

# Web Data Integration

# Introduction to

# MapForce



# Hands-on: Schema Mapping with MapForce - Movies

## 1. Load files

- movie\_list.csv as input
- target.xml as target schema

## 2. Assign IDs

- The id should start with the prefix *movie\_list\_id* followed by an increasing number which starts at 1000.

Increment: **auto-number** function

Add prefix to incrementing number: **concat** function

Foreign Gross: **if** the value is numeric add it **else** add **0**

Date: Add prefix, define the **format**, parse the concat values as date

## 3. Create attribute correspondences from **source** to **target**

- Map **Film** and **Genre** to the corresponding attributes of the target schema
- Set the **gross** attribute value as a summation of the **Domestic** and the **Foreign Gross**. (*hint*: You can only perform summation if all the involved values are numeric)
- Map **year** to **date** and set the day and month to the first of June.

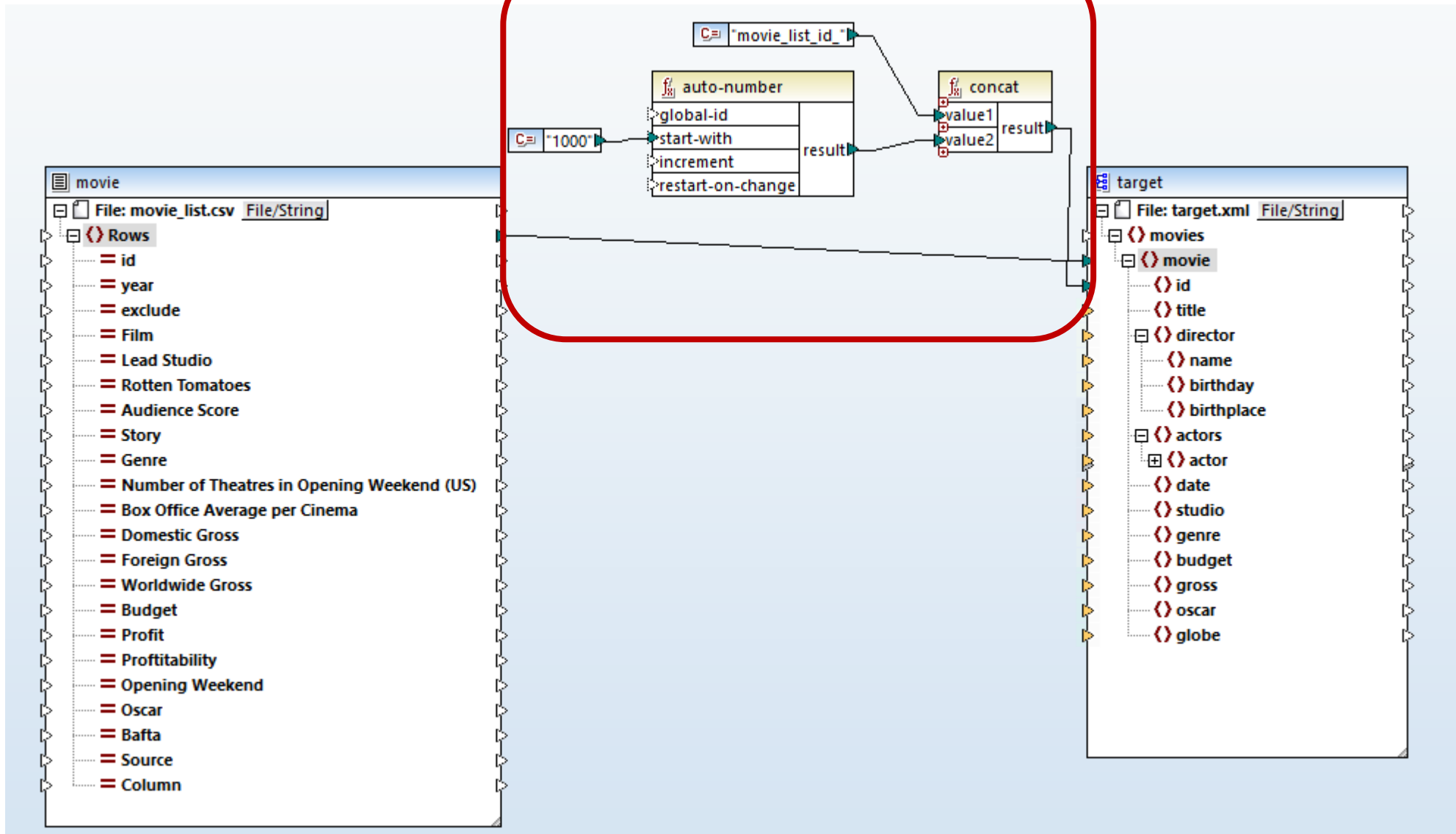
Check if exclude value **equals** y. On **false** map the source row to the target movie

## 4. Filter

- Exclude the movies where the exclude attribute is set to y

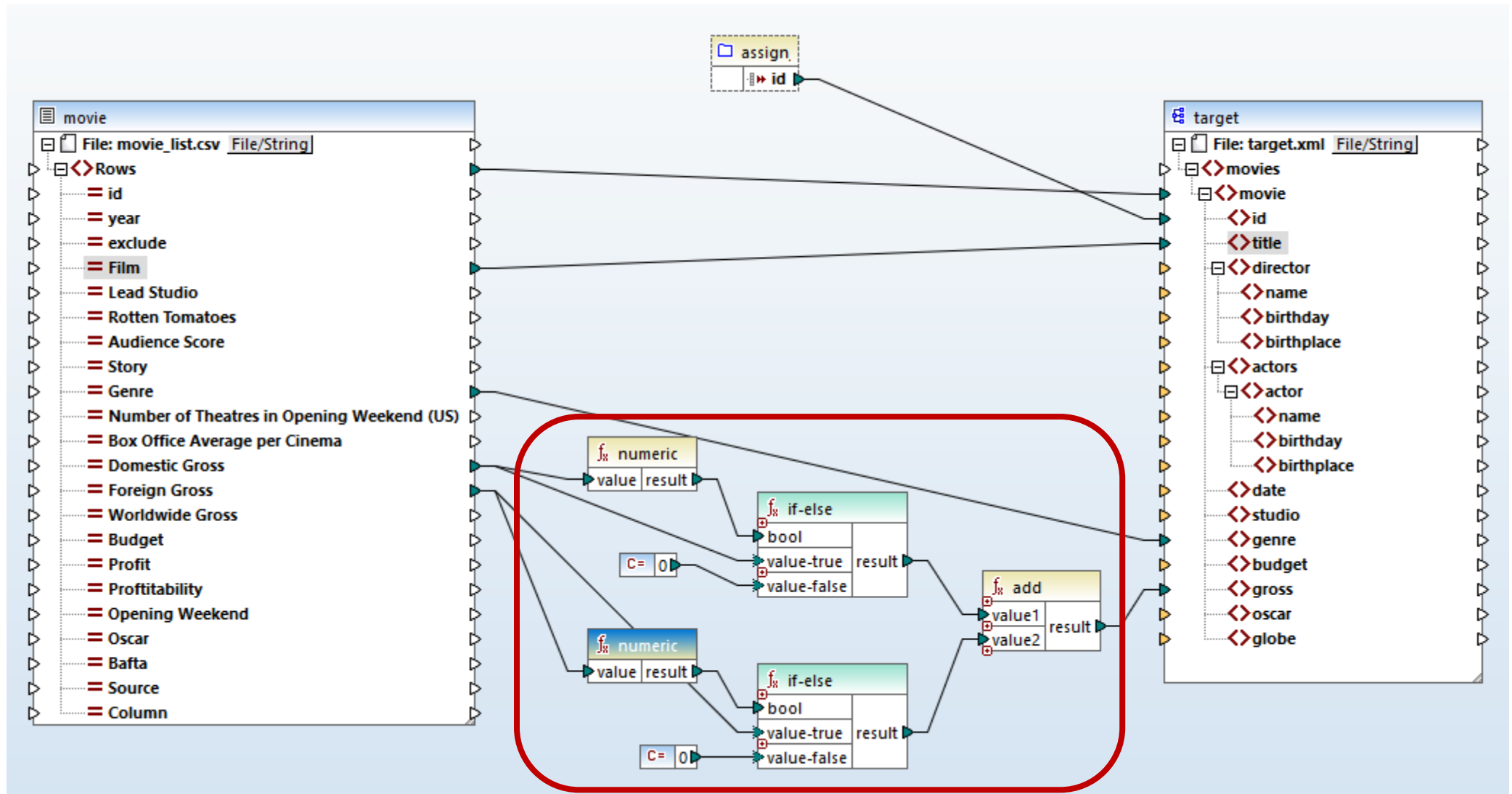
# Solution: Schema Mapping with MapForce – Movies

## – Assign IDs



# Solution: Schema Mapping with MapForce – Movies

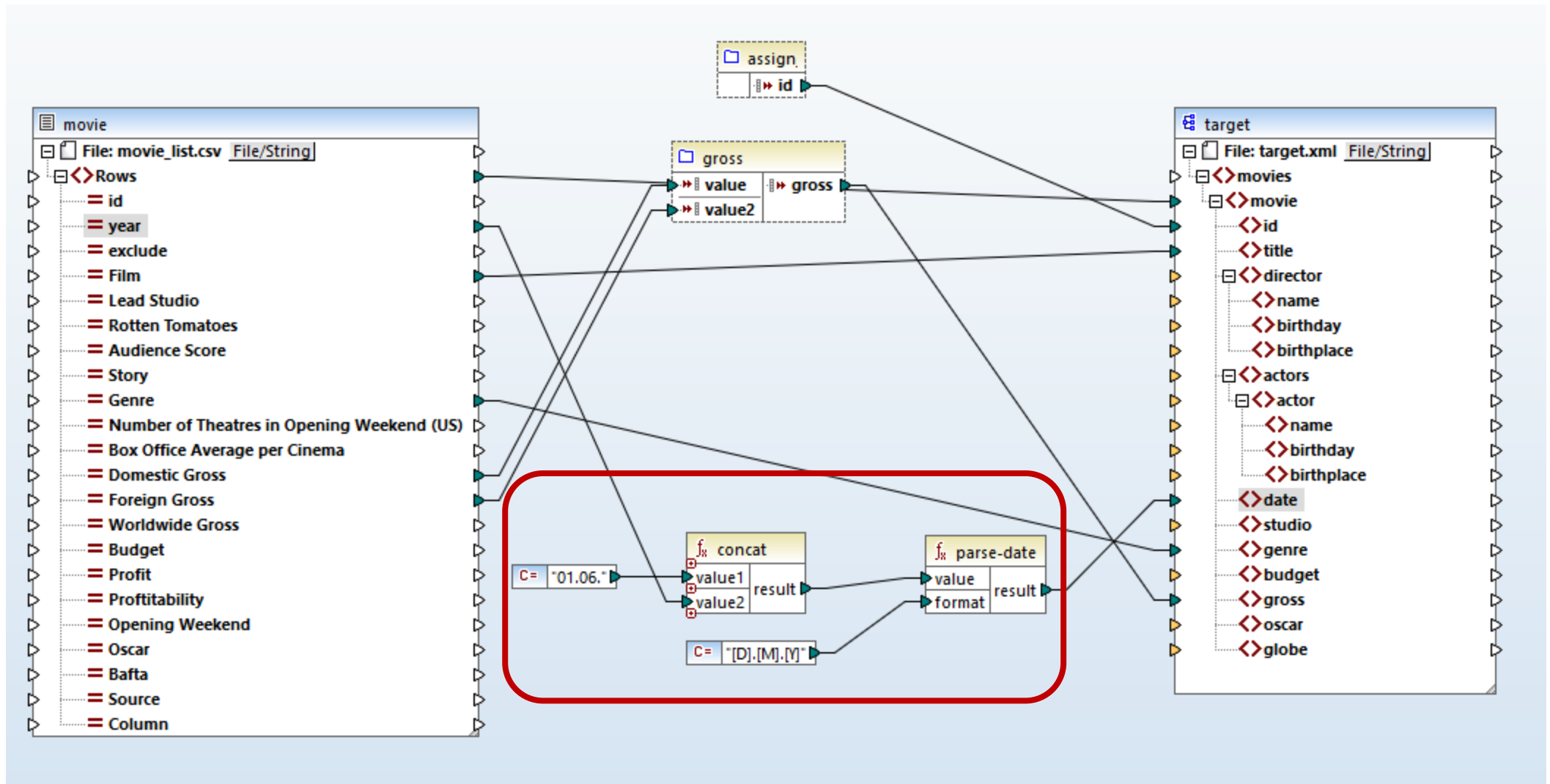
- Create attribute correspondences using summation





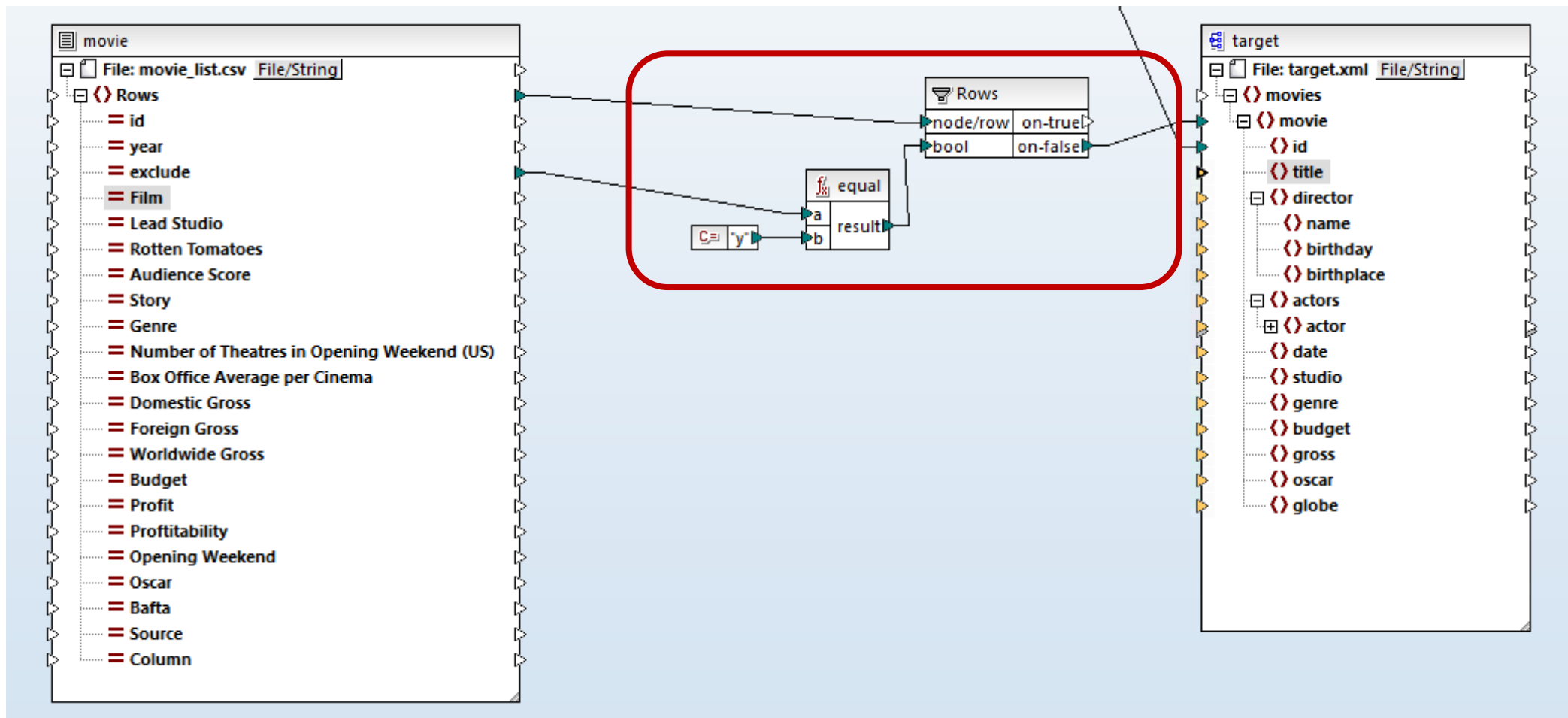
# Solution: Schema Mapping with MapForce – Movies

- Create attribute correspondences using date functions



# Solution: Schema Mapping with MapForce – Movies

## – Filter out entities



# Hands-on: Schema Mapping with MapForce - Actors

## 1. Load files

- actors.csv as input
- target.xml as target schema

The input file is **tab** separated. Define the **datatype** and **names** of the attributes.

## 2. Aggregate by Movie

- Identify an appropriate mapping key (the name of the movie is not enough as there might be movies with the same title)
- Group by the created key. Use the key as id for the target dataset.
- Map the aggregated rows to the movies in the target schema

Use **normalize space** function to remove the spaces. **Concatenate** the birthday day, month, year in a parsable **format** like **[D].[M].[Y]**

## 3. Create actor correspondences

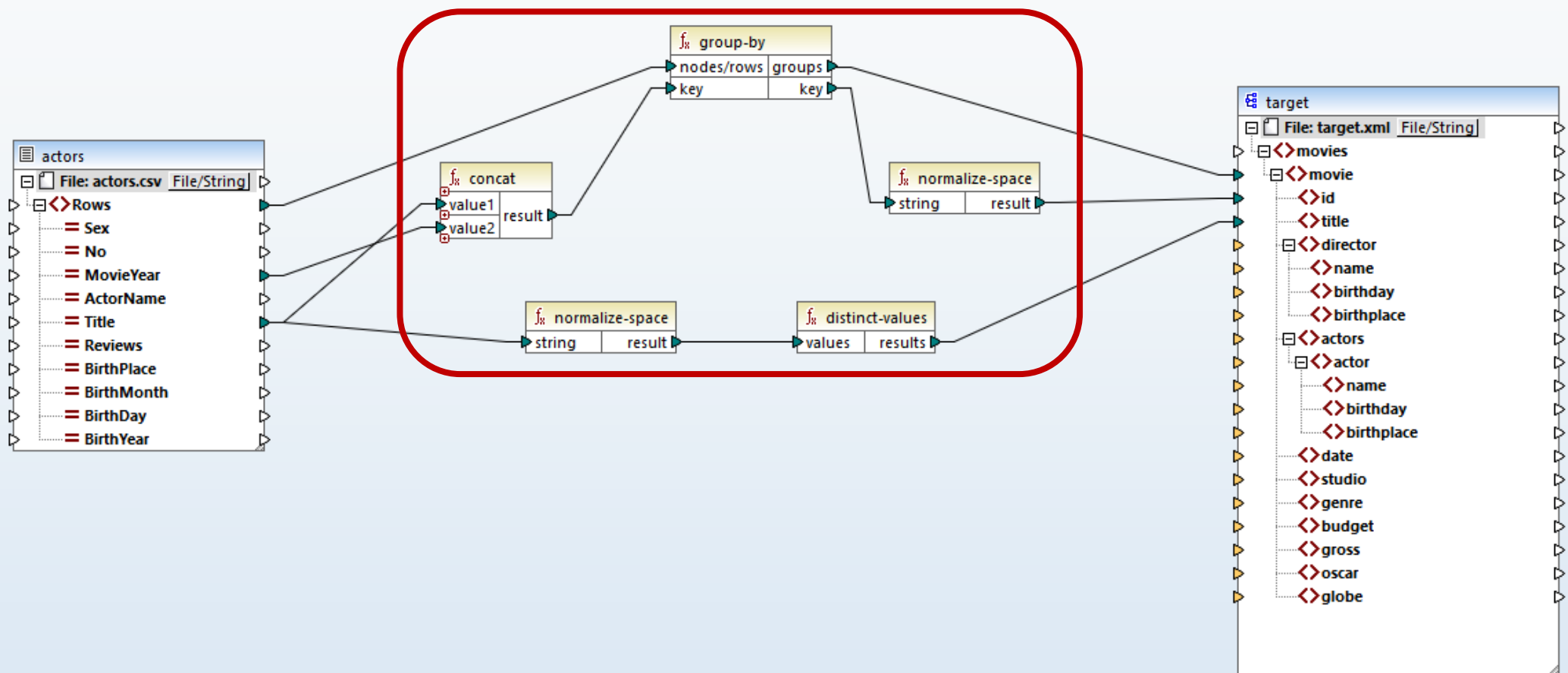
- Create the missing correspondences for the name, birthplace and the full birthday
- In case the values contain spaces, remove them

### File Format:

Gender	No	MovieYear	ActorName	Title	Reviews	BirthPlace	BirthMonth	BirthDay	BirthYear
--------	----	-----------	-----------	-------	---------	------------	------------	----------	-----------

# Solution: Schema Mapping with MapForce – Actors

- Aggregate by movie





# Solution: Schema Mapping with MapForce – Actors

- Create actor correspondences

