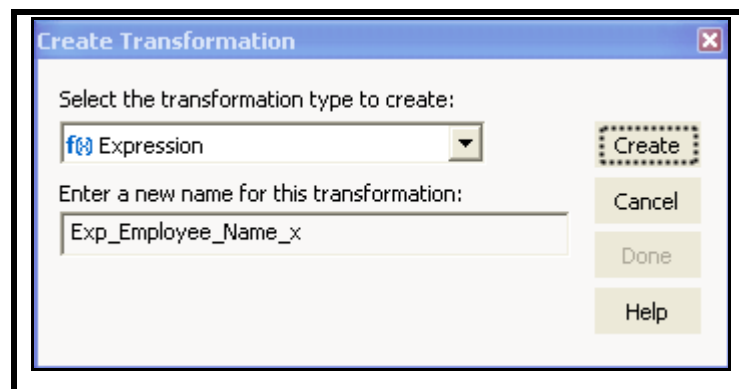
3. Enter the name and click on OK.



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 .

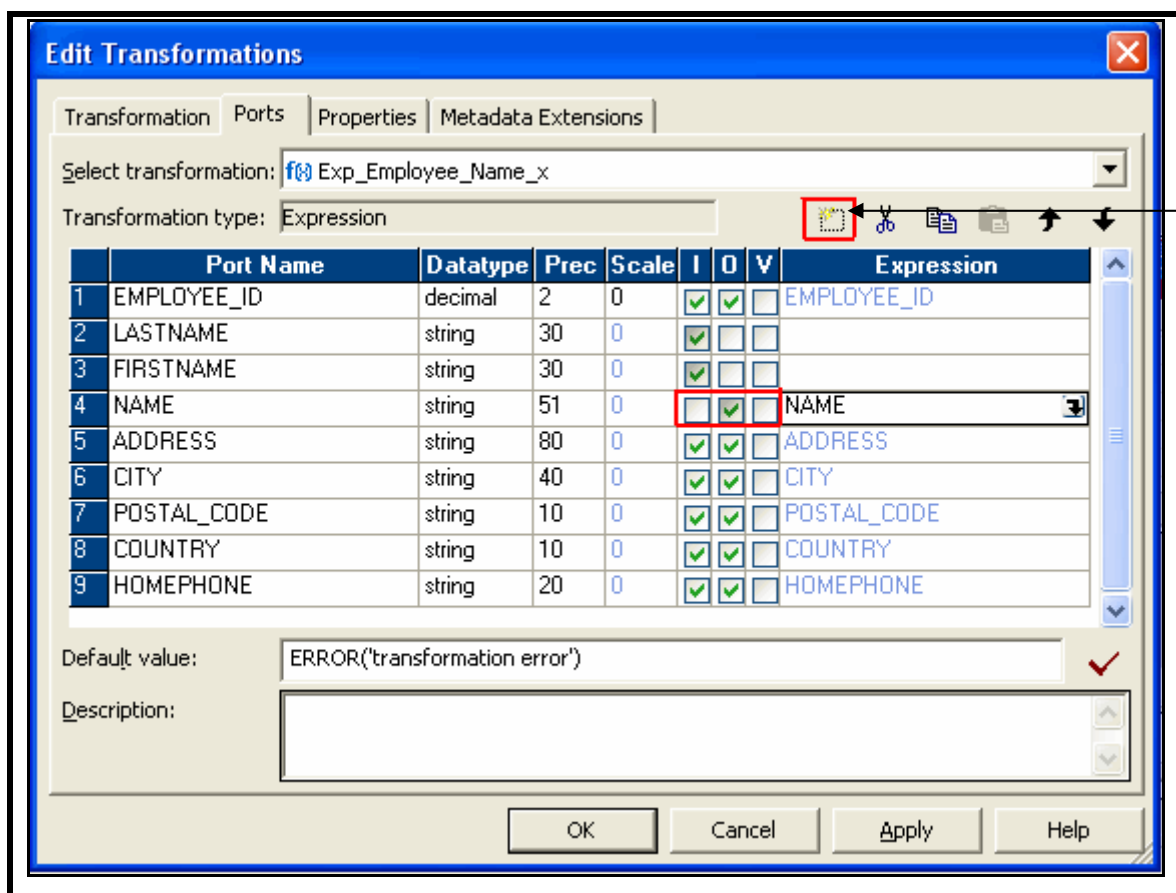
HINT : Hold down the Ctrl or Shift key and select the ports in the **SQ_Employees_x**. Drag them to an empty line on the expression transformation. When the mouse is released, not only will the port names (including data types, precision and scale) will be copied from **SQ_Employees_x**, but links connecting 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.

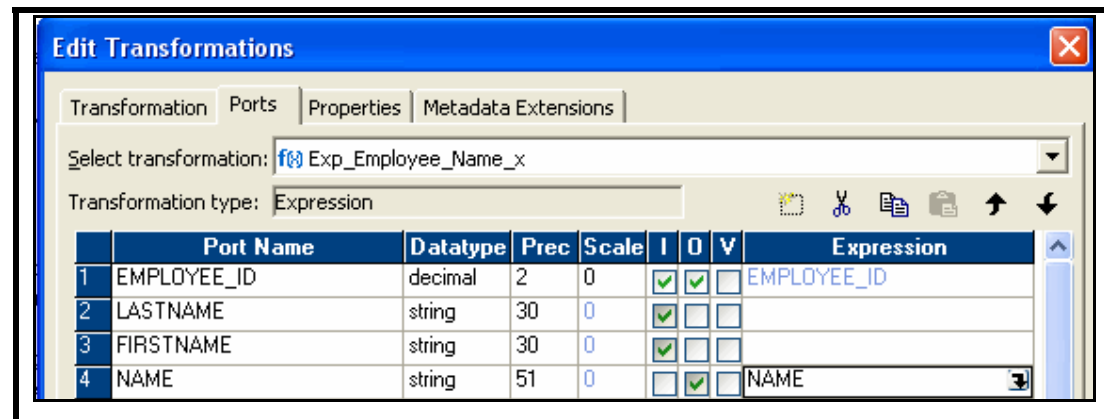
4. Click on the Ports tab.
5. Disable the output ports for FIRSTNAME and LASTNAME by removing the checkmark in the 'O' (output) column – this will define the port as input only.
6. Select the FIRSTNAME port and click on the  icon to add a new port and rename it to Name. This will cause the new port to 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:



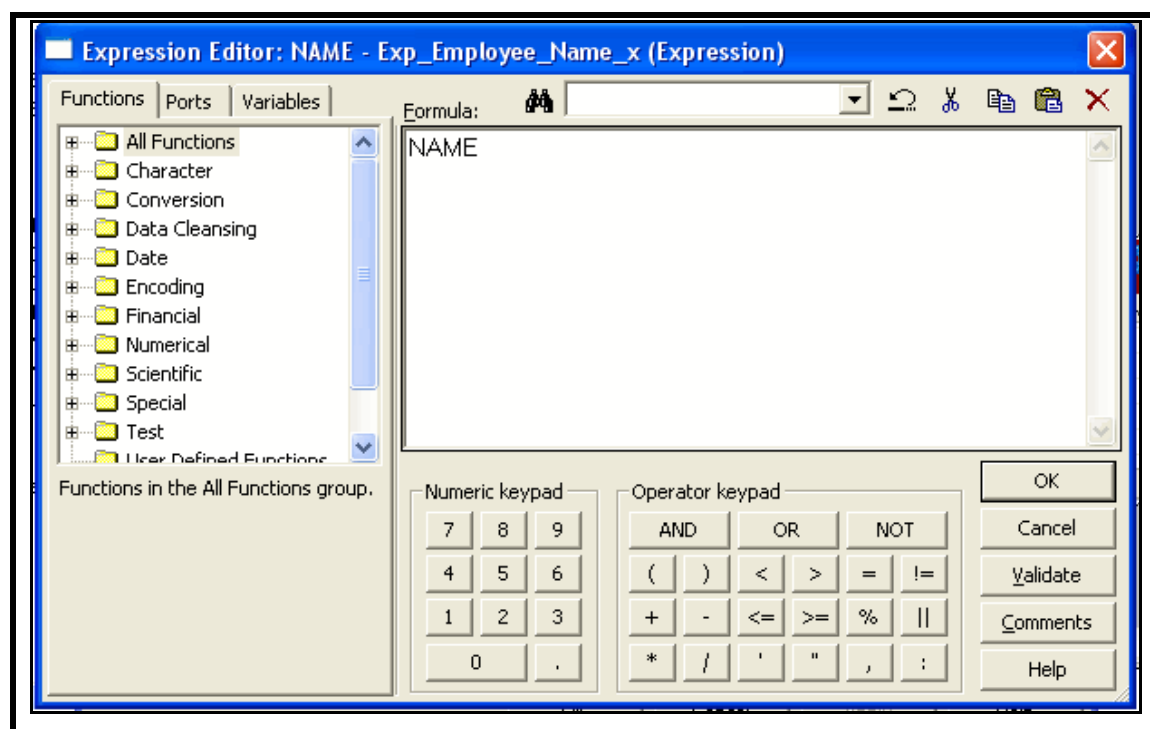
Create Expression Formula

1. Click the down arrow  in the expression column of the NAME port.

Add Port
Icon

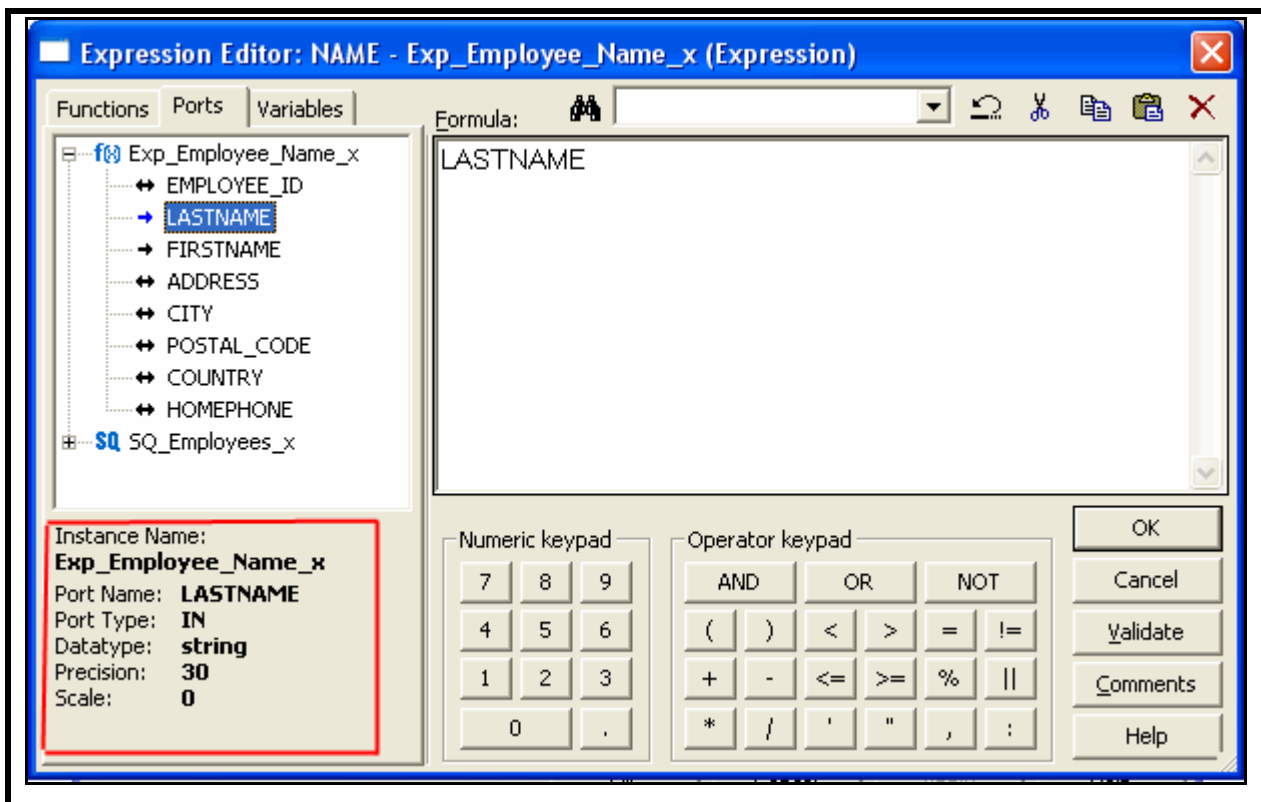


2. Delete the text NAME in the Formula field in the Expression Editor.

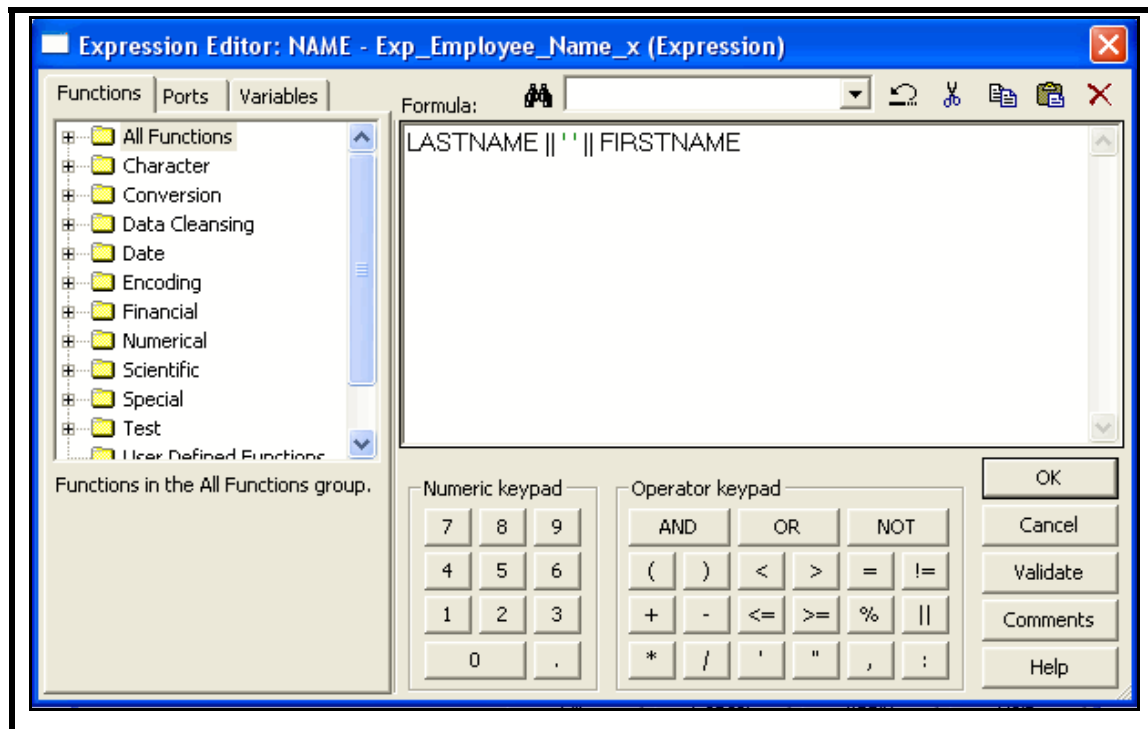


3. Select the Ports tab as shown in the figure below.

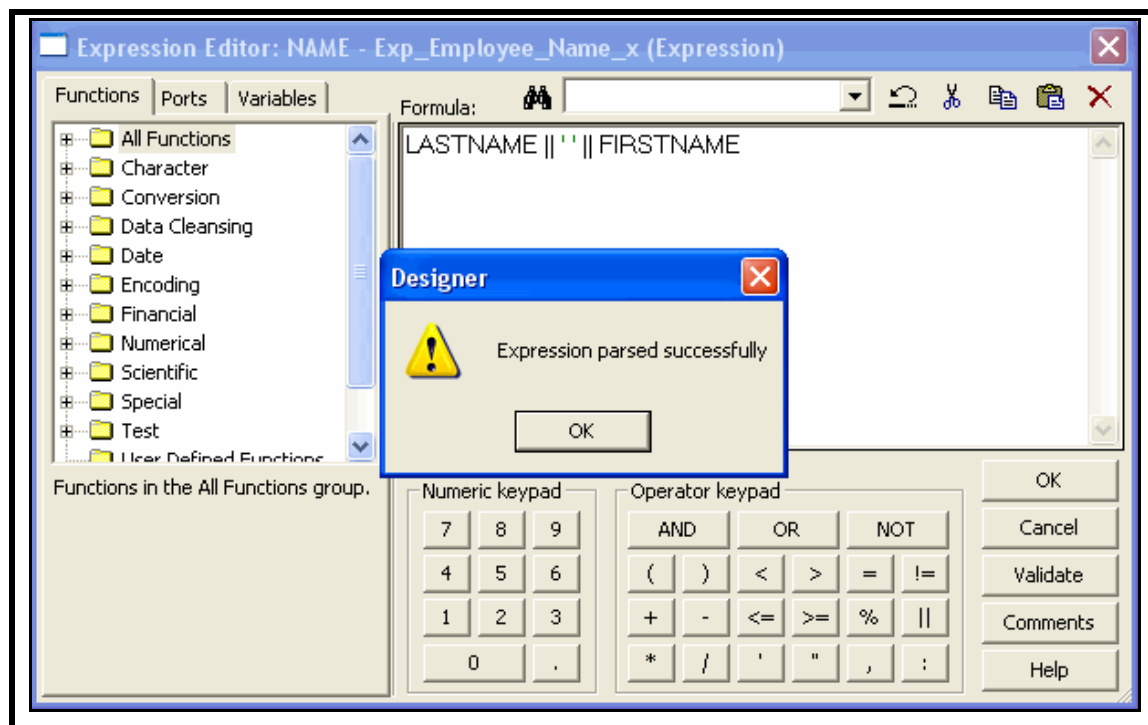
4. Double-click on the port LASTNAME and note its presence in the Formula field. Based on which port is selected, the port details appear under the 'Instance Name:' section of the ports tab.



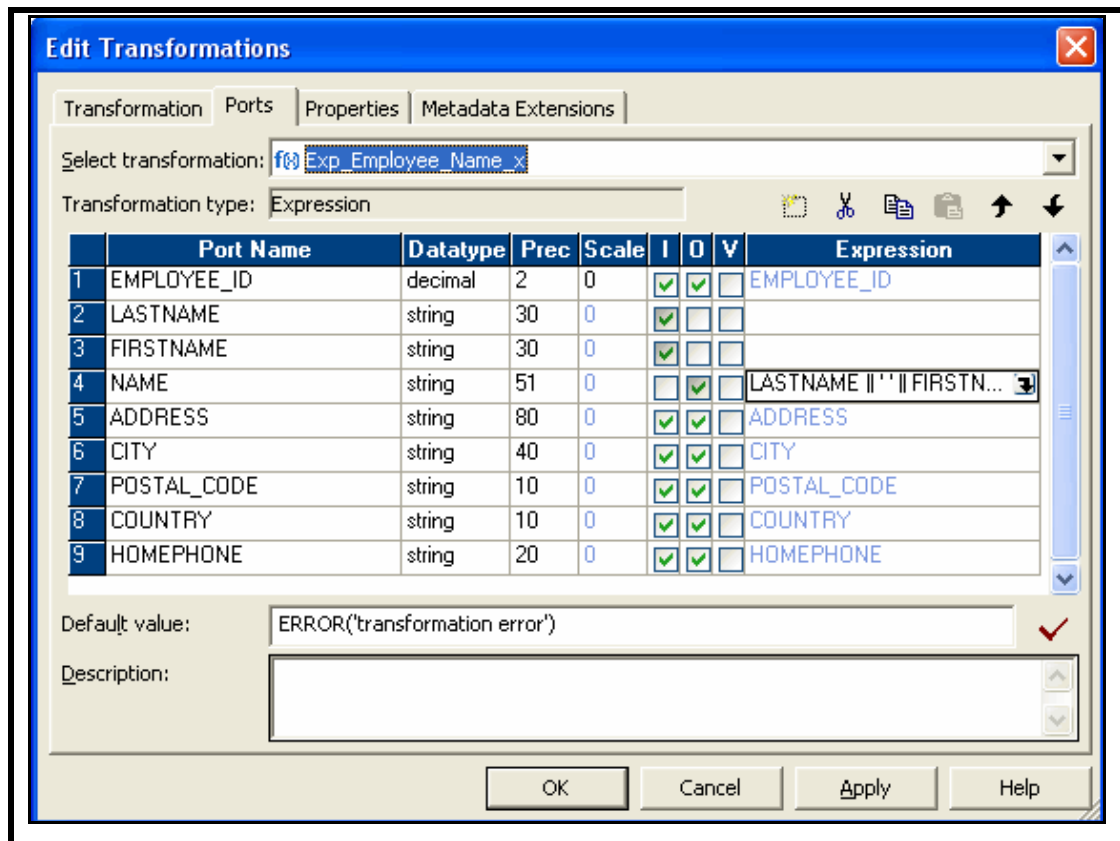
5. Create the expression as shown below:



- Click on Validate. This will parse the expression.

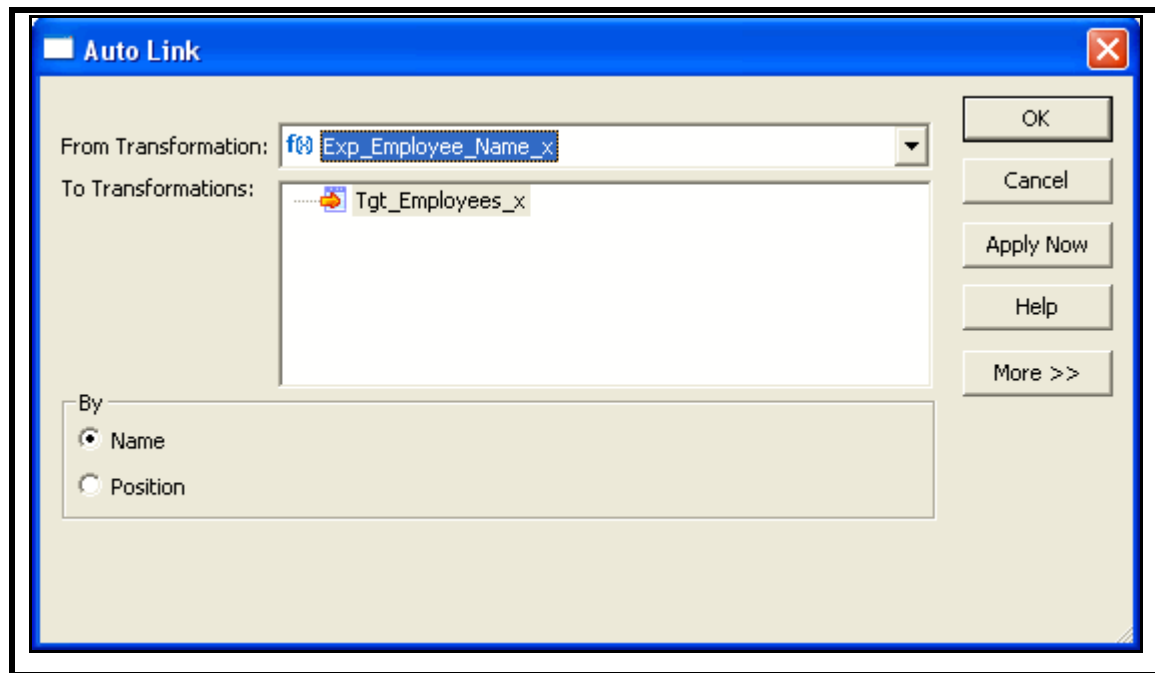


7. Click on OK if the expression is successfully parsed.
8. Click OK to close the Edit Transformation dialog box.
9. To save the changes to the repository, select REPOSITORY | SAVE or Ctrl-S.
10. The finished transformation will look like the following:



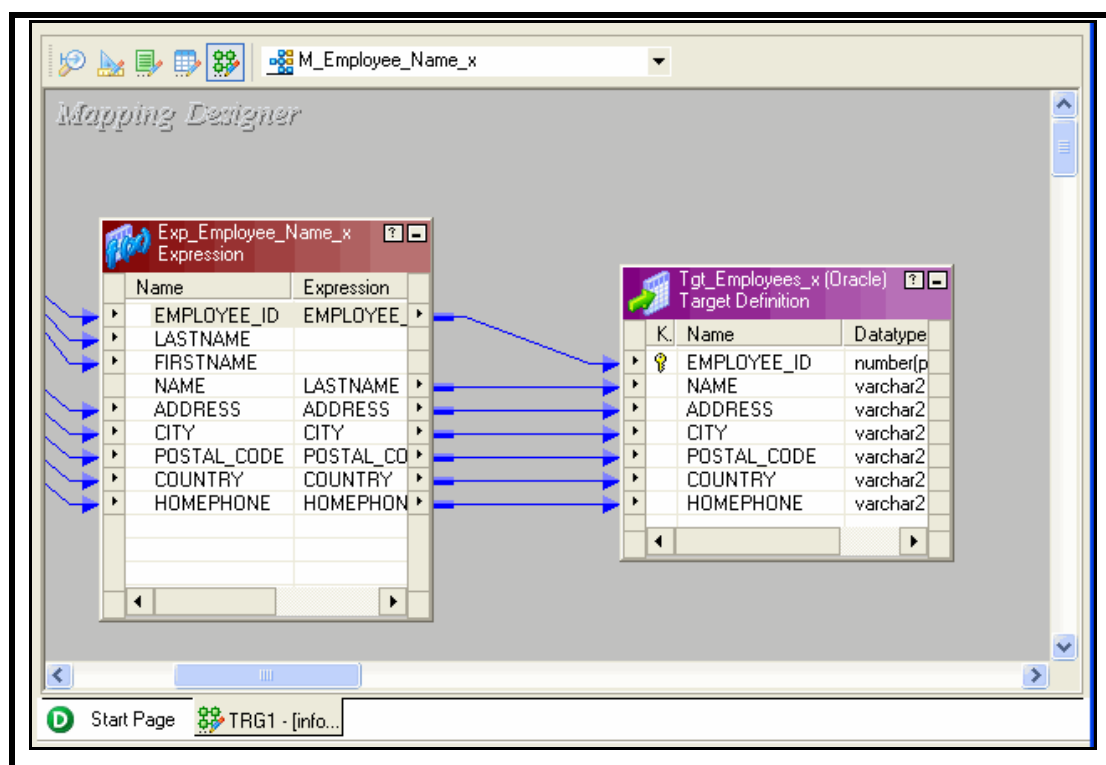
Link Expression Transformation and Target

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.

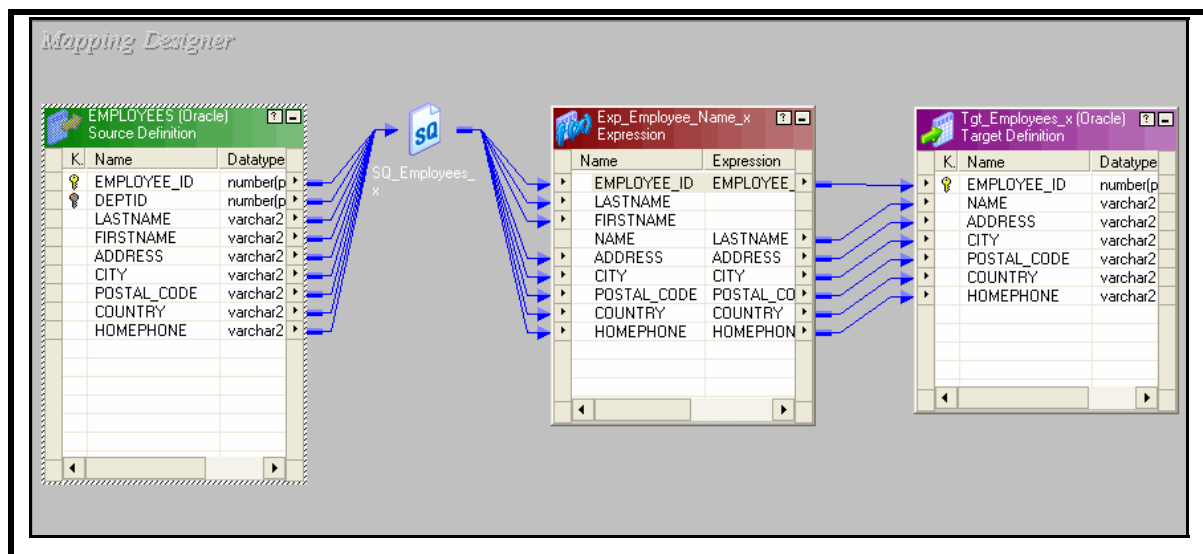


iii. Click on Apply Now and OK.

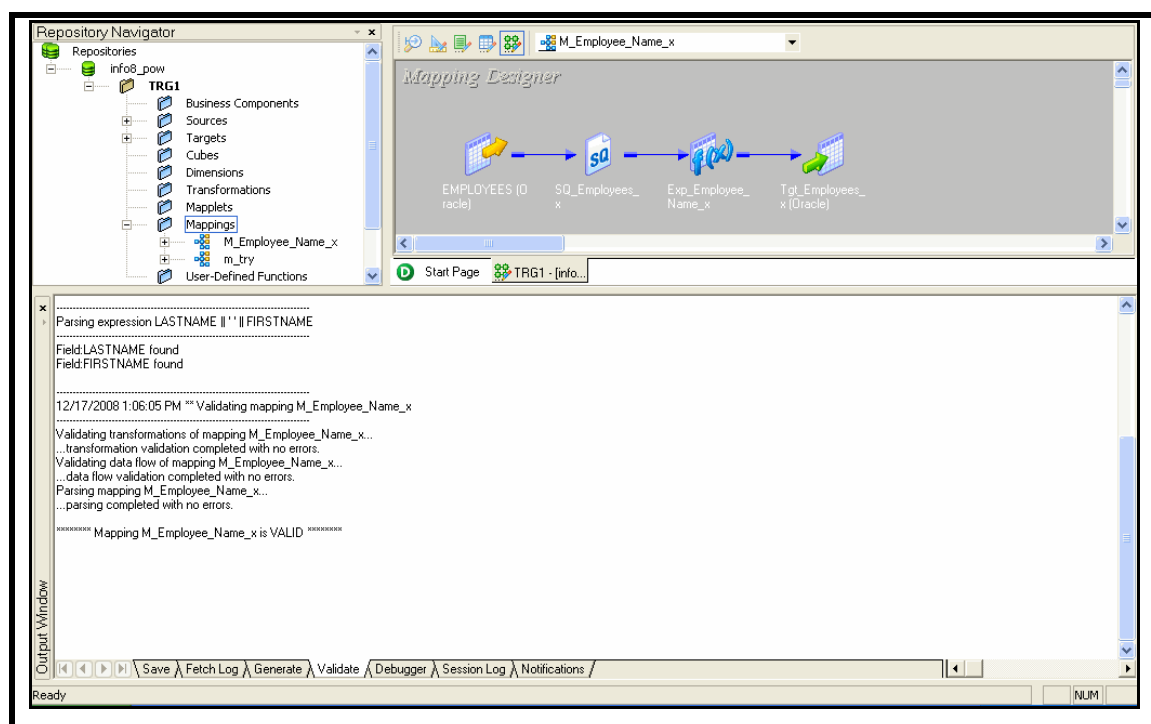
3. Notice the links between expression transformation and target table.



4. The mapping is now complete and should look like the figure shown below:



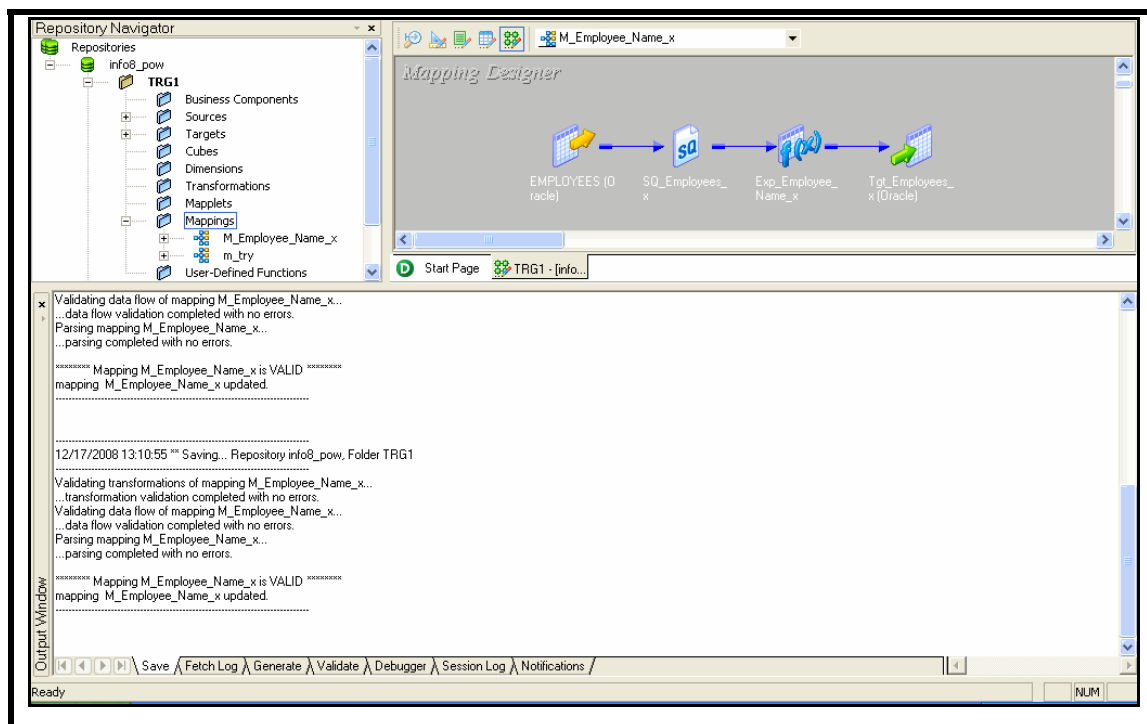
5. Select **MAPPINGS | VALIDATE** to validate the mapping.



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 3-1 Creating Workflow

Goals

- Create database connections for sources and targets
- Learn how to use Workflow Manager.
- Create a simple Workflow
- Create a session task and start task
- Link tasks

Time

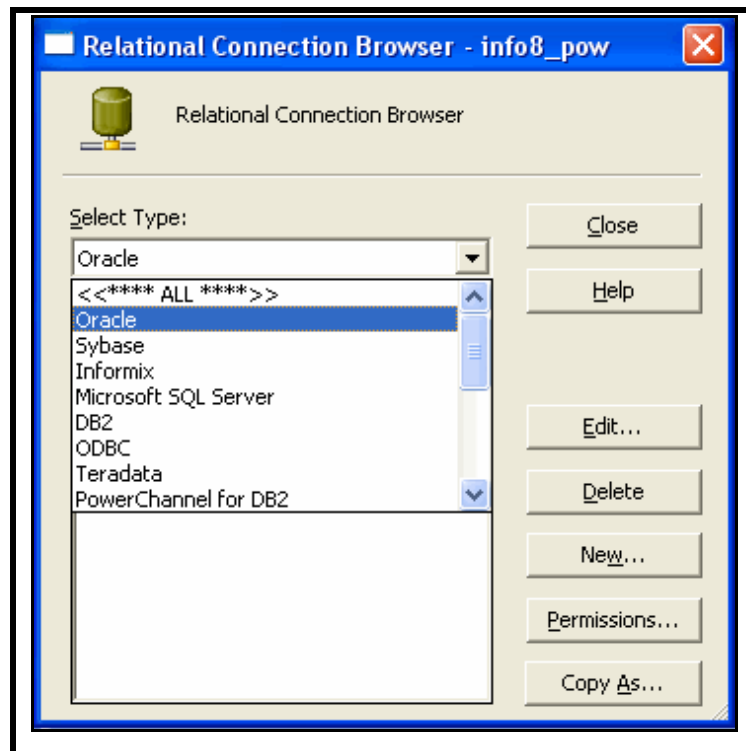
30 Minutes

Lab Setup

- Successful connection to the repository using Workflow Manager
- A valid mapping created

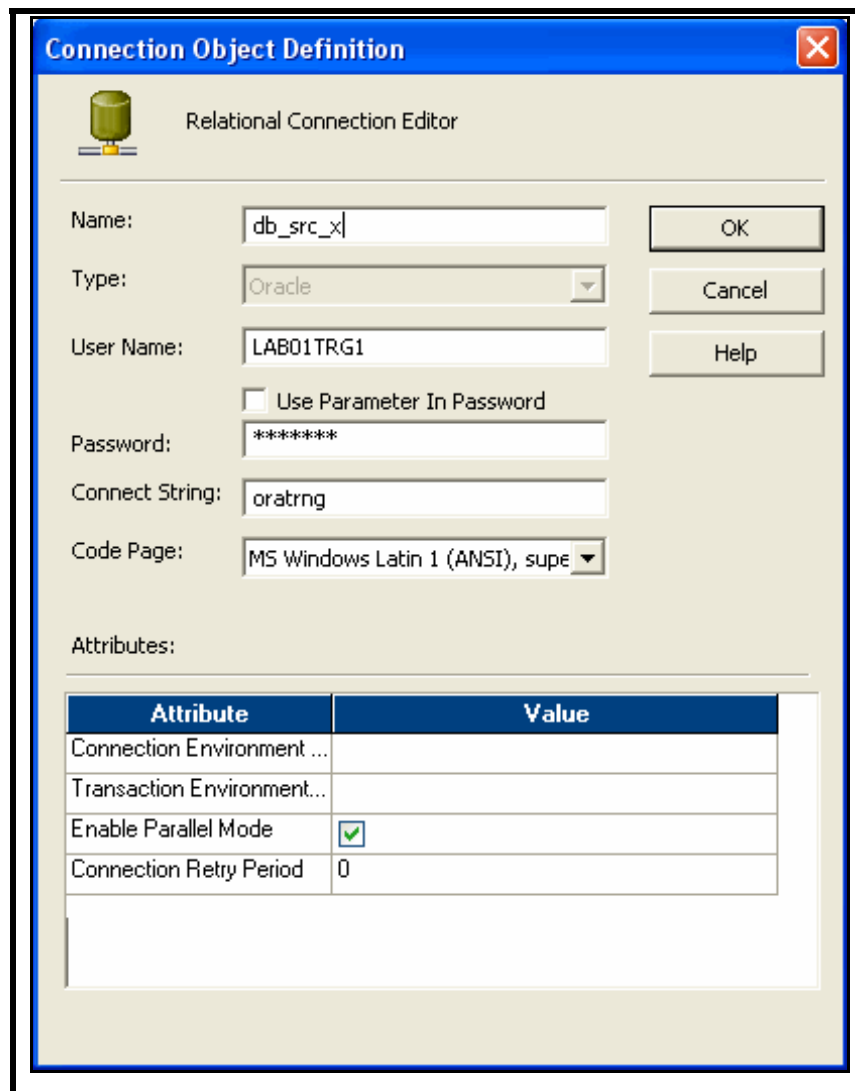
Create Relational Connection

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.



Connection Object Definition

Relational Connection Editor

Name: db_src_x

Type: Oracle

User Name: LAB01TRG1

☐ Use Parameter In Password

Password: *****

Connect String: oratrng

Code Page: MS Windows Latin 1 (ANSI), supe

Attributes:

| Attribute | Value |
|----------------------------|-------------------------------------|
| Connection Environment ... | |
| Transaction Environment... | |
| Enable Parallel Mode | <input checked="" type="checkbox"/> |
| Connection Retry Period | 0 |

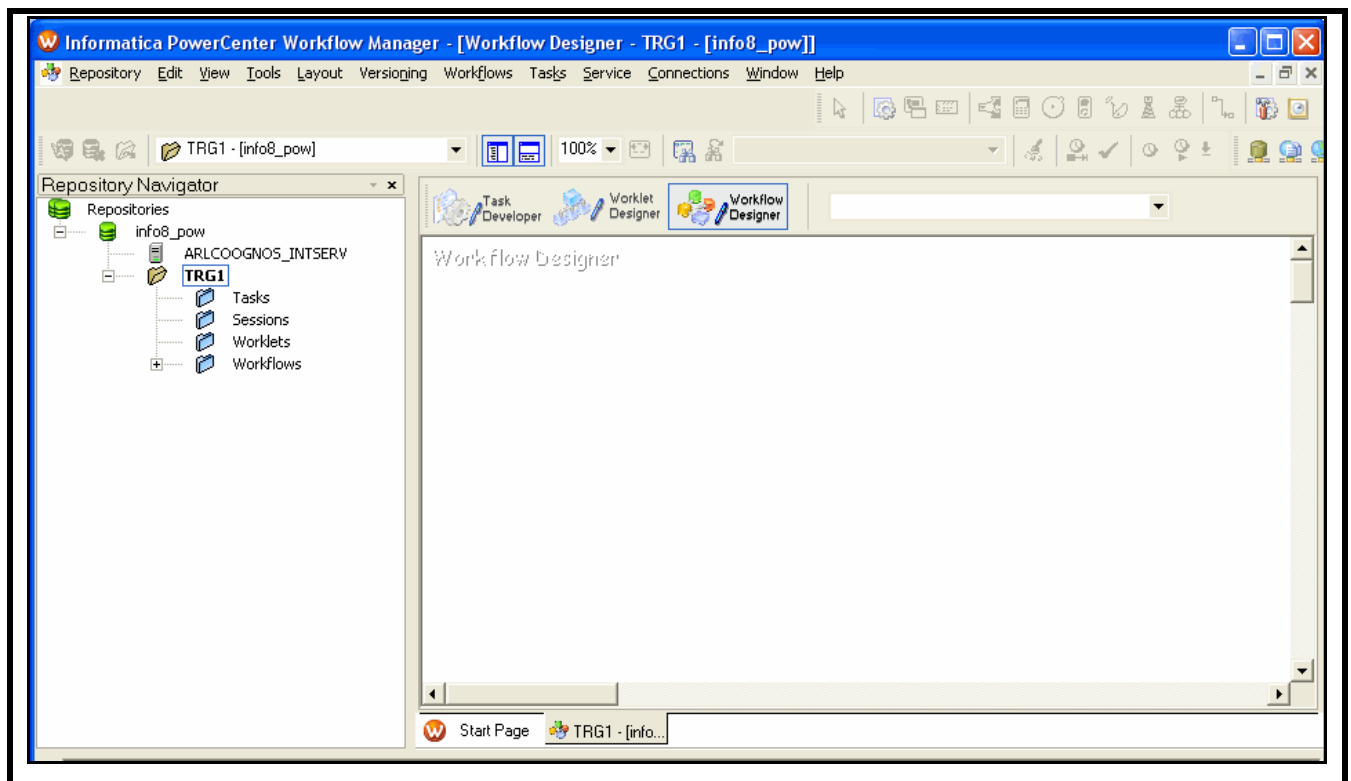
7. To create a target database connection, repeat steps 1 and 2 and in the Connection Object Definition box, enter the name db_tgt_x, username, password and connection string to connect to the target database db_tgt_x.

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).

Create Workflow - NEWWORKFLOW25

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

Name: wf_Employee_Name_x

Comments:

Integration Service: ARLCOGNOS_INTSERV

Suspension email:

Runtime options: ☐ Disabled ☐ Suspend on error

Web Services: ☐ Enabled [Config Service...](#)

Configure Concurrent Execution: ☐ Enabled [Configure Concurrent Execution ...](#)

Load Balancing

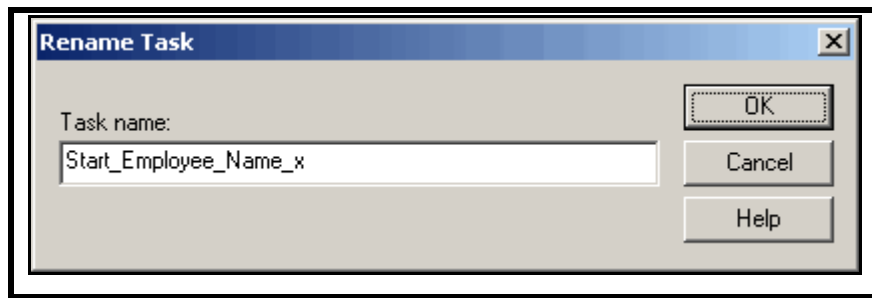
Service Level: Default

OK Cancel Help

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.

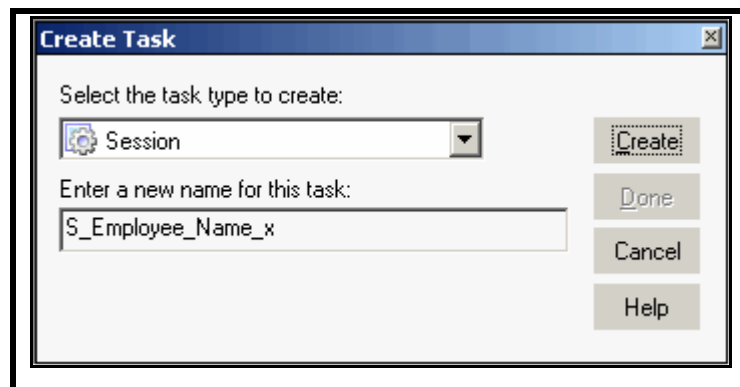
Edit the Start Task

1. Double Click on the Start task
2. Click the Rename button
3. Type Start_Employee_Name_x in the Rename Task dialog box.

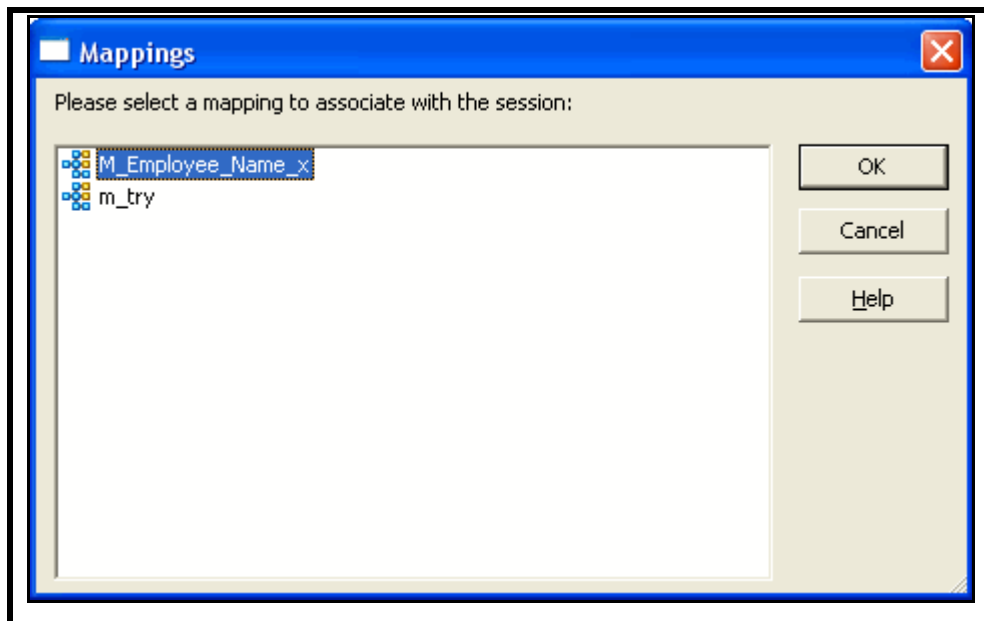


Create a Session Task

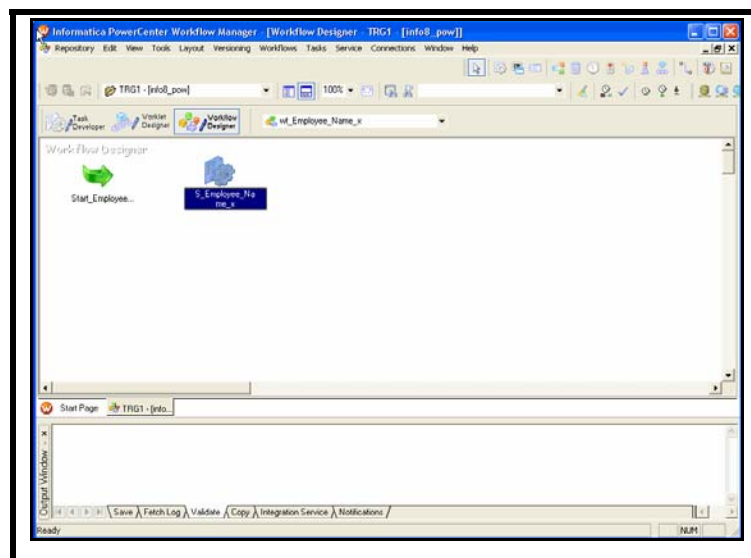
1. Click on TASKS | CREATE and In the Create task dialog box enter a name for the task as S_Employee_Name_x



2. Associate a mapping with the task. The mapping is M_Employee_Name_x, which you just created.

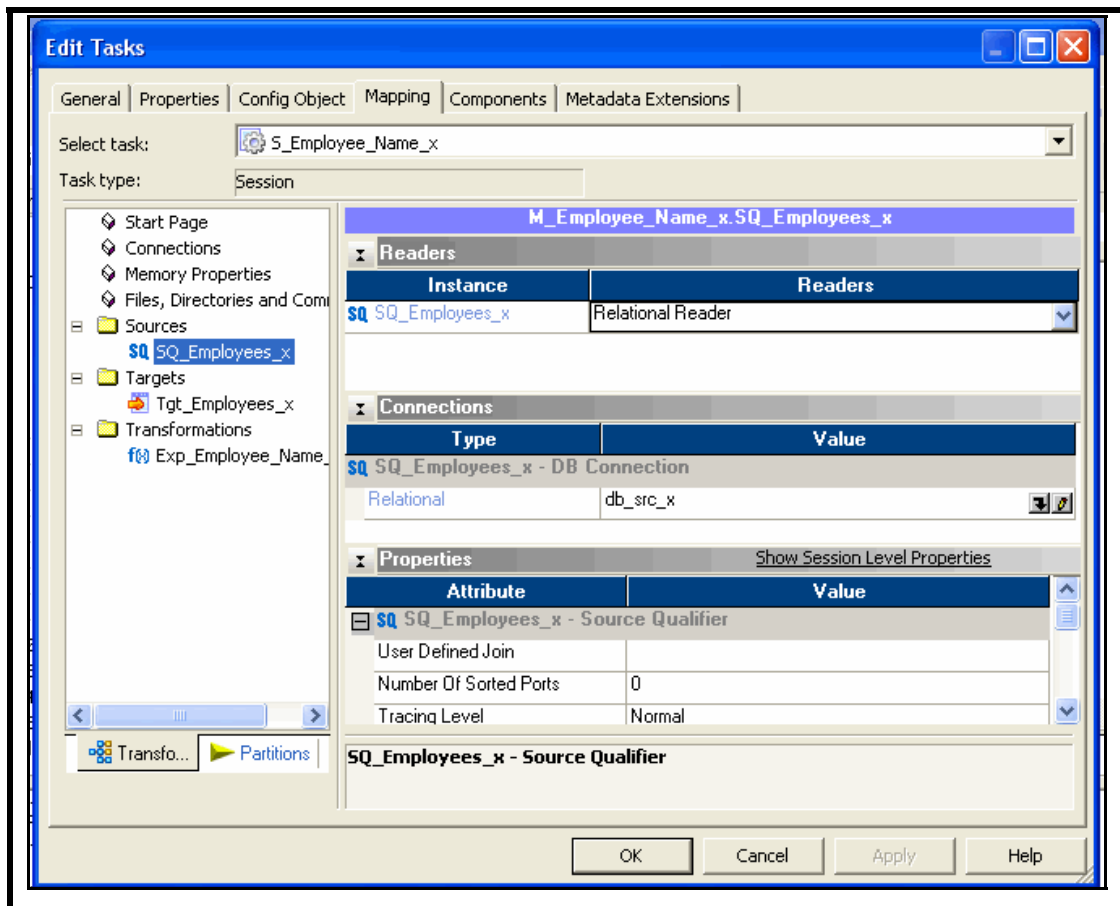


3. Click on OK in Mappings dialog box and Done in Create Task Dialog box.
4. A Session Task appears in the Workspace. Select REPOSITORY |SAVE.

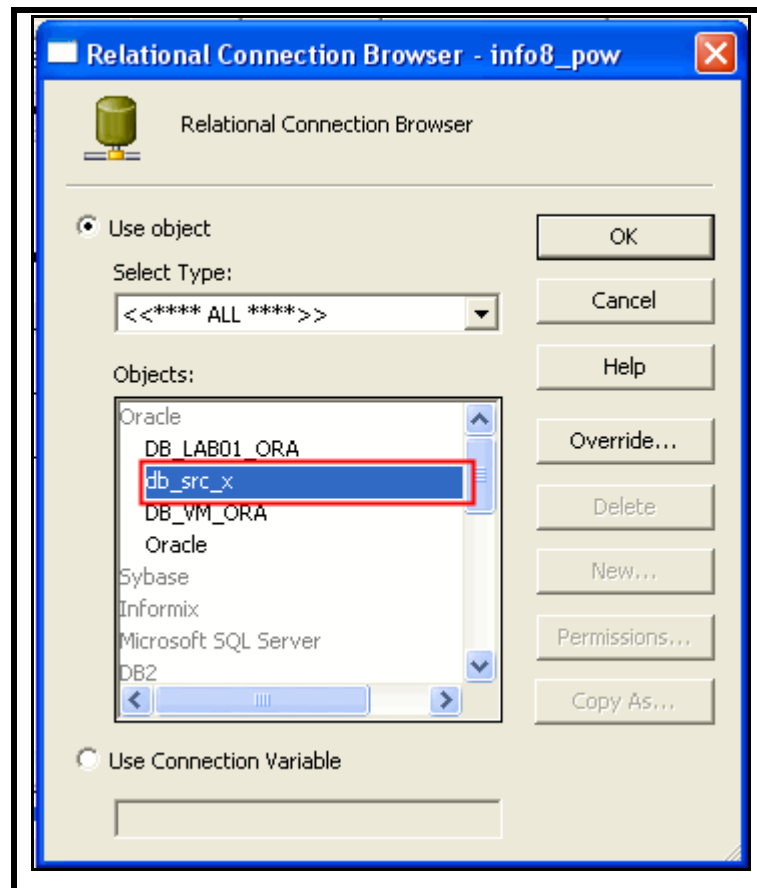


Edit Session Task

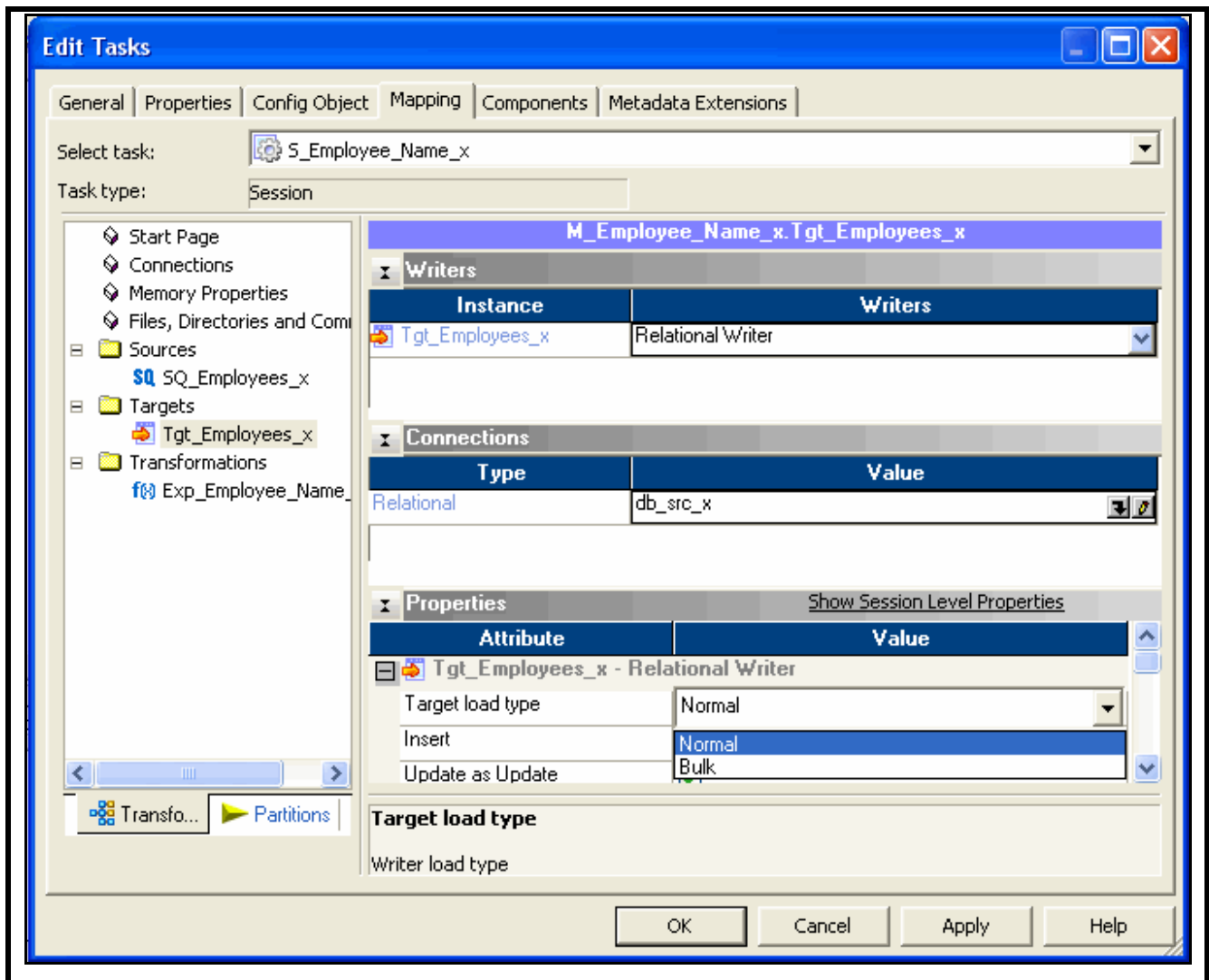
1. Double-click on the S_Employee_Name_x Session Task in the Workspace and click on the Mapping tab.
2. Under the Sources folder, click on the down arrow to open the Relational Connection Browser as shown in the figure below.




- In the Relational Connection Browser, select the database type from the dropdown and select the source database connection just created, from the Objects box.

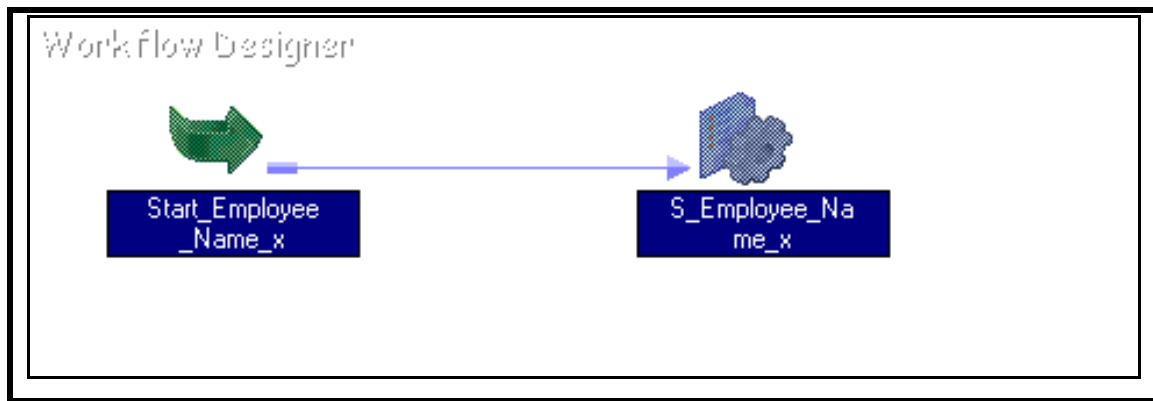


4. Under the Target folder, click on the down arrow to open the Relational Connection Browser and select the target database connection.
5. Under Properties, Select **Normal** for the **Target load type**.

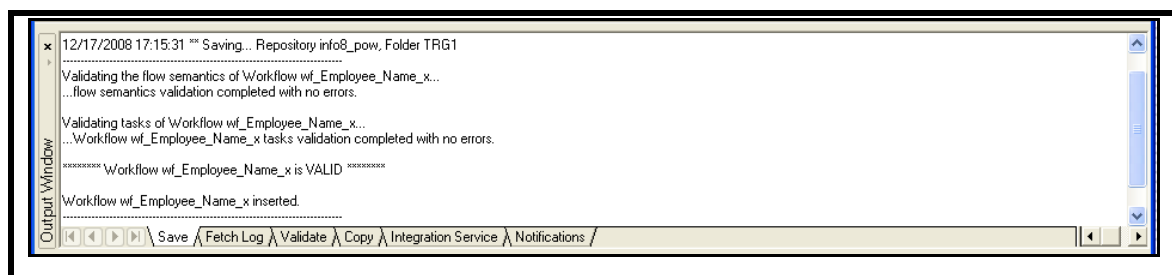


Link Workflow Tasks

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 3-2 Start and Monitor Workflows

| | |
|------------------|---|
| Goals | <ul style="list-style-type: none">• Different methods to start a Workflow• Monitor a Workflow• Create a simple Workflow• Create a session task and start task• Link tasks |
| Time | 15 Minutes |
| Lab Setup | Successful connection to the repository using Workflow Manager A valid Workflow created |

Start and Monitor The Workflow

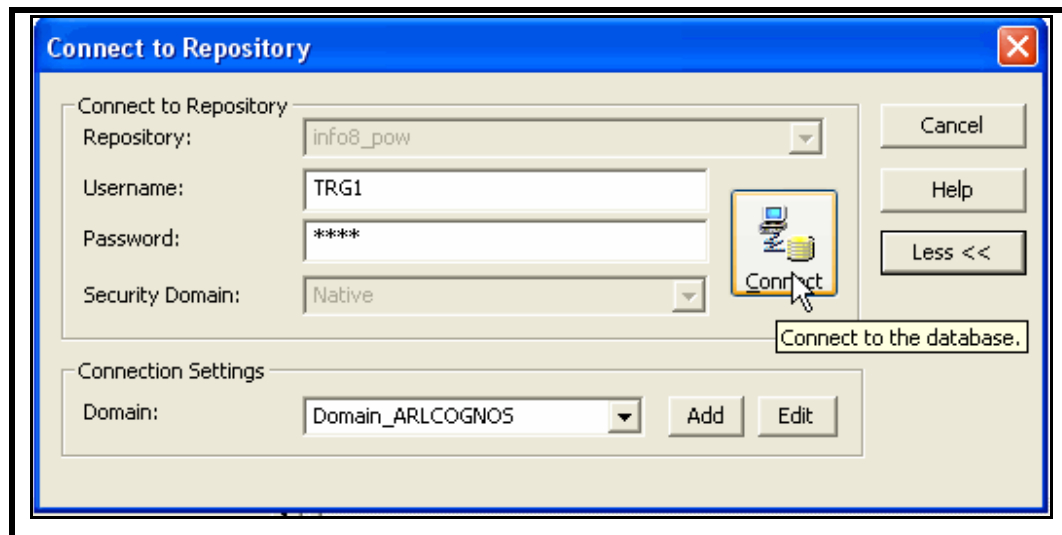
1. If the Workflow is valid, it is ready for execution. In the Workflow Designer, use one of the following methods to start the wf_Employee_Name_x Workflow:
 - i. Select WORKFLOWS | START WORKFLOW.
 - ii. Right-click in the Workspace and select Start Workflow or
 - iii. Right-click on the wf_Employee_Name_x 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

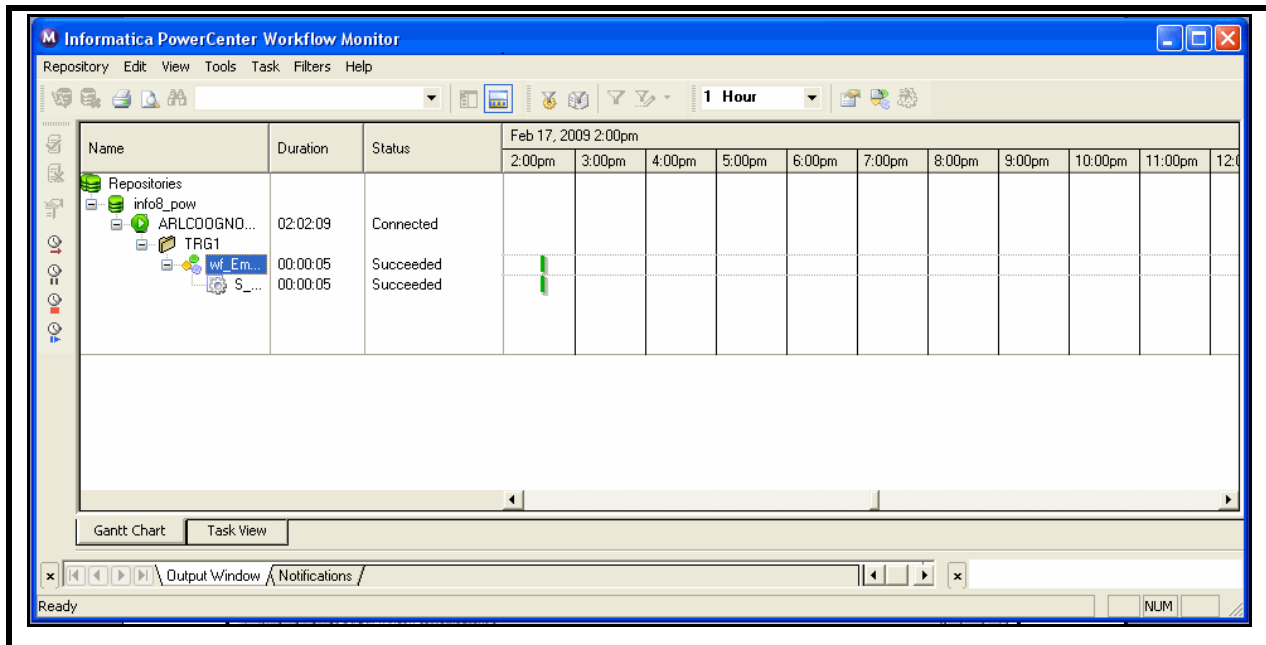
Click on the  icon in the toolbar; or double-click on Repository in the Navigator Window. The **Connect To** Repository box appears.

Note: Enter repository, server and login details as given by your Instructor.

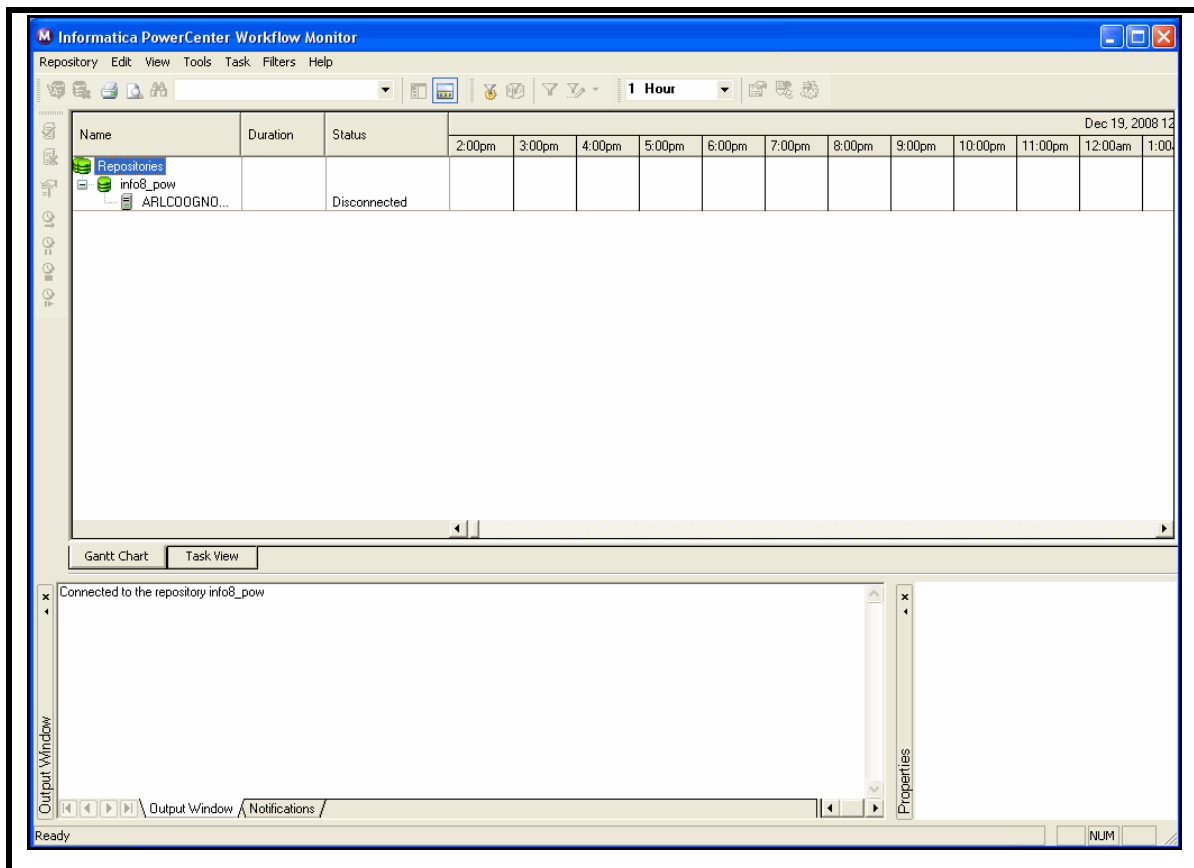
- iii. Select the repository from the Repository pull-down list.
- iv. Enter the Username in the Username box.
- v. Enter the Password in the Password box.
- vi. Select the Domain from the domain pull-down list and connect to database.



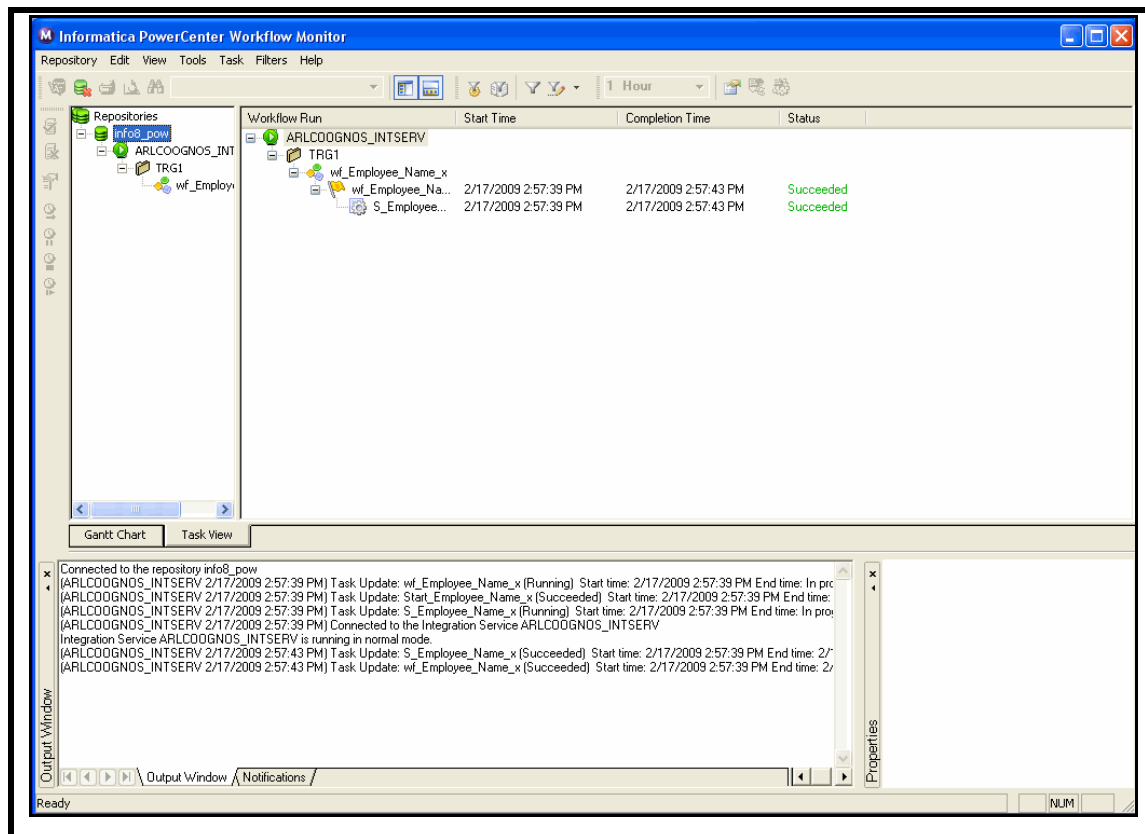
3. Double click on the folder to view Workflow sessions.
4. You get two views, the Gantt Chart View and the Task View.




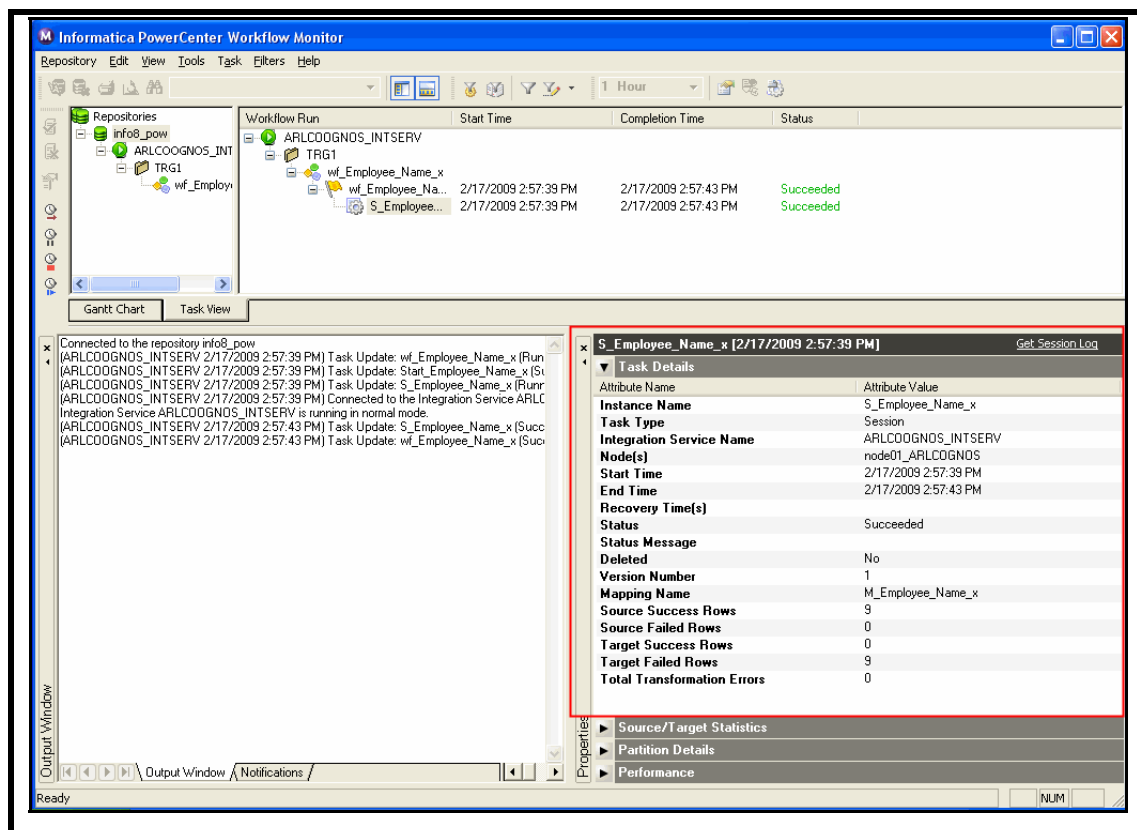
5. Select the Gantt View tab. This view displays details about workflow runs in chronological format. It displays the following information as shown below:



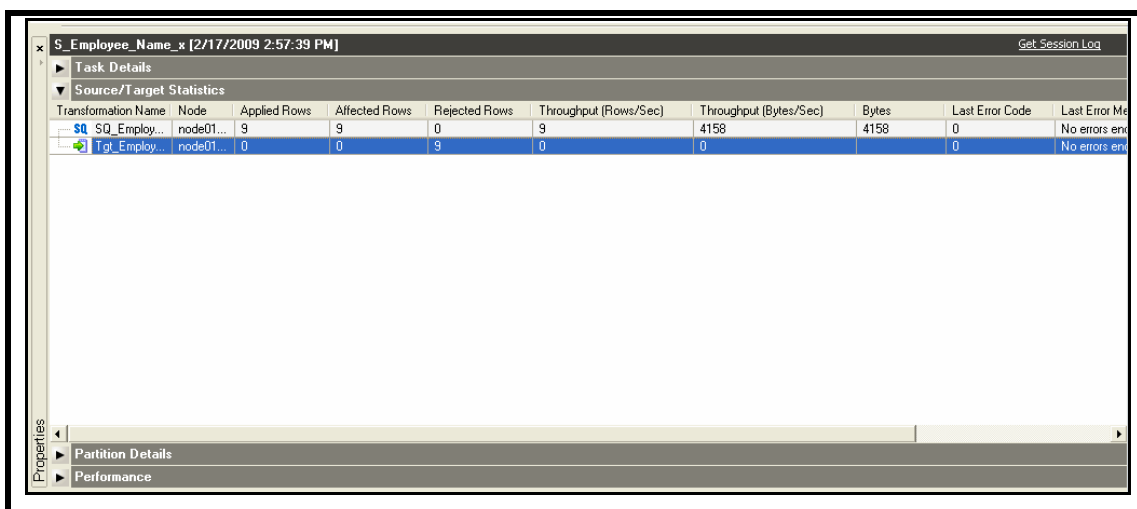
6. Select the Task View. This view displays details about workflow runs in a report format. The Status column gives the following information:




- 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:

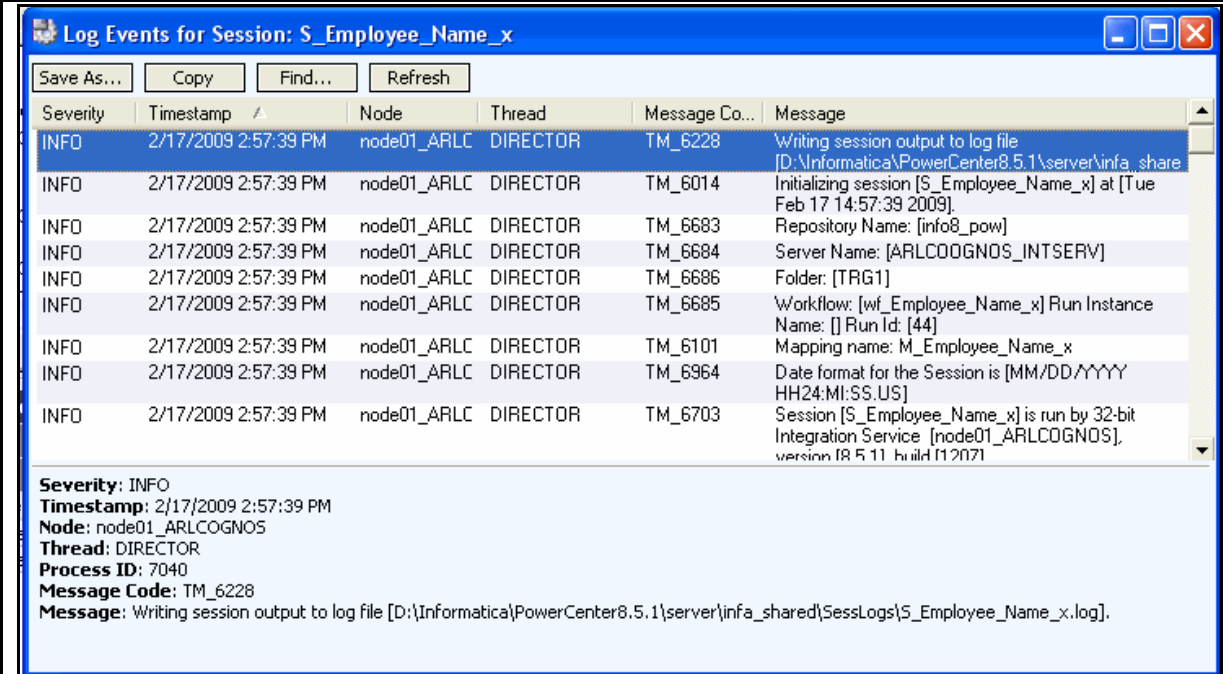


9. Click on the Source/Target Statistics tab in Properties window. More detail on the number of rows handled by the Server are shown here:



- 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



Log Events for Session: S_Employee_Name_x

Save As... Copy Find... Refresh

| Severity | Timestamp | Node | Thread | Message Co... | Message |
|----------|----------------------|-------------|----------|---------------|---|
| INFO | 2/17/2009 2:57:39 PM | node01_ARLC | DIRECTOR | TM_6228 | Writing session output to log file [D:\Informatica\PowerCenter8.5.1\server\infa_share... |
| INFO | 2/17/2009 2:57:39 PM | node01_ARLC | DIRECTOR | TM_6014 | Initializing session [S_Employee_Name_x] at [Tue Feb 17 14:57:39 2009]. |
| INFO | 2/17/2009 2:57:39 PM | node01_ARLC | DIRECTOR | TM_6683 | Repository Name: [info8_pow] |
| INFO | 2/17/2009 2:57:39 PM | node01_ARLC | DIRECTOR | TM_6684 | Server Name: [ARLCOGNOS_INTSERV] |
| INFO | 2/17/2009 2:57:39 PM | node01_ARLC | DIRECTOR | TM_6686 | Folder: [TRG1] |
| INFO | 2/17/2009 2:57:39 PM | node01_ARLC | DIRECTOR | TM_6685 | Workflow: [wf_Employee_Name_x] Run Instance Name: [] Run Id: [44] |
| INFO | 2/17/2009 2:57:39 PM | node01_ARLC | DIRECTOR | TM_6101 | Mapping name: M_Employee_Name_x |
| INFO | 2/17/2009 2:57:39 PM | node01_ARLC | DIRECTOR | TM_6964 | Date format for the Session is [MM/DD/YYYY HH24:MI:SS.US] |
| INFO | 2/17/2009 2:57:39 PM | node01_ARLC | DIRECTOR | TM_6703 | Session [S_Employee_Name_x] is run by 32-bit Integration Service [node01_ARLCOGNOS], version 18.5.11 build 112071 |

Severity: INFO
Timestamp: 2/17/2009 2:57:39 PM
Node: node01_ARLCOGNOS
Thread: DIRECTOR
Process ID: 7040
Message Code: TM_6228
Message: Writing session output to log file [D:\Informatica\PowerCenter8.5.1\server\infa_shared\SessLogs\S_Employee_Name_x.log].

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 | Suyama 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 4-1 Sales Summary

Goals

- Create a mapping that gives the summary of all sales by item description, state, and month
 - Use multiple sources
 - Create and use Expression and Aggregator transformations in the mapping
 - Use functions in the Expression transformation
 - Group columns in the Aggregator transformation

Time

180 Minutes

Lab Setup

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.

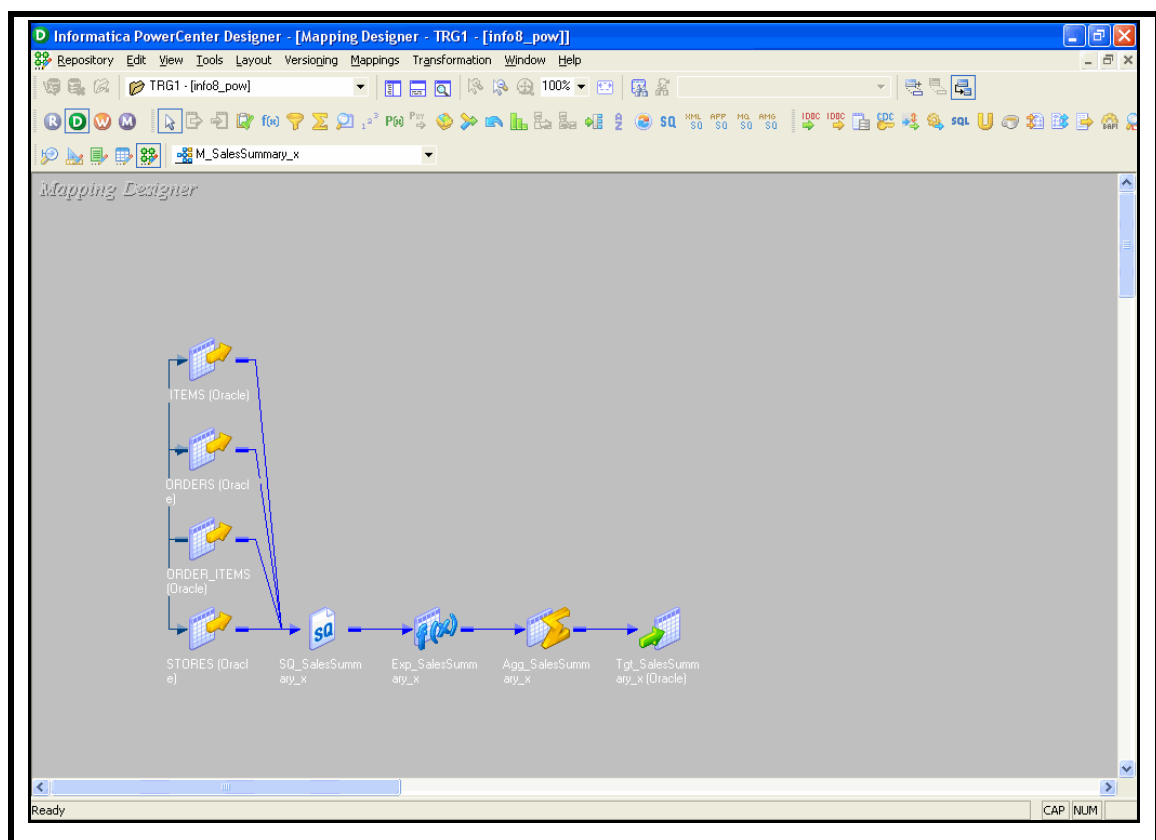
Solution

- 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 |

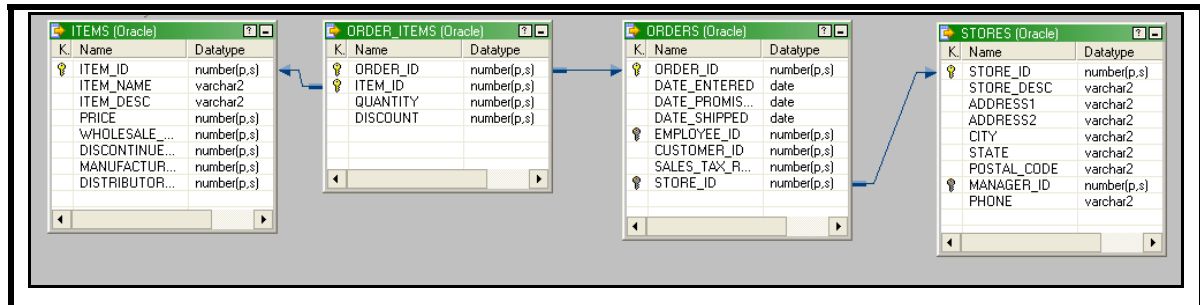
| DESCRIPTION | TOTAL_SOLD | TOTAL_PRICE | MONTH | YEAR | ST |
|-----------------------|------------|-------------|-----------|------|----|
| Buoyancy Compensation | 4 | 1720 | November | 1998 | HI |
| Buoyancy Compensation | 7 | 2305 | October | 1998 | HI |
| Small Instruments | 13 | 736.8 | April | 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 |

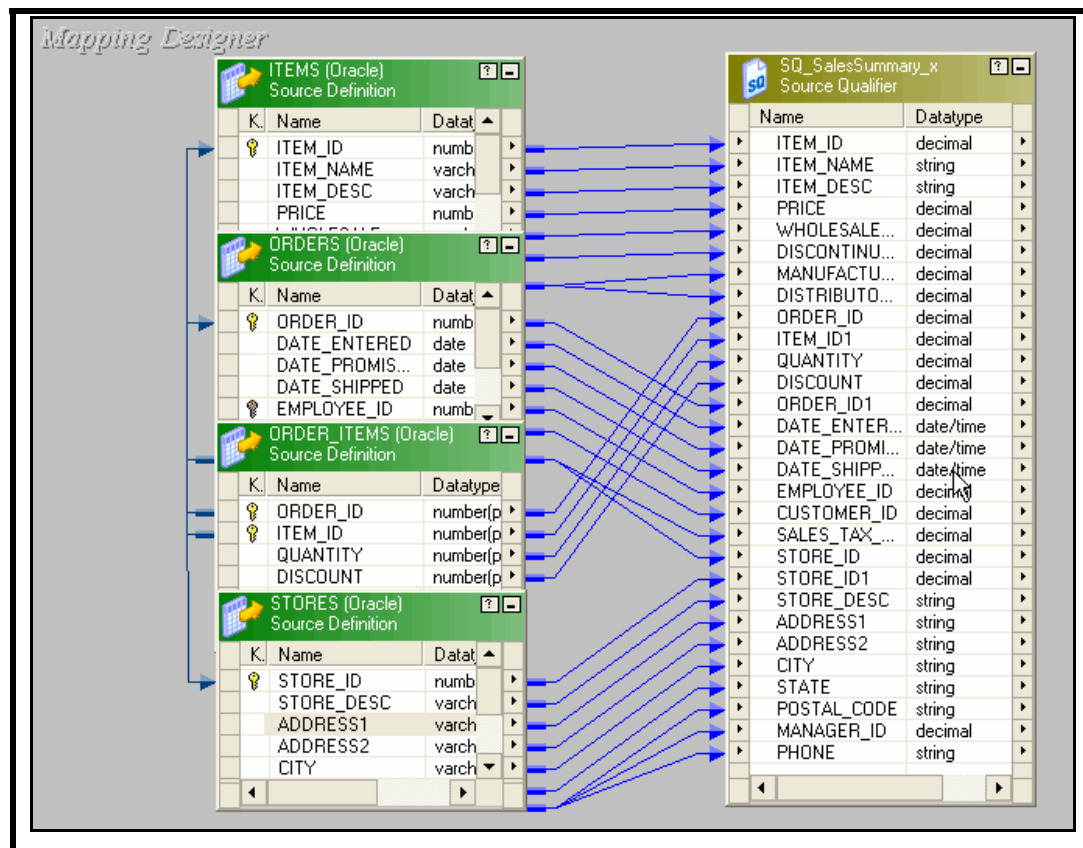
29 rows selected.

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

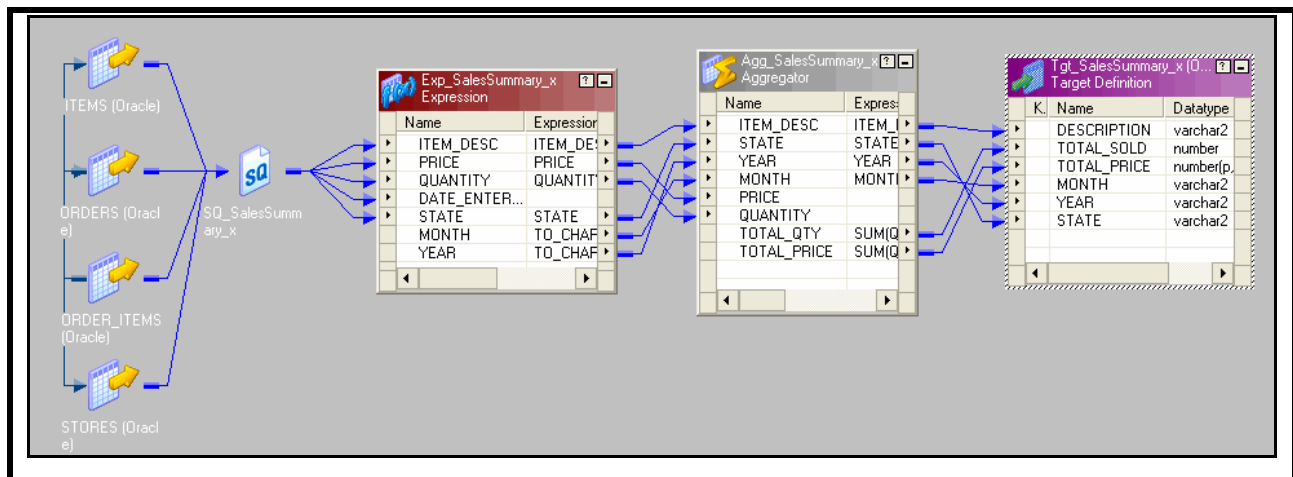
HINT: Select the Link Columns icon in the toolbar for Auto-link.



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.

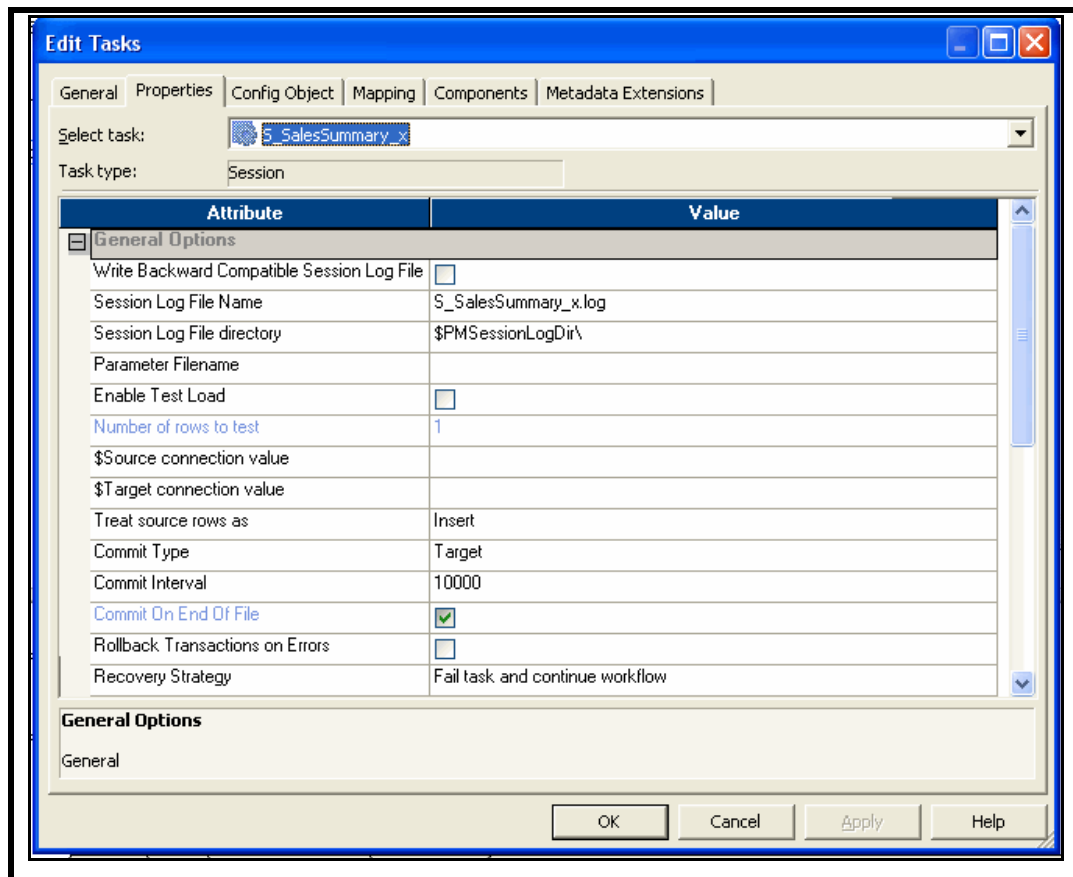


19. Create a Workflow by name wf_SalesSummary_x in Workflow Manager.
20. Under the Properties tab, note name of the **Workflow Log File Name:** wf_SalesSummary_x.log
21. To create non-reusable, local Metadata Extensions, click on the Metadata Extensions tab
 - i. Enter Creation_Date and Author into the Extension Name.
 - ii. Do not select Reusable.

The screenshot shows the 'Edit Tasks' window with the 'Metadata Extensions' tab selected. The task 's_SalesSummary_x' is selected, and the task type is 'Session'. The table below shows the metadata extensions defined for this task.

| Extension name | Datatype | Prec | Value | Reusable |
|------------------------------|----------|------|--------------|--------------------------|
| User Defined Metadata Domain | | | | |
| Author | String | 30 | Student Name | <input type="checkbox"/> |
| Creation_Date | String | 10 | mm/dd/yyyy | <input type="checkbox"/> |

- 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 5-1 New Customer

Goals

- Create a mapping which reads from a flat file and creates a relational table consisting of new customers
- Analyze a fixed width flat file
- Configure a Connected Lookup transformation
- Use a Filter transformation to exclude records from the lookup transformation pipeline

Time

120 Minutes

Lab Setup

Successful Connection to the repository using PowerCenter Client

Background

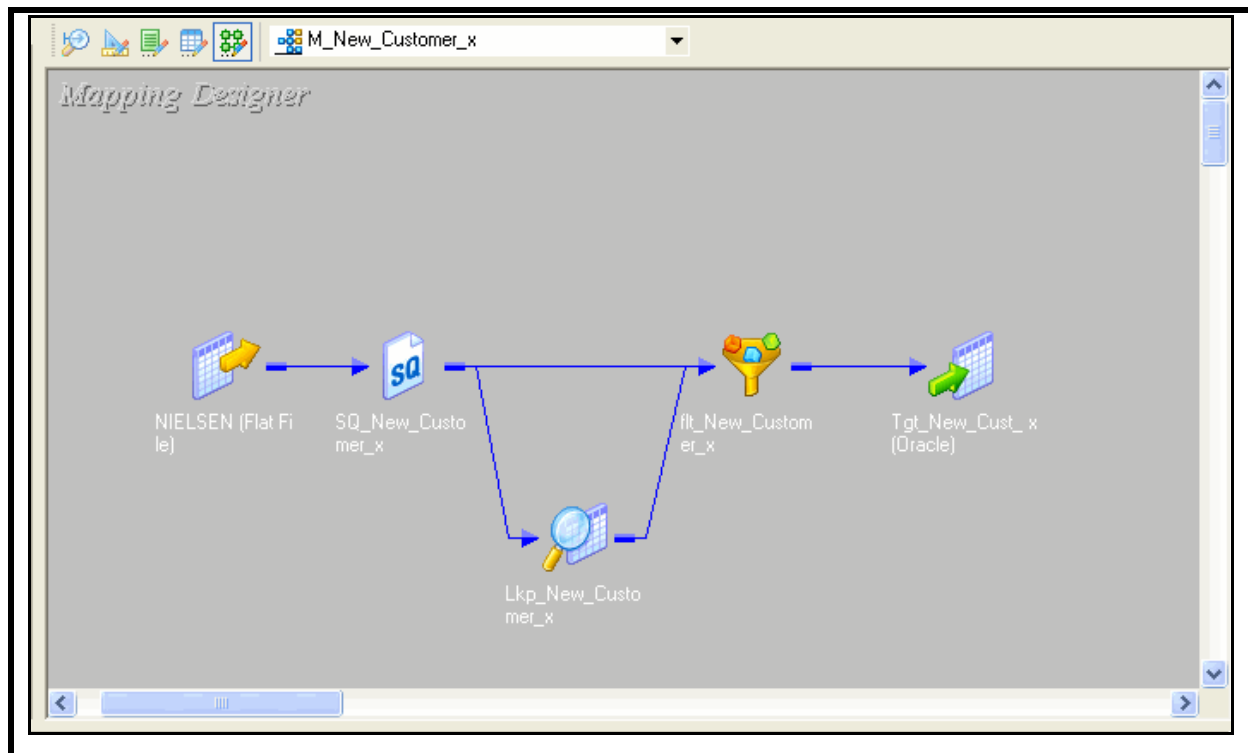
The Marketing department is holding a special promotion for potential customers. The company has purchased an industry listing from the Nielsen Research Company of their target market, which includes potential new customers as well as many of their current customers.

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 | Åkergatan 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 | Vaffeljernet | Smagsløget 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 |

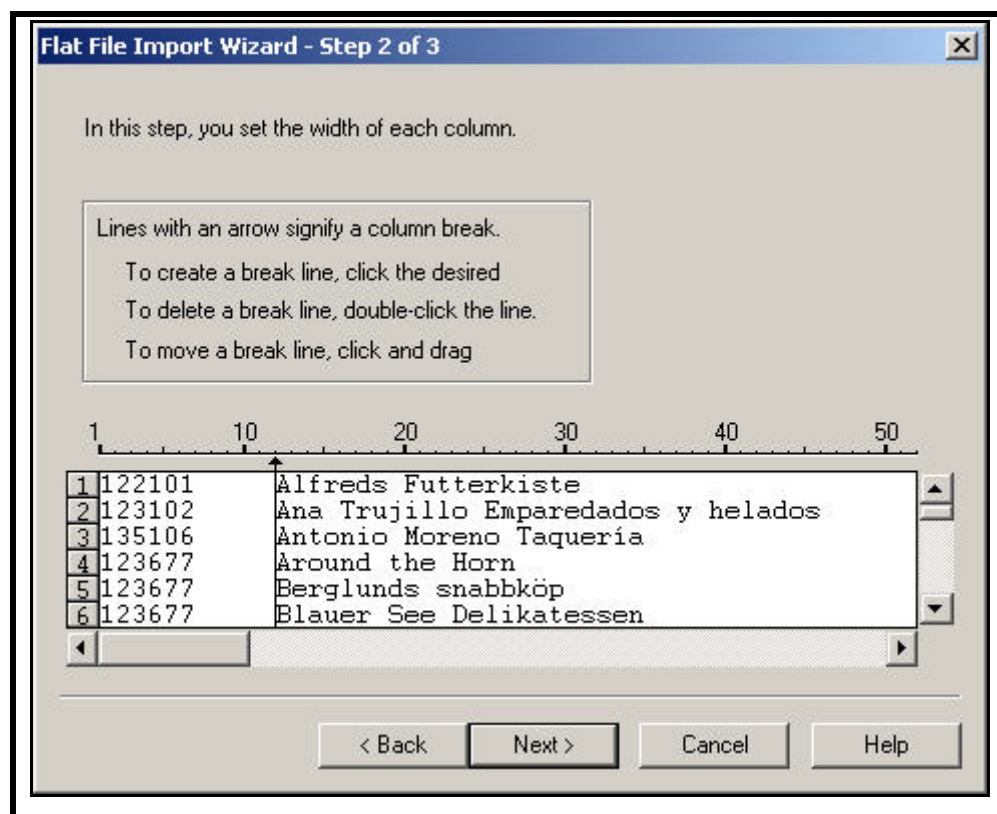
```
84 rows selected.
```

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:

Flat File Import Wizard - Step 3 of 3

Next, select the name and datatype for each column.

Column Information

Name:

Datatype:

☐ Text Length/Prec.

☒ Numeric Scale

☐ Datetime Width

Source definition:

| Name | Type | Len... | Scale |
|-------------|--------|--------|-------|
| Company_... | Text | 50 | - |
| Address1 | Text | 72 | - |
| Address2 | Text | 72 | - |
| City | Text | 30 | - |
| St | Text | 2 | - |
| Code | Num... | 10 | 0 |

| | Address2 | City | S | Code |
|---|----------|-----------|----|-------|
| 1 | | Butte | MT | 59801 |
| 2 | | Charleroi | | 24100 |
| 3 | | Bergamo | | |
| 4 | | | | |
| 5 | | Resende | SP | |
| 6 | | | | |

< Back Finish Cancel Help

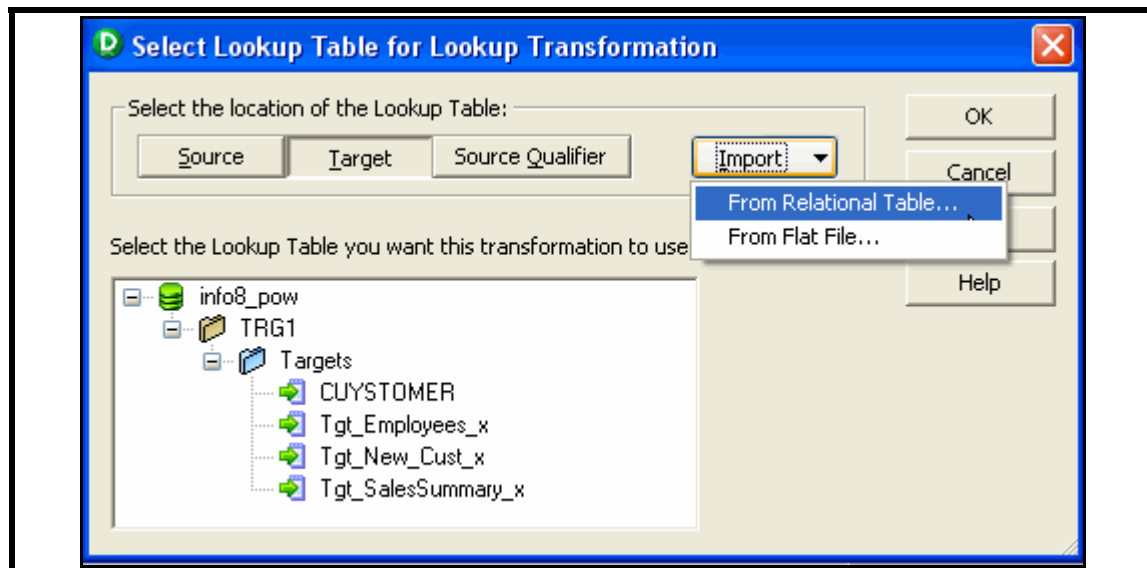
2. Change field name St to State and Code to Postal_Code and click finish.

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.

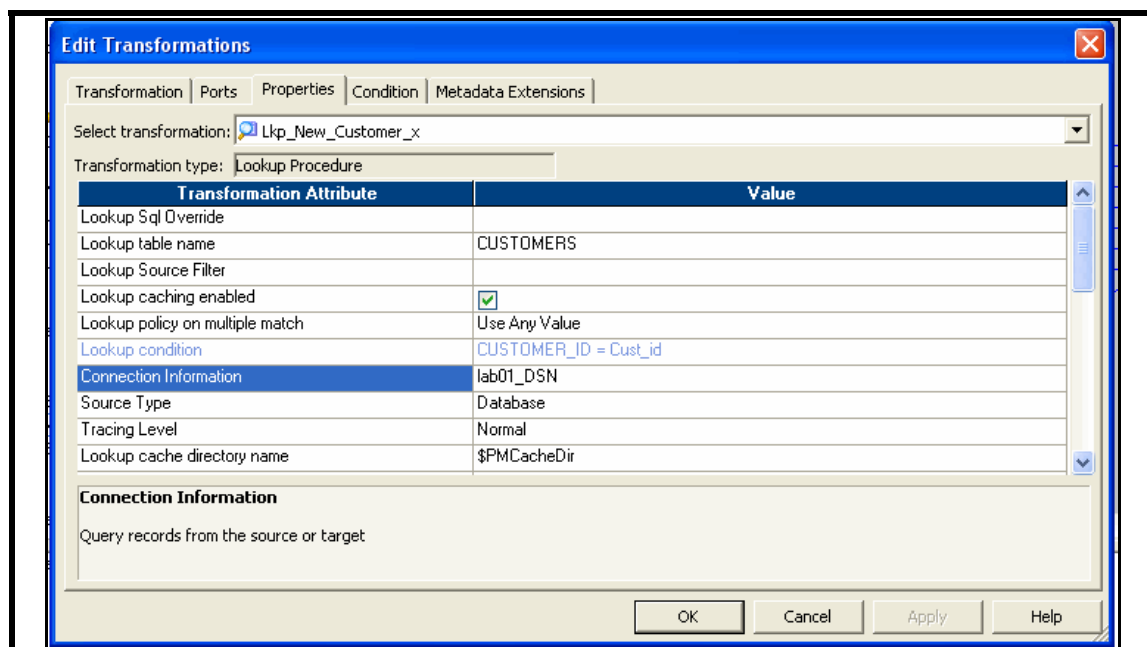
3. Name the new source definition NIELSEN. This is the name that will appear as metadata in the repository, for the source definition.
4. Create a target definition based on the structure of the source file definition with name Tgt_New_Cust_x.

Hint: From the Edit table properties in Target designer, change the database type to oracle.

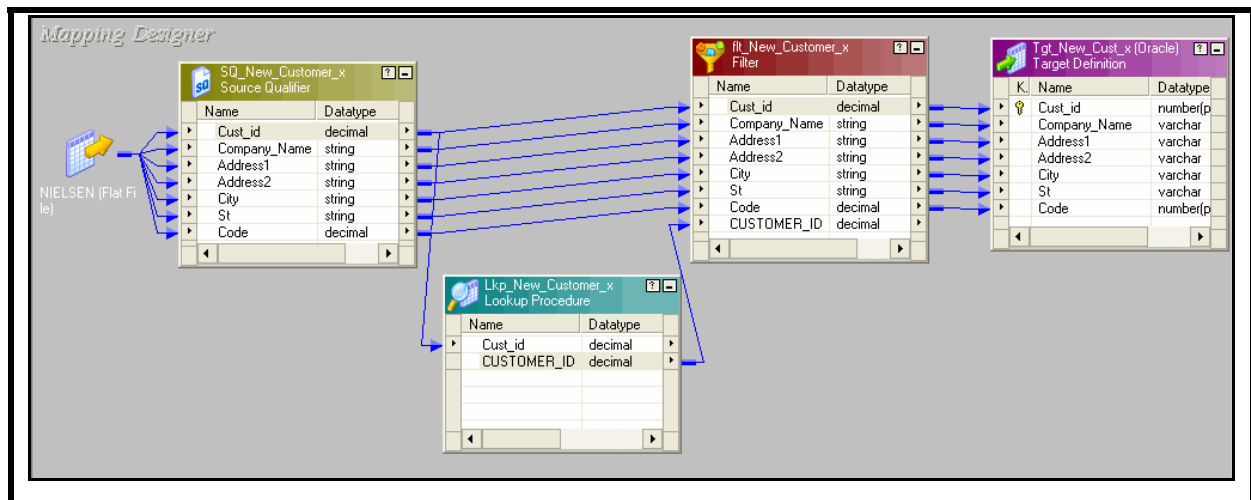
5. Create a new mapping with the name M_New_Customer_x
6. Create Lookup transformation with name Lkp_New_Customer_x.
7. Identify the Lookup table in the Lookup transformation. Use the CUSTOMERS table from the source database to serve as the Lookup table and import it from the database.



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 | Åkergatan 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 | Vaffeljernet | Smagsløget 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.
```

Lab 6-1 Flat File Join

Goals

- Analyze delimited files
 - Join heterogeneous sources
- Use Aggregator and Rank transformations to provide top ten revenue producing items

Time

60 minutes

Lab Setup

Successful connection to the repository using PowerCenter Client

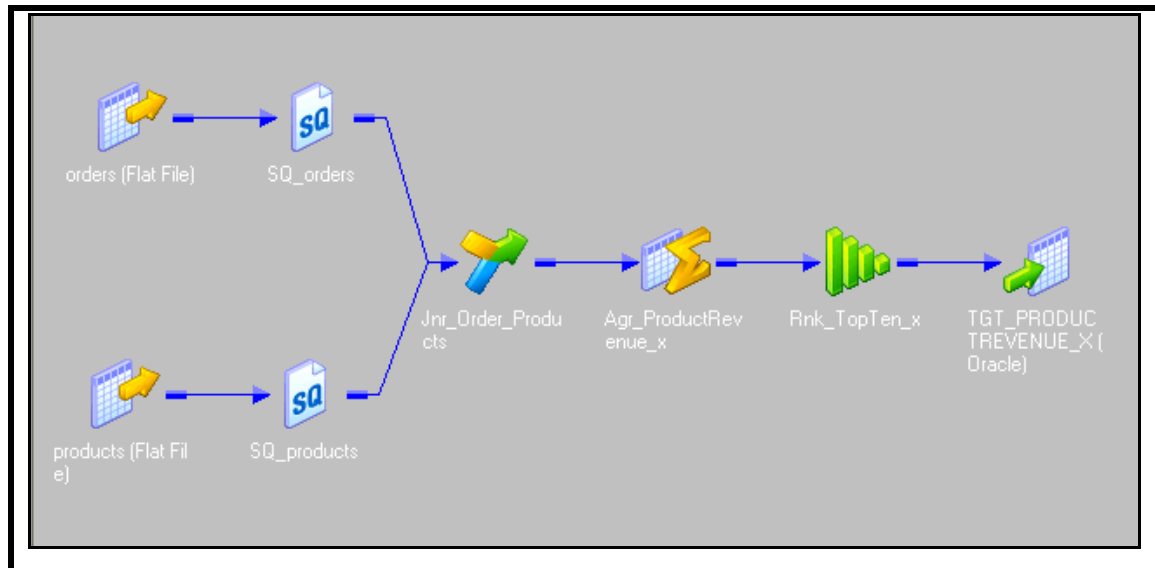
Background

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

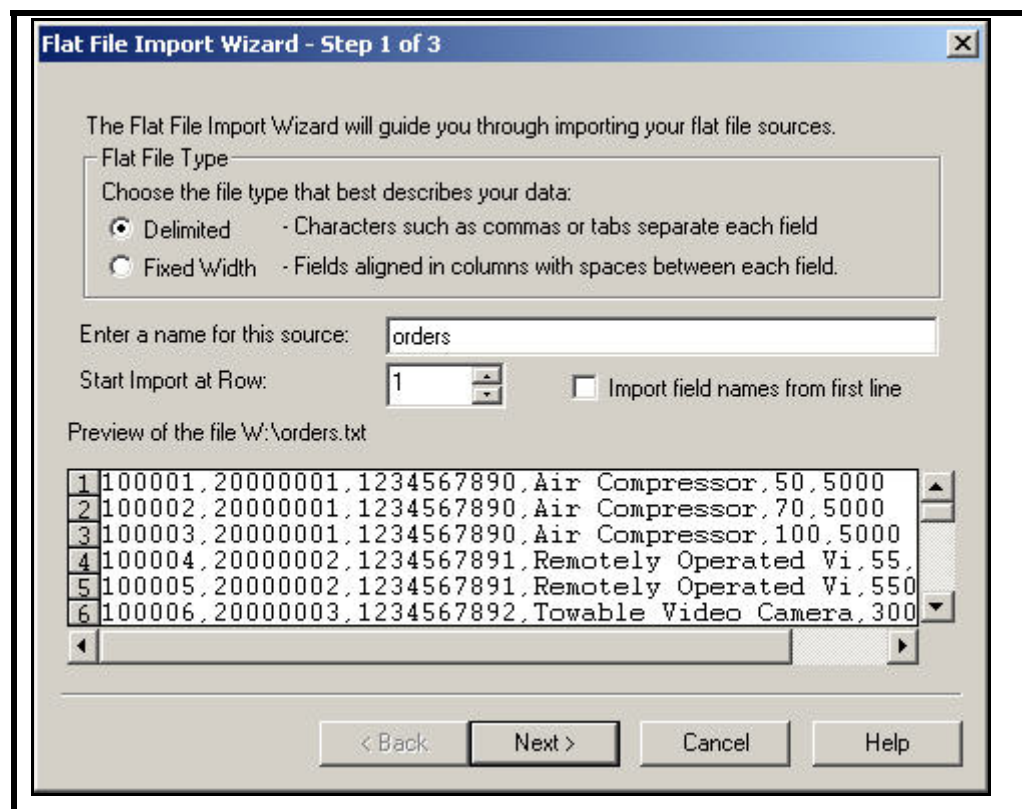
```
SQL> select * from tgt_ProductRevenue_x;
```

| ITEM_NO | ITEM_NAME | PRODUCT_CATEGORY | PRICE | TOTAL_QUANTITY | TOTAL_REVENUE |
|---------|-------------------------|-------------------|-------|----------------|---------------|
| 12347 | Towable Video Camera | Photo Equipment | 7000 | 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 | 8000 | 8900 | 71200000 |
| 12344 | Stabilizing vest | Misc Equipment | 6000 | 10900 | 65400000 |
| 12340 | Under Water Dive vehicl | Vehicle | 7500 | 3600 | 27000000 |
| 12346 | Remotely Operated Vi | Photo Equipment | 6000 | 605 | 3630000 |
| 12345 | Air Compressor | Misc Equipment | 5000 | 220 | 1100000 |

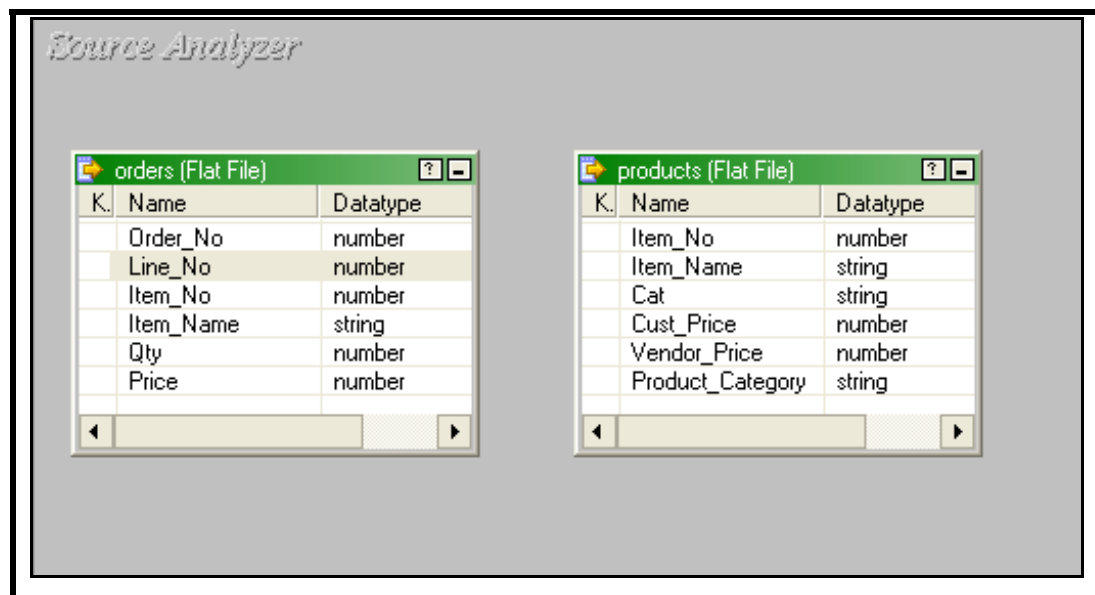
8 rows selected.

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.

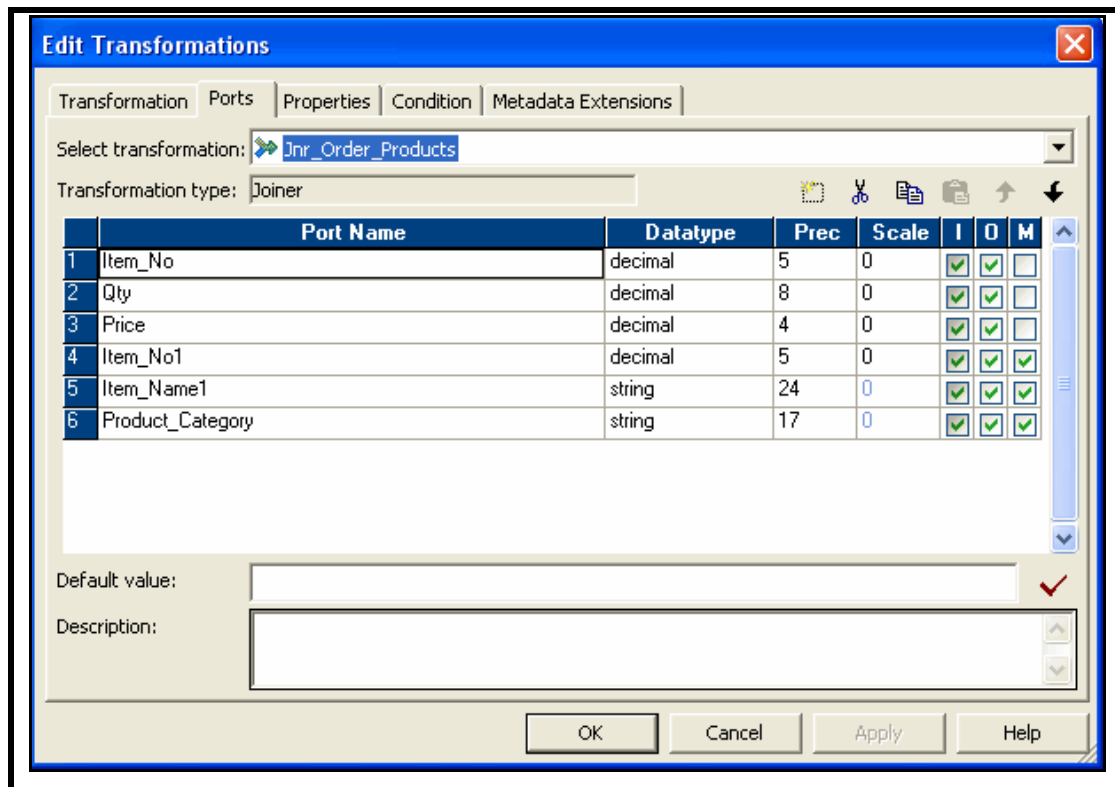


- ii. Name the source definition ORDERS,PRODUCTS.
- iii. Enter the column names and specify data types and field widths as shown below.



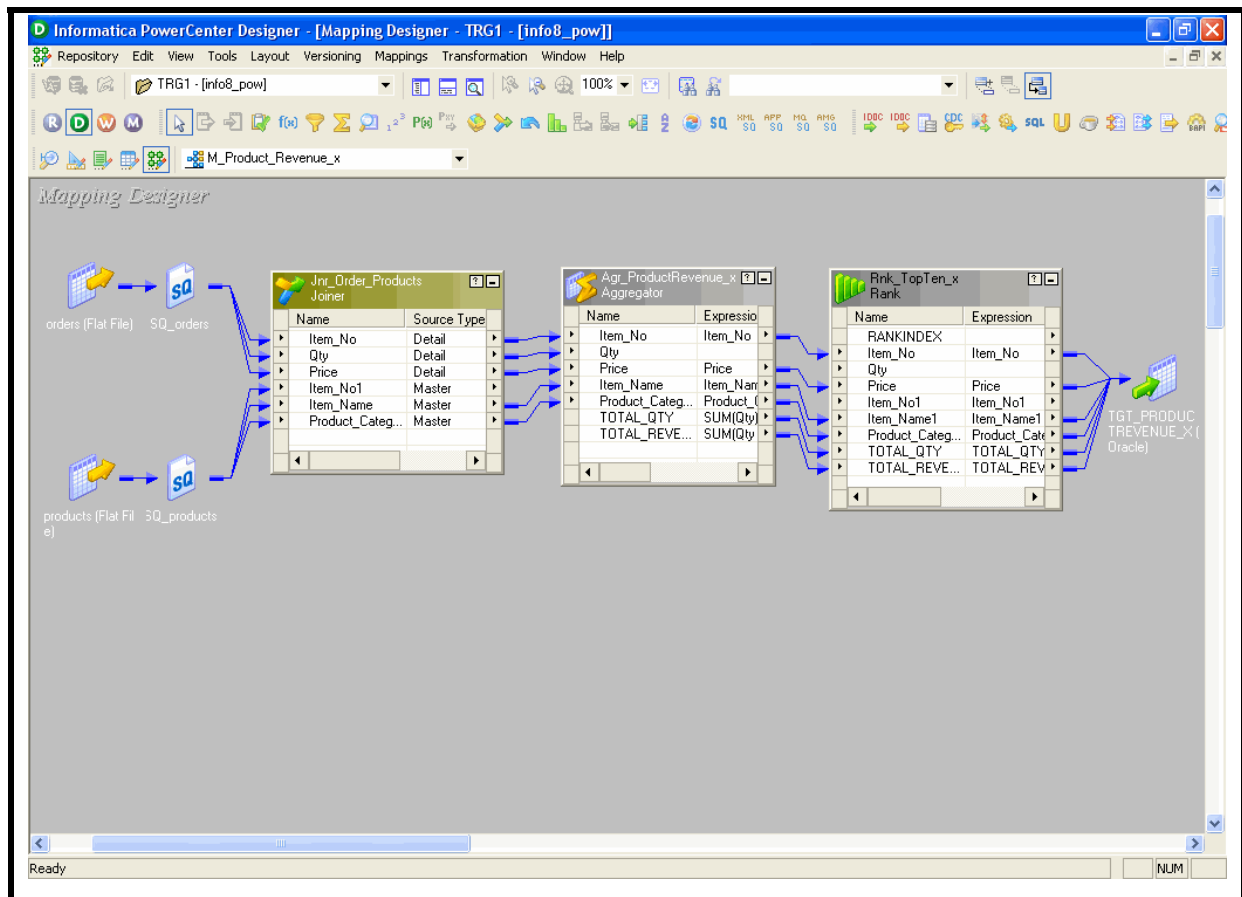
3. Create the target table with name Tgt_ProductRevenue_x .It should have following columns:
ITEM_NO,ITEM_NAME,PRODUCT_CATEGORY,PRICE,TOTAL_QUANTITY,TOTAL_REVENUE.
4. Create mapping with the name M_Product_Revenue_x.
5. DRAG the ORDERS and PRODUCTS source definitions.
6. Drag the Tgt_ProductRevenue_x 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 7-1 Create a Mapplet

Goals

- Learn to create a mapplet
- Understand how to use variables

Time

60 minutes

Lab Setup

PowerCenter Client connectivity

Background

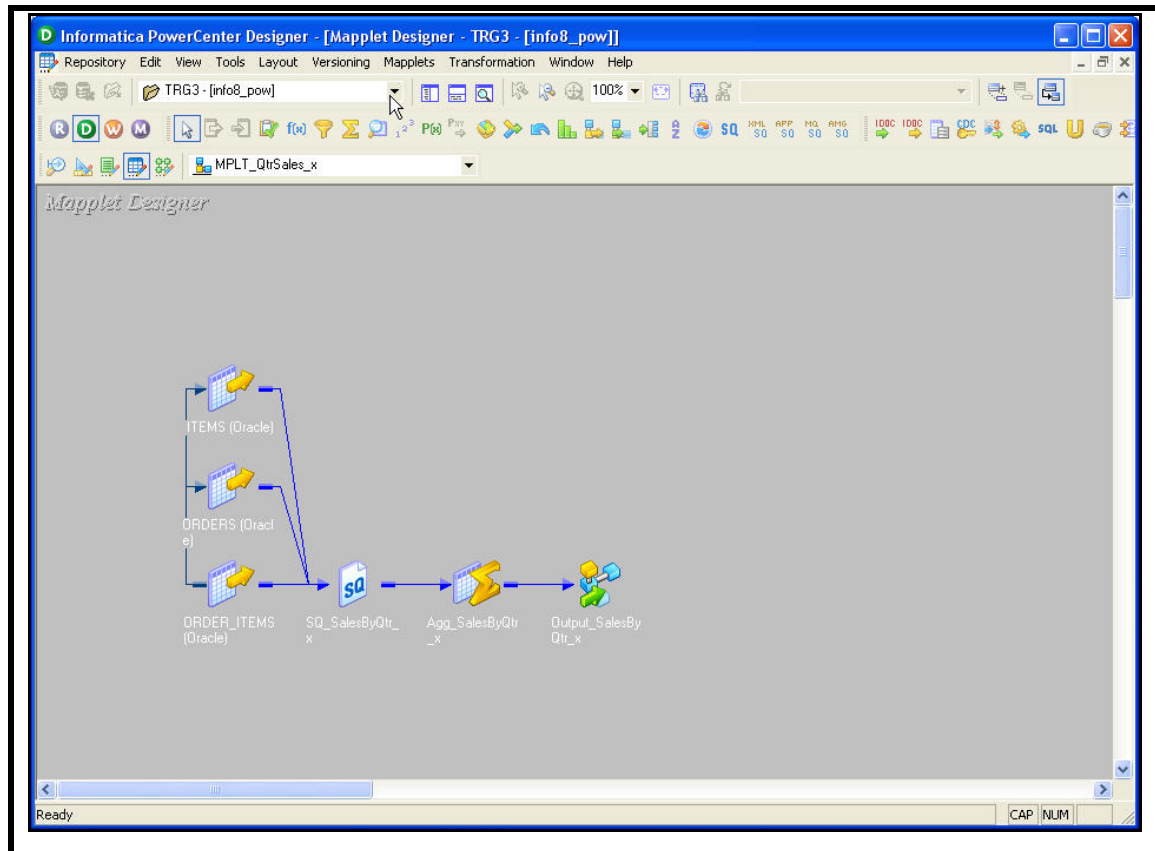
The Sales Department is interested in getting both the quarterly and yearly sales. The Data Mart server has a high CPU usage, so you would like to pre-calculate these amounts and put them in summary tables.

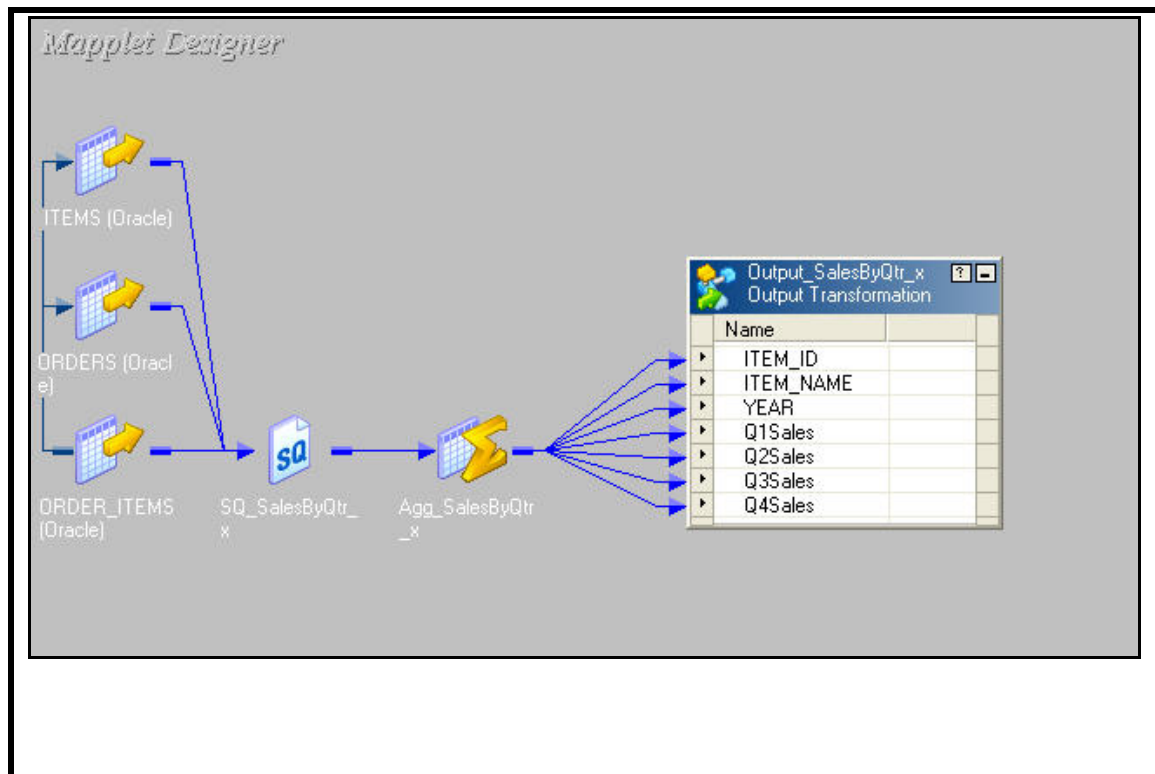
Solution

- Build a mapplet that uses multiple sources and aggregate functions
- Create a variable within the mapplet for use in the aggregate functions
- In the next lab, use this mapplet 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 mapplet |

Mapplet Layout



Mapplet output**Problem Solution**

1. Create a Mapplet 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 |
|---------|---------------|--------|

- 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> |

- Aggregate records by ITEM_ID, ITEM_NAME and YEAR.
- Select the Properties tab. Check the Sorted Input box. Exit the Edit Transformation dialog box.
- 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.
- Open the SQL query window. Click the Generate SQL button. Append the following text to the end of the default SQL statement:
- ORDER BY, ITEMS.ITEM_ID, ITEMS.ITEM_NAME, ORDERS.DATE_ENTERED.
- Verify that there are no errors in the SQL. Exit the SQL editor and the Source Qualifier Transformation.

11. Create a maplet output transformation by the name Output_SalesByQtr_x. Connect all the output ports of the Aggregator to the maplet Output transformation.
12. Verify the results of the maplet validation in the Output Window.

Lab 7-2 Quarterly-Sales Mapping

Goals

- Use a mapplet in a mapping
- Configure a Normalizer transformation

Time

60 minutes

Lab Setup

Quarterly sales mapplet created in the Lab 7-1

Background

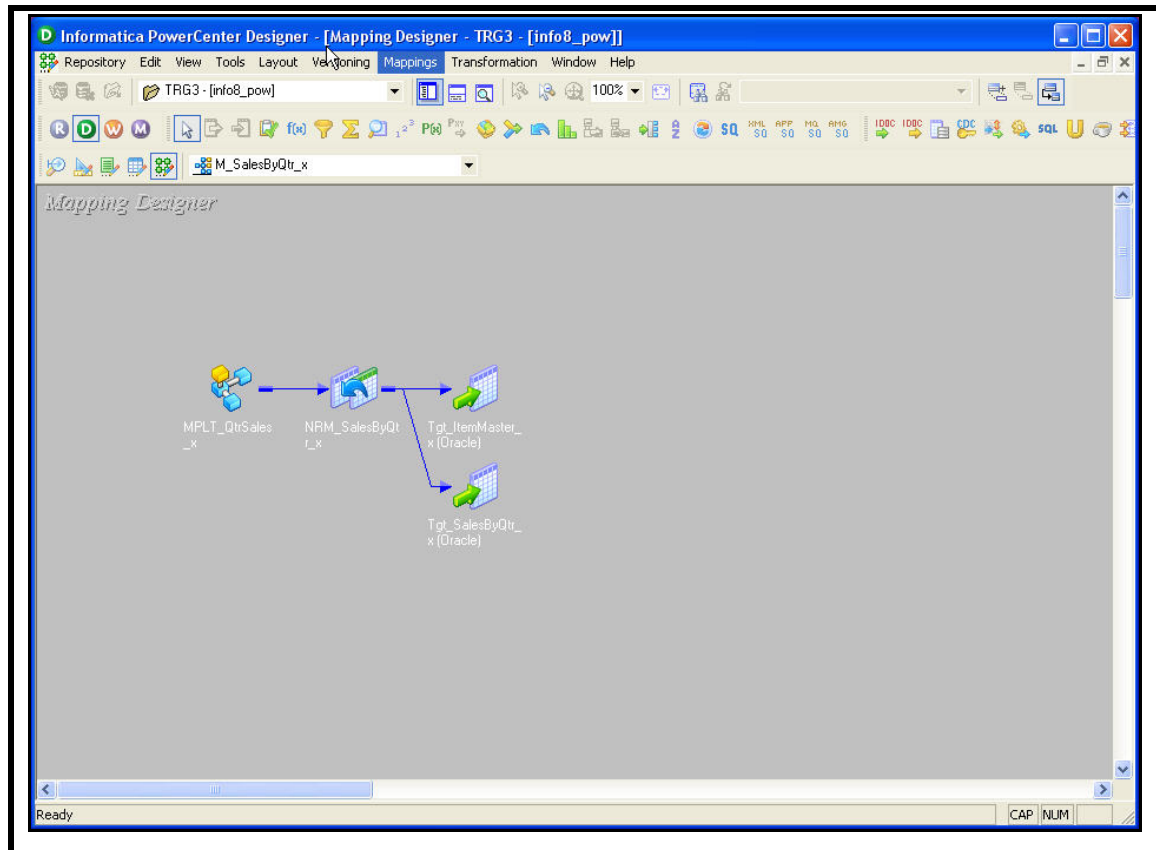
The Sales Department wants to run queries on sales by quarter. The Data Mart server has a high CPU usage, so you would like to pre-calculate these amounts and put them in summary tables.

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 | 1313 | Regulator System | 1998 |
| 2 | 1314 | Second Stage Regulator | 1998 |
| 3 | 1330 | Alternate Inflation Unit | 1998 |
| 4 | 1390 | 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 |

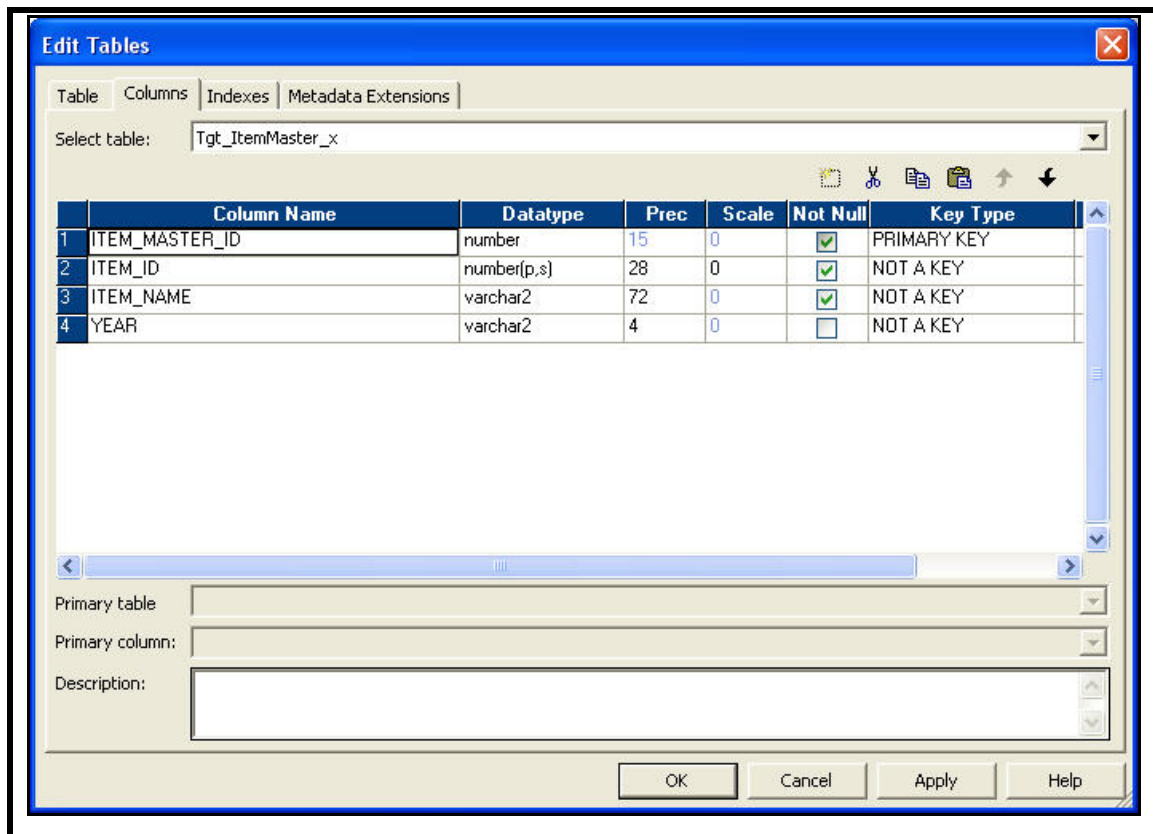
ITEM_MASTER_ID QUARTER QUARTERLY_SALES

| | | |
|----|---|-------|
| 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 |

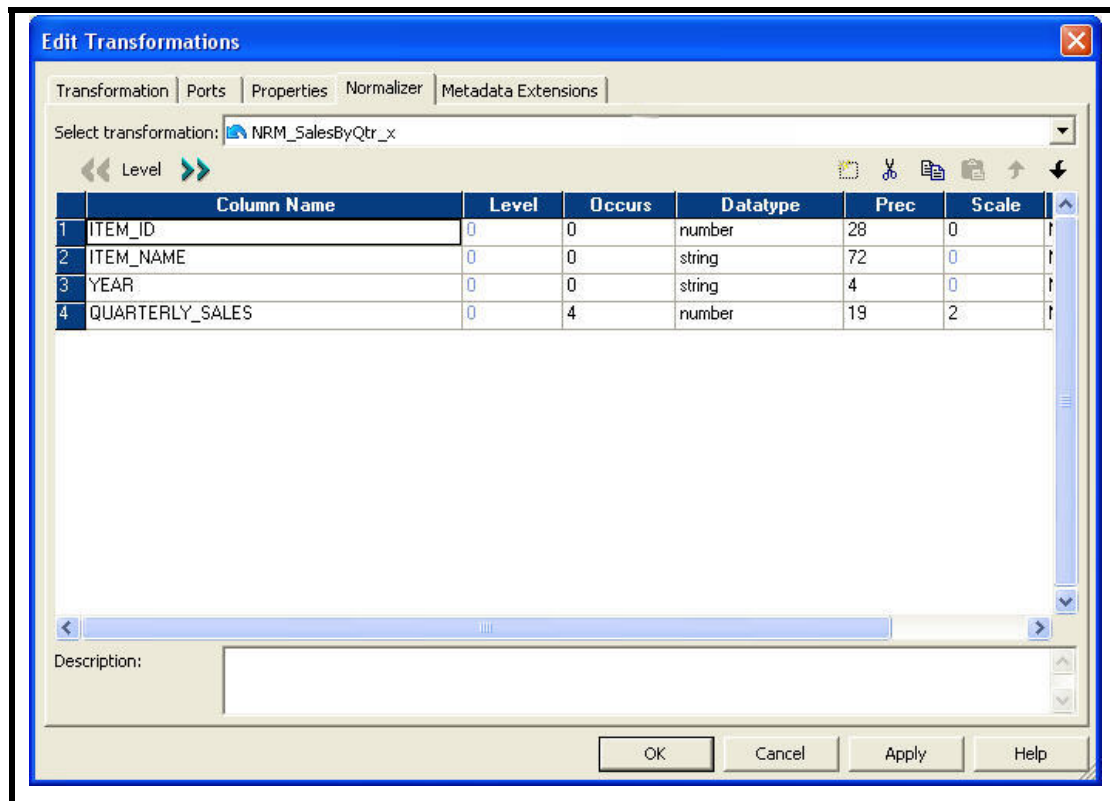
124 rows selected.

Problem Solution

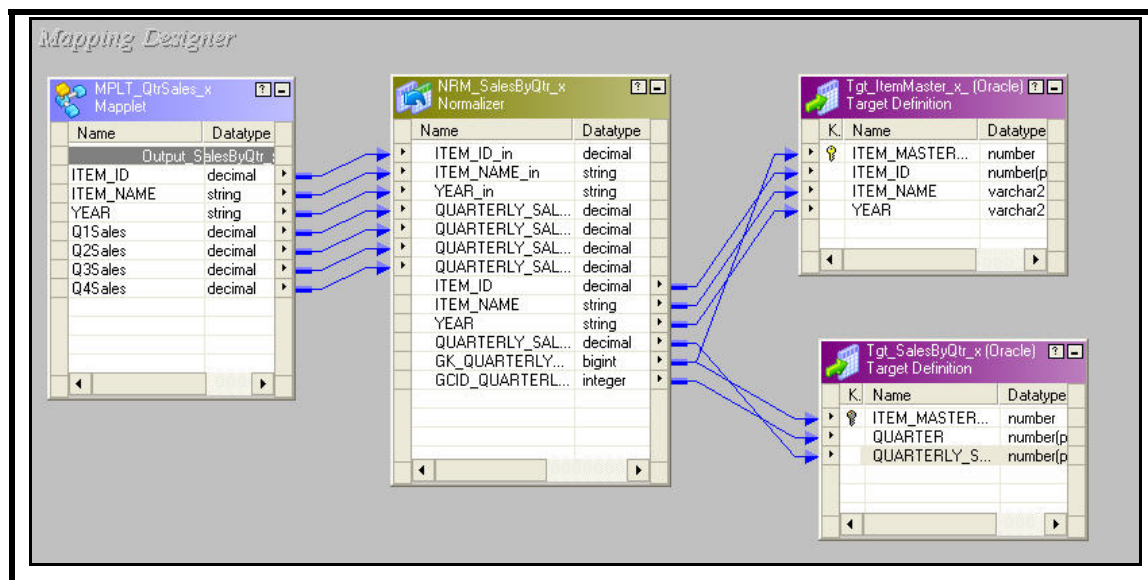
1. Create a new mapping by the name M_SalesByQtr_x.
2. Drag the MPLT_QtrSales_x Mapplet from the Navigator Window into your workspace.
3. Create the target table definition Tgt_ItemMaster_x table. The structure should be as shown below:



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 7-3 Annual Sales Mapping

Goals

- Use a mapplet
- Create a Rank Transformation

Time

60 minutes

Lab Setup

Quarterly sales mapplet created in the Lab 7-1

Background

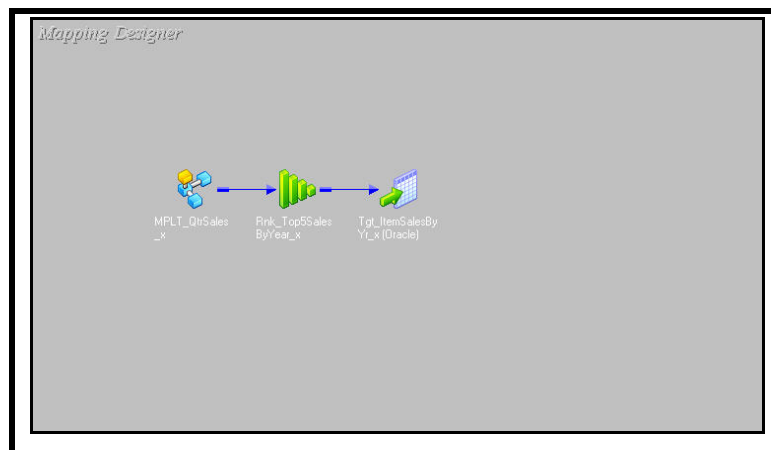
The Sales Department wants to run queries on sales by quarter. The Data Mart server has a high CPU usage, so you would like to pre-calculate these amounts and put them in summary tables.

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_SALESBYR_X | Target | Top 5 selling items for each year |

Mapping Layout



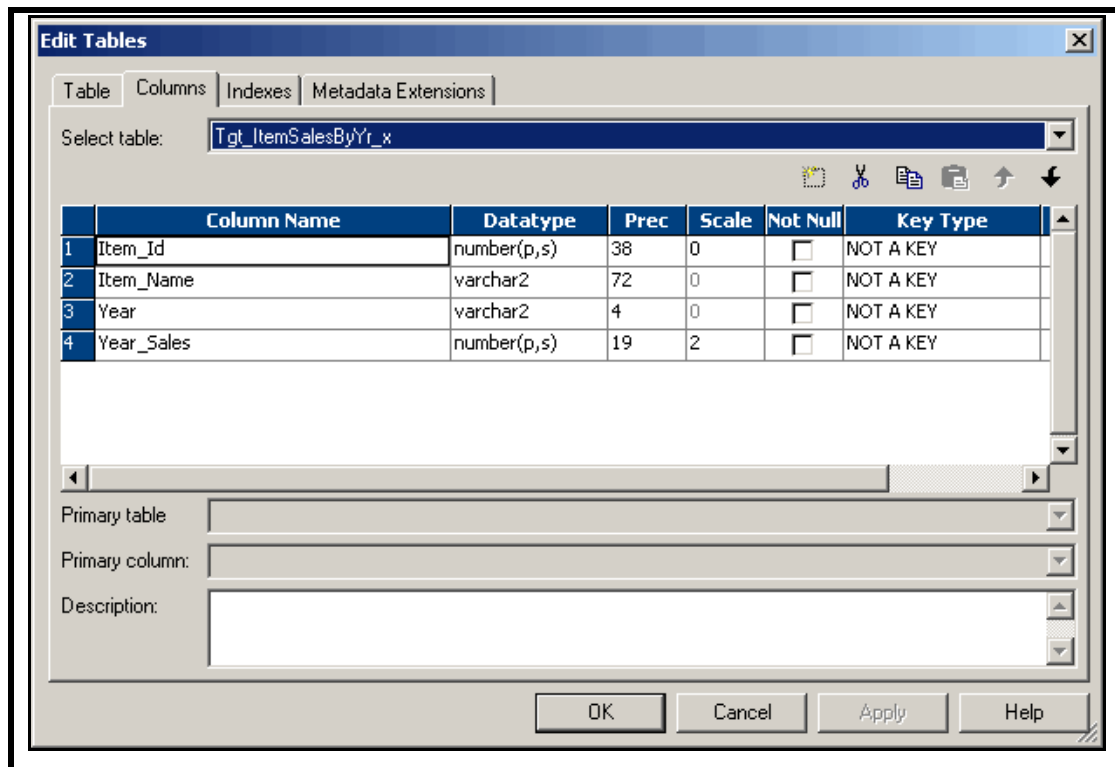
Final Output

```
SQL> select * from tgt_itemsalesbyyr_x;
```

| ITEM_ID | ITEM_NAME | YEAR | YEAR_SALES |
|---------|---------------------------|------|------------|
| 1314 | Second Stage Regulator | 1998 | 4327.98 |
| 2314 | Electronic Console | 1998 | 3078.5 |
| 1313 | Regulator System | 1998 | 2937.91 |
| 3316 | Stabilizing Vest | 1998 | 2549 |
| 3340 | Trim Fit Stabilizing Vest | 1998 | 1954 |

Problem Solution

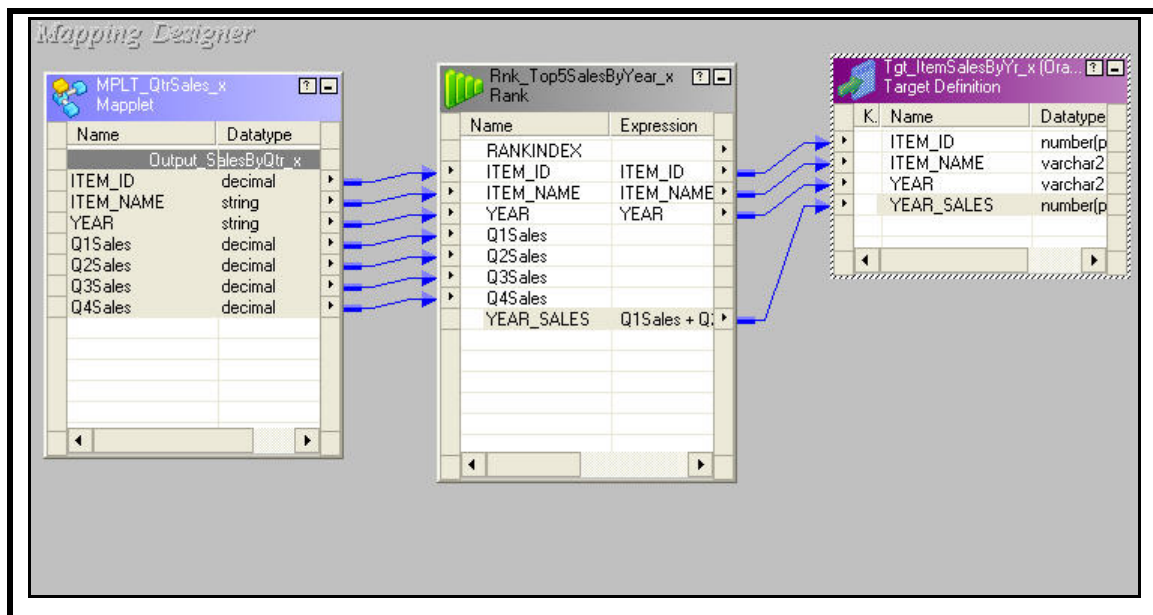
1. Create a new mapping by the name M_Top5SalesByYear_x.
2. Drag the MPLT_QtrSales_x Mapplet from the Navigator Window into your workspace
3. Create the target table definition Tgt_ItemSalesByYr_x and create the physical table in the database. The structure should be as shown below:



4. Create a Rank Transformation and name it Rnk_Top5SalesByYear_x.
5. Connect the ports from MPLT_QtrSales_x (mapplet) 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 8-1 Updating Current Items

Goals

- Configure an Unconnected Lookup Transformation
- Configure an Update Strategy transformation
- Understand Tracing levels in session logs

Time

120 Minutes

Lab Setup

Successful connection to the repository using PowerCenter Designer and Workflow Manager

Background

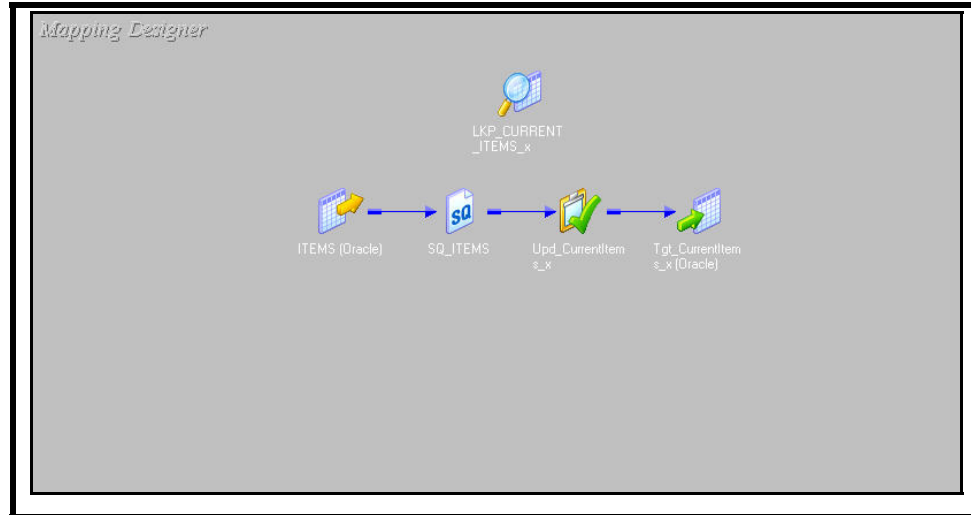
The operational source system that supplies data to your data mart tracks all items that your company has ever sold, even if they have since been discontinued. Your Sales Department wants to run queries against a Data Mart table that contains only currently selling items. They don't want to use views 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

```
SQL> select * from tgt_currentitems_x;
```

| 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 |
| 1390 | 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 | 390 | 290 | 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 | 2010 |
| 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 9-1 Listing Order Details

Goals

- Create a mapping that uses a Sorter Transformation

Time

30 minutes

Lab Setup

A connection to the repository using PowerCenter Designer and Workflow Manager

Background

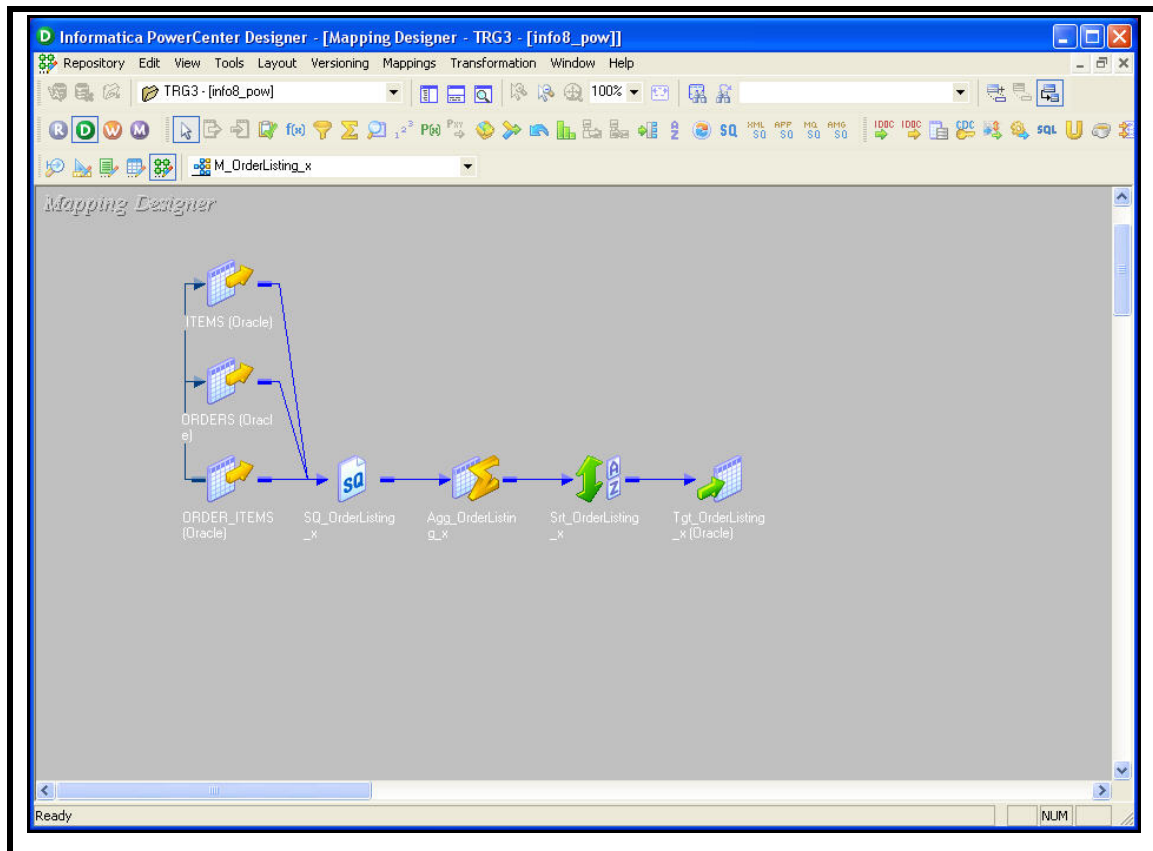
Company requires a report, which will show all order details in descending order of order amount.

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, QUALNTITY, 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

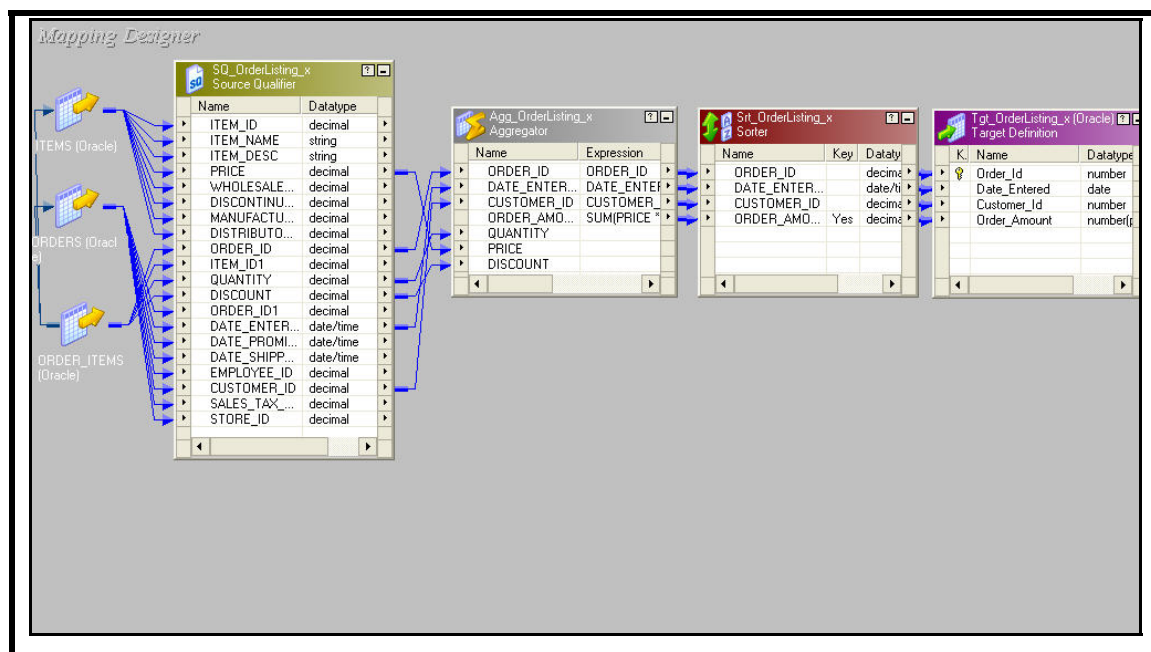
```
SQL> select * from tgt_orderlisting_x;
```

| ORDER_ID | DATE_ENTE | 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 | 1209 |
| 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:
$$\text{SUM}(\text{PRICE} * \text{QTY} - \text{DISCOUNT}).$$
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 10-1 Router Transformation

Goals

- Create a mapping that uses a Router Transformation

Time

60 mins

Lab Setup

A connection to the repository using PowerCenter Designer and Workflow Manager

Background

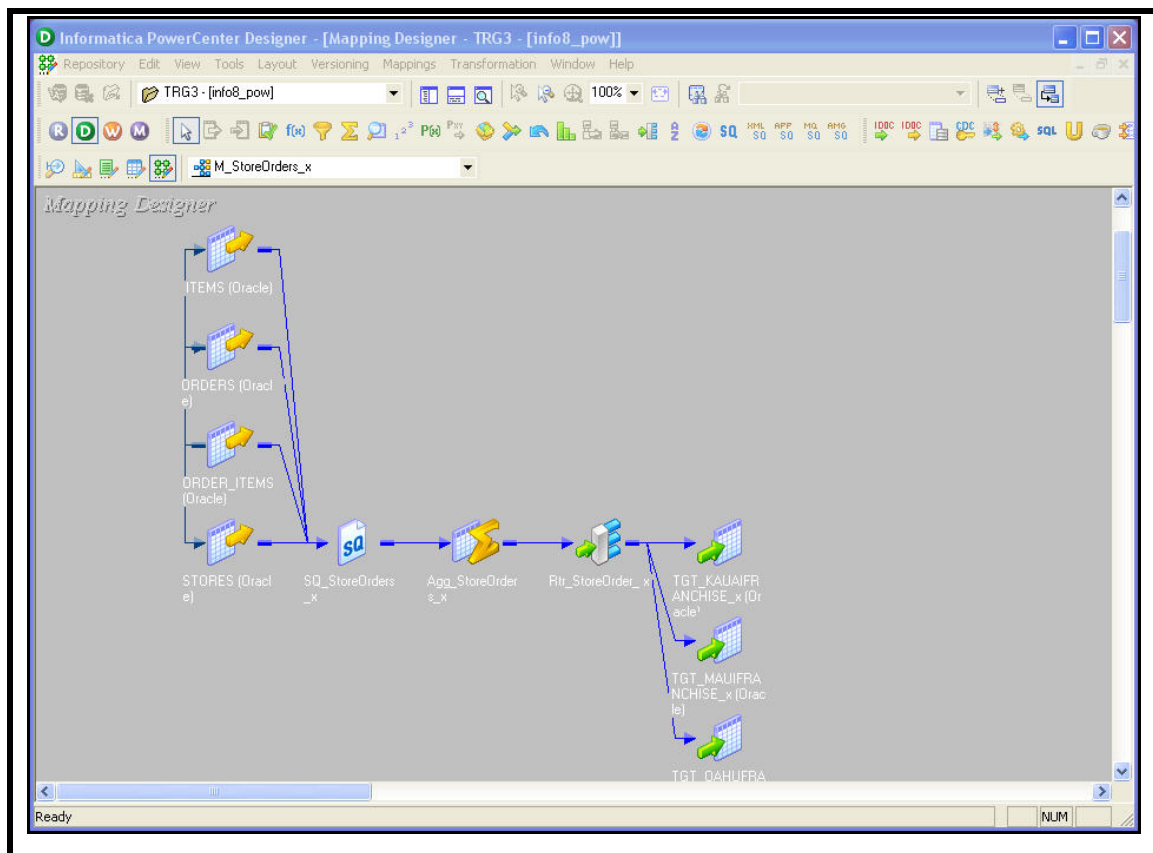
The company requires Store wise order details.

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 | 290 |
| 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_oahufranchise_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:

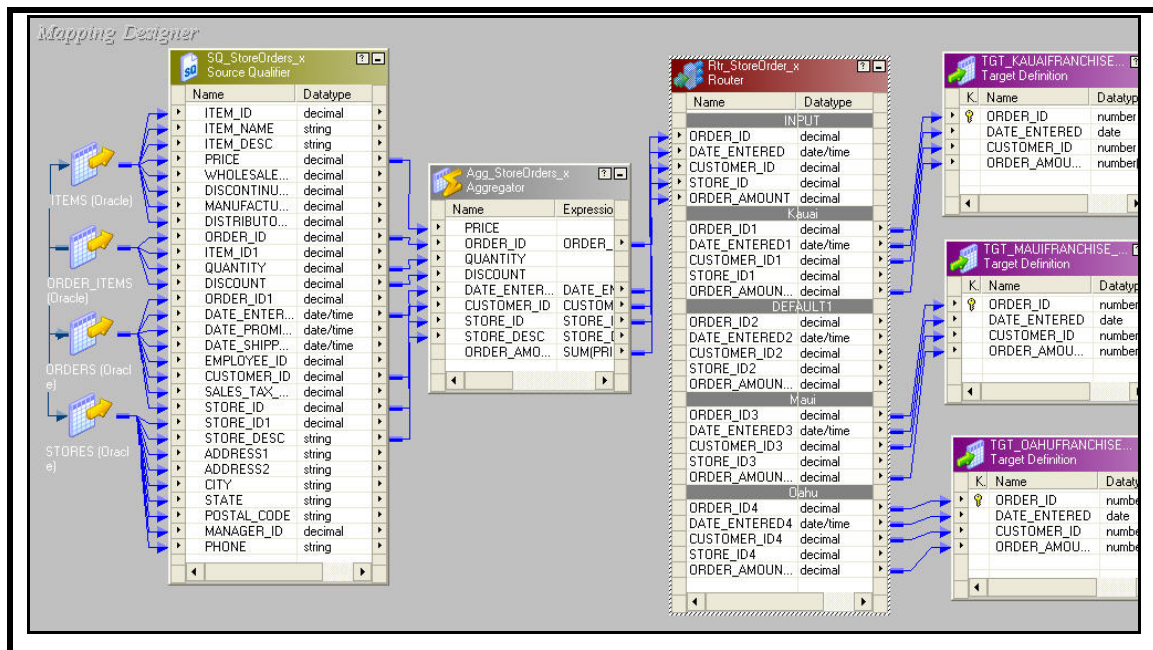
| | Column Name | Datatype | Prec | Scale | Not ... | Key Type |
|---|--------------|-------------|------|-------|-------------------------------------|-------------|
| 1 | ORDER_ID | number | 15 | 0 | <input checked="" type="checkbox"/> | PRIMARY KEY |
| 2 | DATE_ENTERED | date | 19 | 0 | <input checked="" 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:

| 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_KAUAI FRANCHISE_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 11-1 Sequence Generator Transformation

| | |
|------------------|--|
| Goals | <ul style="list-style-type: none">• Create a mapping that uses a Sequence Generator Transformation |
| Time | 60 minutes |
| Lab Setup | A connection to the repository using PowerCenter Designer and Workflow Manager |

Background

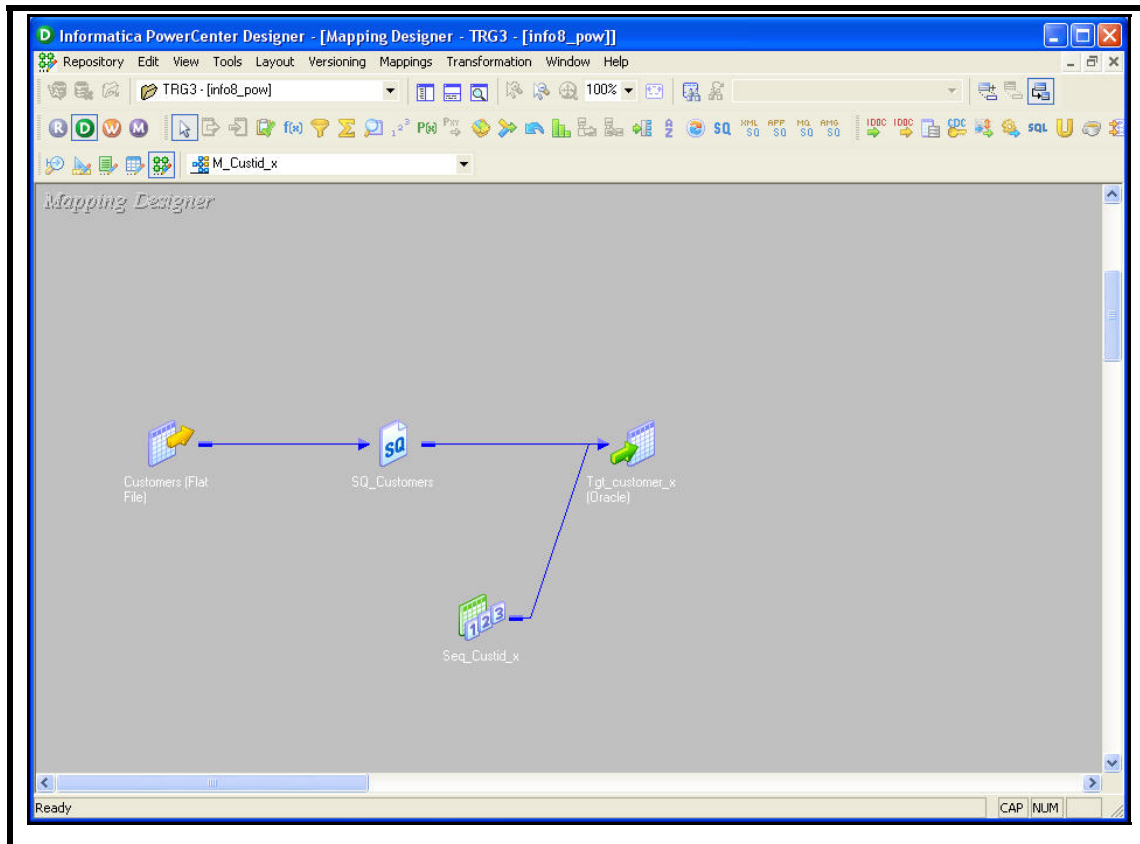
Customer source data arrives at each store in a flat file. Each file contains the customer name and other customer details. However, there is no unique id to identify each customer. The unique id for each customer will be generated through the mapping.

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

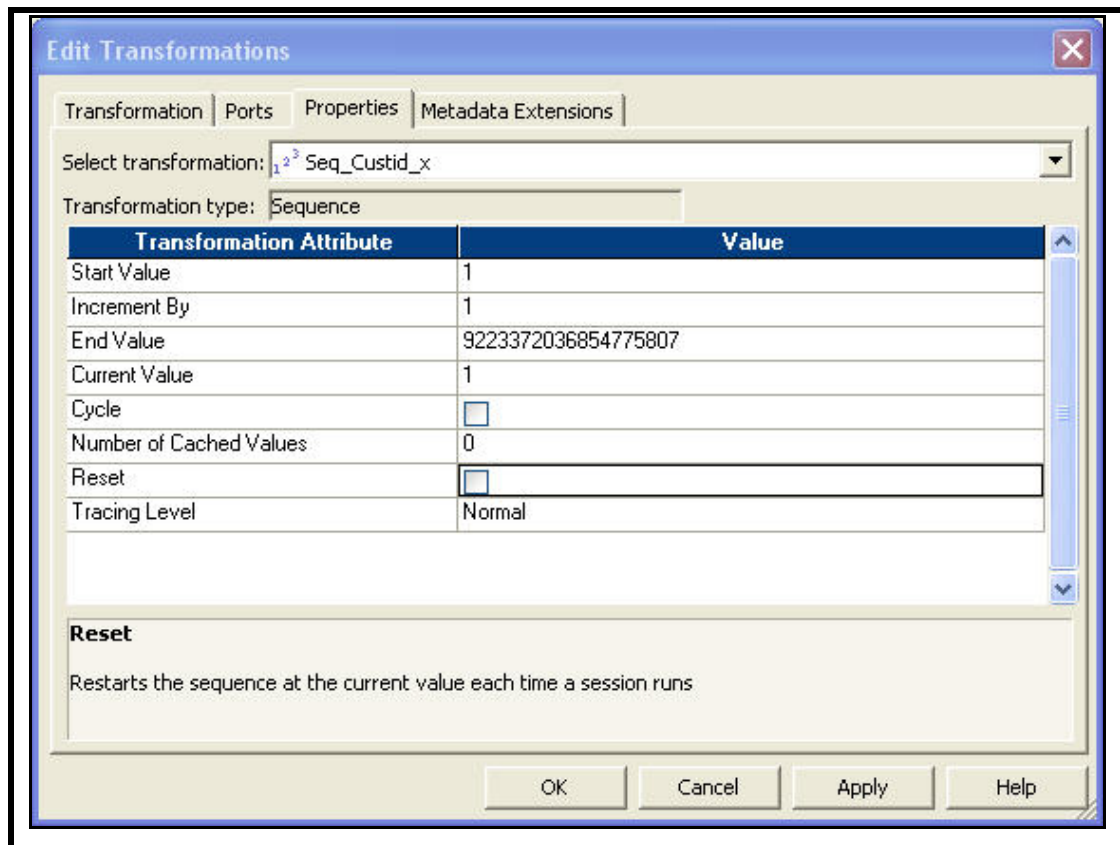
```
SQL> select * from tgt_customer_x;
```

| 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 |

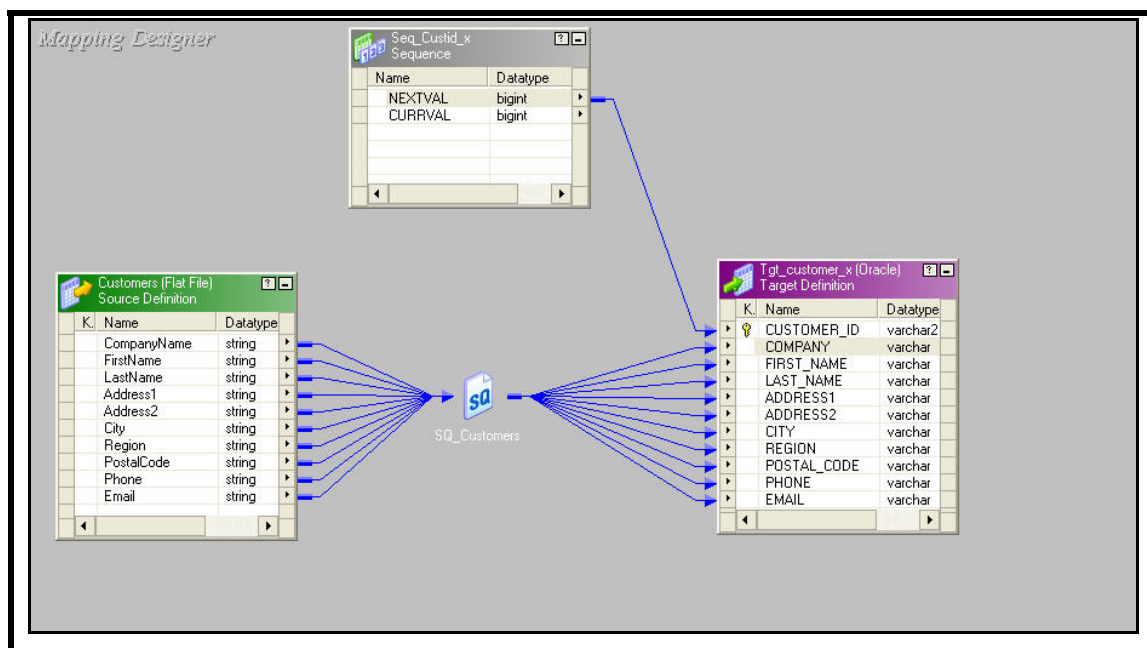
11 rows selected.

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 12-1 Stored Procedure Transformation

Goals

- Create a mapping that uses a Connected and Unconnected Stored Procedure Transformation

Time

60 mins

Lab Setup

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.

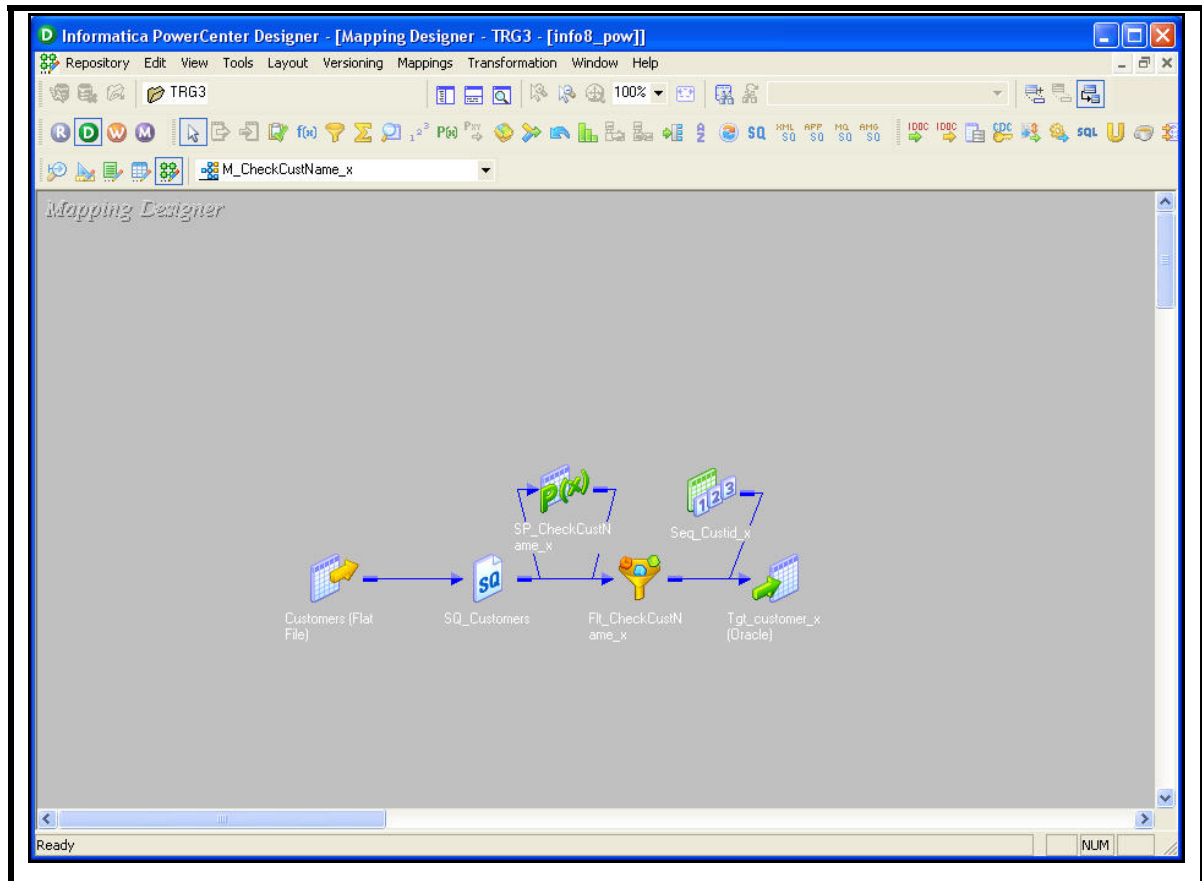
Solution

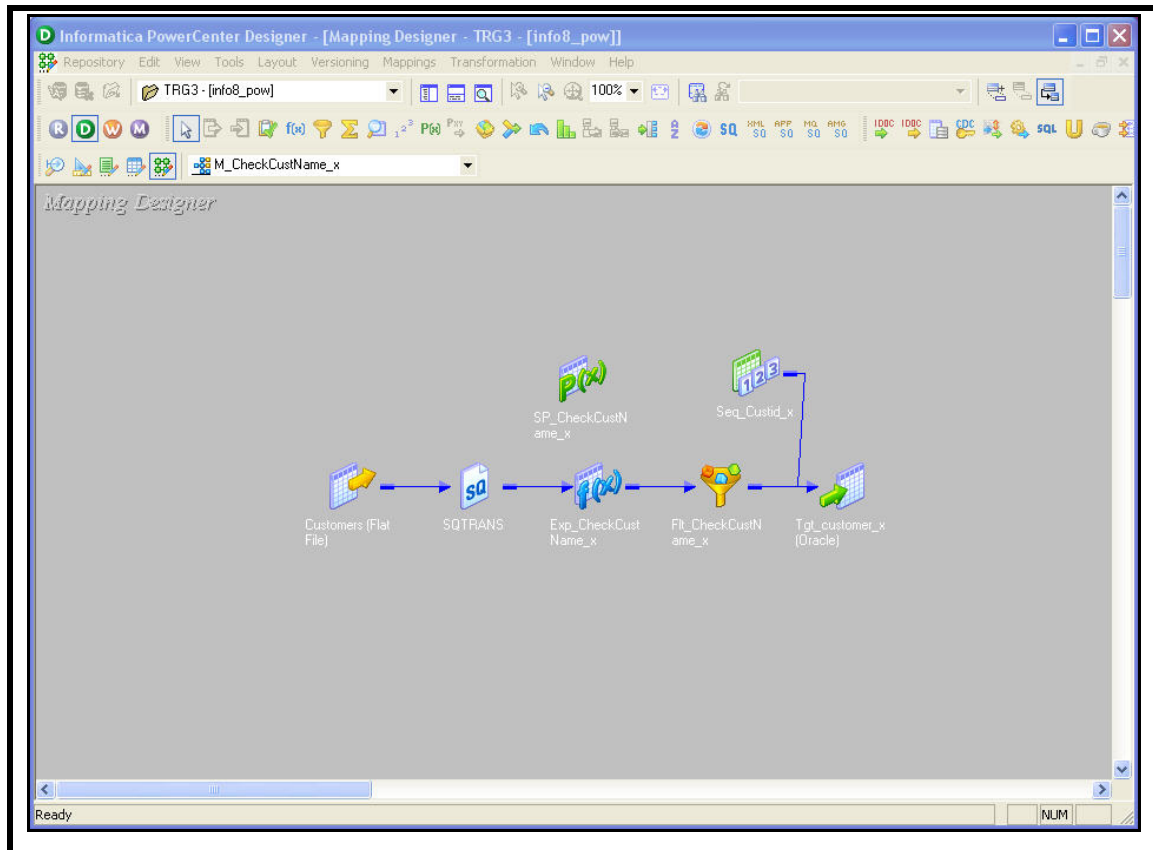
- Use a Connected Stored Procedure transformation to validate the customer name
- The customer name is passed as a parameter to the Stored Procedure
- The Stored Procedure returns a 'V' value for valid names and 'I' for invalid names

| 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;
```

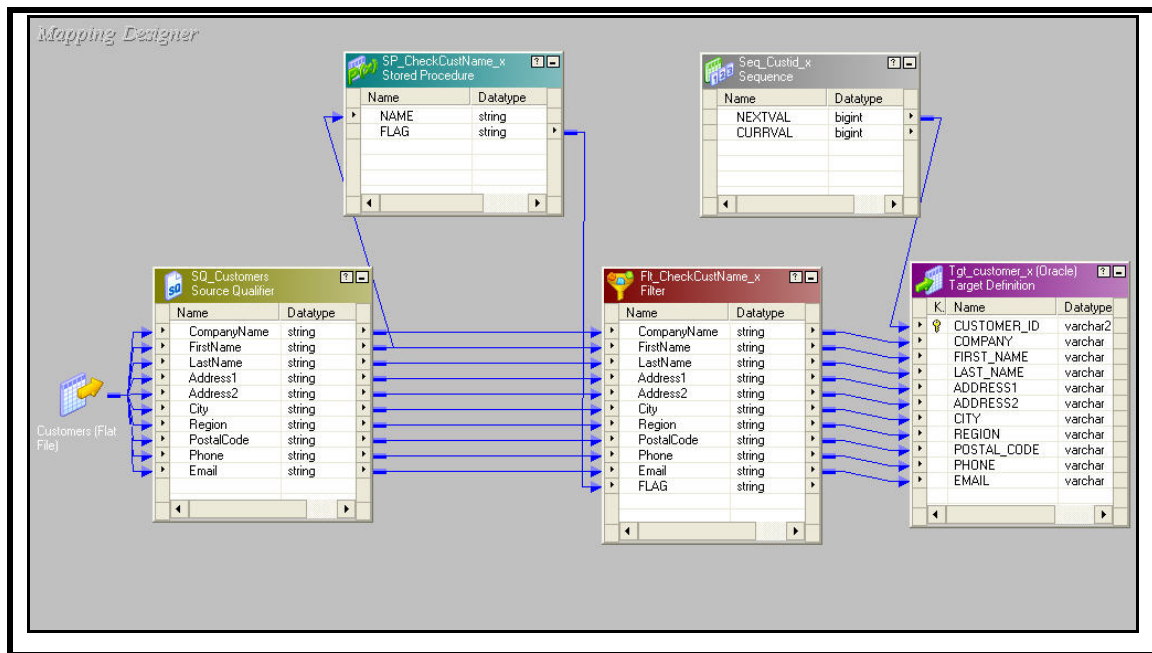
| CUSTOMER_ID | COMPANY | FIRST_NAME |
|-------------|------------------------------------|------------|
| 1 | Ana Trujillo Emparedados y helados | Ana |
| 2 | Antonio Moreno Taquería | Antonio |
| 3 | Around the Horn | Thomas |
| 4 | Berglunds snabbköp | Christina |
| 5 | Blauer See Delikatessen | Hanna |
| 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

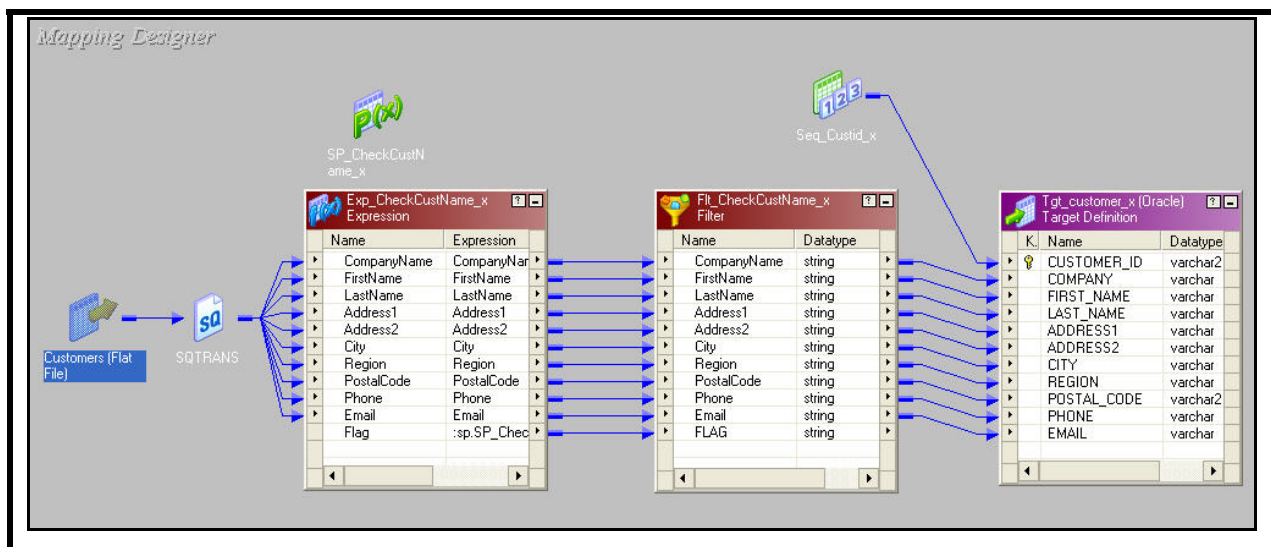
1. Copy the mapping M_Custid_x and rename it as M_CheckCustName_x.
2. Create a connected Stored Procedure transformation and name it as SP_CheckCustName_x.
3. Select the procedure name from the PROCEDURES folder.
4. The Stored Procedure transformation appears with two ports: Name and Flag.
5. Double 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:
:SP.SP_CheckCustName_x(FirstName, Proc_result).
5. 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 13-1 Configure an Email Task

Goals

- Configure a Workflow to send an email to designated recipients when the Integration service runs the workflow

Time

20 minutes

Lab Setup

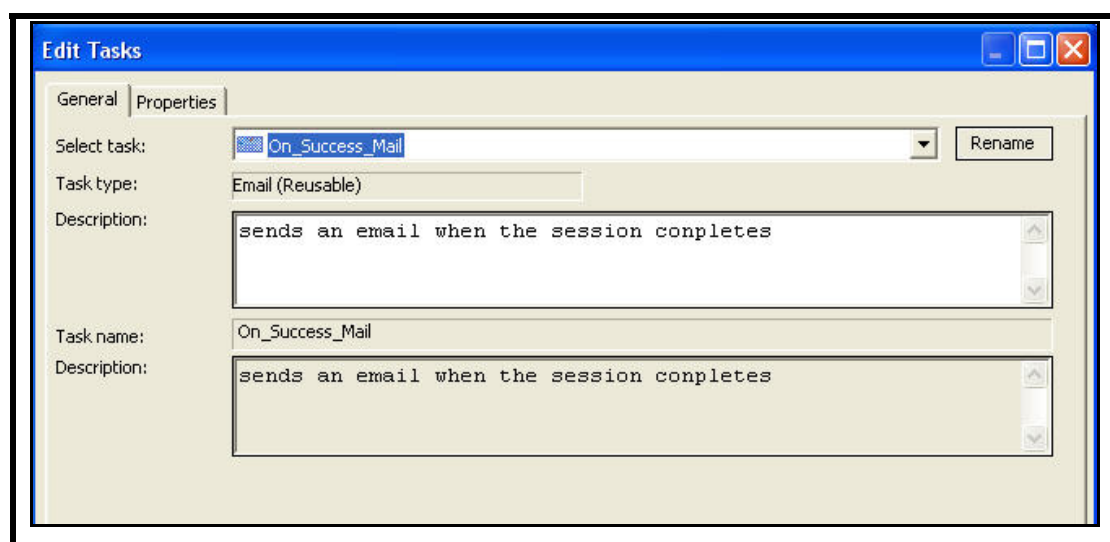
Configured Mail server and a valid Workflow

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.

Edit Tasks

General Properties

Select task: On_Success_Mail


Task type: Email

| Attribute | Value |
|-----------------|--------------------|
| Email User Name | studentx@patni.com |
| Email Subject | Session Details |
| Email Text | |

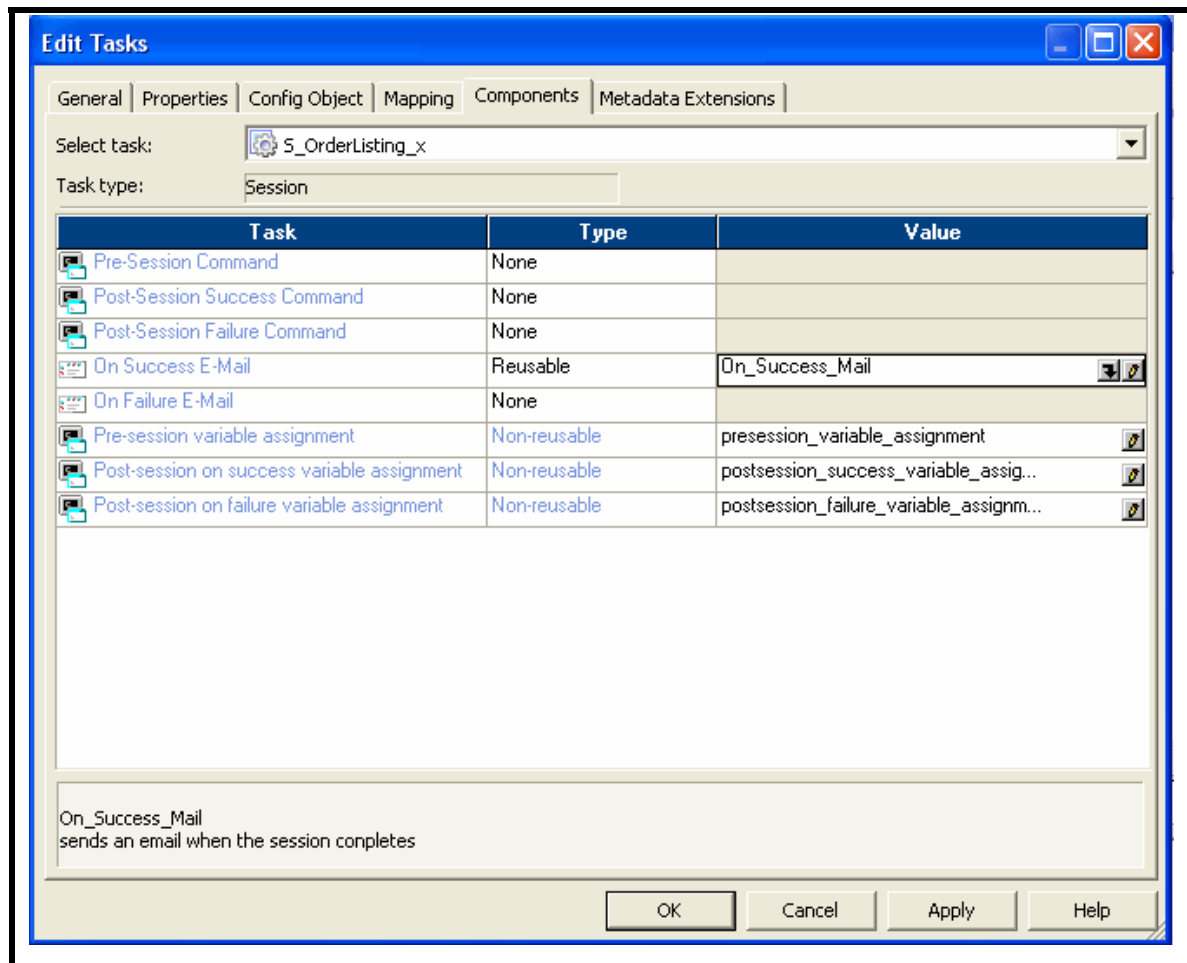
Email Subject

Email subject

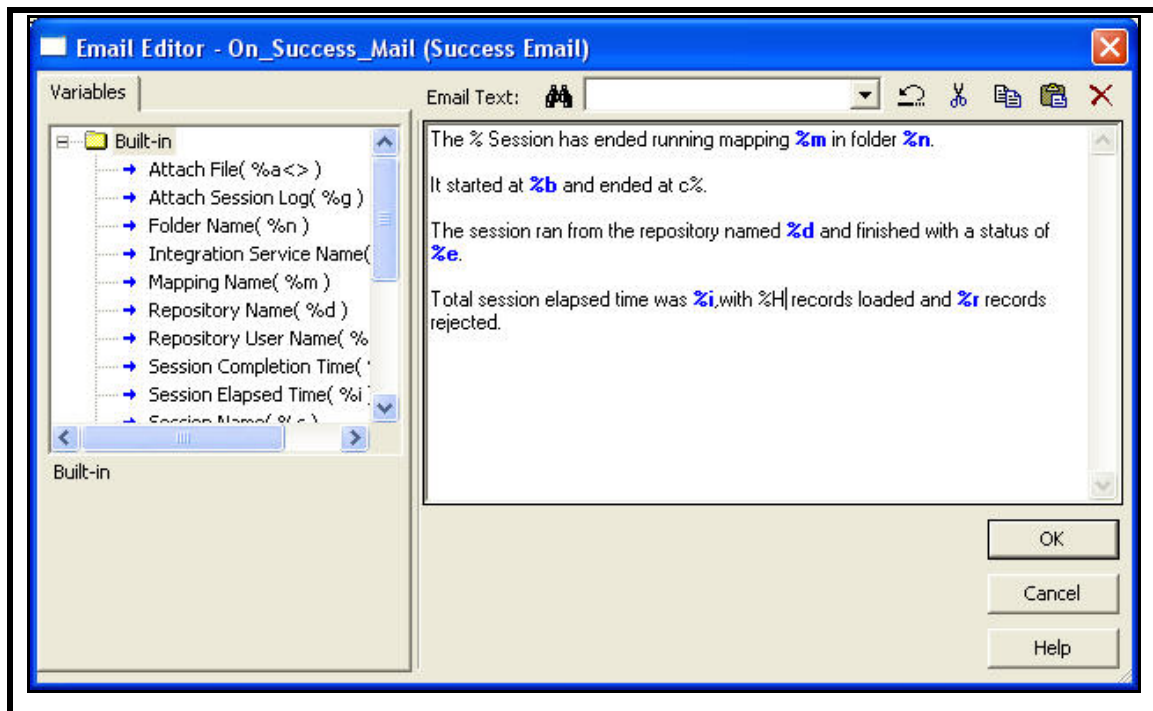
OK Cancel Apply Help

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.

- Click on the icon (shown highlighted in the figure).



- 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 14-1 Configure a Command Task

Goal

- Configure a command task to delete reject files after successful completion of a Workflow

Time

10 minutes

Lab Setup

Successful connection to the repository using PowerCenter Workflow Manager

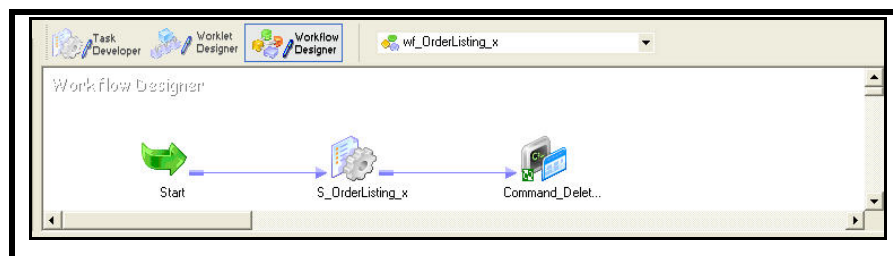
Background

Some reject files created during a Workflow run need to be deleted.

Solution

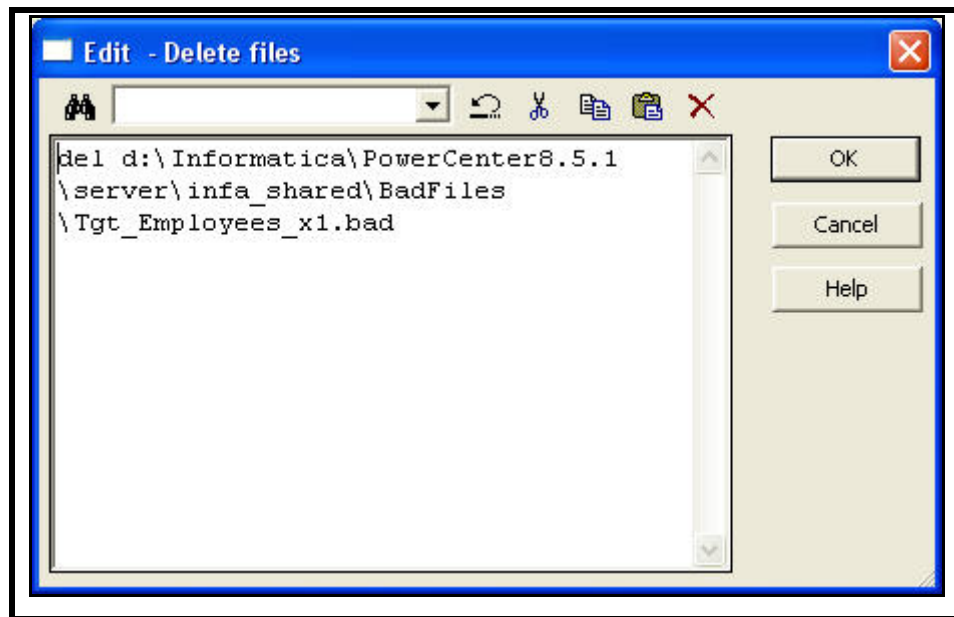
- A Command Task can be configured to specify shell or DOS commands, to delete reject files, copy a file, or archive target files.
- Use the Command Task to delete reject files.

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 15-1 Event-Wait task

| | |
|------------------|---|
| Goal | <ul style="list-style-type: none">• Create Event Wait Task and Event Raise task |
| Time | 20 minutes |
| Lab Setup | 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 16-1 Pre-SQL and Post-SQL

| | |
|------------------|--|
| Goal | <ul style="list-style-type: none">• Create Pre-SQL and Pre-SQL |
| Time | 30 minutes |
| Lab Setup | 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 PKSRCDNO.
3. In the Pre-SQL Create an index PKSRCDNO 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 17-1 Multiple Source Files(Indirect File)

| | |
|------------------|---|
| Goal | <ul style="list-style-type: none">• Use Multiple source files |
| Time | 20 minutes |
| Lab Setup | 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 18-1 Mapping Parameter

| | |
|------------------|--|
| Goal | <ul style="list-style-type: none">• Create a Mapping Parameter |
| Time | 45 minutes |
| Lab Setup | 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.gD:\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 19-1 Mapping Variable

| | |
|------------------|---|
| Goal | <ul style="list-style-type: none">• Create a Mapping Variable |
| Time | 45 minutes |
| Lab Setup | 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 20-1 FTP Connection

| | |
|-----------|--|
| Goal | <ul style="list-style-type: none">• Use an FTP connection |
| Time | 20 minutes |
| Lab Setup | 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).
Note: The above details may vary according to the FTP connection.
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 the source(FTP – Linux server).

Note: Ensure the the FTP service is started on the FTP server(Linux server in this case).

Lab 21-1 Dynamic Lookup

| | |
|------------------|--|
| Goal | <ul style="list-style-type: none">• Create a Dynamic Lookup |
| Time | 45 minutes |
| Lab Setup | 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 22-1 Using Mapping Wizard for SCD Type1

| | |
|------------------|--|
| Goal | <ul style="list-style-type: none"> Create a Type 1 mapping using Wizard |
| Time | 45 minutes |
| Lab Setup | Successful connection to the repository using PowerCenter Workflow Manager |

I. Create source definition

- 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);
```
- Use the flat file customers from the **Srcfiles** folder to populate data in the Customers relational table.
- Create a Mapping to perform the above mentioned task.
- Create a Workflow, task to execute the table.
- The Customers table contains the following data:

| CUSTOMERID | COMPANYNAME | FIRSTNAME | LASTNAME | ADDRESS | CITY |
|------------|------------------------------------|------------|----------|-------------------------------|------------|
| 101 | Alfreds Futterkiste | Mar\$ia | 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 snabbkop | 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 |

- 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:

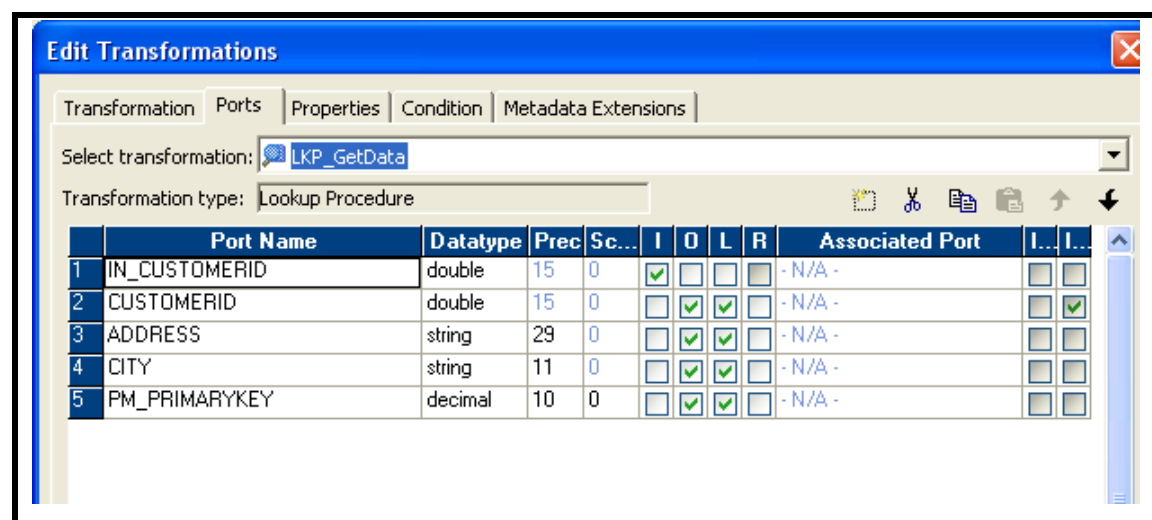
1. Selects all rows.
2. Caches the existing target as a lookup table.
3. Compares logical key columns in the source against corresponding columns in the target lookup table.
4. Compares source columns against corresponding target columns if key columns match.

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

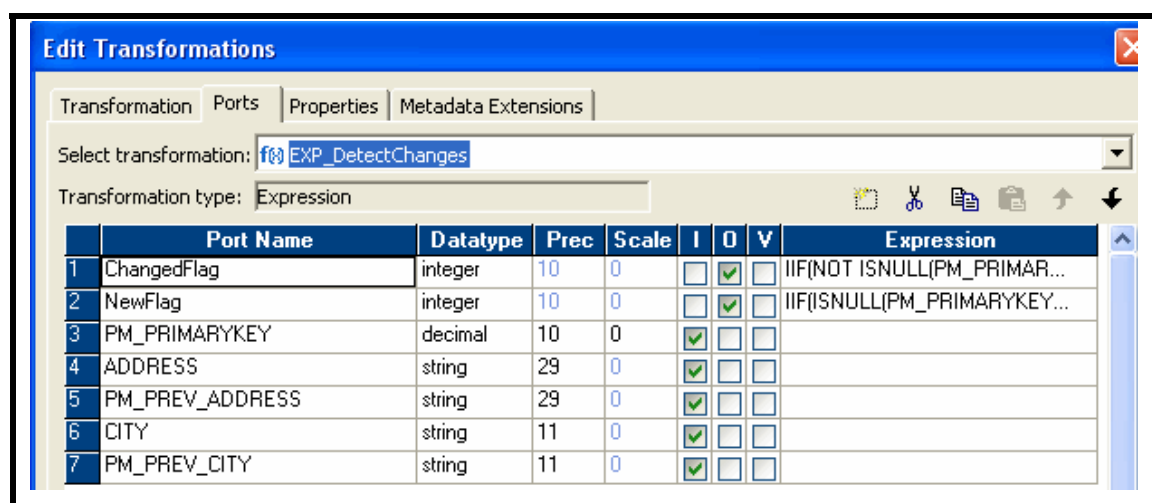
Note: The Type 1 Dimension mapping uses a Lookup and an Expression transformation to compare source data against existing target data. The LookUp table is the target table T_customers.

V. LookUp Transformation

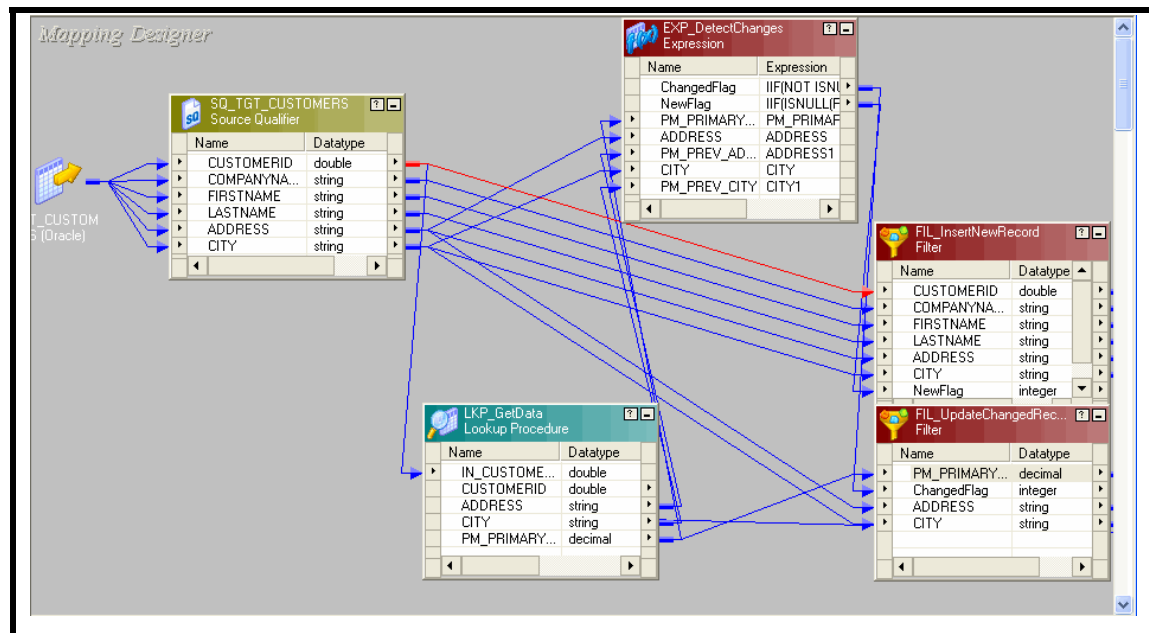
1. The IN_CUSTOMERID field is the field from the source table which is taken from the SQ_Customers.



VI. Expression transformation



1. The ADDRESS and CITY fields are connected from the SQ_Customers, the PM_PREV_ADDRESS, PM_PREV_CITY, PM_PRIMARYKEY are connected from the Lookup transformation.



2. 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)
3. The value of NewFlag is computed as :
`IIF(ISNULL(PM_PRIMARYKEY),TRUE,FALSE)`

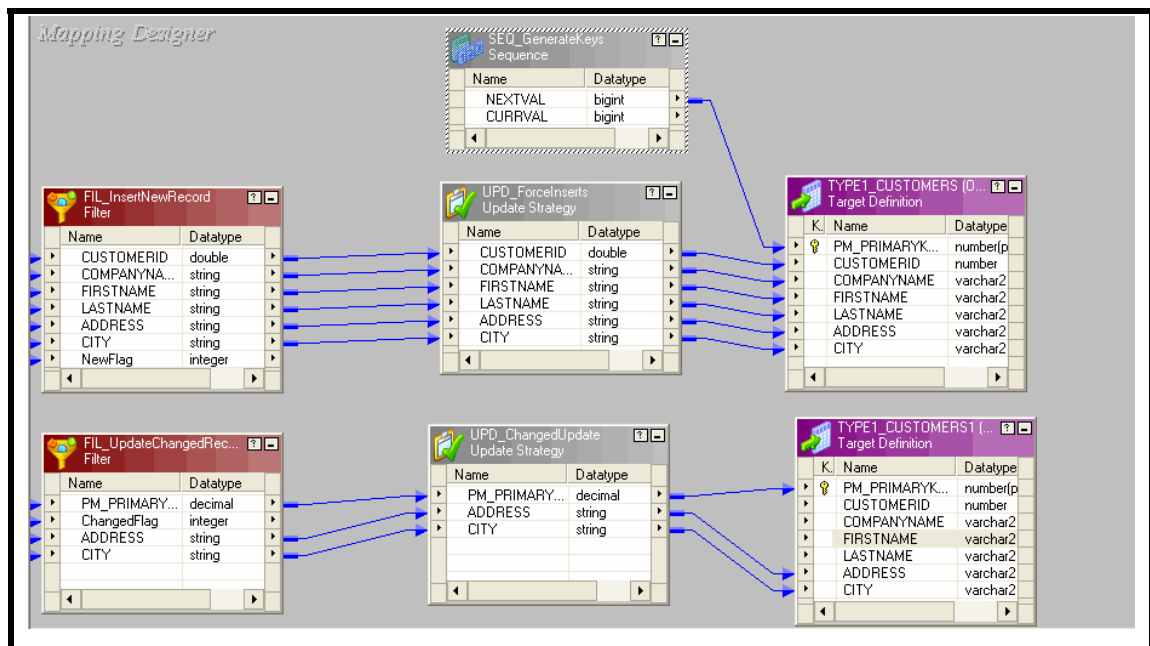
VII. Filter Transformation

1. 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.
2. 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.

- 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

- The UPD_ForceInserts has the Update Strategy Expression DD_INSERT and inserts row in the target table.
- The UPD_ChangedUpdate has the Update Strategy Expression DD_UPDATE and updates rows in the target table.



Lab 23-1 Using Mapping Wizard for SCD Type2

| | |
|------------------|--|
| Goal | <ul style="list-style-type: none"> Create a Type 2 mapping using Wizard |
| Time | 45 minutes |
| Lab Setup | 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 | María | 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 snabbkop | 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**.

The image shows the 'Mapping Wizard' dialog box, specifically the 'Target Field Selection' step. The title bar reads 'Mapping Wizard' with a close button. The main heading is 'Target Field Selection' with a subtitle: 'Choose the fields you would like to compare for change detection, and select the fields that represent the logical key.' Below this, the 'Target Table' is set to 'Tgt_cust_SCD_Type2'. There are two main sections: 'Target Table Fields' and 'Logical Key Fields'. The 'Target Table Fields' list contains 'COMPANYNAME', 'FIRSTNAME', and 'LASTNAME'. The 'Logical Key Fields' list contains 'CUSTOMERID'. Between these two lists are two pairs of buttons: 'Add >>' and '<< Remove'. Below the 'Logical Key Fields' list is a section titled 'Fields to compare for changes' which contains 'ADDRESS' and 'CITY'. At the bottom of the dialog are four buttons: '< Back', 'Next >' (which is highlighted with a mouse cursor), 'Cancel', and 'Help'.

Mapping Wizard

Target Field Selection
Choose the fields you would like to compare for change detection, and select the fields that represent the logical key.

Target Table: Tgt_cust_SCD_Type2

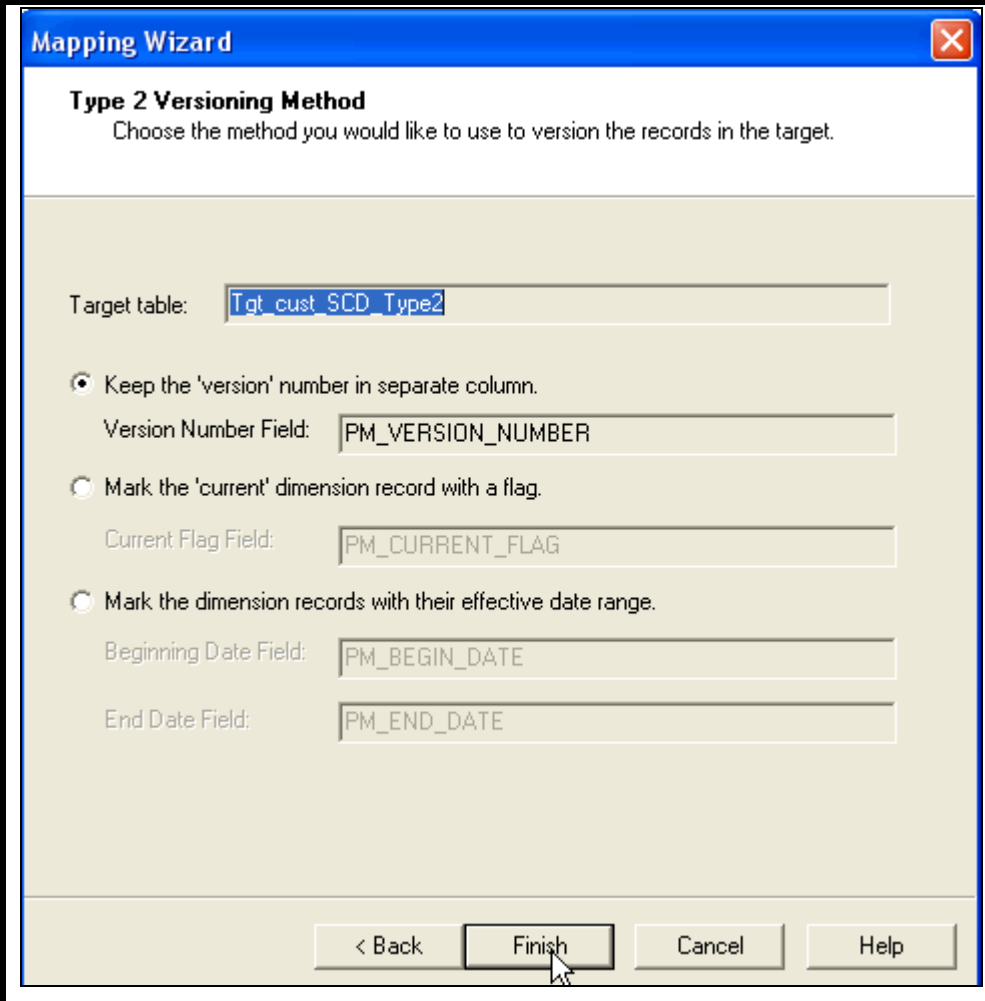
Target Table Fields:
COMPANYNAME
FIRSTNAME
LASTNAME

Logical Key Fields:
CUSTOMERID

Fields to compare for changes:
ADDRESS
CITY

< Back Next > Cancel Help

4. Select the Versioning Method. Click **Finish**.



The image shows a 'Mapping Wizard' dialog box with a blue title bar and a close button (X) in the top right corner. The main title is 'Type 2 Versioning Method' in bold. Below it, a subtitle reads 'Choose the method you would like to use to version the records in the target.' The dialog is divided into a light beige content area and a footer area with buttons. In the content area, the 'Target table:' label is followed by a text box containing 'Tgt_cust_SCD_Type2'. There are three radio button options: 1. 'Keep the 'version' number in separate column.' with a 'Version Number Field:' text box containing 'PM_VERSION_NUMBER'. 2. 'Mark the 'current' dimension record with a flag.' with a 'Current Flag Field:' text box containing 'PM_CURRENT_FLAG'. 3. 'Mark the dimension records with their effective date range.' with 'Beginning Date Field:' text box containing 'PM_BEGIN_DATE' and 'End Date Field:' text box containing 'PM_END_DATE'. The footer area contains four buttons: '< Back', 'Finish' (with a mouse cursor pointing at it), 'Cancel', and 'Help'.

Mapping Wizard

Type 2 Versioning Method
Choose the method you would like to use to version the records in the target.

Target table:

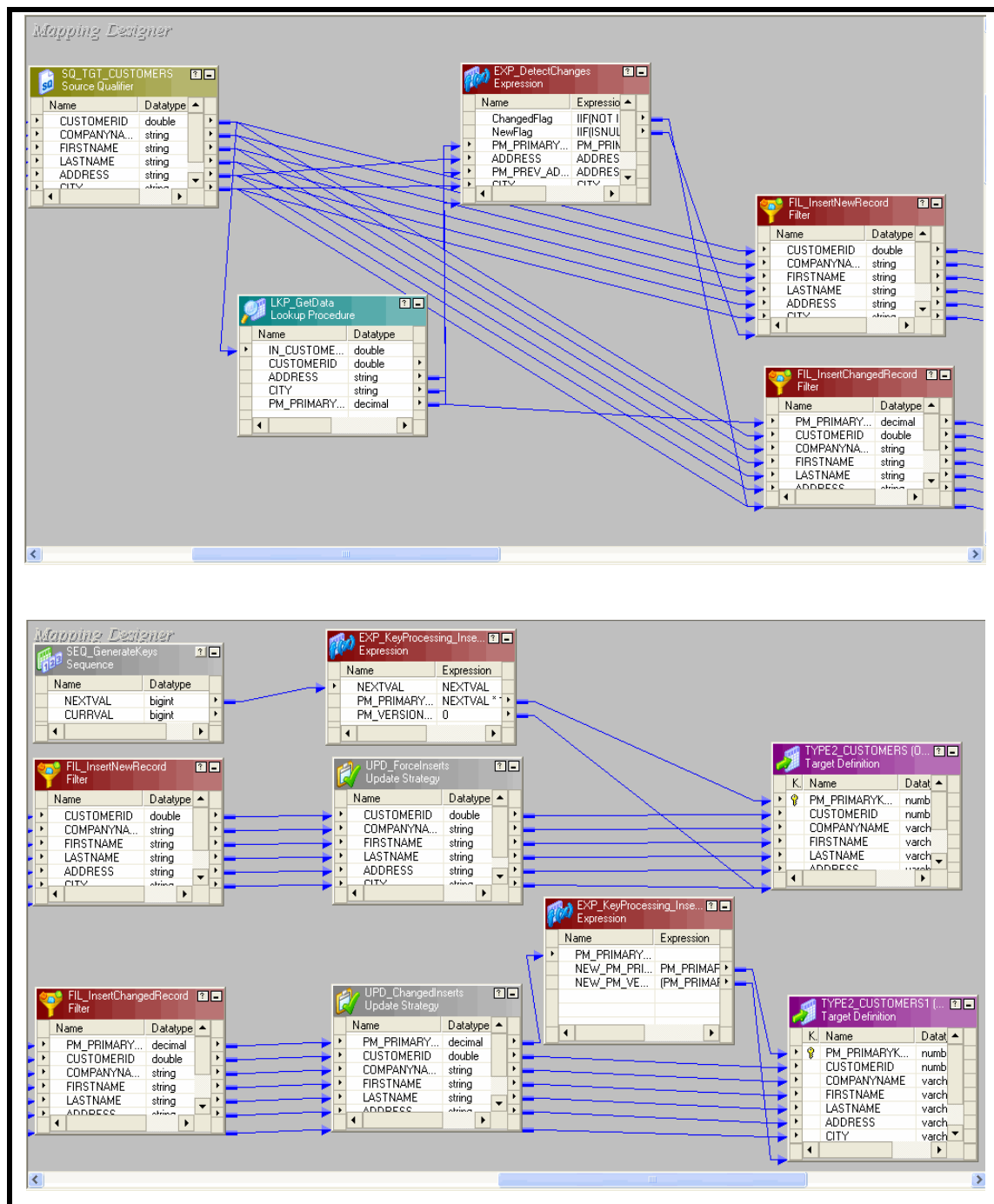
☒ Keep the 'version' number in separate column.
Version Number Field:

☐ Mark the 'current' dimension record with a flag.
Current Flag Field:

☐ Mark the dimension records with their effective date range.
Beginning Date Field:
End Date Field:

< Back **Finish** Cancel Help

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:

$(PM_PRIMARYKEY + 1) \% 1,000.$

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_ITEMS |
| DATE_ENTERED | Date | Datetime | Datetime | NOT NULL | |
| DATE_PROMISED | 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 Vi | 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. 57 | | 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. 57 | | Mannheim |
| Blondel père et fils | Frederique | Citeaux | 24, place | | Strasbourg |
| Bólido Comidas | Martin | Sommer | C/ Araquil, 67 | | Madrid |
| Bon app' | Laurence | Lebihan | 12, rue des | | Marseille |
| Bottom-Dollar | Elizabeth | Lincoln | 23 Tsawassen | | 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 | |