
Informatica (ETL tool)

Introduction to ETL

- What is an ETL?
 - ETL is a technology we use in DSS to populate the data warehouse.
 - **Extracting** data from one or more source systems and **Loading** them to the Data Warehouse after **Transforming** the extracted data according to the structure of the destination tables in Data Warehouse (usually de normalized).
-

ETL (E – Extract)

- Extract – Getting data out of the source systems. This may be just a DTS package which pulls the data, or exporting a table to a flat file in the source system.
 - In Teradata we have Fast Export utility where we can export the data to a flat file.
 - In Oracle we have SQL*Loader to export the data to a flat file.
 - In SQL Server we can use a DTS package to do the same job
-

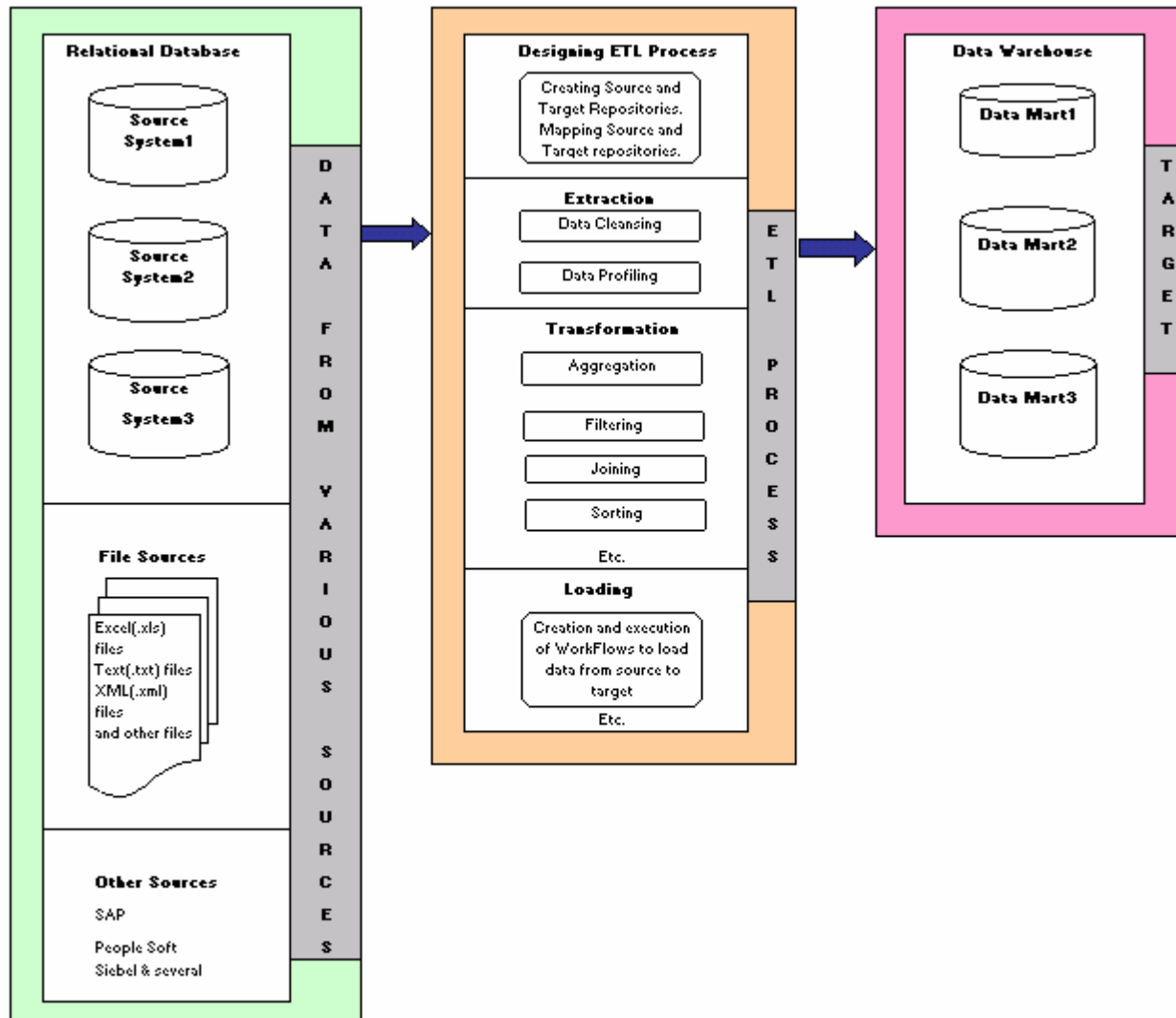
ETL (T – Transform)

- Transform – Its not necessary to have the same data model in source and destination. When the data model is different from source obviously we have to modify the source data to destination's data model. This process is called transformation.
 - Example : When we receive data from various distributors about the sales information we wont get the geo information. So in the transformation logic we will have the code which assigns the respective geo based on the country from which you are getting the data.
-

ETL (L – Load)

- Load – Loading the transformed data into the destination data model (data warehouse).
 - We use ETL tools capability or the SQL to do perform this operation.
 - As there are export functionality available in each RDBMS there is an utility to import the data into the database.
 - Teradata – Fast Import
 - Oracle – SQL*Loader
 - Sybase - bcp
-

Architecture of an ETL



Why ETL tool ?

- Easy to maintain (lot of coding need to be done if we are using db programming)
 - Drag and drop which will enable all of us in the same page and more over turn around time for implementation of project is very less.
 - If we have heterogeneous data sources ETL tool provides the facility in one place to integrate the data.
 - All the ETL tools provide a mechanism to schedule the jobs which enables us to refresh the data in DW automatically.
-

Why ETL tool?

- Handling errors, sending success or failure messages based on the job status is easier in ETL tools
 - It saves lot of time while developing the application as it's a GUI based applications.
 - Meta data driven – which is very important, say per example, if the column length has been changed either in source or in destination all you have to do is change the associate properties in the input / output tabs.
-

Informatica

- Informatica is an ETL tool which is used to extract the data from one or more sources, transform the extracted data according to the structure of the destination tables in the data warehouse and load them in to the data warehouse.
-

Informatica Components

- Power Center Server
 - Power Center Client
 - Repository Manager
 - Repository Server
 - Designer
 - Workflow Manager
 - Workflow Monitor
-

Power Center Server

- Power Center Server is responsible for the extraction of data from the source and then loading data into the targets.
-

Power Center Client

- Power Center Client is used to identify the source and the target definitions, create mappings and mapplets, create sessions, workflows and run the workflows.
-

Repository Manager

What is Repository?

Repository is a place where all the metadata information is stored in the informatica suite.

Repository Manager is used

- To create and manage the Metadata Repository.
 - To create Repository Users and Groups, assign Privileges and Permissions.
-

Repository Server


- Repository Server takes care of all the connections between the repository and the power center client.




Designer


- Designer is a work place to build the mappings and mapplets which helps to move and transform the data between the sources and the targets.
 - Designer also helps to create the source definitions ,target definitions and transformation to build the mappings.
-


Designer Tools

- Source Analyzer 
 - Source Analyzer is used to create or Import source definitions from any RDBMS, flat files, XML, COBOL applications.

 - Warehouse Designer 
 - Warehouse Designer is used to create or Import target definitions.
-

Designer Tools

- Transformation Developer 
 - Transformation Developer is used to create reusable transformations.

 - Mapplet Designer 
 - A mapplet is a reusable object that you create in the Mapplet Designer. It contains a set of transformations and allows you to reuse that transformation logic in multiple mappings.
-

Designer Tools

- Mapping Designer 

Mapping Designer is a collections of source and target definitions linked by transformation objects.

Mapping Designer represents the flow of data between the source and the target.

Workflow Manager

- Workflow Manager is used to create the task for the mappings created in the designer, schedule the task and execute the task



Workflow Manager Tools

- **Task Developer**
 - Task Developer is used to create tasks that you want to execute in the workflow.
 - **Workflow Designer**
 - Workflow Designer is used to create a workflow by connecting tasks with links.
 - **Worklet Designer**
 - Worklet Designer is created to reuse the set of workflow logic in several workflows.
-

Workflow Manager Tools

- Workflow monitor

Workflow monitor is used to monitor the workflows and the tasks.



Transformation

- Transformations help to transform the source data according to the requirements of target system.
 - Transformations ensure the quality of the data being loaded into target and this is done during the mapping process from source to target.
 - Sorting, Filtering, Aggregation, Joining are some of the examples of transformation.
-

Types Of Transformation

■ Active Transformation

Active transformation is a transformation which changes the number of rows that pass from the source to the target.

Ex:

Filter transformation removes the rows that do not meet the filter condition.

Types Of Transformation

- **Passive Transformation**

Passive transformation is a transformation which does not change the number of rows that pass from the source to the target.

Ex:

Expression transformation that performs a calculation on data and passes all rows through the transformation.

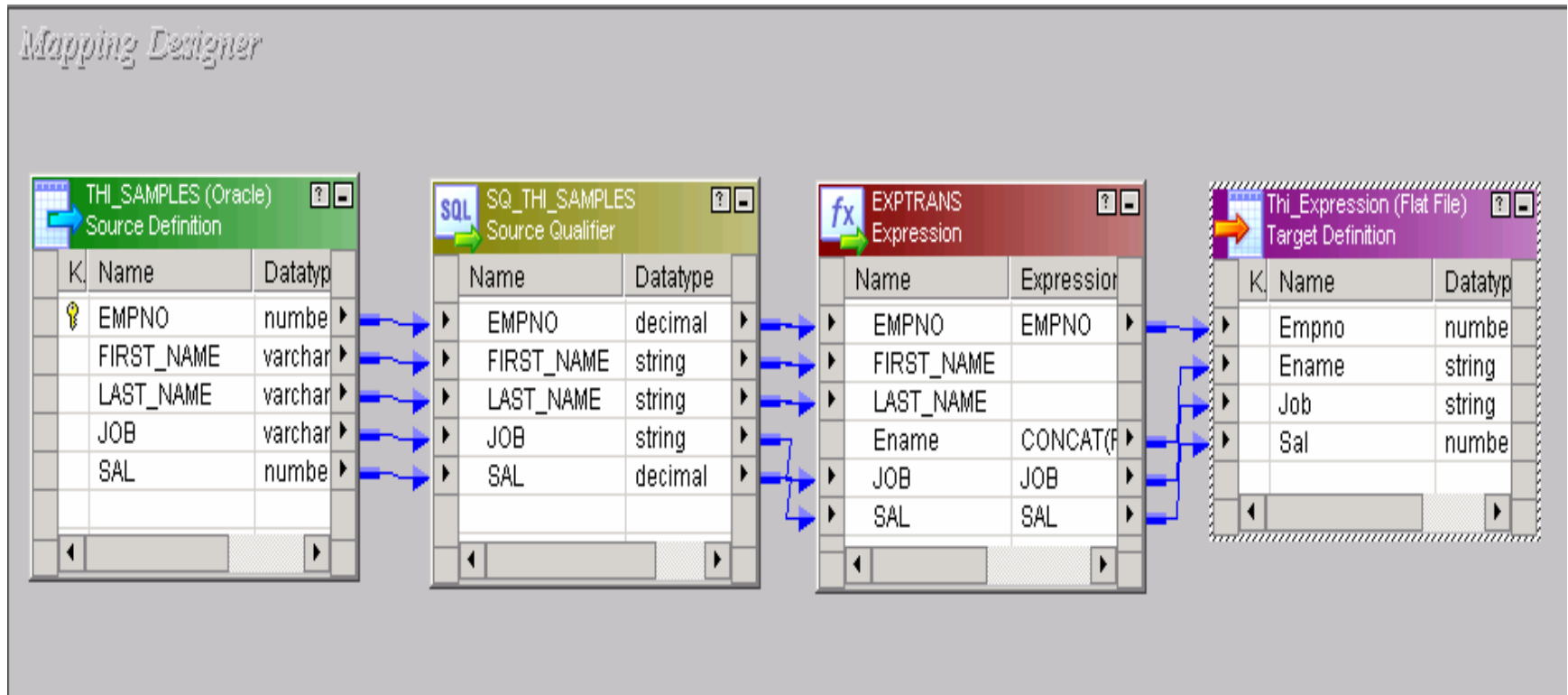
Expression Transformation

- Expression transformation is used to calculate values in a single row before you write to the target.
 - We can use the Expression transformation to perform any non-aggregate calculations.
 - For example, you might need to adjust employee salaries, concatenate first and last names.
-

Example for Expression Transformation

- The Expression Transformation allows to concatenate the two separate columns (first name, last name) load them as a single column (ename) into the target file.
-

Example for Expression Transformation



Filter Transformation

- Filter transformation is used to remove or filter the rows which does not meet the condition.
 - It is an Active transformation.
-

Properties

- Filter condition

Filter condition to be specified in this.

- Tracing level

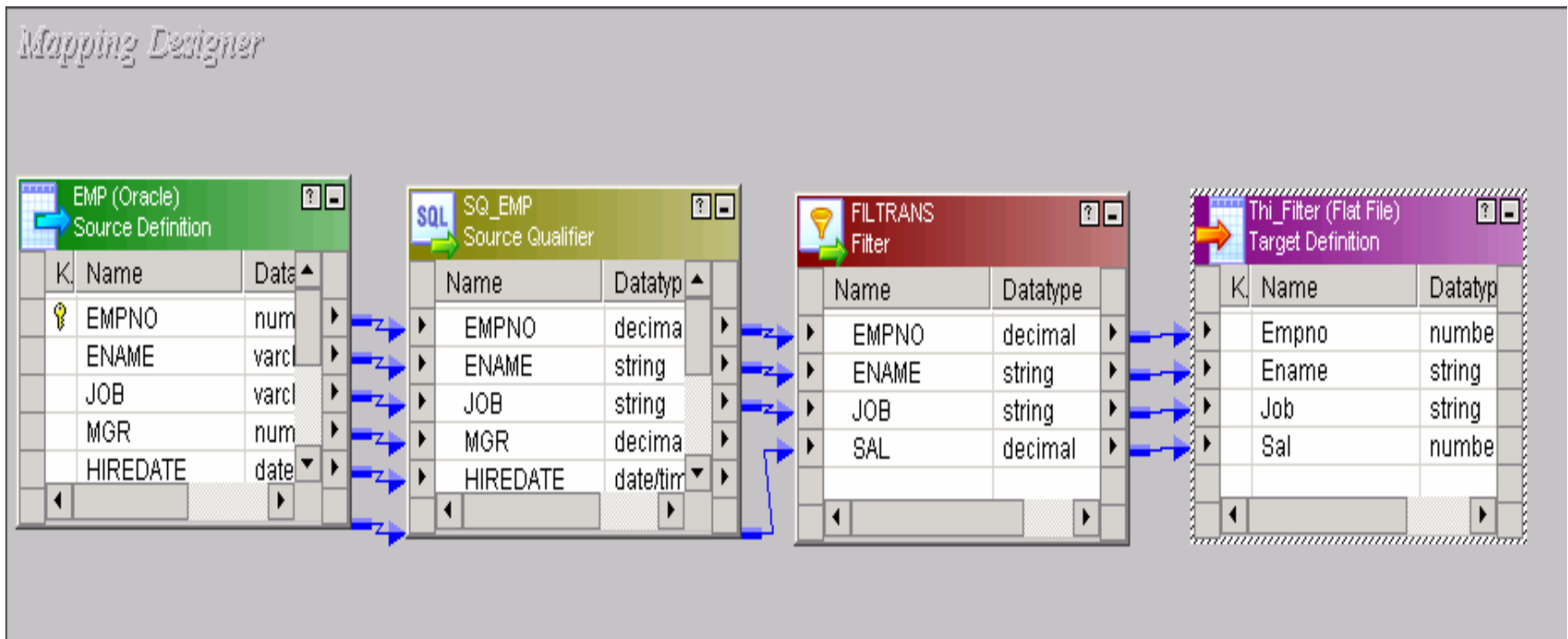
Display the session log in detail for the transformation.

Example for Filter Transformation

- The Filter Transformation allow only rows that meet the condition to pass through it and load the data into the file.



Example for Filter Transformation



Aggregator Transformation

- Aggregator transformation is an active transformation which is used to perform aggregate calculations such as sum, average on a group of rows.
- The following aggregate functions can be used with an aggregator transformations.

AVG, COUNT, FIRST, LAST, MAX, MEDIAN,
MIN, PERCENTILE, STDDEV, SUM, VARIANCE

Components of Aggregator

- Expression

Contains aggregate expressions and non-aggregate expressions based upon the output port .

- Group by port

Used to group the data other than the aggregated columns i.e non-aggregated columns

Properties

- Cache Directory

It is a local directory where the powercenter server creates the index and data caches.

- Tracing level

Display the session log in detail for the transformation.

Properties

- Sorted input

Enable this option only if the data pass to the Aggregator transformation sorted by group by port, either in ascending or descending order.

- Aggregate data cache

The PowerCenter Server stores data in the aggregate cache until it completes aggregate calculations. It stores the row data.

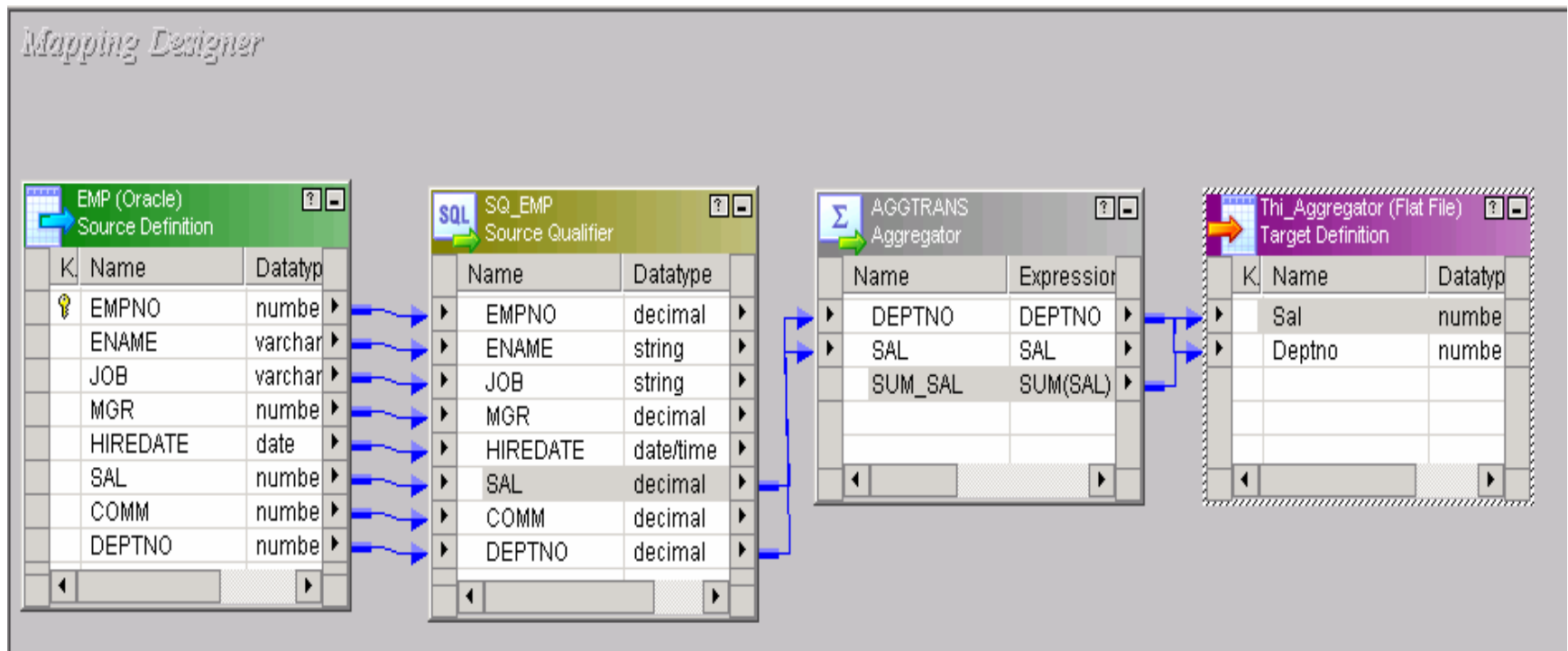
- Aggregate index cache

It stores the values in groups.

Example for Aggregator Transformation

- The Aggregator transformation, performs an aggregation function on a column grouping by the other columns with the data extracted from the DB and sends it to a file.
-

Example for Aggregator Transformation



Sorter Transformation

- Sorter transformation is an active transformation allows to sort the data either in ascending or in descending order based upon the sort key.
 - Sort the data either from relational or from flat file sources.
 - Can sort the data more than one columns.
-

Components

- Key

Enable this option corresponding to the column which u want to sort the data.

- Direction

Select the direction in which order u want to sort the data either in ascending or descending.

Properties

■ Sorter Cache Size

- ❑ The PowerCenter Server uses the Sorter Cache Size property to determine the maximum amount of memory it can allocate to perform the sort operation.
 - ❑ The PowerCenter Server passes all incoming data into the Sorter transformation before it performs the sort operation.
-

Properties

- Case Sensitive

When you enable the Case Sensitive property, the PowerCenter Server sorts uppercase characters higher than lowercase characters.

- Work Directory

You must specify a work directory the PowerCenter Server uses to create temporary files while it sorts data.

Properties

- Null Treated Low

Enable this property if you want the PowerCenter Server to treat null values as lower than any other value when it performs the sort operation. Disable this option if you want the PowerCenter Server to treat null values as higher than any other value.

Properties

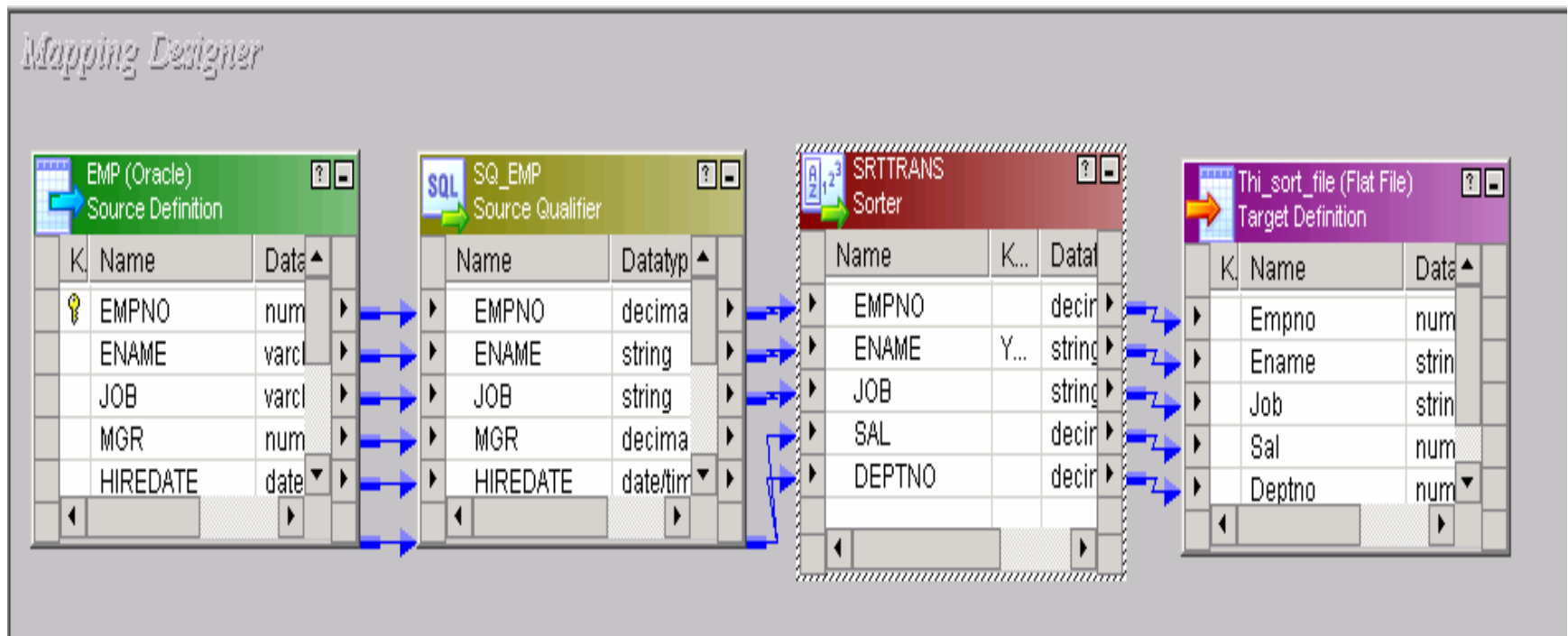
- Tracing level

Configure the Sorter transformation tracing level to control the number and type of Sorter error and status messages the PowerCenter Server writes to the session log.

Example for Sorter Transformation

- The SORT transformation, sorts the data extracted from the DB and sends it to a file.

Example for Sorter Transformation



Rank Transformation

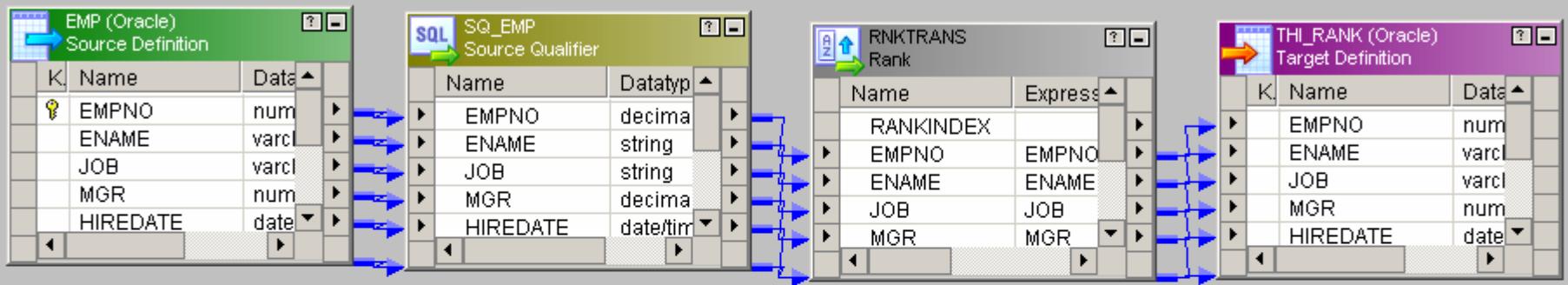
- The Rank transformation allows you to select only the top or bottom rank of data not just one value.
 - It is an Active transformation. Can select only one port to define a rank
 - It can be used return -
 - The largest or smallest numeric value in a port or group.
 - The strings at the top or the bottom of a session sort order.
-

Examples for Rank Transformation

- The Rank Transformation is used to select the top nth or bottom nth rank of data extracted from the database and load them into the target.
-

Examples for Rank Transformation

Mapping Designer



Router Transformation

- It is an Active Transformation which is used to filter the rows similar to Filter transformation.
- It is used to fetch both the satisfied conditions and unsatisfied conditions (if needed).
- The main difference between the filter and router is

In router we can fetch both the satisfied conditions and unsatisfied conditions.

In Filter we can fetch only the satisfied conditions .

Example for Router Transformation

- The Router Transformation routes the extracted data from the database and sends it to one or more different files.
 - Case: When you get some 3rd party records through some files, you may reject some records because of some business reasons. In this case we can router and capture those records in a file and send it back to customer through email to correct the information.
-

Example for Router Transformation

Mapping Designer

K	Name	Datatype
	EMPNO	num
	ENAME	varcl
	JOB	varcl
	MGR	num
	HIREDATE	date

Name	Datatype
EMPNO	decima
ENAME	string
JOB	string
MGR	decima
HIREDATE	date/tim

Name	Datatype
INPUT	
EMPNO	decimal
ENAME	string
JOB	string
SAL	decimal
Sal1	
EMPNO1	decimal
ENAME1	string
JOB1	string
SAL1	decimal
Sal2	
EMPNO3	decimal
ENAME3	string
JOB3	string
SAL3	decimal
DEFAULT	
EMPNO2	decimal
ENAME2	string
JOB2	string
SAL2	decimal

K	Name	Datatype
	Empno	numbe
	Ename	string
	Job	string
	Sal	numbe

K	Name	Datatype
	Empno	numbe
	Ename	string
	Job	string
	Sal	numbe

K	Name	Datatype
	Empno	numbe
	Ename	string
	Job	string
	Sal	numbe

Mapping Designer

EMP (Oracle) Source Definition		
K	Name	Data
PK	EMPNO	num
	ENAME	varcl
	JOB	varcl
	MGR	num
	HIREDATE	date

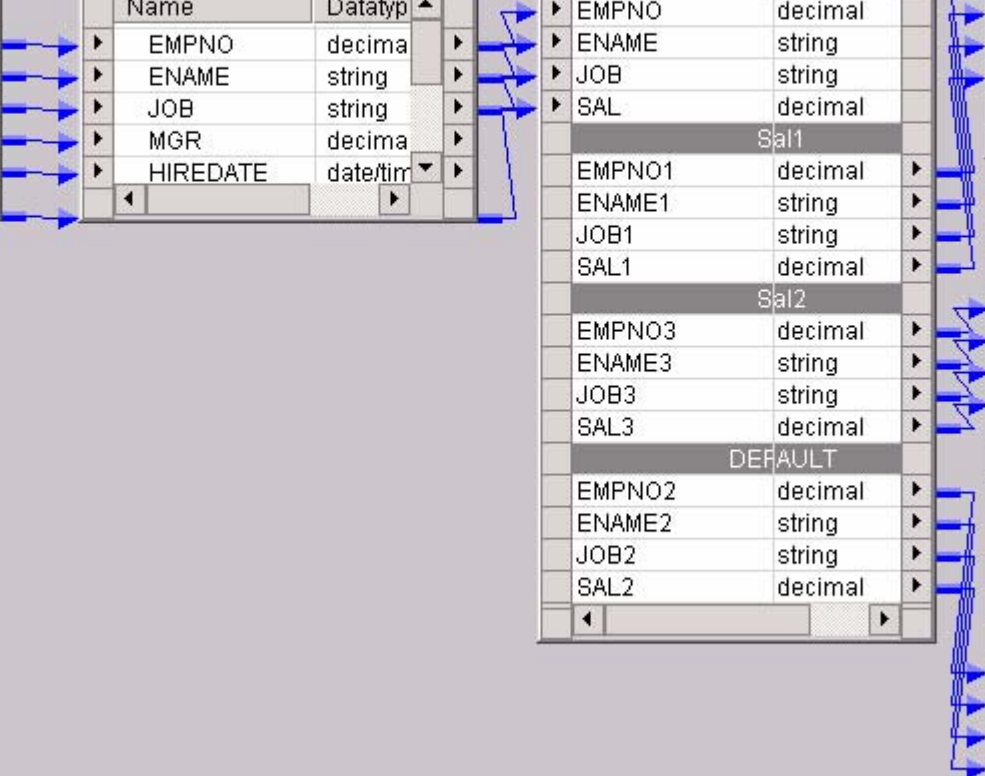
SQL SQ_EMP Source Qualifier		
	Name	Datatypes
	EMPNO	decima
	ENAME	string
	JOB	string
	MGR	decima
	HIREDATE	date/tim

RTRTRANS Router	
Name	Datatype
INPUT	
▶ EMPNO	decimal
▶ ENAME	string
▶ JOB	string
▶ SAL	decimal
Sal1	
▶ EMPNO1	decimal
▶ ENAME1	string
▶ JOB1	string
▶ SAL1	decimal
Sal2	
▶ EMPNO3	decimal
▶ ENAME3	string
▶ JOB3	string
▶ SAL3	decimal
DEFAULT	
▶ EMPNO2	decimal
▶ ENAME2	string
▶ JOB2	string
▶ SAL2	decimal

Thi_Router1 (Flat File) Target Definition		
K	Name	Datatypes
▶	Empno	numbe
▶	Ename	string
▶	Job	string
▶	Sal	numbe

Thi_Router2 (Flat File) Target Definition		
K	Name	Datatypes
▶	Empno	numbe
▶	Ename	string
▶	Job	string
▶	Sal	numbe

Thi_Router3 (Flat File) Target Definition		
K	Name	Datatypes
▶	Empno	numbe
▶	Ename	string
▶	Job	string
▶	Sal	numbe



Joiner Transformation

- Joiner transformation is used to join the source data from two related heterogeneous sources residing in different locations or file systems. Or, you can join data from the same source.
 - It is an Active Transformation.
 - A join is a relational operator that combines data from multiple tables into a single result set.
-

Properties

- Join Type :

We can define the join type on the Properties tab in the transformation.

The various join types are

- 1) Normal join
 - 2) Master outer join
 - 3) Detail outer join
 - 4) Full outer join
-

Properties

- Normal Join :

Discards all rows of data from the master and detail source that do not match, based on the condition.

- Master outer Join :

A master outer join keeps all rows of data from the detail source and the matching rows from the master source. It discards the unmatched rows from the master source.

Properties

- Detail outer Join :

A detail outer join keeps all rows of data from the master source and the matching rows from the detail source. It discards the unmatched rows from the detail source.

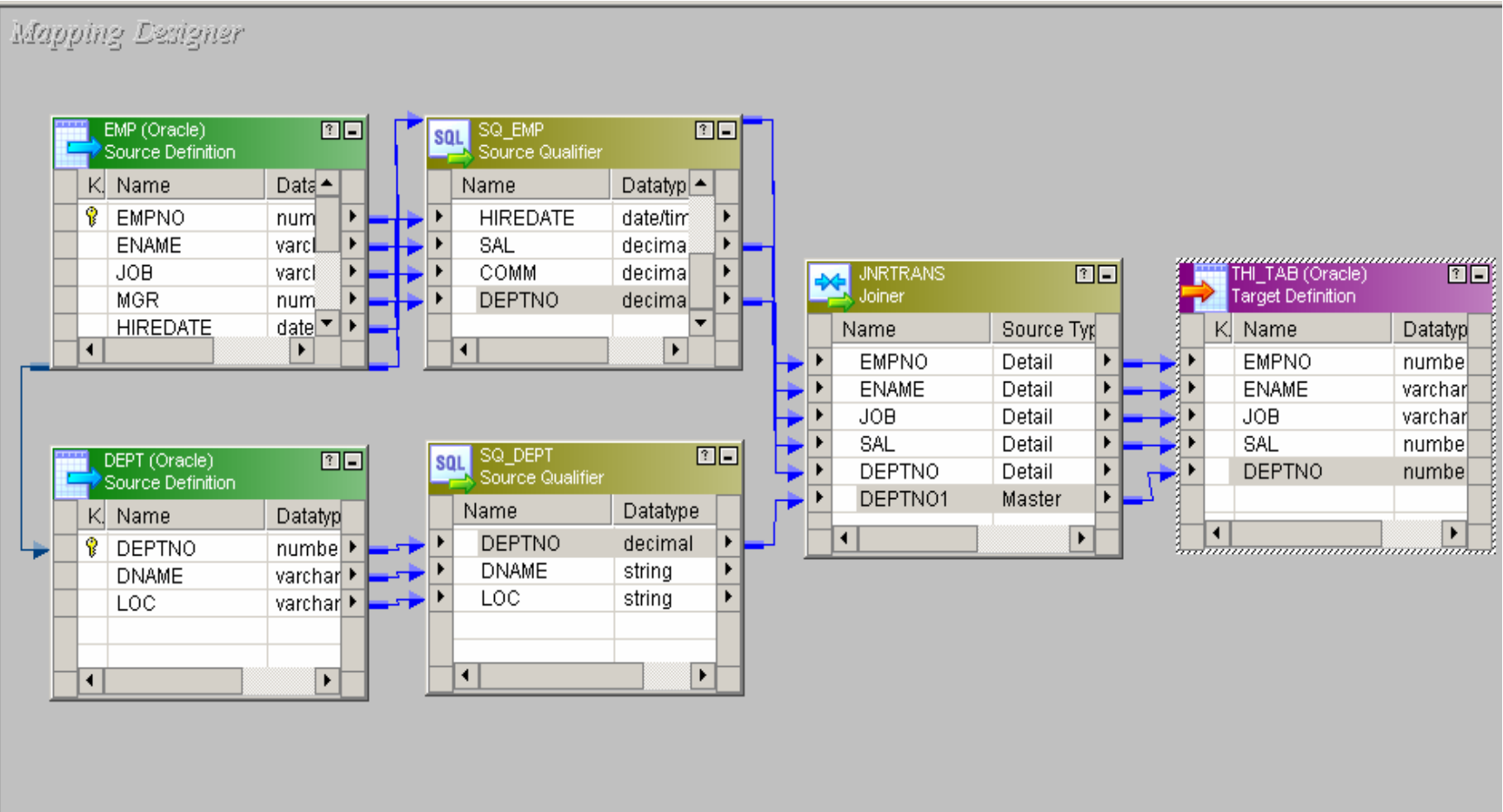
- Full outer Join :

A full outer join keeps all rows of data from both the master and detail sources.

Example for Joiner Transformation

- The Joiner Transformation is used to join the tables data extracted from the database based on the join condition and send it to the target table.
-

Example for Joiner Transformation

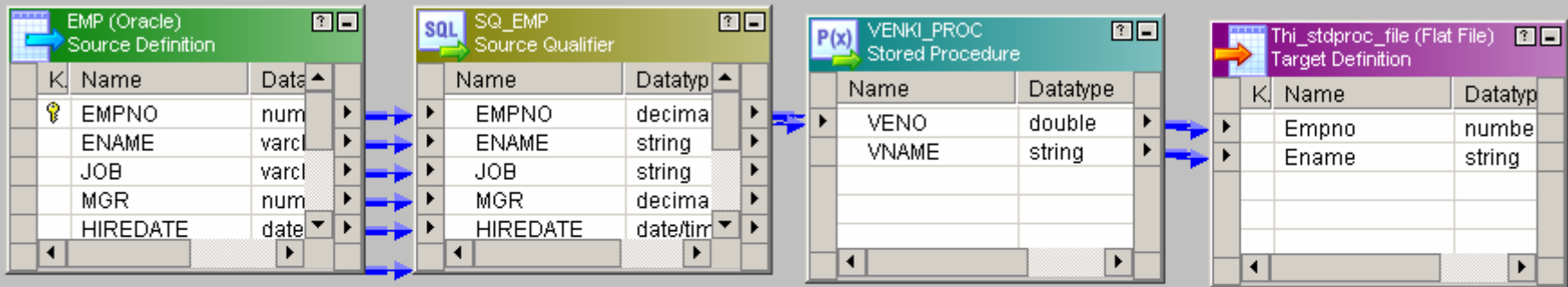


Stored Procedure

- A Stored Procedure transformation is an important tool for populating and maintaining databases
 - Stored procedure is a passive transformation.
 - The stored procedure must exist in the database before creating a Stored Procedure transformation
 - One of the most useful features of stored procedures is the ability to send data to the stored procedure, and receive data from the stored procedure
-

Example for Stored procedure

Mapping Designer



Sequence Generator Transformation

- The sequence generator transformation generates numeric values.
 - It contains two o/p ports that can be connected to one or more transformations. The informatica server generates a value each time a row enters a connected transformation , even if that value is not used.
 - The sequence generator is reusable and use it in multiple mappings .
 - It is a passive transformation.
-

Sequence Generator Transformation

- It can be used to
 - Create unique primary key values
 - replace missing primary keys

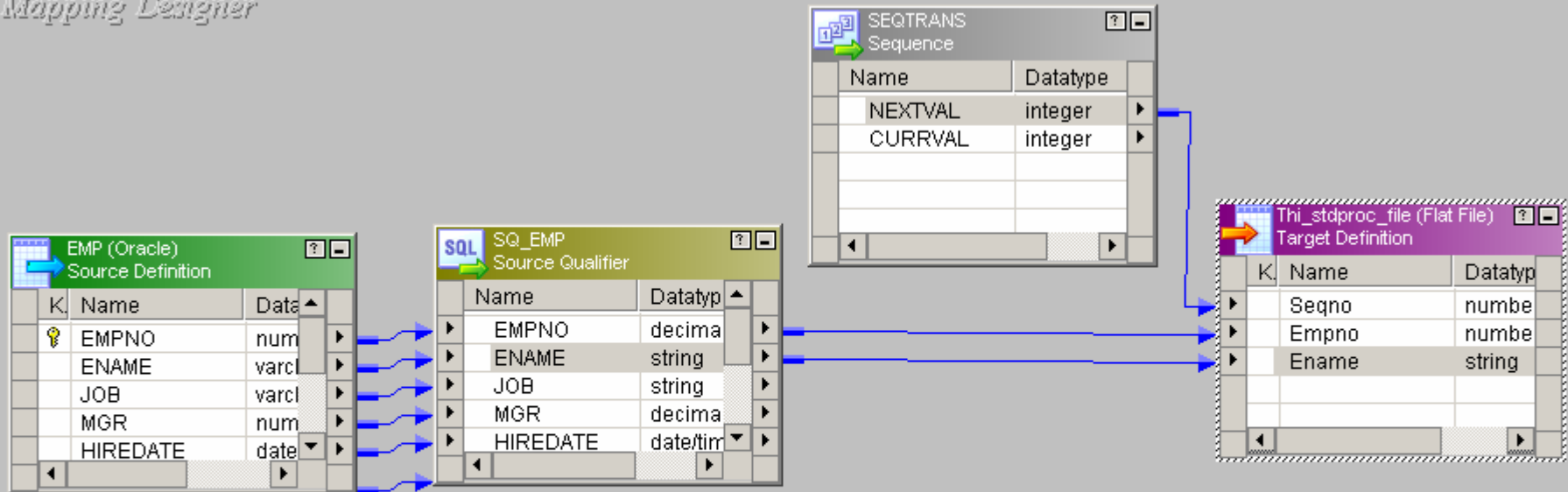


Example for Sequence Generator Transformation

- Sequence generator is used to generate the sequence values to the data extracted from the database and load them into the target.
-

Example for Sequence Generator Transformation

Mapping Designer



Source Qualifier Transformation

- When we add relational or flat file source definition to a mapping, you need to connect to a source qualifier transformation.
 - It is an Active Transformation.
 - Can use the Source Qualifier to perform the following tasks:
 - Join data originating from the same source database.
-

Source Qualifier Transformation

Filter records when the Informatica Server reads source data.

Specify an outer join rather than the default inner join.

Specify sorted ports.

Select only distinct values from the source.

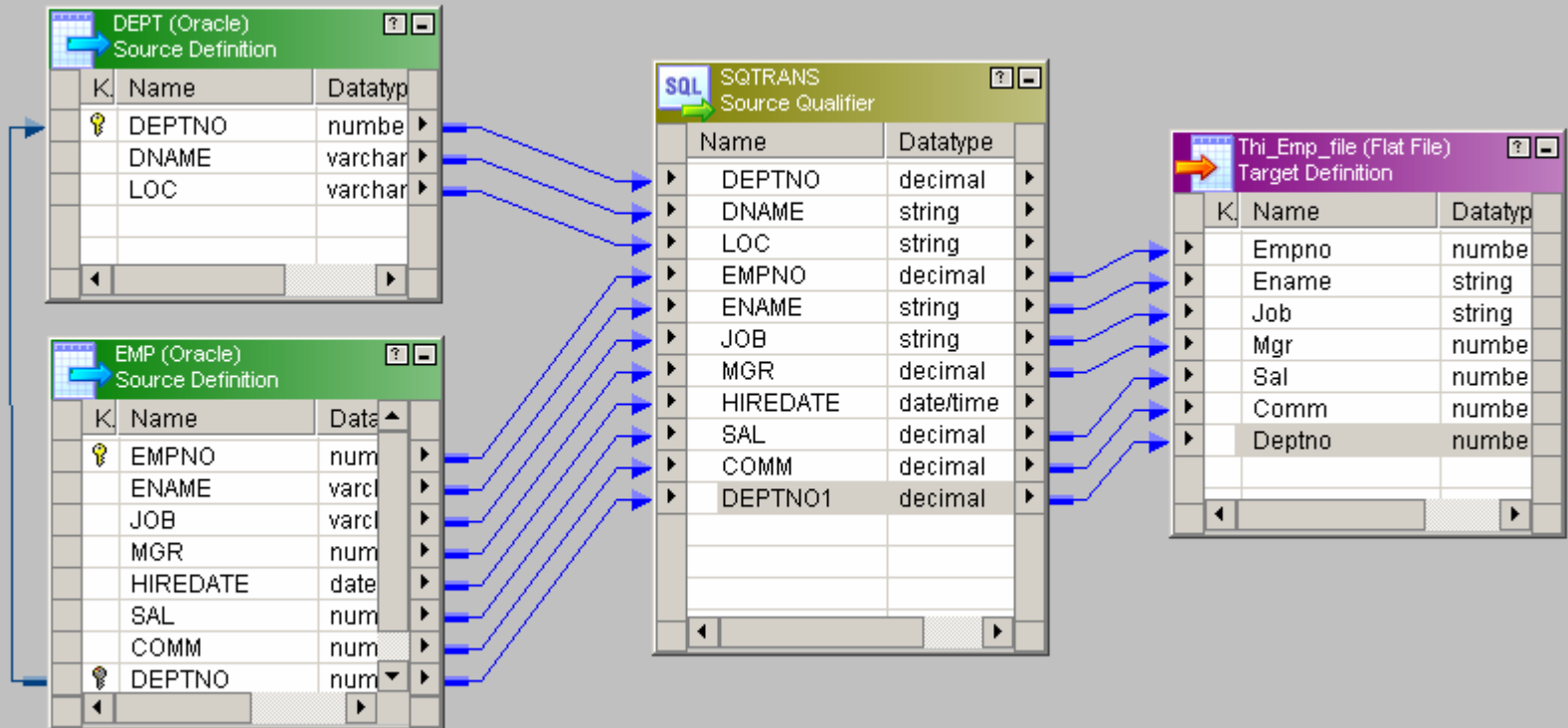
Example for Source Qualifier Transformation

- Source Qualifier transformation is used to Combine the data from two sources and load them into the target.



Example for Source Qualifier Transformation

Mapping Designer



Union Transformation

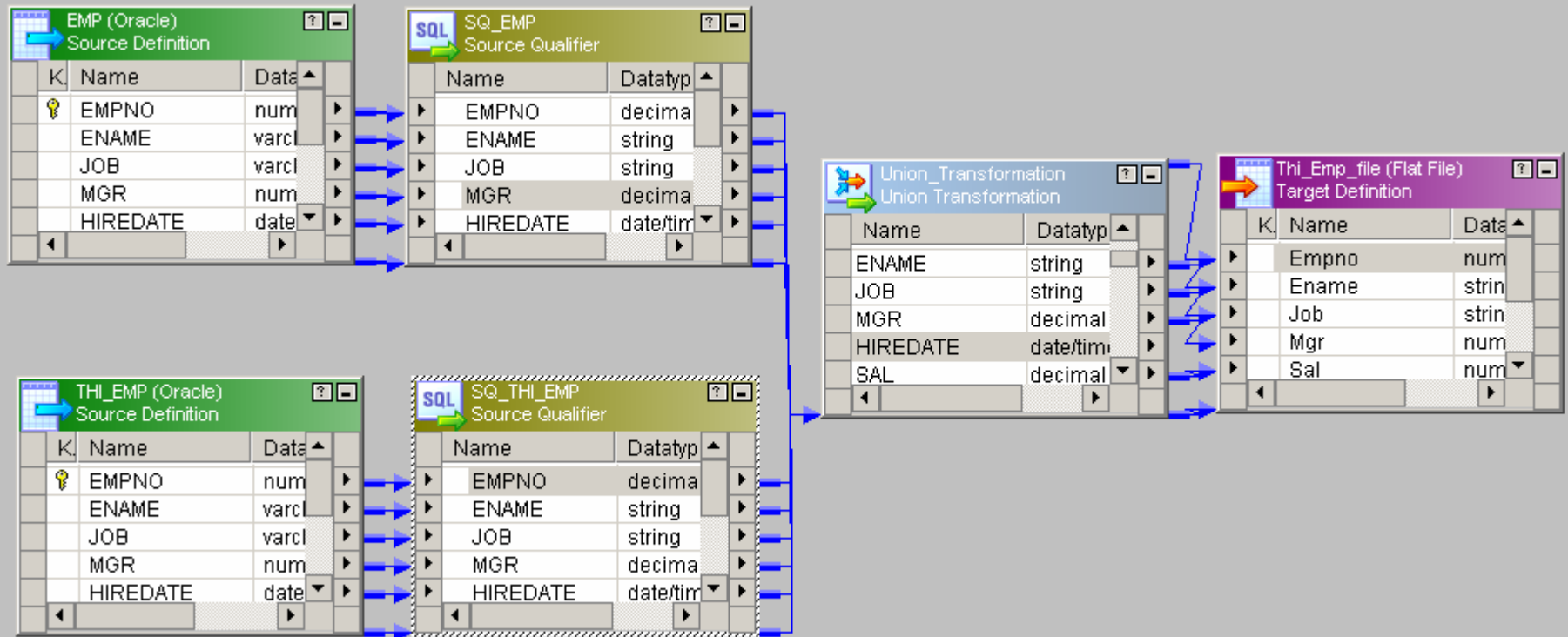
- Union Transformation is used to merge data from two or more sources.
 - It is an Active transformation.
 - It is similar to Union All sql statement to combine the results of two or more sql statements.
 - It does not remove duplicate rows
-

Example for Union Transformation

- Union transformation is used to merge the two or more source data extracted from the database and load them into the target
-

Example for Union Transformation

Mapping Designer



Lookup Transformation

- Lookup transformation is used in mapping to lookup data in a flat file or relational table, view, or synonym.
 - Can use multiple lookup transformations in a mapping.
 - It is a passive transformation.
 - Can use the lookup transformations to perform the following tasks.
-

Lookup Transformation

Get a related value :

For example, your source includes employee ID, but you want to include the employee name in your target table to make your summary data easier to read.

Update slowly changing dimension tables :

You can use a Lookup transformation to determine whether rows already exist in the target.

Connected Transformation

- Connected Transformation

- Receives input values directly from another transformation in the pipeline.

- Passes return values from the query to the next transformation.

Unconnected Transformation

- Unconnected Transformation

With unconnected Lookups, you can pass multiple input values into the transformation, but only one column of data out of the transformation.

Use the return port to specify the return value in an unconnected lookup transformation

Example for Lookup Transformation

- Lookup transformation is used to lookup the data in a flat file or relational data extracted from the database and load them into the target.
-

Example for Lookup Transformation

Mapping Designer

THI_EMP (Oracle) Source Definition		
K	Name	Datatype
	EMPNO	num
	ENAME	varcl
	JOB	varcl
	MGR	num
	HIREDATE	date

LKPTRANS Lookup Procedure		
Name	Datatype	
▶	EMPNO1	decimal
▶	ENAME1	string
▶	JOB1	string
▶	MGR1	decimal
▶	HIREDATE1	date/time
▶	SAL1	decimal
▶	COMM1	decimal
▶	DEPTNO1	decimal
▶	EMPNO	decimal
▶	ENAME	string
▶	JOB	string
▶	MGR	decimal
▶	HIREDATE	date/time
▶	SAL	decimal
▶	COMM	decimal
▶	DEPTNO	decimal

EXPTRANS Expression		
Name	Expression	
▶	EMPNO	EMPNO
▶	cond	DECODE(
▶	EMPNO1	EMPNO1
▶	ENAME1	ENAME1
▶	JOB1	JOB1
▶	MGR1	MGR1
▶	HIREDATE1	HIREDATE
▶	SAL1	SAL1
▶	COMM1	COMM1
▶	DEPTNO1	DEPTNO1

EMP (Oracle) Target Definition		
K	Name	Datatype
▶	EMPNO	numbe
▶	ENAME	varchar
▶	JOB	varchar
▶	MGR	numbe
▶	HIREDATE	date
▶	SAL	numbe
▶	COMM	numbe
▶	DEPTNO	numbe

SQ_THI_EMP Source Qualifier		
Name	Datatype	
▶	EMPNO	decima
▶	ENAME	string
▶	JOB	string
▶	MGR	decima
▶	HIREDATE	date/tim

RTRTRANS Router		
Name	Datatype	
	cond	
▶	cond1	decimal
▶	EMPNO11	decimal
▶	ENAME11	string
▶	JOB11	string
▶	MGR11	decimal
▶	HIREDATE11	date/tim
▶	SAL11	decimal
▶	COMM11	decimal
▶	DEPTNO11	decimal

Update strategy Transformation

- Update strategy transformation determines whether to insert, update, delete, or reject rows.
 - It is an active transformation.
 - Example : When a customer address changes, you may want to save the original address in the table instead of updating that portion of the customer row. In this case, you would create a new row containing the updated address, and preserve the original row with the old customer address. This illustrates how you might store historical information in a target table.
-

Update strategy Transformation

- Constants for each database operation

Operations	Constant	Numeric Value
Insert	DD_Insert	0
Update	DD_Update	1
Delete	DD_Delete	2
Reject	DD_Reject	3

Example for Update strategy Transformation

- Update strategy Transformation is used to insert ,delete ,reject the data, or update the existing data extracted from the database and load them into the target.
-

Example for Update strategy Transformation

Mapping Designer

