hale Workshop

Introduction to new features

22.05.2018

Agenda (1/2)

hale studio

- Project variables
- Improved alignment migration
- hale connect integration
- Spatial Index / Join
- Split GML by feature type
- SQL queries as data source

Agenda (2/2)

hale Command Line Interface

- Merging alignments
- Custom success conditions for transformations
- Rewrite data

Plugins

- GeoServer AppSchema: Support for isolated workspaces
- XtraServer configuration Import & Export

hale studio

Project variables (3.0)

Use case:

- Make variables in a transformation configurable for individual transformation runs

How to use?

- Configure project variables (default values) in the transformation project
- Use them in transformation functions (e.g. Assign, Formatted String, Groovy scripts)
- Override using environment variables or system properties

Improved alignment migration (3.4)

Use case:

- Change / update source or target schema (e.g. to a new version)
  - Examples:
    - Migrate alignment from one INSPIRE version to the next
    - Updated aligmnent to changed source schema

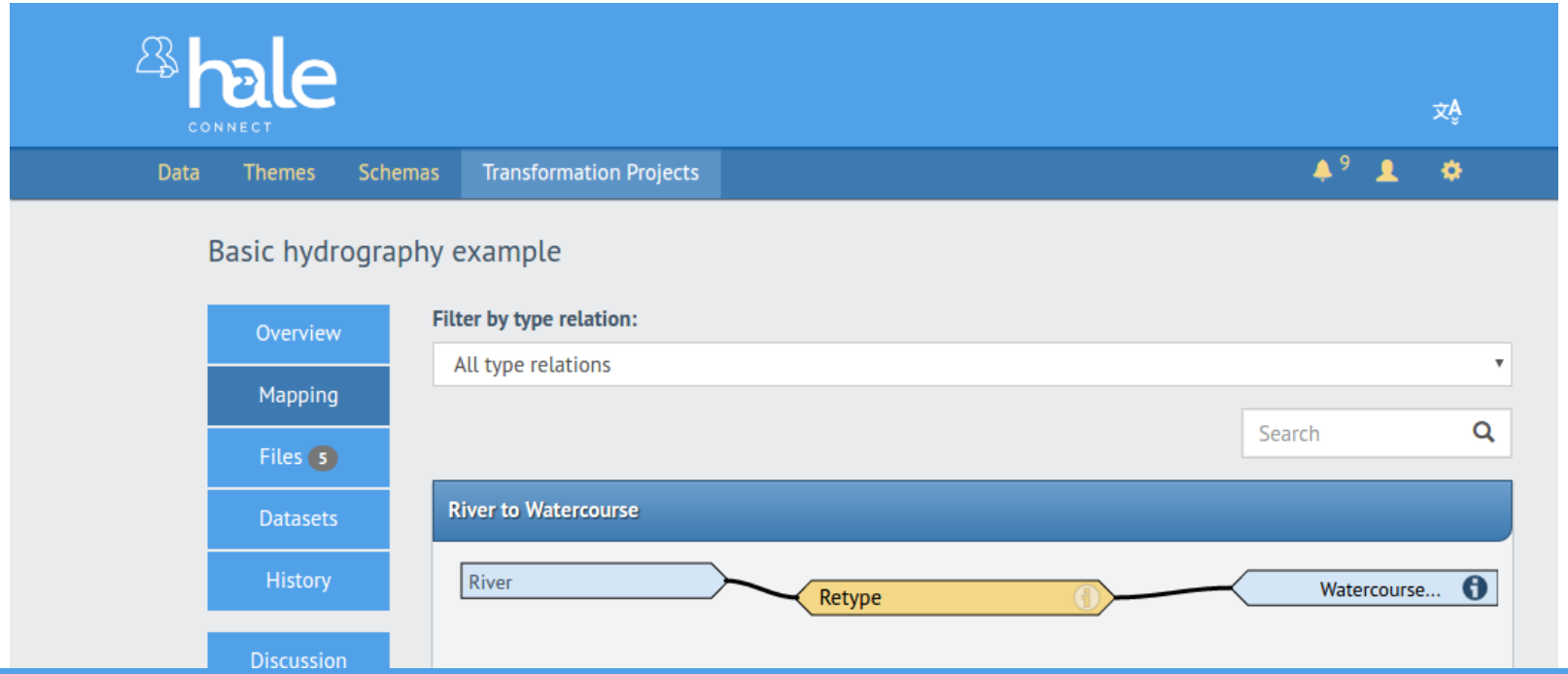
How to use?

- Use „Reload and update schema“ (replace target schema) **or**
- Import existing alignment into a new project **or**
- „Merge“ two alignments (see example shown later)

hale connect integration (3.3)

Directly interact with hale connect from hale studio:

- Load and save projects
- Export projects for sharing



Spatial Index / Join (3.3)

Use case:

- Find objects that are related spatially to a certain geometry / object

How to use?

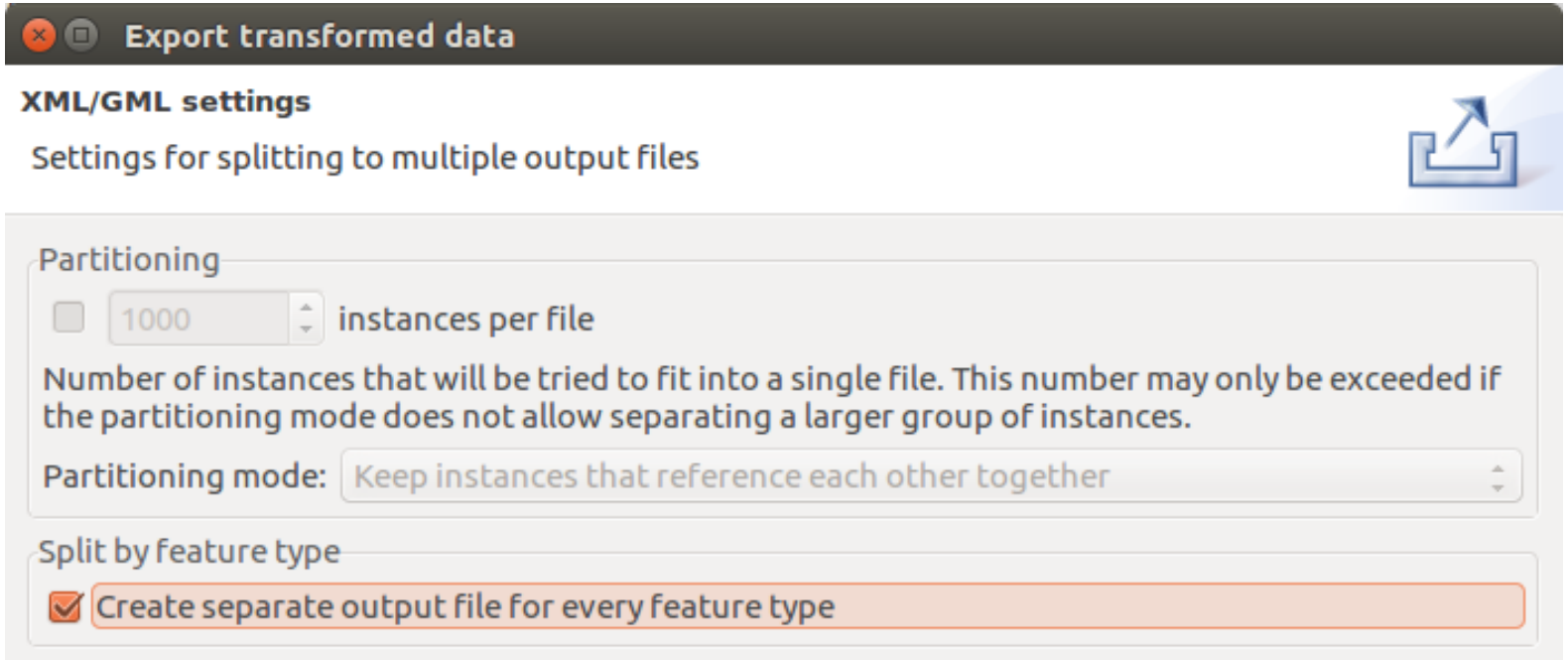
- Using the Spatial Join transformation function **or**
- Using the *spatialIndexQuery* helper function in Groovy scripts

Split GML by feature type (3.4)

GML output can be split by feature types

- One file per feature type
- References across feature types are adapted to include the file name

Note: In hale studio 3.4 there is a bug regarding the naming of the created files.

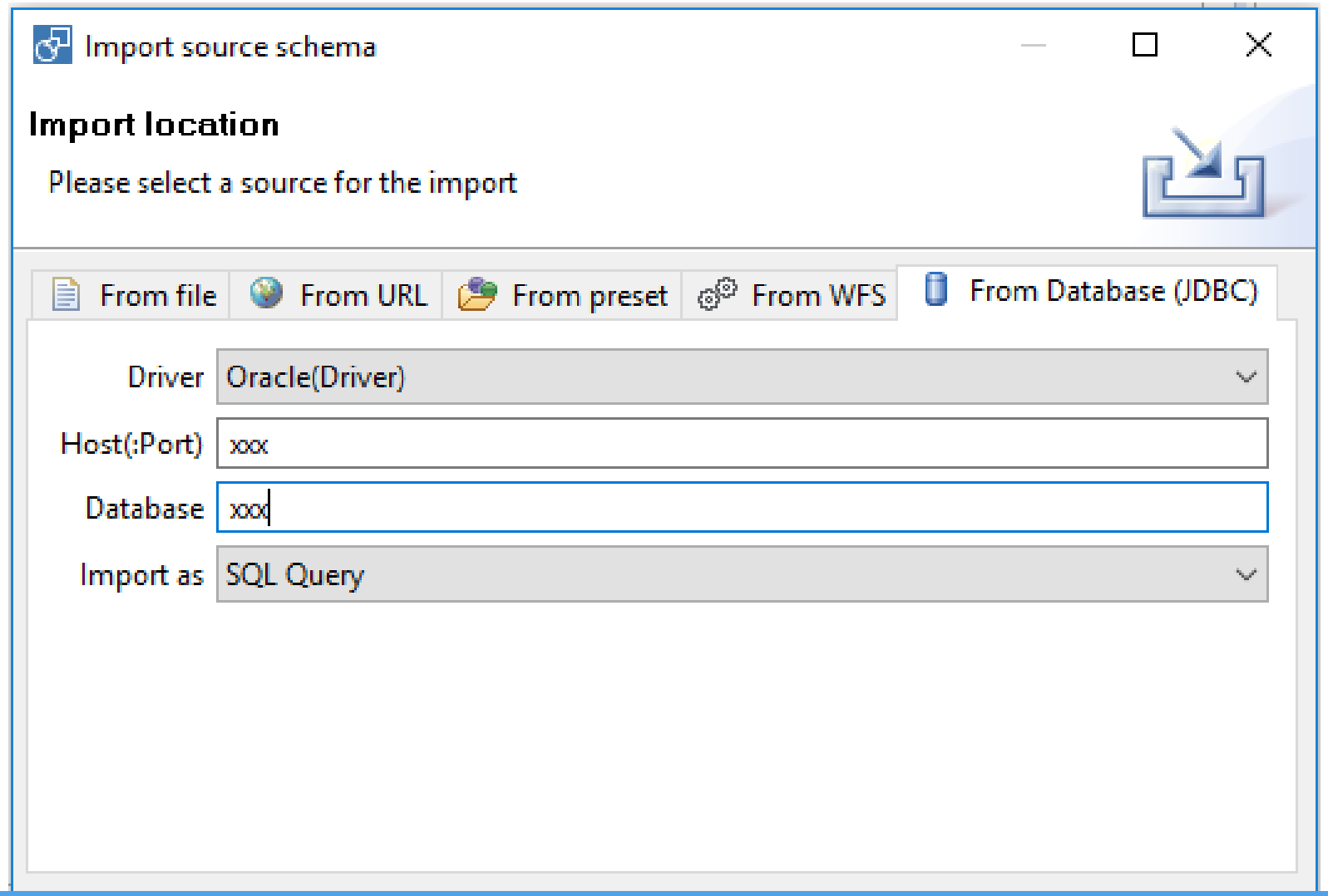


SQL queries as data source (3.3)

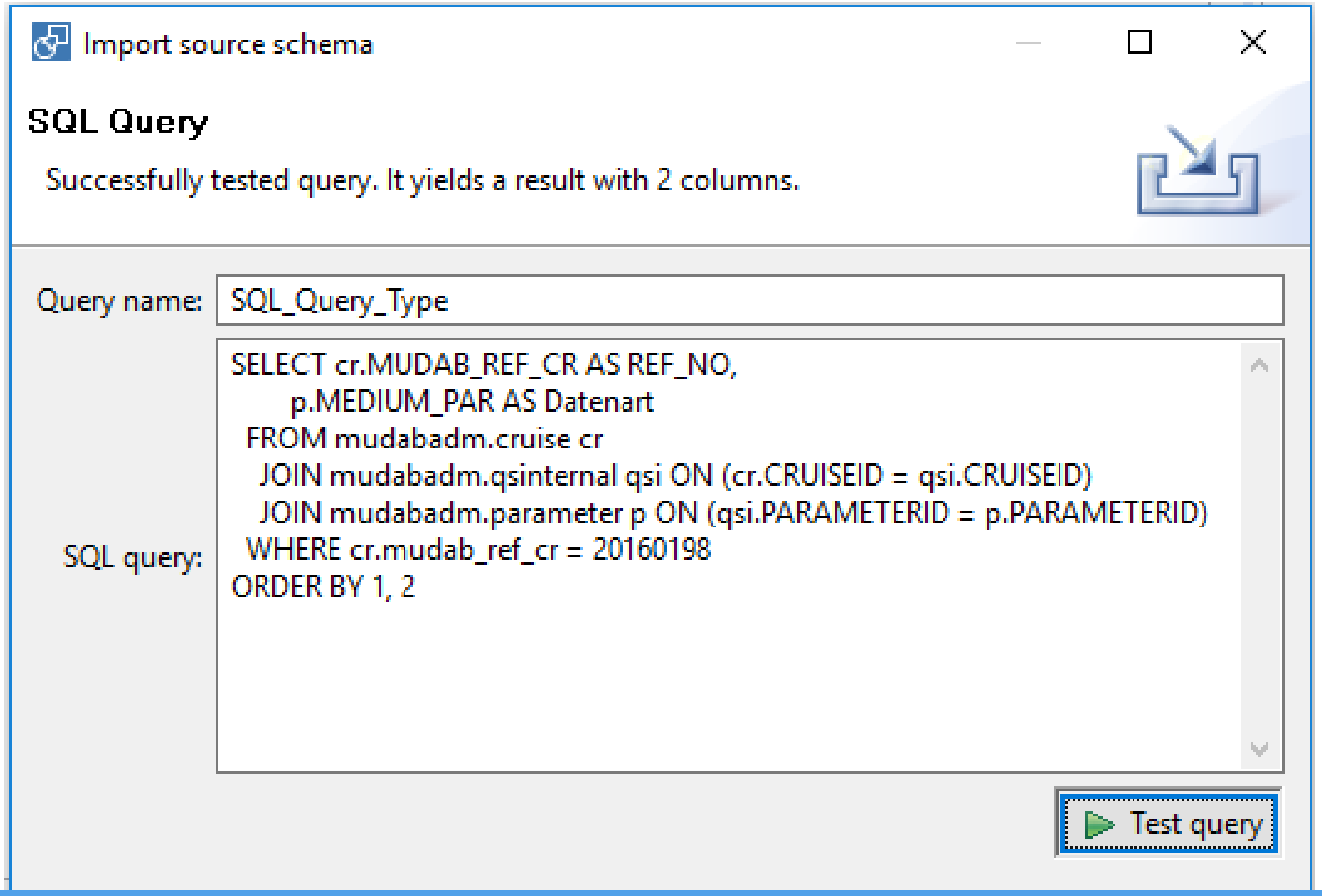
Use case:

- Support transformation from database side (but views can or shall not be used)
- Parameterisable queries via project variables (e.g. dynamically adjust WHERE clause)

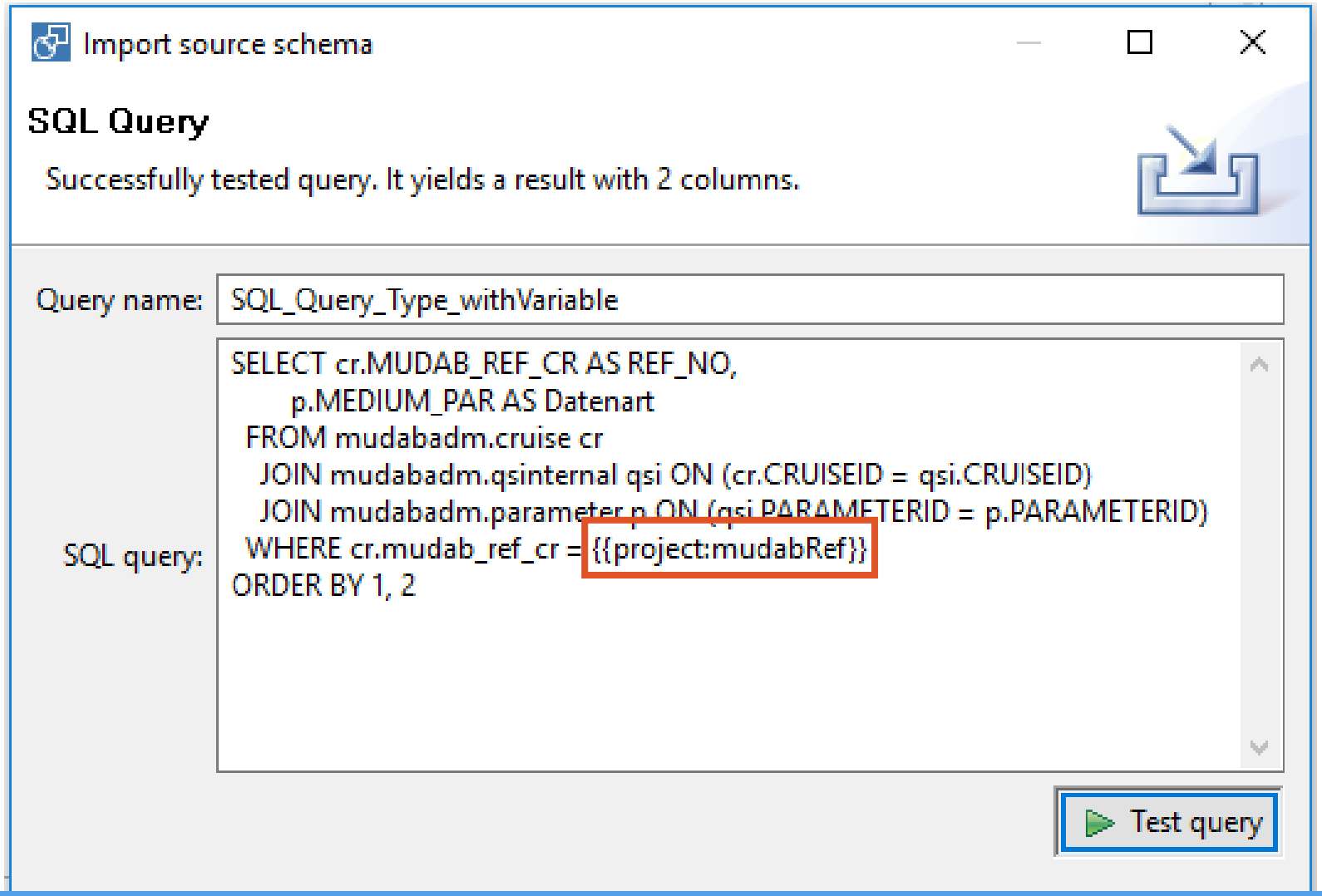
SQL queries as data source (3.3)



SQL queries as data source (3.3)



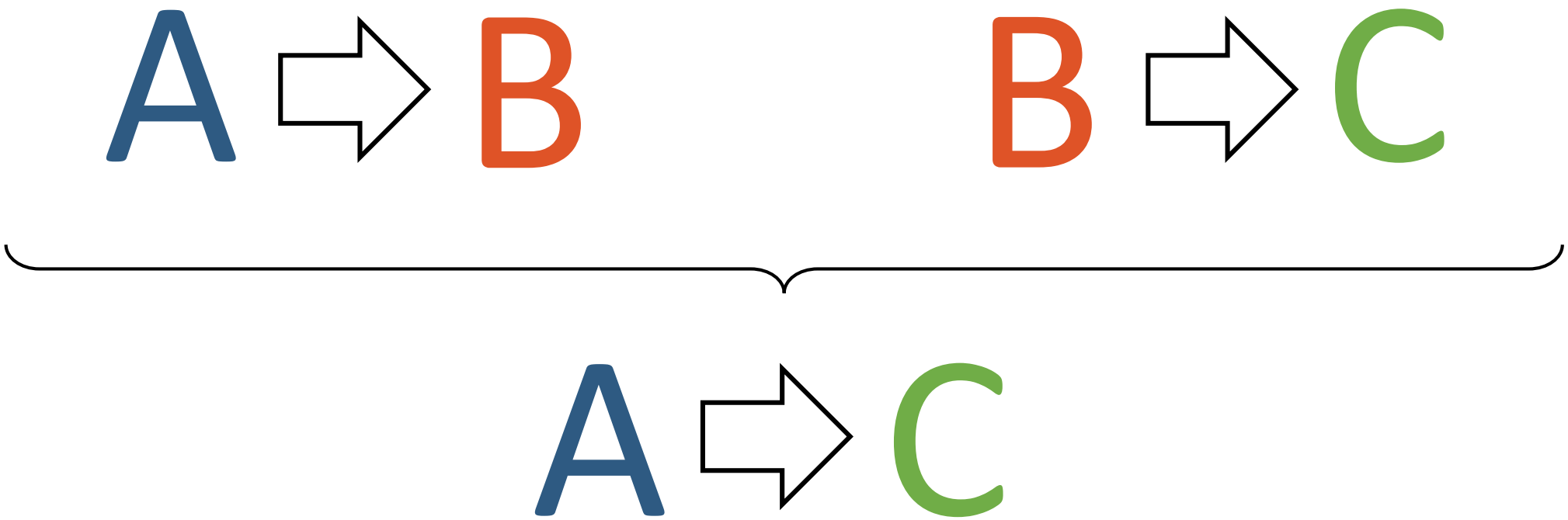
SQL queries as data source (3.3)



hale CLI

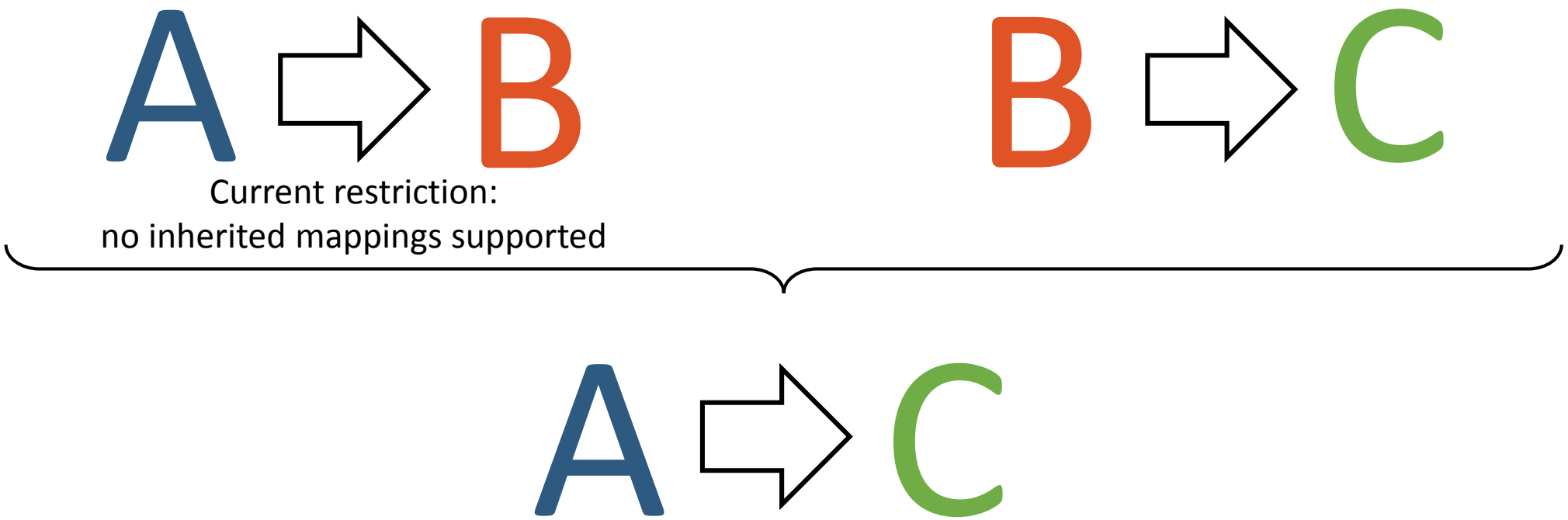
Merging alignments (3.4)

Combine two hale alignments



Merging alignments (3.4)

Combine two hale alignments

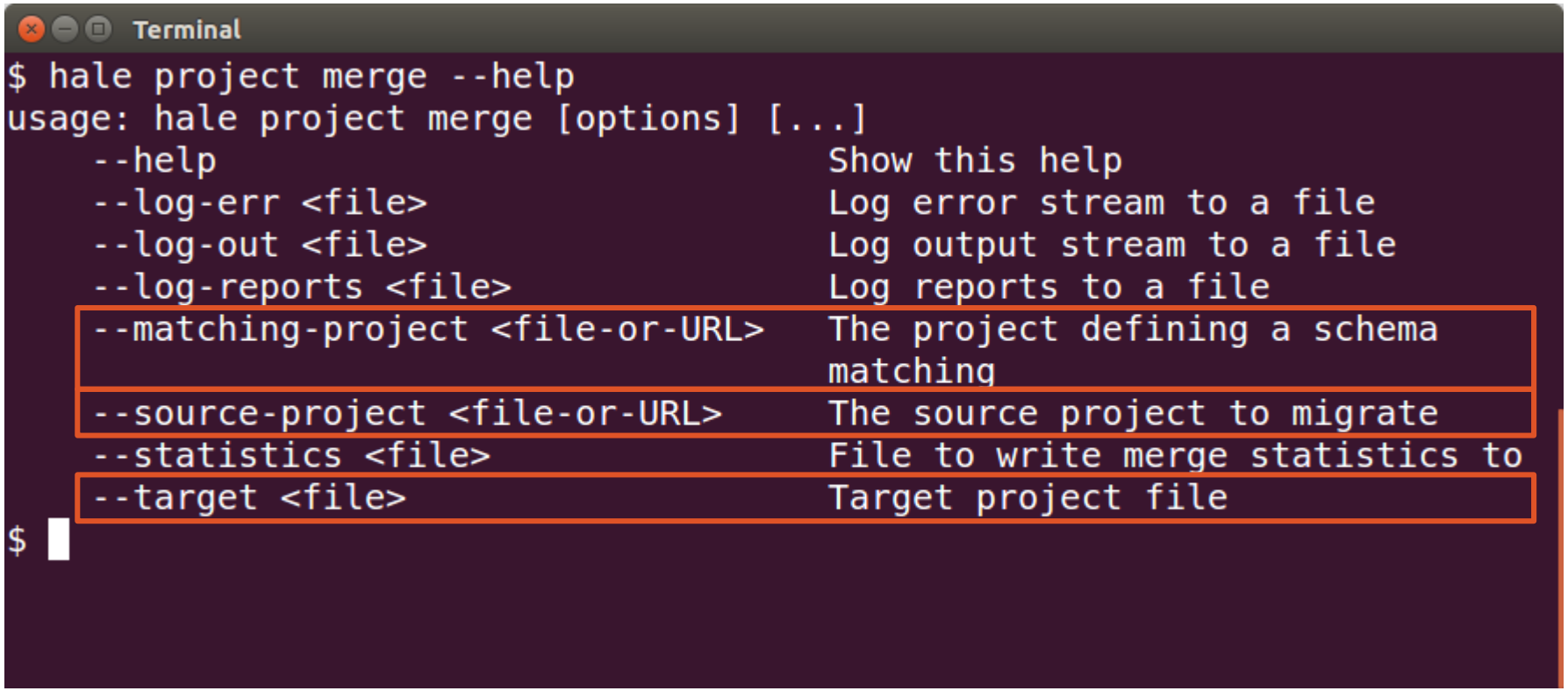


Merging alignments (3.4)

Command in hale-cli

A ⇨ B

B ⇨ C



A ⇨ C

Merging alignments (3.4)

Result:

- Combined project/alignment
- Necessary changes can only be done automatically in part
- Manual verification/changes required
  - Support via hints where changes may be necessary

Error Log | Properties | Tasks

Description	#	Status
Cell messages	19	
o42001, o42001__mat, o42001__anl, o42001__fdv, o42001__p020001020002, o42001__p020005020006, o42001__p0000103000 to RoadA	2	
A script was associated to the merged Groovy Retype this cell is derived from, it was used as script for this cell's script and needs to		new
The filter on the original source has been dropped because transfering it was not possible automatically. The original filter was: "CQ		new
o71001, o71001__mat, o71001__anl, o71001__fdv, o71001__p020001020002, o71001__p020005020006, o71001__p0000103000 to RoadA	1	
A script was associated to the merged Groovy Retype this cell is derived from, it was used as script for this cell's script and needs to		new
o42009, o42009__mat, o42009__anl, o42009__fdv, o42009__p020001020002, o42009__p020005020006, o42009__p0000103000 to RoadS	2	
A script was associated to the merged Groovy Retype this cell is derived from, it was used as script for this cell's script and needs to		new

Custom success conditions for transformations (3.4)

Use case:

- Specific conditions when a data set should be further processed after transformation
  - Example: Transformation is only considered successful if
    - XML Schema validation was run and yielded no errors / warnings
    - At least one object (of a certain type?) was created during the transformation
    - No errors occurred during the transformation

Note: The default behavior for transformation success (return code zero) is that the transformation completed (errors may have occurred), a result was written and validations on the written data were successful.

Custom success conditions for transformations (3.4)

How to use?

- Specify Groovy script to be evaluated on transformation statistics **or**
- Process transformation statistics (Json) with a tool of your choice

Options for transformation command line:

```
-successEvaluation <file-or-URI-to-script>
-statisticsOut <statistics-file>
```

Rewrite data (3.3.1)

Use case:

- Rewrite data with hale without the need to rerun a transformation
  - Examples:
    - Apply a coordinate transformation
    - Split output in multiple files
    - Do an upload via WFS-T interface

How to use?

```
hale data rewrite --help
```

Plugins

GeoServer AppSchema: Support for isolated workspaces (3.4)

- Support for isolated workspaces (mutliple workspaces for the same namespace may exist)
- Requires GeoServer 2.13 (+ AppSchema plugin)

XtraServer configuration Import & Export (3.4)

- Requires XtraServer by interactive instruments
- Import of XtraServer mapping configuration (Database to XML/GML)
- Export to XtraServer mapping configuration (limited set of functions)

Questions?