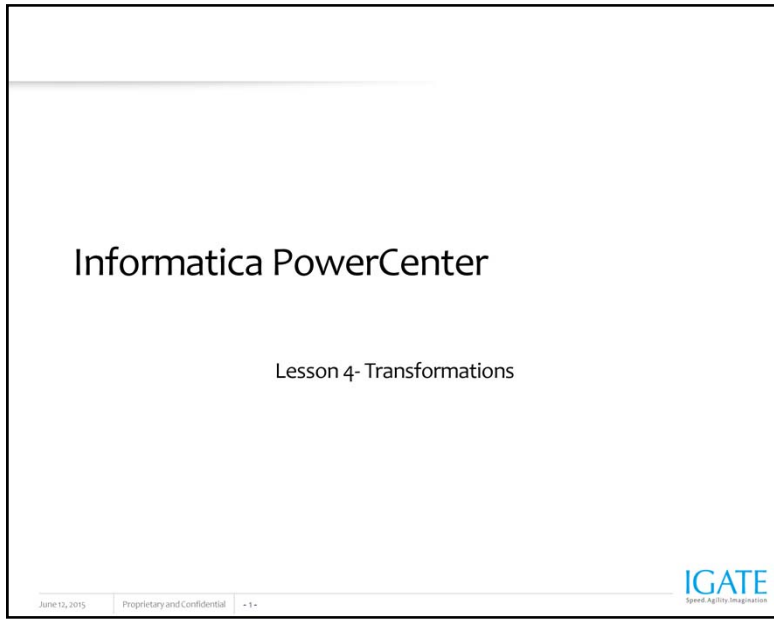

Informatica PowerCenter
Lesson 4- Transformations
June 12, 2015
Proprietary and Confidential
- 1 -
IGATE
Speed. Agility. Imagination

Lesson Objectives

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

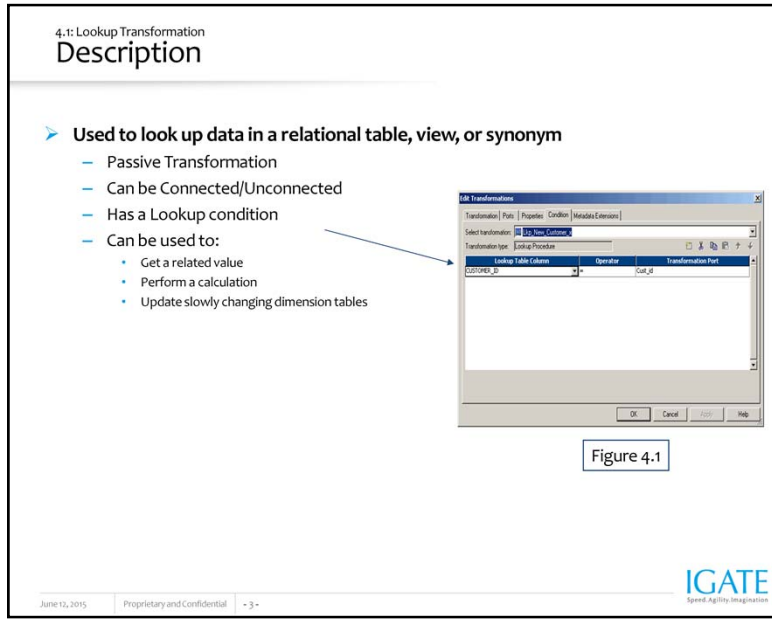


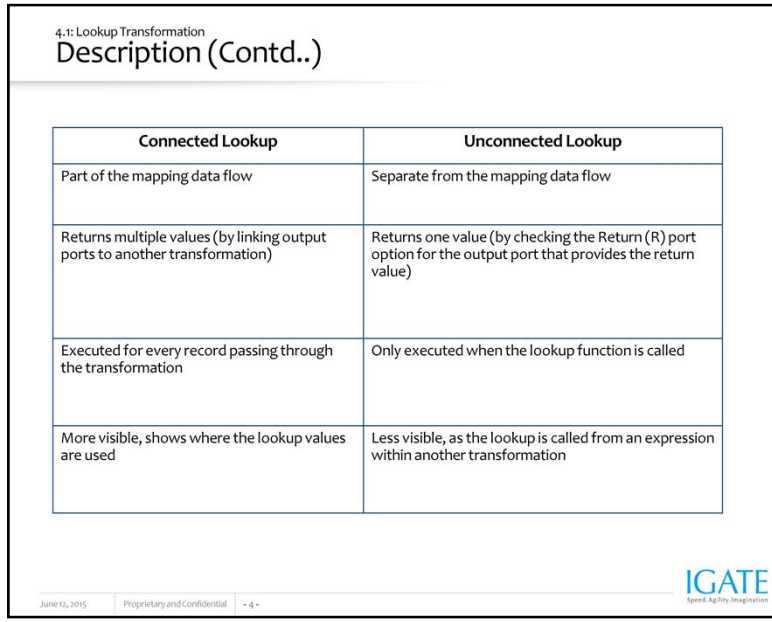
Figure 4.1

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

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

The Lookup Transformation can be used to:

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



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

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

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

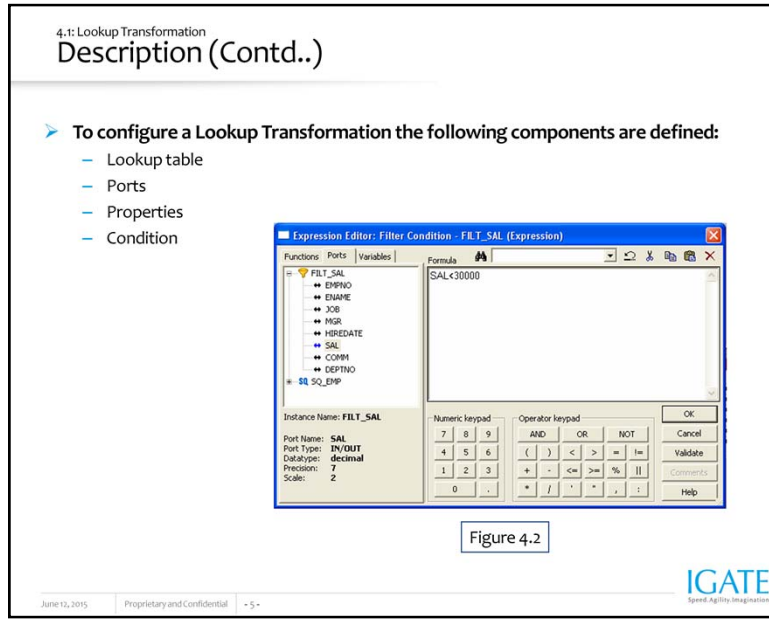
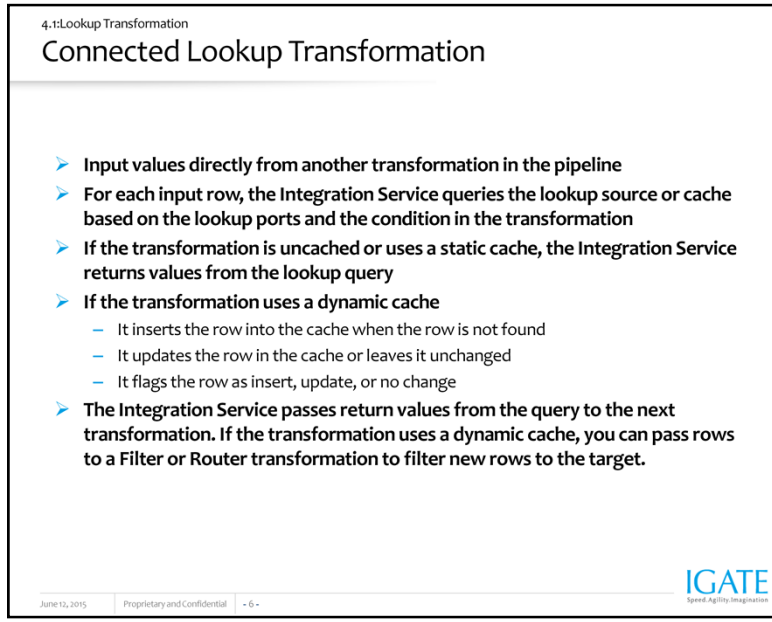


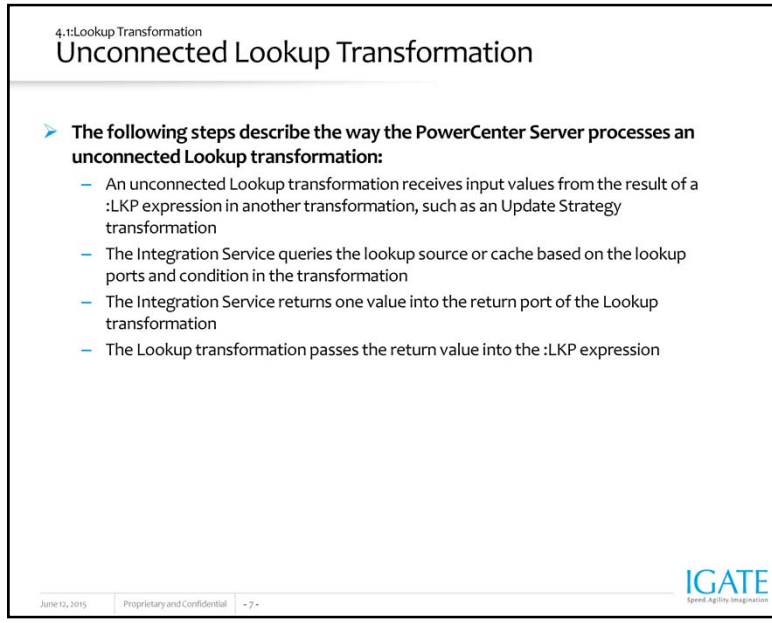
Figure 4.2

To configure a Lookup Transformation the following components are defined:

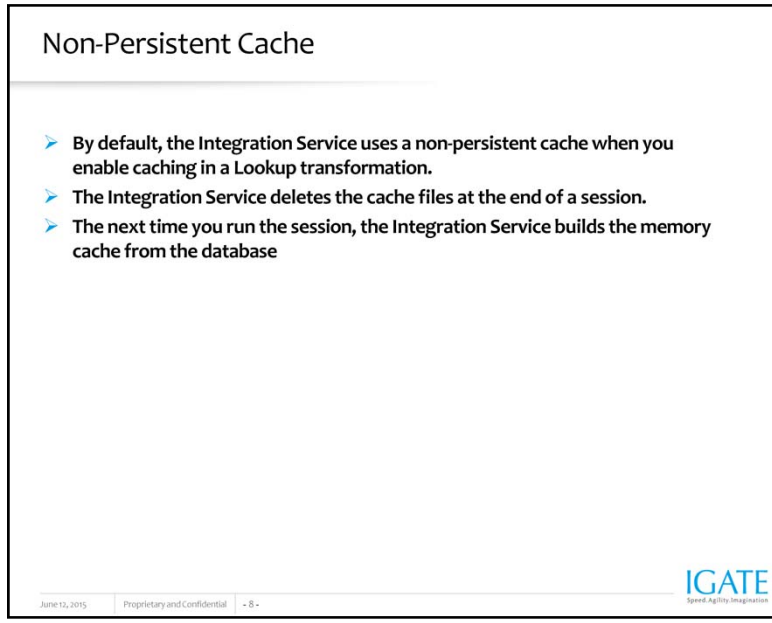
- **Lookup source -** Lookup source can be created from the following locations:
  - Relational or flat file source or target definition in repository
  - Source Qualifier definition in Mapping
  - Table or file that the Integration service and client can connect to
- **Lookup Ports -** The Ports tab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



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



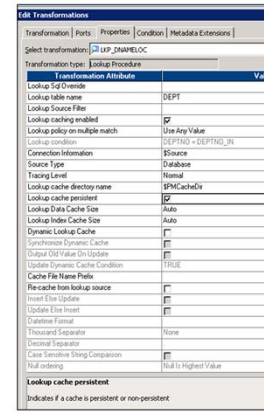
An unconnected Lookup transformation receives input values from the result of a :LKP expression in another transformation. You can call the Lookup transformation more than once in a mapping.
A common use for unconnected Lookup transformations is to update slowly changing dimension tables.



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

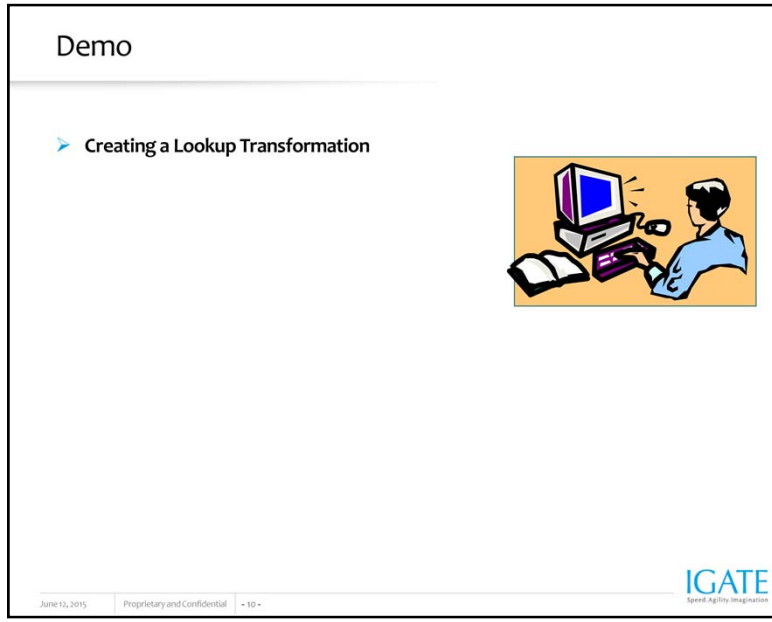
Persistent Cache

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



IGATE
Speed. Agility. Imagination

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



Steps to create a Lookup transformation:

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

4.2. Filter Transformation

Description

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

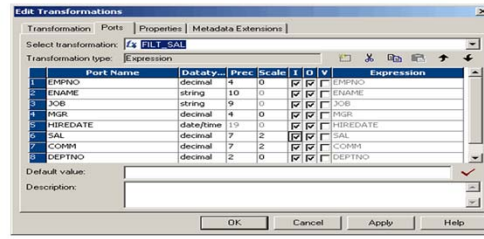


Figure 4.3

IGATE
Speed. Agility. Imagination

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

Filter Condition

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

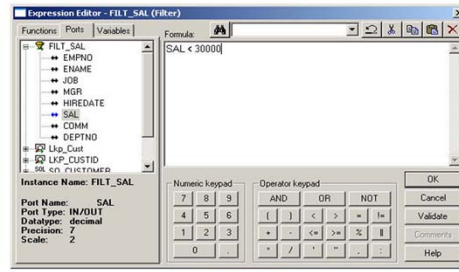


Figure 4.4

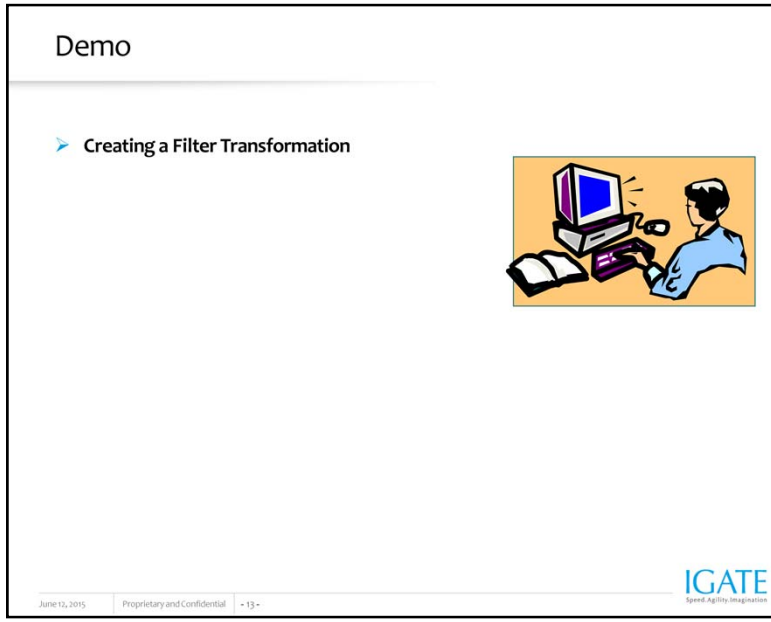
- The transformation language is used to enter the filter condition expression

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

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

SALARY < 30000

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



Steps to create a Filter transformation:

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

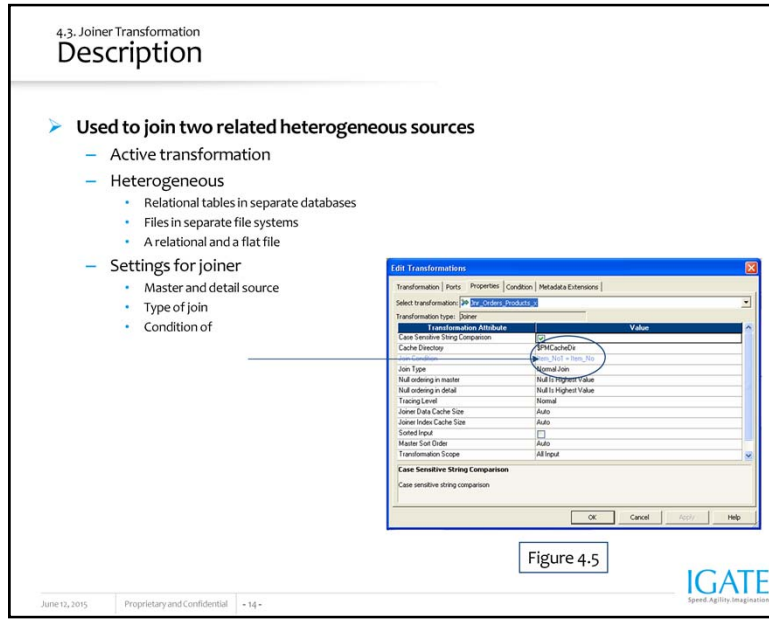


Figure 4.5

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

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

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

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

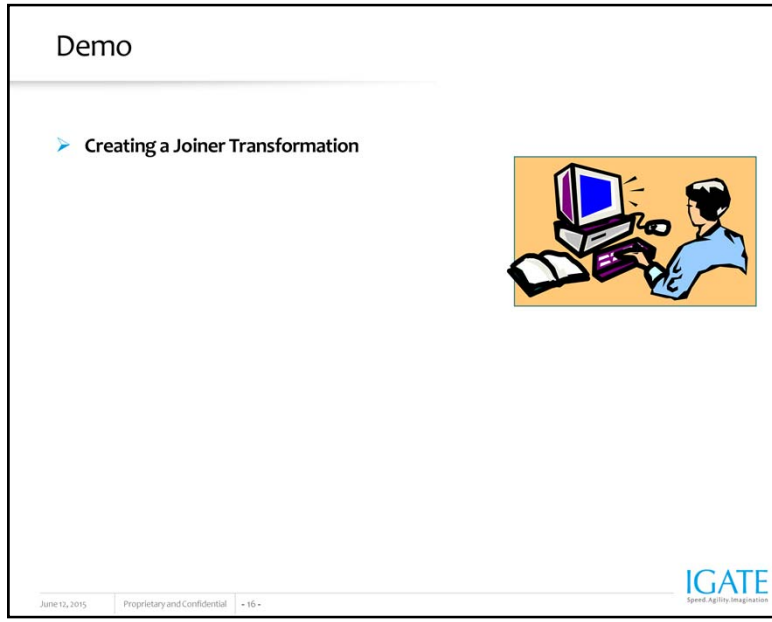
One of the sources is specified as the master source, and the other as the detail source. This is specified in the Ports tab of the transformation by clicking the ***M*** (Master/Detail checkbox) column. When ports of a transformation are added to a Joiner transformation, the ports from the first source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

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

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

The **condition** of the join is a mandatory condition defining at least one field from each data source that the transformation uses to perform the join. These fields must be declared to be of the same data type. For example, the following condition joins data from two sources based on an item ID:
ITEM_NO = ITEM_NO1

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



Steps to create a Joiner Transformation:

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

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

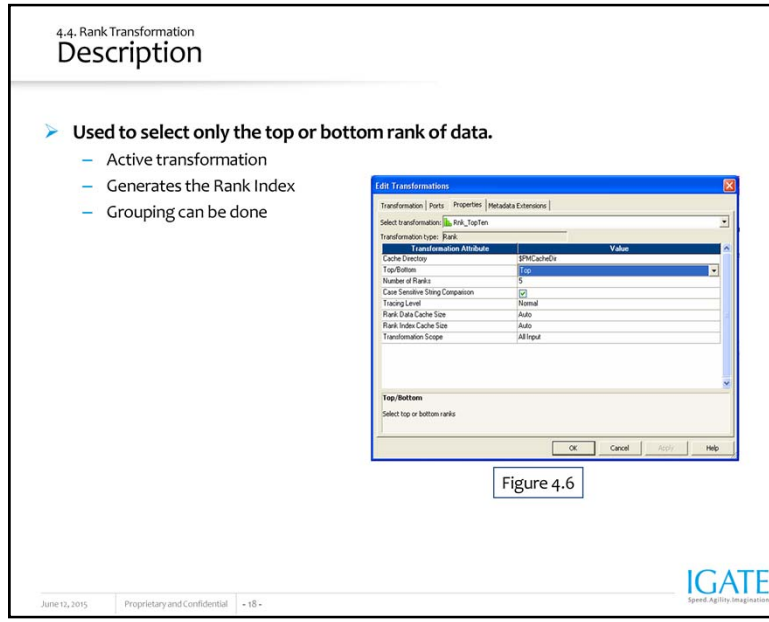
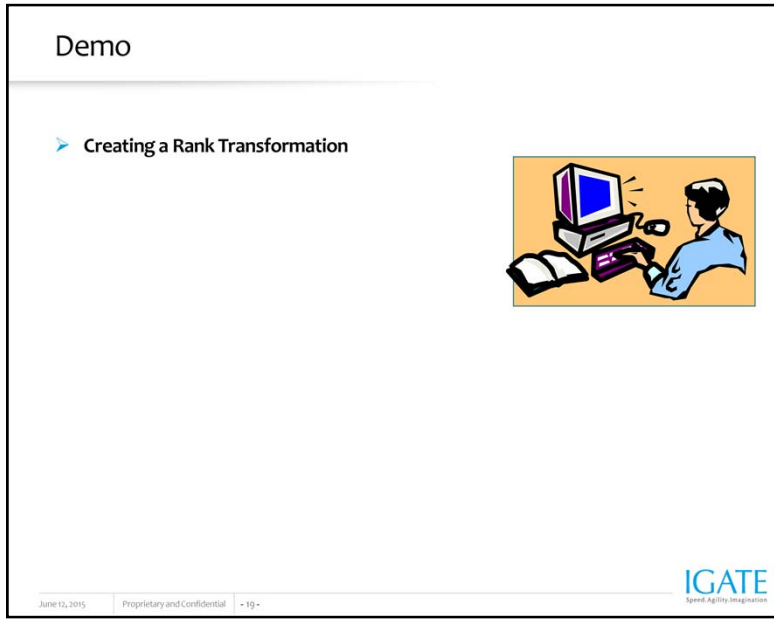


Figure 4.6

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

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

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



Steps to create a Rank Transformation:

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

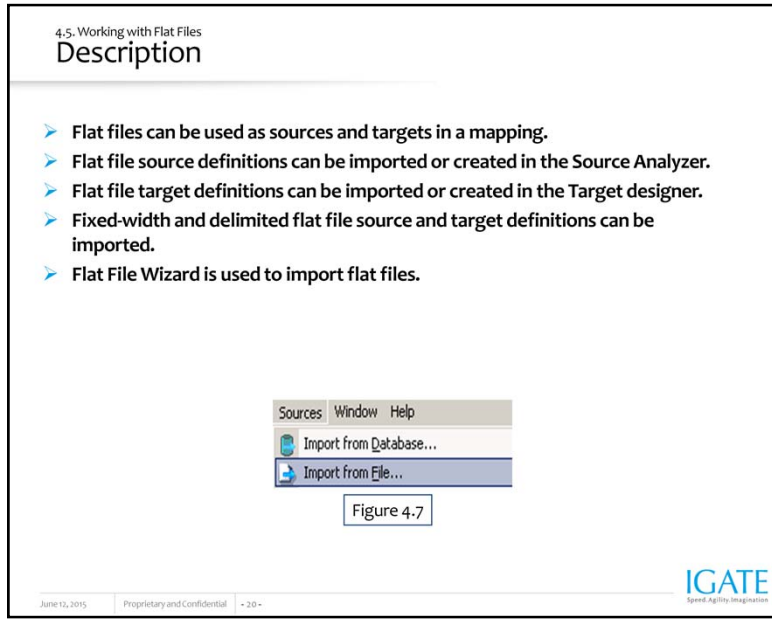


Figure 4.7

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

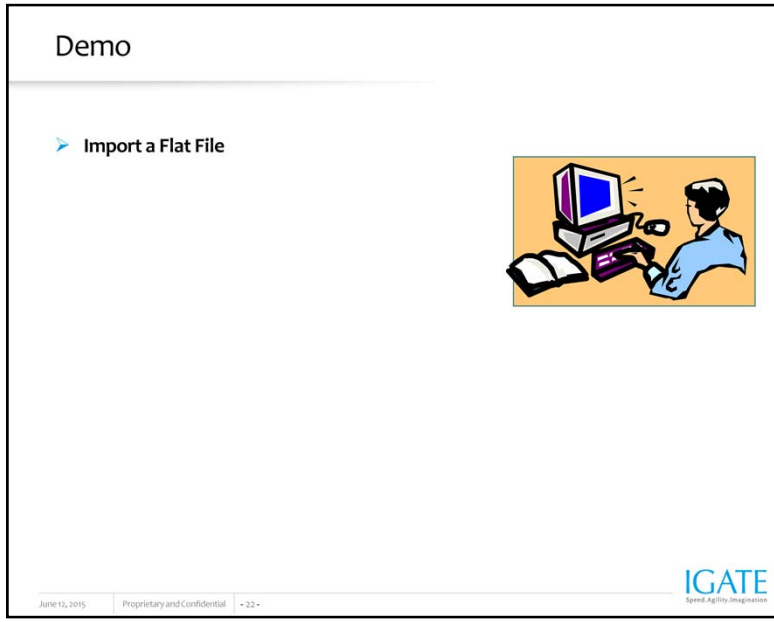
After the file source and target definitions are created, they can be used in mappings.

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

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

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

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



Steps to import a Flat file definition:

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

Summary

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

Review Question

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