

Semarchy

xDI DEV

C Delving Mapping



xDI DEV

C Delving Mapping

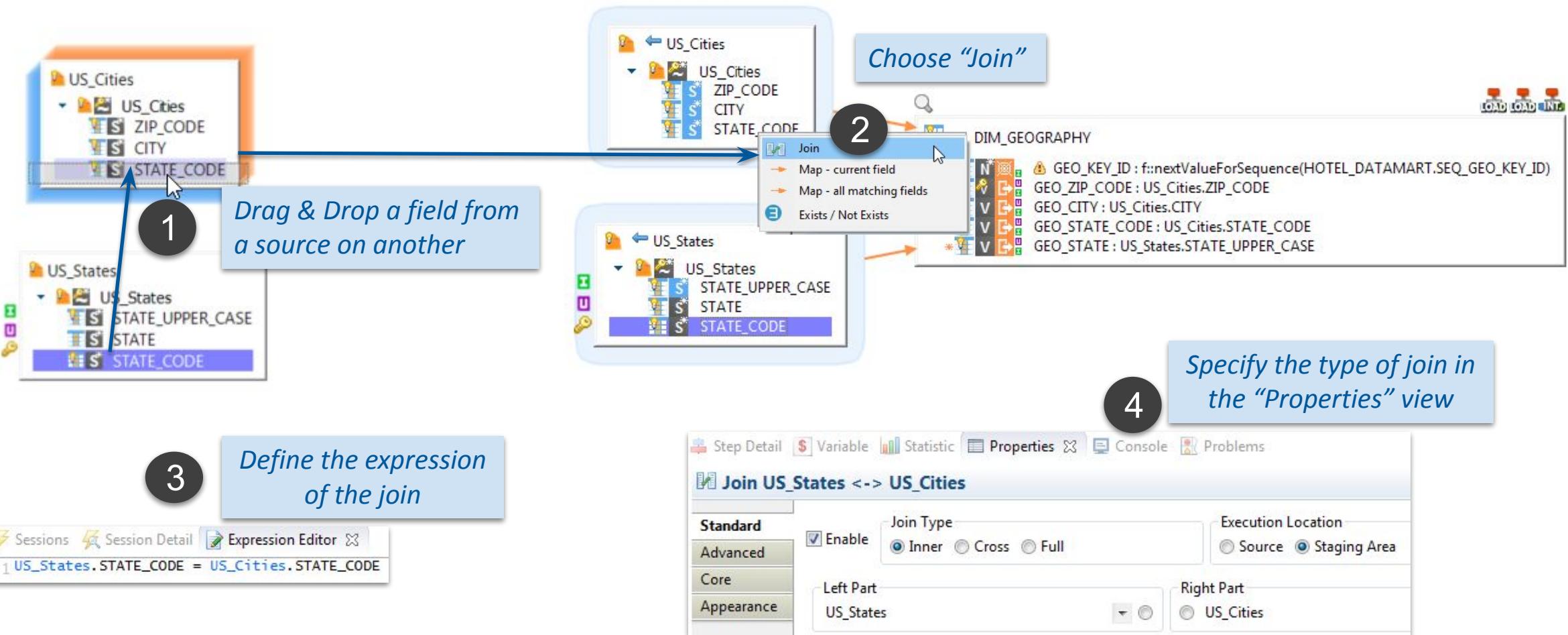
C1 - Go deeper in the Mappings



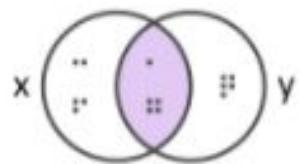
Join source data sets



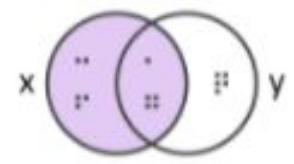
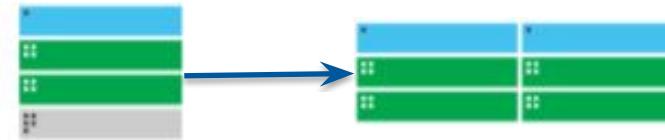
Defining joins



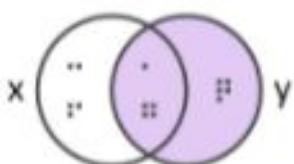
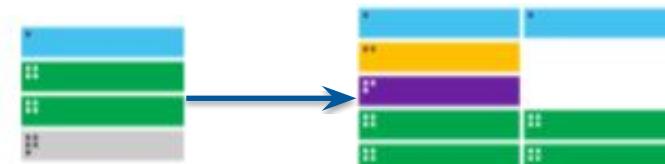
Join types



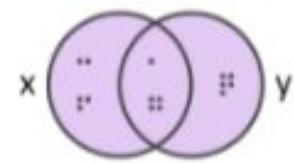
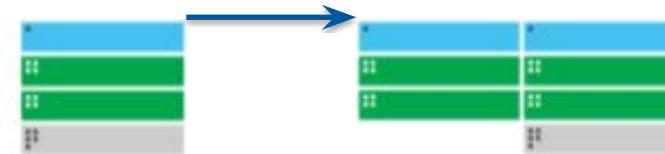
x (inner) join y
on/using <condition>



x left (outer) join y
on/using <condition>



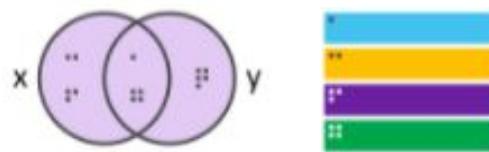
x right (outer) join y
on/using <condition>



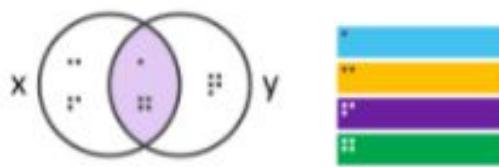
x full (outer) join y
on/using <condition>



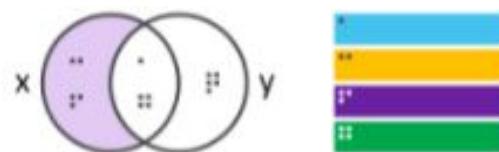
Join types



$x \text{ cross join } y$
or
 x, y



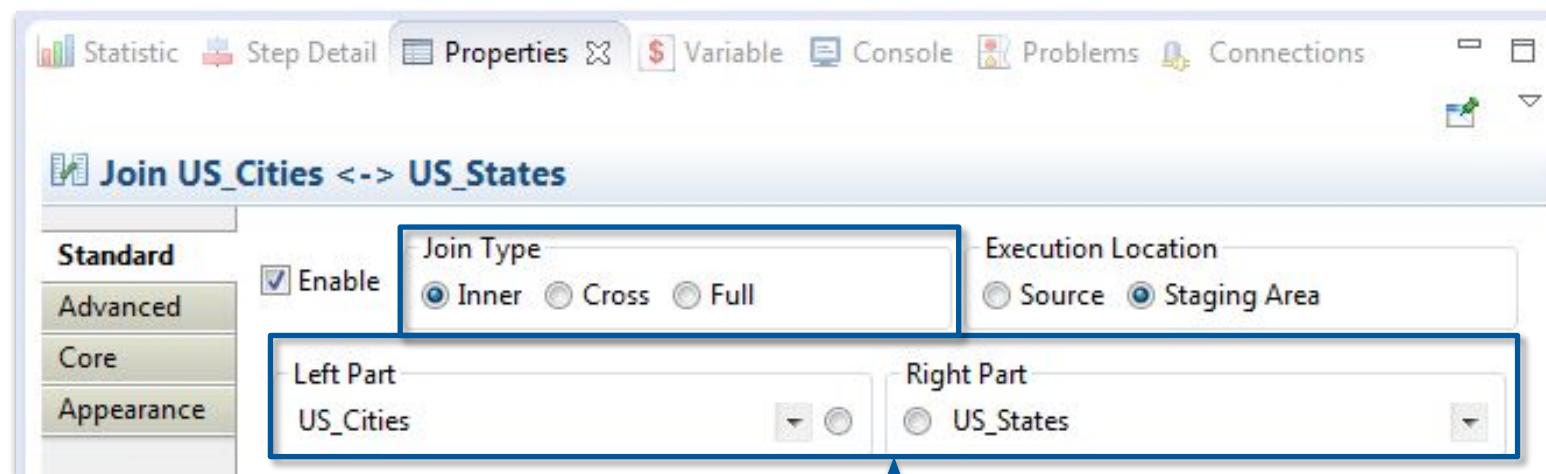
Semi-Join:
... from x where **exists**
(... from y where <cond.>)



Anti-Join:
... from x where **not exists**
(... from y where <cond.>)



Join properties

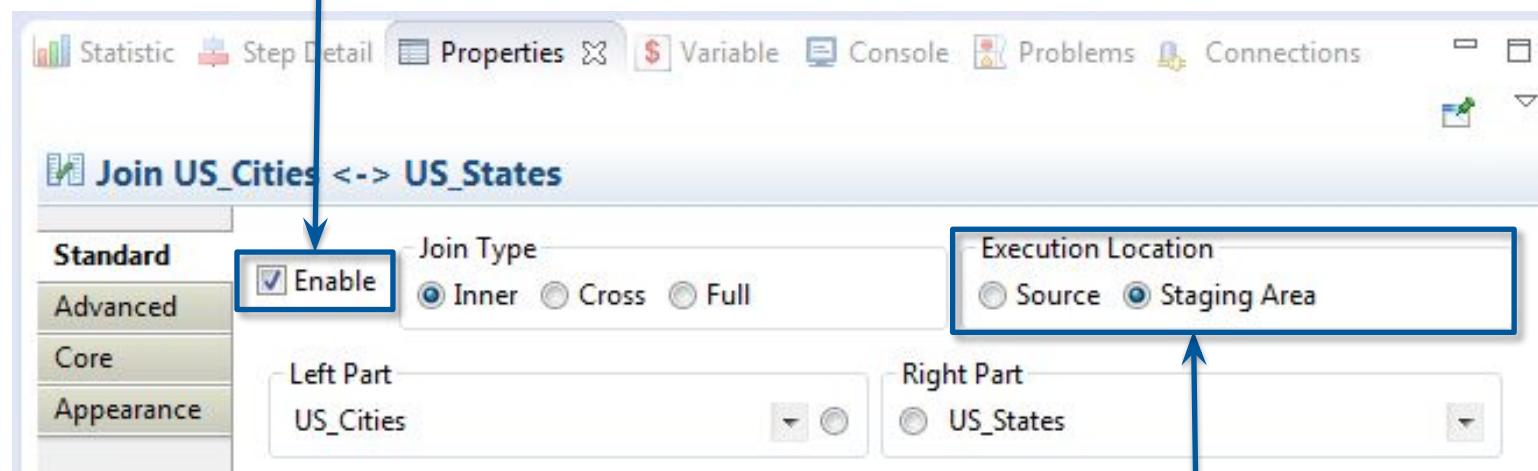


Join type

- **Inner** : Inner join between the tables
- **Cross** : Cross join (the expression won't be used)
- **Full** : Full outer join between the tables
- **Left / Right Part** : Click on the radio button to choose the right or left part

Join properties

**Enable / disable
the expression**



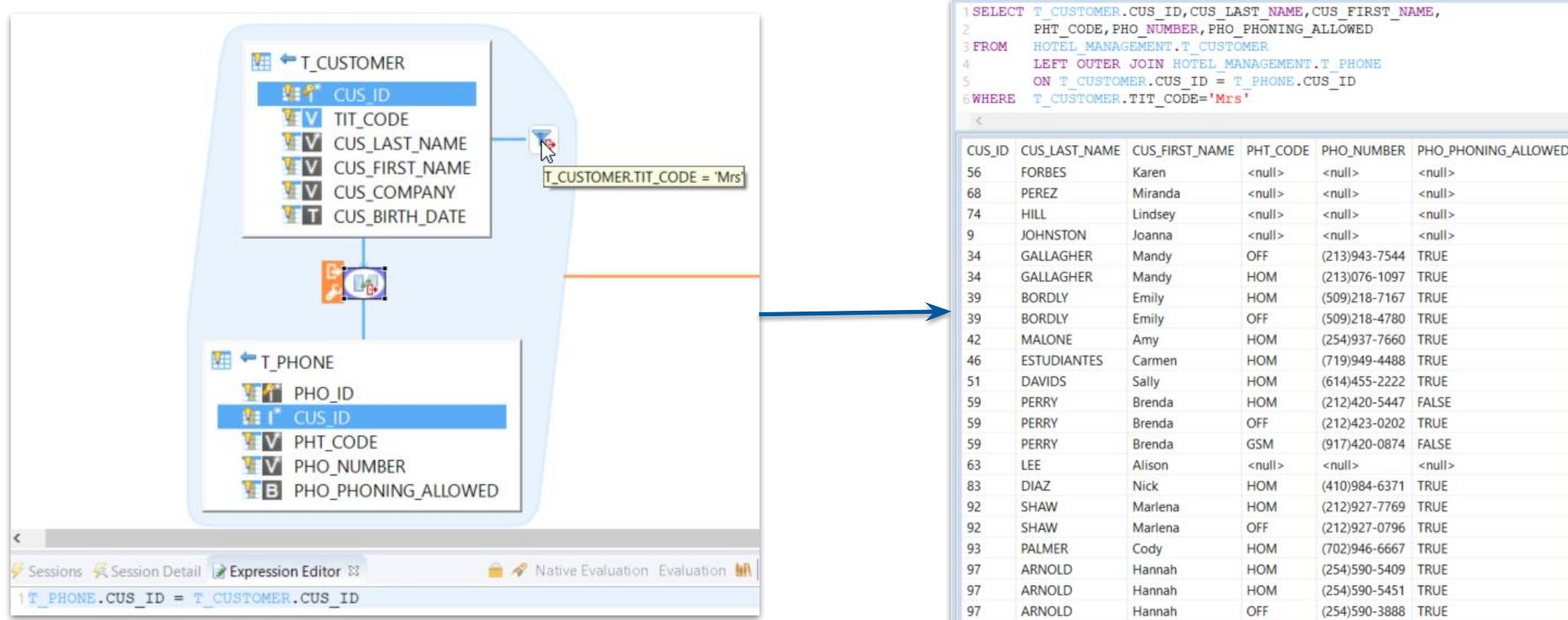
Location of the execution

- **Source** : Expression executed during the select on the source
- **Staging Area** : Expression executed during the select in the staging area

Outer Join and restrictions

If a restriction must be added on a source table with an “Outer Join”

- on the side of the master table (T_CUSTOMER in the sample): no particular issue



Outer Join and restrictions

If a restriction must be added on a source table with an “Outer Join”

- not on the side of the master table (master table is T_CUSTOMER in the sample), set the restriction in the join clause instead of in the filter





Demo

Inner & Outer Joins





Practice exercise

Create the Load DIM_Customer mapping

- Use “Computed Fields”
- Use “User Defined Functions”
- Use “Activated filters”

T ogether
E veryone
A chieves
M ore





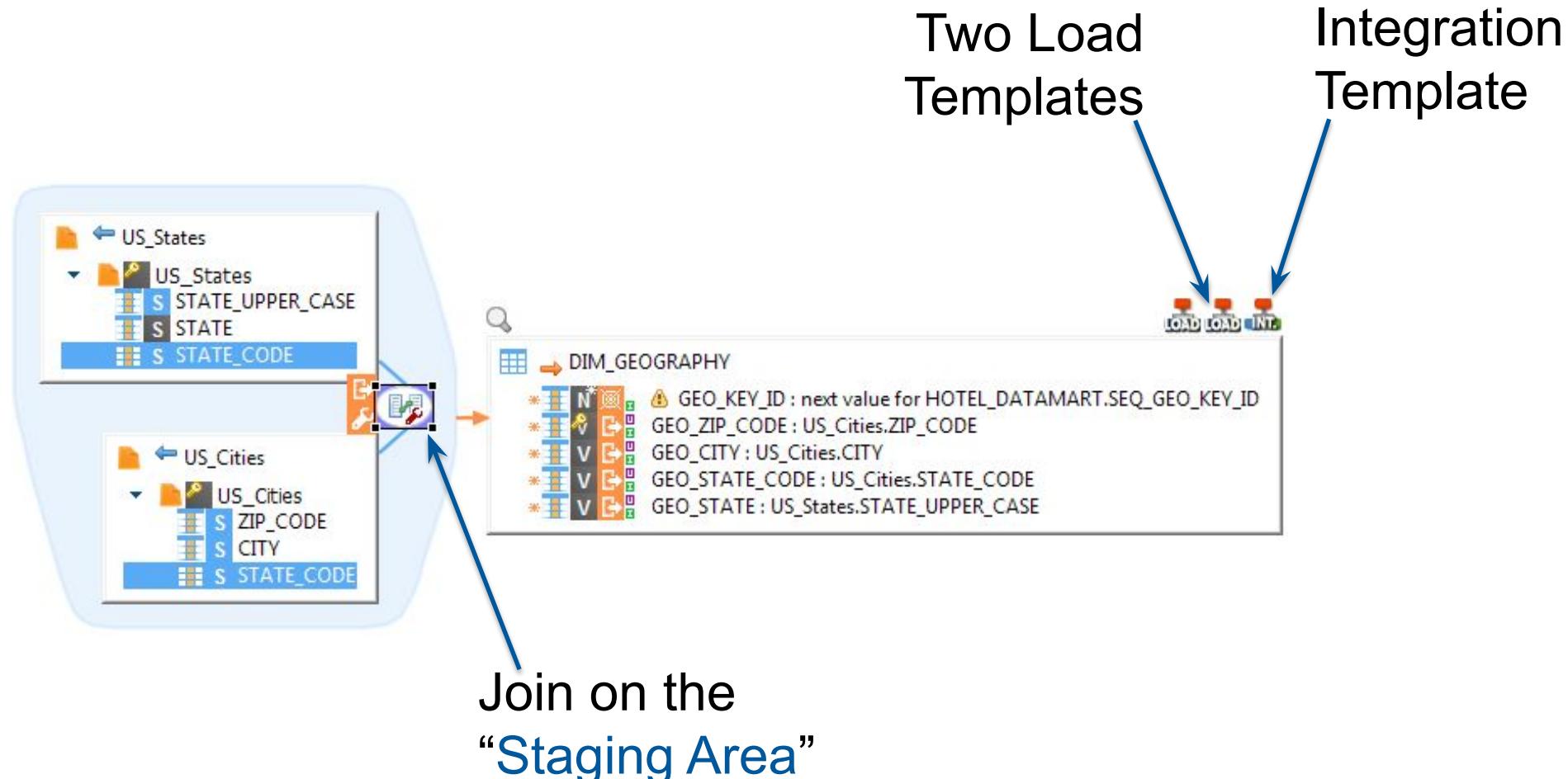
Related tutorial exercises

Creation a Mapping

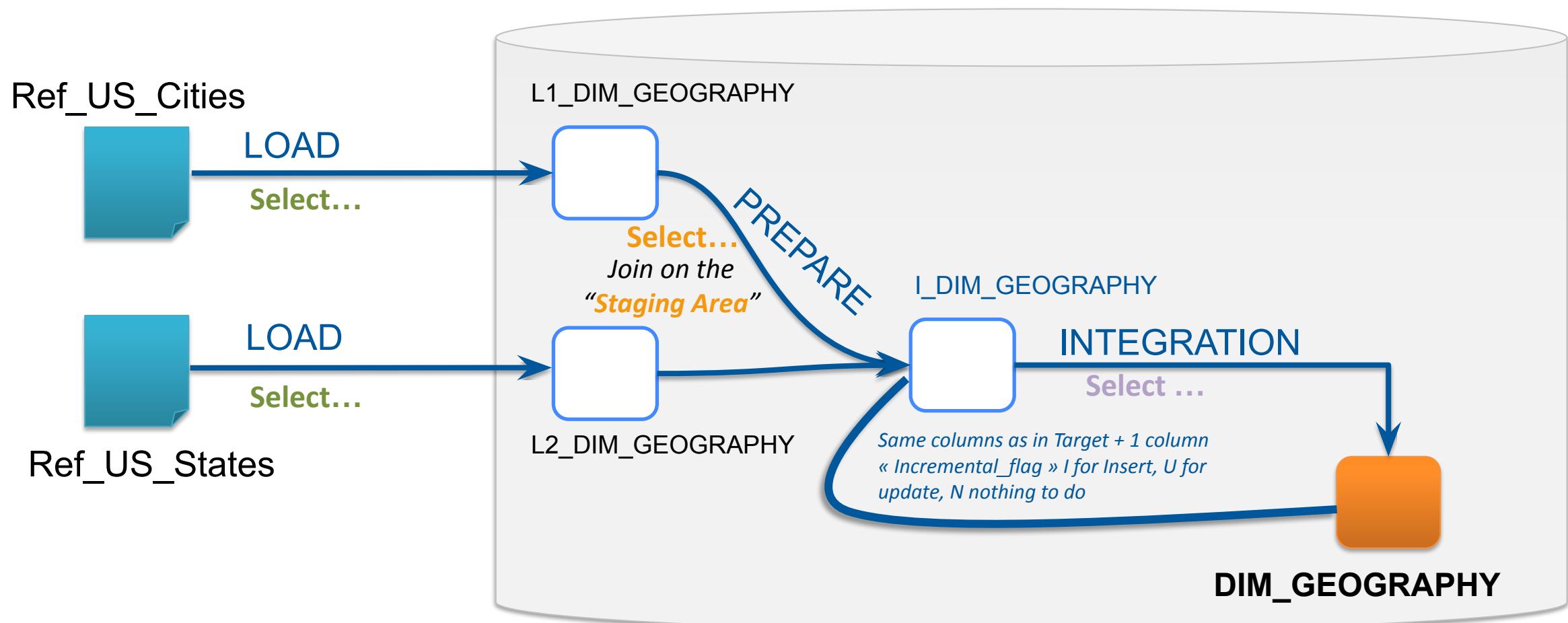
- with joins
- with Outer joins



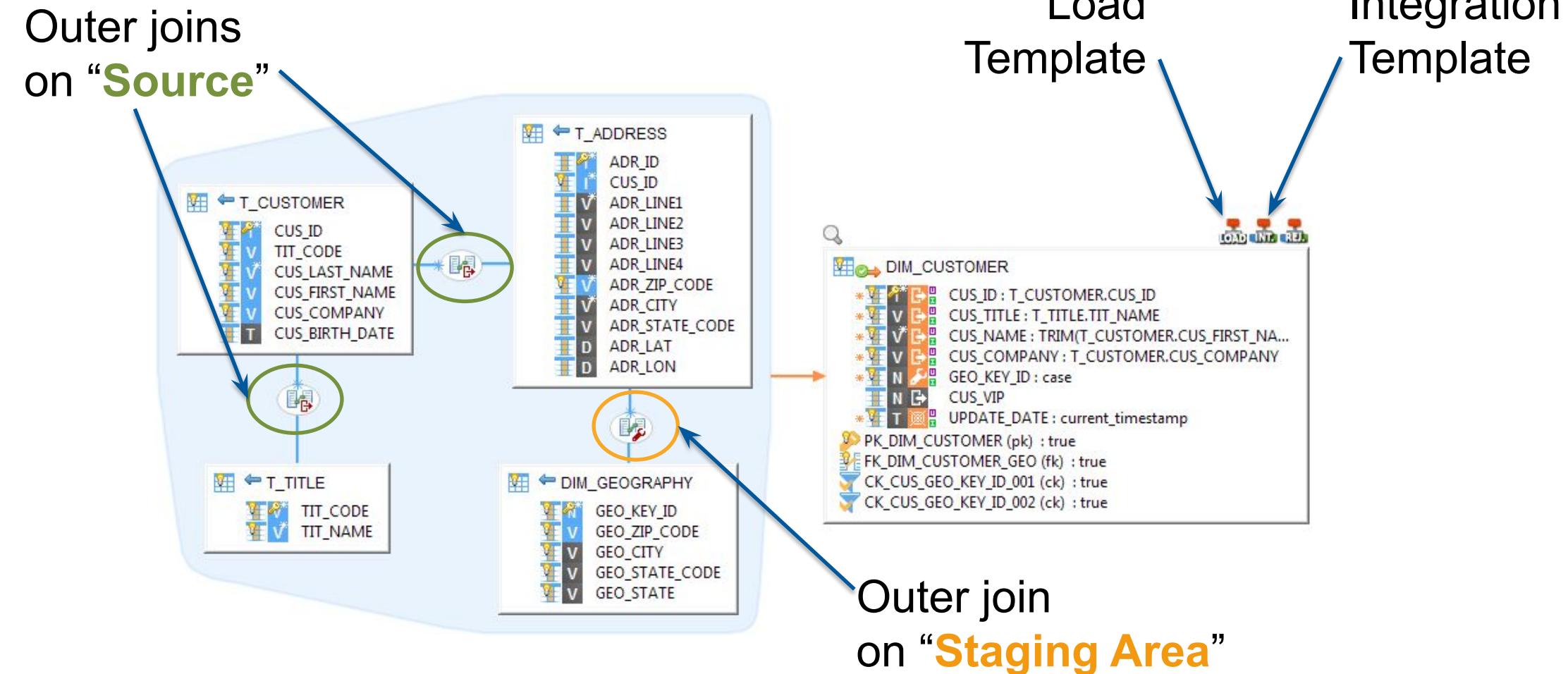
Analyzing the result on “Load DIM_GEOGRAPHY” mapping



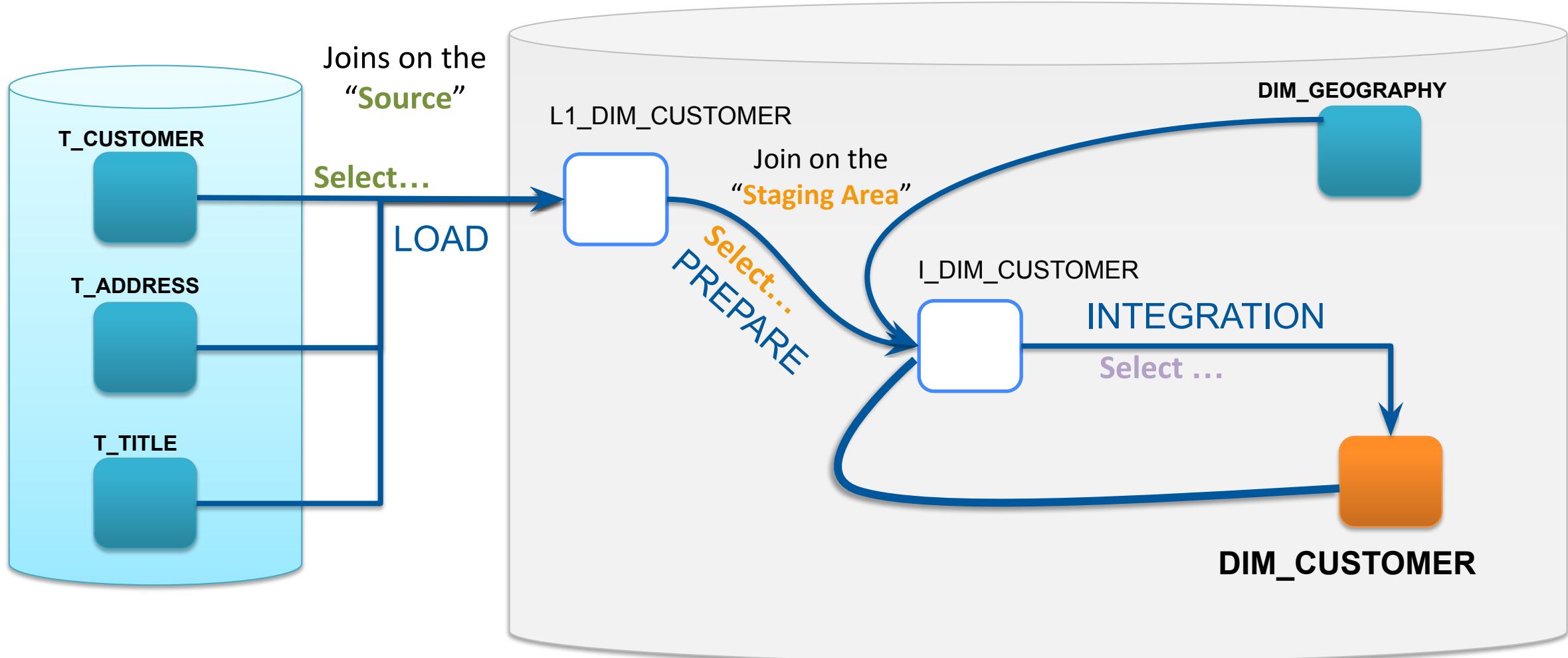
Understanding the result



Analyzing the result : “Load DIM_CUSTOMER” mapping



Understanding the result



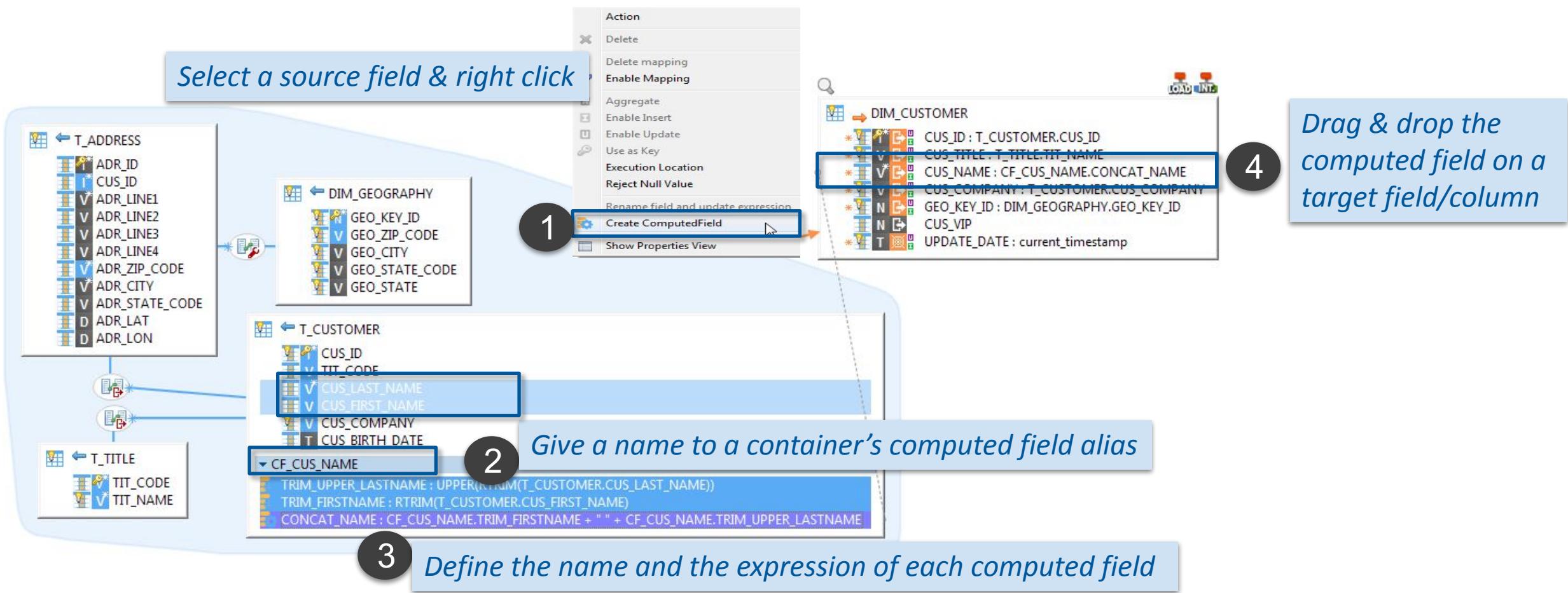


Computed fields and User
defined functions



Computed fields in a mapping

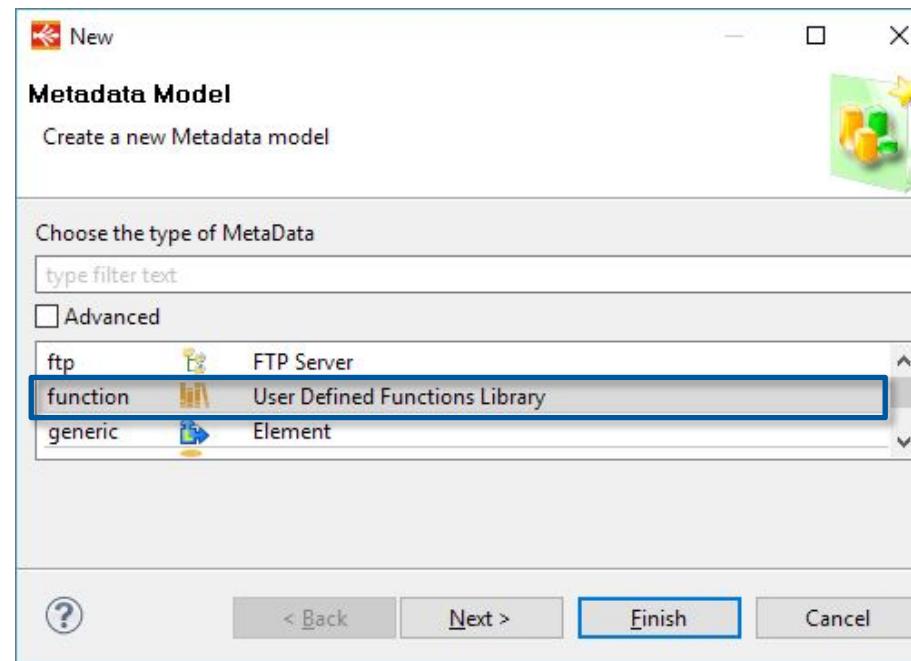
It is possible to add computed fields in the source of a mapping



User Defined Functions

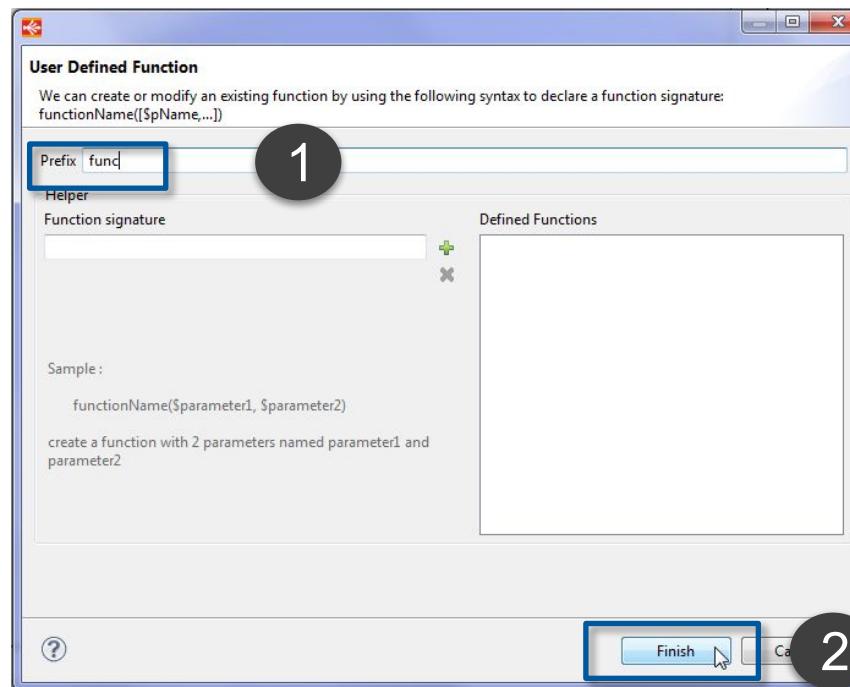
Used to create functions libraries

- Generated code can depend on the technology
- Can be used to define/centralize complex SQL expressions



User Defined Functions - Metadata

Define a prefix for the list of functions grouped in a given metadata

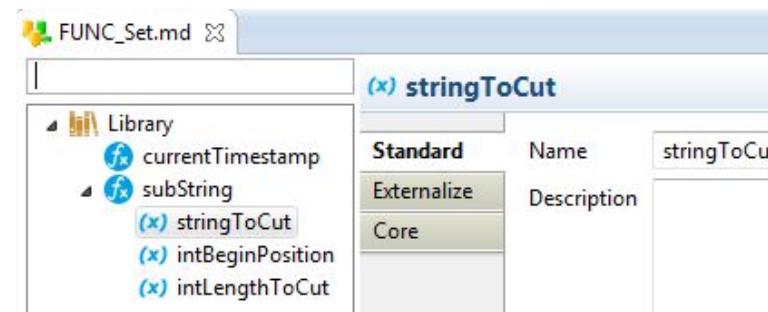


```
func::my_function(my_param1,my_param2)
```

User Defined Functions - Parameters

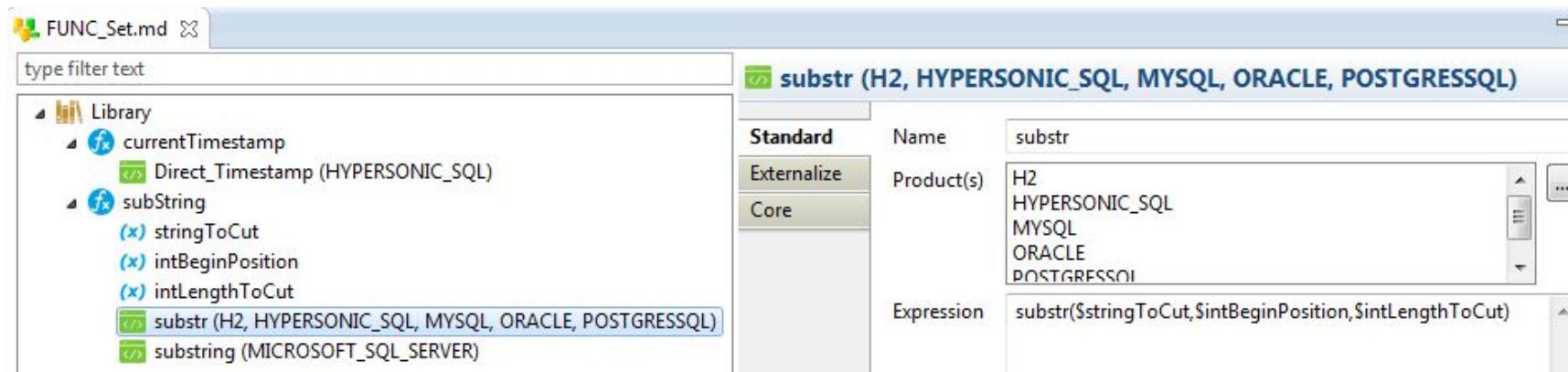
To add a function select **New > Function** on the root node. Give it a name and optionally a description

- *Then, create input and output parameters : right click on the function and select **New > Parameter***



User Defined Functions - Implementation

- Right click on the function and select
 - New > Implementation

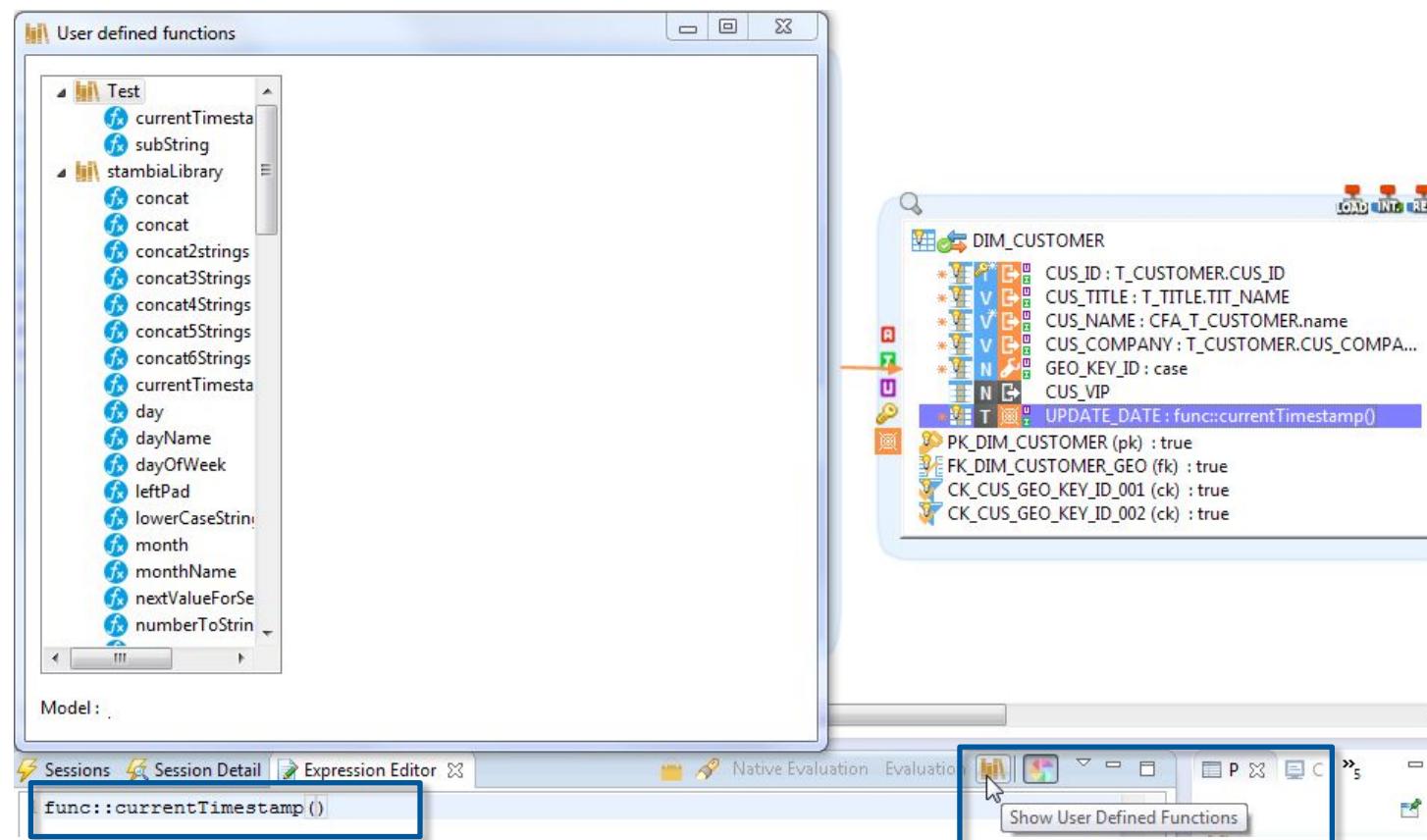


- In the Product(s) box, you'll specify the technologies the implementation will be used for

When using the function in a Mapping, the right implementation will be automatically selected

User Defined Functions - Use

You can use auto completion (Ctrl – space) or open the UDF window as shown below.





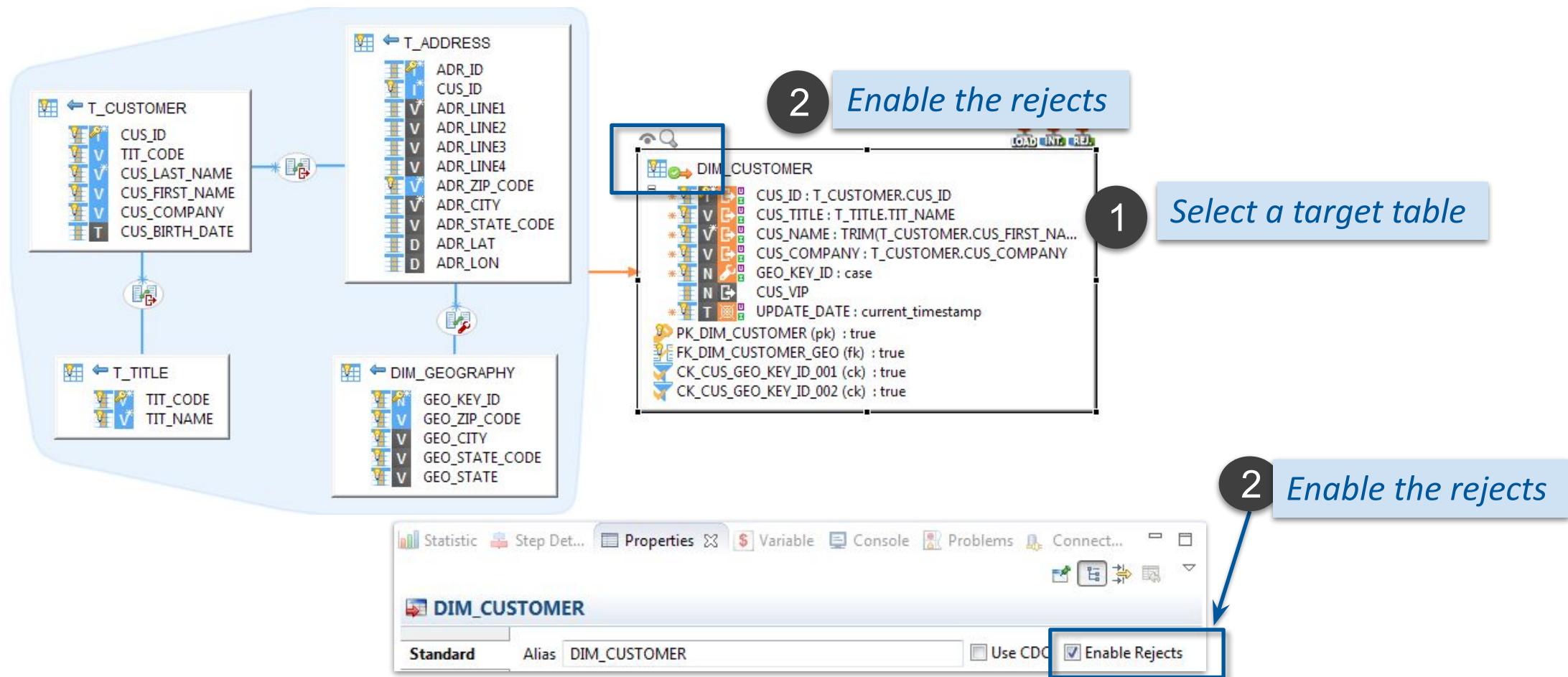
Reject management

<https://www.youtube.com/watch?v=iJPSVGjDUpM&feature=youtu.be>



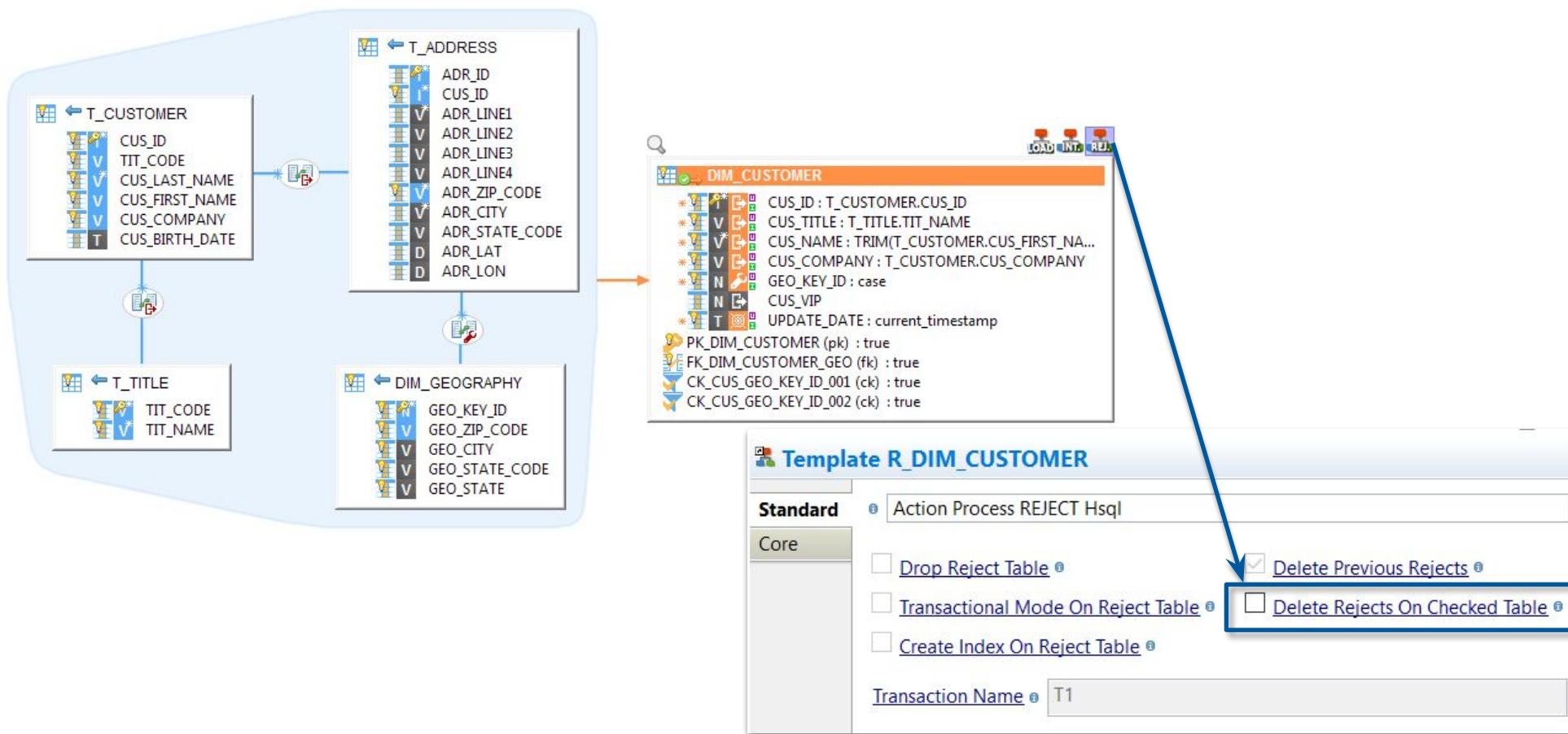
Managing the rejects

When designing the mappings, it is possible to activate the « Rejects Management »



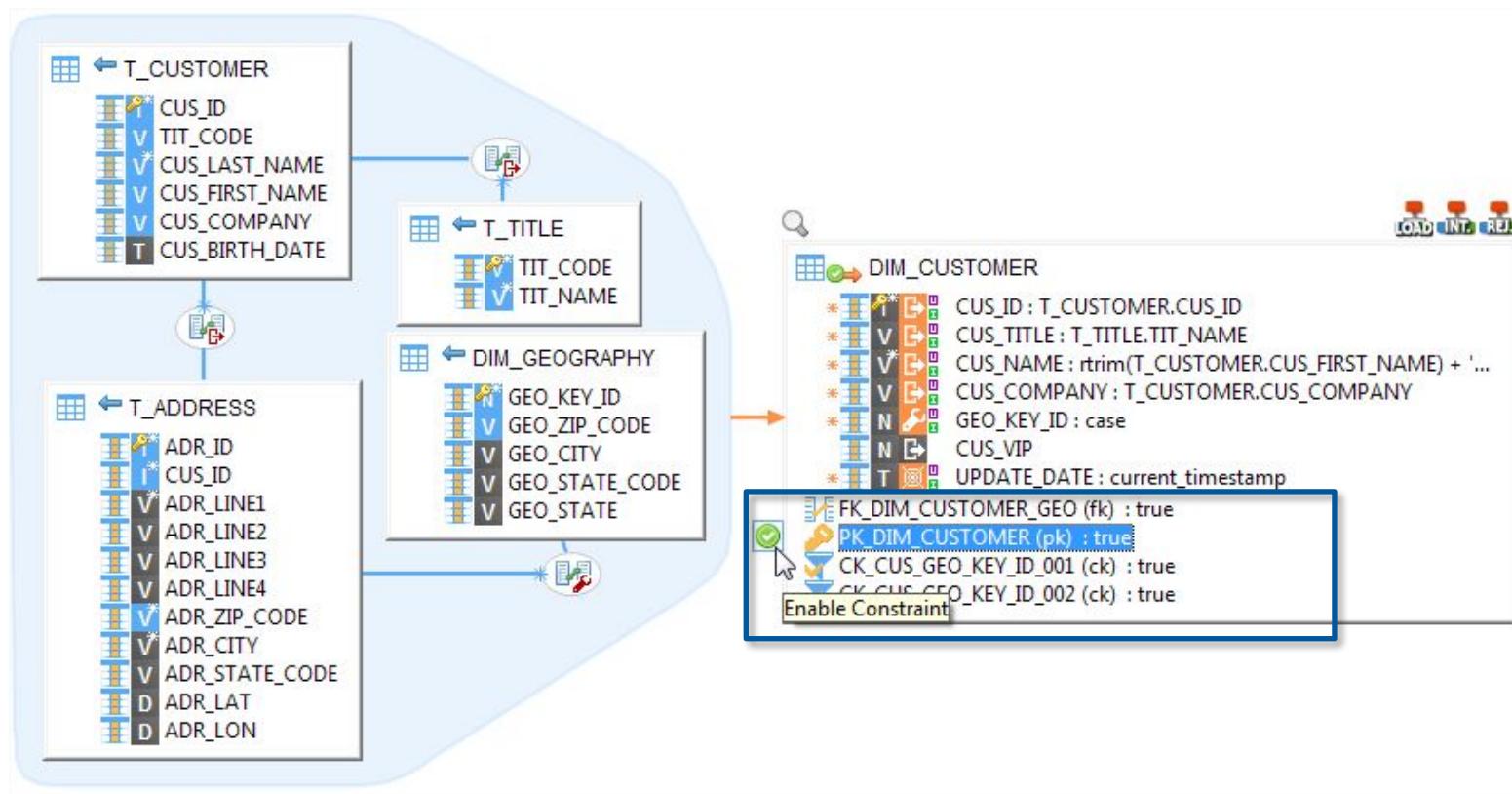
Managing the rejects

The mapping will use a Reject template to generate the appropriate SQL code



Managing the rejects

The mapping will check the data according to the rules activated on the target :

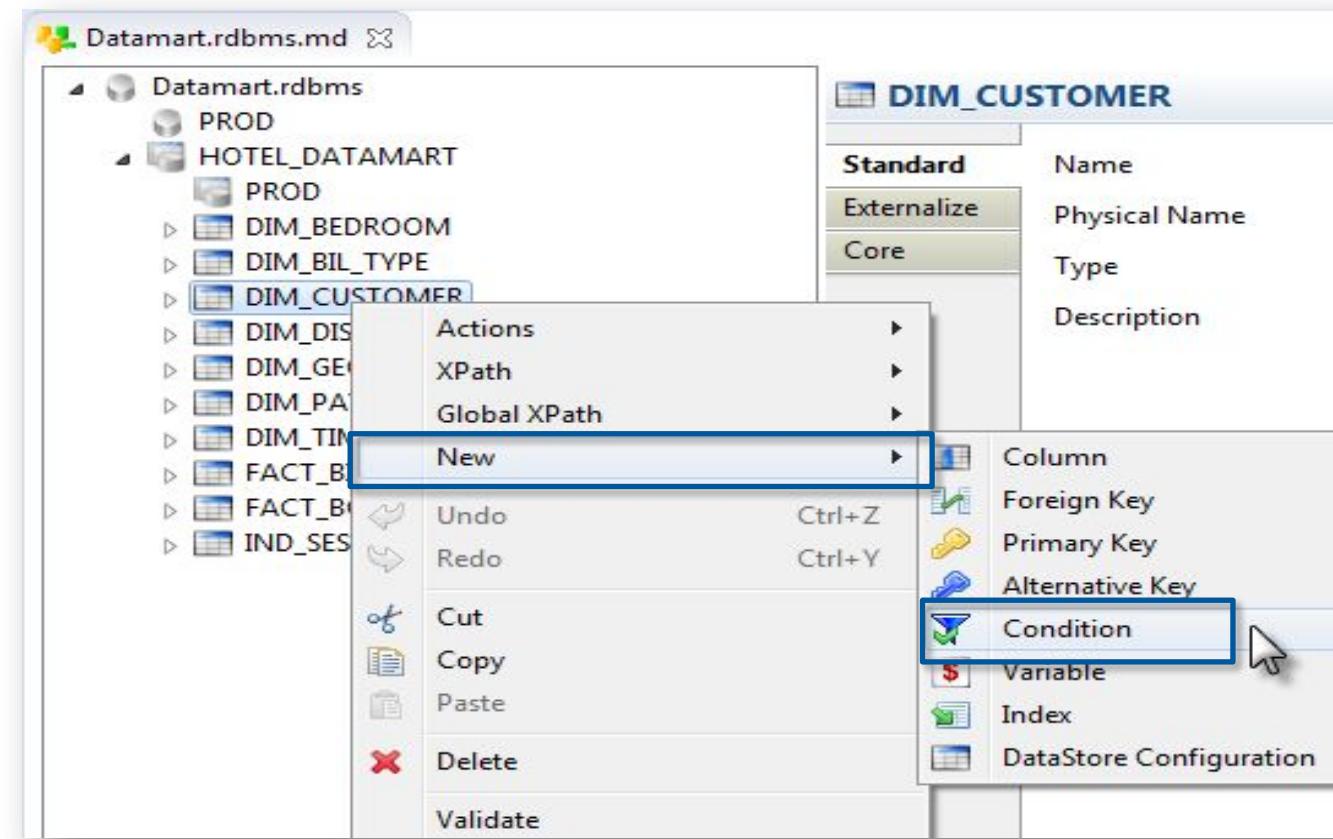


Adding constraints

It is possible to add constraints in the metadata file in order to check them in a mapping



The Metadata file must be in the work zone: double click on it in the project explorer to open it



Adding constraints

Six properties to be defined for each condition

The screenshot shows two windows. On the left is the 'HSQL_Datamart.md' interface, displaying a tree structure of tables and constraints under 'HSQL_Datamart'. On the right is a detailed view of the constraint 'CK_CUS_GEO_KEY_ID_001'. The constraint details are as follows:

Property	Value
Name	CK_CUS_GEO_KEY_ID_001
User Message	No address
Reject Code	CUS_001
Severity Level	Reject
Description	Keep track of each record loaded into DIM_CUSTOMER without any address
Condition Expression	DIM_CUSTOMER.GEO_KEY_ID <> 0

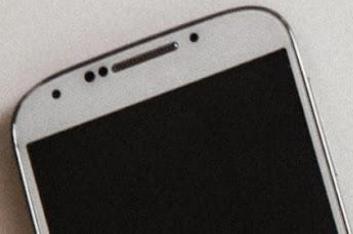
A blue arrow points from the text 'Occurs of "Fatal"' to the 'Severity Level' field.

Occurs of “Fatal”
“Severity Level” will
generate an error for
the mapping during
execution



Demo

Manage rejects

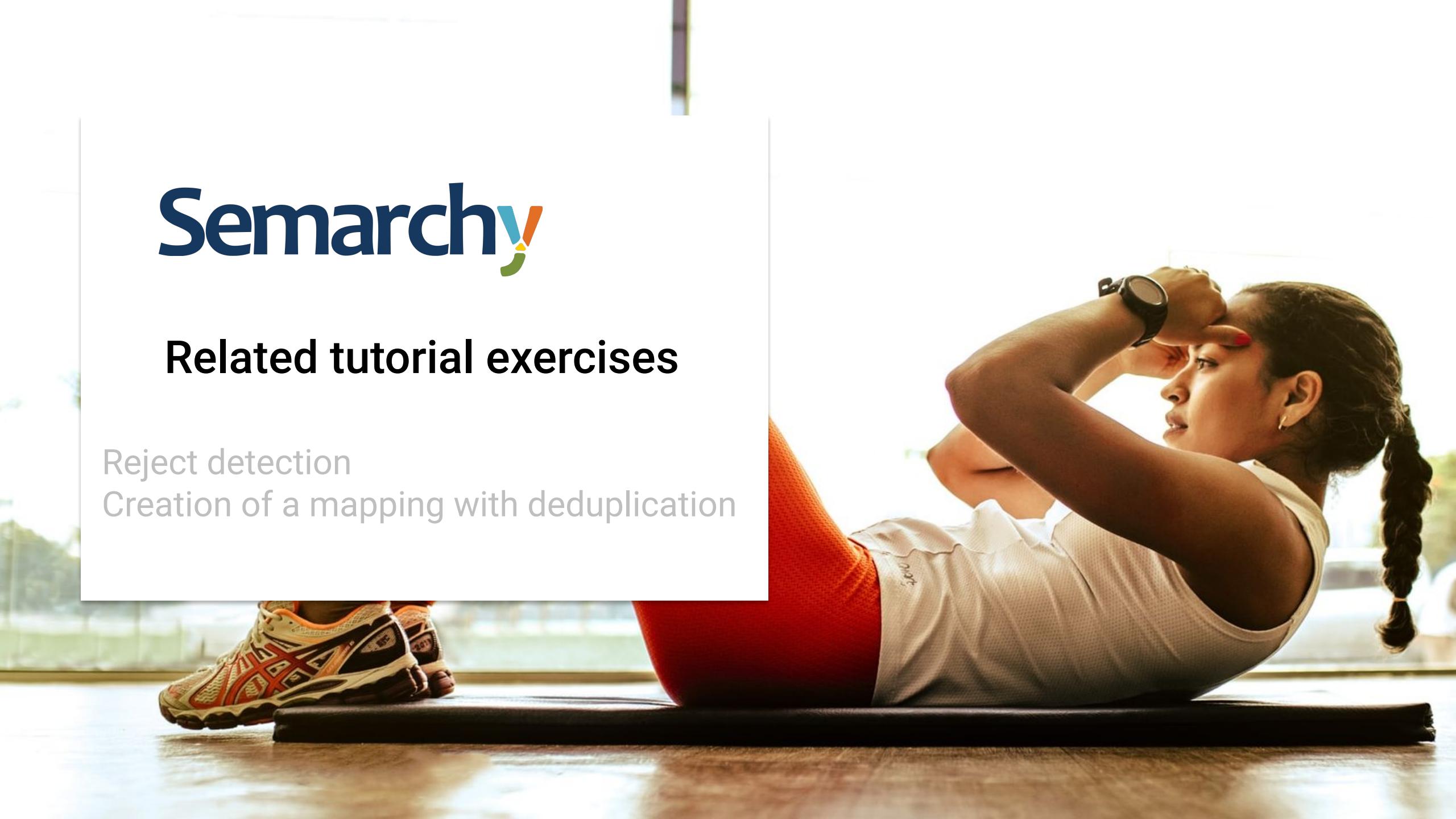




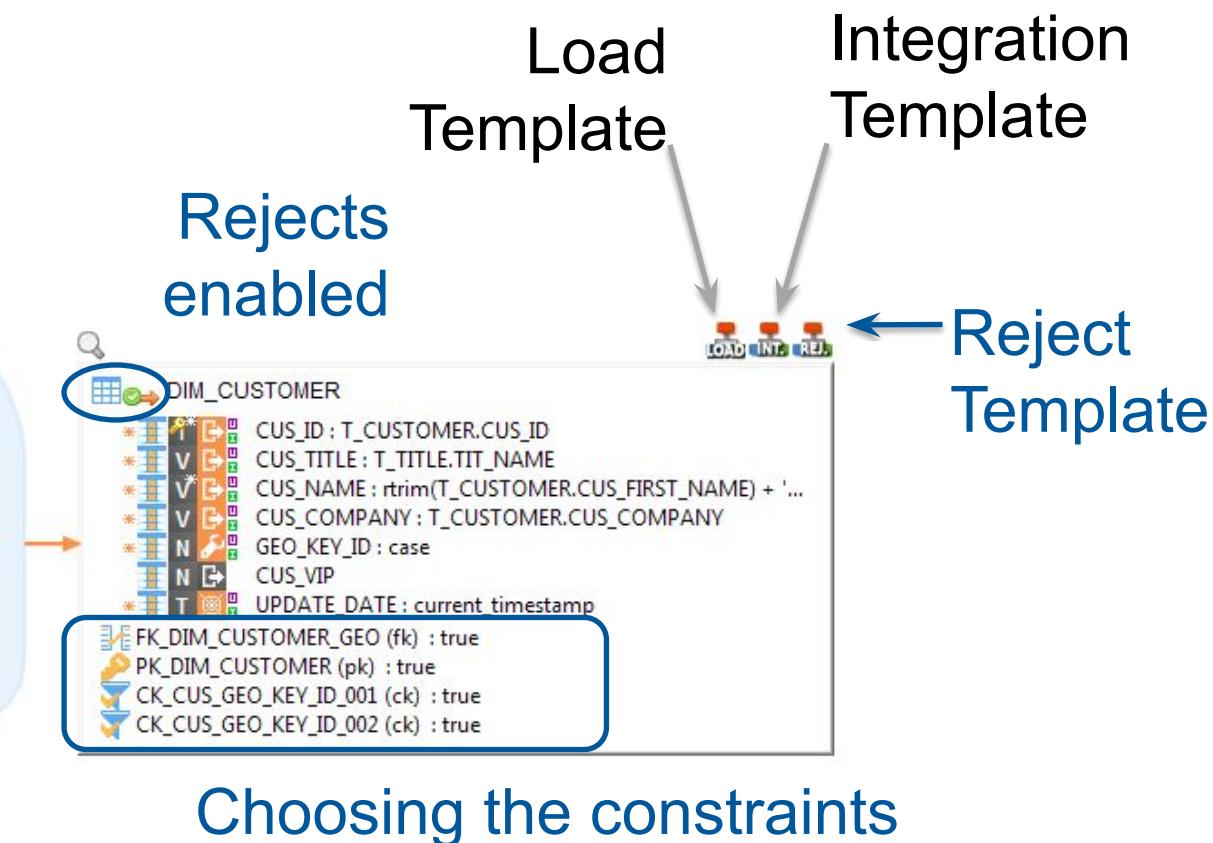
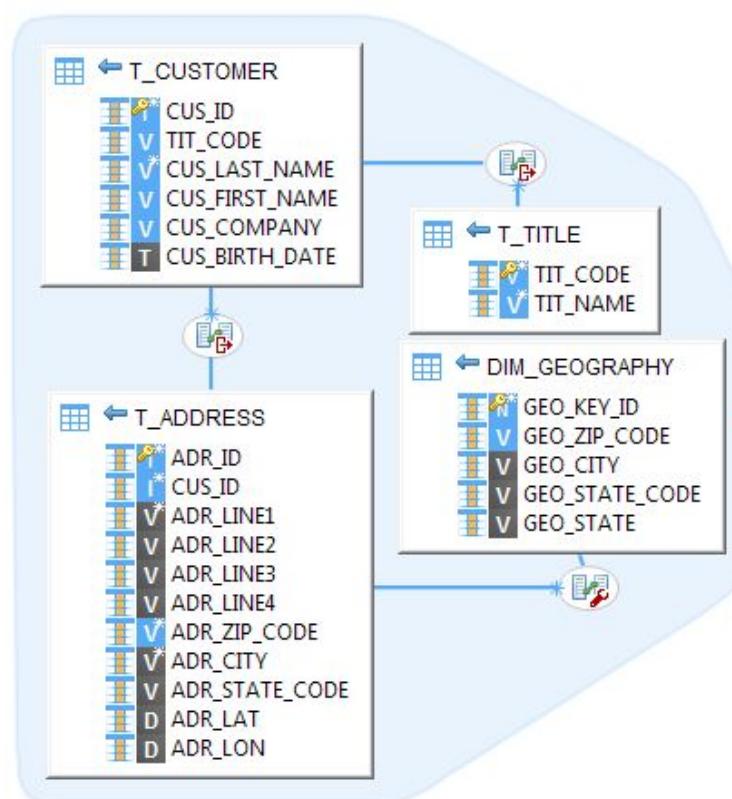
Related tutorial exercises

Reject detection

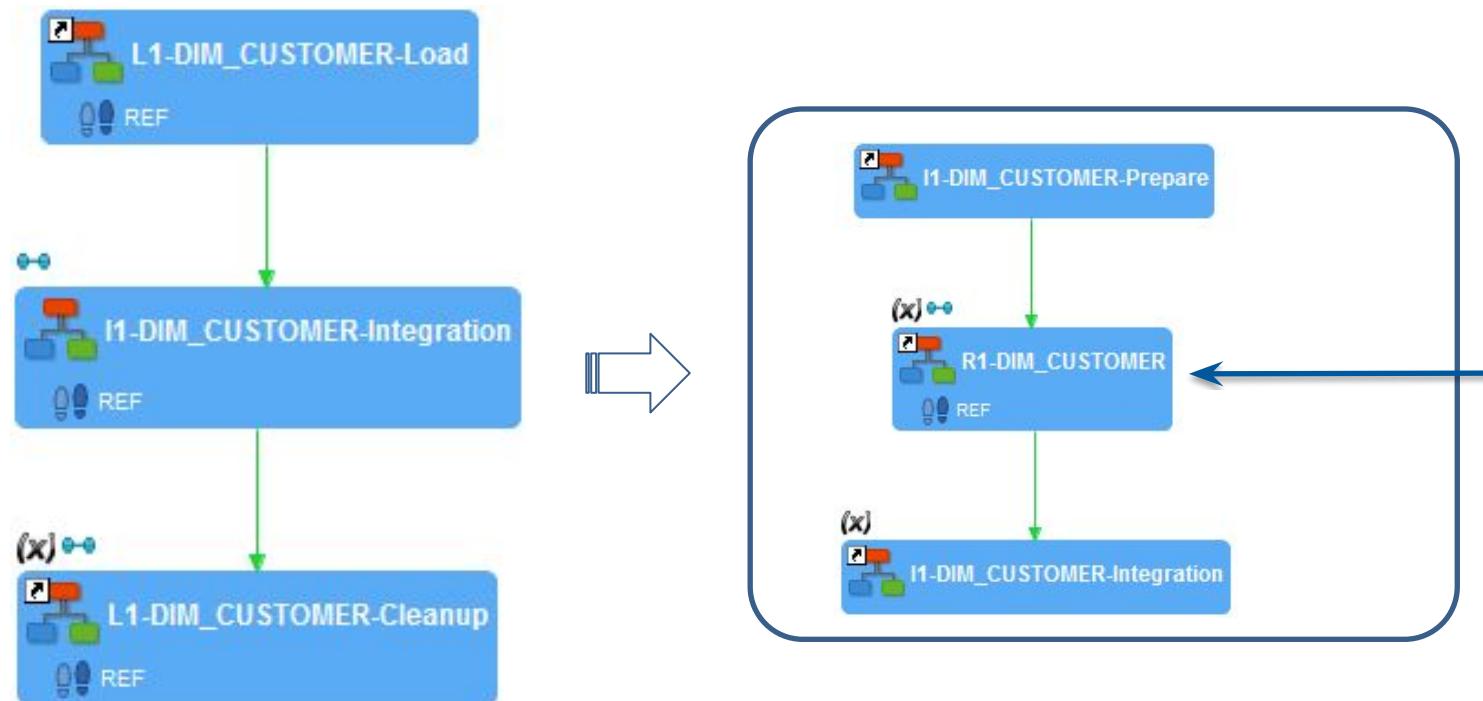
Creation of a mapping with deduplication



Analyzing the result : Load DIM_CUSTOMER

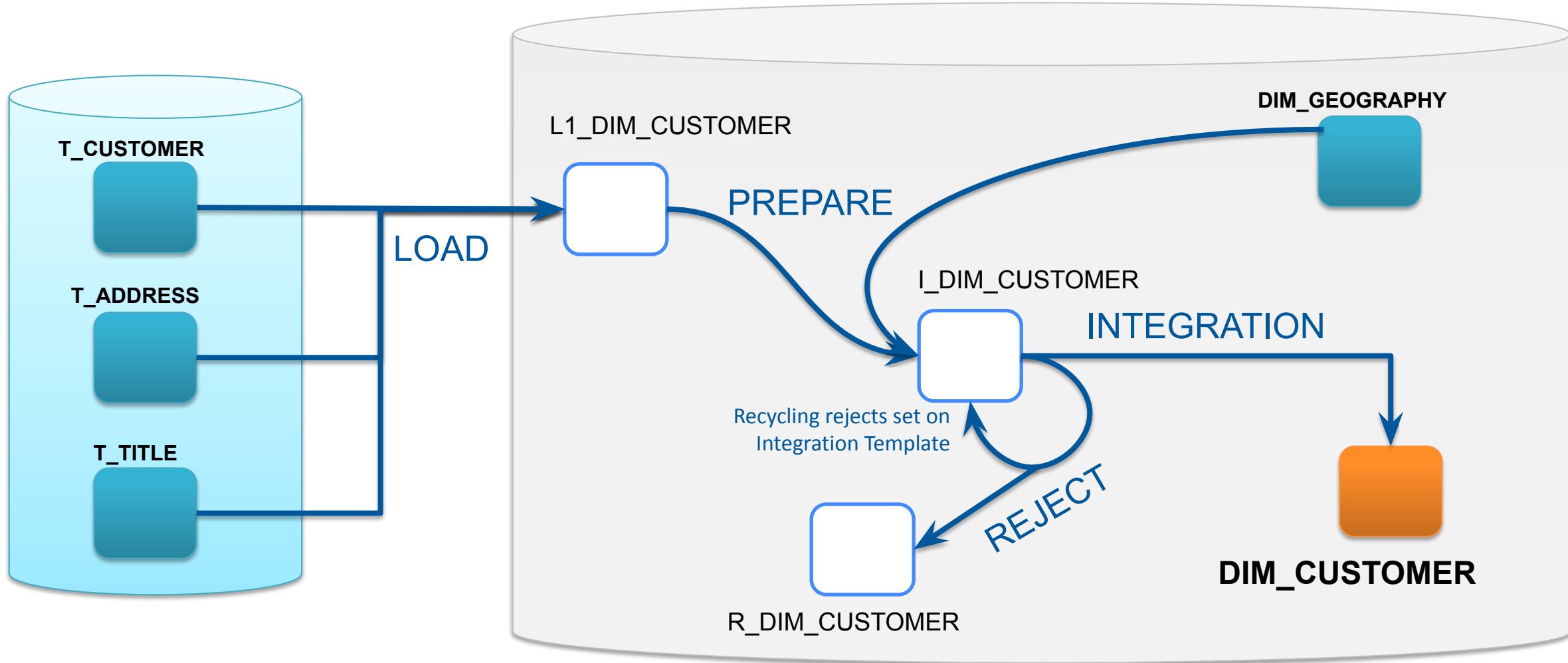


Analyzing the results

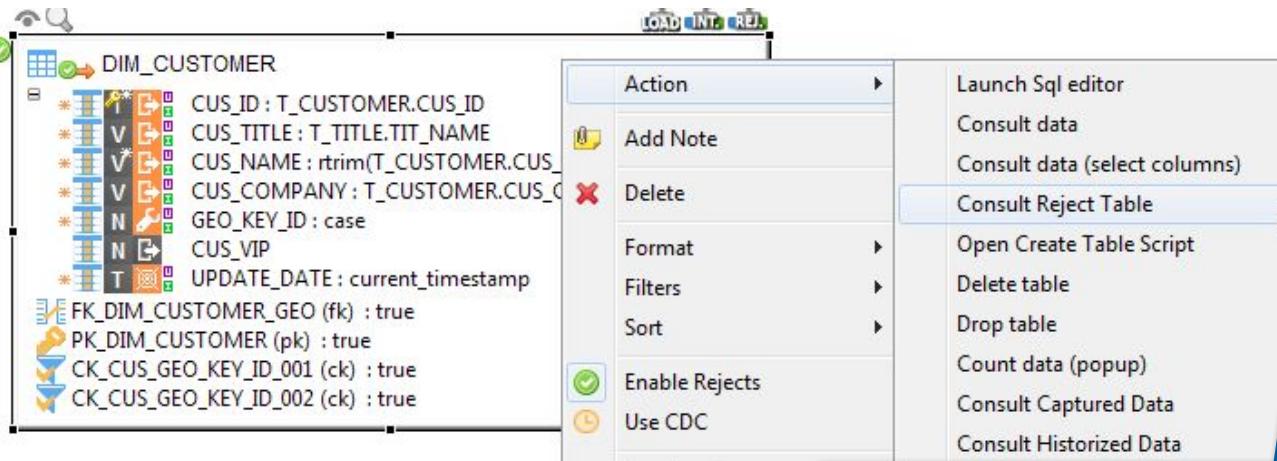


A new step appears,
generated by the
reject template

Understanding the result



Analyzing the results



The screenshot shows the 'DIM_CUSTOMER' table in the 'LOAD INTO CREA' view. A context menu is open over the table, with the 'Consult Reject Table' option highlighted. A blue arrow points from the text 'Have a look on the reject table' to the 'Messages' tab of the SQL editor window below.

Have a look on the reject table

Ctrl DIM_CUSTOMER(0).sql

```
select * from HOTEL_DATAMART.R_DIM_CUSTOMER
```

SESSION_ID	REJECT_MODE	REJECT_MESSAGE	REJECT_CODE	REJECTED_BY	RULE_NAME	RULE_TYPE	RULE_SEV
c0a800200141094b4cc5648e0f2f8a84	F	Unknown address detected	CUS_001	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	CK_CUS_GEO_KEY_ID_001	CK	REJECT
c0a800200141094b4cc5648e0f2f8a84	F	Unknown address detected	CUS_001	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	CK_CUS_GEO_KEY_ID_001	CK	REJECT
c0a800200141094b4cc5648e0f2f8a84	F	Unknown address detected	CUS_001	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	CK_CUS_GEO_KEY_ID_001	CK	REJECT
c0a800200141094b4cc5648e0f2f8a84	F	Unknown address detected	CUS_001	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	CK_CUS_GEO_KEY_ID_001	CK	REJECT
c0a800200141094b4cc5648e0f2f8a84	F	Unknown address detected	CUS_001	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	CK_CUS_GEO_KEY_ID_001	CK	REJECT
c0a800200141094b4cc5648e0f2f8a84	F	Unknown address detected	CUS_001	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	CK_CUS_GEO_KEY_ID_001	CK	REJECT
c0a800200141094b4cc5648e0f2f8a84	F	Unknown address detected	CUS_001	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	CK_CUS_GEO_KEY_ID_001	CK	REJECT
c0a800200141094b4cc5648e0f2f8a84	F	Unknown address detected	CUS_001	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	CK_CUS_GEO_KEY_ID_001	CK	REJECT
c0a800200141094b4cc5648e0f2f8a84	F	Unknown address detected	CUS_001	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	CK_CUS_GEO_KEY_ID_001	CK	REJECT
c0a800200141094b4cc5648e0f2f8a84	F	Unknown address detected	CUS_001	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	CK_CUS_GEO_KEY_ID_001	CK	REJECT
c0a800200141094b4cc5648e0f2f8a84	F	Unknown address detected	CUS_001	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	CK_CUS_GEO_KEY_ID_001	CK	REJECT
c0a800200141094b4cc5648e0f2f8a84	F	Unknown address detected	CUS_001	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	CK_CUS_GEO_KEY_ID_001	CK	REJECT
c0a800200141094b4cc5648e0f2f8a84	F	Unknown Zip Code	CUS_002	Load DIM_CUSTOMER (_dyiTgMnzEeKhW67aUTK58Q)	"CK_CUS_GEO_KEY_ID_002"	CK	REJECT



Stage step usage



Staging transformation

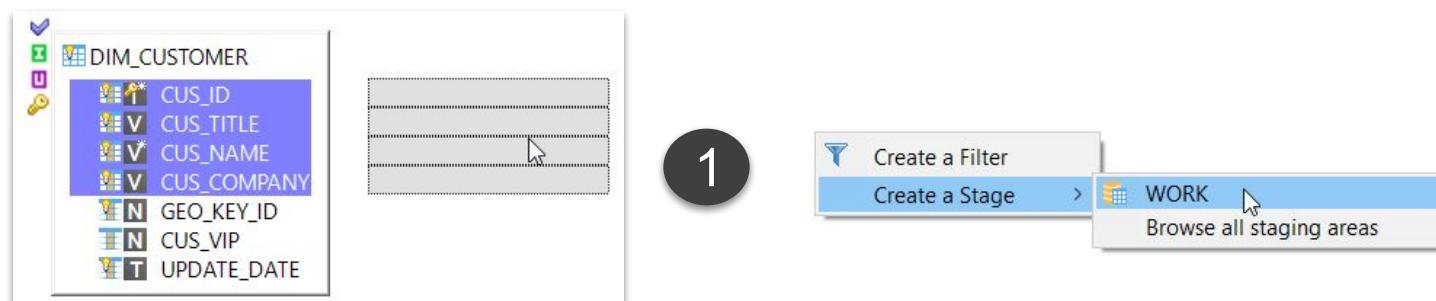
You can use a “*stage area*” to build intermediate transformation

Two ways to create a stage

1. Drag & drop a RDBMS schema



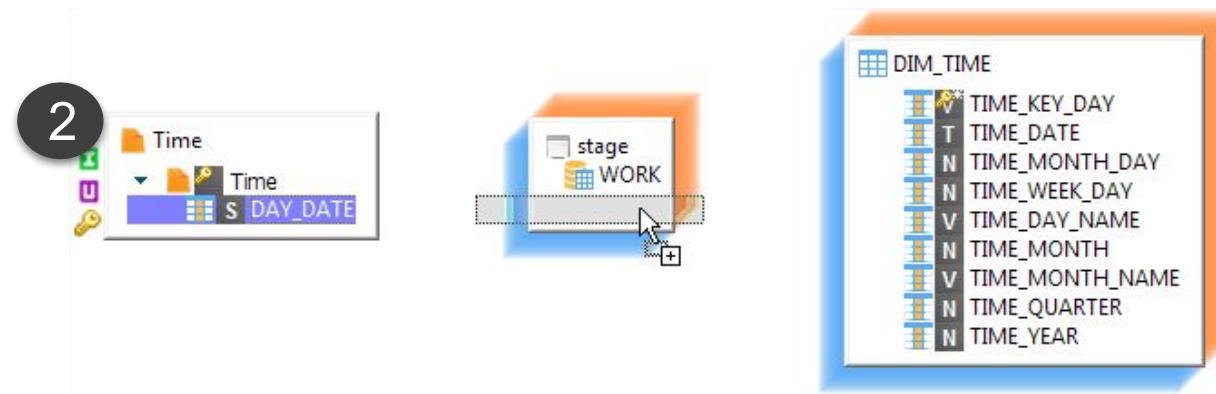
2. Drag & drop a list of source columns directly



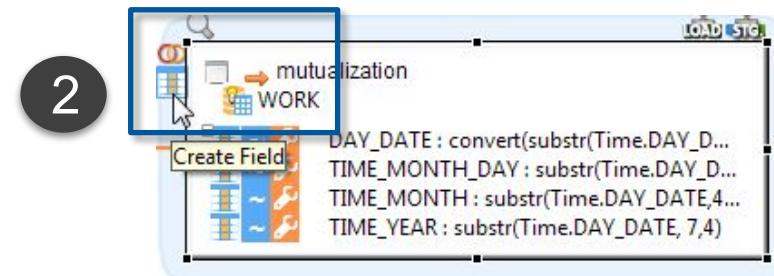
Staging transformation

Two ways to add fields

1. Drag & Drop source columns in the stage component

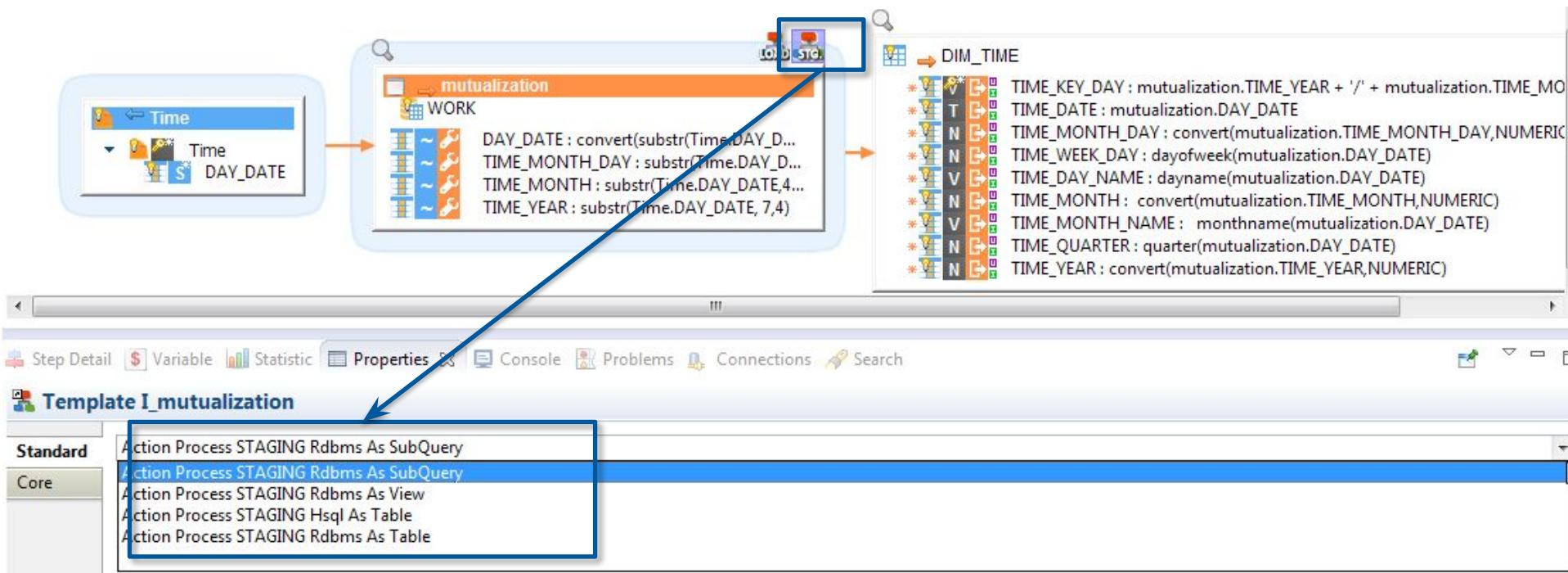


2. Or click on Create Field button



Staging transformation

You can choose the action on the stage in the list of stage Templates



Stage As Table & Data Types

If you want to create a table with the stage fields, it's possible to define their data types :

The screenshot illustrates the configuration of a data mapping process in the Semarchy Intelligent Data Hub.

Left Panel (Action Process STAGING Hsqi As Table):

- Source:** A "Time" source with a "Time DAY_DATE" field.
- Transformation:** A "gen_date" step with a "WORK_DM" target. It contains four SQL expressions:
 - DAY_DATE : convert(subst..., TIME_MONTH_DAY : subst..., TIME_MONTH : substr(Tim..., TIME_YEAR : substr(Time...
 - TIME_MONTH_DAY : f::subString(Time.DAY_DATE,1,2)
 - TIME_MONTH : f::subString(Time.DAY_DATE,4,2)
 - TIME_YEAR : f::subString(Time.DAY_DATE,7,4)
- Target:** A "Time" target with a "S DAY_DATE" field.
- Properties:**
 - Action:** Action Process STAGING Hsqi As Table
 - Checkboxes:** Use Distinct, Clean Temporary Objects, Use Stage Name For Temporary Object Name, Transactional Mode On Work Tables, Commit Transaction, Lock Cdc Table, Transaction Name (set to T1), Cdc Subscriber.

Middle Panel (mutualization):

- Source:** A "Time" source with a "Time DAY_DATE" field.
- Transformation:** A "mutualization" step with a "WORK" target. It contains four SQL expressions:
 - DAY_DATE : f::stringToTimestamp(f::concat6Strings(f::subString(Time.DAY_DATE,7,4),'-',f::subString(Time.DAY_DATE,1,2),'-',f::subString(Time.DAY_DATE,4,2),'-',f::subString(Time.DAY_DATE,1,2)))
 - TIME_MONTH_DAY : f::subString(Time.DAY_DATE,1,2)
 - TIME_MONTH : f::subString(Time.DAY_DATE,4,2)
 - TIME_YEAR : f::subString(Time.DAY_DATE,7,4)
- Target:** A "DIM_TIME" target with various fields:
 - TIME_KEY_DAY : f::concat5Strings(mutualization.TIME_YEAR,'/',mutualization.TIME_MONTH,'-',mutualization.TIME_DAY)
 - TIME_DATE : mutualization.DAY_DATE
 - TIME_WEEK_DAY : f::dayOfWeek(mutualization.DAY_DATE)
 - TIME_DAY_NAME : f::dayName(mutualization.DAY_DATE)
 - TIME_MONTH : f::stringToNumber(mutualization.TIME_MONTH)
 - TIME_MONTH_NAME : f::monthName(mutualization.DAY_DATE)
 - TIME_QUARTER : f::quarter(mutualization.DAY_DATE)
 - TIME_YEAR : f::stringToNumber(mutualization.TIME_YEAR)

Right Panel (Query Field DAY_DATE):

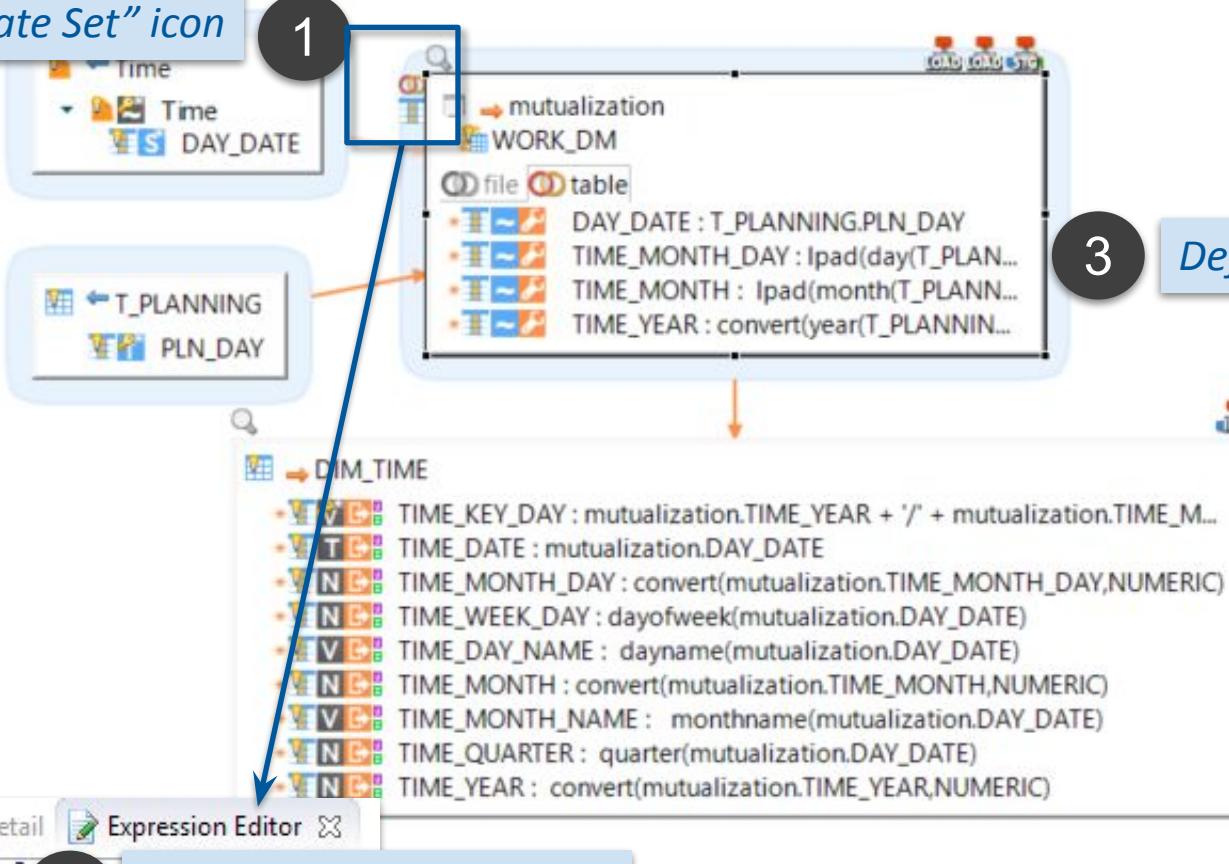
- Alias:** DAY_DATE
- Core Properties:**
 - Enable:** Checked
 - Execution Location:** Staging Area
 - Mapping:** Aggregate, Enable Datatype (checked)
- Structure:** Shows the data type is set to **TIMESTAMP**.
- Standard:** Shows the data type is set to **TIMESTAMP**.
- Type:** Shows the data type is set to **TIMESTAMP**.
- Precision:** Shows the data type is set to **TIMESTAMP**.
- Scale:** Shows the data type is set to **TIMESTAMP**.

Create Set in a stage

You can choose ‘Create Set’ in a stage to add a new set of sources

Click on the “Create Set” icon

1



Define the expressions for the new set

3

2

Define the set operation



Demo

Create stages & sets



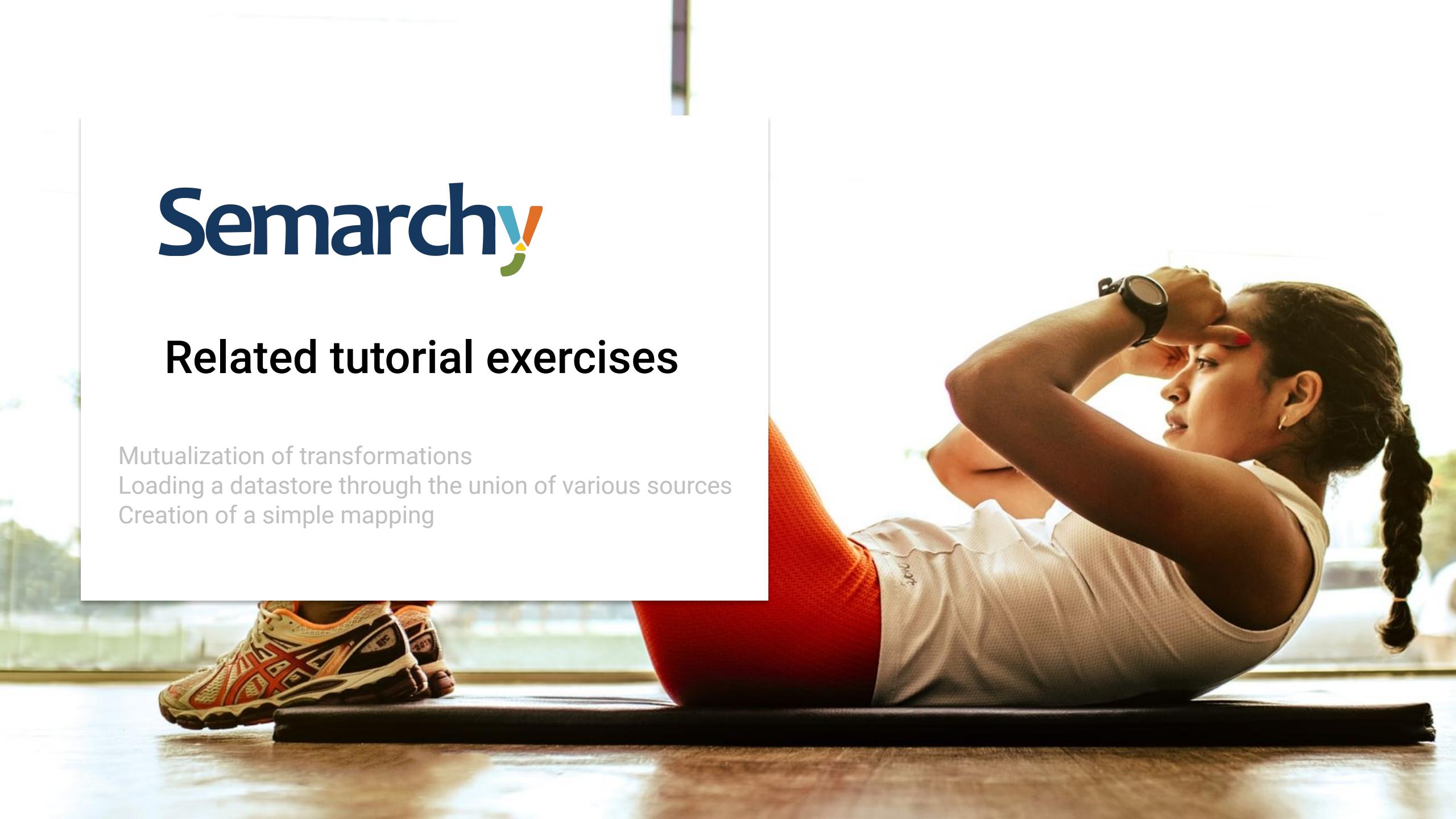


Related tutorial exercises

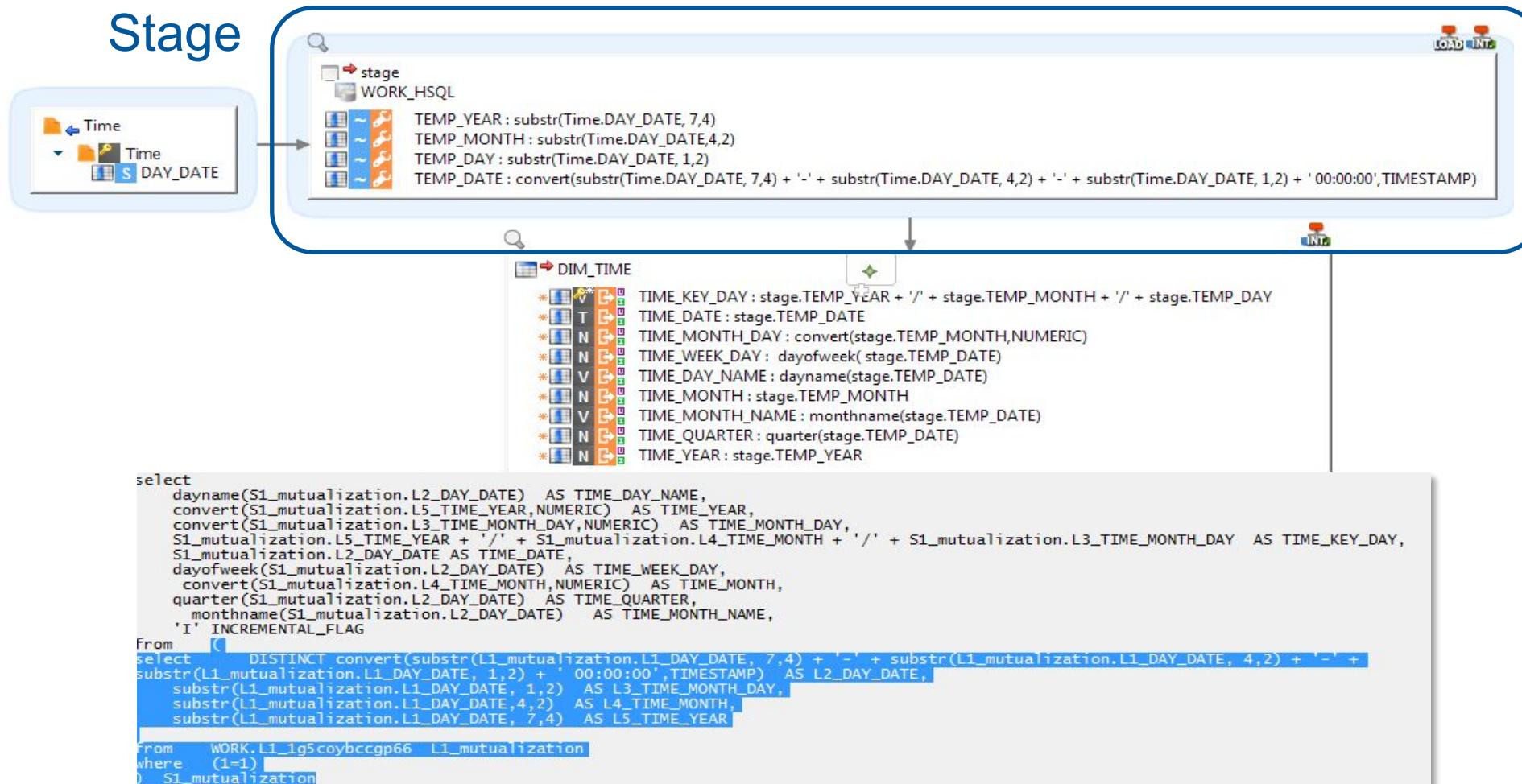
Mutualization of transformations

Loading a datastore through the union of various sources

Creation of a simple mapping



Analyzing the result : DIM_TIME



Analyzing the result : DIM_TIME





xDI DEV

C Delving Mapping

C2 - Feedback on Mapping deepening

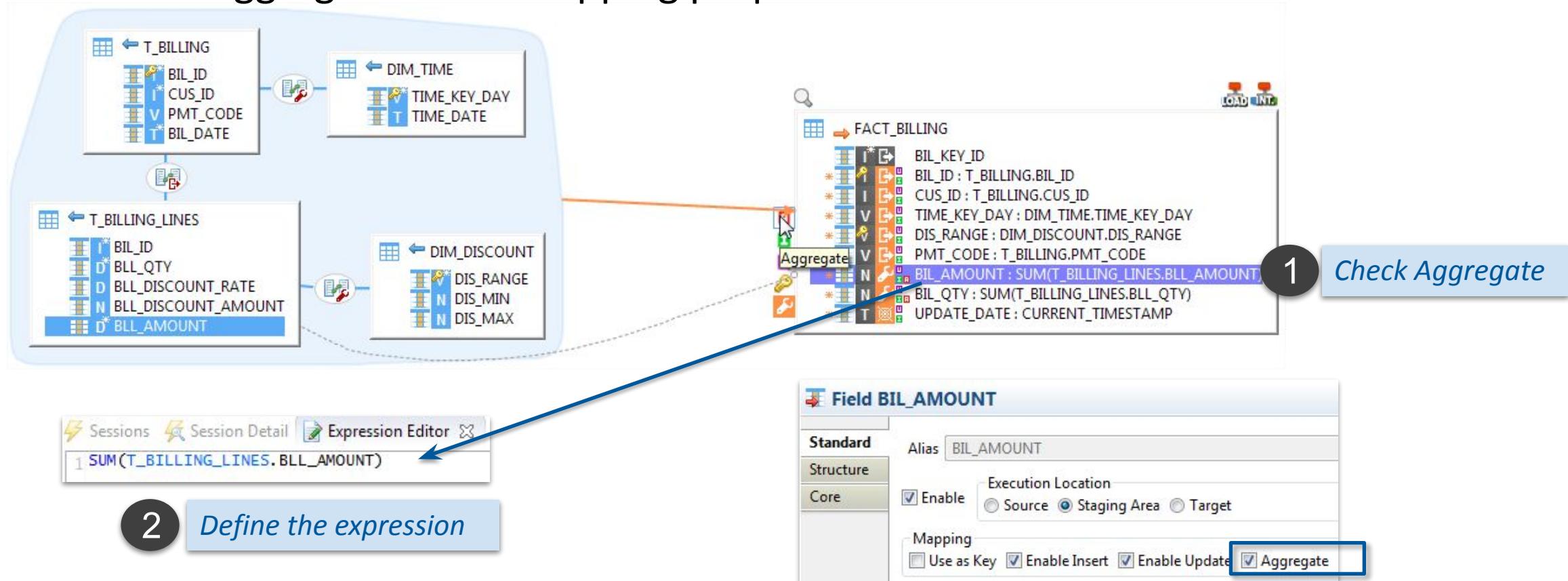


Manage aggregates



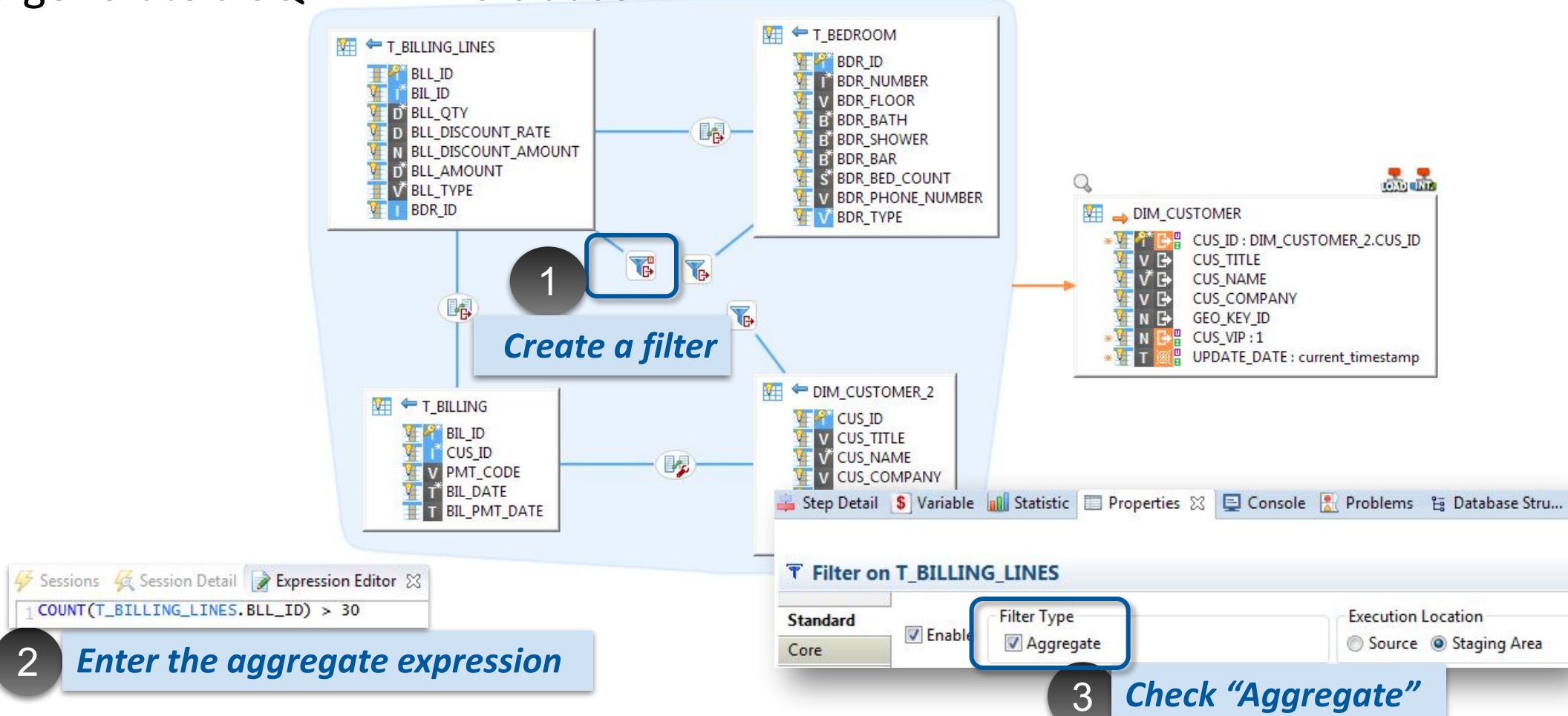
Managing aggregates

To do an aggregation, just define your expression in the Expression Editor and choose Aggregate in the mapping properties



Managing HAVING clause

To generate a SQL HAVING clause :





Demo

Aggregates



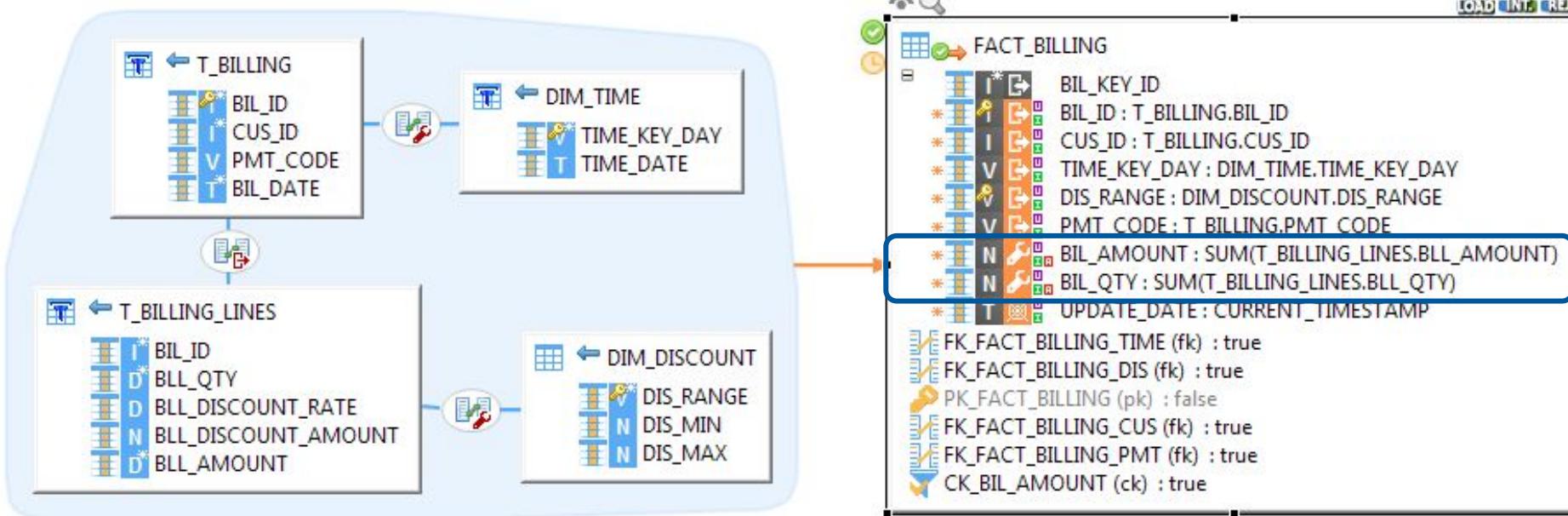


Related tutorial exercises

Creation of a Mapping with aggregate calculation
Additional exercises

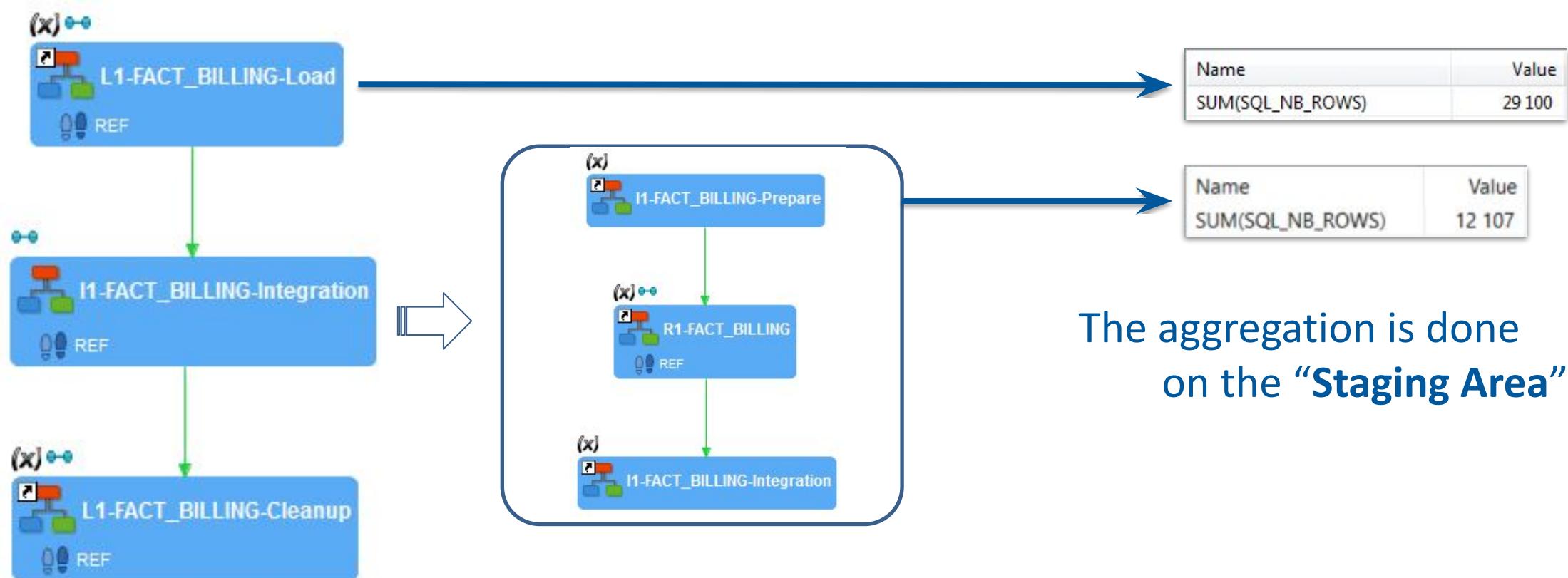


Analyzing the result

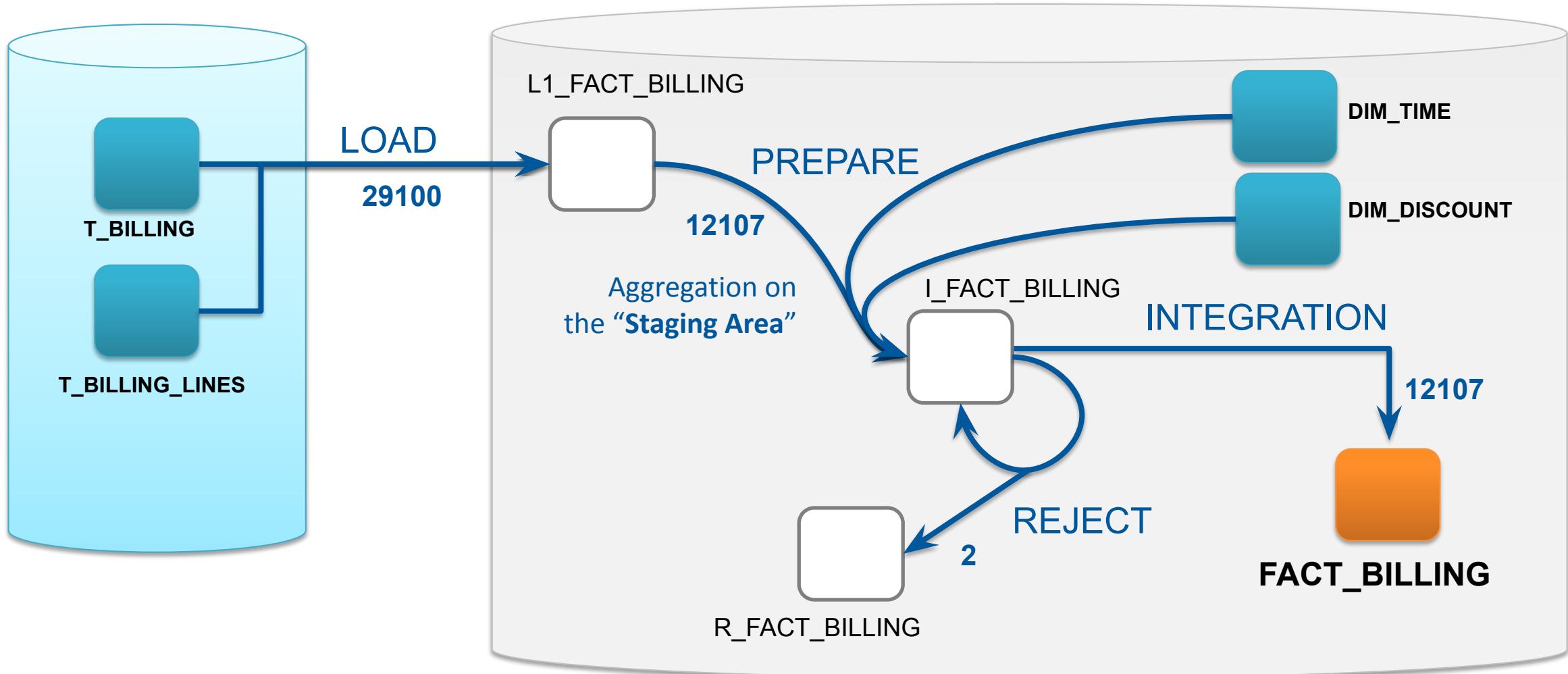


Aggregation on the
“Staging Area” location

Analyzing the result

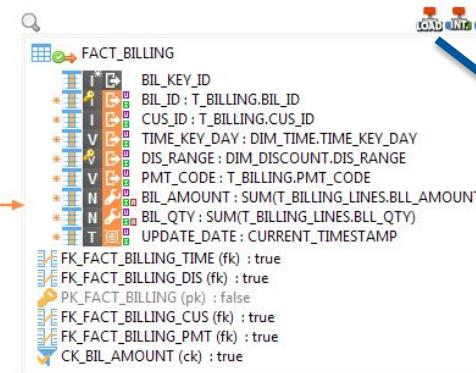
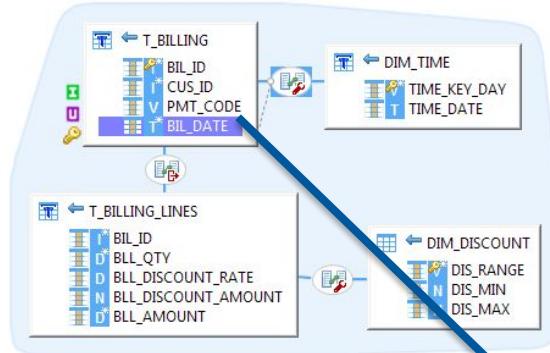


Understanding the result



Increase the performances

In order to improve the performance, it's possible to add an index on the load table



Field BIL_DATE

Standard	Alias <input type="text" value="BIL_DATE"/>
Structure	Execution Location <input type="radio"/> Source <input type="radio"/> Staging Area <input checked="" type="radio"/> Target
Core	Enable <input type="checkbox"/> Mapping <input type="checkbox"/> Use as Key <input type="checkbox"/> Enable Insert <input type="checkbox"/> Enable Update <input type="checkbox"/> Aggregate Description

Tags

1

Add a tag on the column

Note : the index tags must begin with « IDX » in uppercase

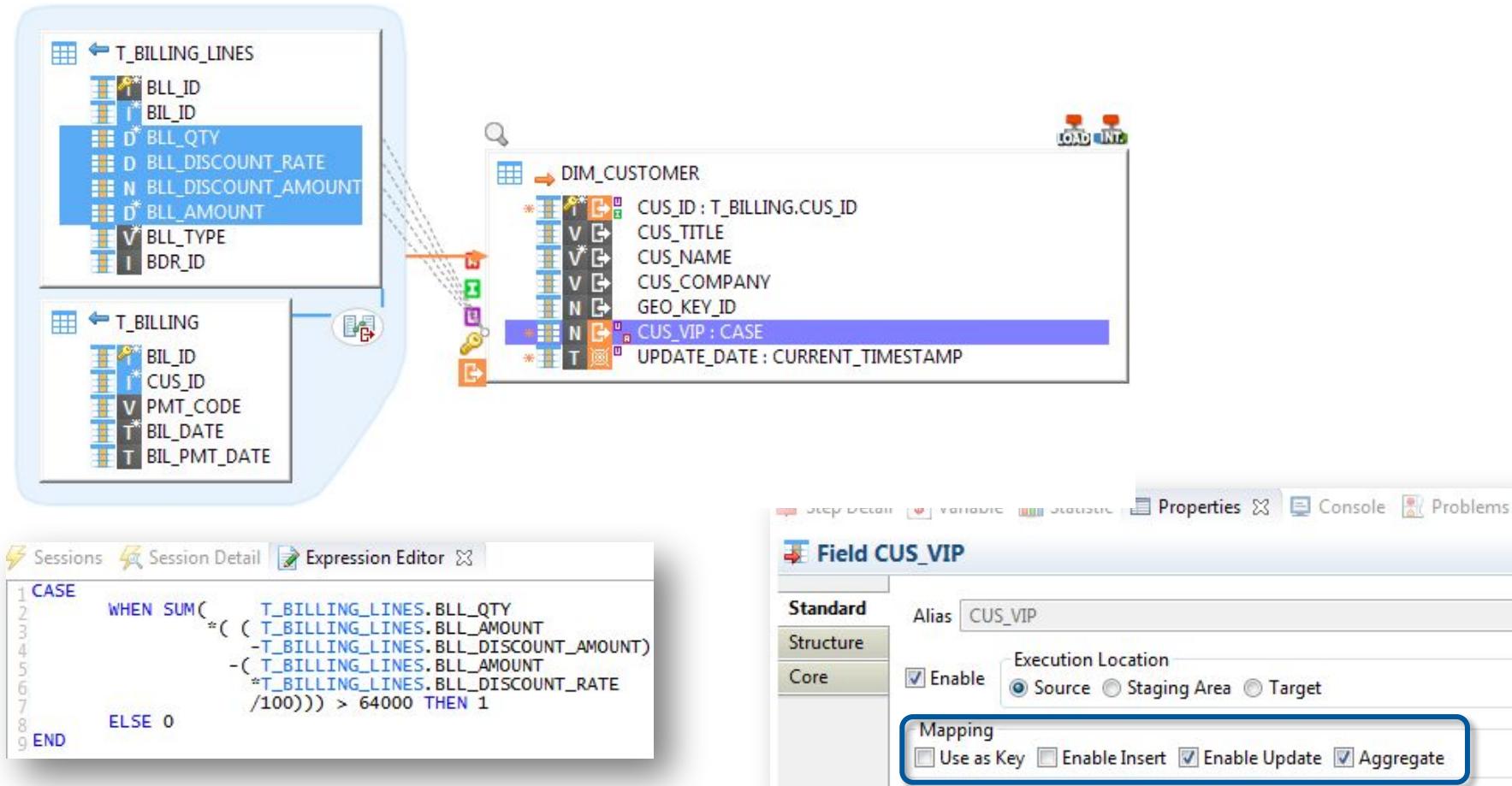
Template L1 FACT_BILLING

Action Process LOAD Rdbms to Rdbms	<input type="checkbox"/> Clean Temporary Objects <input type="checkbox"/> Use Distinct
Core	<input type="checkbox"/> Cdc Wait Mode <input checked="" type="checkbox"/> Create Load Indexes
	<input checked="" type="checkbox"/> Lock Cdc Table <input type="checkbox"/> Use Transaction On Work Tables

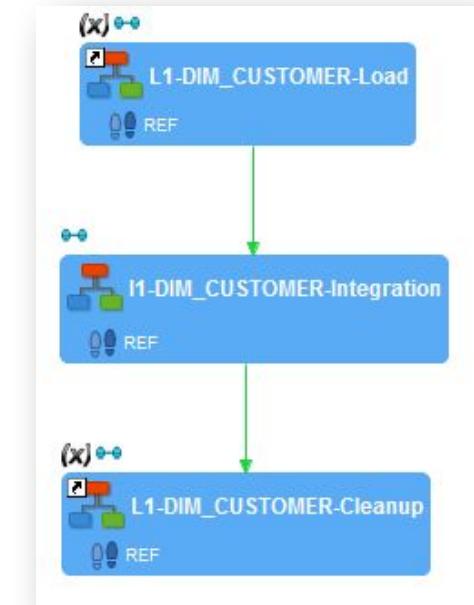
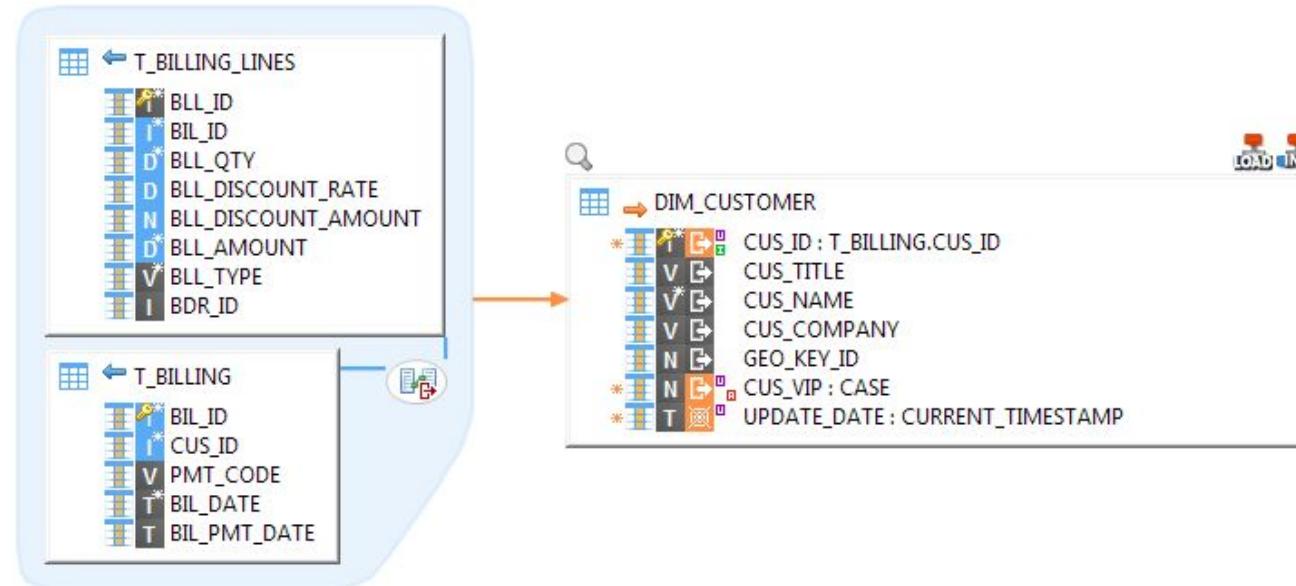
2

Activate the index creation

Analyzing the result



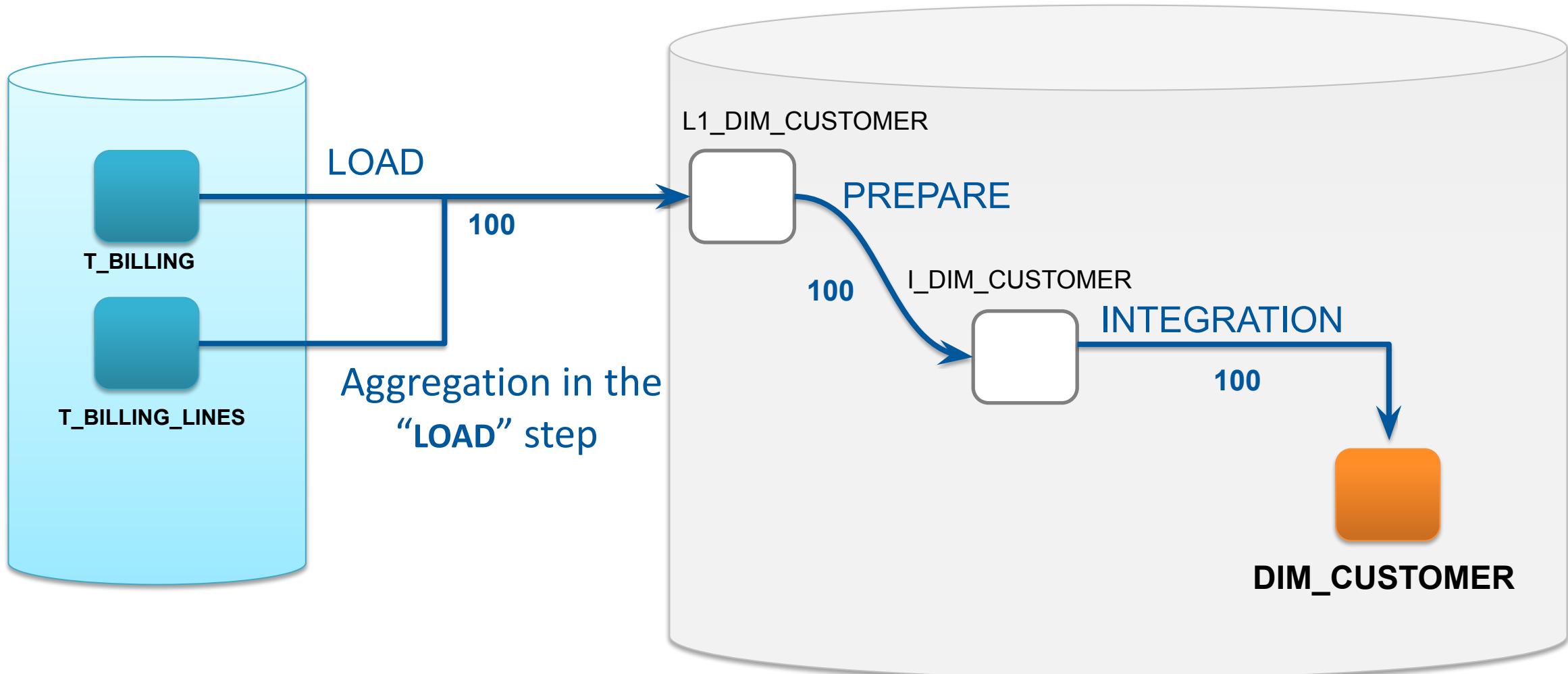
Analyzing the result



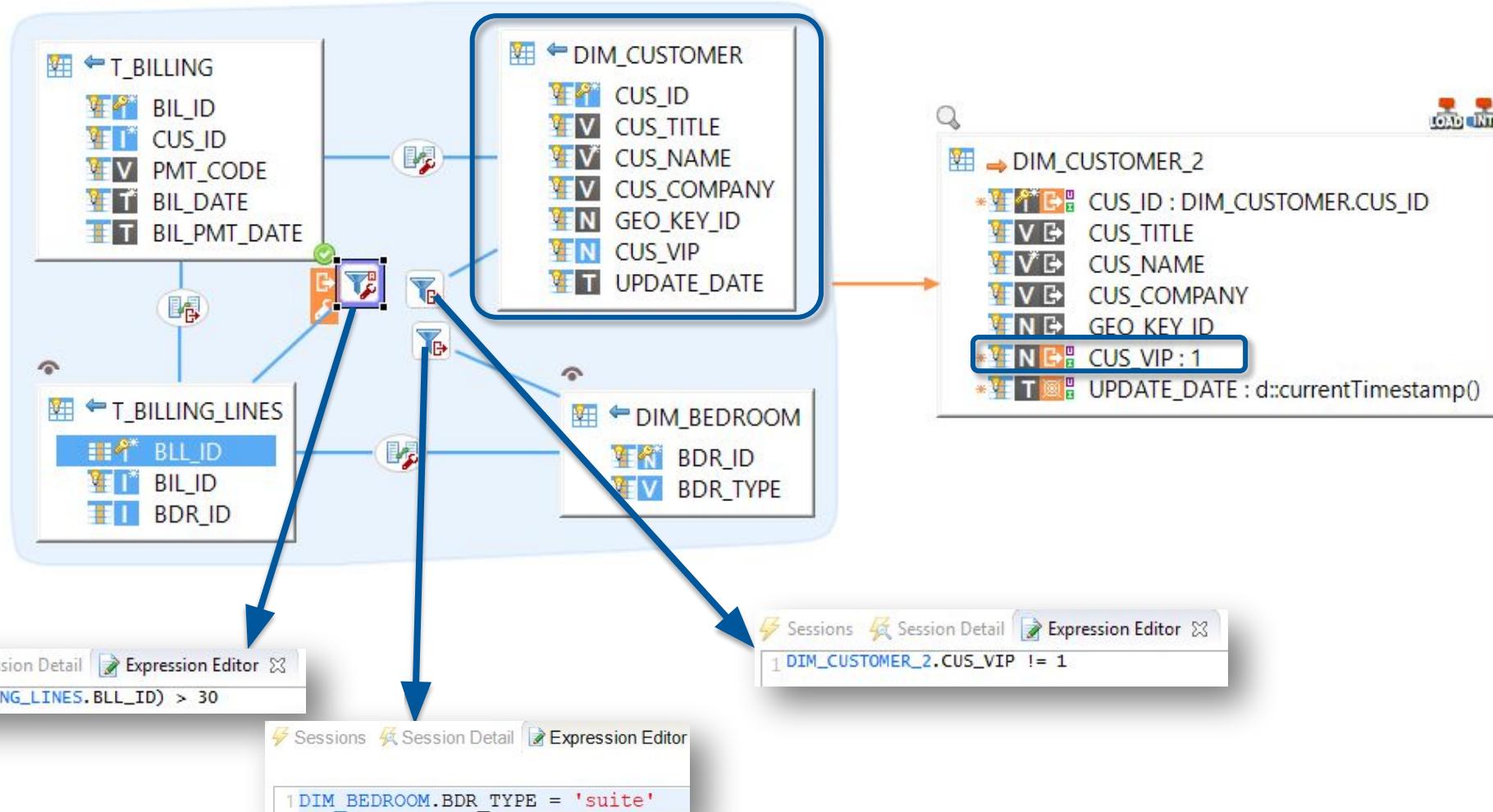
Step Detail \$ Variable Statistic

Name	Value
SUM(SQL_NB_ROWS)	400
SUM(SQL_STAT_INSERT)	0
SUM(SQL_STAT_UPDATE)	100

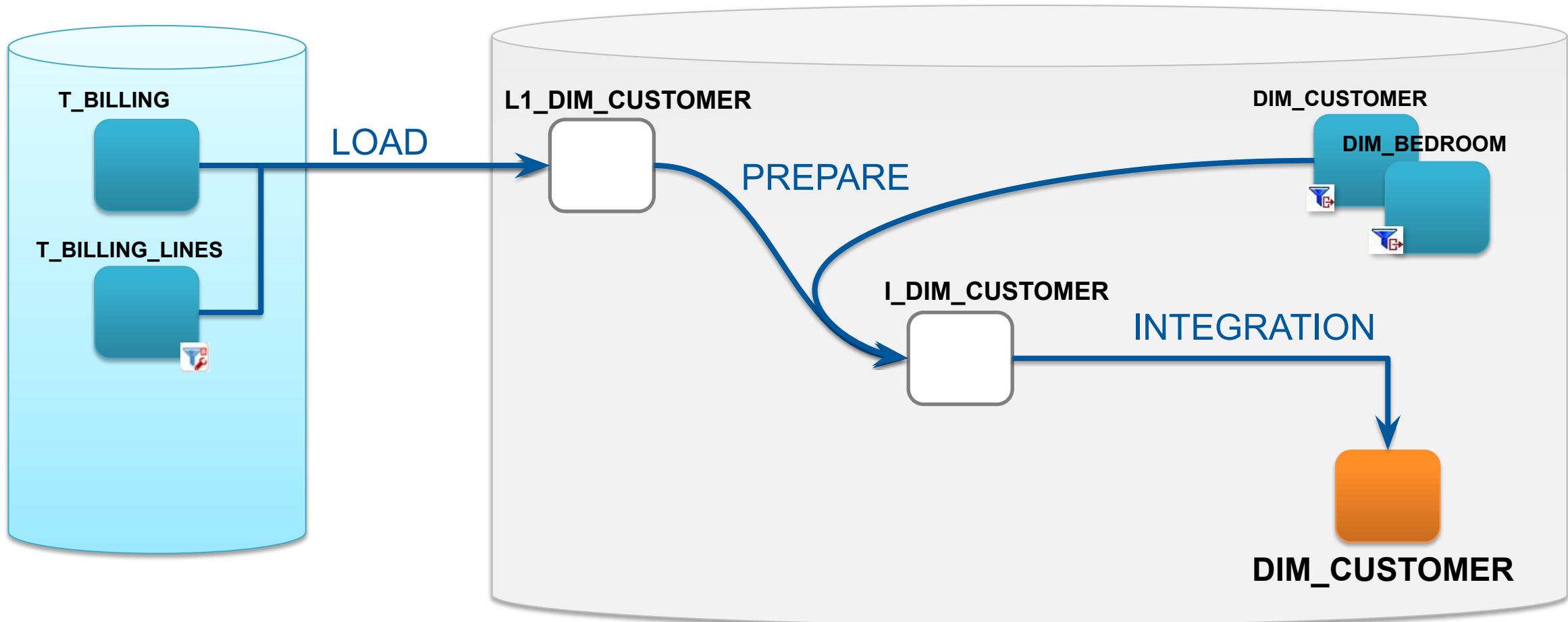
Understanding the result



Analyzing the result



Understanding the result





**Impact management &
navigate in metadatas**



Impact management

Impact view can be used to help developers find the relationships between objects :

The screenshot shows the Semarchy application interface. The Project Explorer view on the left displays a tree structure of metadata, including Imported Metadata, FILE_Server, FUNC_Set, HSQL_Datamart, and HSQL_Datamart subfolders containing HOTEL_DATAMART, Hierarchy, DIM_BEDROOM, DIM_BIL_TYPE, and DIM_CUSTOMER. The main workspace is titled 'Impact' and shows the details for the selected object, 'DIM_CUSTOMER'. The 'Used By' section is expanded, listing several entries that reference the 'DIM_CUSTOMER' object. A toolbar at the top of the workspace has a button labeled 'Show Impact (Ctrl+Shift+I)', which is highlighted with a blue box. The status bar at the bottom indicates the copyright information: '© 2020 Semarchy'.

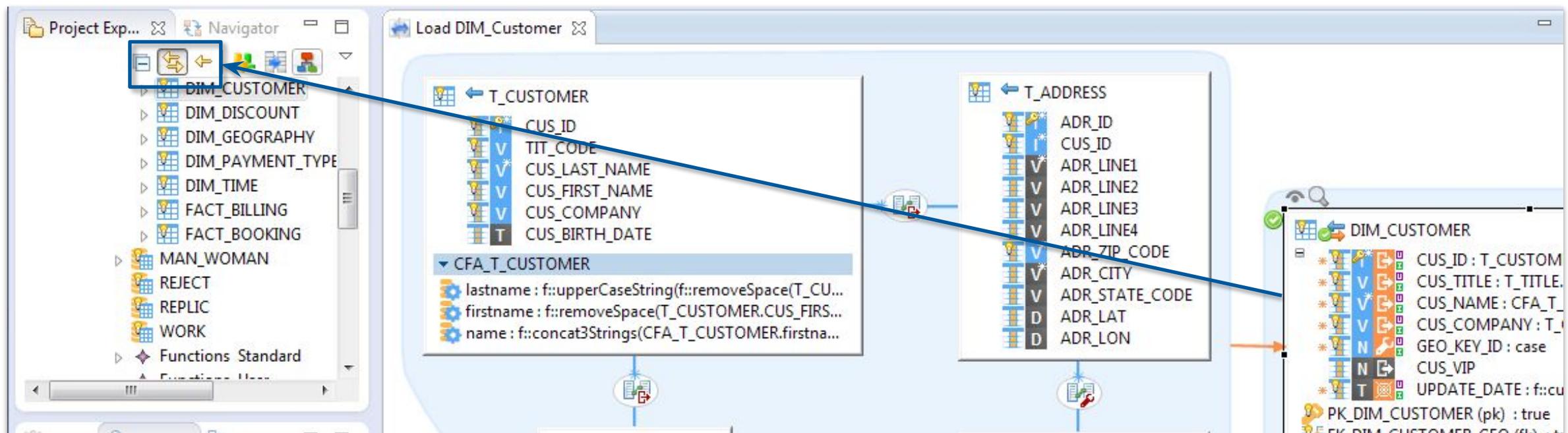
Object	File
DIM_CUSTOMER	/Metadata/HSQL_Datamart.md

Used By:

- Meta Data Link DIM_CUS /Training/Additional/Bind link0.proc
- DIM_CUSTOMER /Tutorial - Fundamentals/Dimension/Load DIM_Customer.map
- DIM_CUSTOMER /Tutorial - Fundamentals/Dimension/Update CUS_VIP/Update CUS_VIP step1.map
- DIM_CUSTOMER_2 /Tutorial - Fundamentals/Dimension/Update CUS_VIP/Update CUS_VIP step2.map
- DIM_CUSTOMER /Tutorial - Fundamentals/Dimension/Update CUS_VIP/Update CUS_VIP step2.map
- Meta Data Link SOURCE_ /Tutorial - Fundamentals/Load All Datamart.proc

Navigate in Metadatas

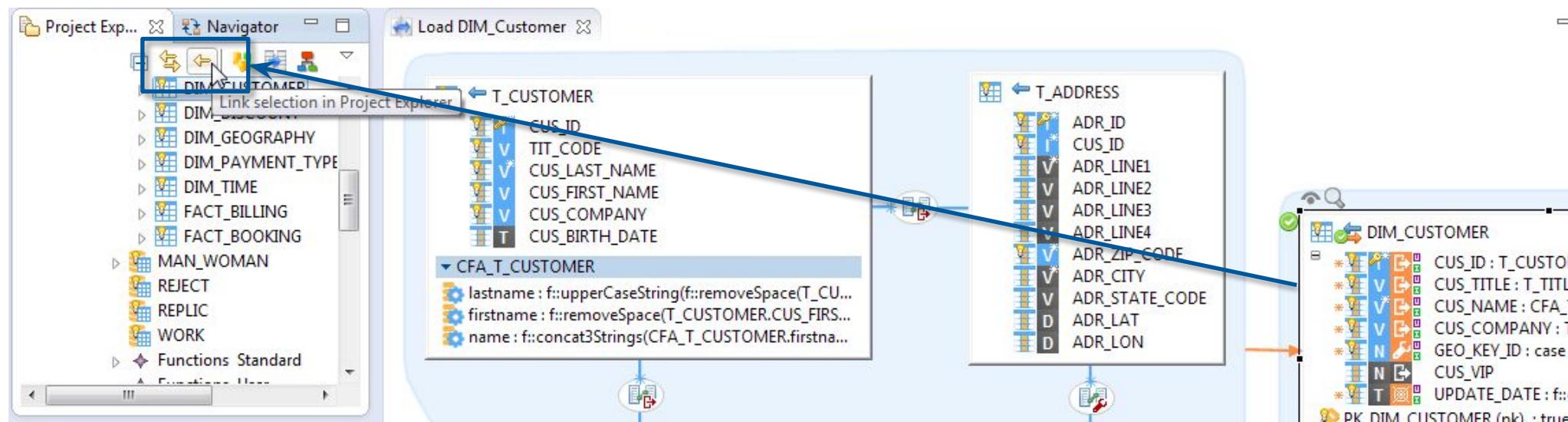
Use the “Link with Editor” button (push button switch) to find the related Metadata in the Project Explorer



Take care not to forget to unset the button after !

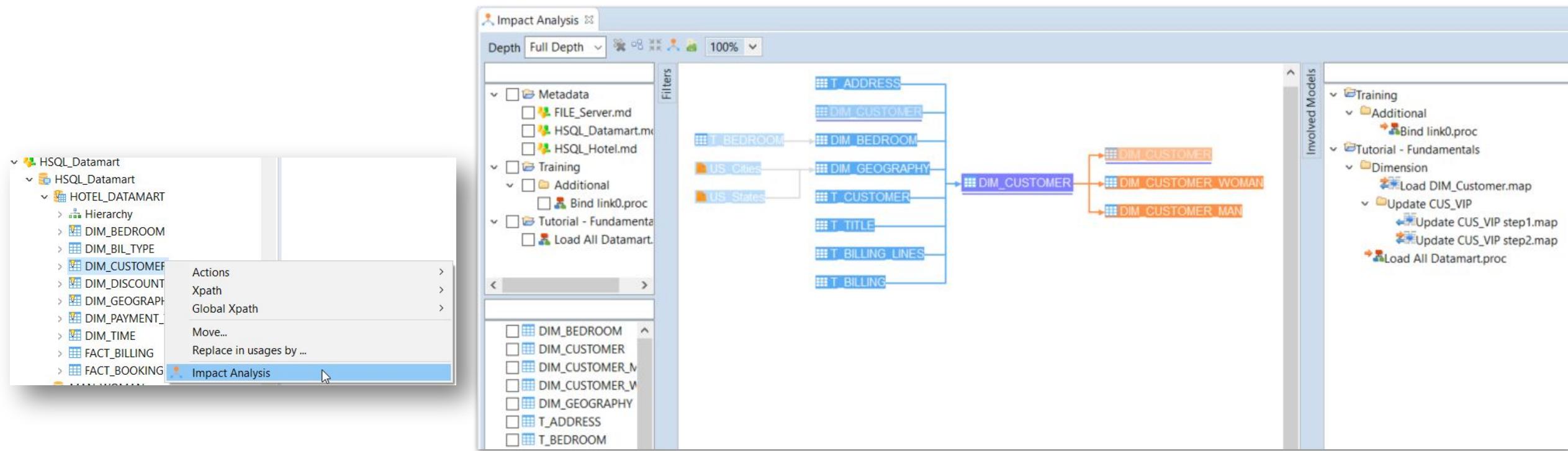
Navigate in Metadatas

The “Link selection in Project Explorer” button to find the related Metadata in the Project Explorer



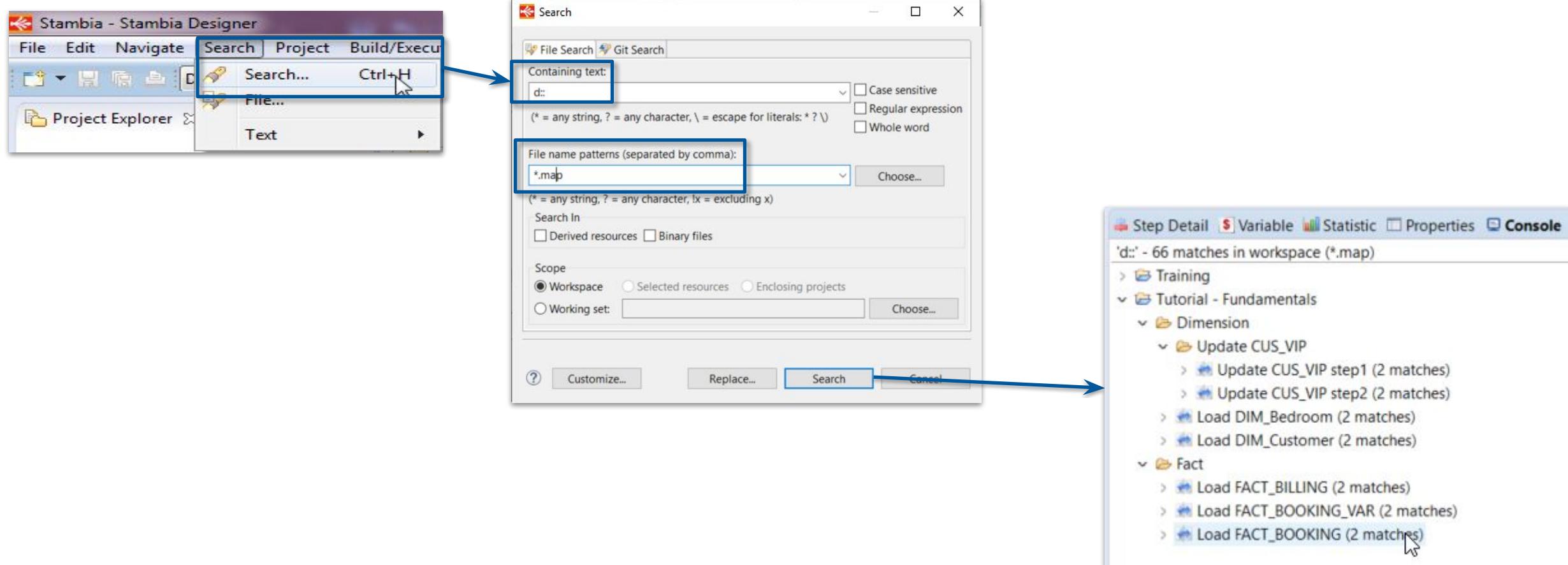
As the 'Link with Editor' button, it allows to focus in the Project Explorer on the select item, with the difference that the new one performs a one-shot link

Impact management : impact analysis



Navigate in resource files

Use the search menu to find text in the Workspace





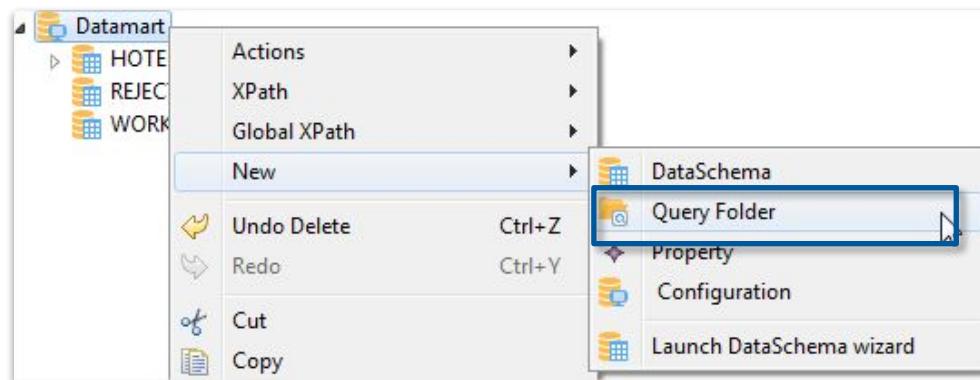
Query Metadata



Query metadata

It's possible to define a query as a metadata to use it in a mapping

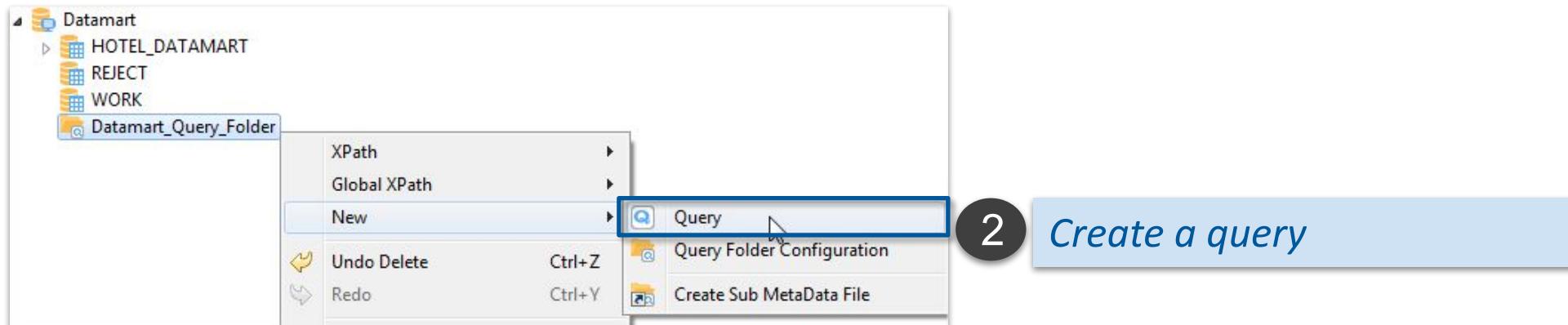
1. Create a query folder



Create a query folder (define a name) to store the query to use on this database

Query metadata

2. Create a query and define the SQL expression



3. Define the name & the SQL expression

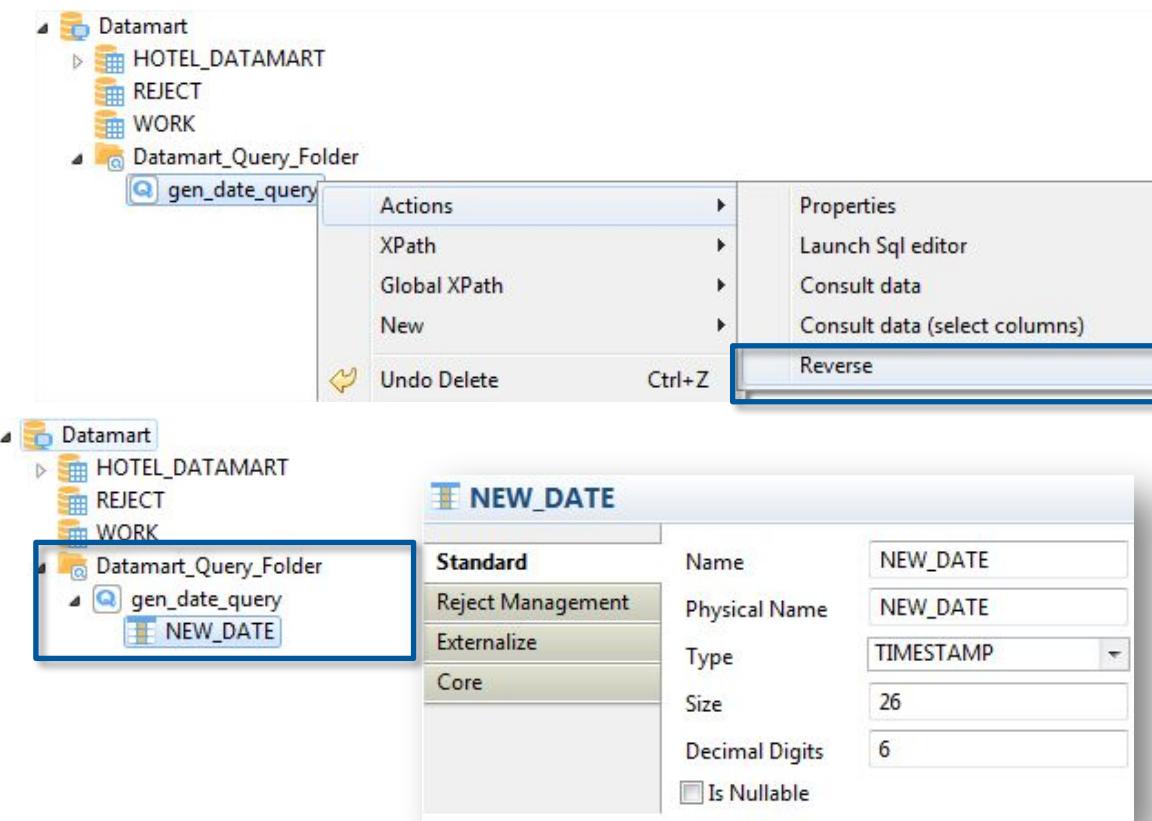
The screenshot shows a configuration dialog for a query named 'gen_date_query'. The dialog has tabs: 'Standard' (selected), 'Externalize', and 'Core'. The 'Name' field is set to 'gen_date_query'. The 'Expression' field contains the following SQL code:

```
SELECT CAST(current_date - value_to_subtract DAY AS TIMESTAMP) AS NEW_DATE  
FROM UNNEST(SEQUENCE_ARRAY(1, 1000, 1)) AS generate_series(value_to_subtract)
```

A callout bubble labeled '3' points to the 'Expression' field, with the text 'Define a name and the SQL expression WITH a defined alias on each expression on the SELECT clause' inside.

Query metadata

4. Save the query and finally “Reverse” the query to retrieve the columns

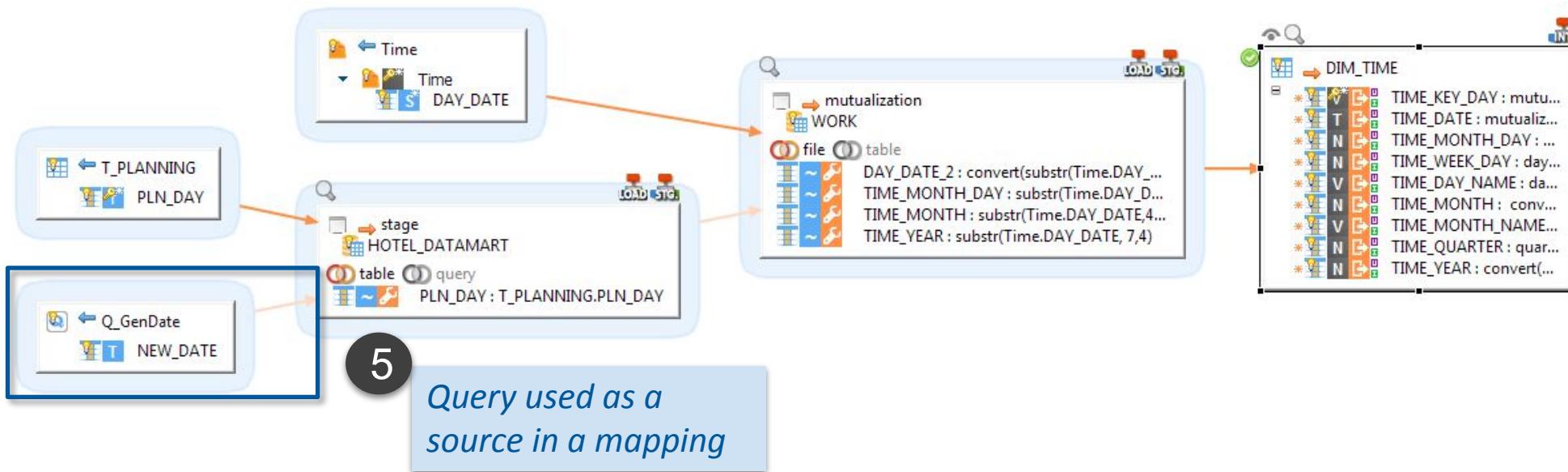


4

Reverse to have the corresponding columns of the alias in the expressions of the SELECT clause

Query metadata

5. It's now possible to use this query in mappings





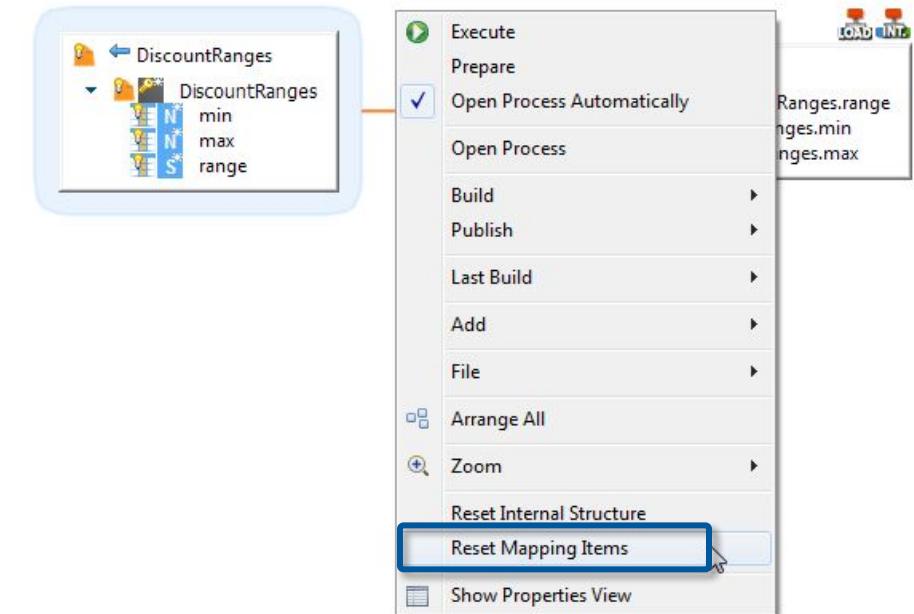
**Tips to fix mapping,
processes and locked
demo databases**



How to repair mapping and processes ?

To repair mapping

- Right click & “Reset Mapping Items” inside a mapping



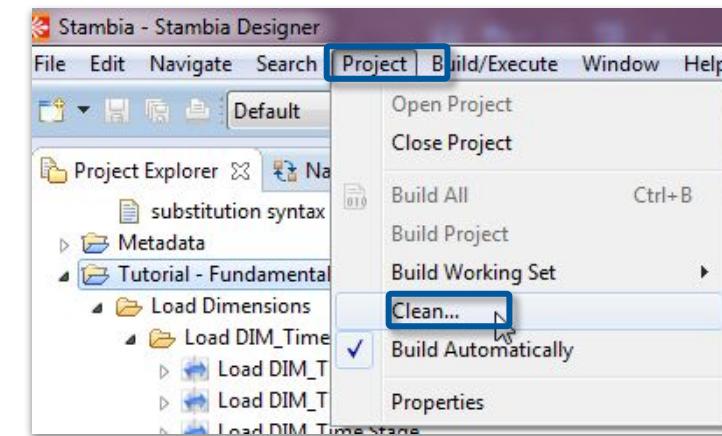
- Right click in Project Explorer, Advanced, Reinit diagram



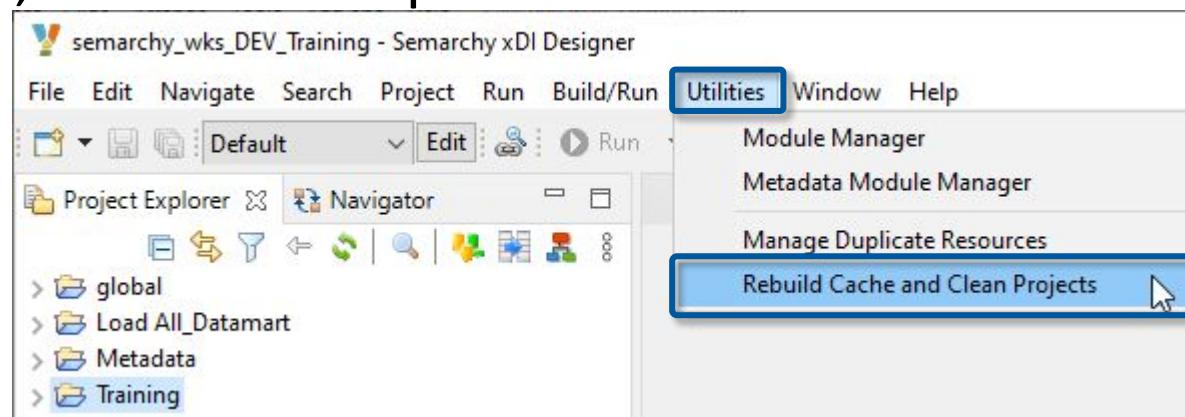
How to repair mapping and processes ?

To repair mapping or processes

- In menu, to clean a project



- In impact view, for the workspace

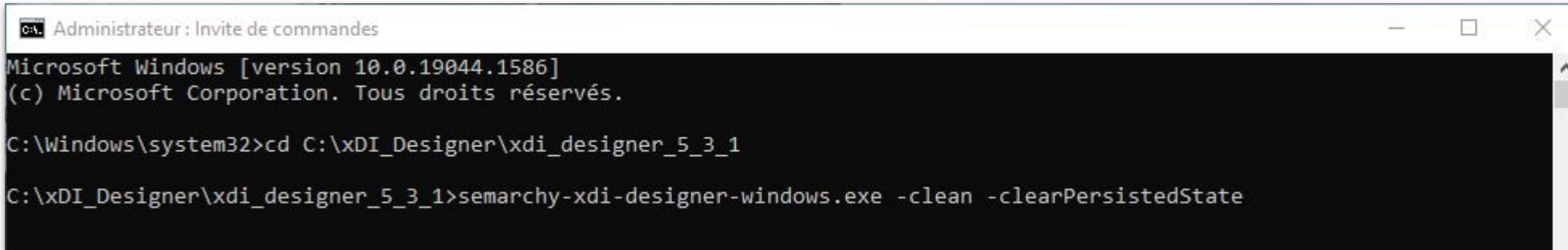


How to repair a workspace

To clean a workspace with errors

- Close your Designer
- Enter in command line, under the Designer folder

semarchy-xdi-designer-windows.exe -clean -clearPersistedState



```
C:\ Administrateur : Invite de commandes
Microsoft Windows [version 10.0.19044.1586]
(c) Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>cd C:\xDI_Designer\xdi_designer_5_3_1

C:\xDI_Designer\xdi_designer_5_3_1>semarchy-xdi-designer-windows.exe -clean -clearPersistedState
```

To go further

Document Type	Link
English video Learn about Data Reject management	https://www.youtube.com/watch?v=iJPSVGjDUpM
Stambia.org article Using Metadata queries	https://stambia.org/doc/77-development-hints-and-tips/metadata/216-using-metadata-queries
Stambia.org article Getting started with User Defined Functions	https://stambia.org/doc/218-development-hints-and-tips/user-defined-functions/537-getting-started-with-user-defined-functions
Stambia.org article Getting started with User Defined Functions (2)	https://stambia.org/doc/218-development-hints-and-tips/user-defined-functions/685-getting-started-with-user-defined-functions-2
Stambia.org article Changing the name of the temporary tables	https://stambia.org/doc/77-development-hints-and-tips/metadata/176-changing-the-name-of-temporary-tables
Stambia.org article Combining multiple sources with SQL set operator	https://stambia.org/doc/221-development-hints-and-tips/mapping-stages/672-combining-multiples-sources-with-sql-set-operators
Stambia.org article Defining Data Types on stage fields	https://stambia.org/doc/221-development-hints-and-tips/mapping-stages/539-defining-data-types-on-stage-fields



Questions?

