Informatica PC Training

Day-3

Agenda:

- Transformation and related components
- Source Qualifier transformation
- Expression Transformation
- Filter Transformation
- Router Transformation
- Joiner Transformation
- Union Transformation
- Demo

Debadatta Mohanty

Housekeeping Tips

- **Please mute your phone during the presentation.**
- **If there is too much noise, participants will be put on auto-mute.**
- **We shall open up the table for Q&A at the end of the session.**
- **Please feel free to post your questions over Chat as well.**
- **This session will be recorded and an email will be sent with links to the recordings after the session.**
- **At the end of the course, TEX will request you to provide feedback on the training.**

Transformations used in Informatica

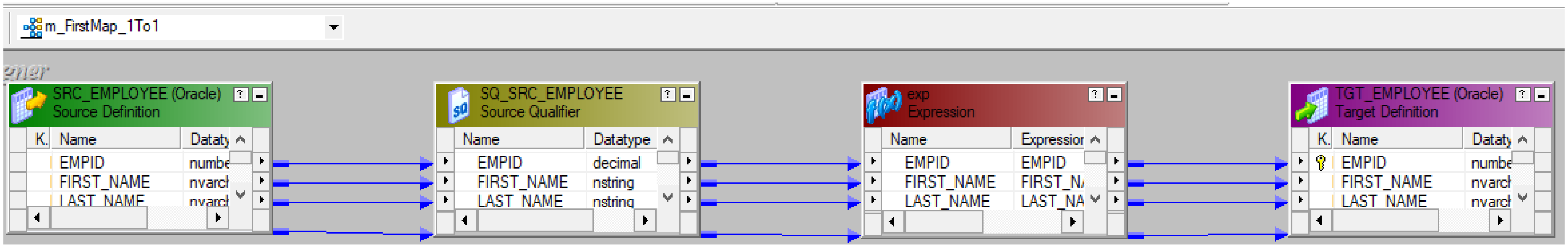
What is a transformation?

- A transformation is repository object that generates, modifies and/or passes data to other repository object.
- The Designer provides a set of transformations that perform specific functions.
- Transformations in a mapping represent the operations the Integration Service performs on data.
- Data passes into and out of transformations through ports that you connect in a Mapping or Mapplet.
- Transformations can be active or passive and Connected or unconnected.

Transformation Types

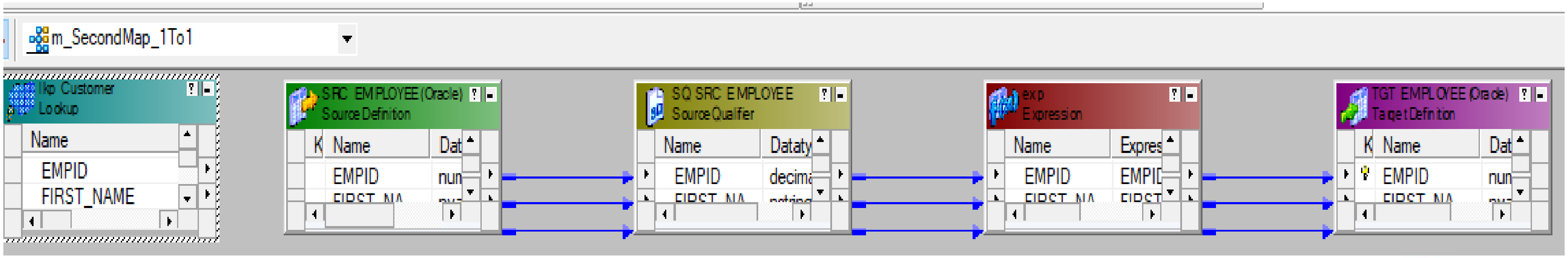
Connected Transformation

- The transformation which connected to other Transformation in the data flow Mapping.
- *Example* – SQ and Exp transformations are connected transformations in below mapping.



Unconnected Transformation

- Transformation which is not connected to other transformations in the mapping. An unconnected transformation is called within another transformation and returns a value to that transformation.
- Example – Lookup Transformation is used as unconnected transformations in below mapping.



Transformation Types

- ***Active Transformation***

  An active transformation can perform any of the following actions:

  (a) Change the number of rows that pass through the transformation:- for eg, the Filter transformation is active because it removes rows that do not meet the filter condition.

  (b) Change the transaction boundary:- for eg, the Transaction Control transformation is active because it defines a commit or roll back transaction based on an expression evaluated for each row.

  (c) Change the row type:- for eg, the Update Strategy transformation is active because it flags rows for insert, delete, update, or reject.

  Example: Filter transformation

- ***Passive Transformation***

  A Passive transformation which will satisfy all below conditions:

  (a) Do not Change the number of rows that pass through the transformation

  (b) Maintains the transaction boundary

  (c) Maintains the row type

  Example: Expression transformation

Transformations

Transformation	Type	Description
Aggregator	Active & Connected	Performs aggregate calculations.
Application Source Qualifier	Active & Connected	Represents the rows that the Integration Service reads from an application, such as an ERP source, when it runs a session.
Custom	Active or Passive & Connected.	Calls a procedure in a shared library or DLL.
Data Masking	Passive & Connected	Replaces sensitive production data with realistic test data for non-production environments.
Expression	Passive & Connected	Calculates a value.
External Procedure	Passive & Connected or Unconnected	Calls a procedure in a shared library or in the COM layer of Windows.
Filter	Active & Connected	Filters data.
HTTP	Passive & Connected	Connects to an HTTP server to read or update data.
Input	Passive & Connected	Defines mapplet input rows. Available in the Mapplet Designer.

Transformations

Transformation	Type	Description
Java	Active or Passive & Connected.	Executes user logic coded in Java. The byte code for the user logic is stored in the repository.
Joiner	Active & Connected	Joins data from different databases or flat file systems.
Lookup	Active or Passive & Connected or Unconnected	Look up and return data from a flat file, relational table, view, or synonym.
Normalizer	Active & Connected	Source qualifier for COBOL sources. Can also use in the pipeline to normalize data from relational or flat file sources.
Output	Passive & Connected	Defines mapplet output rows. Available in the Mapplet Designer.
Rank	Active & Connected	Limits records to a top or bottom range.
Router	Active & Connected	Routes data into multiple transformations based on group conditions.
Sequence Generator	Passive & Connected	Generates primary keys.
Sorter	Active & Connected	Sorts data based on a sort key.
Source Qualifier	Active & Connected	Represents the rows that the Integration Service reads from a relational or flat file source when it runs a session.

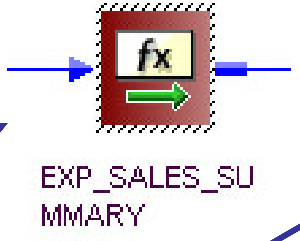
Transformations

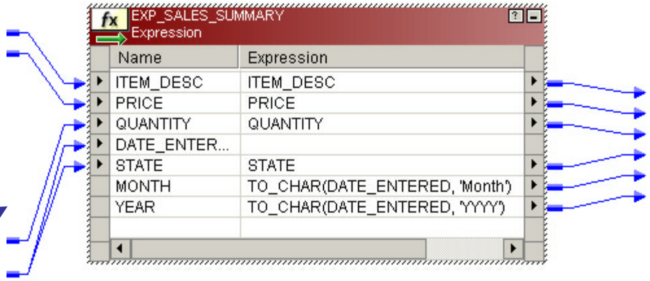
Transformation	Type	Description
SQL	Active or Passive & Connected.	Executes SQL queries against a database.
Stored Procedure	Passive & Connected or Unconnected	Calls a stored procedure.
Transaction Control	Active & Connected	Defines commit and rollback transactions.
Union	Active & Connected	Merges data from different databases or flat file systems.
Unstructured Data	Active or Passive & Connected.	Transforms data in unstructured and semi-structured formats.
Update Strategy	Active & Connected	Determines whether to insert, delete, update, or reject rows.
XML Generator	Active & Connected	Reads data from one or more input ports and outputs XML through a single output port.
XML Parser	Active & Connected	Reads XML from one input port and outputs data to one or more output ports.
XML Source Qualifier	Active & Connected	Represents the rows that the Integration Service reads from an XML source when it runs a session.

Transformation Views

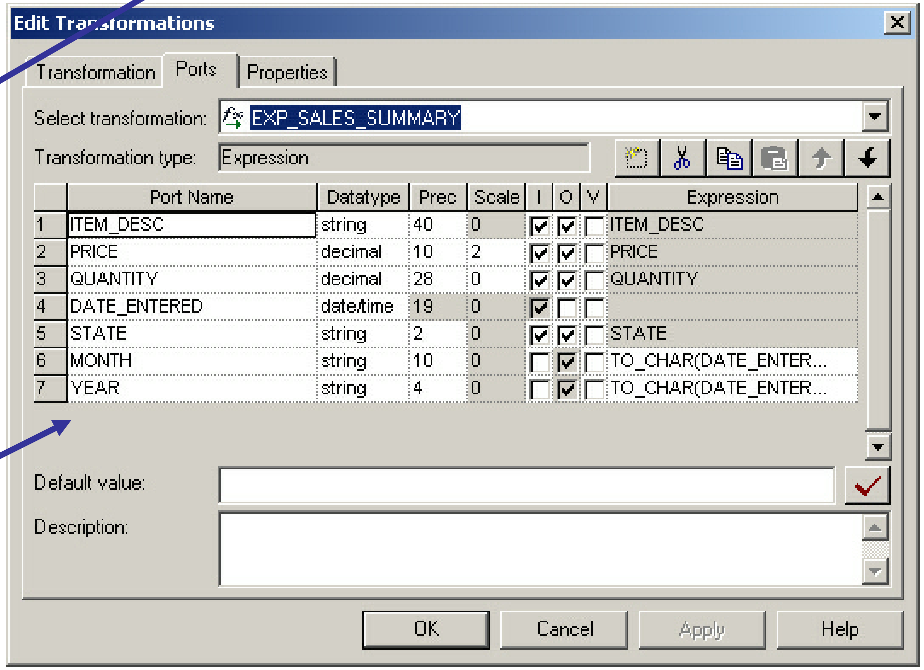
A transformation has three views:

- Iconized - shows the transformation in relation to the rest of the mapping
- Normal - shows the flow of data through the transformation
- Edit - shows transformation ports and properties; allows editing




EXP_SALES_SUMMARY
Expression

Name	Expression
ITEM_DESC	ITEM_DESC
PRICE	PRICE
QUANTITY	QUANTITY
DATE_ENTER...	
STATE	STATE
MONTH	TO_CHAR(DATE_ENTERED, 'Month')
YEAR	TO_CHAR(DATE_ENTERED, 'YYYY')


Edit Transformations

Transformation | Ports | Properties

Select transformation: EXP_SALES_SUMMARY

Transformation type: Expression

	Port Name	Datatype	Prec	Scale	I	O	V	Expression
1	ITEM_DESC	string	40	0	☑	☑	□	ITEM_DESC
2	PRICE	decimal	10	2	☑	☑	□	PRICE
3	QUANTITY	decimal	28	0	☑	☑	□	QUANTITY
4	DATE_ENTERED	date/time	19	0	☑	□	□	
5	STATE	string	2	0	☑	☑	□	STATE
6	MONTH	string	10	0	□	☑	□	TO_CHAR(DATE_ENTER...
7	YEAR	string	4	0	□	☑	□	TO_CHAR(DATE_ENTER...

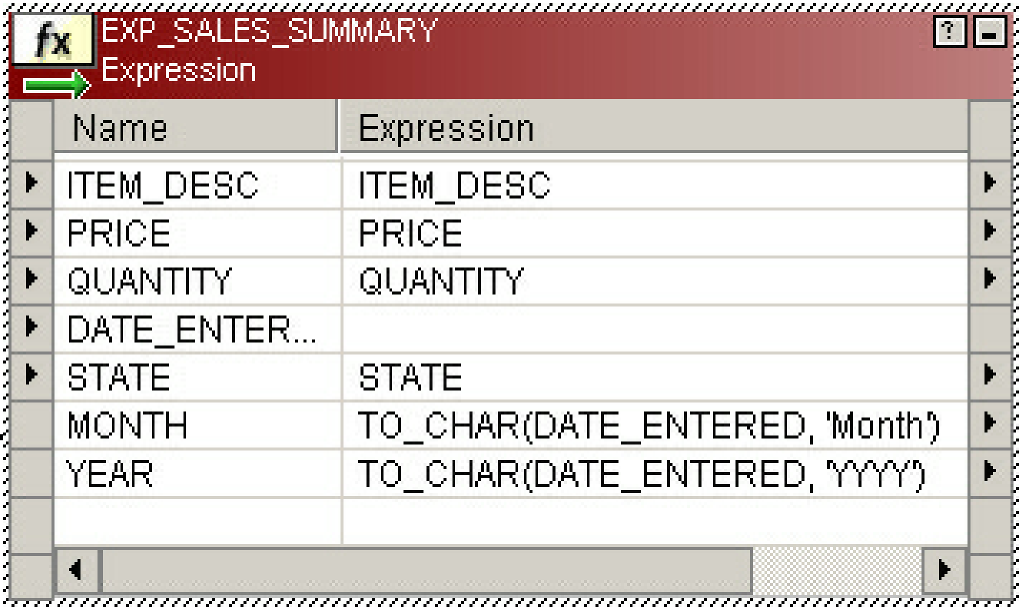
Default value:

Description:

Transformation Normal View

Shows data flow through the transformation

Data passes through I/O ports unchanged



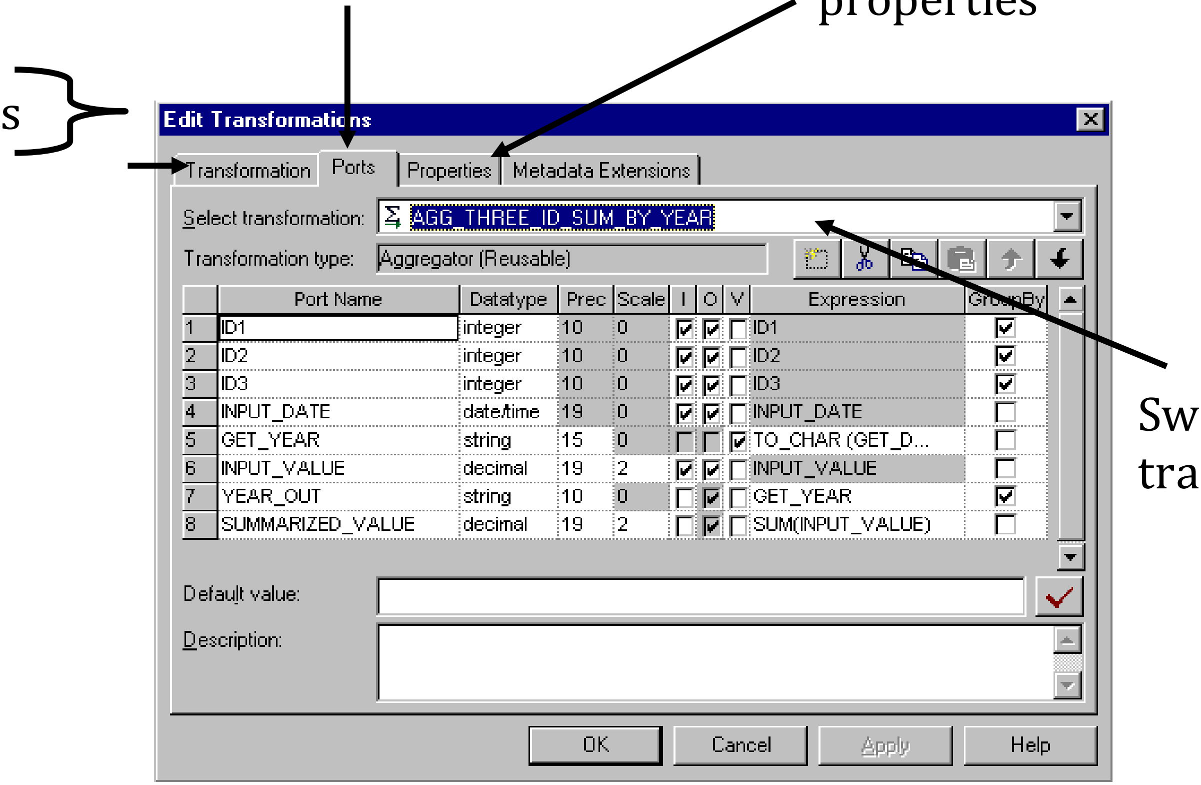
EXP_SALES_SUMMARY
Expression

Name	Expression
ITEM_DESC	ITEM_DESC
PRICE	PRICE
QUANTITY	QUANTITY
DATE_ENTER...	
STATE	STATE
MONTH	TO_CHAR(DATE_ENTERED, 'Month')
YEAR	TO_CHAR(DATE_ENTERED, 'YYYY')

- DATE_ENTERED passes into the transformation through an input port
- It is used in the MONTH port to extract the month
- The month is passed through MONTH output port

Transformation Edit Mode

Allows users to change or create transformation ports and properties



Define port level handling

Define transformation level properties

Enter comments

Make reusable

Switch between transformations

Edit Transformations

Select transformation: AGG_THREE_ID_SUM_BY_YEAR

Transformation type: Aggregator (Reusable)

	Port Name	Datatype	Prec	Scale	I	O	V	Expression	GroupBy
1	ID1	integer	10	0	☑	☑	□	ID1	☑
2	ID2	integer	10	0	☑	☑	□	ID2	☑
3	ID3	integer	10	0	☑	☑	□	ID3	☑
4	INPUT_DATE	date/time	19	0	☑	☑	□	INPUT_DATE	□
5	GET_YEAR	string	15	0	□	□	☑	TO_CHAR (GET_D...	□
6	INPUT_VALUE	decimal	19	2	☑	☑	□	INPUT_VALUE	□
7	YEAR_OUT	string	10	0	□	☑	□	GET_YEAR	☑
8	SUMMARIZED_VALUE	decimal	19	2	□	☑	□	SUM(INPUT_VALUE)	□

Default value:

Description:

Transformations - Properties

- **Port Name**

  Copied ports will inherit the name of contributing port
  Copied ports with the same name will be appended with a number
  Types Of Ports:
  - ***Input:*** Input Data from previous stage.
  - ***Output:*** Output Data to the next stage.
  - ***Variable:*** The port that stores value temporarily.
  - ***Lookup:*** Port to be used to compare Data.
  - ***Return:*** The Port which returns value from Lookup.

- **Data types**

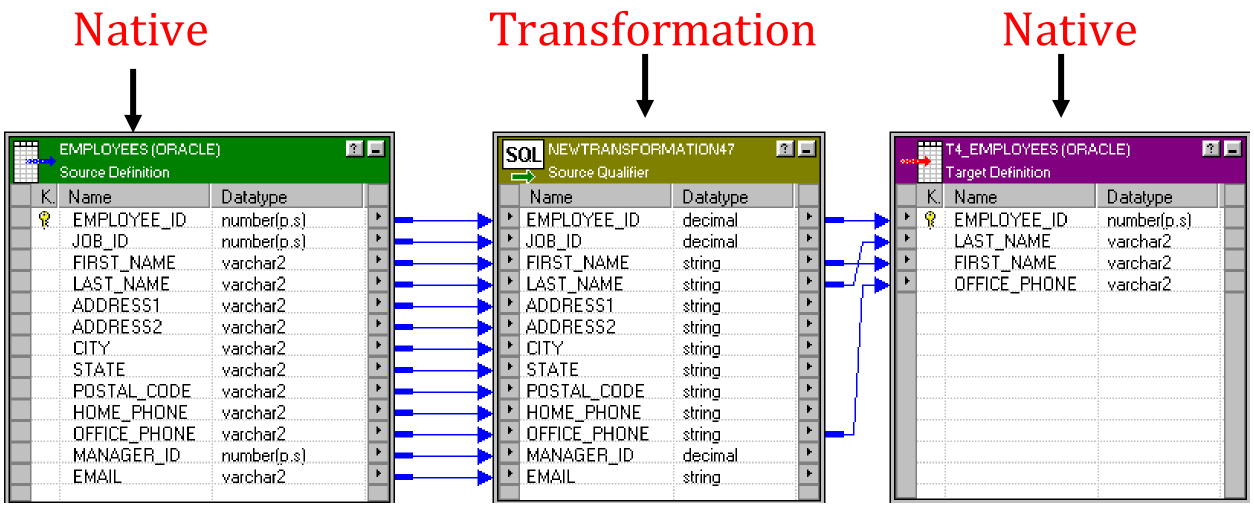
  Transformations use internal data types.
  Data types of input ports must be compatible with data types of the feeding output port.

- **Port Default values** - can be set to handle nulls and errors

- **Description** - can enter port comments

Informatica Data Types

NATIVE DATATYPES	TRANSFORMATION DATATYPES
Specific to the source and target database types	PowerCenter internal datatypes based on ANSI SQL-92
Display in source and target tables within Mapping Designer	Display in transformations within Mapping Designer



Native

Transformation

Native

EMPLOYEES (ORACLE)
Source Definition

K.	Name	Datatype
🔑	EMPLOYEE_ID	number(p,s)
	JOB_ID	number(p,s)
	FIRST_NAME	varchar2
	LAST_NAME	varchar2
	ADDRESS1	varchar2
	ADDRESS2	varchar2
	CITY	varchar2
	STATE	varchar2
	POSTAL_CODE	varchar2
	HOME_PHONE	varchar2
	OFFICE_PHONE	varchar2
	MANAGER_ID	number(p,s)
	EMAIL	varchar2

SQL NEWTRANSFORMATION47
Source Qualifier

Name	Datatype
EMPLOYEE_ID	decimal
JOB_ID	decimal
FIRST_NAME	string
LAST_NAME	string
ADDRESS1	string
ADDRESS2	string
CITY	string
STATE	string
POSTAL_CODE	string
HOME_PHONE	string
OFFICE_PHONE	string
MANAGER_ID	decimal
EMAIL	string

T4_EMPLOYEES (ORACLE)
Target Definition

K.	Name	Datatype
🔑	EMPLOYEE_ID	number(p,s)
	LAST_NAME	varchar2
	FIRST_NAME	varchar2
	OFFICE_PHONE	varchar2

- Transformation datatypes allow *mix and match* of source and target database types
- When connecting ports, native and transformation datatypes must be compatible (or must be explicitly converted)

Data type Conversions

	Integer	Decimal	Double	Char	Date
Integer	X	X	X	X	
Decimal	X	X	X	X	
Double	X	X	X	X	
Char	X	X	X	X	X
Date				X	X

- All numeric data can be converted to all other numeric datatypes, e.g. - integer, double, and decimal
- All numeric data can be converted to string, and vice versa
- Date can be converted only to date and string, and vice versa
- Other conversions not listed above are not supported

Source Qualifier Transformation

Source Qualifier Transformation

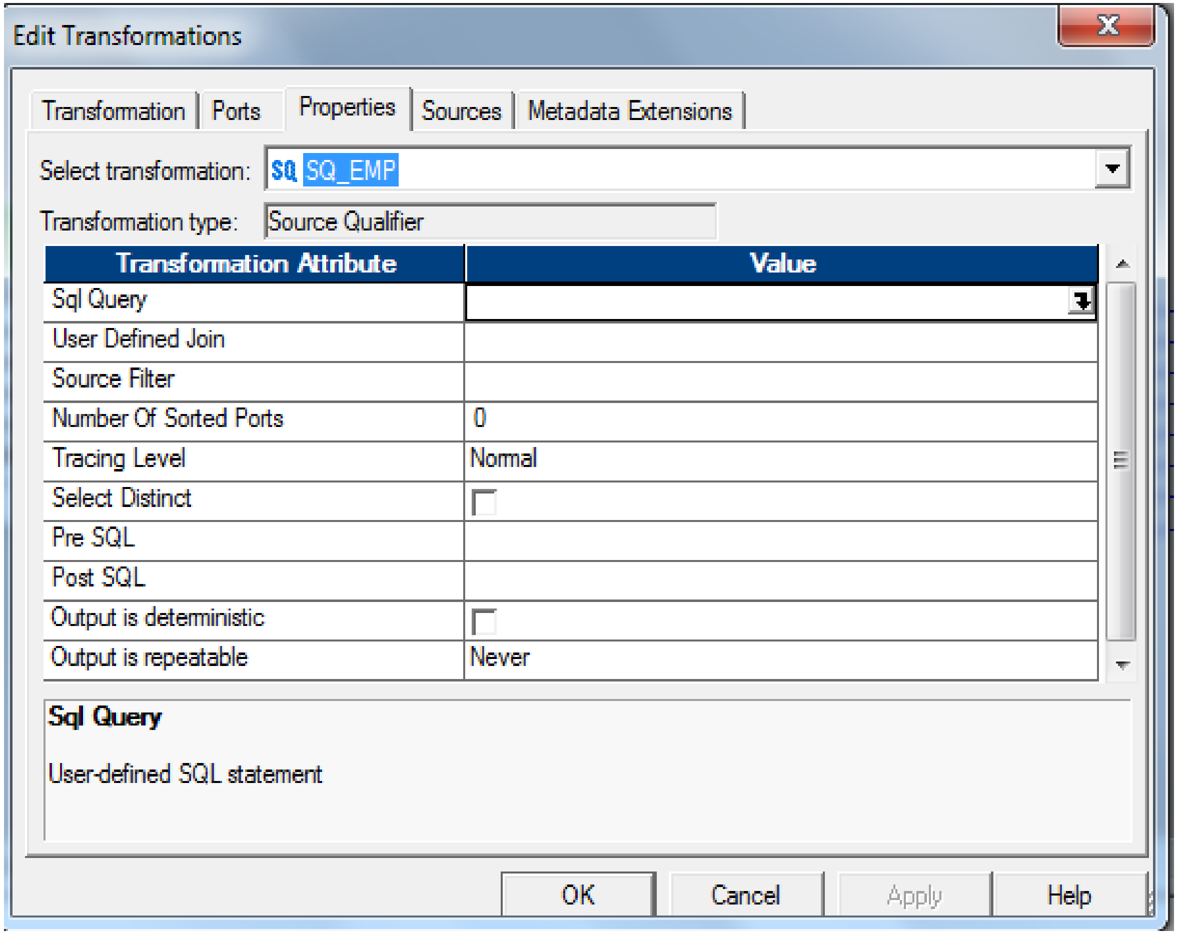
Represents the source record set queried by the Server.
Mandatory in Mappings using relational or flat file sources.

- ➢ Join data originating from the same source database.
- ➢ Filter rows when the Integration Service reads source data.
- ➢ Specify an outer join rather than the default inner join.
- ➢ Specify sorted ports.
- ➢ Select only distinct values from the source.
- ➢ Create a custom query to issue a special SELECT statement for the Integration Service to read source data.

Source Qualifier Transformation

SQ Transformation – Properties:

- •SQL Query
- •User Defined Join
- •Source Filter
- •No of Sorted Ports
- •Tracing Level
- •Select Distinct
- •Pre SQL
- •Post SQL



Select transformation: SQ SQ_EMP

Transformation type: Source Qualifier

Transformation Attribute	Value
Sql Query	
User Defined Join	
Source Filter	
Number Of Sorted Ports	0
Tracing Level	Normal
Select Distinct	□
Pre SQL	
Post SQL	
Output is deterministic	□
Output is repeatable	Never

Sql Query

User-defined SQL statement

Pre-SQL and Post-SQL Rules

- Can use any command that is valid for the database type; no nested comments.
- Can use Mapping Parameters and Variables in SQL executed against the source.
- Use a semi-colon (;) to separate multiple statements
- Informatica Server ignores semi-colons within /* ...*/
- To use a semi-colon outside of quotes or comments, ‘escape’ it with a back slash (\)
- Workflow Manager does not validate the SQL.

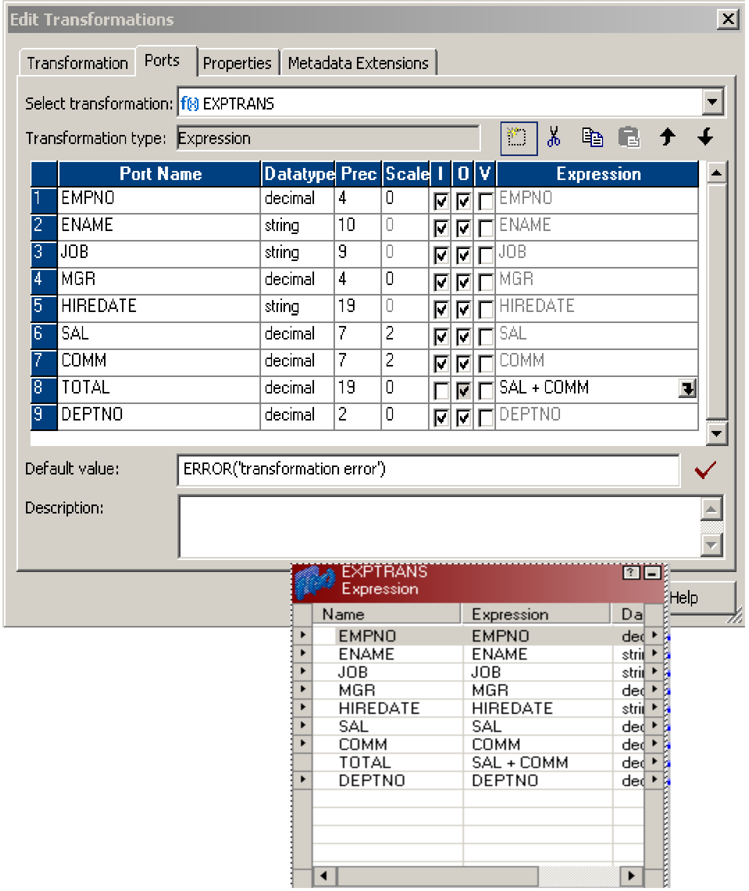
Expression Transformation

Expression Transformation

- Can use the Expression transformation to
  - perform any non-aggregate calculations
  - Calculate values in a single row
  - test conditional statements before you output the results to target tables or other transformations

- Ports that must be included in an Expression Transformation:
  - Input or input/output ports for each value used in the calculation
  - Output port for the expression
  - Variable Ports are used to temporarily store values.

Expression Transformation



Edit Transformations

Select transformation: EXPTRANS

Transformation type: Expression

	Port Name	Datatype	Prec	Scale	I	O	V	Expression
1	EMPNO	decimal	4	0	☑	☑	□	EMPNO
2	ENAME	string	10	0	☑	☑	□	ENAME
3	JOB	string	9	0	☑	☑	□	JOB
4	MGR	decimal	4	0	☑	☑	□	MGR
5	HIREDATE	string	19	0	☑	☑	□	HIREDATE
6	SAL	decimal	7	2	☑	☑	□	SAL
7	COMM	decimal	7	2	☑	☑	□	COMM
8	TOTAL	decimal	19	0	□	☑	□	SAL + COMM
9	DEPTNO	decimal	2	0	☑	☑	□	DEPTNO

Default value: ERROR('transformation error')

Description:

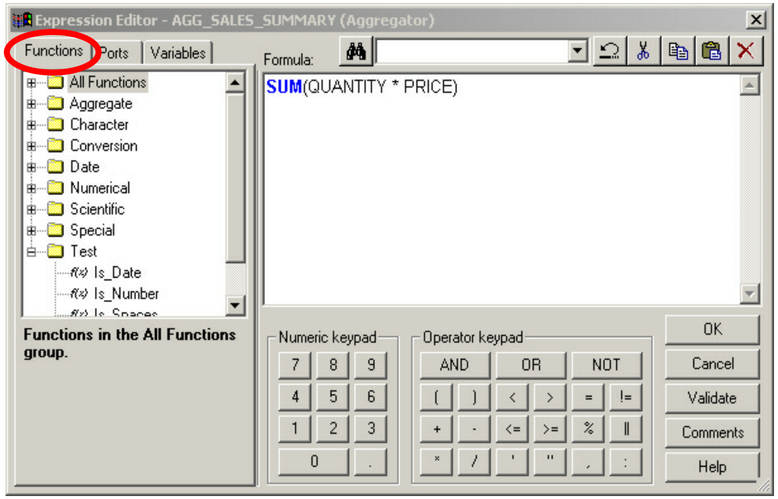
EXPTRANS
Expression

Name	Expression	Da
EMPNO	EMPNO	dec
ENAME	ENAME	stri
JOB	JOB	stri
MGR	MGR	dec
HIREDATE	HIREDATE	stri
SAL	SAL	dec
COMM	COMM	dec
TOTAL	SAL + COMM	dec
DEPTNO	DEPTNO	dec

- Can enter multiple expressions in a single expression transformation
- Can enter only one expression for each output port
- Can create any number of output ports in the transformation
- Can create Variable ports to store Data temporarily.

Expression Editor

- An expression formula is a calculation or conditional statement
- Used in Expression, Aggregator, Rank, Filter, Router, Update Strategy
- Performs calculation based on ports, functions, operators, variables, literals, constants and return values from other transformations

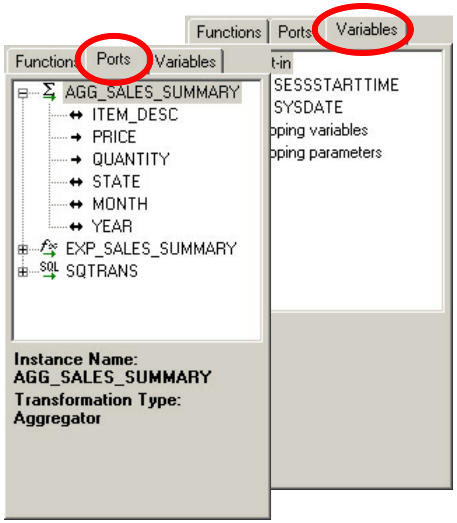


Expression Editor - AGG_SALES_SUMMARY (Aggregator)

Formula:

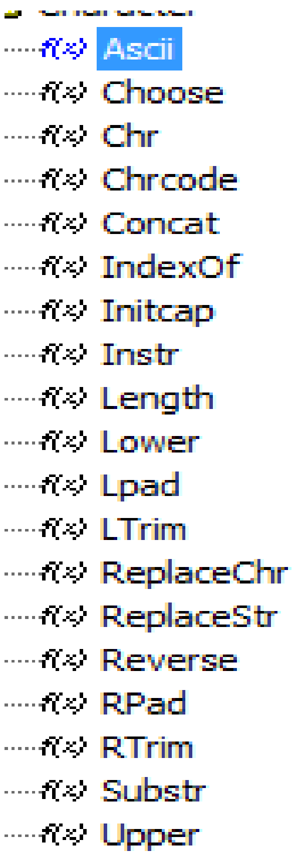
```
SUM(QUANTITY * PRICE)
```

Functions in the All Functions group.



Instance Name: AGG_SALES_SUMMARY
Transformation Type: Aggregator

Informatica Functions



Character

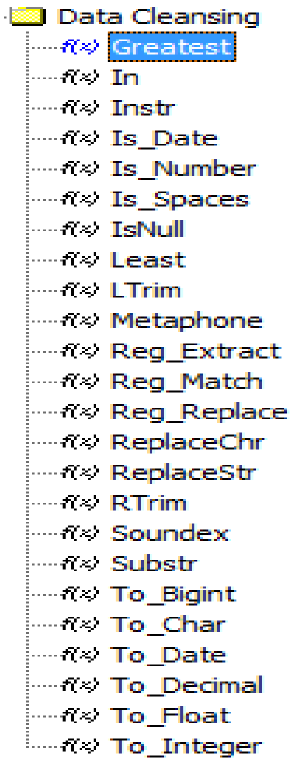
- Ascii
- Choose
- Chr
- Chrcode
- Concat
- IndexOf
- Initcap
- Instr
- Length
- Lower
- Lpad
- LTrim
- ReplaceChr
- ReplaceStr
- Reverse
- RPad
- RTrim
- Substr
- Upper

Character Functions

Used to manipulate character data

- Length- returns the length of string
- Concat- Concatenate two strings

Informatica Functions

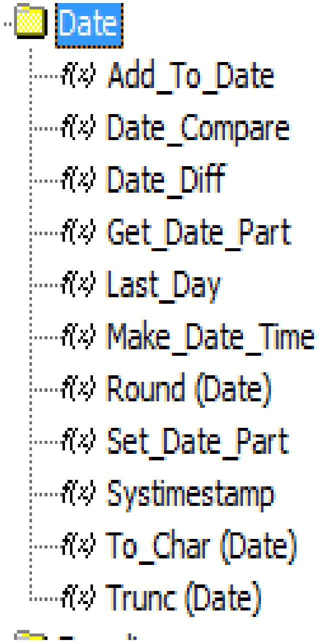


Data Cleansing
- Greatest
- In
- Instr
- Is_Date
- Is_Number
- Is_Spaces
- IsNull
- Least
- LTrim
- Metaphone
- Reg_Extract
- Reg_Match
- Reg_Replace
- ReplaceChr
- ReplaceStr
- RTrim
- Soundex
- Substr
- To_Bigint
- To_Char
- To_Date
- To_Decimal
- To_Float
- To_Integer

Data Cleansing and Conversion Functions

- Used to convert datatypes.
- Used to Validate the data for required format.
- Used to test if a lookup result is null.
- Used to validate data

Informatica Functions



- Date
  - f(x) Add_To_Date
  - f(x) Date_Compare
  - f(x) Date_Diff
  - f(x) Get_Date_Part
  - f(x) Last_Day
  - f(x) Make_Date_Time
  - f(x) Round (Date)
  - f(x) Set_Date_Part
  - f(x) Systimestamp
  - f(x) To_Char (Date)
  - f(x) Trunc (Date)

Date Functions

- Used to round, truncate, or compare dates; extract one part of a date; or perform arithmetic on a date

- To pass a string to a date function, first use the TO_DATE function to convert it to a date/time datatype

Informatica Functions

ABS
CEIL
CUME
EXP
FLOOR
LN
LOG
MOD
MOVINGAVG
MOVINGSUM
POWER
ROUND
SIGN
SQRT
TRUNC

Numerical Functions

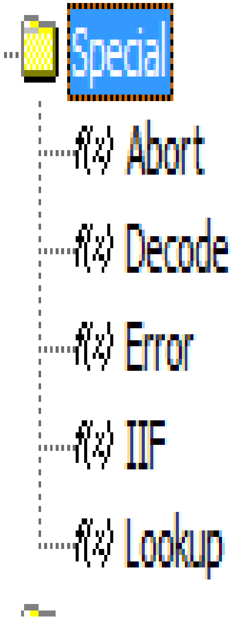
- Used to perform mathematical operations on numeric data

Scientific Functions

- Used to calculate geometric values of numeric data

COS
COSH
SIN
SINH
TAN
TANH

Informatica Functions



Special Functions:

ABORT(): Stops the session and issues a specified error message to the session log file. When the Integration Service encounters an ABORT function, it stops transforming data at that row.

DECODE(): Searches a port for a value you specify. If the function finds the value, it returns a result value, which you define.

ERROR(): Causes the Integration Service to skip a row and issue an error message, which you define. The error message displays in the session log. The Integration Service does not write these skipped rows to the session reject file.

IIF(): Returns one of two values you specify, based on the results of a condition.

LOOKUP(): The LOOKUP function compares data in a lookup source to a value you specify. When the Integration Service finds the search value in the lookup table, it returns the value from a specified column in the same row in the lookup table.

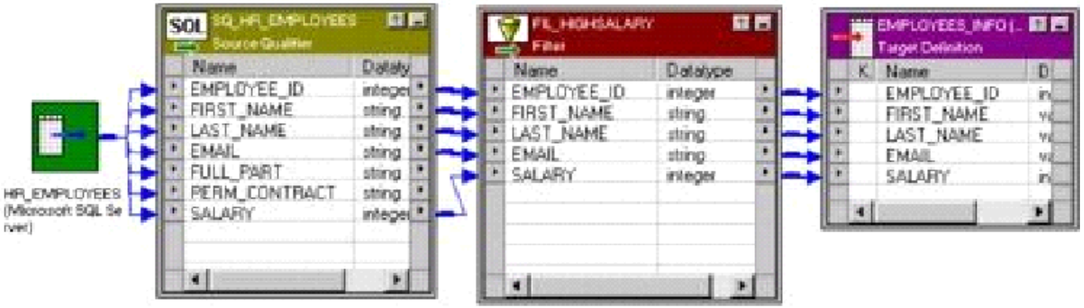
Expression Validate:

The Validate or 'OK' button in the Expression Editor will:

- Parse the current expression
  - e.g. check expression syntax is correct, references are valid or not
- Parse transformation attributes
  - e.g. - filter condition, lookup condition, SQL Query
- Parse default values
- Check spelling, correct number of arguments in functions, other syntactical errors

Filter Transformation

Filter Transformation



HR_EMPLOYEES (Microsoft SQL Server)

SQ_HR_EMPLOYEES
Source Qualifier

Name	Dataty
EMPLOYEE_ID	integer
FIRST_NAME	string
LAST_NAME	string
EMAIL	string
FULL_PART	string
PERM_CONTRACT	string
SALARY	integer

FIL_HIGHSALARY
Filter

Name	Datatype
EMPLOYEE_ID	integer
FIRST_NAME	string
LAST_NAME	string
EMAIL	string
SALARY	integer

EMPLOYEES_INFO (...
Target Definition

K	Name	D
	EMPLOYEE_ID	in
	FIRST_NAME	va
	LAST_NAME	va
	EMAIL	va
	SALARY	in

!! Filter should always be used as close to the Source, so that the Load of data carried ahead is decreased at / or near to the Source Itself.

- It provides the means for filtering rows in a mapping.
- All ports in a Filter transformation are input/output.
- Only rows that meet the condition pass through it.
- Cannot concatenate ports from more than one transformation into the Filter transformation.
- To maximize session performance, include the Filter transformation as close to the sources in the mapping as possible.
- Does not allow setting output default values.

Router Transformation

Router Transformation

Rows sent to multiple filter conditions

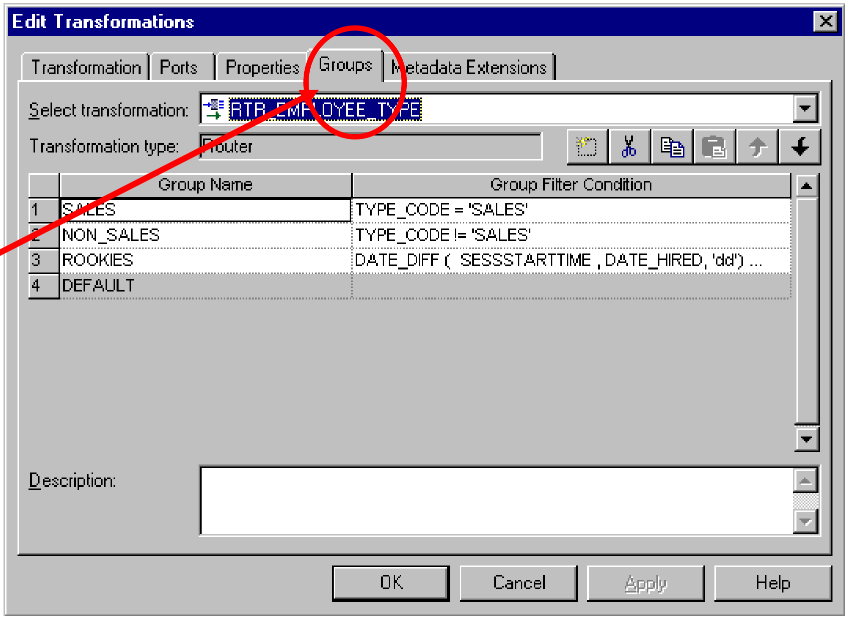
Active Transformation Connected

Ports

- All input/output
- Specify filter conditions for each Group

Usage

- Link source data in one pass to multiple filter conditions



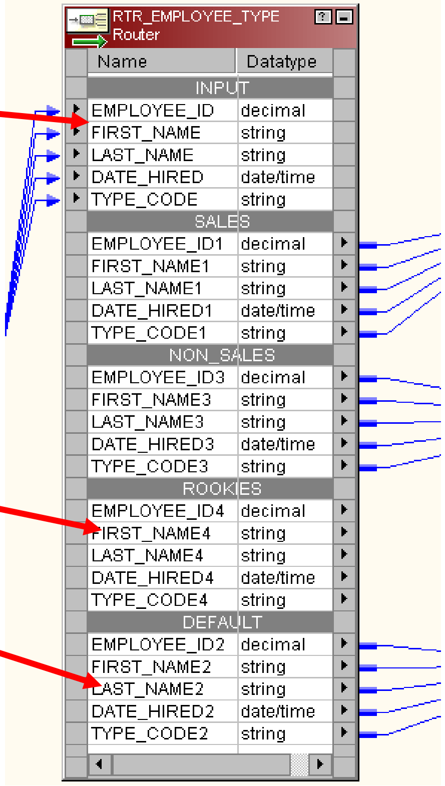
Select transformation: RTR_EMPLOYEE_TYPE

Transformation type: Router

	Group Name	Group Filter Condition
1	SALES	TYPE_CODE = 'SALES'
2	NON_SALES	TYPE_CODE != 'SALES'
3	ROOKIES	DATE_DIFF (SESSSTARTTIME , DATE_HIRED, 'dd') ...
4	DEFAULT	

Router Groups

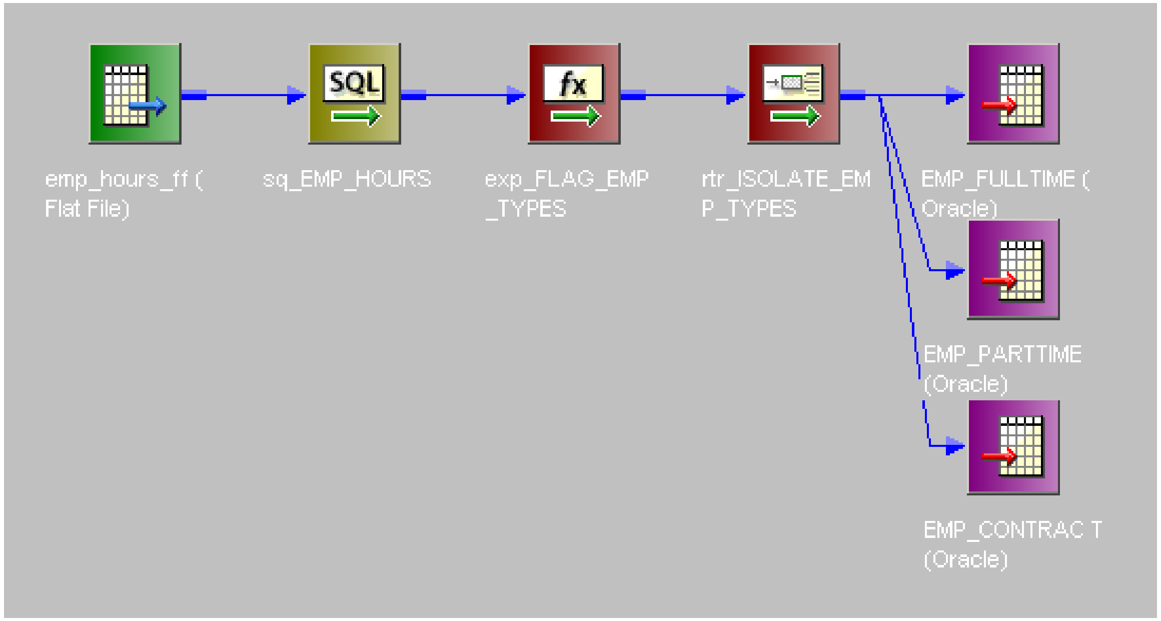
- Input group (always one)
- User-defined groups
- Each group has one condition
- ALL group conditions are evaluated for EACH row
- One row can pass multiple conditions
- Unlinked Group outputs are ignored
- Default group (always one) *can* capture rows that fail all Group conditions



RTR_EMPLOYEE_TYPE
Router

Name	Datatype
INPUT	
EMPLOYEE_ID	decimal
FIRST_NAME	string
LAST_NAME	string
DATE_HIRED	date/time
TYPE_CODE	string
SALES	
EMPLOYEE_ID1	decimal
FIRST_NAME1	string
LAST_NAME1	string
DATE_HIRED1	date/time
TYPE_CODE1	string
NON_SALES	
EMPLOYEE_ID3	decimal
FIRST_NAME3	string
LAST_NAME3	string
DATE_HIRED3	date/time
TYPE_CODE3	string
ROOKIES	
EMPLOYEE_ID4	decimal
FIRST_NAME4	string
LAST_NAME4	string
DATE_HIRED4	date/time
TYPE_CODE4	string
DEFAULT	
EMPLOYEE_ID2	decimal
FIRST_NAME2	string
LAST_NAME2	string
DATE_HIRED2	date/time
TYPE_CODE2	string

Router Transformation in a Mapping

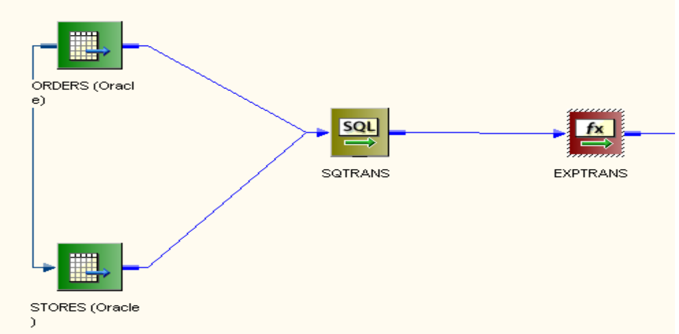


Joiner Transformation

Joiner Transformation

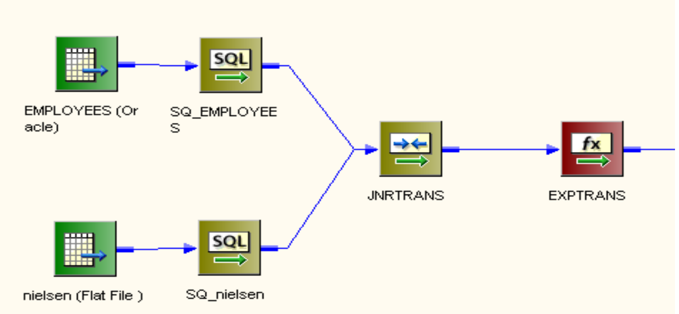
❑ ***Homogeneous Joins***: Joins that can be performed with a SQL SELECT statement

- ➢ Source Qualifier contains a SQL join
- ➢ Tables on same database server (or are synonyms)
- ➢ Database server does the join “work”
- ➢ Multiple homogenous tables can be joined



❑ ***Heterogeneous Joins:*** Joins that cannot

- ➢ An Oracle table and a Sybase table
- ➢ Two Informix tables on different database servers
- ➢ Two flat files
- ➢ A flat file and a database table



Joiner Transformation

Performs heterogeneous joins on records from different databases or flat file sources

Active Transformation
Connected

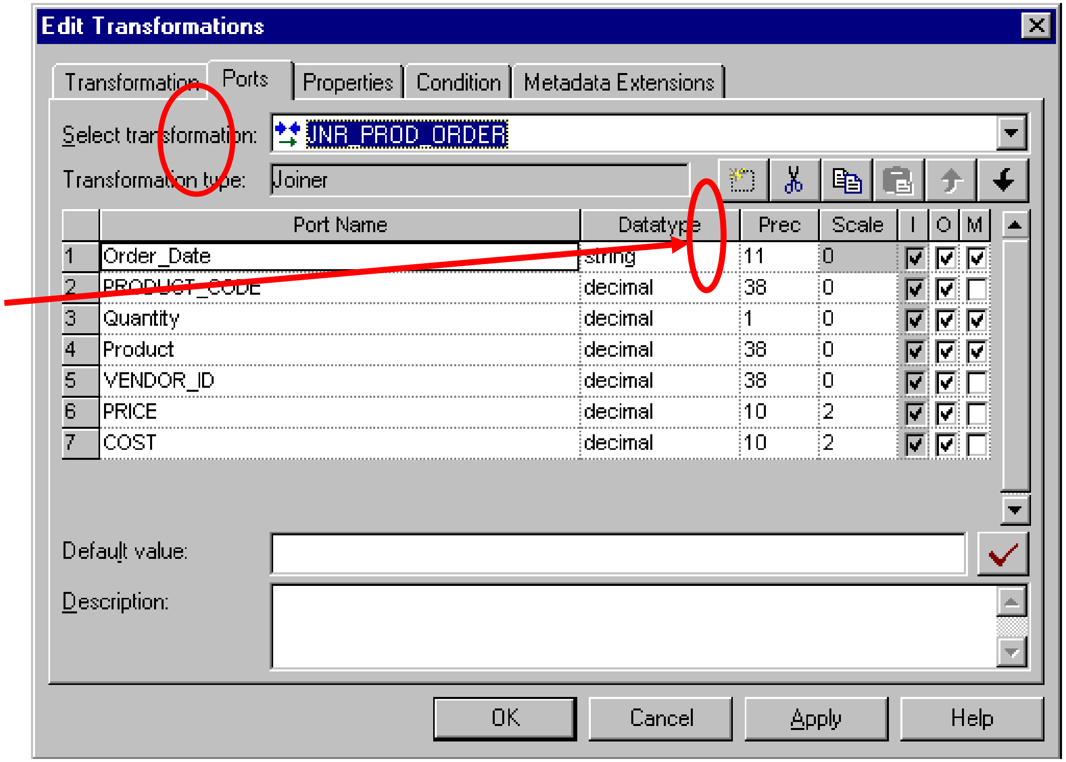
Ports

- All input or input / output
- "M" denotes port comes from master source

Specify the Join condition

Usage

- Join two flat files
- Join two tables from different databases
- Join a flat file with a relational table



Edit Transformations

Select transformation: JNR_PROD_ORDER

Transformation type: Joiner

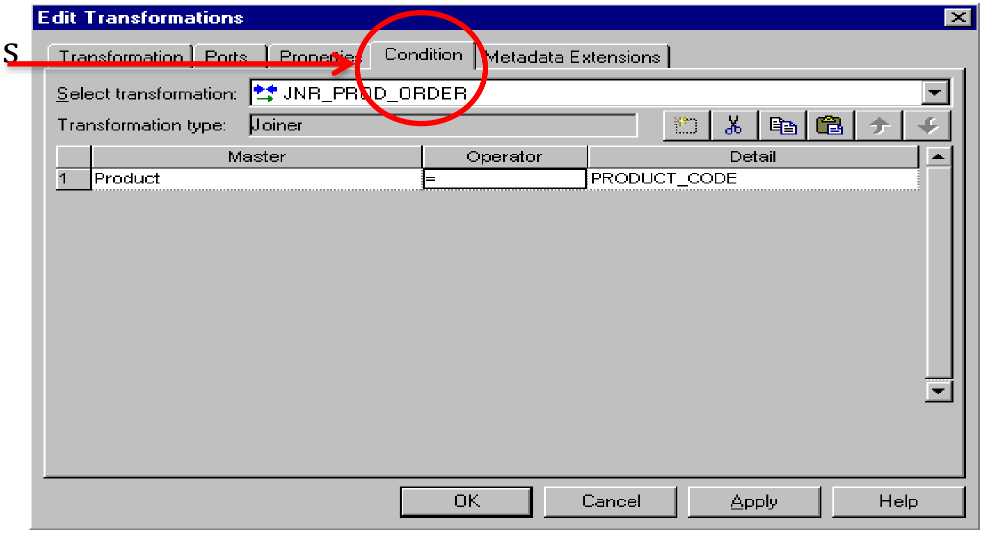
	Port Name	Datatype	Prec	Scale	I	O	M
1	Order_Date	string	11	0	☑	☑	☑
2	PRODUCT_CODE	decimal	38	0	☑	☑	□
3	Quantity	decimal	1	0	☑	☑	☑
4	Product	decimal	38	0	☑	☑	☑
5	VENDOR_ID	decimal	38	0	☑	☑	□
6	PRICE	decimal	10	2	☑	☑	□
7	COST	decimal	10	2	☑	☑	□

Default value:

Description:

Joiner Conditions

Multiple join Conditions are Supported



Select transformation: JNR_PROD_ORDER

Transformation type: Joiner

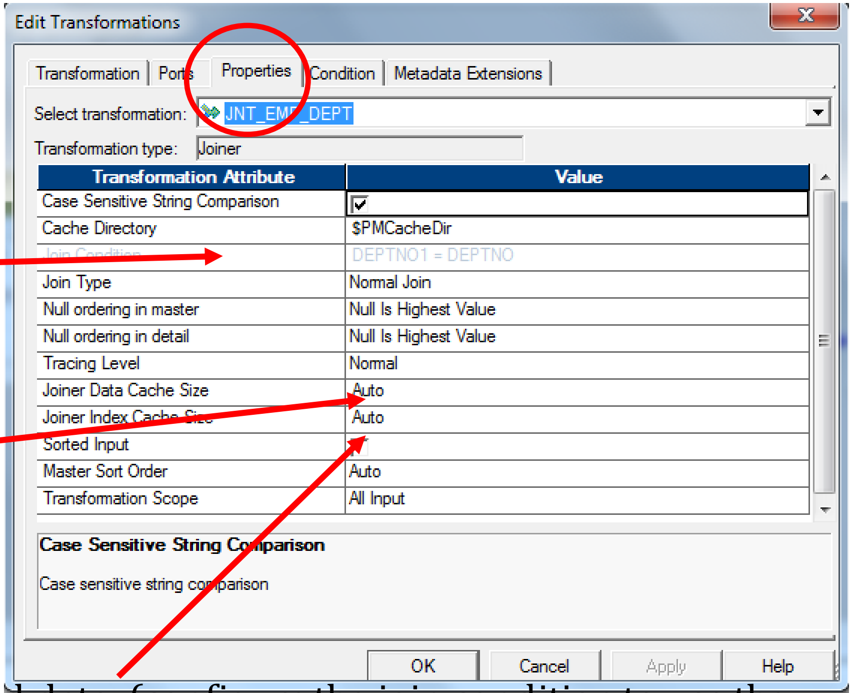
	Master	Operator	Detail
1	Product	=	PRODUCT_CODE

Joiner Properties

Join types:

- “Normal” (inner)
- Master outer
- Detail outer
- Full outer

Set Joiner Cache



Edit Transformations

Transformation | Ports | Properties | Condition | Metadata Extensions

Select transformation: JNT_EMP_DEPT

Transformation type: Joiner

Transformation Attribute	Value
Case Sensitive String Comparison	☑
Cache Directory	$PMCacheDir
Join Condition	DEPTNO1 = DEPTNO
Join Type	Normal Join
Null ordering in master	Null Is Highest Value
Null ordering in detail	Null Is Highest Value
Tracing Level	Normal
Joiner Data Cache Size	Auto
Joiner Index Cache Size	Auto
Sorted Input	☐
Master Sort Order	Auto
Transformation Scope	All Input

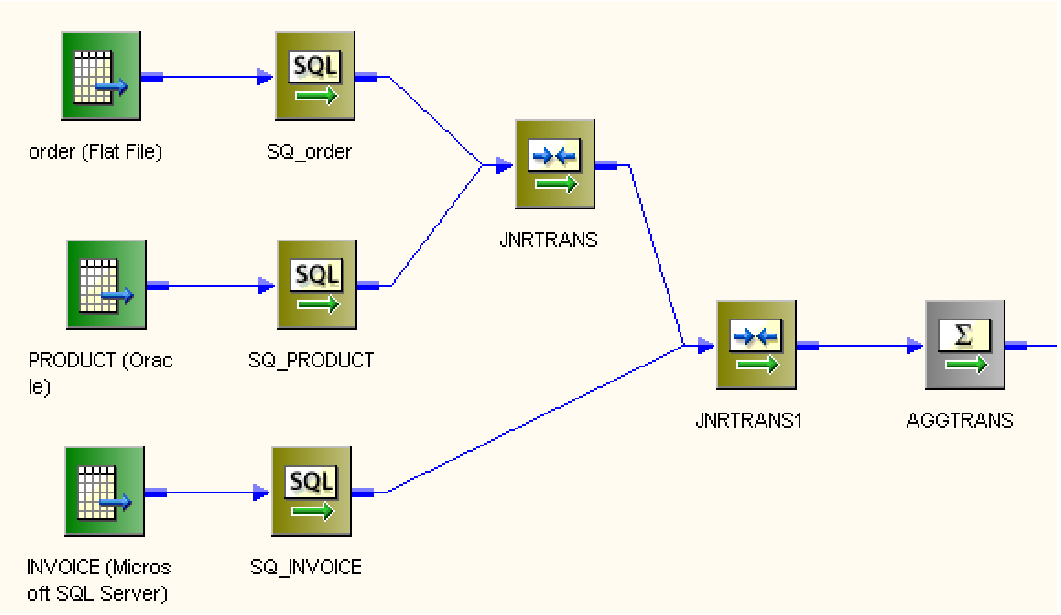
Case Sensitive String Comparison

Case sensitive string comparison

Joiner can accept sorted data (configure the join condition to use the sort origin ports)

Nested Joins

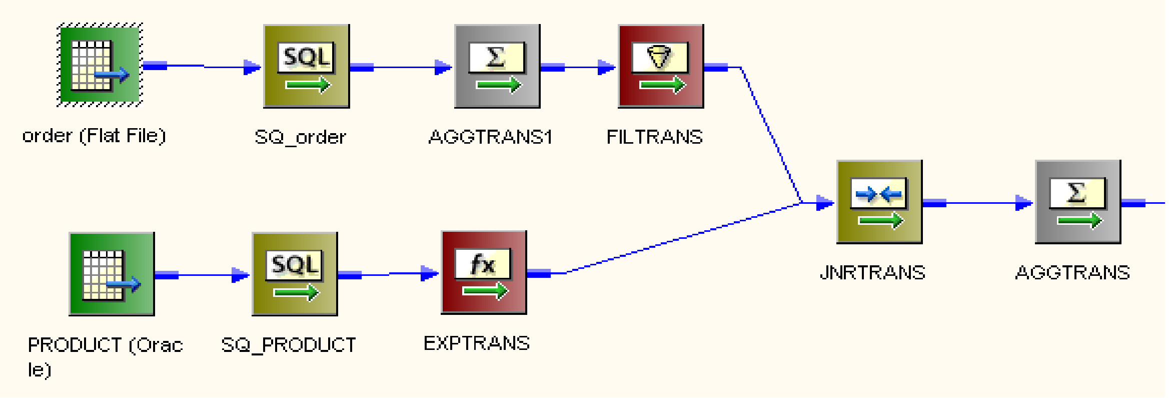
Used to join three or more heterogeneous sources



Mid-Mapping Join

The Joiner does NOT accept input in the following situations:

- Both input pipelines begin with the same Source Qualifier
- Both input pipelines begin with the same Normalizer
- Both input pipelines begin with the same Joiner
- Either input pipeline contains an Update Strategy
- Cannot be used Sequence Generator transformation directly before the Joiner transformation.



Union Transformation

Union Transformation

Merge data from multiple pipelines or pipeline branches into one pipeline branch

Active Transformation

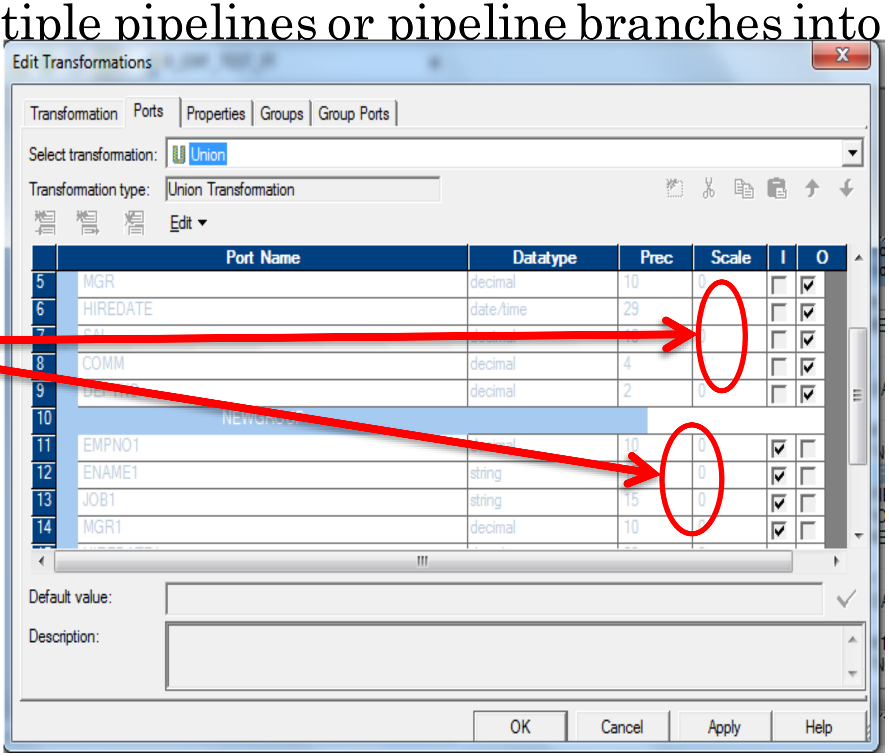
Connected

Ports

- Input , Output
- Define No of Groups and No of input pipelines.

Usage

- Merge data from Multiple Source similar to one output.



Edit Transformations

Transformation | Ports | Properties | Groups | Group Ports

Select transformation: Union

Transformation type: Union Transformation

	Port Name	Datatype	Prec	Scale	I	O
5	MGR	decimal	10	0	□	☑
6	HIREDATE	date/time	29	0	□	☑
7	SAL	decimal	10	0	□	☑
8	COMM	decimal	4	0	□	☑
9	DEPTNO	decimal	2	0	□	☑
10	NEWGROUP					
11	EMPNO1	decimal	10	0	☑	□
12	ENAME1	string		0	☑	□
13	JOB1	string	15	0	☑	□
14	MGR1	decimal	10	0	☑	□

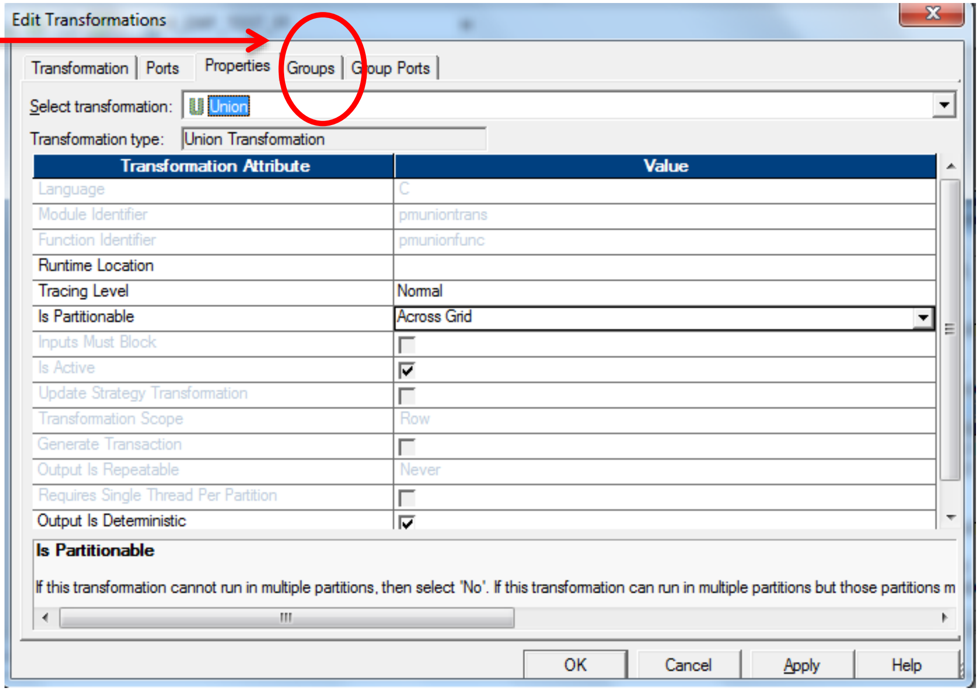
Default value:

Description:

Union Transformation - Properties

Can Create No of Input source as Group.

Can Set the Partitions level



Select transformation: Union

Transformation type: Union Transformation

Transformation Attribute	Value
Language	C
Module Identifier	pmuniontrans
Function Identifier	pmunionfunc
Runtime Location	
Tracing Level	Normal
Is Partitionable	Across Grid
Inputs Must Block	□
Is Active	☑
Update Strategy Transformation	□
Transformation Scope	Row
Generate Transaction	□
Output Is Repeatable	Never
Requires Single Thread Per Partition	□
Output Is Deterministic	☑

Is Partitionable

If this transformation cannot run in multiple partitions, then select 'No'. If this transformation can run in multiple partitions but those partitions m

Union Transformation

- Rules
  - Can create multiple input groups, but only one output group.
  - All input groups and the output group must have matching ports.
  - The precision, data type, and scale must be identical across all groups.
  - The Union transformation does not remove duplicate rows. This is similar to Union All in SQL.
  - Cannot use a Sequence Generator or Update Strategy transformation upstream from a Union transformation.

Demo and Q&A

- Demo
  - Expression transformation and simple mapping
  - Source Qualifier transformation & simple mapping
  - Filter Transformation & simple mapping
  - Router Transformation & simple mapping
  - Joiner Transformation & simple mapping
  - Union Transformation & simple mapping

Thank You