Business intelligence

Tutorial 4 – Pentaho Data Integration - Extension

UNIVERSIDAD DE
MURCIA

Content

UNIVERSIDAD DE MURCIA

- 1 – Load closure
- 2 – Load dimension
  - Read XML with XPath
  - Set and read global variables
  - Update dimension
- 3 – Load prescription
  - Extract from CSV file
  - Stream lookup
  - Javascript

Setup

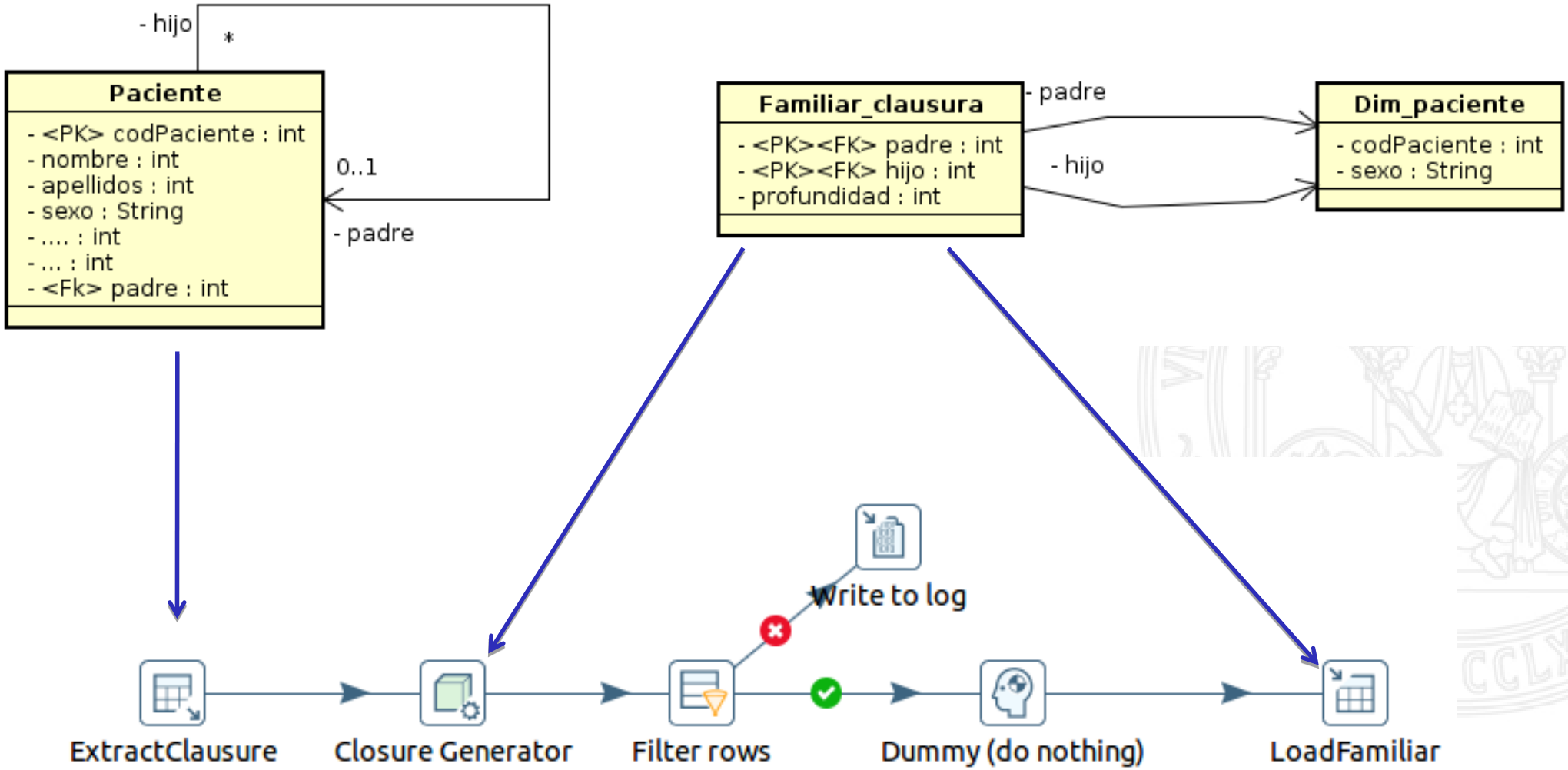
UNIVERSIDAD DE MURCIA

- We will extend the tut3 database with new tables
- Uncompress IN-2017-T4-extra_subir.rar
- Load SQL scripts
  - sql2-alter-schema-pacientes.sql
  - sql3-alter-schema-alumno.sql
    - IMPORTANT: change the “dw_prof” by your own schema.
  - sql-1-borrar-datos-esquema.sql: USE IT WHEN NEEDED.
- We use the same database connection as in tutorial 3

1- Load closure

UNIVERSIDAD DE MURCIA

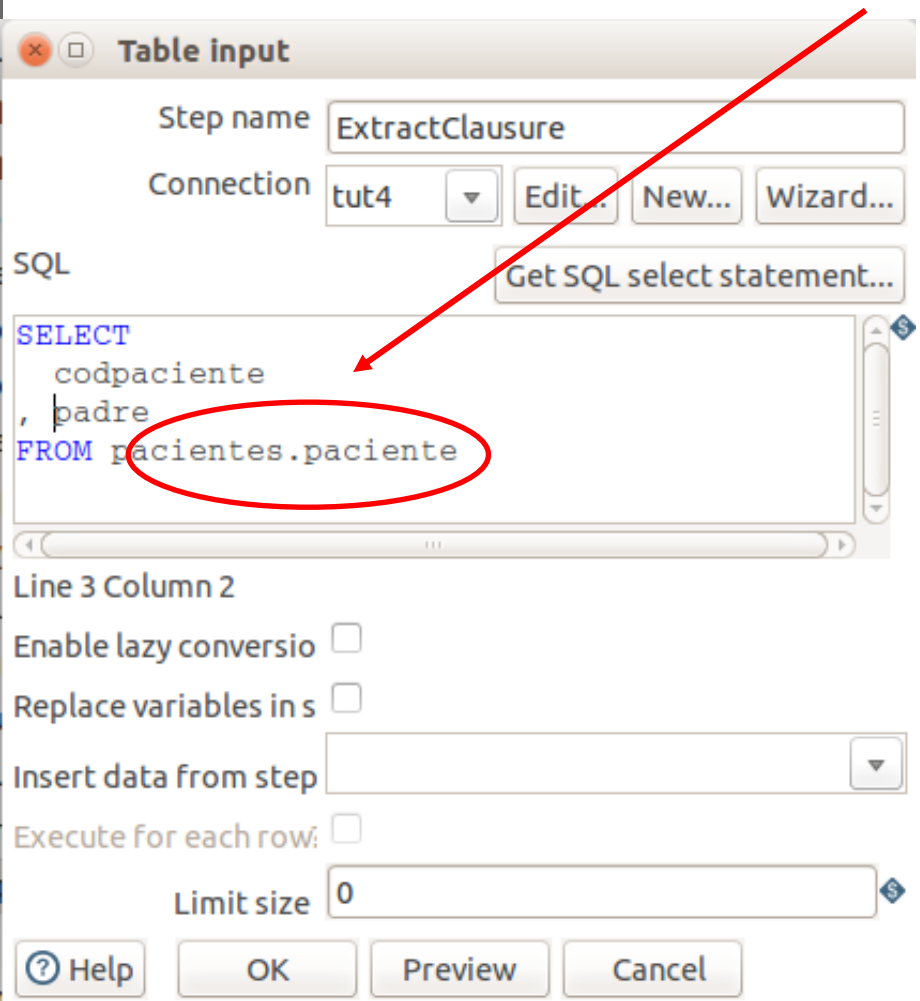
- Create a new PDI transformation



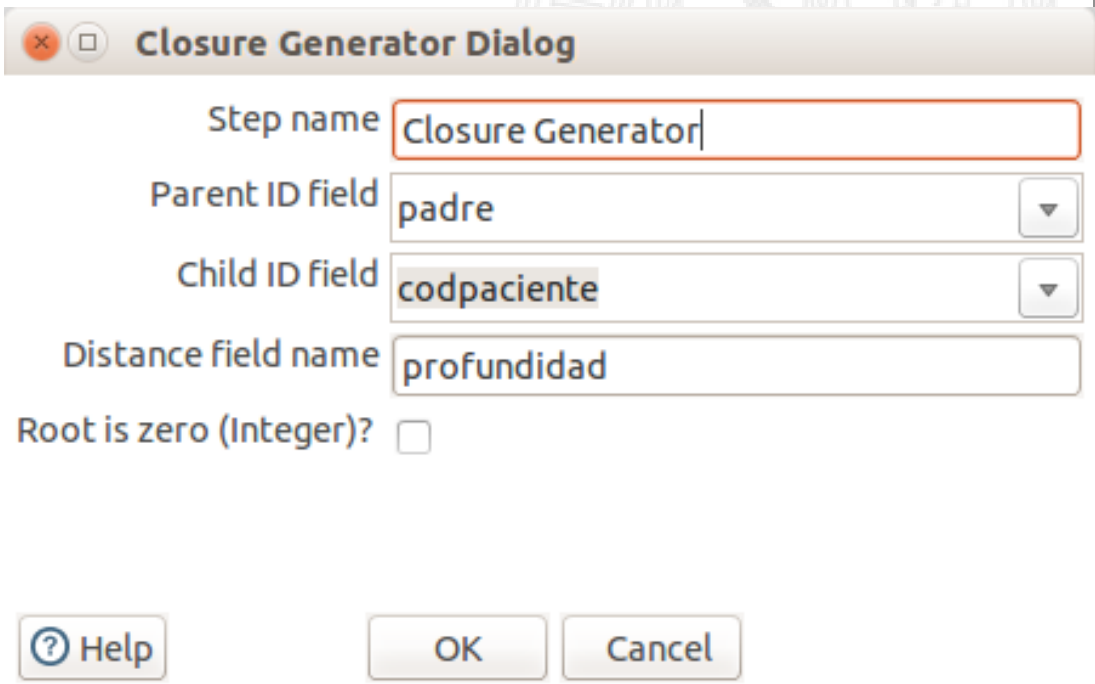
1- Load closure

UNIVERSIDAD DE MURCIA

- Input-> Table input
- Schema: “PACIENTES” clausura
- Fields: codpaciente, padre



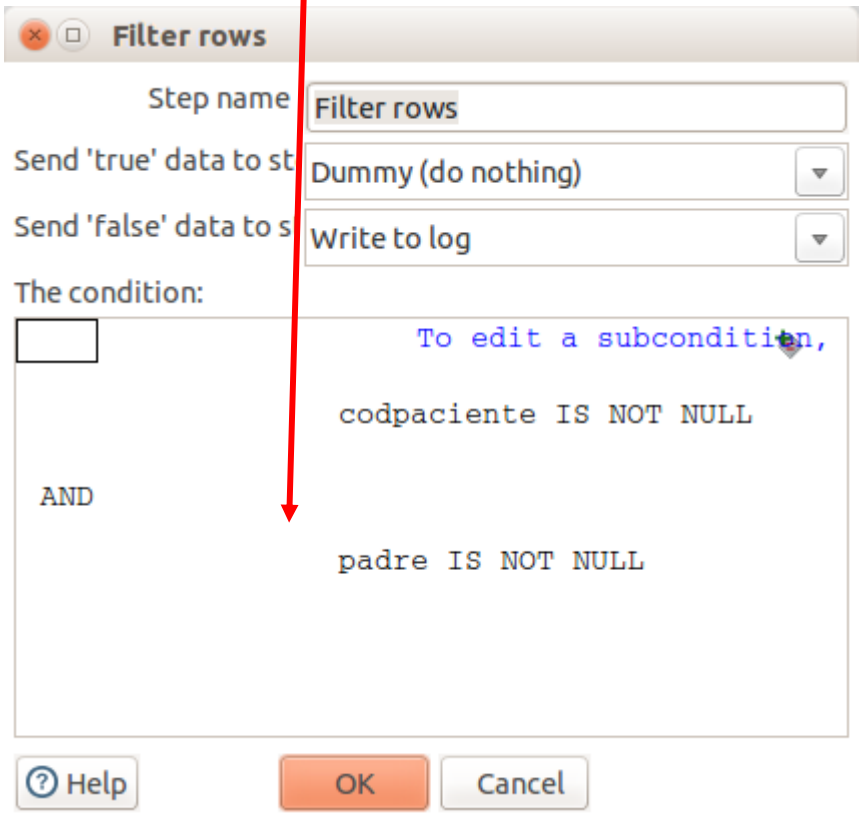
- Closure:
  - Transform -> Closure generator
  - Set “padre”, “hijo”, “profundidad”



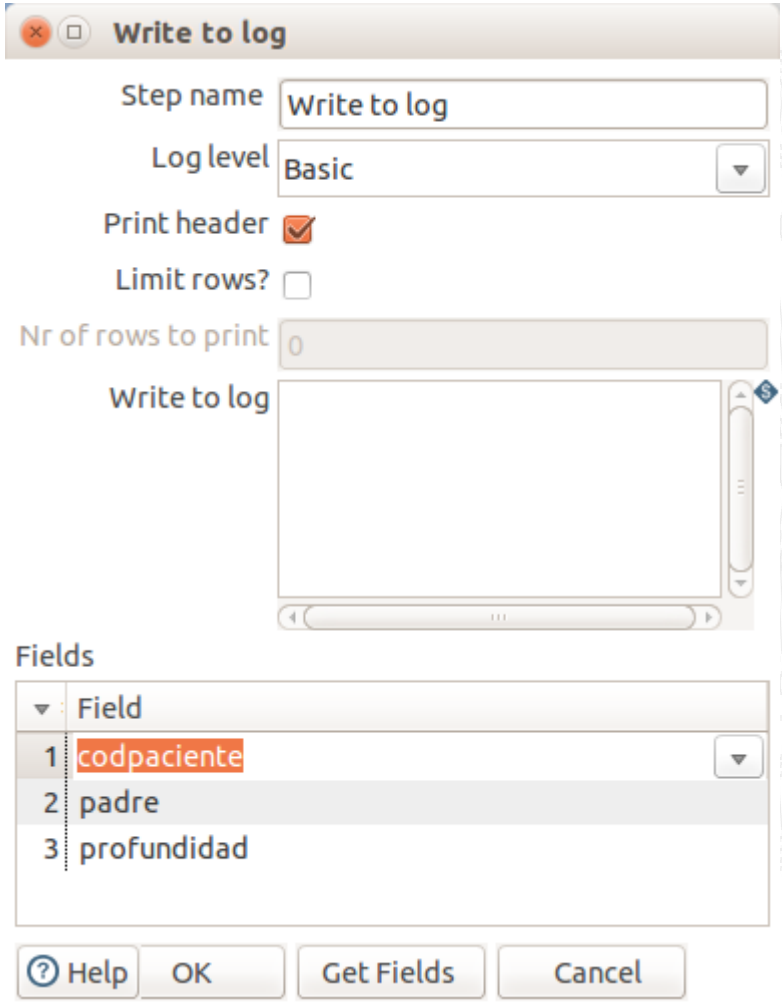
1- Load closure

UNIVERSIDAD DE MURCIA

- Filter:
  - Flow -> Filter
  - True: Dummy step
  - False: Write to log
  - Condition



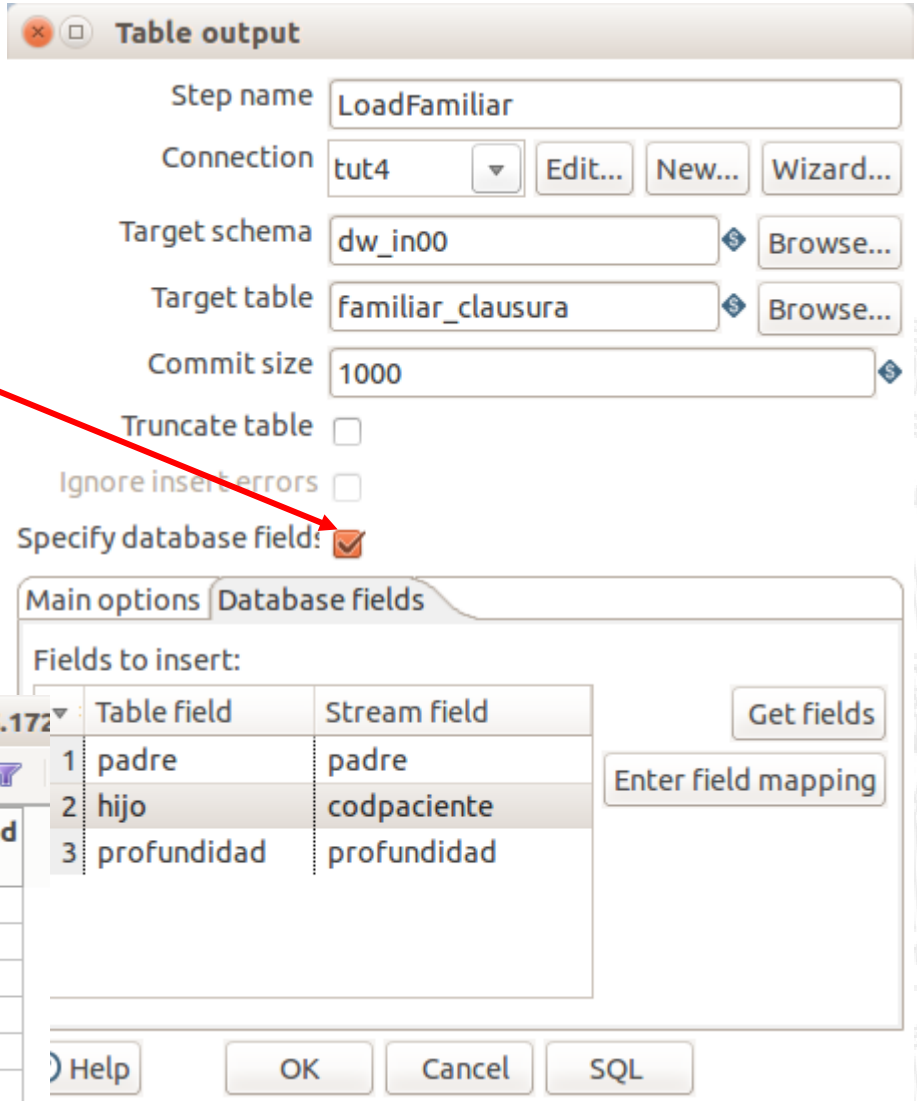
- Flow -> Dummy
- Utility -> Write to log
  - Set fields and log level

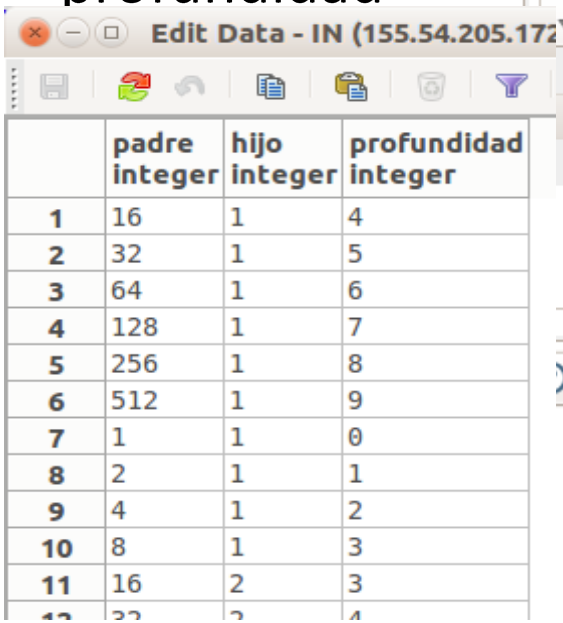


1- Load closure

UNIVERSIDAD DE MURCIA

- Store dimension:
  - Output -> Table
  - Target: “familiar_clausura”
  - Map fields
    - Padre -> padre
    - Hijo -> codpaciente
    - Profundidad -> profundidad



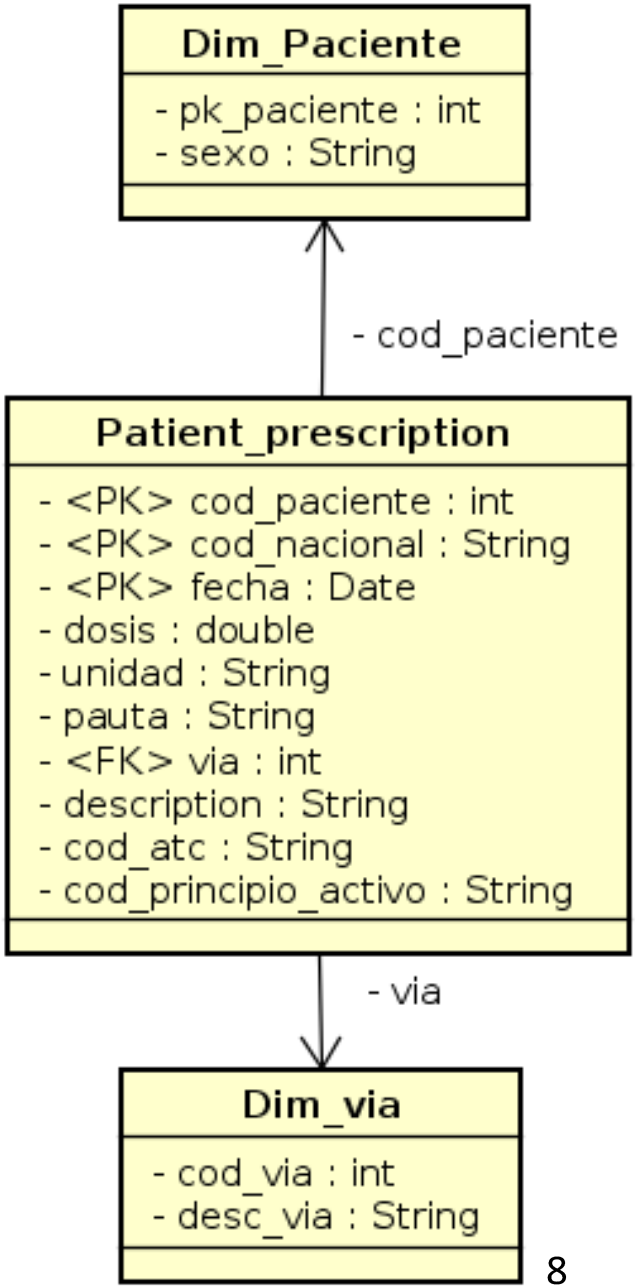


	padre integer	hijo integer	profundidad integer
1	16	1	4
2	32	1	5
3	64	1	6
4	128	1	7
5	256	1	8
6	512	1	9
7	1	1	0
8	2	1	1
9	4	1	2
10	8	1	3
11	16	2	3
12	32	2	4

2- Load dimension

UNIVERSIDAD DE MURCIA

- Steps
  - Configure global variables
  - Load XML VIAS – Xml
    - Read path from global variable
    - Read XML
    - Update dimension



2 – Use global variables from config files

UNIVERSIDAD DE MURCIA

- Files:
  - $HOME/.kettle/shared.xml
  - $HOME/.kettle/kettle.properties
    - Set pathPrescription=MIRUTA1
      - Unzip there the file given
    - Set rutaErrors=MIRUTA2
- Note1: without blank around "=".
- Note2: set simple path (C:\folder).

Terminal

```
mcampos@periquito:~/.kettle$ ls
db.cache                db.cache-6.0.0.0-353   shared.xml.backup
db.cache-5.2.0.0        kettle.properties      xulSettings.properties
db.cache-5.3.0.0-213    shared.xml
mcampos@periquito:~/.kettle$
```

```
mcampos@periquito:~/.kettle$ cat shared.xml
<?xml version="1.0" encoding="UTF-8"?>
<sharedobjects>
  <connection>
    <name>IN_LOCAL</name>
    <server>localhost</server>
    <type>POSTGRESQL</type>
    <access>Native</access>
    <database>jjj</database>
    <port>5433</port>
    <username>inbd</username>
    <password>Encrypted 2be98afc86aa7f2e4cb79ce10d79cadde<
    <servername/>
    <data_tablespace/>
    <index_tablespace/>
    <attributes>
      <attribute><code>FORCE_IDENTIFIERS_TO_LOWERCASE</cod
      <attribute><code>FORCE_IDENTIFIERS_TO_UPPERCASE</cod
      IS_CLUSTERED</code><attribute>N</at
      PORT_NUMBER</code><attribute>5433</
      PRESERVE_RESERVED_WORD_CASE</code><
      QUOTE_ALL_FIELDS</code><attribute>N
      SUPPORTS_BOOLEAN_DATA_TYPE</code><a
      SUPPORTS_TIMESTAMP_DATA_TYPE</code>
      USE_POOLING</code><attribute>N</att
```

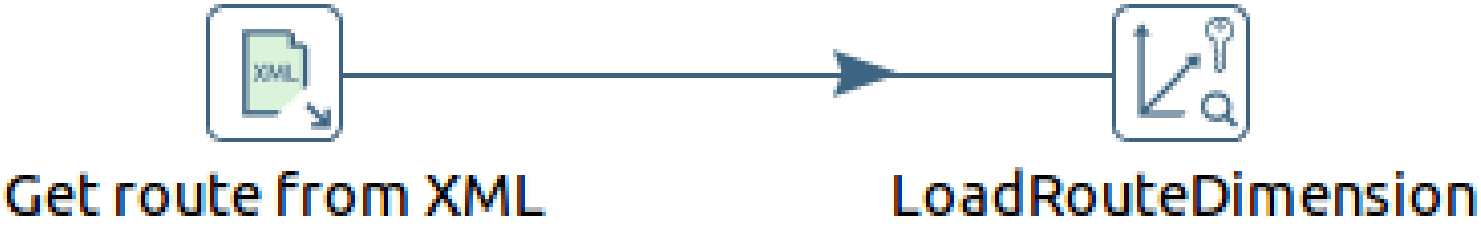
kettle.properties + (~/.kettle) - VIM

```
1 pathPrescription=/home/mcampos/tutorials/t4-paths/prescripcion_201407
2 pathErrors=/home/mcampos/tutorials/t4-paths/errors
~
-- INSERTAR --                                   1,36          Todo
```

2- Read XML and Update dimension

UNIVERSIDAD DE MURCIA

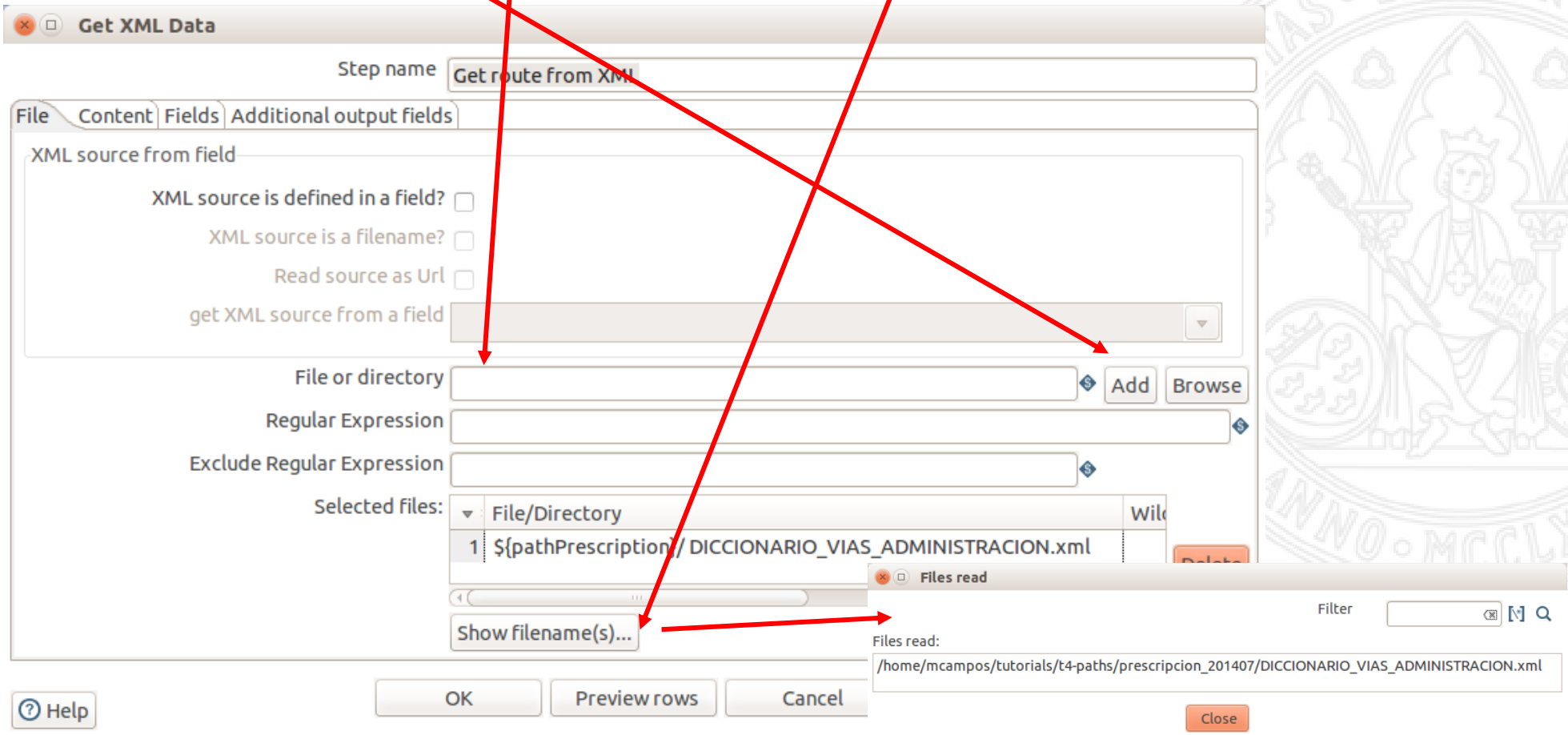
- Transformation
  - Input -> Get data from XML
    - Read content with XPath expressions
  - Datawarehouse -> Combination lookup/update
    - For SCD use Dimension lookup/update



2- Read XML file

UNIVERSIDAD DE MURCIA

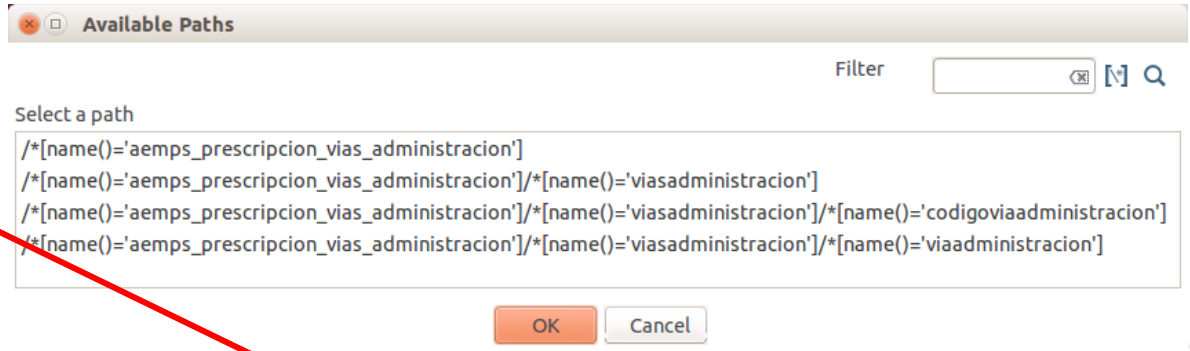
- Read file using global variable previously defined
  - ${pathPrescription}/DICCIONARIO_VIAS_ADMINISTRACION.xml
  - NOTE: Restart PDI to read global variables
- Click on “Add”. Check with “Show filenames” that file is ok.

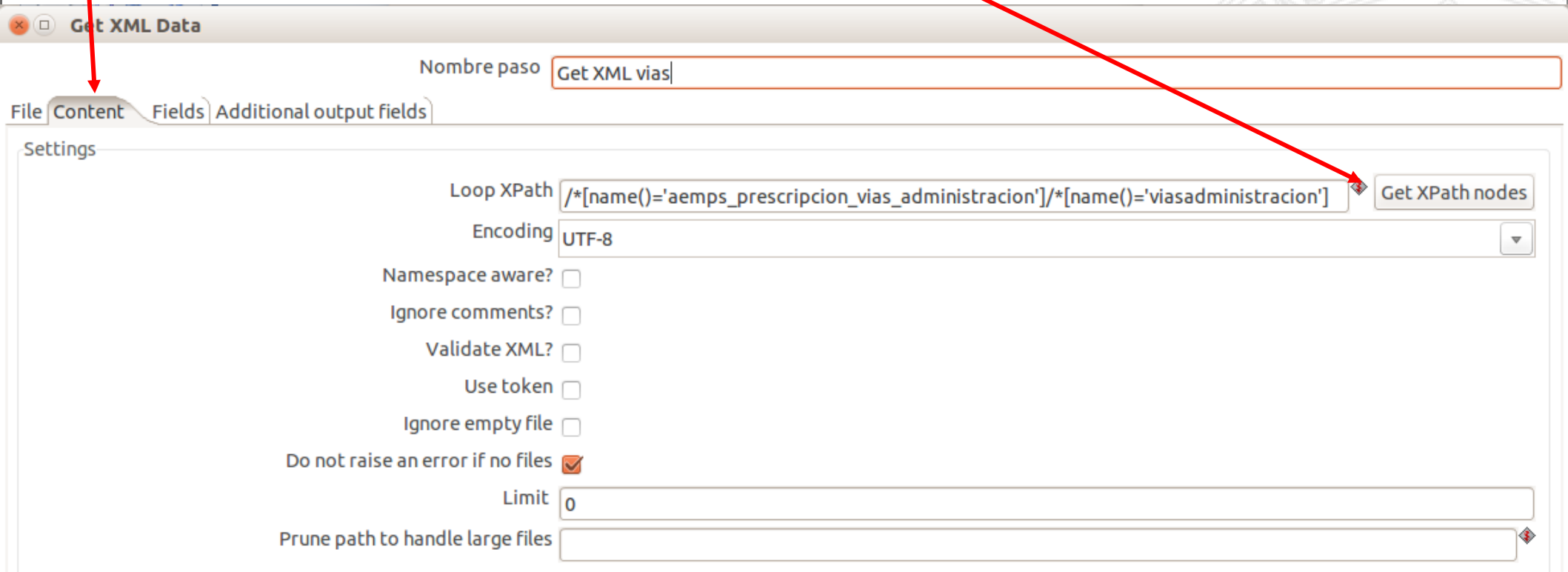


2- Read XML file

UNIVERSIDAD DE MURCIA

- Content tab
  - Click “Get XPath Nodes”
  - Select Xpath to “viasadministracion’-> We read its leaves





2- Read XML file

UNIVERSIDAD DE MURCIA

- Fields tab
  - Read fields
  - Preview rows

Get XML Data

Step name: Get route from XML

File | Content | Fields | Additional output fields

	Name	XPath	Element	Result type	Type
1	codigoviaadministracion	*[name()='codigoviaadministracion']	Node	Value of	Integer
2	viaadministracion	*[name()='viaadministracion']	Node	Value of	String

Get fields

OK | Preview rows | Cancel

Examine preview data

Rows of step: Get route from XML (50 rows)

	codigoviaadministracior	viaadministracion
1	7	HEMODIÁLISIS
2	10	OTRA VÍA
3	15	USO CUTÁNEO
4	16	USO DENTAL
5	19	USO GINGIVAL
6	21	USO BUCOFARÍNGEO
7	22	VÍA ENDOCERVICAL
8	24	VÍA ENDOTRAQUEOPULN
9	25	VÍA EPIDURAL
10	29	VÍA INTRAARTERIAL
11	30	VÍA INTRAARTICULAR
12	31	VÍA INTRACARDIACA
13	32	VÍA INTRACAVERNOSA
14	35	VÍA INTRADÉRMICA
15	38	VÍA INTRALESIONAL
16	41	VÍA INTRAMUSCULAR
17	42	VÍA INTRAOCULAR

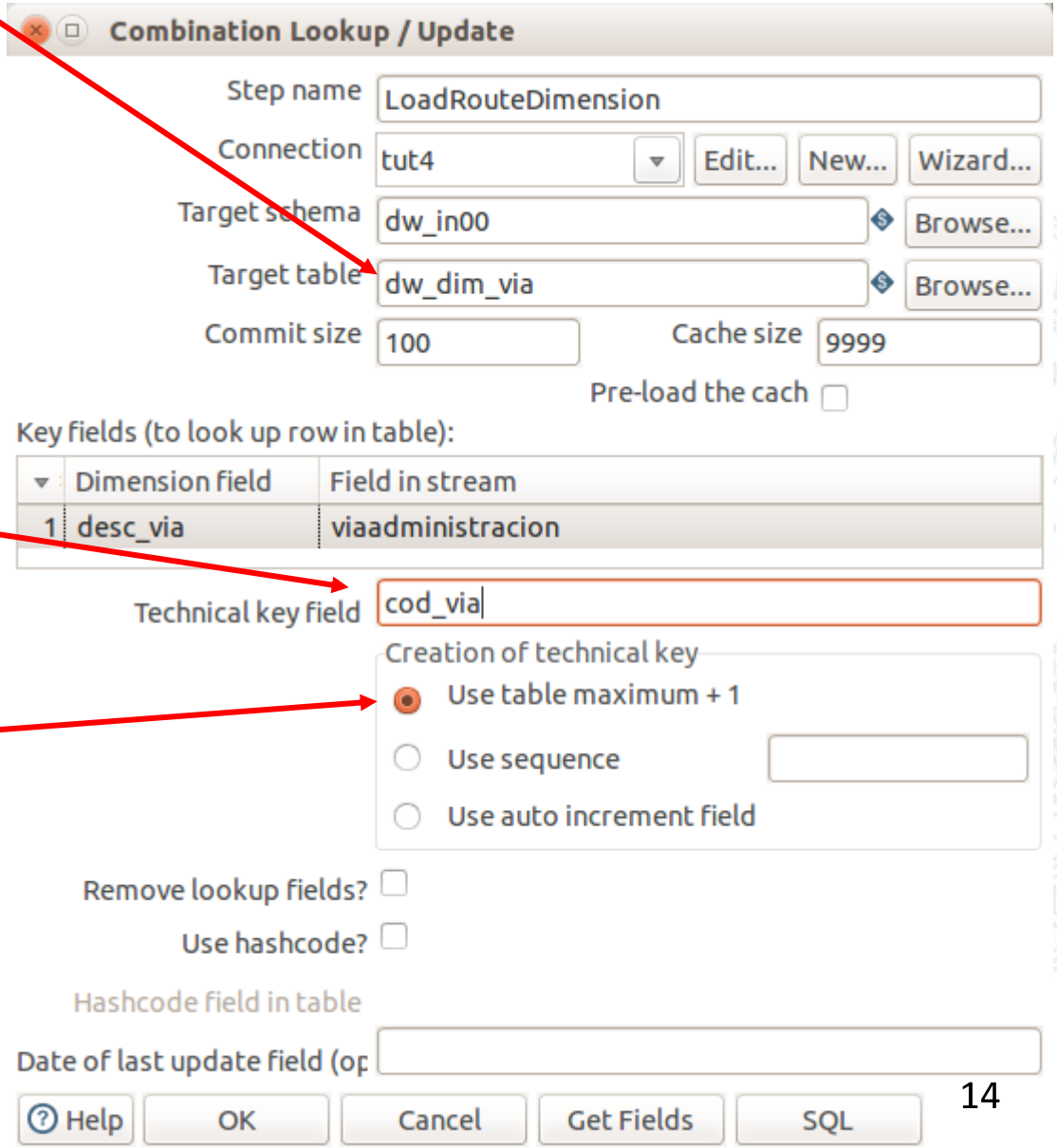
Close | Show Log

2- Update dimension

UNIVERSIDAD DE MURCIA

- Set target table: dw_YOURSCHEMA.dw_dim_via

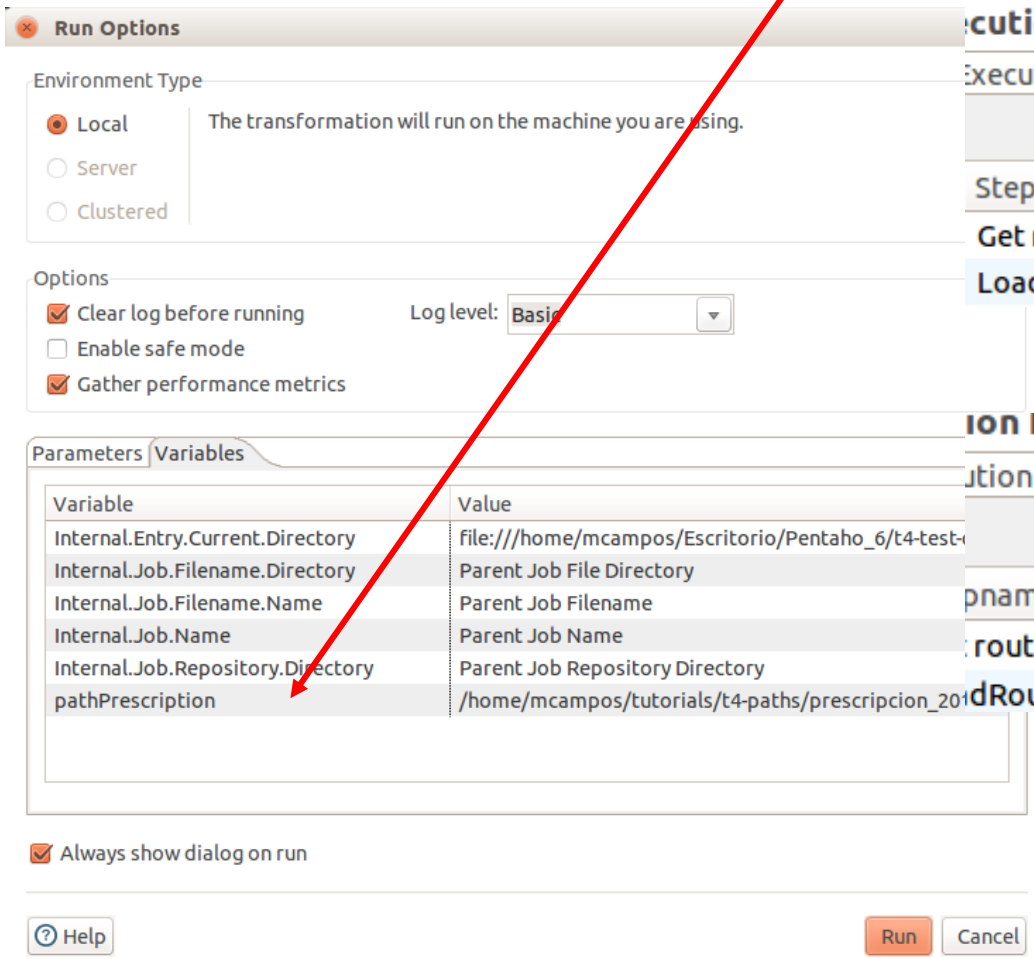
- Set key fields
- Set table column for subrogate key: cod_via
- Create subrogate key
  - As max table+1
- It will insert a new row when necessary

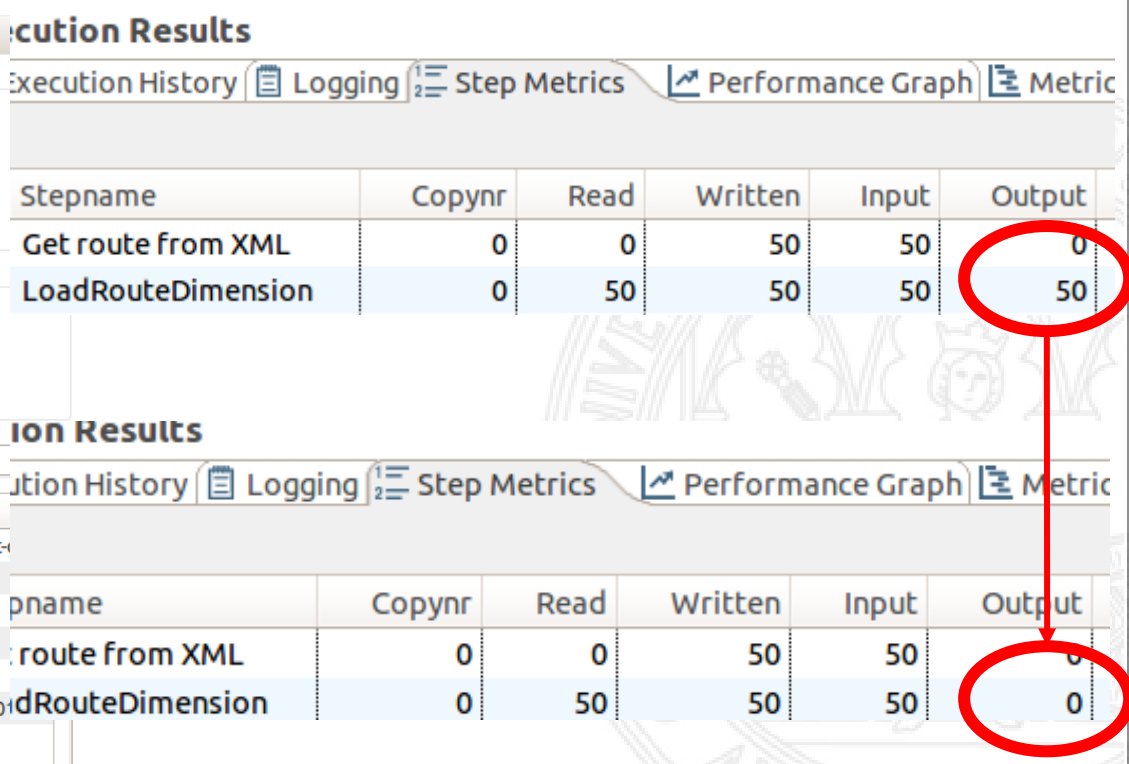


2 – Load Dimension

UNIVERSIDAD DE MURCIA

- When running Check variables
- Now rerunning doesn't try to overwrite values of dimensions





3 – Load prescriptions

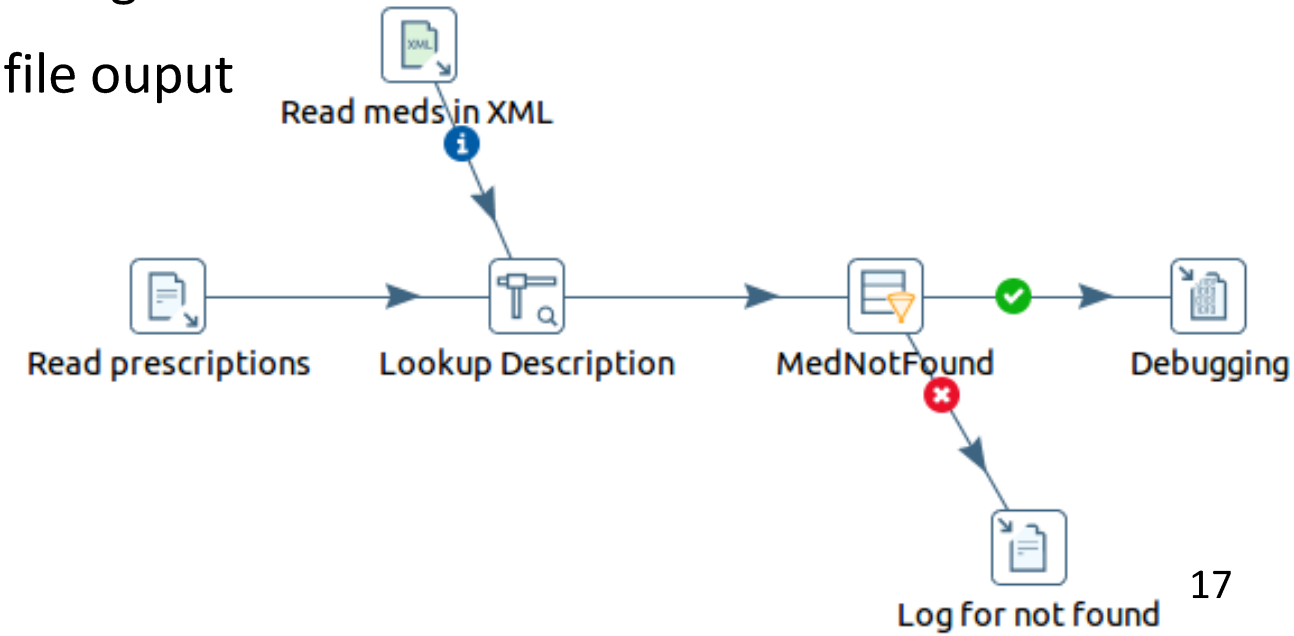
UNIVERSIDAD DE MURCIA

- Part 1
  - Read csv with prescriptions for patients. With fields
    - Active ingredient, dosis, route, etc.
  - Read XML of meds to get description of active ingredient
    - Stream lookup
- Part 2
  - Validate hour format with javascript + regular expression
    - Set constant value if invalid
  - Lookup route in dimension
  - Merge date and time in a "timestamp" field
    - Include metadata
  - Store in database

3 - Load prescriptions

UNIVERSIDAD DE MURCIA

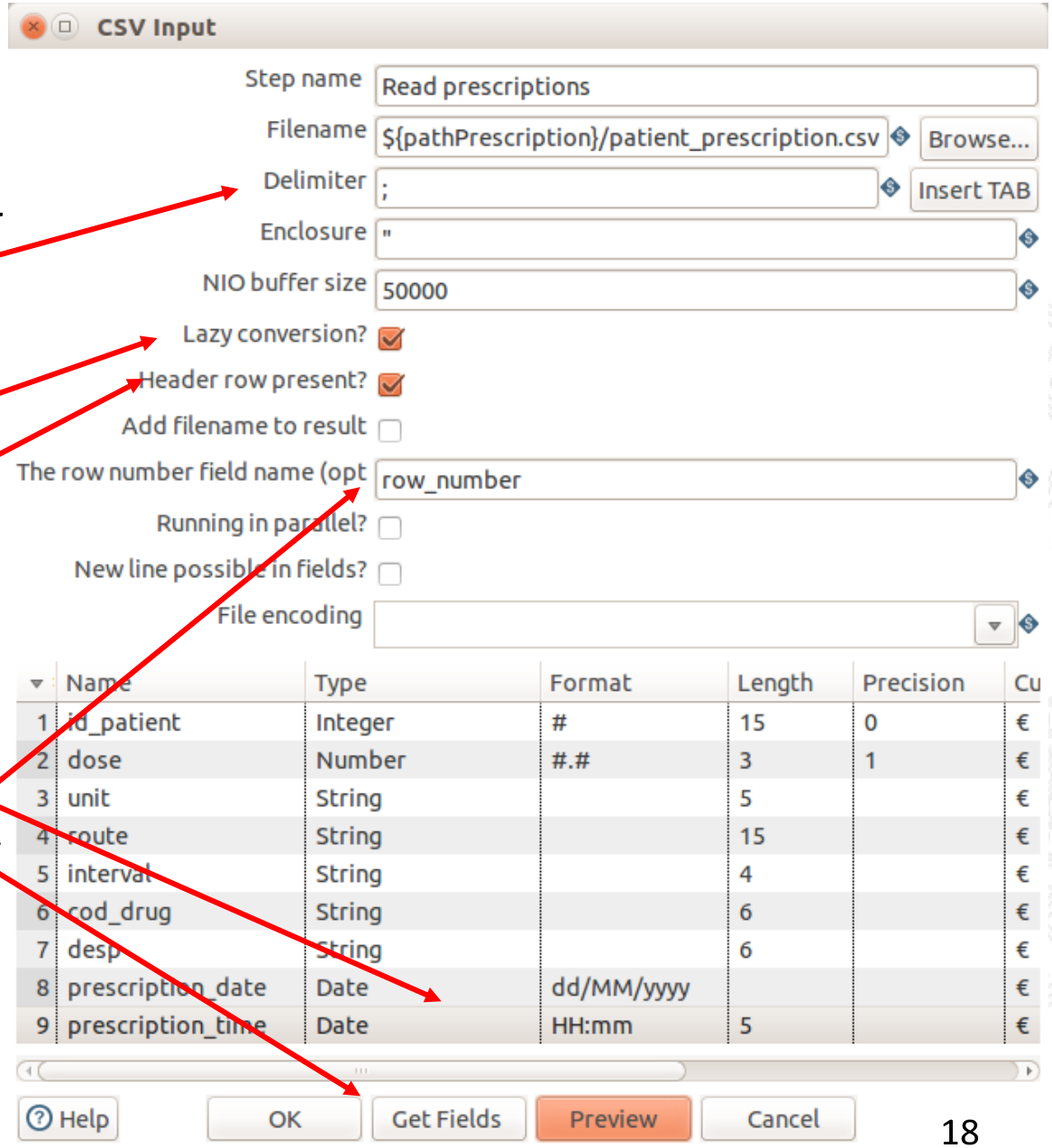
- Patient prescriptions: Input -> CSV file input
- Meds description: Input ->Get data from XML
- Lookup description of active ingredient: Lookup -> Stream lookup
- Filter prescription without description (not found in stream):
  - Flow -> Filter rows
  - If true: Utility -> Write to log
  - If false: Output -> Text file ouput



3- Read CSV file

UNIVERSIDAD DE MURCIA

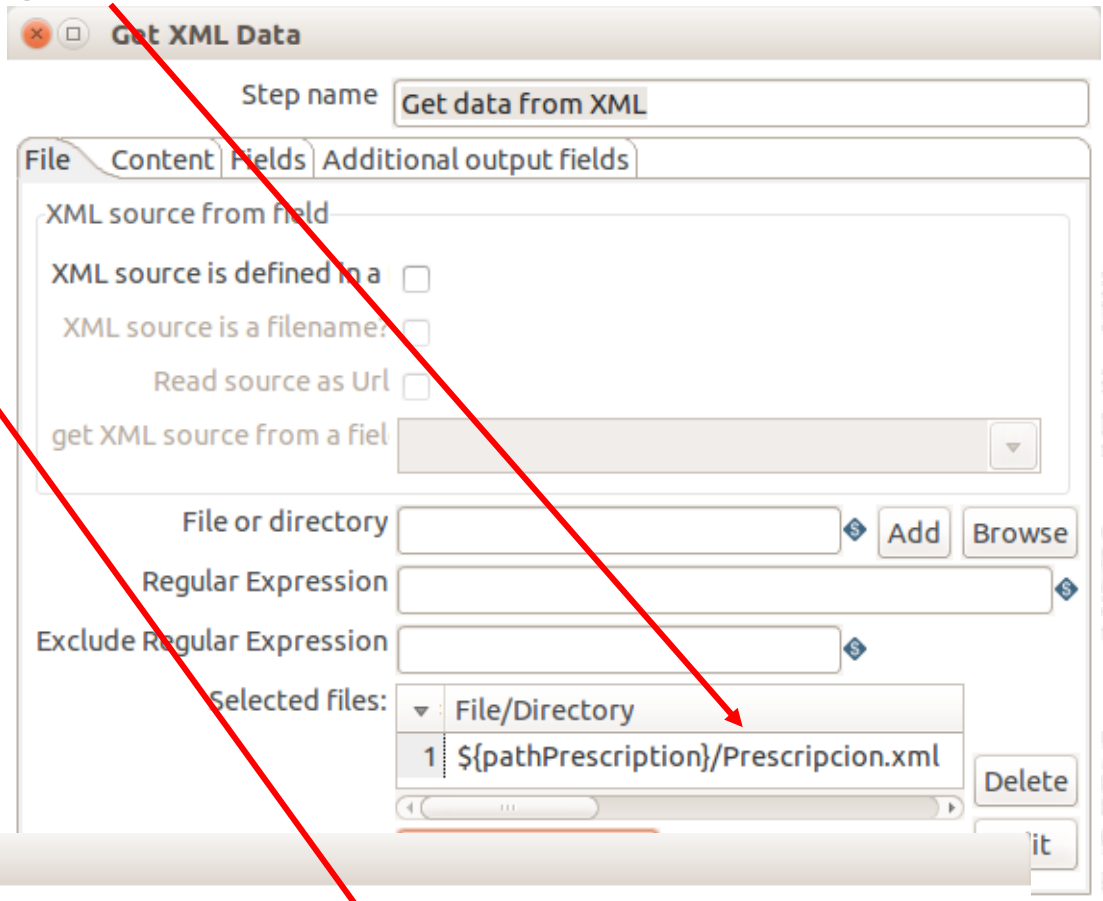
- Set filename using global var
  - ${pathPrescription}/patient_prescription.csv
- Set delimiter “;”
- Enable lazy conversion
- Check header present
- Click “Get fields”
- Important: data format
- Include new column with row number for debugging: “row_number”

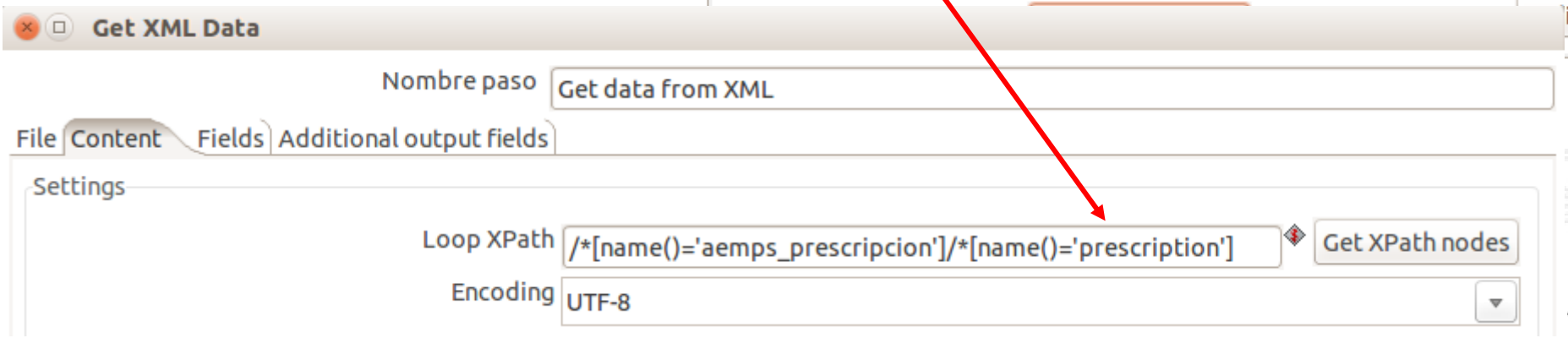


3- Read XML file

UNIVERSIDAD DE MURCIA

- Same as before but “Prescription.xml”
- XPath for ‘prescription’

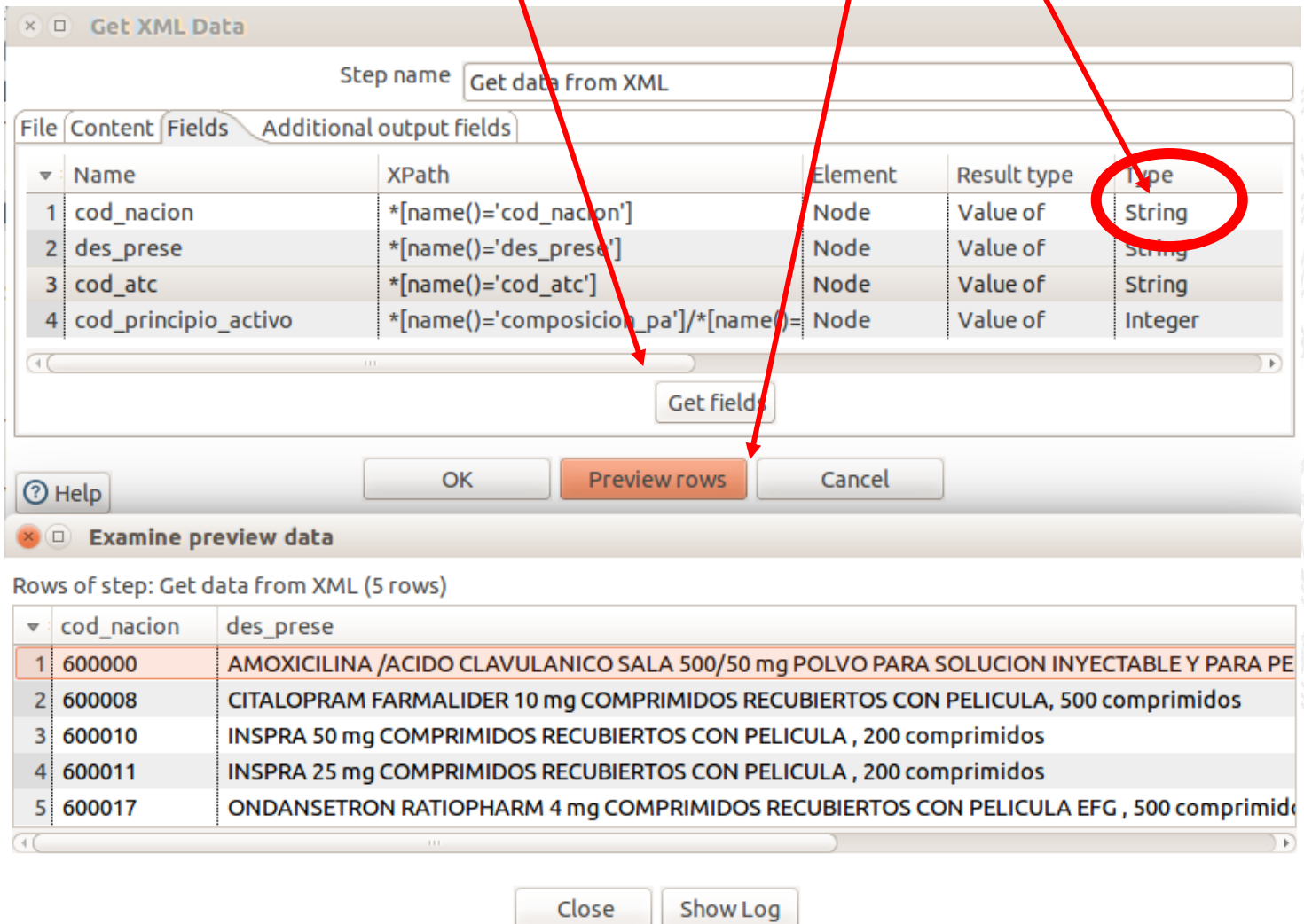




3- Read XML file

UNIVERSIDAD DE MURCIA

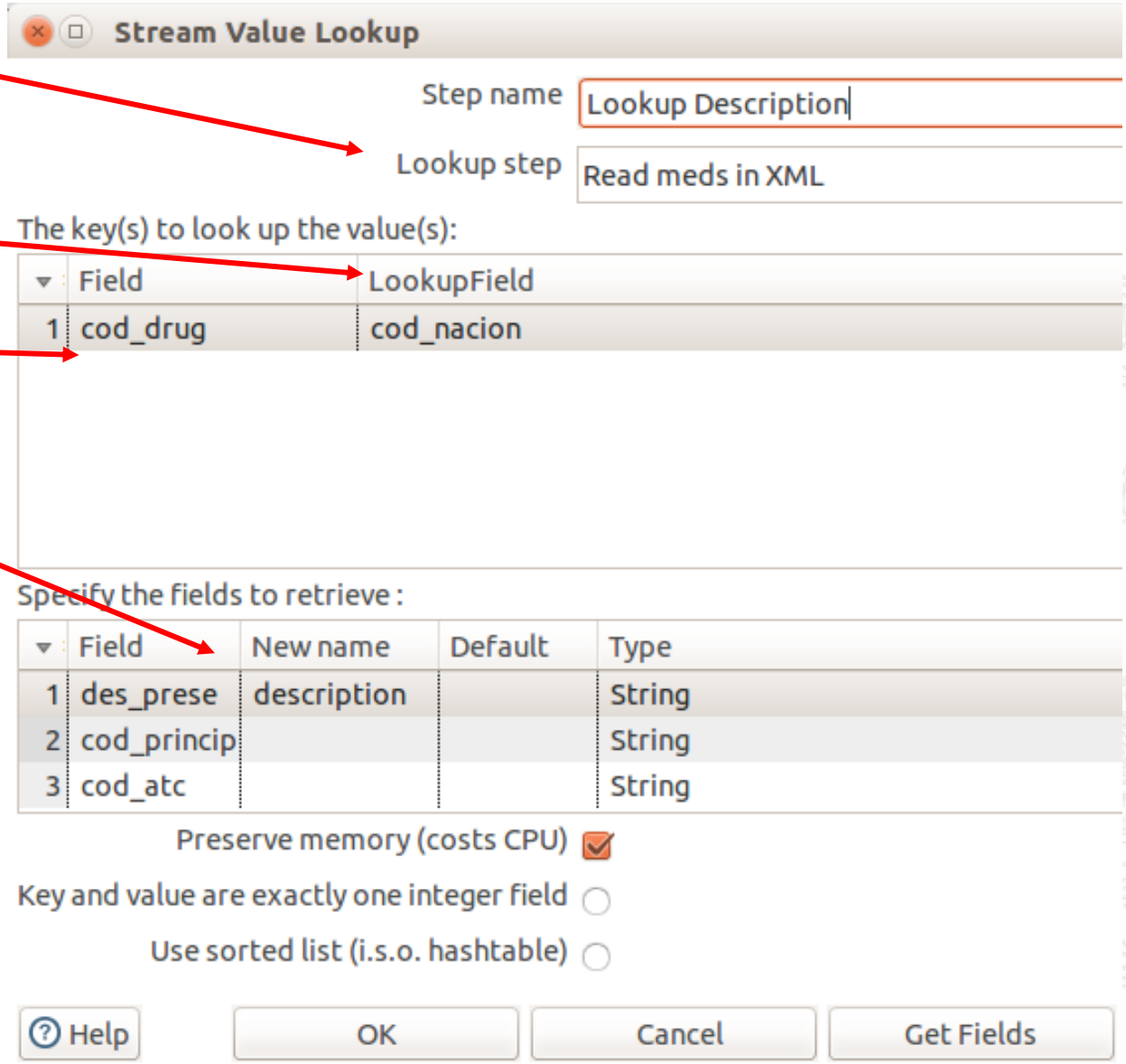
- Tab “Fields” -> Get fields and preview rows.
  - Get only ‘cod_nacion’, ‘des_prese’, ‘cod_atc’, ‘cod_principio_activo’
  - Note: Codigo_nacional change data type: String



3- Lookup in stream

UNIVERSIDAD DE MURCIA

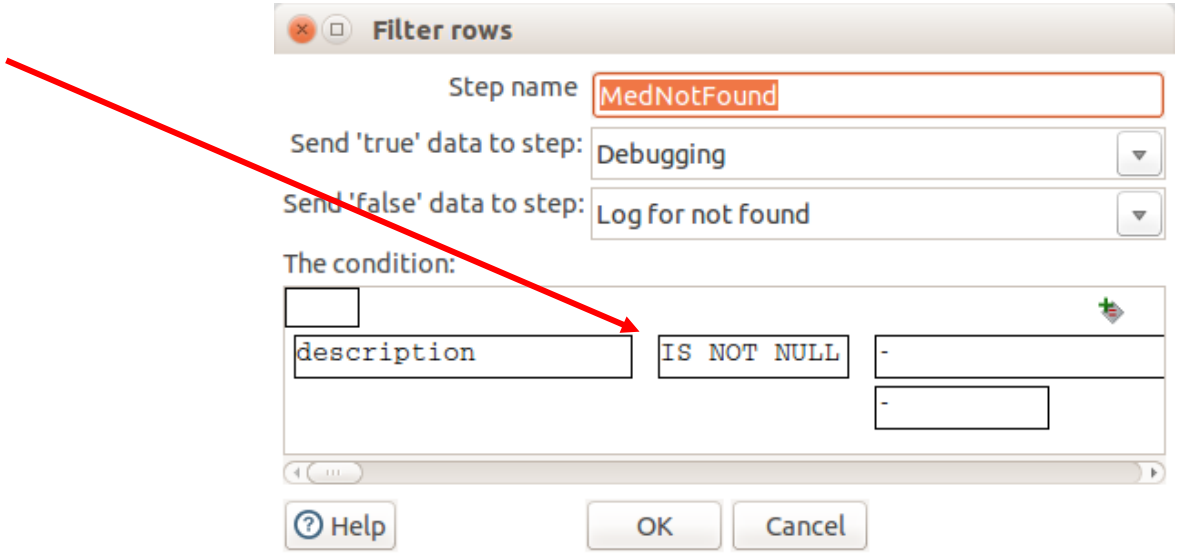
- Set stream to lookup in
  - The XML file
- Key field in XML
- Field in CSV
- Fields from XML that will be inserted in the stream
  - Set data type
  - Optional renaming



3- Filter

UNIVERSIDAD DE MURCIA

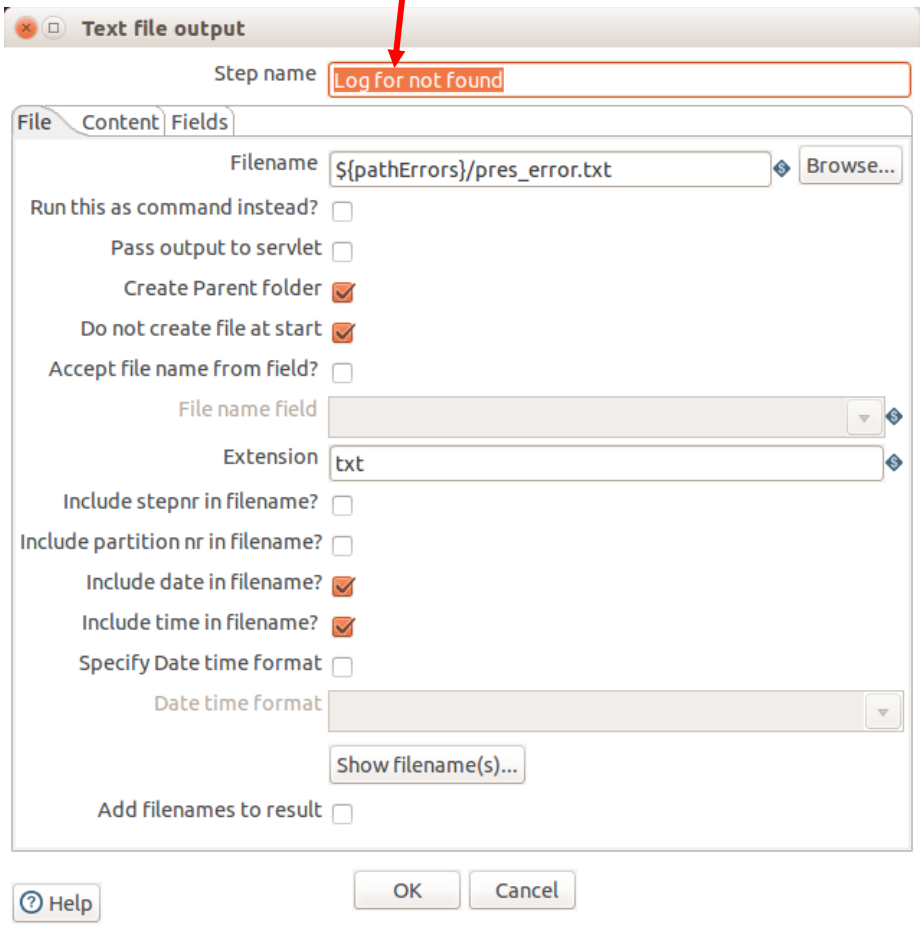
- Discard if description is null



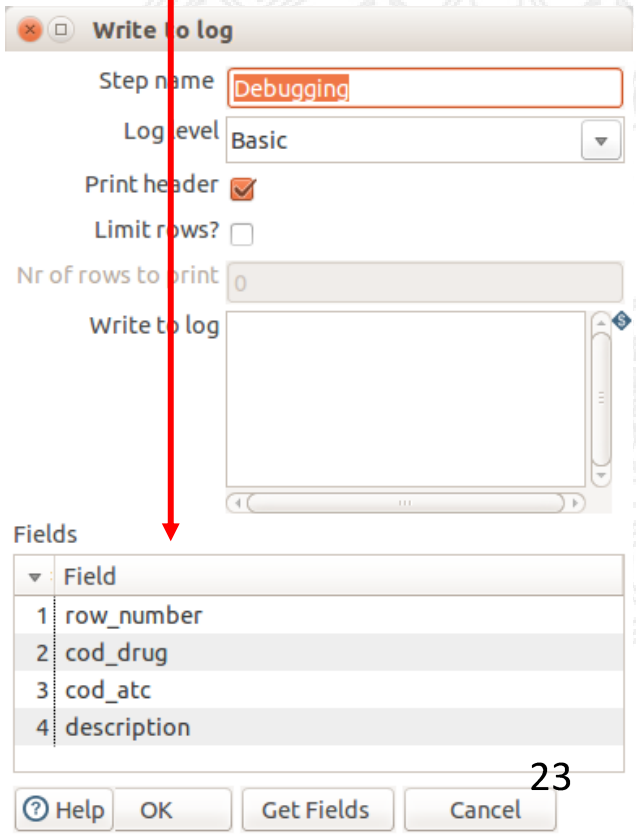
3- Write to log and errors to file

UNIVERSIDAD DE MURCIA

- Set a text file to register errors
  - Set folder ${pathErrors}/pres_error.txt



- Set fields needed for debugging
  - Row number is usefull



Stop and verify everything is ok

UNIVERSIDAD DE MURCIA

- Execute the transformation
  - Get data from xml: 19994 rows
  - Text file: 10 rows

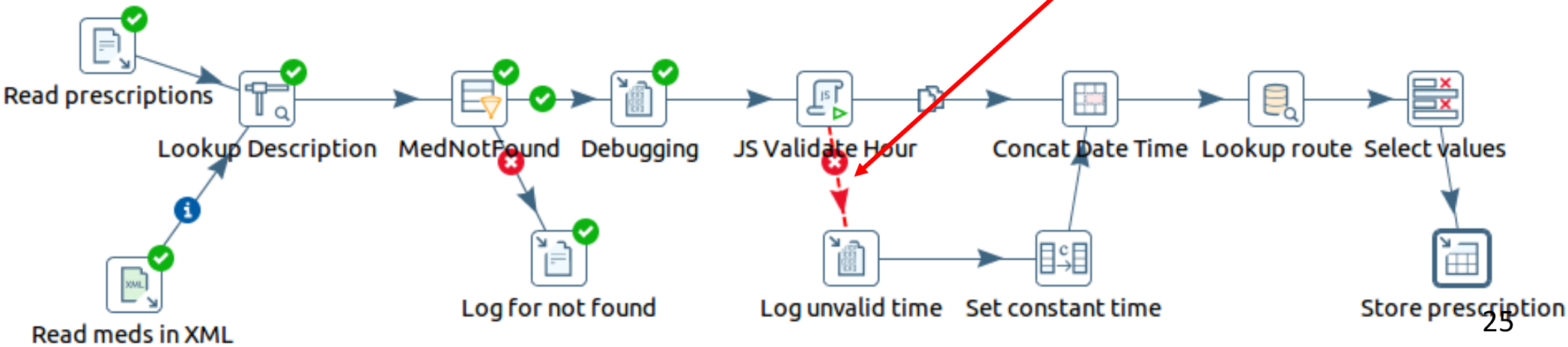
- Some error in log writing?
  - Modify “Read prescription” and change data type of column “prescription_time” to string

3- Load prescriptions - Part 2

UNIVERSIDAD DE MURCIA

- Scripting -> Modified Javascript Value
  - Validate hour format with javascript + regular expression
  - Transform -> Set field value to a constant: Set constant value if invalid
- Transform->Concat fields: Concat date and time in a "timestamp"
- Lookup->Database lookup: lookup route in route dimension
- Transform->Select fields: Remove unneeded fields. Set type, rename.
- Output->Table output: Store in database

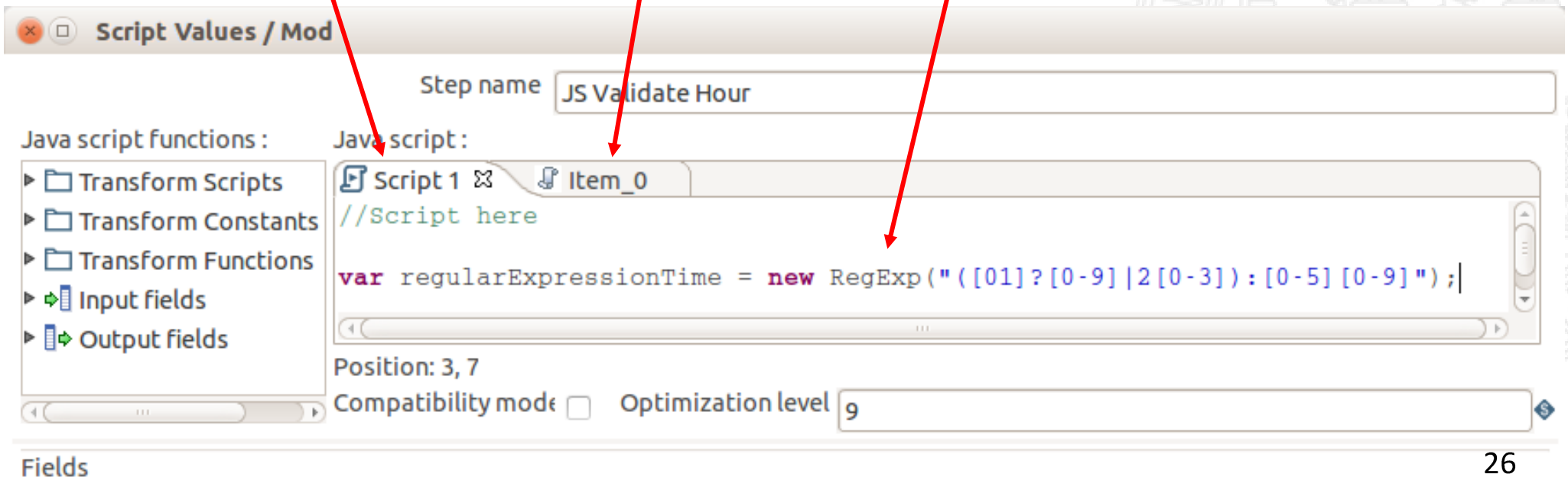
Note: error handling



3 – JS Validate time

UNIVERSIDAD DE MURCIA

- Insert new start script -> **Right click on Tab "Script 1"**
  - "Set start script" -> Executed once for the transformation
    - Initialize regular expression: **RegExp("([01]?[0-9]|2[0-3]):[0-5][0-9]")**
  - "Add New" -> "Set transform script" -> Once per row
    - Verify regular expression
  - Alternative: Validation->DataValidator



3 – JS Validate time

UNIVERSIDAD DE MURCIA

- Validate regular expression with value
  - regularExpressionTime.test(prescription_time);
  - Get variables from “Input fields”
- Launch error in transformation, also for debugging

Script Values / Mod

Step name: JS Validate Hour

Java script functions:

- Transform Functions
- Input fields
  - id_patient
  - dose
  - unit
  - route
  - interval
  - cod_drug
  - desp

Java script:

Script 1 | Item_0

```
correctTime = regularExpressionTime.test(prescription_time);

if (!correctTime) {
    Alert("Unvalid time:" + prescription_time); // Debug
    _step_.putError(getInputRowMeta(), row, e.nr_errors, e.message, e.field, e.errcode);
    trans_Status = ERROR_TRANSOFRMATION; // Raise error for manager
}
```

Position: 6, 67

Compatibility mode? ☐ Optimization level 9

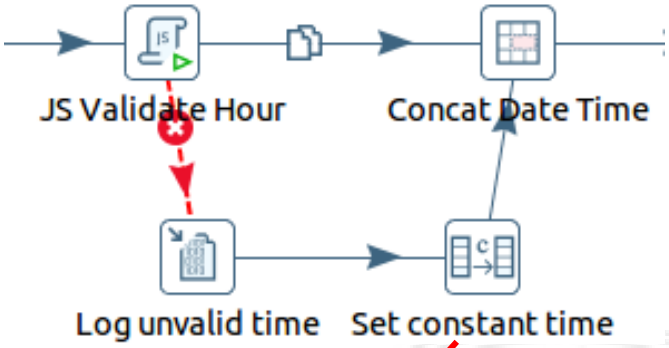
Fields

Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'

3 – JS Validate time

UNIVERSIDAD DE MURCIA

- If invalid time log
- Set constant value in field “prescription_time”



Write to log

Step name: Log unvalid time
Log level: Basic
Print header: ☑
Limit rows?: ☐
Nr of rows to print: 0
Write to log:

Fields

	Field
1	id_patient
2	row_number
3	cod_drug
4	prescription_date
5	prescription_time

Help | OK | Get Fields | Cancel

Set field value to a constant

Step name: Set constant time
Use variable in constant: ☐

Fields

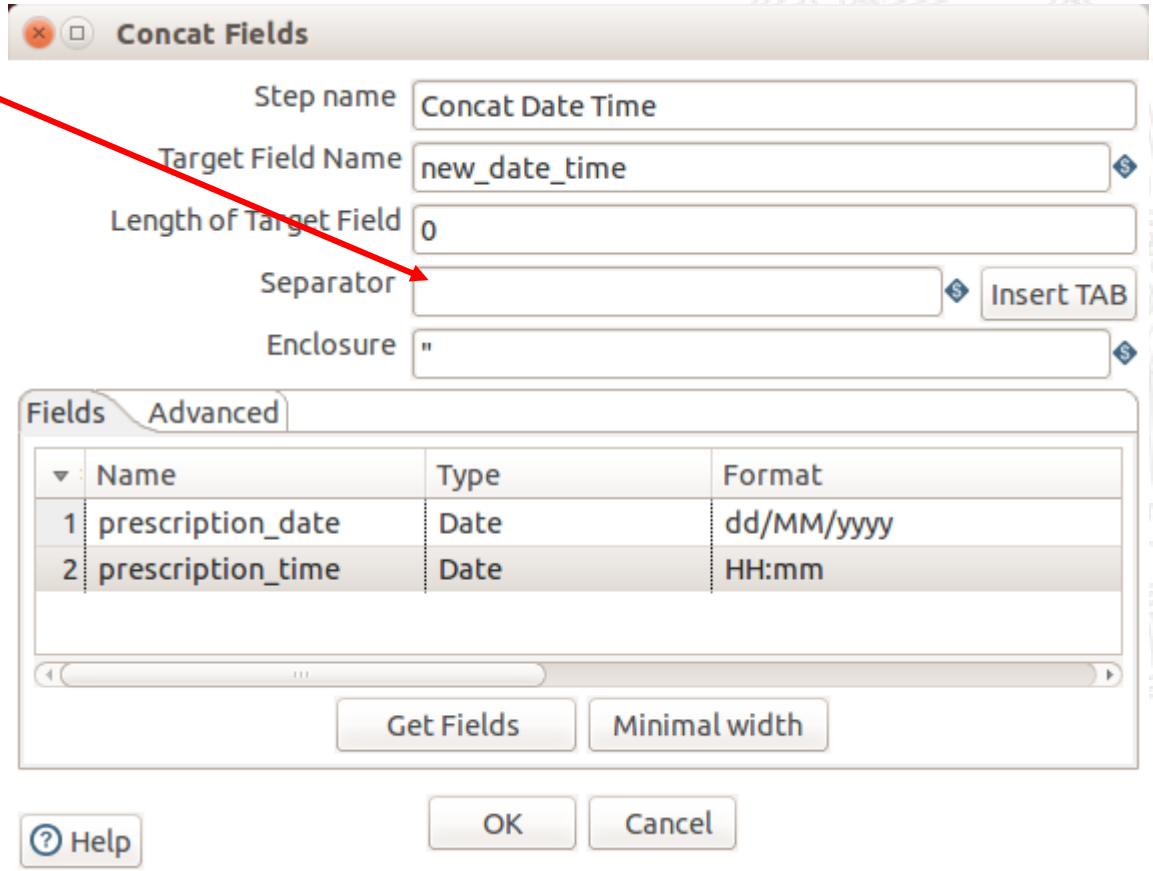
	Field	Replace by value	Conversion mask (Date)
1	prescription_time	08:00	HH:mm

Help | OK | Get Fields | Cancel

3 – Concat time and date

UNIVERSIDAD DE MURCIA

- Concat fields “prescription_date” and “prescription_time” in new stream field: “new_date_time”
- Check data types
- Note Separator: blank space (not empty field)



3 – Lookup route

UNIVERSIDAD DE MURCIA

- Select your_SCHEMA.dw_dim_via
  - Set lookup field: “route”
  - Set table field: “desc_via
  - Return “cod_via”
    - Include type

Database Value Lookup

Step name: Lookup route

Connection: tut4 | Edit... | New... | Wizard...

Lookup schema: dw_in00 | Browse...

Lookup table: dw_dim_via | Browse...

Enable cache?

Cache size in rows (0=cache everything): 0

Load all data from table

The key(s) to look up the value(s):

	Table field	Comparator	Field1	Field2
1	desc_via	=	route	

Values to return from the lookup table :

	Field	New name	Default	Type
1	cod_via			Integer

Do not pass the row if the lookup fails

Fail on multiple results?

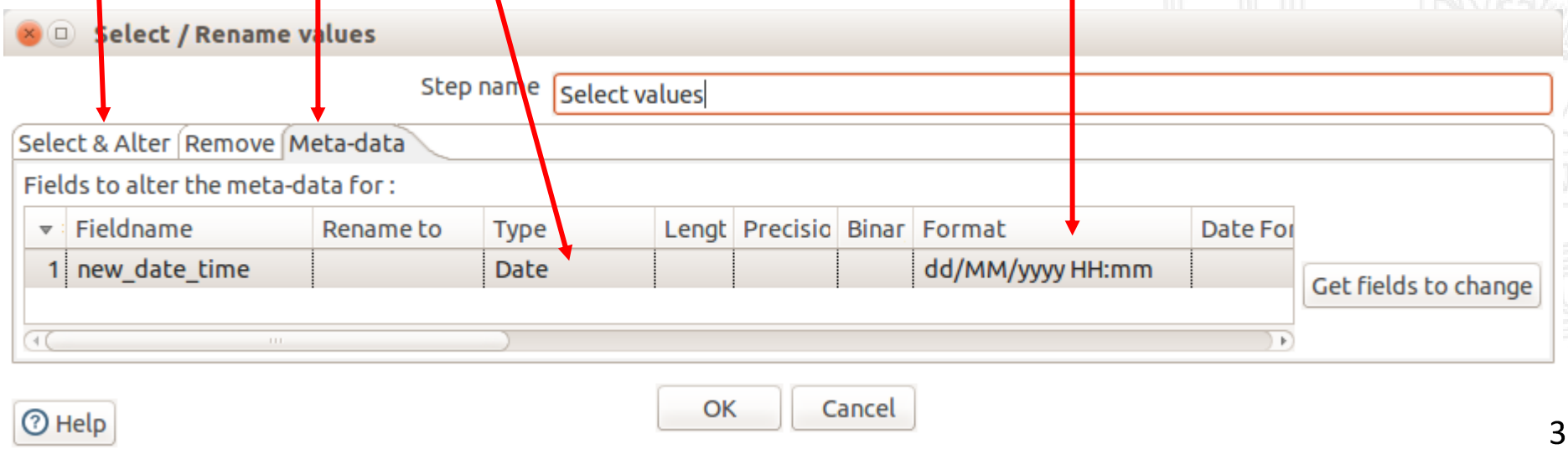
Order by

Help | OK | Cancel | Get Fields | Get lookup fields

3 – Select data and introduce metadata

UNIVERSIDAD DE MURCIA

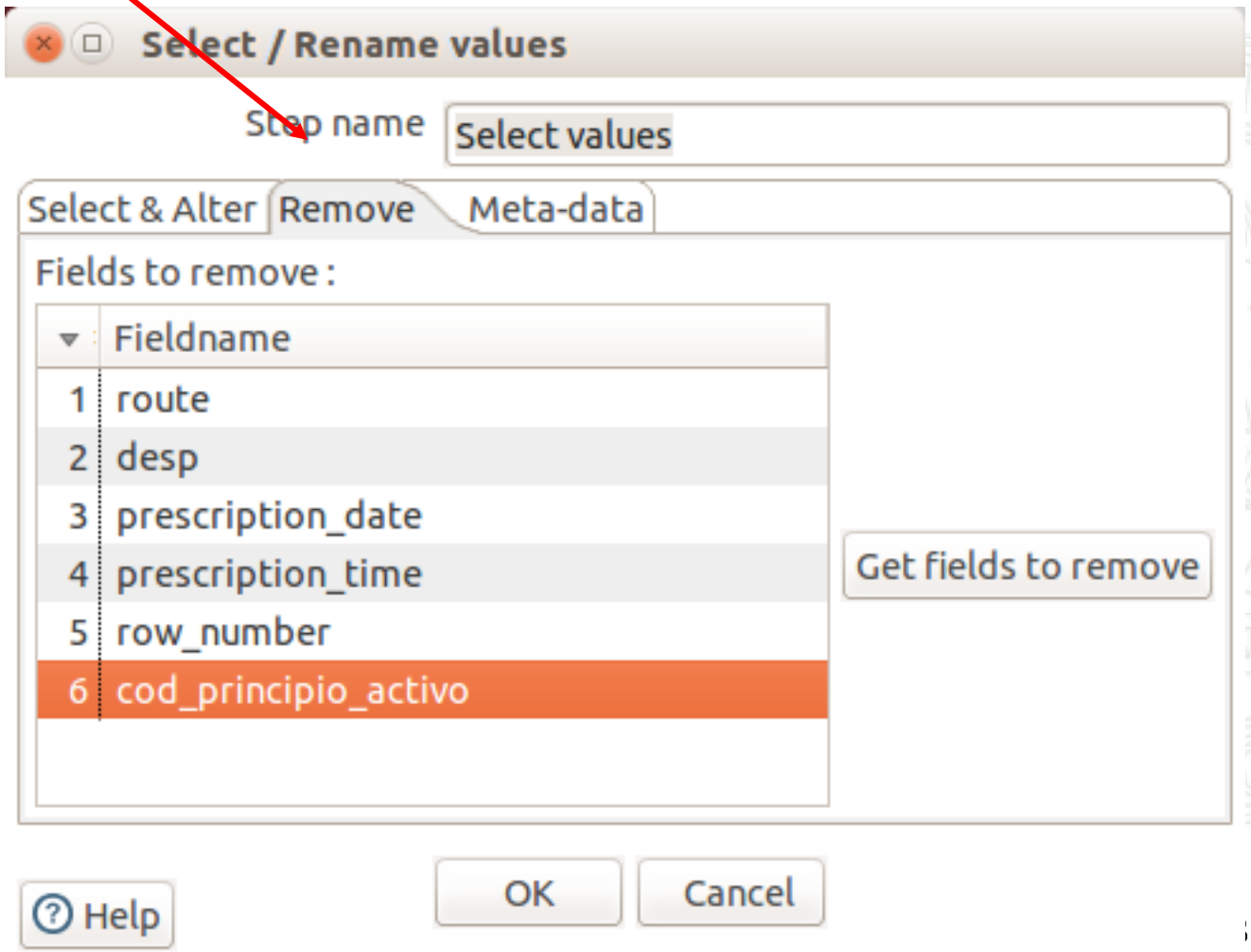
- Tab “Select & alter -> Get fields: all
- Tab “Meta-data:”
- Set “Timestamp” type and format “dd/MM/yyyy HH:mm” for field “new_date_time” we concat



3 – Select data and introduce metadata

UNIVERSIDAD DE MURCIA

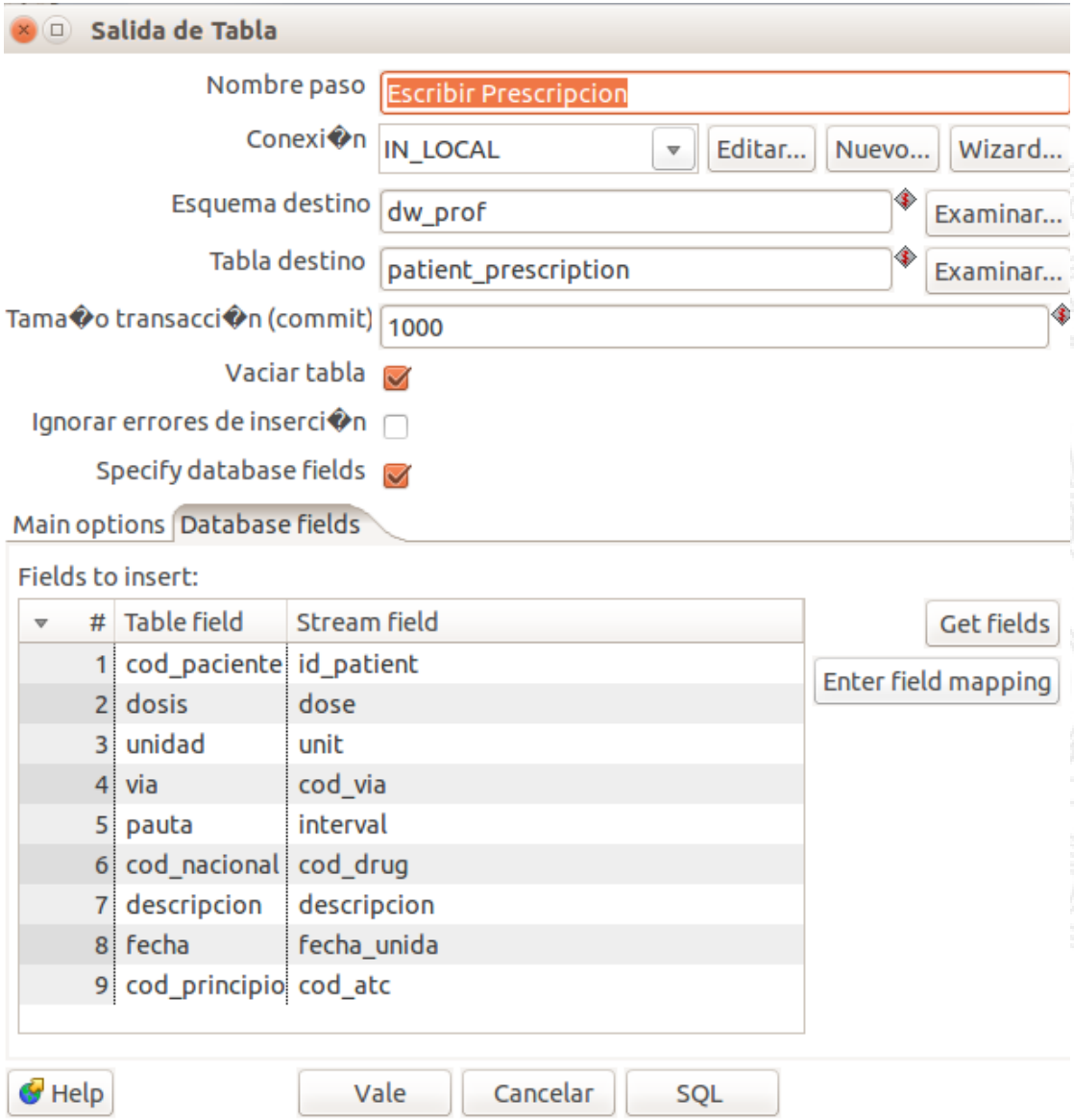
- Tab “Remove”
- Include only unneeded fields:



3- Store prescriptions

UNIVERSIDAD DE MURCIA

- Select your _SCHEMA.patient_prescription



Ready

UNIVERSIDAD DE MURCIA

- Remember: Before running, verify transformation
  - In case of problem with Date format parsing one of:
    - A) In input disable “lazy load” with type string
    - B) Set prescription_time with type Date format HH:mm, and not print in in log steps