

**Máster universitario de Ciencia de Datos**

**Práctica 2**

**Diseño y uso de bases de datos analíticas – identificación, diseño y desarrollo de los procesos ETL.**

Autor:

Mario Ubierna San Mamés

|  |
| --- |
| Índice de Contenido |

[Índice de Contenido 3](#_Toc71554266)

[Índice de tablas 4](#_Toc71554267)

[Índice de ilustraciones 5](#_Toc71554268)

[1. Introducción 6](#_Toc71554269)

[2. Bibliografía 31](#_Toc71554270)

|  |
| --- |
| Índice de tablas |

**No se encuentran elementos de tabla de ilustraciones.**

|  |
| --- |
| Índice de ilustraciones |

**No se encuentran elementos de tabla de ilustraciones.**

|  |
| --- |
| Introducción |

## Presentación

A partir de la solución oficial de la primera práctica (PRA1), el estudiante debe diseñar, implementar y ejecutar los procesos de extracción, transformación y carga de los datos de las fuentes de datos proporcionadas.

Así pues, esta actividad tiene como objetivo identificar y desarrollar los procesos de carga del almacén de datos y que esta sea efectiva.

## Descripción

Si nos centramos en los subobjetivos, esta segunda parte del caso práctico consiste en lo siguiente:

* Identificar los procesos de extracción, transformación y carga de datos (ETL) hacia el almacén de datos.
* Diseñar y desarrollar los procesos ETL mediante las herramientas de diseño proporcionadas.
* Implementar con los trabajos (*jobs*) los procesos ETL para que su carga planificada sea efectiva.

Además del documento con la solución de la PRA2 que se debe entregar, también se tendrá en consideración la implementación sobre la máquina virtual proporcionada en el curso.

En resumen, el documento de la solución de la PRA2 debe incluir los siguientes aspectos:

* Descripción de todas las acciones que se han realizado.
* Capturas de pantalla que muestren todas las partes significativas del ETL, sus características y su correspondiente explicación.
* Capturas de pantalla que demuestren la correcta ejecución de la ETL y el tiempo de ejecución.
* Capturas de pantalla que demuestren las correcta carga de los datos (cargados en la base de datos).

|  |
| --- |
| Identificación de los procesos ETL |

A la hora de diseñar los procesos de carga de una base de datos analítica no hay una única estrategia. Es habitual estructurar los procesos ETL sobre la base de las entidades de datos que se deben actualizar, ya que existen diferencias conceptuales en la actualización de una dimensión con respecto a la de una tabla de hechos. La división del proceso de carga inicial en diferentes bloques de actualización facilitará el diseño de un orden de ejecución y la gestión de las dependencias. Cada uno de estos bloques de actualización se dividirá en las correspondientes etapas de extracción, transformación y carga.

Se identifican los dos bloques siguientes:

* **Bloque IN:** procesos de carga de los datos desde las fuentes a las tablas intermedias en el área de maniobras (*staging area*). Estos procesos se distinguen por el prefijo «IN\_» en el nombre.
* **Bloque TR:** procesos de transformación para cargar los datos desde las tablas intermedias hasta nuestro almacén, según el modelo multidimensional diseñado. Así pues, son diferentes los procesos ETL de transformación para cargar las dimensiones de aquellos que se realizan para cargar las tablas de hechos. Estos procesos se distinguen con el prefijo «TR\_» en el nombre.

## Bloque IN

Respecto al bloque In, el cual nos va a permitir almacenar la información en el staging area, tenemos los siguientes procesos:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre ETL** | **Descripción** | **Orígenes de los dastos** | **Tabla de destino (stage)** |
| IN\_ DENUNCIAS\_ INFRACCIONES | Carga de los datos correspondientes a las estadísticas sobre los expedientes incoados por el artículo 36.6 LOPSC de desobediencia durante el estado de emergencia sanitaria COVID-19 en la comunidad de Euskadi. | ACUMULADO-DENUNCIAS-INFRACCIONES.xlsx | STG\_Denuncias\_Infracciones |
| IN\_POBLACION | Carga los datos respectivos a las cifras de la población española. | población\_9687bsc.csv | STG\_Poblacion |
| IN\_MOVILIDAD | Movilidad de la población durante el estado de alarma. | 35167bsc.csv | STG\_Movilidad |
| IN\_AGLOMERACION | Porcentaje de la población que evitaba las aglomeraciones con motivo del coronavirus, por grupo de edad y provincia. | statistic\_id1104235\_covid19\_-poblacion-que-evitabalas-aglomeraciones-segunedad-en-espana-2020.xlsx | STG\_Evitar\_Aglomeracion |
| IN\_LLAMADAS\_112 | Llamadas al 112 por ámbito geográfico y tipología (accidentes de tráfico, civismo, incendios, asistencia sanitaria, seguridad...) | rows.xml | SGT\_Llamadas112 |

Tabla - Procesos ETL Bloque IN.

## Bloque TR

Respecto al bloque TR tenemos tanto los procesos para dotar de datos a las dimensiones como a los hechos.

### Dimensiones

Los procesos ETL que se encargar de añadir la información a la dimensiones son los siguientes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del ETL** | **Descripción** | **Tabla de origen** | **Tabla de destino (dimensión)** |
| TR\_DIM\_FECHA | Carga y transformación de la dimensión temporal. |  |  |
| TR\_DIM\_AMBITO\_GEOGRAFICO | Carga y transformación de la dimensión con los datos de los ámbitos geográficos. |  |  |
| TR\_DIM\_GRUPO\_EDAD | Carga y transformación de la dimensión con los datos de los grupo de edad. |  |  |
| TR\_DIM\_MEDICION | Carga y transformación de la dimensión con los datos de las mediciones. |  |  |
| TR\_DIM\_TIPOLOGIA | Carga y transformación de la dimensión con los datos de la tipología. |  |  |

Tabla - Procesos ETL Bloque TR Dimensiones.

### Hechos

Respecto a los hechos tenemos los siguientes procesos de carga:

|  |  |  |
| --- | --- | --- |
| **Nombre del ETL** | **Descripción** | **Tabla de origen** |
| TR\_FACT\_LLAMADAS112 | Carga y transformación de la tabla de hechos Fact\_Llamadas112. | STG\_Llamadas112 |
| TR\_FACT\_MEDICIONES | Carga y transformación de la tabla de hechos Fact\_Mediciones |  |

Tabla - Procesos ETL Bloque TR Hechos.

|  |
| --- |
| Diseño y desarrollo de los procesos ETL |

En este apartado, se deben diseñar los procesos de carga identificados en el punto anterior con la herramienta de diseño proporcionada. En este caso es Pentho Data Integration (PDI).

## Creación de tablas

El primer paso para la implementación de los procesos ETL consiste en la creación de las tablas. Esto se llevará a cabo una única vez, mediante *scripts*, sobre la base de dastos proporcionada (en nuestro caso: SQL Server). Se deberán crear las tablas intermedias y las tablas del modelo dimensional de la solución oficial, es decir, las dimensiones y las tablas de hechos. Para hacerlo, deben utilizarse los *scripts* facilitados junto a la solución de la PRA1.

### Tablas del área intermedia (*staging area*)

Lo primero que vamos a hacer es la creación de las tablas intermedias:

#### Tabla intermedia STG\_Denuncias\_Infracciones

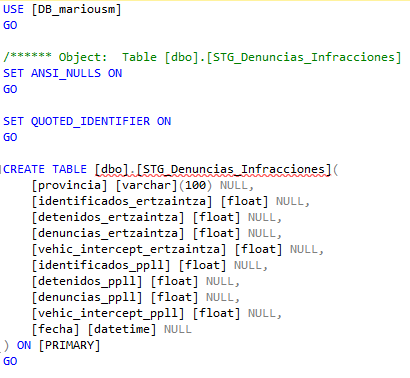


Ilustración - STG\_Denuncias\_Infracciones.

#### Tabla intermedia STG\_Poblacion

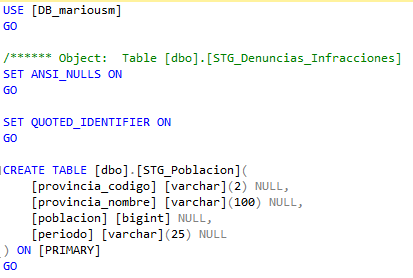


Ilustración - STG\_Poblacion.

#### Tabla intermedia STG\_Llamadas112

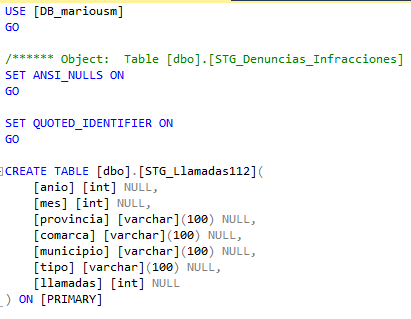


Ilustración - STG\_Llamadas112.

#### Tabla intermedia STG\_Movilidad

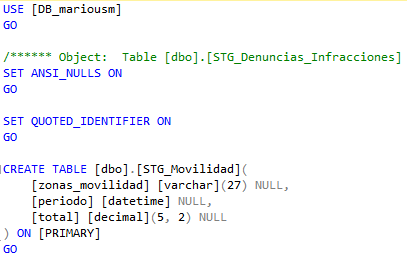


Ilustración - STG\_Movilidad.

#### Tabla intermedia STG\_Evitar\_Aglomeracion

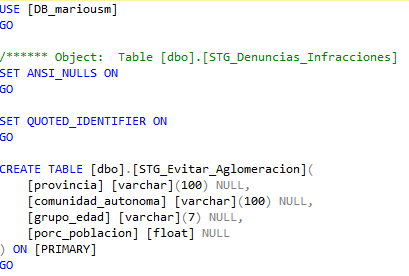


Ilustración - STG\_Evitar\_Aglomeracion.

Comprobamos que todas las tabla intermedias se han creado correctamente:

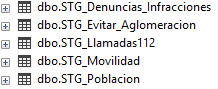


Ilustración - Tablas de staging area.

### Tablas de las dimensiones

Lo segundo que debemos de hacer es la creación de las tablas de dimensiones:

#### Tabla dimensión DIM\_Ambito\_Geografico

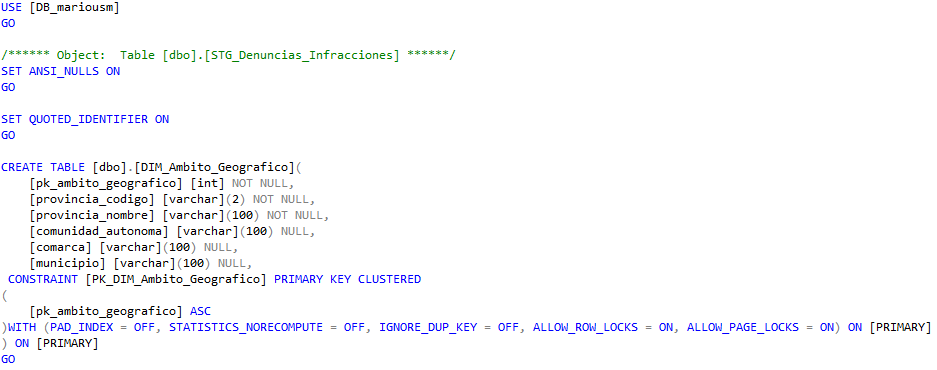


Ilustración - DIM\_Ambito\_Geografico.

#### Tabla dimensión DIM\_Fecha

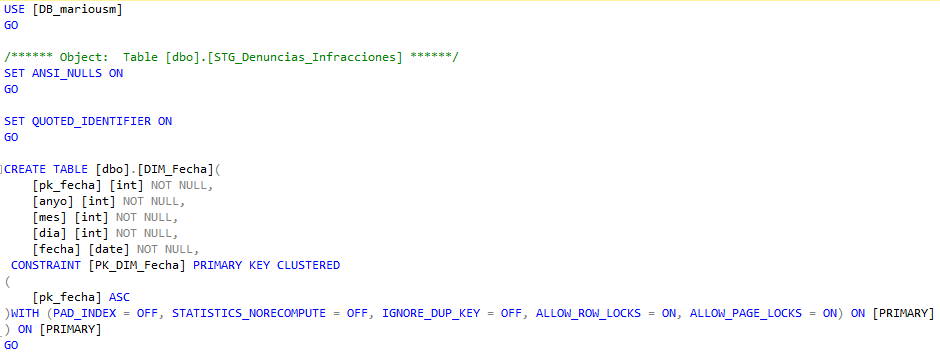


Ilustración - DIM\_Fecha.

#### Tabla dimensión DIM\_Grupo\_Edad

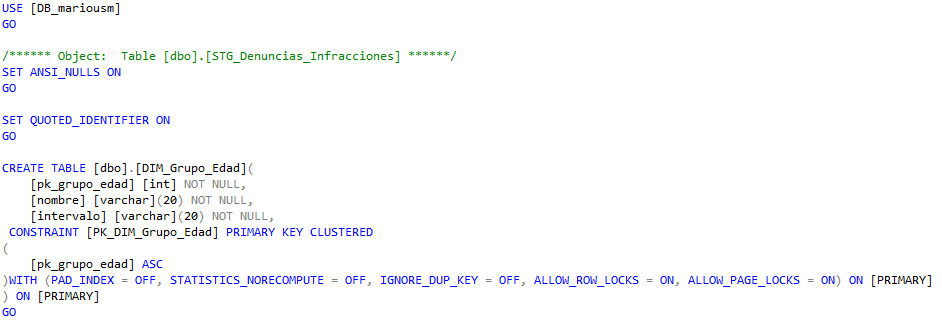


Ilustración - DIM\_Grupo\_Edad.

#### Tabla dimensión DIM\_Medicion

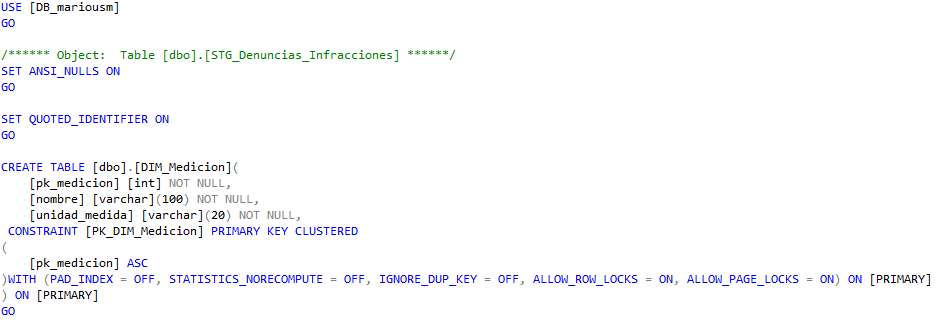


Ilustración - DIM\_Medicion.

#### Tabla dimensión DIM\_Tipologia

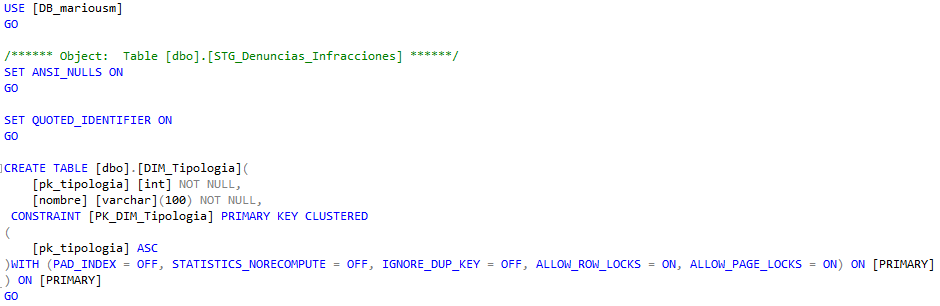


Ilustración - DIM\_Tipologia.

Comprobamos que todas las tablas de dimensiones se han creado correctamente:

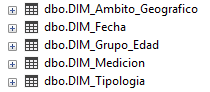


Ilustración - Tablas de dimensiones.

### Tablas de hechos

Finalmente creamos las diferentes tablas de los hechos:

#### Tabla hecho FACT\_Llamadas112

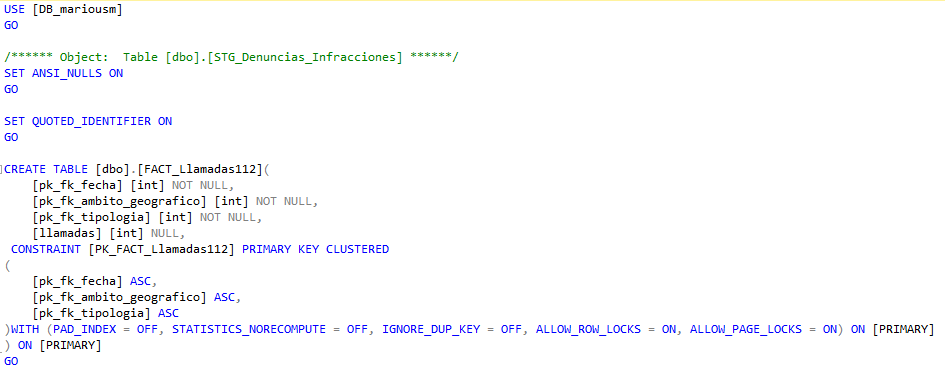


Ilustración - FACT\_Llamadas112.

#### Tabla hecho FACT\_Mediciones

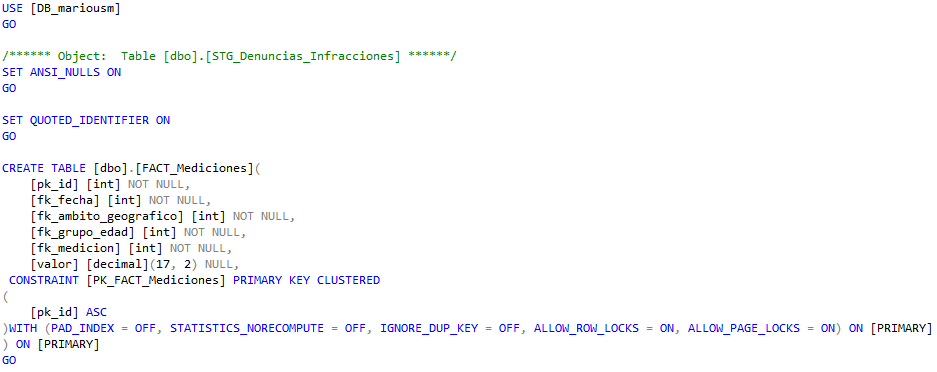


Ilustración - FACT\_Mediciones.

Realizamos los alter table de las tablas de hechos:

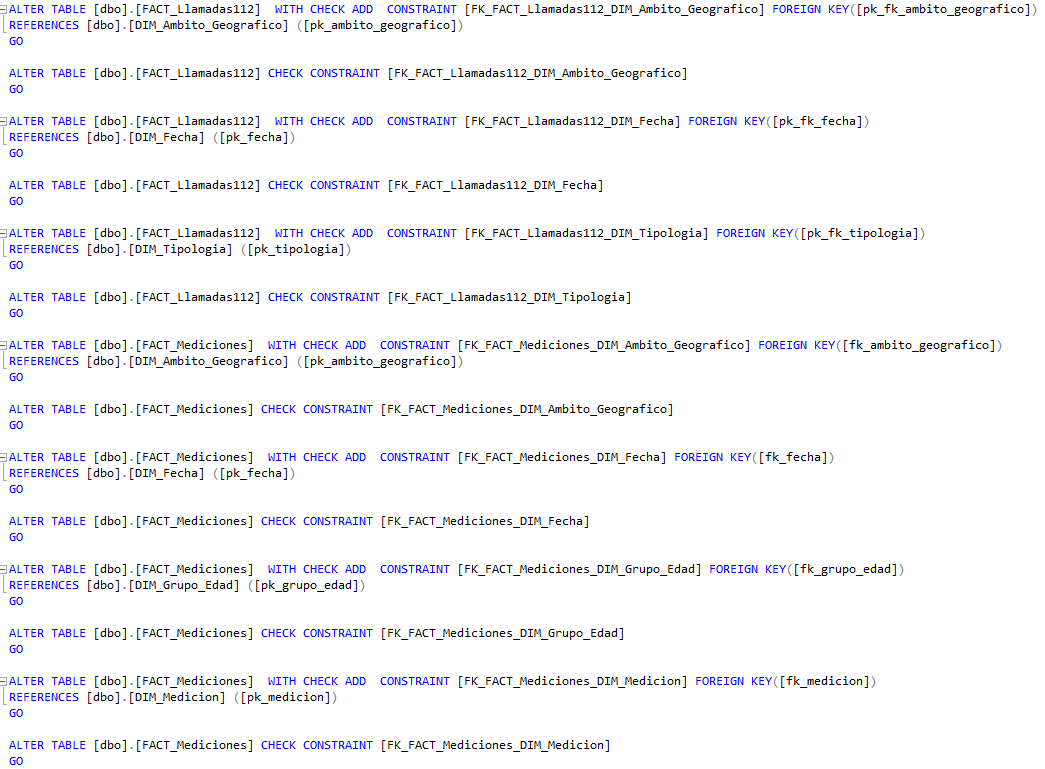


Ilustración - Alter table hechos.

Comprobamos que se han creado todas las tablas correspondientes:



Ilustración - Tablas de hechos.

## Bloque IN

En este bloque se van a realizar las transformaciones para que la información en forma bruta se pase a las tablas intermedias, y luego haremos uso de éstas para crear las transformaciones de dimensiones y hechos.

### Definición de variables de entorno

Es una buena práctica utilizar variables de entorno para así poder evitar errores en el definiciones futuras. Para ello accedemos a *kettle.properties* y definimos las siguientes variables:

Para el origen en el que se encuentran todos los archivos definimos la variables DIR\_ENT:

* Nombre: DIR
* Valor: F:\Mario\PRA2\data

Para la cadena de conexión a la base de datos vamos a usar:

* Nombre: HOSTNAME
* Valor: UCS1R1UOCSQL02
* Nombre: DBNAME
* Valor: DB\_mariousm
* Nombre: PORT
* Valor: 1433
* Nombre: USERNAME
* Valor: STUDENT\_mariousm

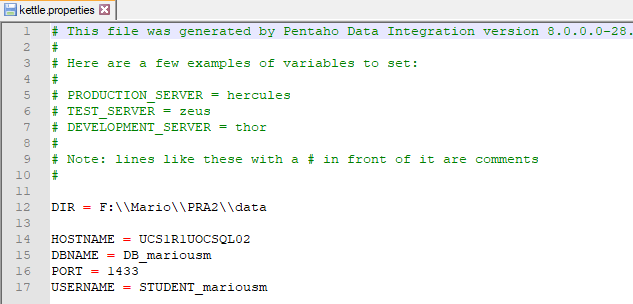


Ilustración - Variables de entorno.

### Conexión base de datos SQL Server

El siguiente paso es crear la conexión a la base de datos que va a ser usada tanto por las transformaciones como por los jobs que se realicen en esta práctica.

Para ello creamos la nueva conexión y establecemos los valores definidos en las variables de entorno:

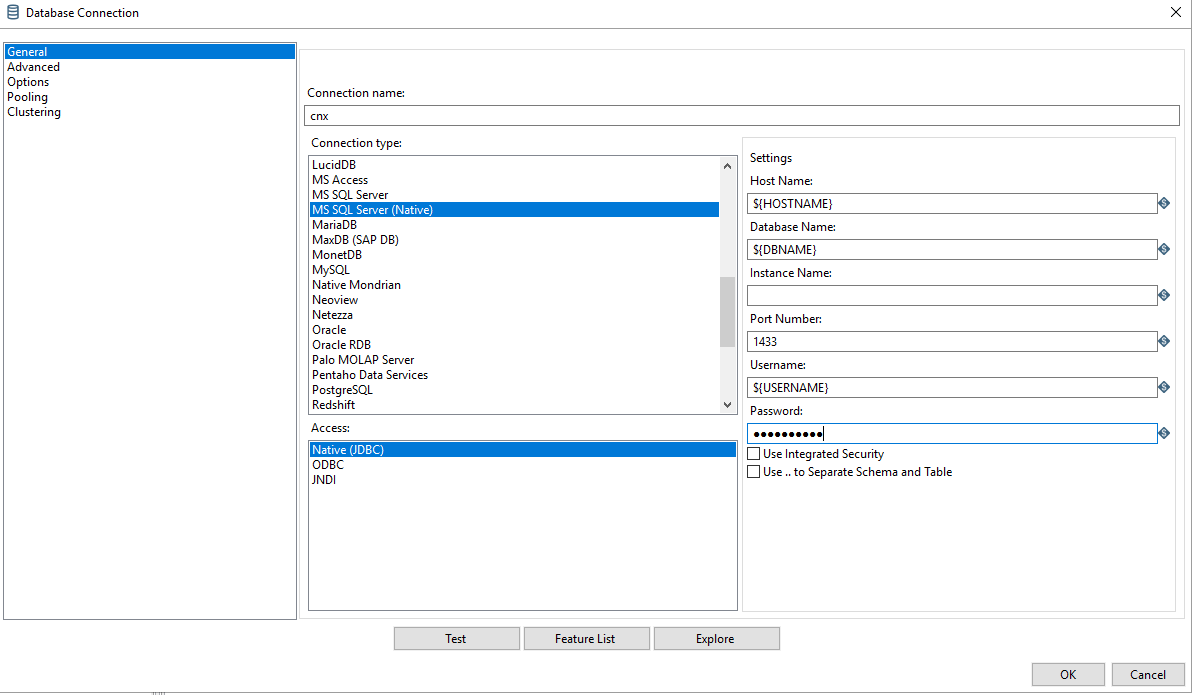


Ilustración - Conexión a la base de datos.

### Transformación IN\_DENUNCIAS\_INFRACCIONES

Una vez que ya hemos definido las variables de entorno y la conexión podemos proceder a realizar todas las transformaciones y trabajos.

La primera transformación que vamos a realizar se llama “IN\_DENUNCIAS\_INFRACCIONES”, su objetivo es leer todos los datos del archivo “ACUMULADO-DENUNCIAS-INFRACCIONES.xlsx” en la tabla intermedia “STG\_Denuncias\_Infracciones”.

En este caso no hemos hecho ninguna modificación en el Excel original, por lo que la transformación nos queda de la siguiente forma:

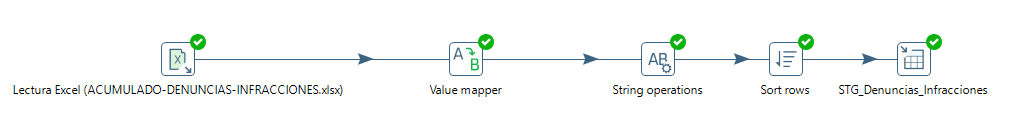


Ilustración - IN\_DENUNCIAS\_INFRACCIONES.

Ahora vamos a explicar paso a paso lo que hemos hecho:

#### Lectura del Excel

Lo primero de todo es leer el fichero Excel que se nos proporciona, y para ello usamos el componente “Microsoft Excel Input”, una vez hecho eso escribimos el nombre del paso, le indicamos el fichero que va a utilizar, y le indicamos que el formato del fichero Excel es la XLSX:

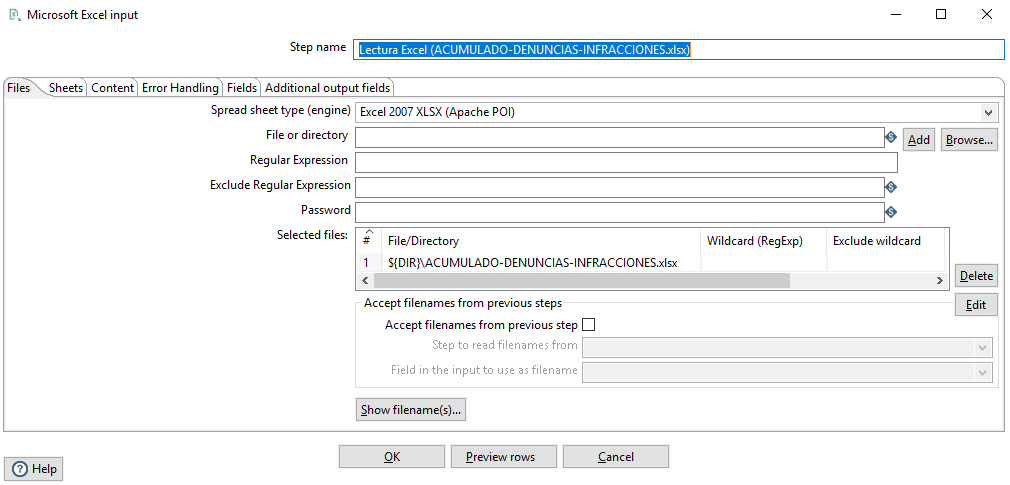


Ilustración - Lectura IN\_DENUNCIAS\_INFRACCIONES.

Una vez hecho eso, le indicamos qué hoja tiene que leer y desde qué fila y columna, en nuestro caso la hoja “Datos\_tratados” y la fila 5 columna 0:

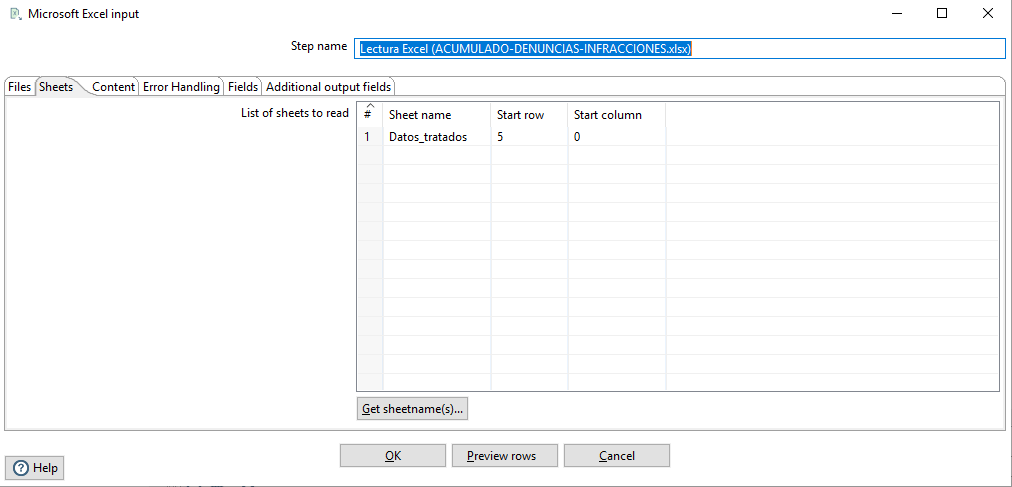


Ilustración - Lectura IN\_DENUNCIAS\_INFRACCIONES.

Posteriormente obtenemos los campos leídos en la pestaña “Field”:

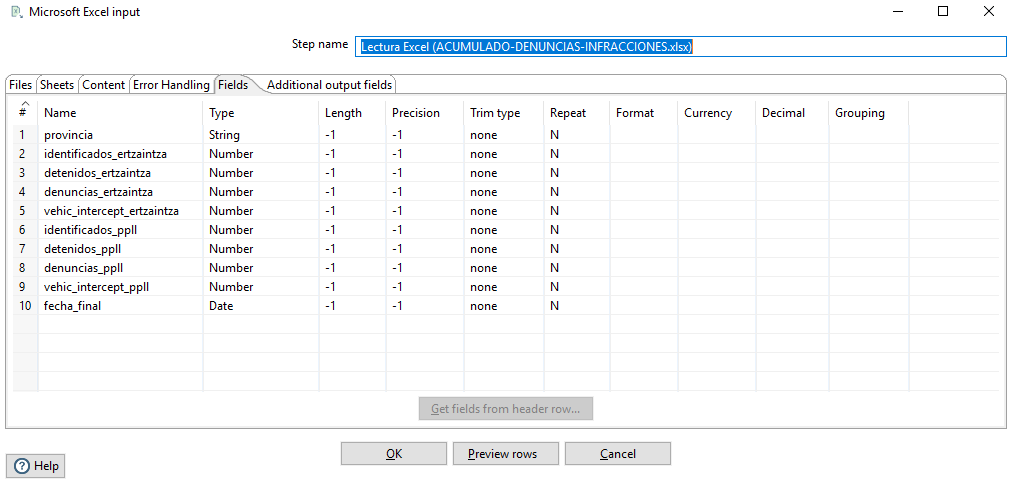


Ilustración - Lectura IN\_DENUNCIAS\_INFRACCIONES.

#### Mapeo

Una vez leídos los datos vemos que las provincias están escritas en euskera, por lo que para homogeneizar los datos hemos decidido convertirlas al castellano. Por lo tanto, hacemos la traducción tal y como vemos en la siguiente captura:

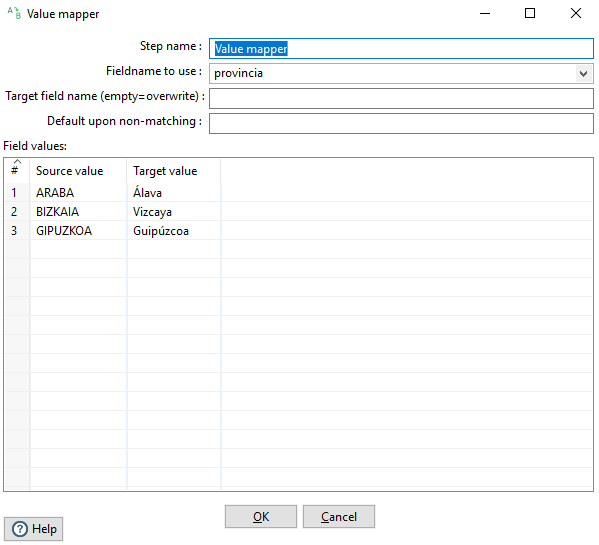


Ilustración - Mapeo Valores IN\_DENUNCIAS\_INFRACCIONES.

#### Normalización

Posteriormente hacemos una normalización de los campos que son de tipo “string”, ya que en éstos vamos a convertir los valores a mayúscula y sin espacios, tal y como vemos en la siguiente ilustración:

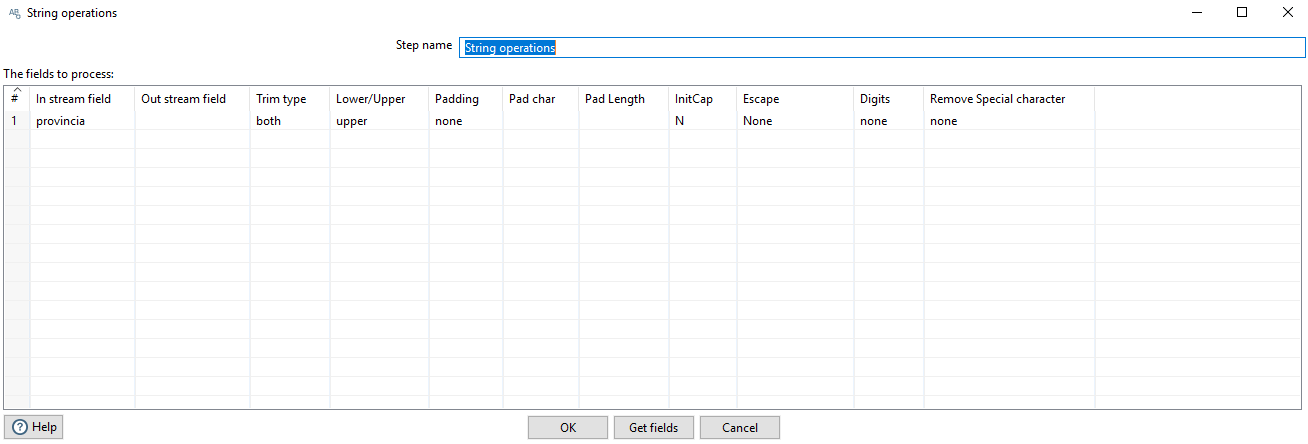


Ilustración - Normalización Strings IN\_DENUNCIAS\_INFRACCIONES.

#### Ordenación

Posteriormente, ordenamos todos los campos de forma ascendente:

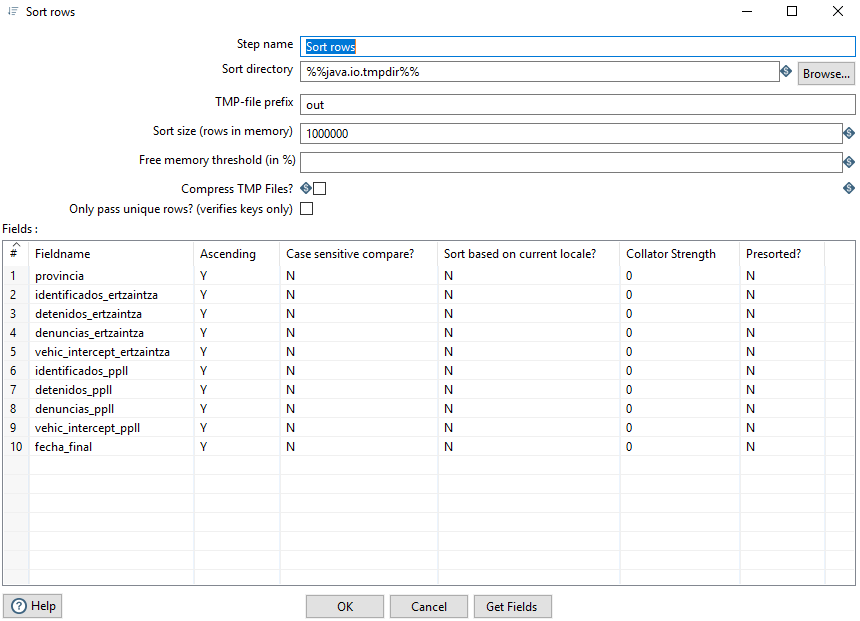


Ilustración - Ordenación IN\_DENUNCIAS\_INFRACCIONES.

#### Guardado

Finalmente, introducimos todos los valores en la base de datos, es decir, en la tabla intermedia STG\_Denuncias\_Infracciones, indicamos que haga un truncate de la tabla y con la conexión definida guardamos los valores en la tabla correspondiente:

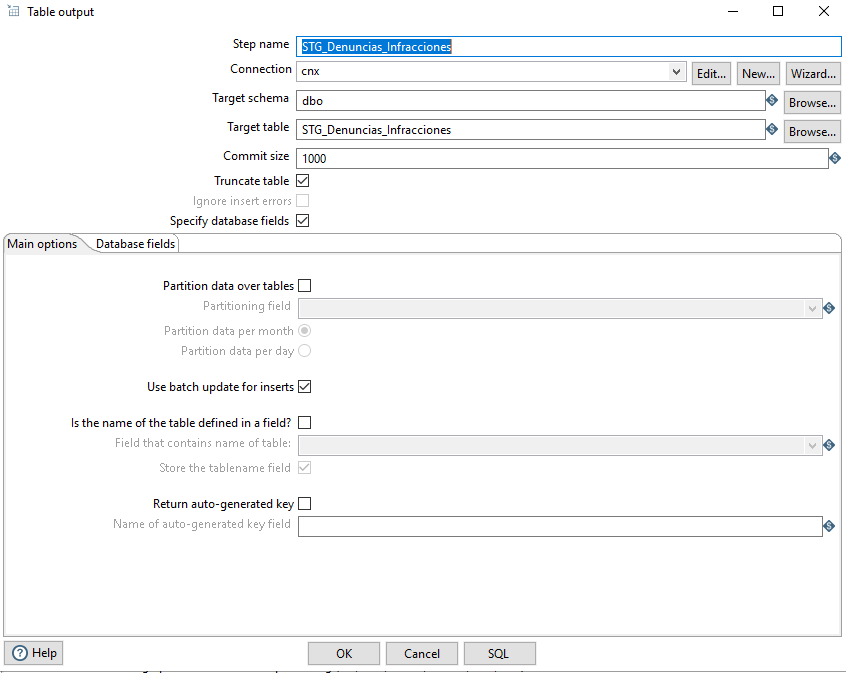


Ilustración - Guardado IN\_DENUNCIAS\_INFRACCIONES.

Al ejecutar la anterior transformación obtenemos las siguiente métricas:

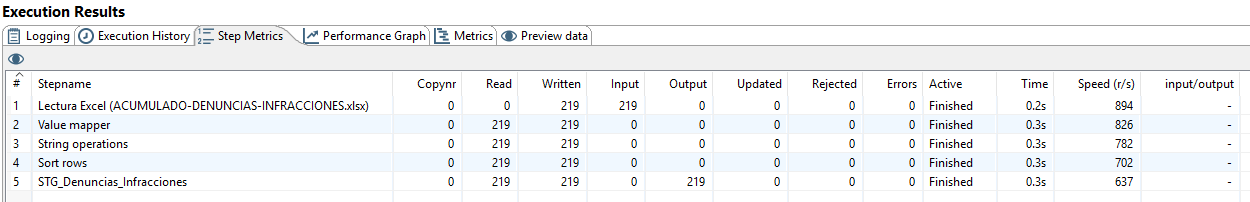


Ilustración - Métricas IN\_DENUNCIAS\_INFRACCIONES.

Observamos que tenemos 219 registros leídos y en nuestra base de datos se han almacena también 219 registros, por lo que la información es correcta.

### Transformación IN\_POBLACION

La segunda transformación que vamos a realizar se llama “IN\_POBLACION”, su objetivo es leer todos los datos del archivo “poblacion\_9687bsc.csv” y almacenarlos en la taba intermedia “STG\_Poblacion”.

En este caso no hemos hecho ninguna modificación al fichero original, por lo que la transformación nos queda de la siguiente forma:

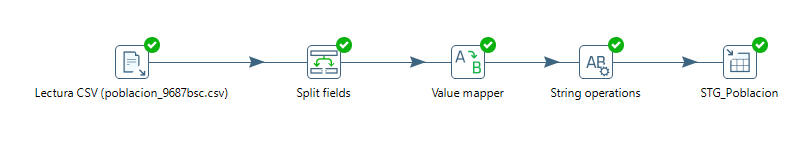


Ilustración - IN\_POBLACION.

#### Lectura CSV

Lo primero que debemos de hacer es cargar la información que se nos proporciona a partir del fichero CSV correspondiente. Por lo tanto, lo primero escribimos el nombre del paso, indicamos el fichero y el delimitador del CSV, en nuestro caso “;”:

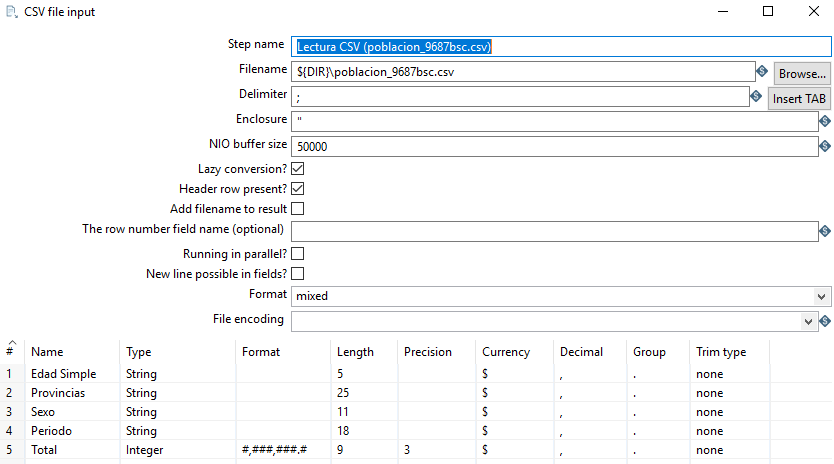


Ilustración - Lectura IN\_POBLACION.

Cabe destacar que hemos tenido que modificar el tipo del campo “Total” ya que lo reconocía como decimal cuando realmente es un entero.

#### Split

Luego separamos el campo “Provincias”, para así obtener tanto el código como la provincia correspondiente, para ello indicamos que el campo que queremos separar es “Provincias”, y luego en el grid establecemos los nuevos campos:

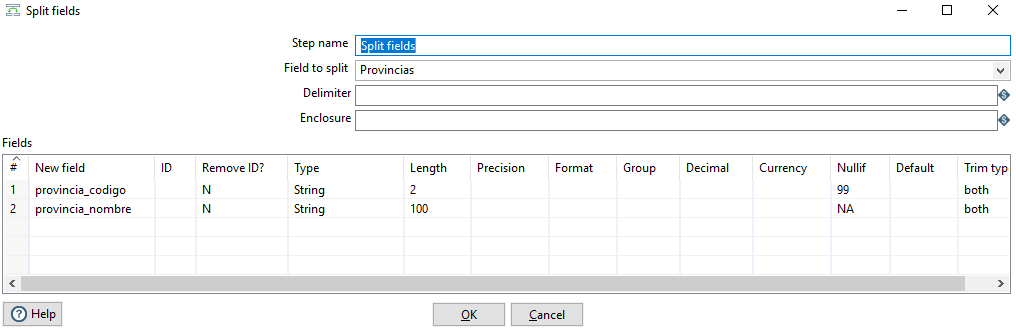


Ilustración - Separación Campos IN\_POBLACION.

#### Mapeo

Una vez hecho el paso anterior tenemos que mapear valores, esto se debe a que los nombres de las provincias no son del todo correctos (aparecen en gallego, euskera, catalán y valenciano). Además, al hacer la separación algunos nombres de provincias compuestas han desaparecido, es por ello que necesitamos de este paso para solventar los problemas:

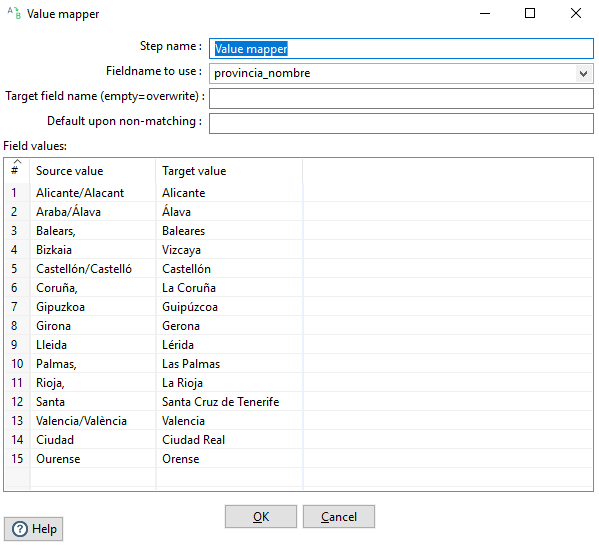


Ilustración - Mapeo Valores IN\_POBLACION.

#### Normalización

Antes de almacenar los datos en la base de datos, vamos a normalizar los strings para que todos estén en mayúsculas y no tengan espacios ni al principio ni al final:

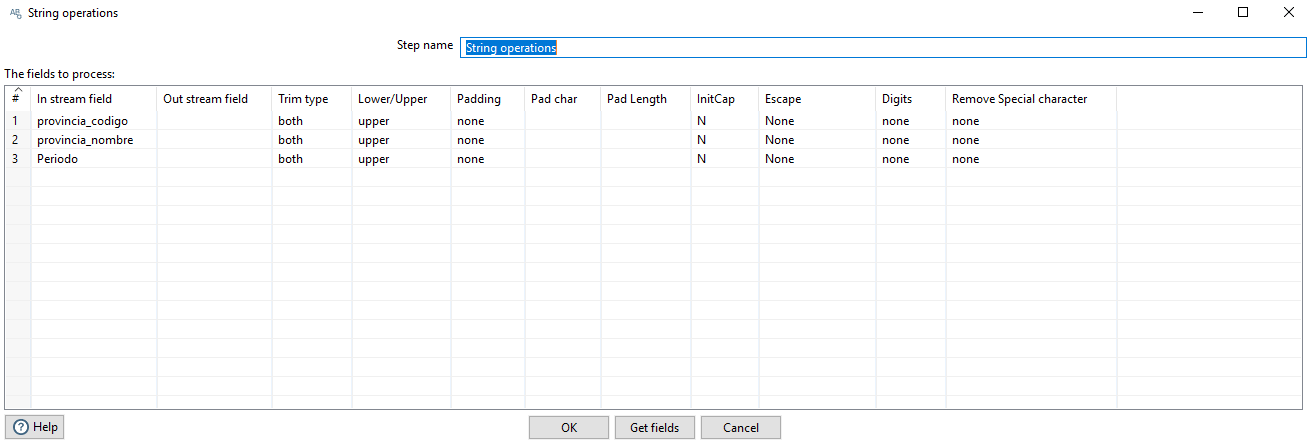


Ilustración - Normalización Strings IN\_POBLACION.

#### Guardado

Finalmente, guardamos los datos en la tabla “STG\_Poblacion”, indicando que haga un truncate de la tabla y asociamos los campos:

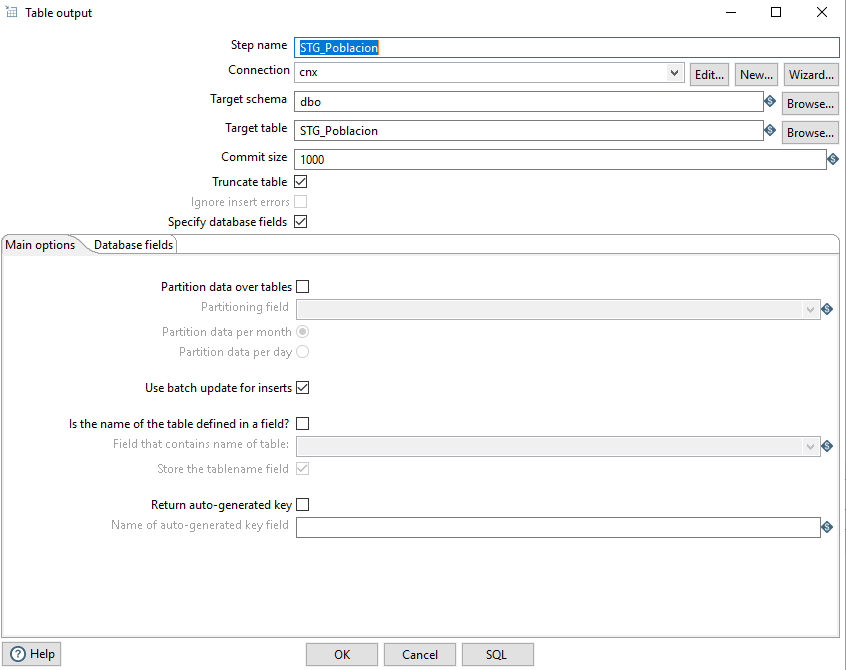


Ilustración - Guardado IN\_POBLACION.

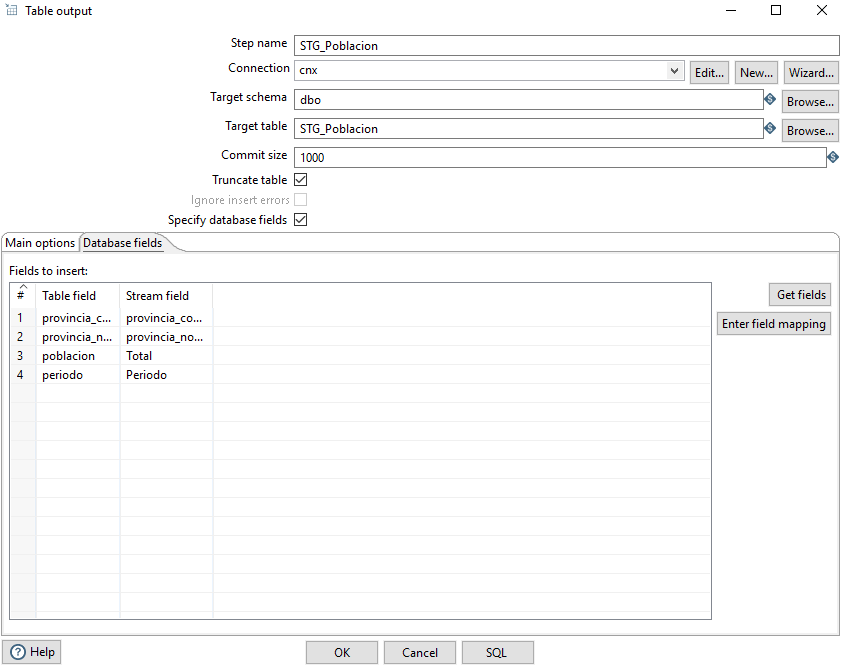


Ilustración - Guardado IN\_POBLACION.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

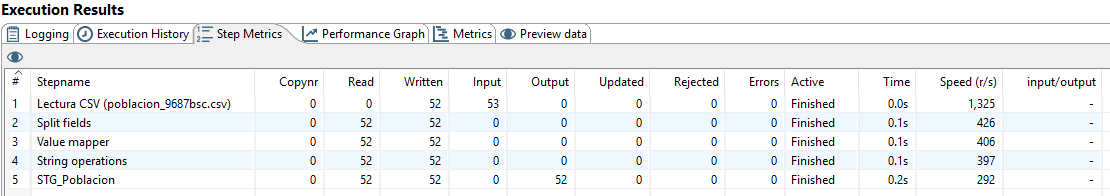


Ilustración - Métricas IN\_POBLACION.

Observamos que tenemos 53 registros (52 registros + 1 cabecera) y se han almacenado 52 registros, por lo que la información es correcta.

### Transformación IN\_MOVILIDAD

La tercera transformación que vamos a realizar se llama “IN\_MOVILIDAD”, su objetivo es leer todos los datos del archivo “35167bsc.csv” y guardarlos en la tabla intermedia “STG\_Movilidad”.

En este caso no hemos hecho ninguna modificación en el fichero CSV original, por lo que la transformación nos queda de la siguiente manera:

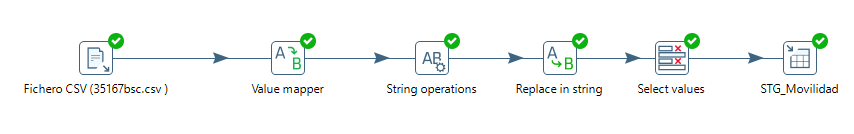


Ilustración - IN\_MOVILIDAD.

Ahora vamos a explicar paso a paso lo que hemos hecho:

#### Lectura CSV

Lo primero que tenemos que hacer es leer el fichero CSV que se nos proporciona, luego escribimos el nombre del paso, indicamos el fichero y el delimitador del CSV, en nuestro caso “,”:

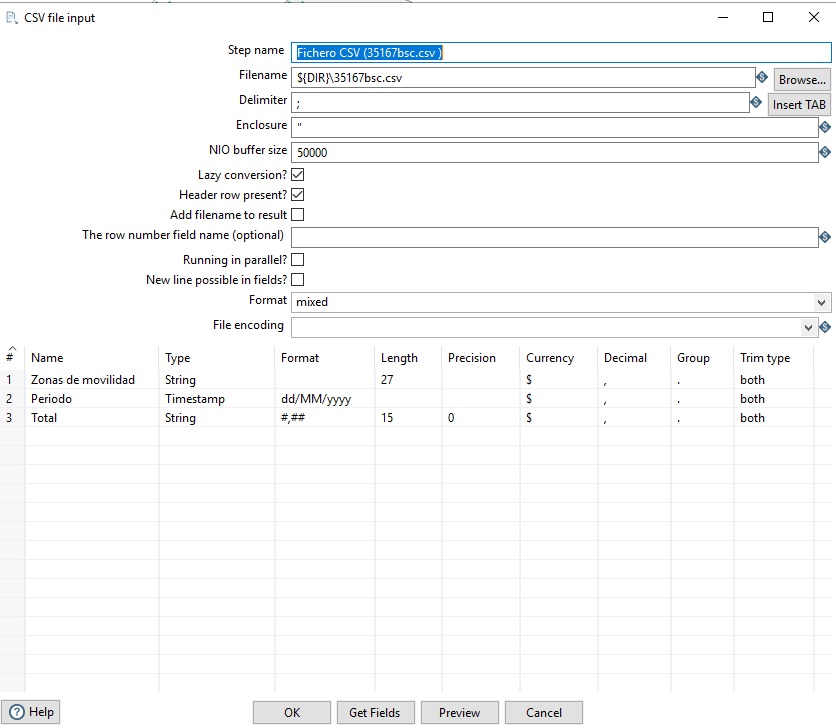


Ilustración - Lectura IN\_MOVILIDAD.

Cabe destacar que el atributo “Total” hemos indicado que sea de tipo string, ya que si considerábamos que fuera numérico a la hora de introducirlo en la base de datos no guardaba los decimales.

#### Mapeo

Una vez leídos todos los datos, tenemos que realizar un mapeo del campo “Zonas de movilidad”, esto se debe a que los nombres de las provincias vienen en (euskera, gallego, catalán, valenciano y balear), sin embargo para homogeneizar todas las provincias las traducimos al castellano.

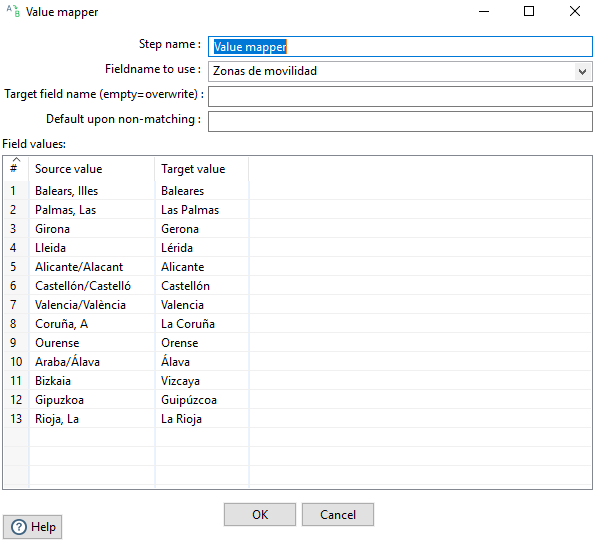


Ilustración - Mapeo Valores IN\_MOVILIDAD.

#### Normalización

Una vez que tenemos ya los datos de forma correcta, normalizamos las provincias para que no haya un espacio al principio o al final de la cadena, y establecemos que todas las cadenas estén en mayúsculas:

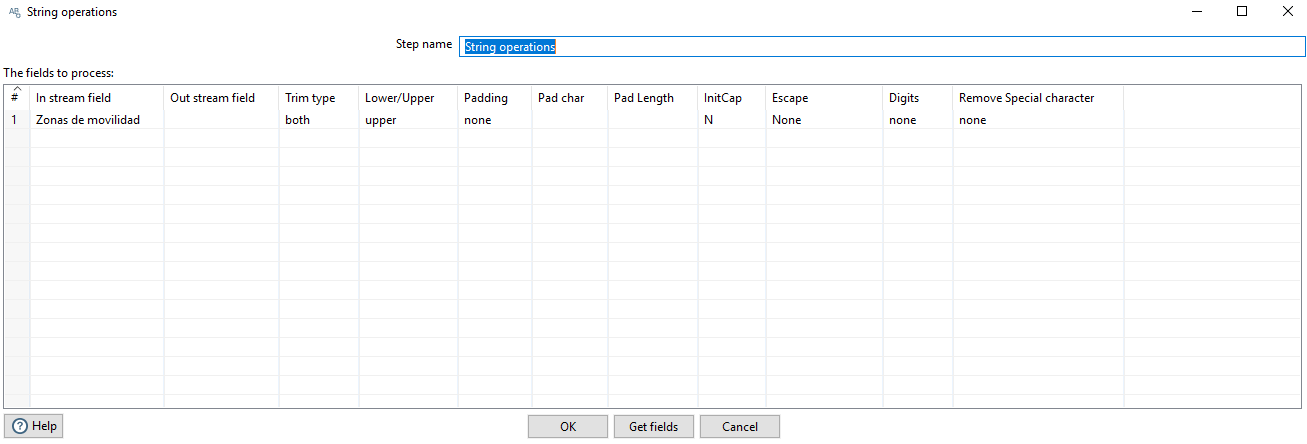


Ilustración - Normalización IN\_MOVILIDAD.

#### Replace

Posteriormente reemplazamos en el string de “Total” la coma por el punto, ya que de esta manera cuando luego introduzcamos el valor en la base de datos sí que no va a aparecer con decimales:

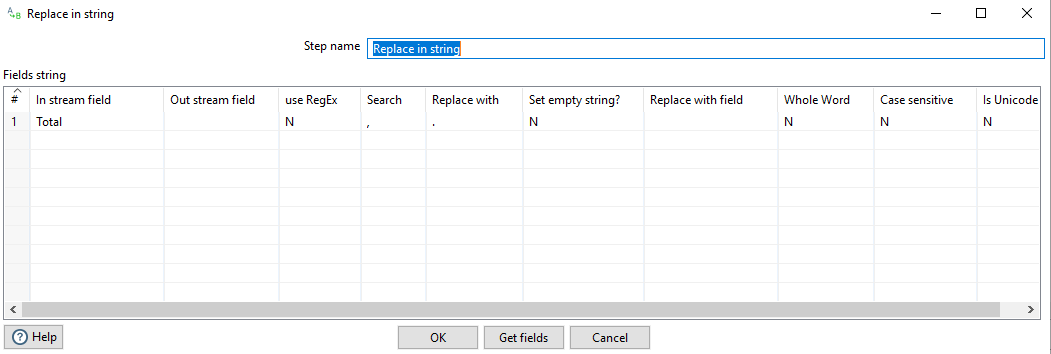


Ilustración - Replace IN\_MOVILIDAD.

#### Select Values

Cuando ya tenemos el string modificado, hay que convertirlo de decimal, ya que en la tabla de la base de datos el “Total” lo almacenamos como un número. Para ello, hacemos uso del componente “Select\_Values” y en “Meta-data” establecemos que cree un nuevo campo que sea de tipo numérico con el campo original “Total”, el resultado de esta operación lo guardamos en “totalSQL”:

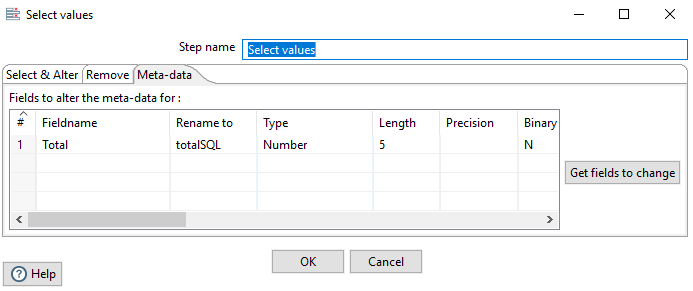


Ilustración - Select Values IN\_MOVILIDAD.

#### Guardado

Finalmente, guardamos todo el proceso realizado en la tabla “STG\_Movilidad”, indicando que hay un truncate de la tabla y asociamos los campos obtenidos con los de la tabla:

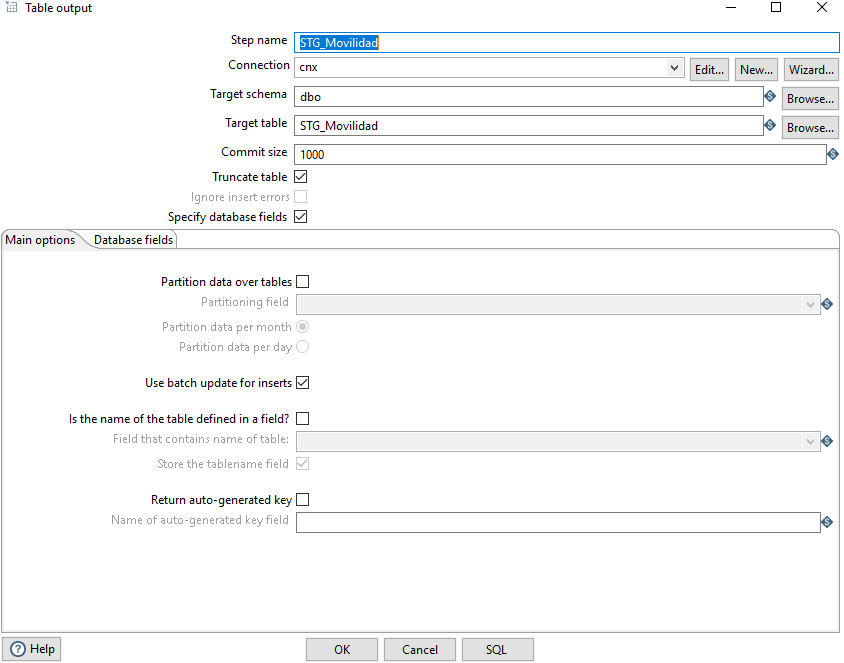


Ilustración - Guardado IN\_MOVILIDAD.

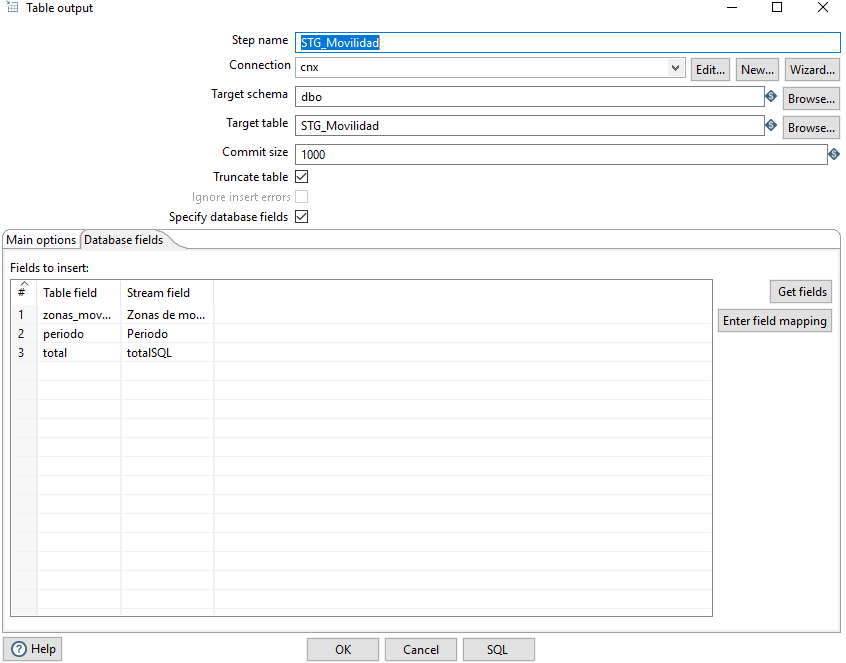


Ilustración - Guardado IN\_MOVILIDAD.

Para terminar con esta transformación obtenemos las métricas de su ejecución:

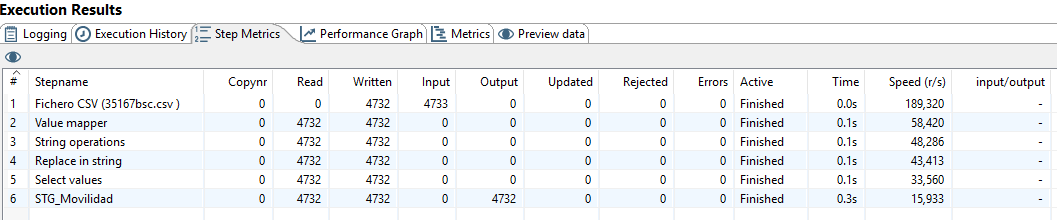


Ilustración - Métricas IN\_MOVILIDAD.

Como podemos observar, leemos 4733 registros (4732 observaciones + 1 cabecera) y almacenamos 4732, por lo que la información es correcta.

### Transformación IN\_AGLOMERACION

La cuarta transformación que vamos a realizar se llama “IN\_AGLOMERACION”, su objetivo es leer todo los datos del Excel “statistic\_id1104235\_covid-19\_-poblacion-que-evitaba-las-aglomeraciones-segun-edad-en-espana-2020.xlsx” y guardarlos en la tabla intermedia “STG\_AGLOMERACION”.

En este caso sí hemos hecho una modificación en el fichero Excel, no sabemos por qué motivo determinadas provincias tenían un espacio o caracter especial que no era visible y al hacer la lectura, independientemente de si hacíamos un “trim” o usábamos un “string operations” no eliminaba ese “espacio/carácter especial”, es por ello que hemos eliminado de forma manual dicho espacio en el campo “provincia” de los registros afectados.

La transformación nos queda de la siguiente manera:

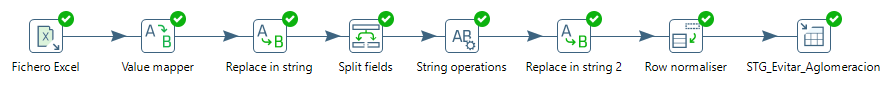


Ilustración - IN\_AGLOMERACION.

#### Lectura

Lo primero que tenemos que hacer es leer el fichero Excel que se nos ha proporcionado, para ello escribimos el nombre del paso, indicamos el fichero y su formato correspondiente a XLSX:

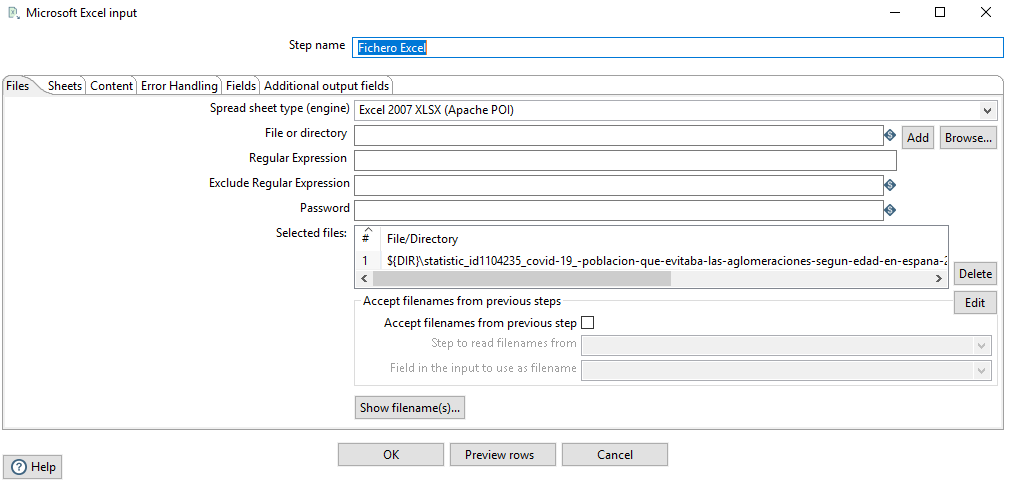


Ilustración - Lectura IN\_AGLOMERACION.

Una vez hecho eso, le indicamos qué hoja tiene que leer y desde qué fila y columna, en nuestro caso la hoja “Datos\_provincias” y la fila 5 columna 2:

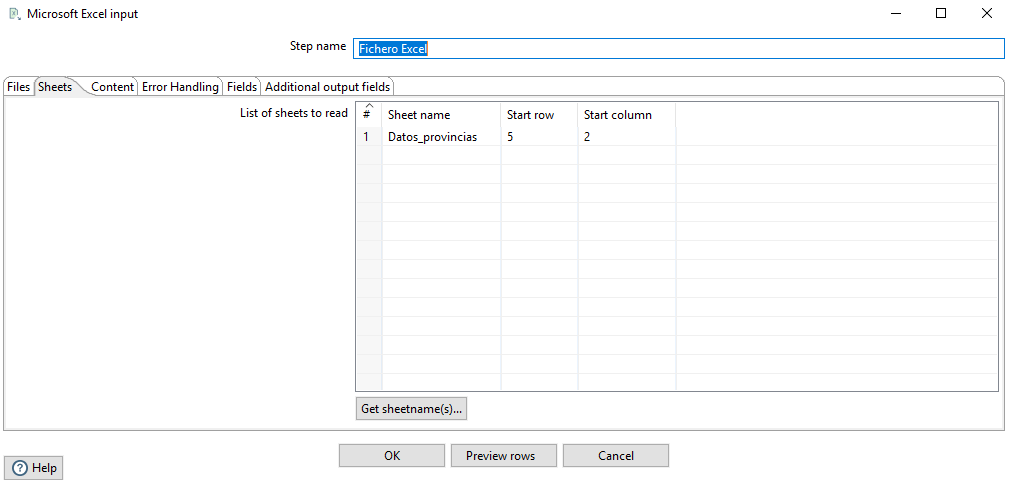


Ilustración - Lectura IN\_AGLOMERACION.

Posteriormente obtenemos los campos leídos en la pestaña “Fields”, los nombres de los campos han sido definidos de forma manual:

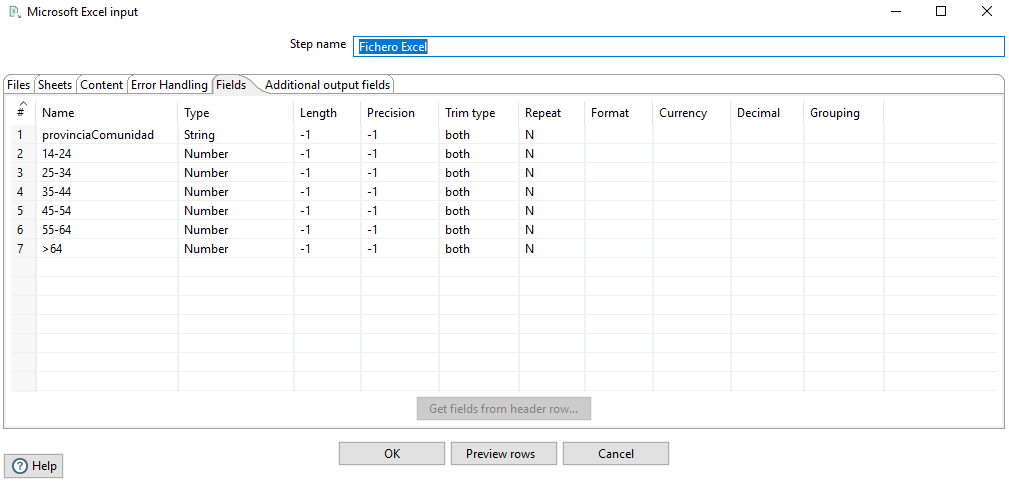


Ilustración - Lectura IN\_AGLOMERACIONES.

#### Mapeo

Al igual que ha sucedido con transformaciones anteriores, muchas provincias vienen también con su nombre en catalán/gallego/euskera/valenciano… Es por ello que hemos decido mantener el nombre en castellano, mapeando así los valores de las provincias que venían en otro idioma:

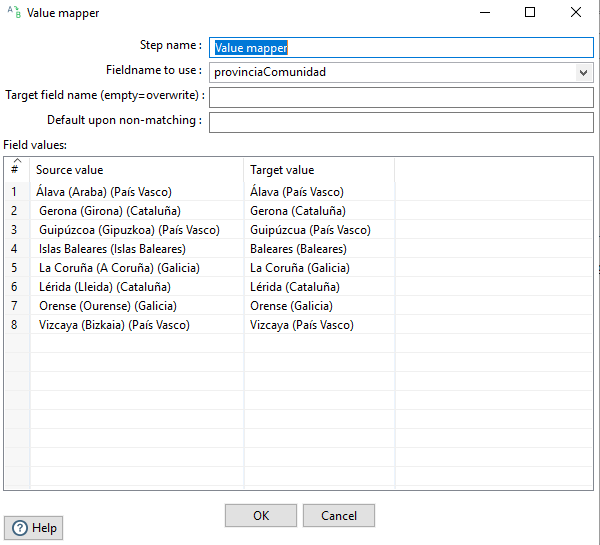


Ilustración - Mapeo Valores IN\_AGLOMERACION.

#### Replace

Posteriormente tenemos que hacer un replace del símbolo “)” por nada, de esta forma luego podemos dividir el campo en dos, para así obtener el nombre de la provincia y su comunidad autónoma:

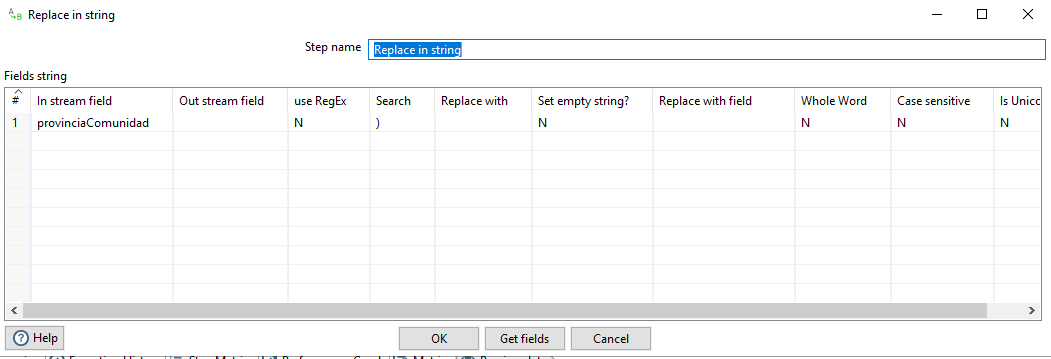


Ilustración - Replace IN\_AGLOMERACION.

#### Split

Una vez que ya tenemos la información que queremos, la podemos separar estableciendo como separado “[espacio](“, de esta forma creamos dos nuevos campos: uno para la provincia y otro para la comunidad.

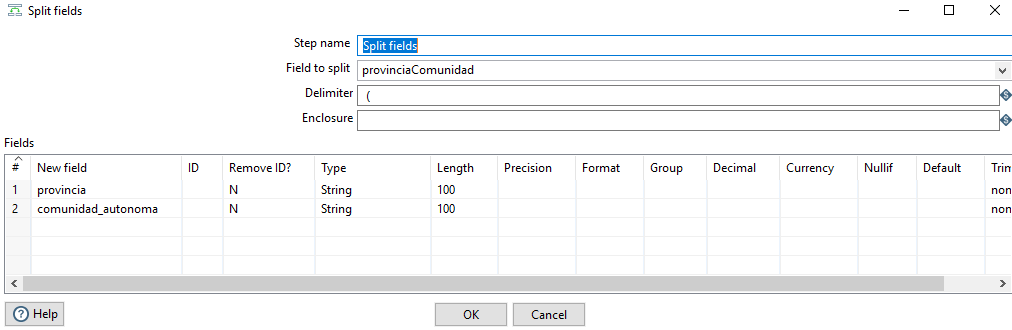


Ilustración - Split IN\_AGLOMERACION.

#### Normalización

Cuando ya tenemos la información separada, podemos hacer uso de un “string operations” para normalizar todos los strings, es decir, establecer mayúsculas y eliminar espacios:

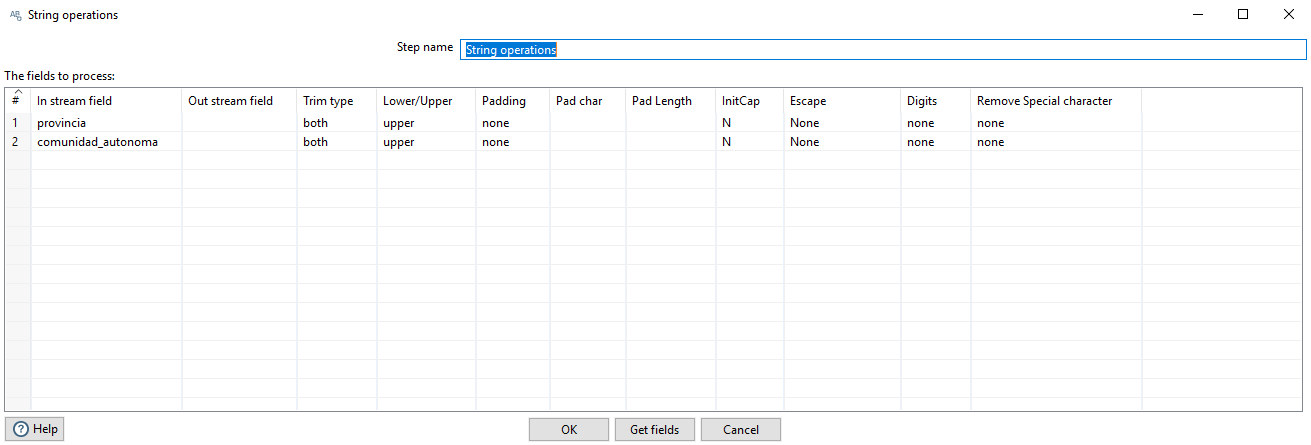


Ilustración - Normalización Strings IN\_AGLOMERACION.

#### Replace

Analizando los datos hemos visto que todas las comunidades y provincias cumplían las reglas ortográficas, sin embargo, la comunidad Aragón la escribían sin tilde. Es por ello que para mantener la misma lógica en todas las transformaciones hemos corregido dicho problema:

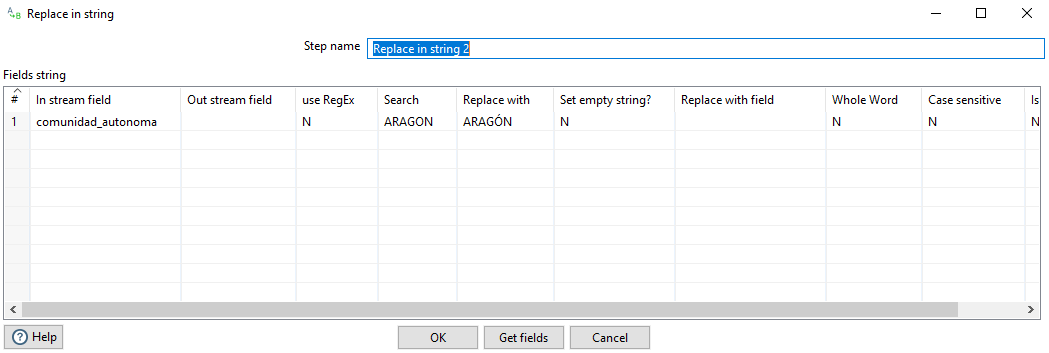


Ilustración - Replace IN\_AGLOMERACION.

#### Normalización filas

Posteriormente, hemos tenido que normalizar filas para que las columnas respectivas al grupo de edad fueran filas y no columnas. Para ello establecemos el nuevo campo que vamos a crear y los valores que va a tener dicho campo:

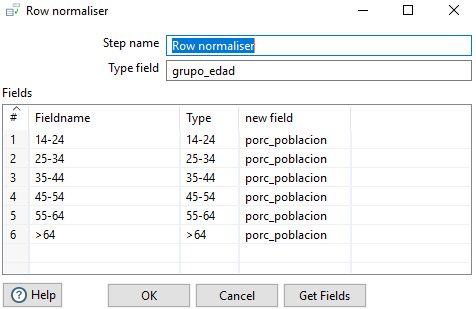


Ilustración - Normalización Filas IN\_AGLOMERACION.

#### Guardado

Finalmente, una vez que tenemos ya todos los datos normalizados podemos proceder al guardado de los mismo en la tabla intermedia “STG\_AGLOMERACION”. Tenemos que marcar el truncate table y asociar los campos:

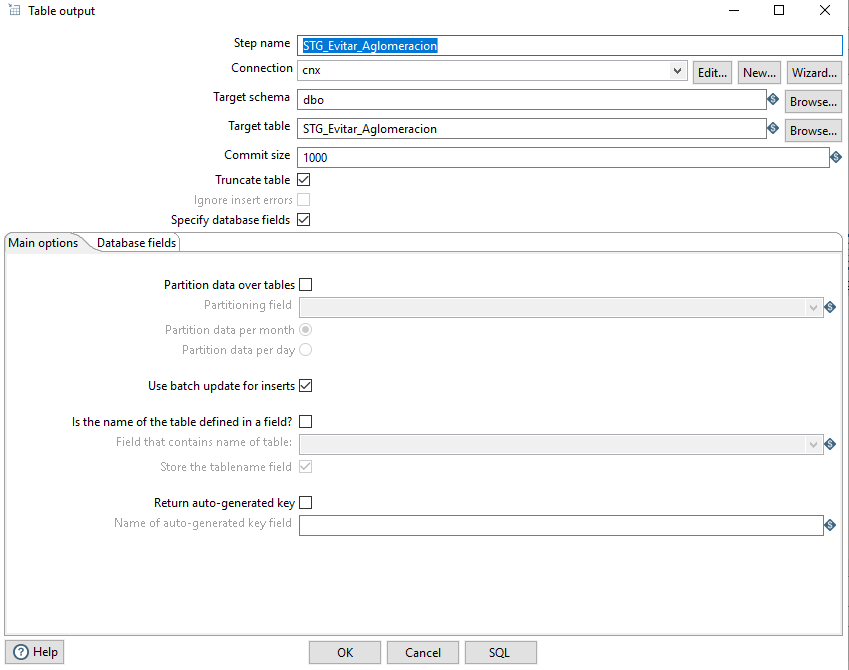


Ilustración - Guardado IN\_AGLOMERACIONES.

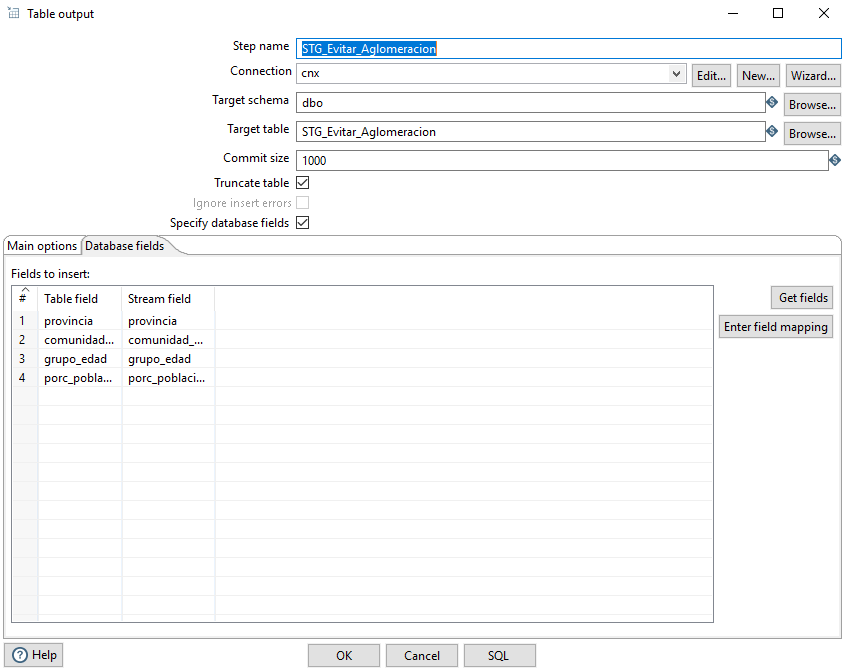


Ilustración - Guardado IN\_AGLOMERACIONES.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

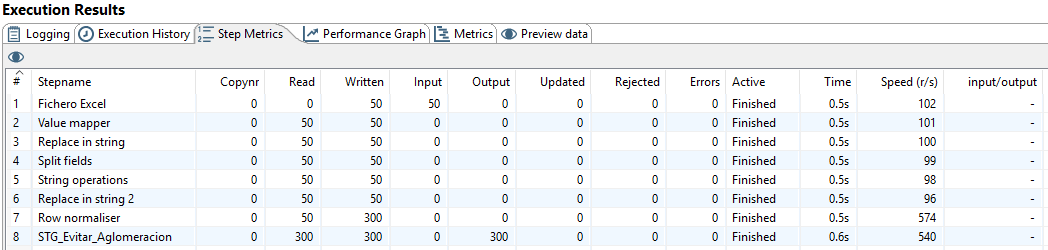


Ilustración - Métricas IN\_AGLOMERACION.

Como podemos observar leemos 50 registros y almacenamos 300, esto se debe a la normalización de las filas para el atributo “grupo\_edad”.

### Transformación IN\_LLAMADAS112

La última transformación respecto al bloque IN es “IN\_LLAMADAS112”, ésta se encarga de hacer la lectura del archivo “rows.xml” el cual contiene todas las llamadas, y las vamos a guardar en la tabla intermedia “STG\_Llamadas112”.

En este caso no hemos hecho ninguna modificación en el fichero XML original, por lo que la transformación nos queda de la siguiente forma:

|  |
| --- |
| Bibliografía |

a