

**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](#_Toc71889270)

[Índice de tablas 5](#_Toc71889271)

[Índice de ilustraciones 6](#_Toc71889272)

[1. Introducción 9](#_Toc71889273)

[1.1. Presentación 9](#_Toc71889274)

[1.2. Descripción 9](#_Toc71889275)

[2. Identificación de los procesos ETL 11](#_Toc71889276)

[2.1. Bloque IN 11](#_Toc71889277)

[2.2. Bloque TR 12](#_Toc71889278)

[2.2.1. Dimensiones 12](#_Toc71889279)

[2.2.2. Hechos 13](#_Toc71889280)

[3. Diseño y desarrollo de los procesos ETL 15](#_Toc71889281)

[3.1. Creación de tablas 15](#_Toc71889282)

[3.1.1. Tablas del área intermedia (*staging area*) 15](#_Toc71889283)

[3.1.2. Tablas de las dimensiones 18](#_Toc71889284)

[3.1.3. Tablas de hechos 20](#_Toc71889285)

[3.2. Bloque IN 22](#_Toc71889286)

[3.2.1. Definición de variables de entorno 22](#_Toc71889287)

[3.2.2. Conexión base de datos SQL Server 23](#_Toc71889288)

[3.2.3. Transformación IN\_DENUNCIAS\_INFRACCIONES 24](#_Toc71889289)

[3.2.4. Transformación IN\_POBLACION 28](#_Toc71889290)

[3.2.5. Transformación IN\_MOVILIDAD 32](#_Toc71889291)

[3.2.6. Transformación IN\_AGLOMERACION 37](#_Toc71889292)

[3.2.7. Transformación IN\_LLAMADAS112 44](#_Toc71889293)

[3.3. Bloque TR Dimensiones 51](#_Toc71889294)

[3.3.1. Transformación TR\_DIM\_GRUPO\_EDAD 51](#_Toc71889295)

[3.3.2. Transformación TR\_DIM\_Medicion 54](#_Toc71889296)

[3.3.3. Transformación TR\_DIM\_TIPOLOGIA 58](#_Toc71889297)

[3.3.4. Transformación TR\_DIM\_AMBITO\_GEOGRAFICO 62](#_Toc71889298)

[4. Bibliografía 63](#_Toc71889299)

|  |
| --- |
| Índice de tablas |

[Tabla 1 - Procesos ETL Bloque IN. 12](#_Toc71889332)

[Tabla 2 - Procesos ETL Bloque TR Dimensiones. 13](#_Toc71889333)

[Tabla 3 - Procesos ETL Bloque TR Hechos. 14](#_Toc71889334)

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

[Ilustración 1 - STG\_Denuncias\_Infracciones. 17](#_Toc71889337)

[Ilustración 2 - STG\_Poblacion. 17](#_Toc71889338)

[Ilustración 3 - STG\_Llamadas112. 18](#_Toc71889339)

[Ilustración 4 - STG\_Movilidad. 18](#_Toc71889340)

[Ilustración 5 - STG\_Evitar\_Aglomeracion. 19](#_Toc71889341)

[Ilustración 6 - Tablas de staging area. 19](#_Toc71889342)

[Ilustración 7 - DIM\_Ambito\_Geografico. 19](#_Toc71889343)

[Ilustración 8 - DIM\_Fecha. 20](#_Toc71889344)

[Ilustración 9 - DIM\_Grupo\_Edad. 20](#_Toc71889345)

[Ilustración 10 - DIM\_Medicion. 20](#_Toc71889346)

[Ilustración 11 - DIM\_Tipologia. 21](#_Toc71889347)

[Ilustración 12 - Tablas de dimensiones. 21](#_Toc71889348)

[Ilustración 13 - FACT\_Llamadas112. 21](#_Toc71889349)

[Ilustración 14 - FACT\_Mediciones. 22](#_Toc71889350)

[Ilustración 15 - Alter table hechos. 22](#_Toc71889351)

[Ilustración 16 - Tablas de hechos. 23](#_Toc71889352)

[Ilustración 17 - Variables de entorno. 24](#_Toc71889353)

[Ilustración 18 - Conexión a la base de datos. 24](#_Toc71889354)

[Ilustración 19 - IN\_DENUNCIAS\_INFRACCIONES. 25](#_Toc71889355)

[Ilustración 20 - Lectura IN\_DENUNCIAS\_INFRACCIONES. 25](#_Toc71889356)

[Ilustración 21 - Lectura IN\_DENUNCIAS\_INFRACCIONES. 26](#_Toc71889357)

[Ilustración 22 - Lectura IN\_DENUNCIAS\_INFRACCIONES. 26](#_Toc71889358)

[Ilustración 23 - Mapeo Valores IN\_DENUNCIAS\_INFRACCIONES. 27](#_Toc71889359)

[Ilustración 24 - Normalización Strings IN\_DENUNCIAS\_INFRACCIONES. 27](#_Toc71889360)

[Ilustración 25 - Ordenación IN\_DENUNCIAS\_INFRACCIONES. 28](#_Toc71889361)

[Ilustración 26 - Guardado IN\_DENUNCIAS\_INFRACCIONES. 29](#_Toc71889362)

[Ilustración 27 - Métricas IN\_DENUNCIAS\_INFRACCIONES. 29](#_Toc71889363)

[Ilustración 28 - IN\_POBLACION. 30](#_Toc71889364)

[Ilustración 29 - Lectura IN\_POBLACION. 30](#_Toc71889365)

[Ilustración 30 - Separación Campos IN\_POBLACION. 31](#_Toc71889366)

[Ilustración 31 - Mapeo Valores IN\_POBLACION. 31](#_Toc71889367)

[Ilustración 32 - Normalización Strings IN\_POBLACION. 32](#_Toc71889368)

[Ilustración 33 - Guardado IN\_POBLACION. 32](#_Toc71889369)

[Ilustración 34 - Guardado IN\_POBLACION. 33](#_Toc71889370)

[Ilustración 35 - Métricas IN\_POBLACION. 33](#_Toc71889371)

[Ilustración 36 - IN\_MOVILIDAD. 34](#_Toc71889372)

[Ilustración 37 - Lectura IN\_MOVILIDAD. 34](#_Toc71889373)

[Ilustración 38 - Mapeo Valores IN\_MOVILIDAD. 35](#_Toc71889374)

[Ilustración 39 - Normalización IN\_MOVILIDAD. 35](#_Toc71889375)

[Ilustración 40 - Replace IN\_MOVILIDAD. 36](#_Toc71889376)

[Ilustración 41 - Select Values IN\_MOVILIDAD. 36](#_Toc71889377)

[Ilustración 42 - Guardado IN\_MOVILIDAD. 37](#_Toc71889378)

[Ilustración 43 - Guardado IN\_MOVILIDAD. 38](#_Toc71889379)

[Ilustración 44 - Métricas IN\_MOVILIDAD. 38](#_Toc71889380)

[Ilustración 45 - IN\_AGLOMERACION. 39](#_Toc71889381)

[Ilustración 46 - Lectura IN\_AGLOMERACION. 39](#_Toc71889382)

[Ilustración 47 - Lectura IN\_AGLOMERACION. 40](#_Toc71889383)

[Ilustración 48 - Lectura IN\_AGLOMERACIONES. 40](#_Toc71889384)

[Ilustración 49 - Mapeo Valores IN\_AGLOMERACION. 41](#_Toc71889385)

[Ilustración 50 - Replace IN\_AGLOMERACION. 41](#_Toc71889386)

[Ilustración 51 - Split IN\_AGLOMERACION. 42](#_Toc71889387)

[Ilustración 52 - Normalización Strings IN\_AGLOMERACION. 42](#_Toc71889388)

[Ilustración 53 - Replace IN\_AGLOMERACION. 43](#_Toc71889389)

[Ilustración 54 - Normalización Filas IN\_AGLOMERACION. 43](#_Toc71889390)

[Ilustración 55 - Guardado IN\_AGLOMERACIONES. 44](#_Toc71889391)

[Ilustración 56 - Guardado IN\_AGLOMERACIONES. 45](#_Toc71889392)

[Ilustración 57 - Métricas IN\_AGLOMERACION. 45](#_Toc71889393)

[Ilustración 58 - IN\_LLAMADAS112. 46](#_Toc71889394)

[Ilustración 59 - Lectura IN\_LLAMADAS112. 46](#_Toc71889395)

[Ilustración 60 - Lectura IN\_LLAMADAS112. 47](#_Toc71889396)

[Ilustración 61 - Lectura IN\_LLAMADAS112. 47](#_Toc71889397)

[Ilustración 62 - Mapeo Valores IN\_LLAMADAS112. 48](#_Toc71889398)

[Ilustración 63 - Mapeo Valores IN\_LLAMADAS112. 49](#_Toc71889399)

[Ilustración 64 - Normalización IN\_LLAMADAS112. 49](#_Toc71889400)

[Ilustración 65 - Guardado IN\_LLAMADAS112. 50](#_Toc71889401)

[Ilustración 66 - Guardado IN\_LLAMADAS112. 51](#_Toc71889402)

[Ilustración 67 - Métricas IN\_LLAMADAS112. 51](#_Toc71889403)

[Ilustración 68 - TR\_DIM\_GRUPO\_EDAD. 52](#_Toc71889404)

[Ilustración 69 - Borrado TR\_DIM\_GRUPO\_EDAD. 52](#_Toc71889405)

[Ilustración 70 - Grid TR\_DIM\_GRUPO\_EDAD. 53](#_Toc71889406)

[Ilustración 71 - Grid TR\_DIM\_GRUPO\_EDAD. 53](#_Toc71889407)

[Ilustración 72 - Normalización TR\_DIM\_GRUPO\_EDAD. 53](#_Toc71889408)

[Ilustración 73 - Secuenciación TR\_DIM\_GRUPO\_EDAD. 54](#_Toc71889409)

[Ilustración 74 - Guardado TR\_DIM\_GRUPO\_EDAD. 54](#_Toc71889410)

[Ilustración 75 - Guardado TR\_DIM\_GRUPO\_EDAD. 55](#_Toc71889411)

[Ilustración 76 - Métricas TR\_DIM\_GRUPO\_EDAD. 55](#_Toc71889412)

[Ilustración 77 - TR\_DIM\_MEDICION. 56](#_Toc71889413)

[Ilustración 78 - Borrado TR\_DIM\_MEDICION. 56](#_Toc71889414)

[Ilustración 79 - Grid TR\_DIM\_DIM\_MEDICION. 56](#_Toc71889415)

[Ilustración 80 - Grid TR\_DIM\_MEDICION. 57](#_Toc71889416)

[Ilustración 81 - Normalización TR\_DIM\_MEDICION. 57](#_Toc71889417)

[Ilustración 82 - Secuenciación TR\_DIM\_MEDICION. 58](#_Toc71889418)

[Ilustración 83 - Guardado TR\_DIM\_MEDICION. 58](#_Toc71889419)

[Ilustración 84 - Guardado TR\_DIM\_MEDICION. 59](#_Toc71889420)

[Ilustración 85 - Métricas TR\_DIM\_MEDICIONES. 59](#_Toc71889421)

[Ilustración 86 - TR\_DIM\_TIPOLOGIA. 60](#_Toc71889422)

[Ilustración 87 - Borrado TR\_DIM\_TIPOLOGIA. 60](#_Toc71889423)

[Ilustración 88 - Lectura TR\_DIM\_TIPOLOGIA. 61](#_Toc71889424)

[Ilustración 89 - Secuenciación TR\_DIM\_TIPOLOGIA. 62](#_Toc71889425)

[Ilustración 90 - Guardado TR\_DIM\_TIPOLOGIA. 62](#_Toc71889426)

[Ilustración 91 - Guardado TR\_DIM\_TIPOLOGIA. 63](#_Toc71889427)

[Ilustración 92 - Métricas TR\_DIM\_TIPOLOGIA. 63](#_Toc71889428)

|  |
| --- |
| 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 datos** | **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 | STG\_Llamadas112 |
| IN\_FECHAS | Almacenamos todas las fechas de todos los ficheros de datos proporcionados. | STG\_Llamadas112  STG\_Denuncias\_Infracciones  STG\_Movilidad | STG\_Fechas |

Tabla 1 - 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. | STG\_Fechas | DIM\_Fecha |
| TR\_DIM\_AMBITO\_GEOGRAFICO | Carga y transformación de la dimensión con los datos de los ámbitos geográficos. | STG\_Poblacion  STG\_Llamadas112  STG\_Evitar\_Aglomeracion | DIM\_Ambito\_Geografico |
| TR\_DIM\_GRUPO\_EDAD | Carga y transformación de la dimensión con los datos de los grupo de edad. | Manual, a partir de un grid. | DIM\_Grupo\_Edad |
| TR\_DIM\_MEDICION | Carga y transformación de la dimensión con los datos de las mediciones. | Manual, a partir de un grid. | DIM\_Medicion |
| TR\_DIM\_TIPOLOGIA | Carga y transformación de la dimensión con los datos de la tipología. | STG\_Llamadas112 | DIM\_Tipologia |

Tabla 2 - 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 | STG\_Denuncias\_infracciones  STG\_Evitar\_Aglomeracion  STG\_Movilidad  STG\_Poblacion |

Tabla 3 - 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

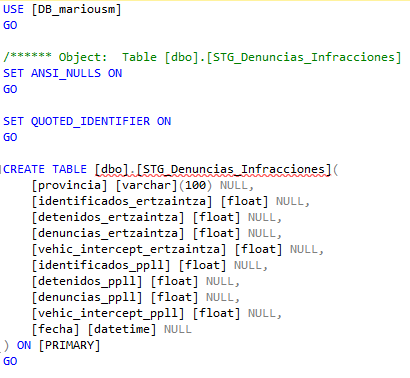


Ilustración 1 - STG\_Denuncias\_Infracciones.

#### Tabla intermedia STG\_Poblacion

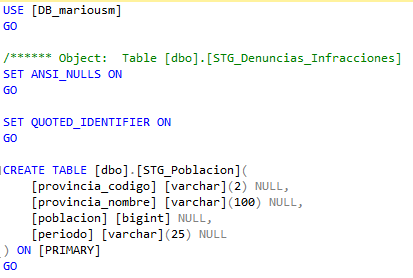


Ilustración 2 - STG\_Poblacion.

#### Tabla intermedia STG\_Llamadas112

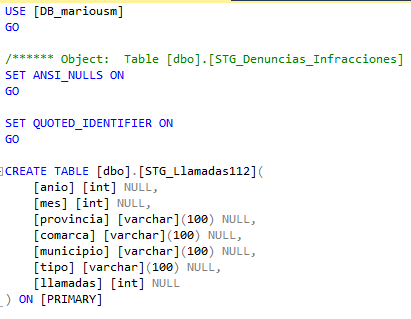


Ilustración 3 - STG\_Llamadas112.

#### Tabla intermedia STG\_Movilidad

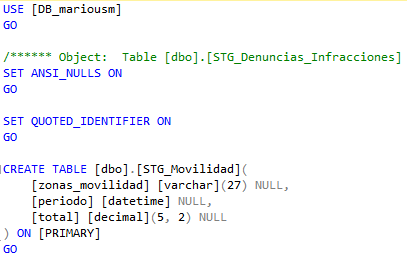


Ilustración 4 - STG\_Movilidad.

#### Tabla intermedia STG\_Evitar\_Aglomeracion

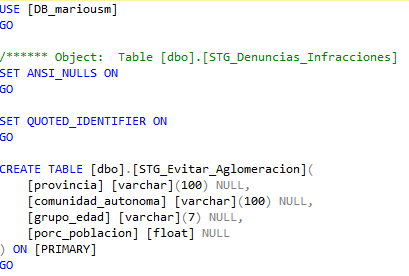


Ilustración 5 - STG\_Evitar\_Aglomeracion.

#### Tabla intermedia STG\_Fechas

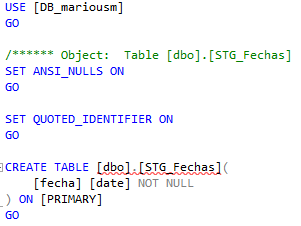


Ilustración - STG\_Fechas.

Comprobamos que todas las tabla intermedias se han creado correctamente:

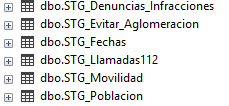


Ilustración 7 - Tablas de staging area.

### Tablas de las dimensiones

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

#### Tabla dimensión DIM\_Ambito\_Geografico

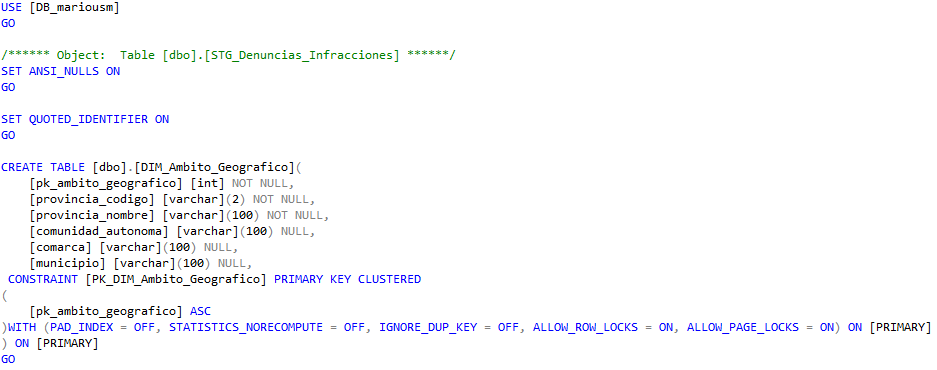


Ilustración 8 - DIM\_Ambito\_Geografico.

#### Tabla dimensión DIM\_Fecha

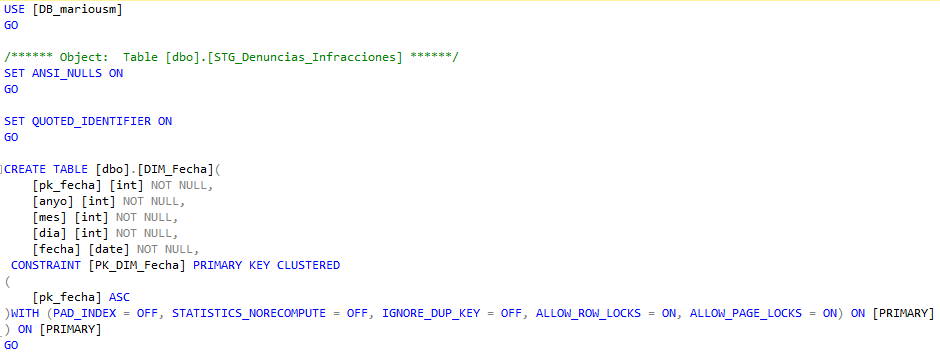


Ilustración 9 - DIM\_Fecha.

#### Tabla dimensión DIM\_Grupo\_Edad

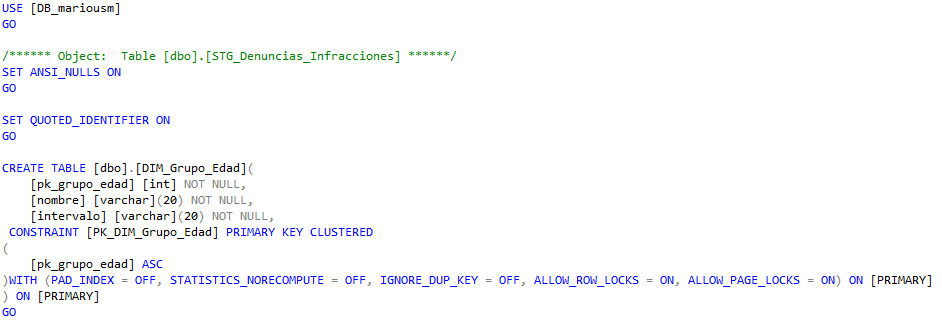


Ilustración 10 - DIM\_Grupo\_Edad.

#### Tabla dimensión DIM\_Medicion

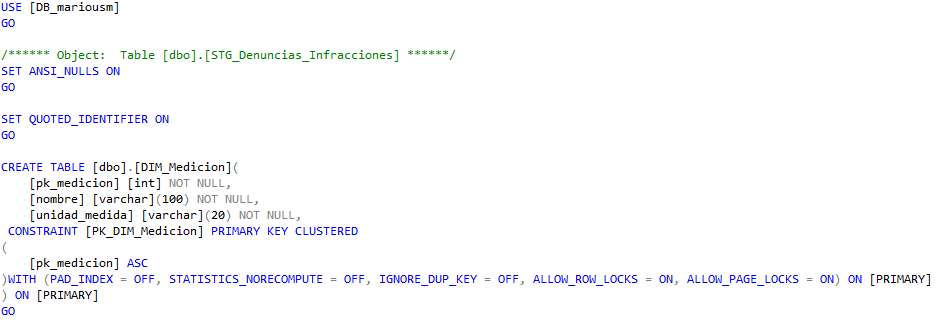


Ilustración 11 - DIM\_Medicion.

#### Tabla dimensión DIM\_Tipologia

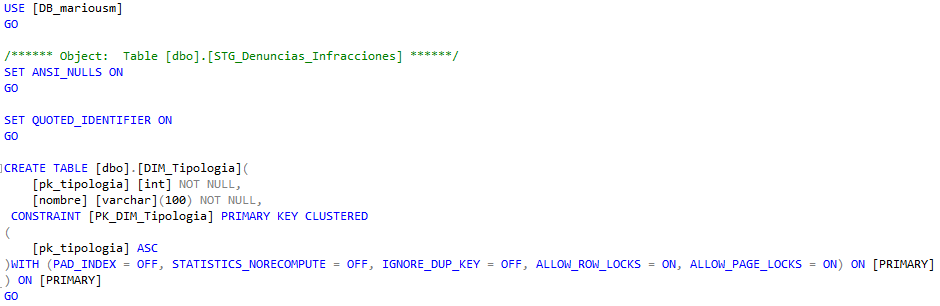


Ilustración 12 - DIM\_Tipologia.

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

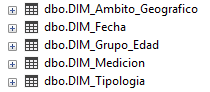


Ilustración 13 - Tablas de dimensiones.

### Tablas de hechos

Finalmente creamos las diferentes tablas de los hechos:

#### Tabla hecho FACT\_Llamadas112

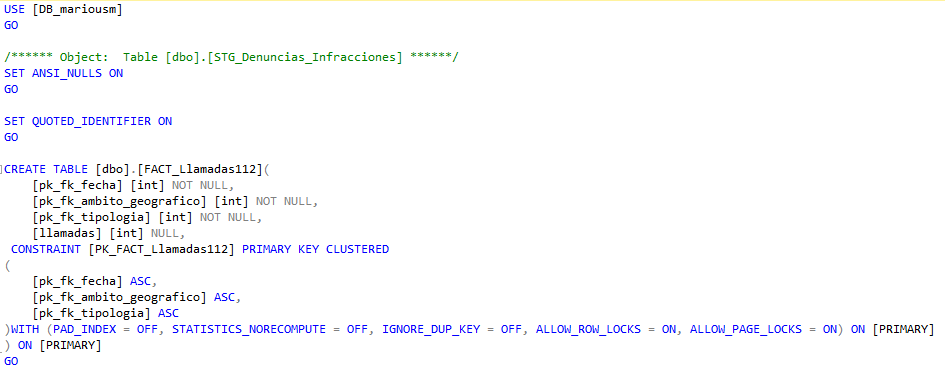


Ilustración 14 - FACT\_Llamadas112.

#### Tabla hecho FACT\_Mediciones

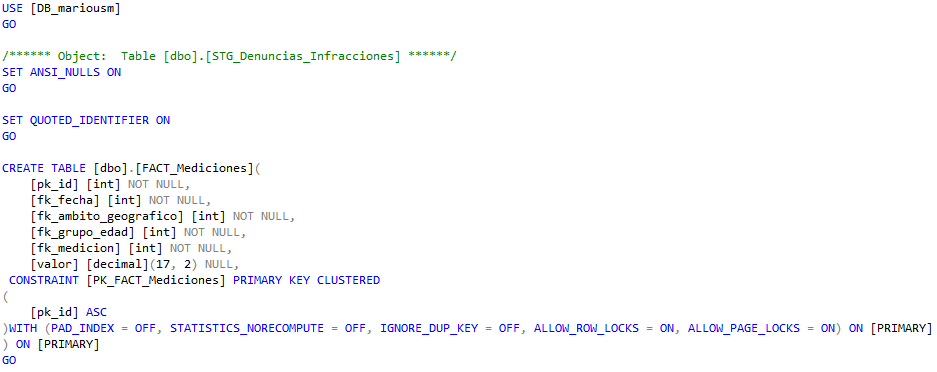


Ilustración 15 - FACT\_Mediciones.

Realizamos los alter table de las tablas de hechos:

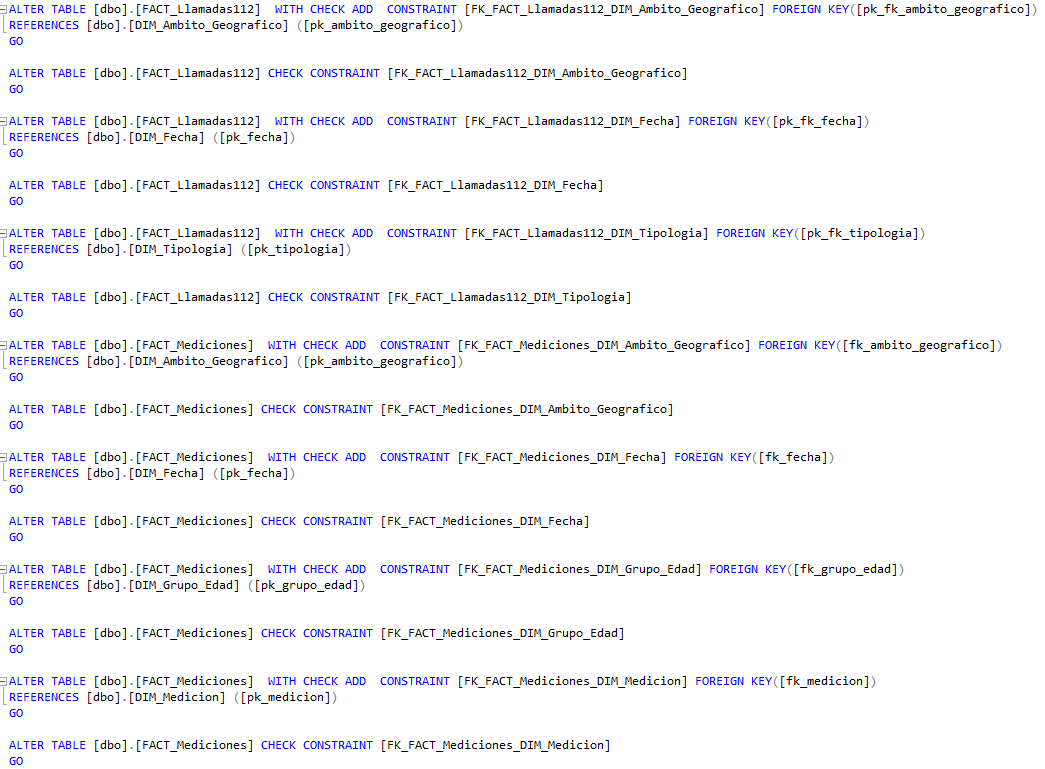


Ilustración 16 - Alter table hechos.

Comprobamos que se han creado todas las tablas correspondientes:



Ilustración 17 - 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

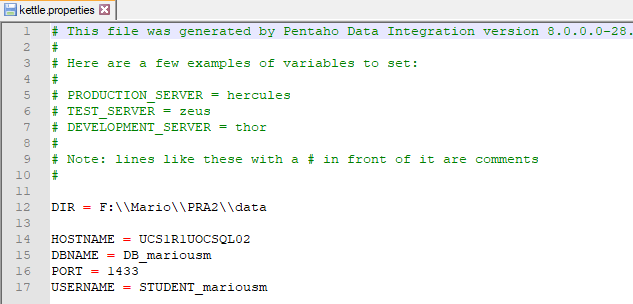


Ilustración 18 - Variables de entorno.

### 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:

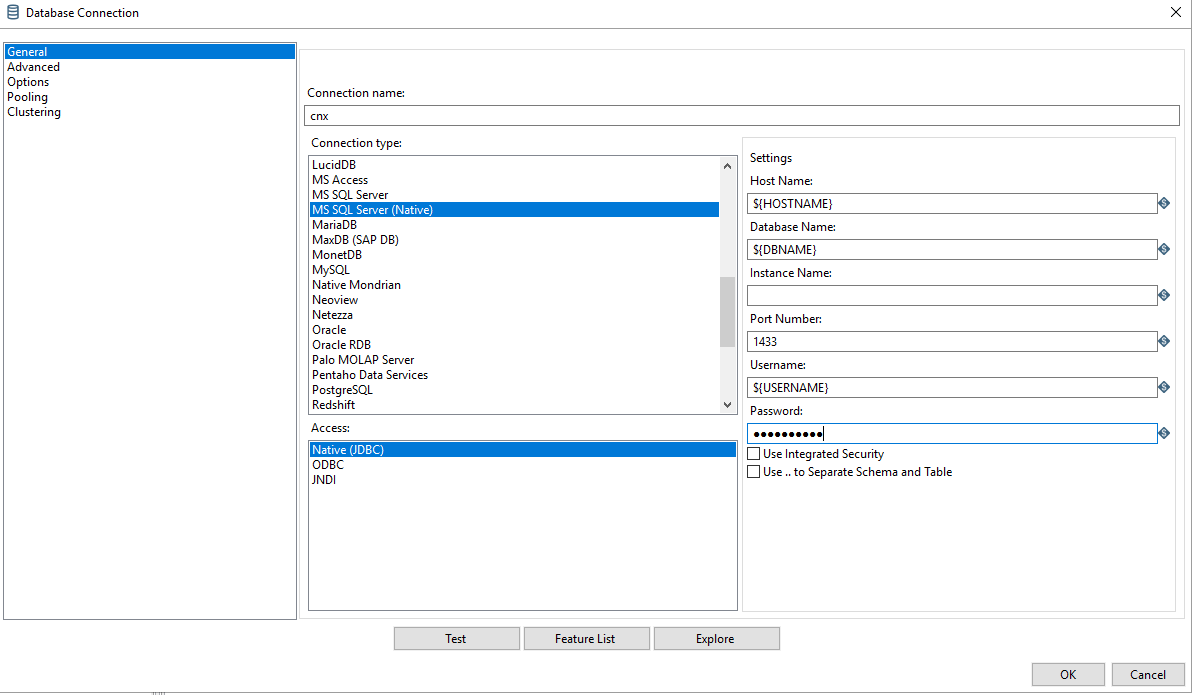


Ilustración 19 - Conexión a la base de datos.

### 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:

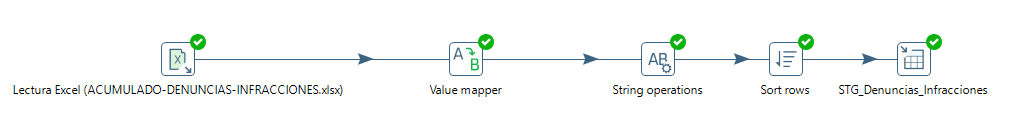


Ilustración 20 - IN\_DENUNCIAS\_INFRACCIONES.

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:

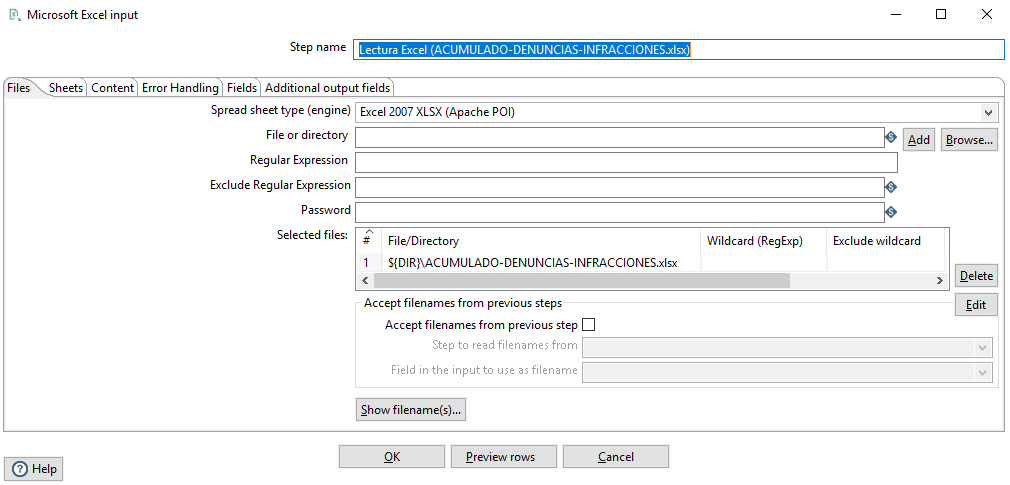


Ilustración 21 - Lectura IN\_DENUNCIAS\_INFRACCIONES.

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:

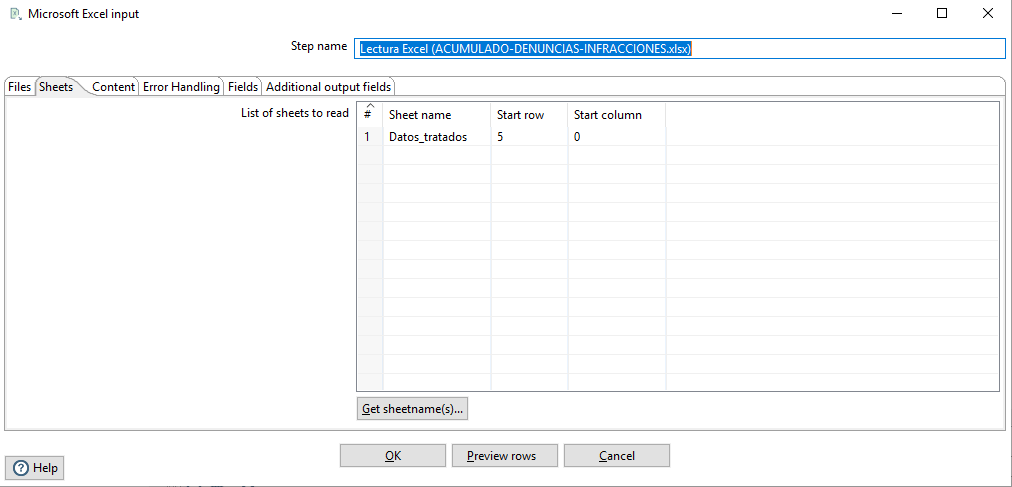


Ilustración 22 - Lectura IN\_DENUNCIAS\_INFRACCIONES.

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

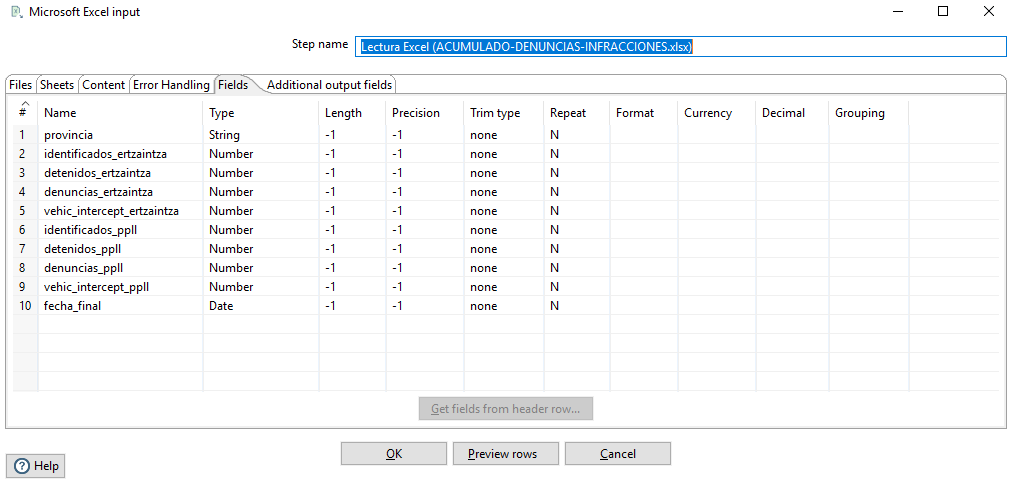


Ilustración 23 - Lectura IN\_DENUNCIAS\_INFRACCIONES.

#### 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:

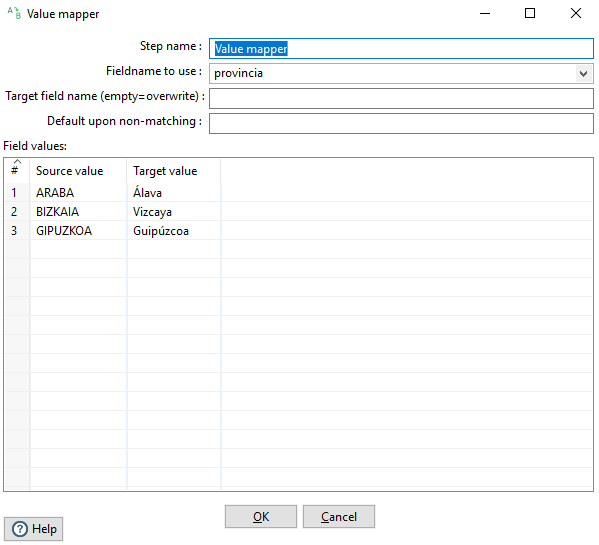


Ilustración 24 - Mapeo Valores IN\_DENUNCIAS\_INFRACCIONES.

#### 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:

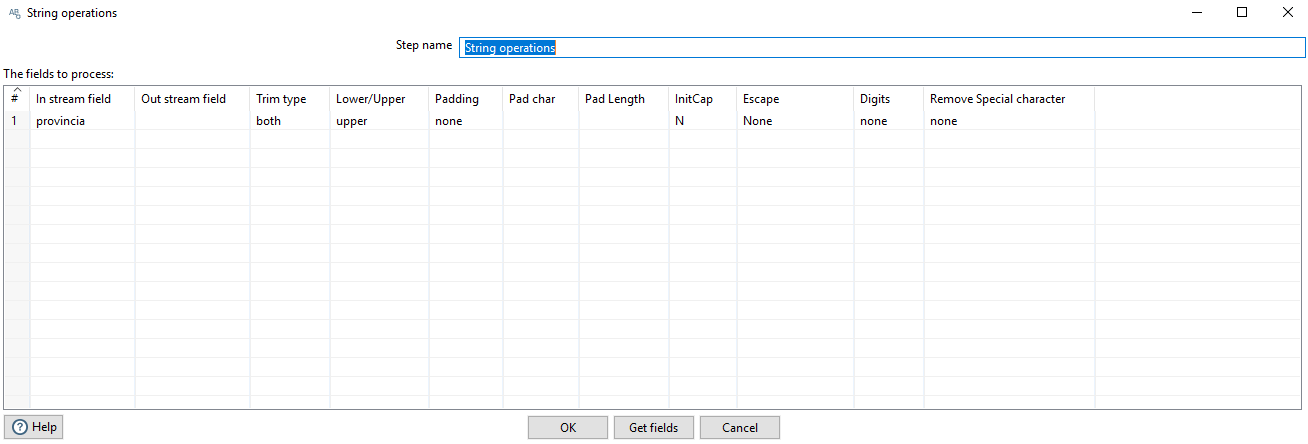


Ilustración 25 - Normalización Strings IN\_DENUNCIAS\_INFRACCIONES.

#### Ordenación

Posteriormente, ordenamos todos los campos de forma ascendente:

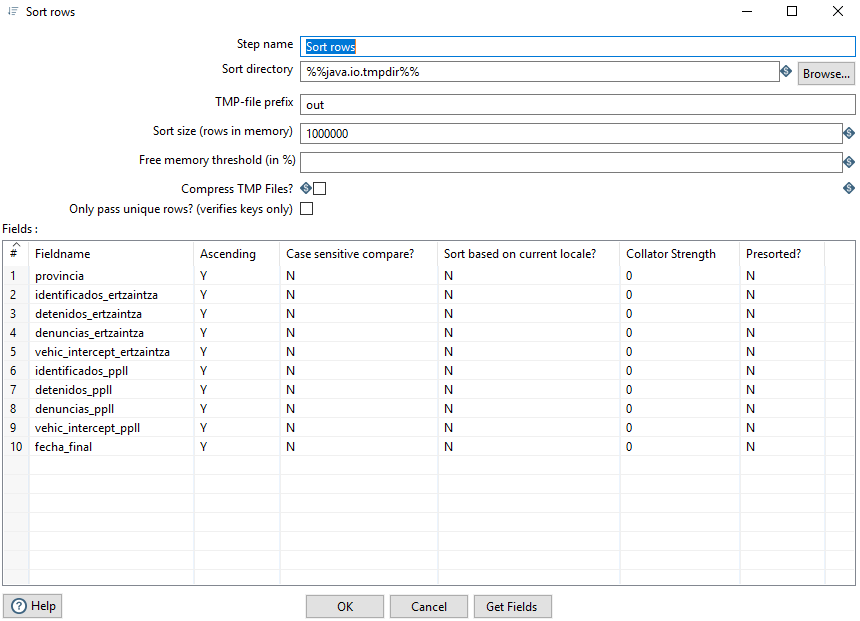


Ilustración 26 - Ordenación IN\_DENUNCIAS\_INFRACCIONES.

#### 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:

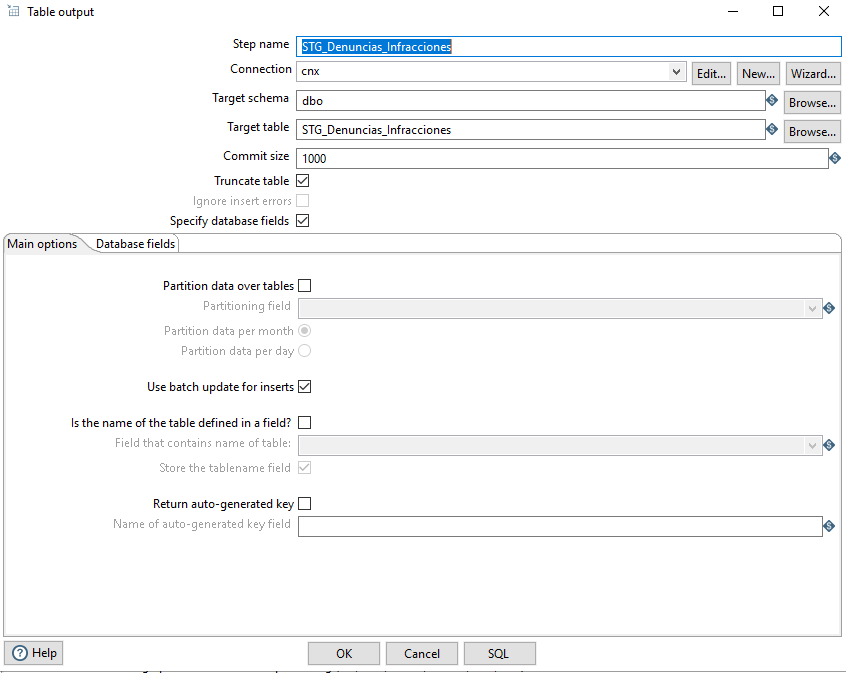


Ilustración 27 - Guardado IN\_DENUNCIAS\_INFRACCIONES.

Al ejecutar la anterior transformación obtenemos las siguiente métricas:

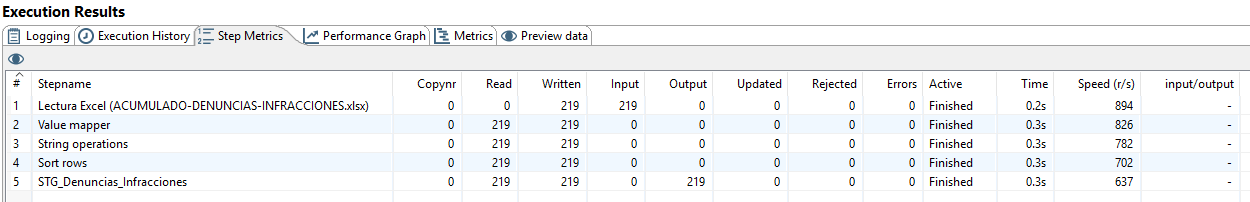


Ilustración 28 - Métricas IN\_DENUNCIAS\_INFRACCIONES.

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:

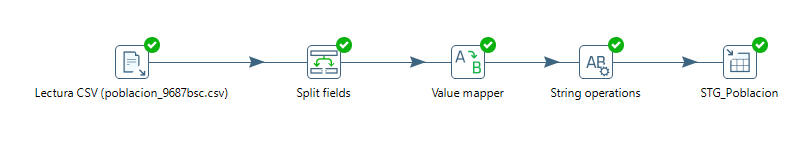


Ilustración 29 - IN\_POBLACION.

#### 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 “;”:

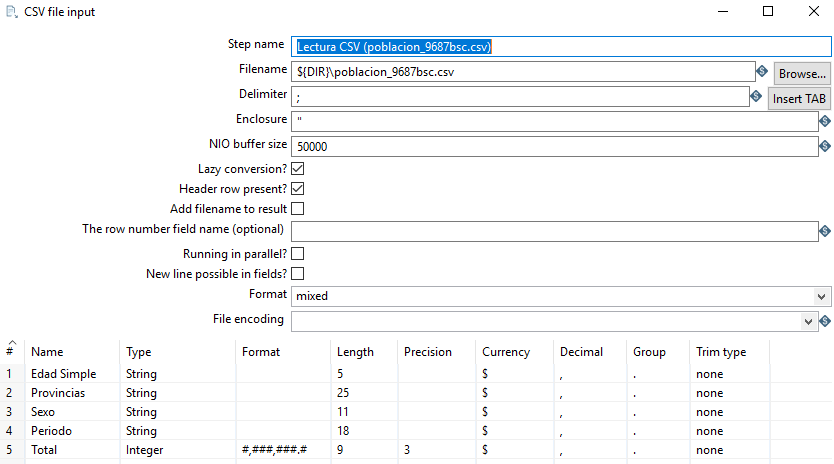


Ilustración 30 - Lectura IN\_POBLACION.

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:

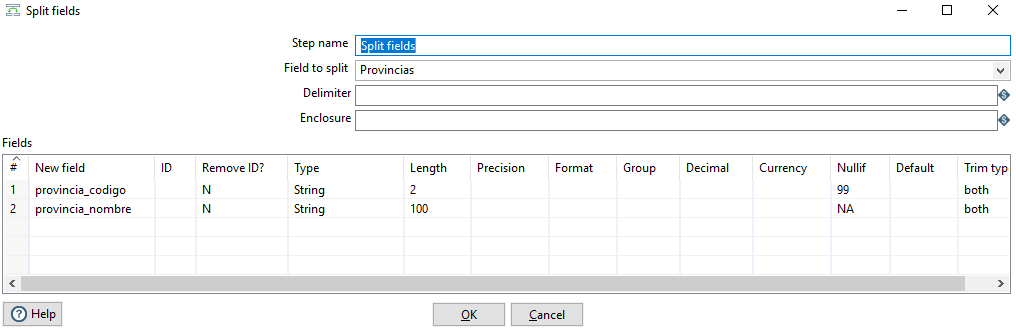


Ilustración 31 - Separación Campos IN\_POBLACION.

#### 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:

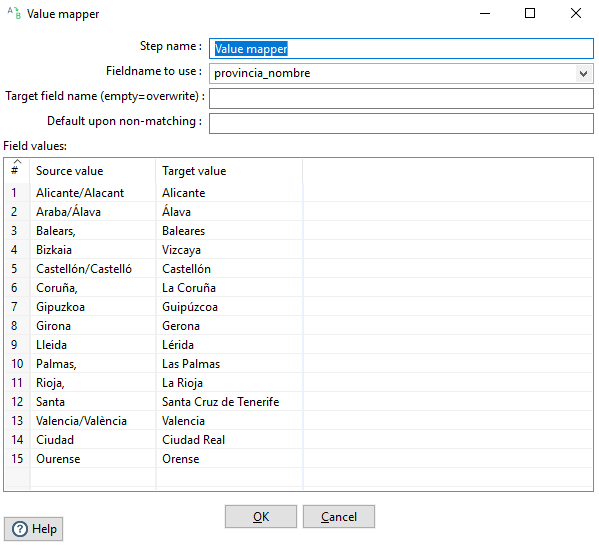


Ilustración 32 - Mapeo Valores IN\_POBLACION.

#### 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:

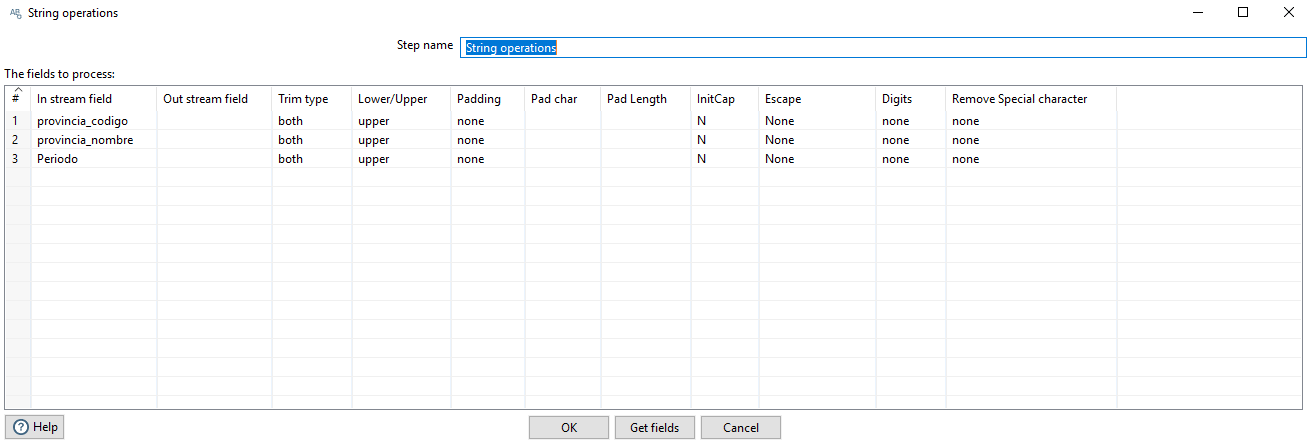


Ilustración 33 - Normalización Strings IN\_POBLACION.

#### Guardado

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

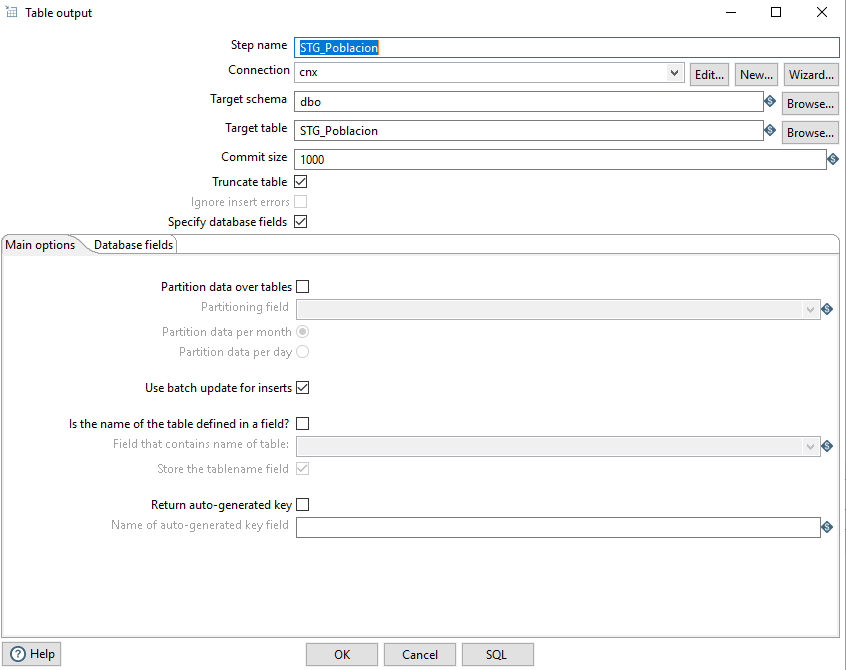


Ilustración 34 - Guardado IN\_POBLACION.

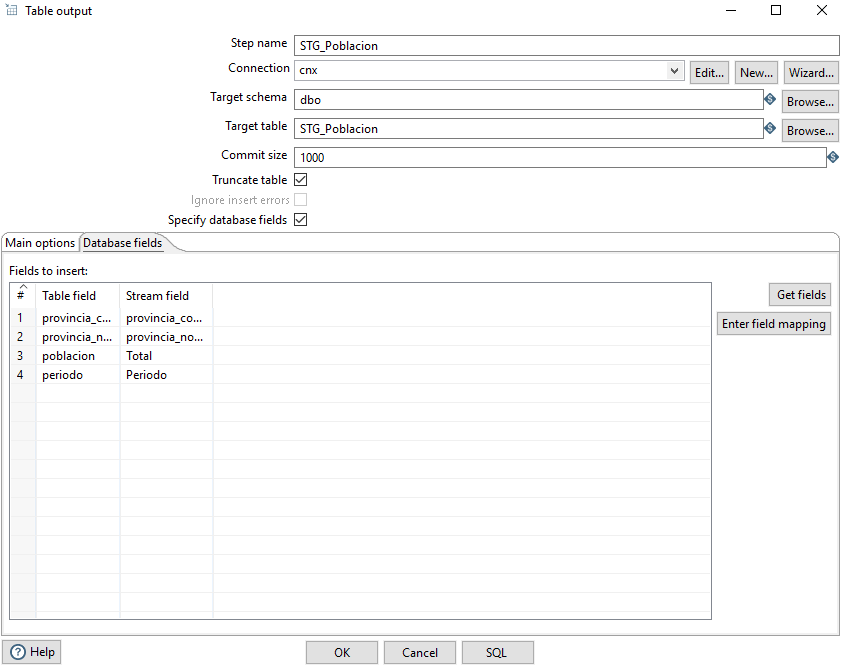


Ilustración 35 - Guardado IN\_POBLACION.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

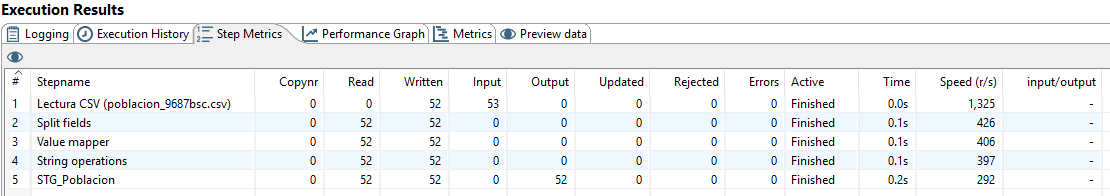


Ilustración 36 - Métricas IN\_POBLACION.

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:

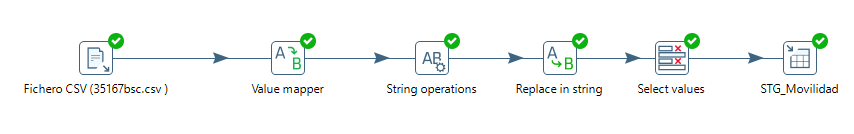


Ilustración 37 - IN\_MOVILIDAD.

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 “,”:

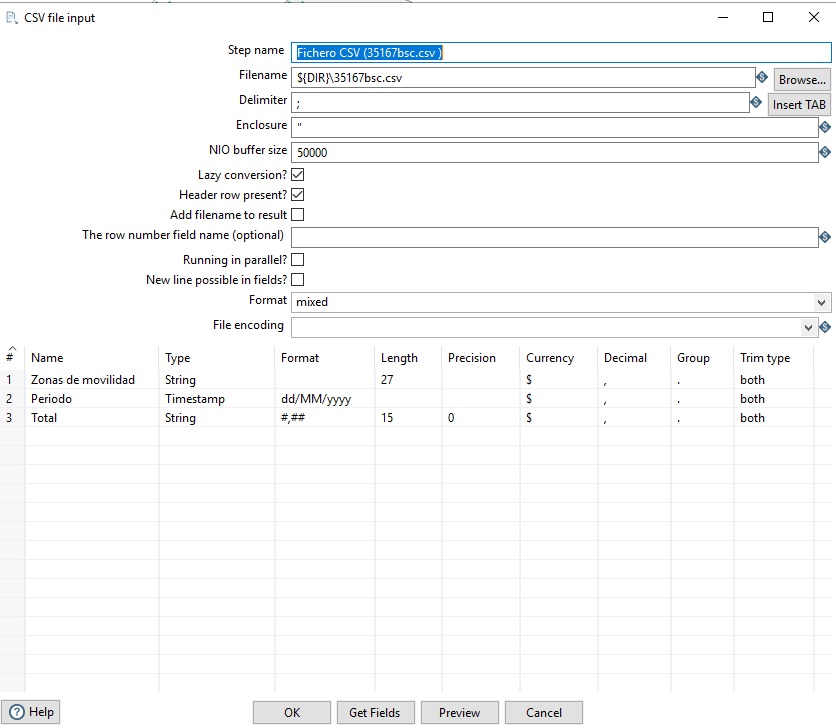


Ilustración 38 - Lectura IN\_MOVILIDAD.

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.

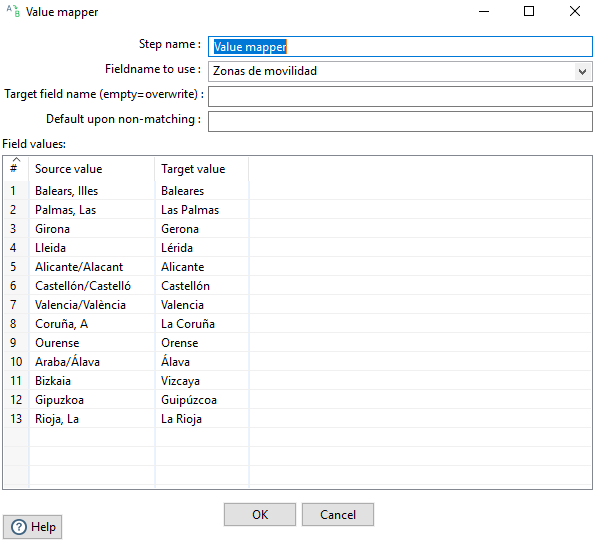


Ilustración 39 - Mapeo Valores IN\_MOVILIDAD.

#### 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:

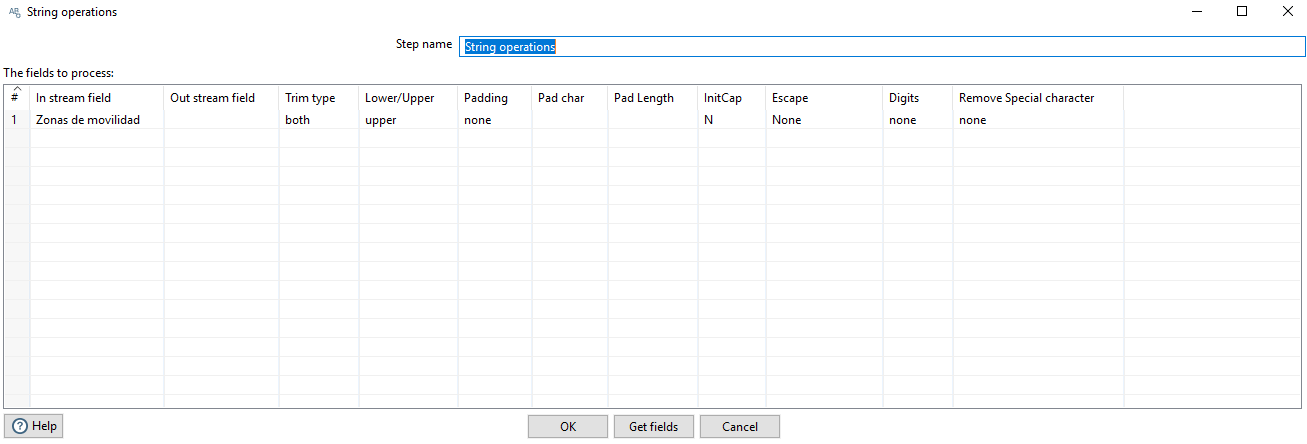


Ilustración 40 - Normalización IN\_MOVILIDAD.

#### 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:

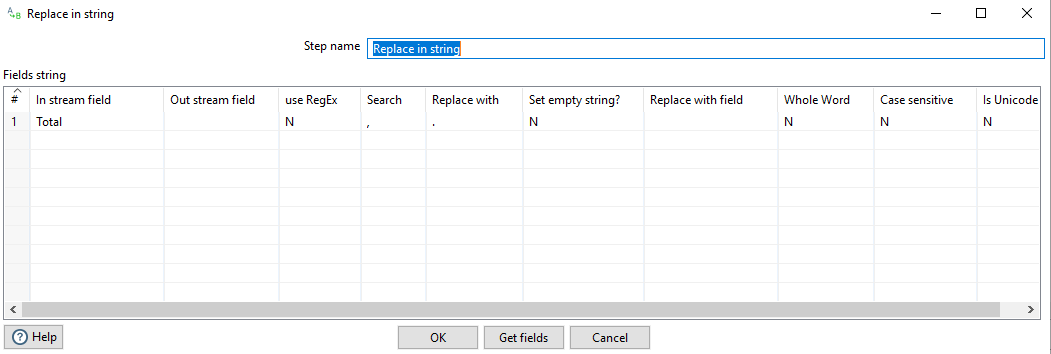


Ilustración 41 - Replace IN\_MOVILIDAD.

#### 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”:

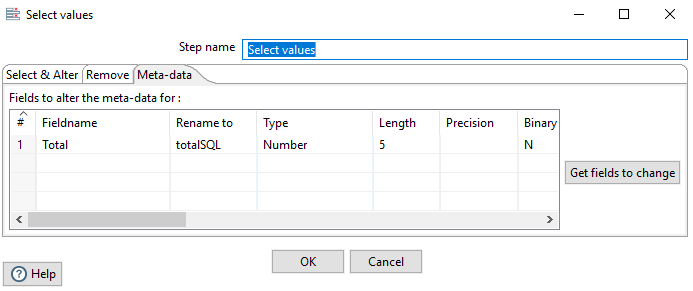


Ilustración 42 - Select Values IN\_MOVILIDAD.

#### 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:

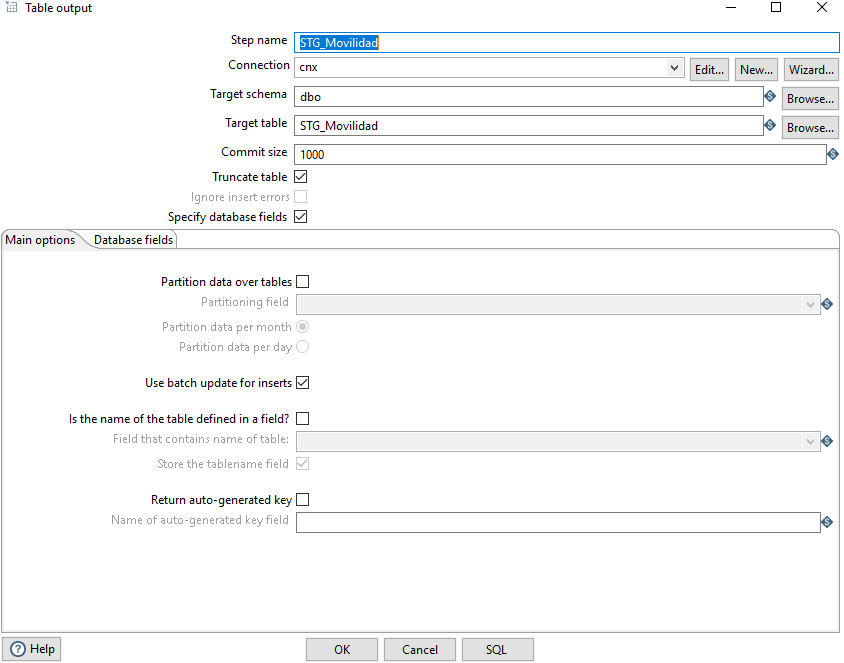


Ilustración 43 - Guardado IN\_MOVILIDAD.

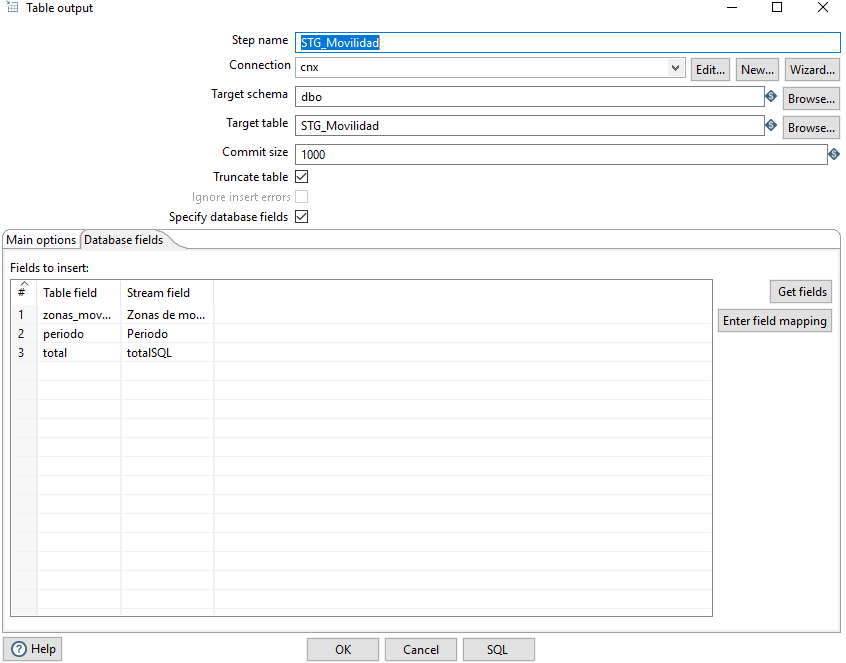


Ilustración 44 - Guardado IN\_MOVILIDAD.

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

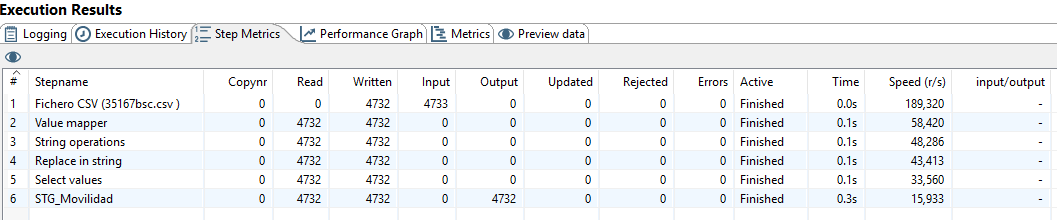


Ilustración 45 - Métricas IN\_MOVILIDAD.

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:

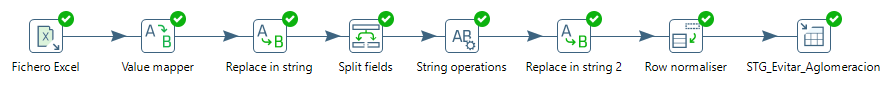


Ilustración 46 - IN\_AGLOMERACION.

#### 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:

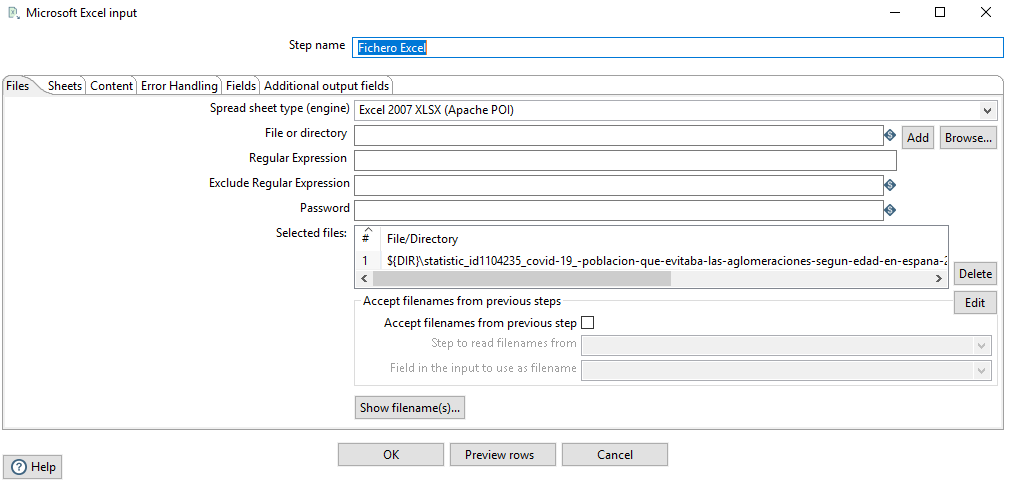


Ilustración 47 - Lectura IN\_AGLOMERACION.

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:

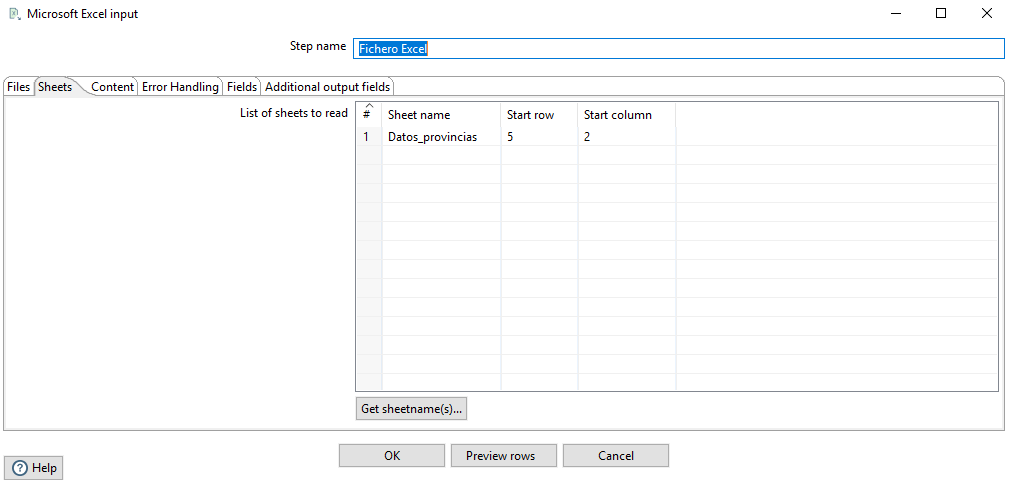


Ilustración 48 - Lectura IN\_AGLOMERACION.

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

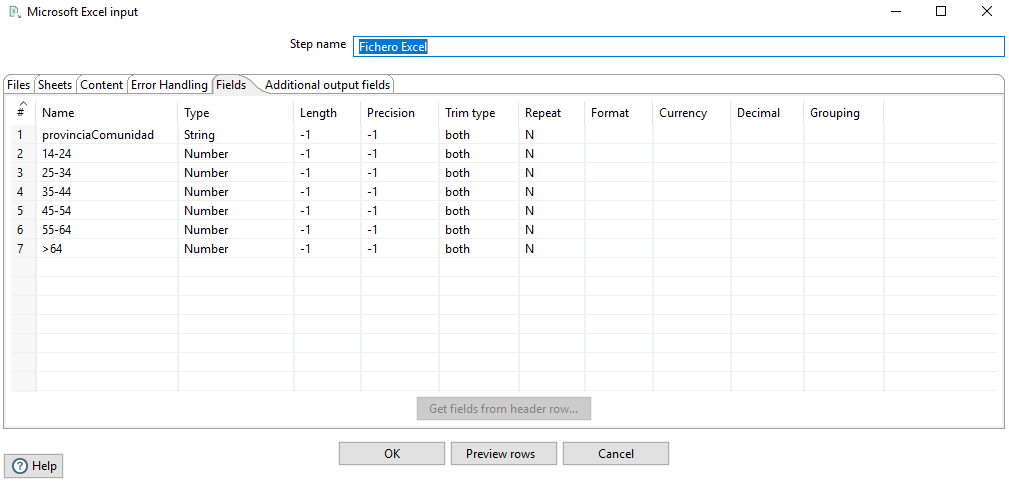


Ilustración 49 - Lectura IN\_AGLOMERACIONES.

#### 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:

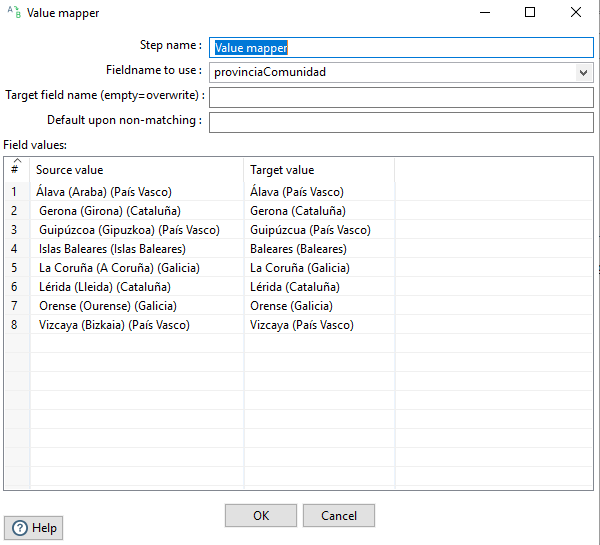


Ilustración 50 - Mapeo Valores IN\_AGLOMERACION.

#### 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:

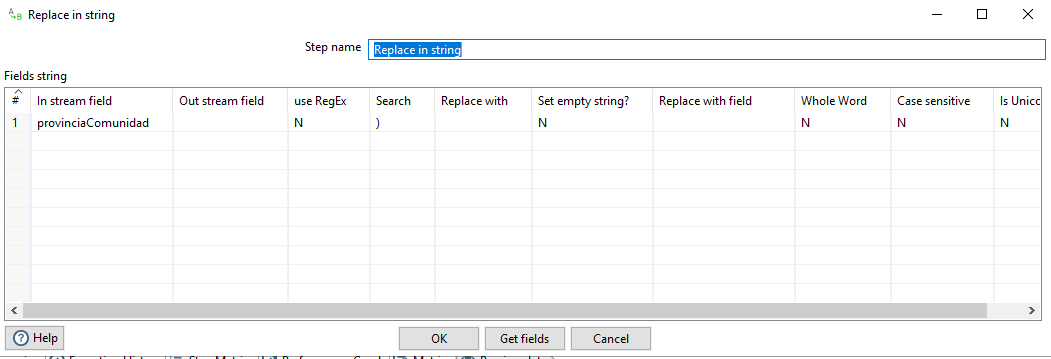


Ilustración 51 - Replace IN\_AGLOMERACION.

#### 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.

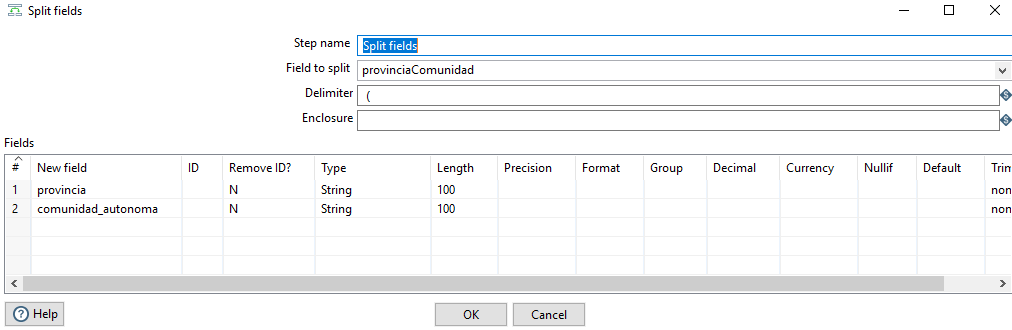


Ilustración 52 - Split IN\_AGLOMERACION.

#### 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:

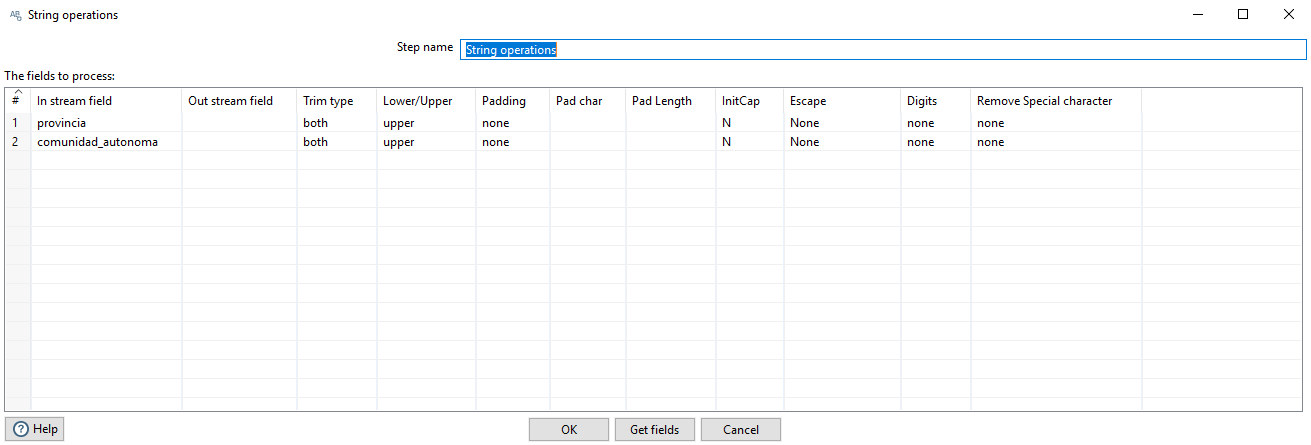


Ilustración 53 - Normalización Strings IN\_AGLOMERACION.

#### 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:

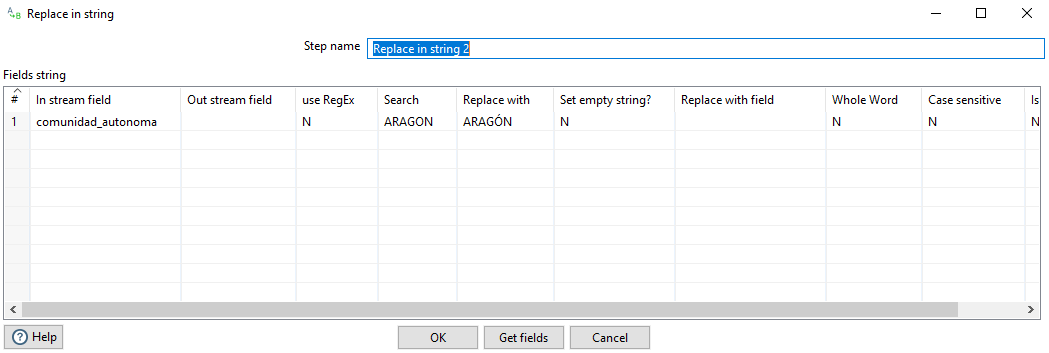


Ilustración 54 - Replace IN\_AGLOMERACION.

#### 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:

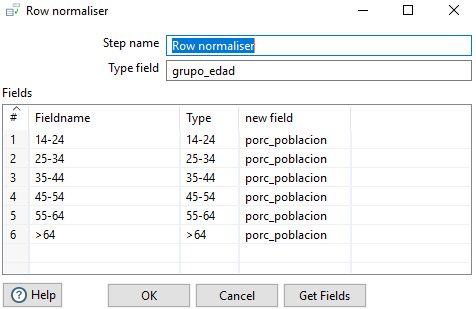


Ilustración 55 - Normalización Filas IN\_AGLOMERACION.

#### 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:

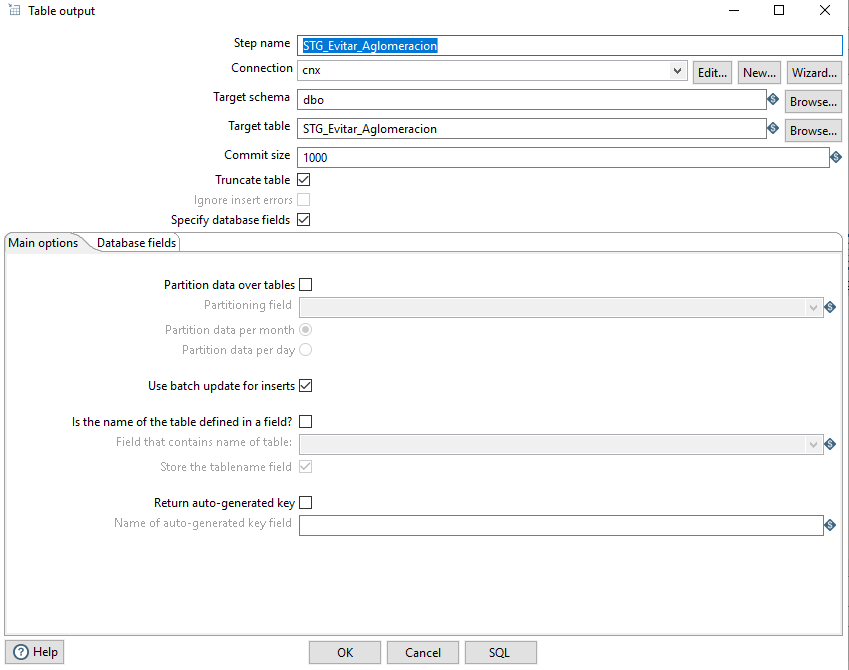


Ilustración 56 - Guardado IN\_AGLOMERACIONES.

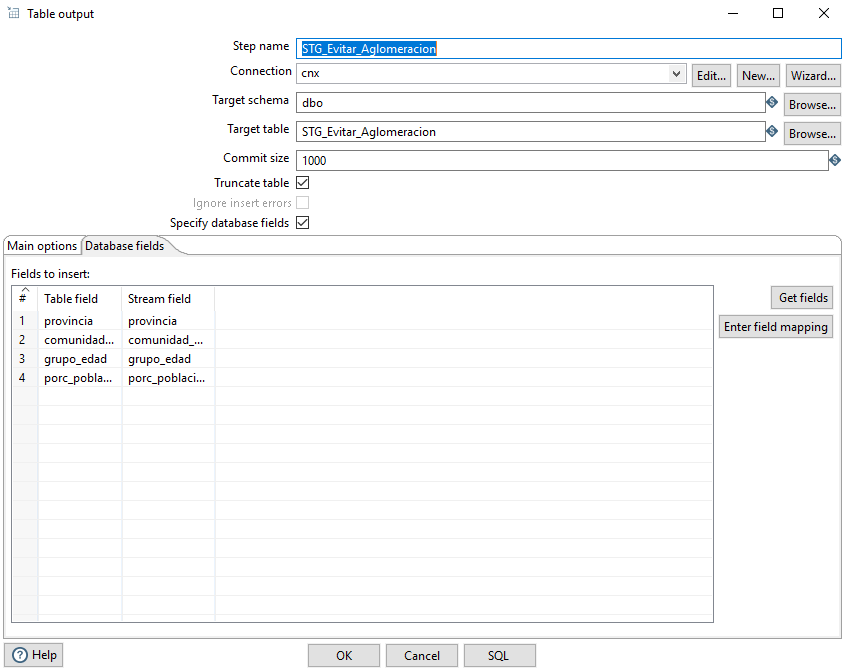


Ilustración 57 - Guardado IN\_AGLOMERACIONES.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

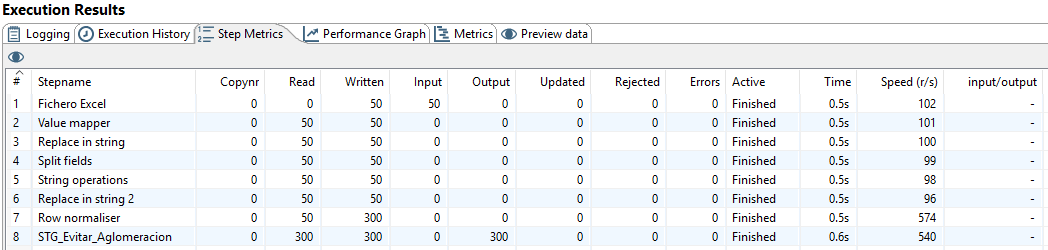


Ilustración 58 - Métricas IN\_AGLOMERACION.

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:

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

Ilustración - IN\_LLAMADAS112.

#### Lectura XML

Lo primero que tenemos que hacer es leer la información que se nos proporciona en el fichero XML. Para ello escribimos el nombre del paso e indicamos el fichero:

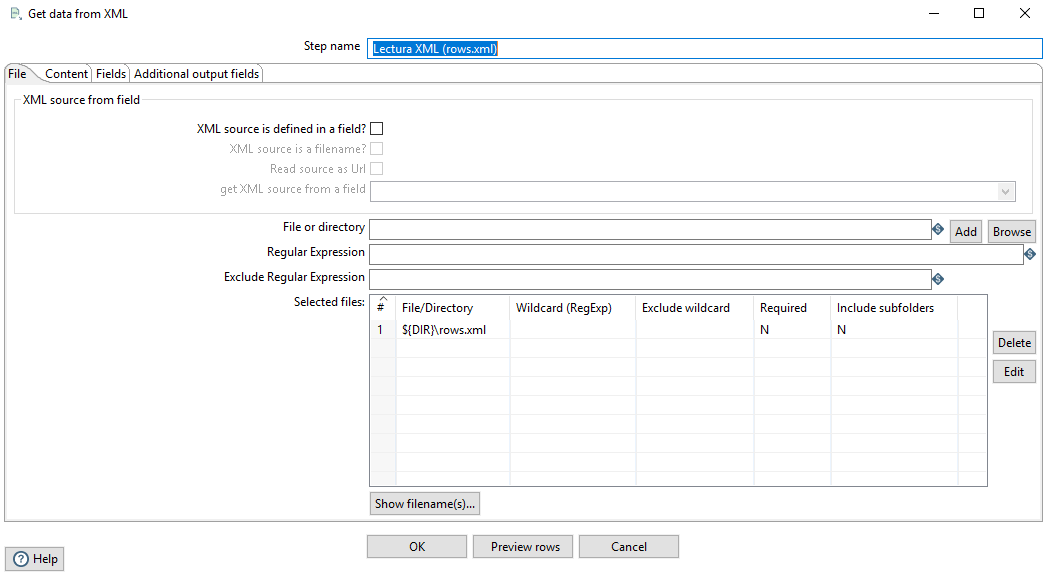


Ilustración - Lectura IN\_LLAMADAS112.

Luego nos dirigimos a la pestaña “Content” y definimos desde qué loop tiene que empezar a leer nuestro fichero XML:

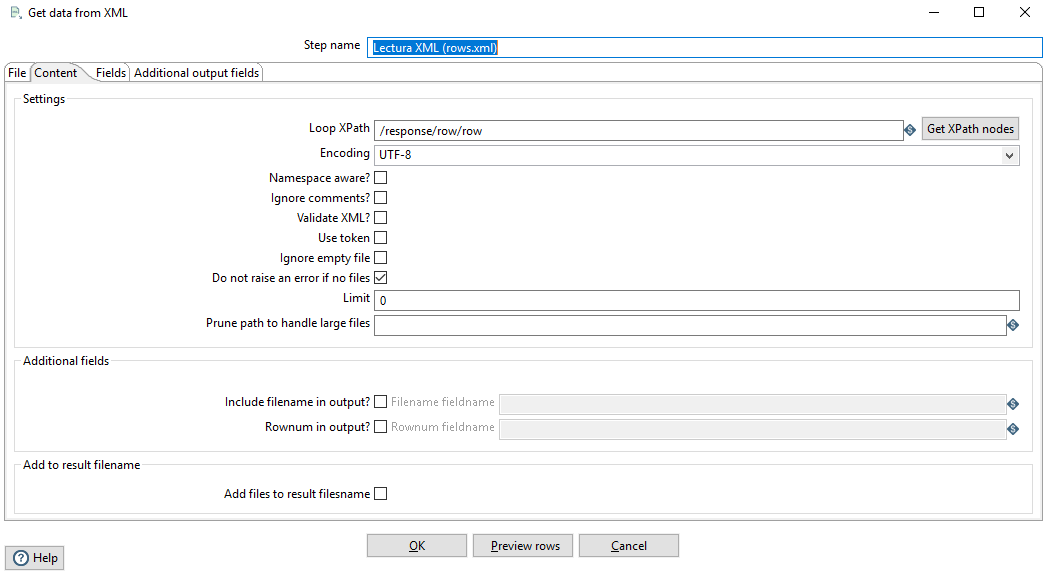


Ilustración - Lectura IN\_LLAMADAS112.

Finalmente, obtenemos los campos y los definimos nosotros de forma manual:

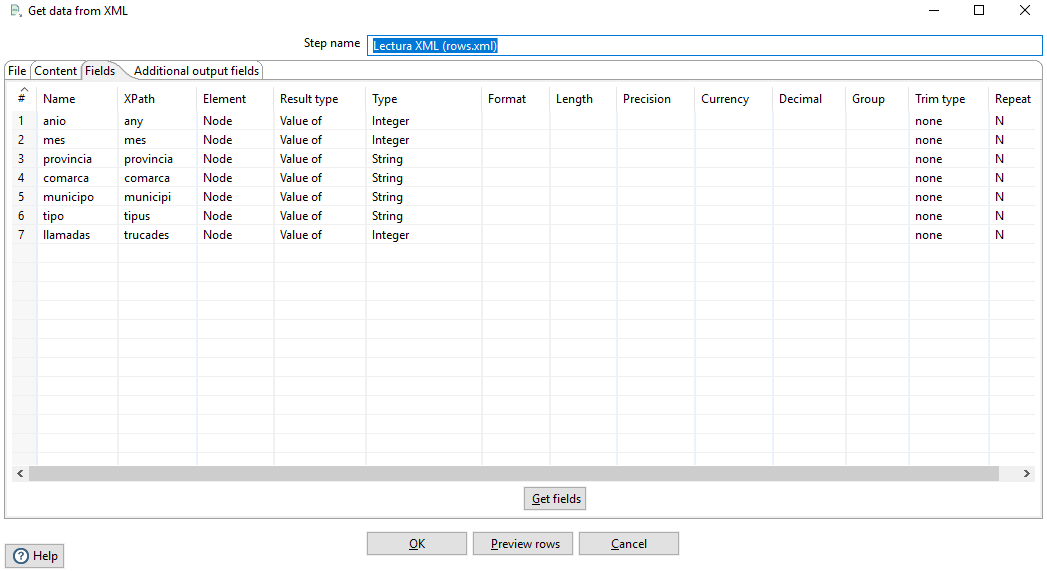


Ilustración - Lectura IN\_LLAMADAS112.

#### Mapeo

Una vez leído los registros tenemos que hacer un cambio de valor de algunas provincias, ya que éstas aparecen en catalán y vamos a mantener en la base de datos solamente la traducción al castellano:

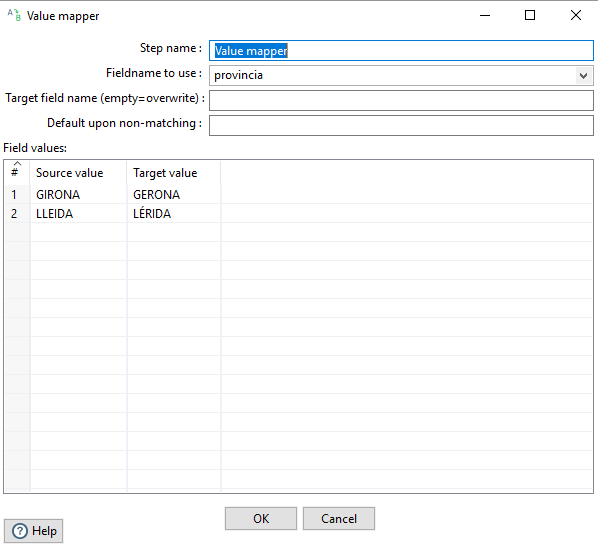


Ilustración - Mapeo Valores IN\_LLAMADAS112.

#### Mapeo

Necesitamos hacer otro mapeo de valores para el campo “tipo” ya que éste representa el motivo de la llamada y sus valores están en catalán. Por lo tanto, realizamos lo mismo que en el paso anterior:

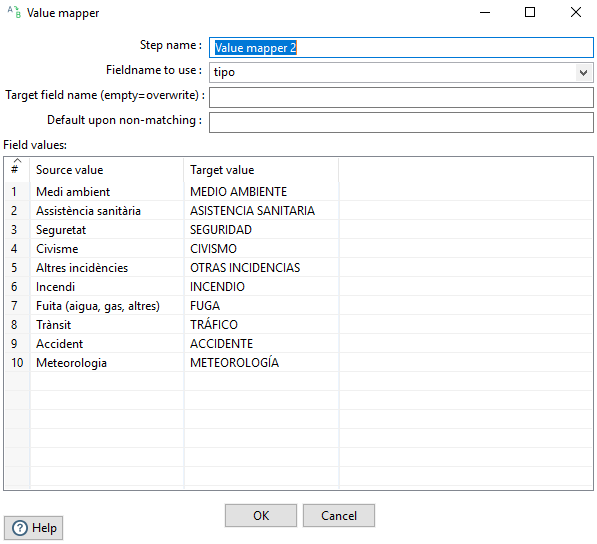


Ilustración - Mapeo Valores IN\_LLAMADAS112.

#### Normalización

Antes de introducir todos los datos a la base de datos, tenemos que normalizar las cadenas de valores, es decir, establecer los campos string a mayúscula y sin espacios al comienzo ni al final:

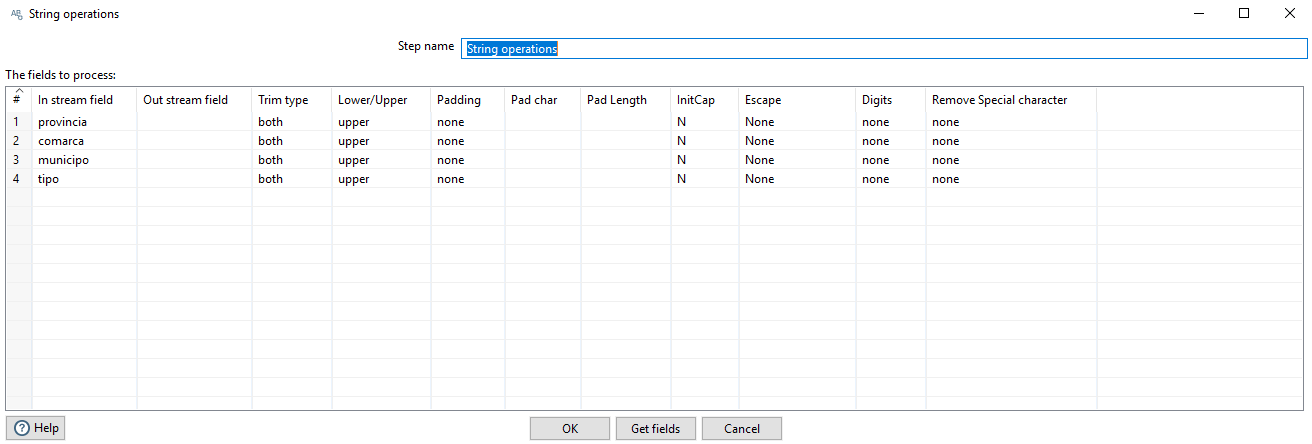


Ilustración - Normalización IN\_LLAMADAS112.

#### Guardado

Finalmente, guardamos los datos en la tabla “STG\_Llamadas112”, indicando que haga un truncate de la tabla y asociamos los campos:

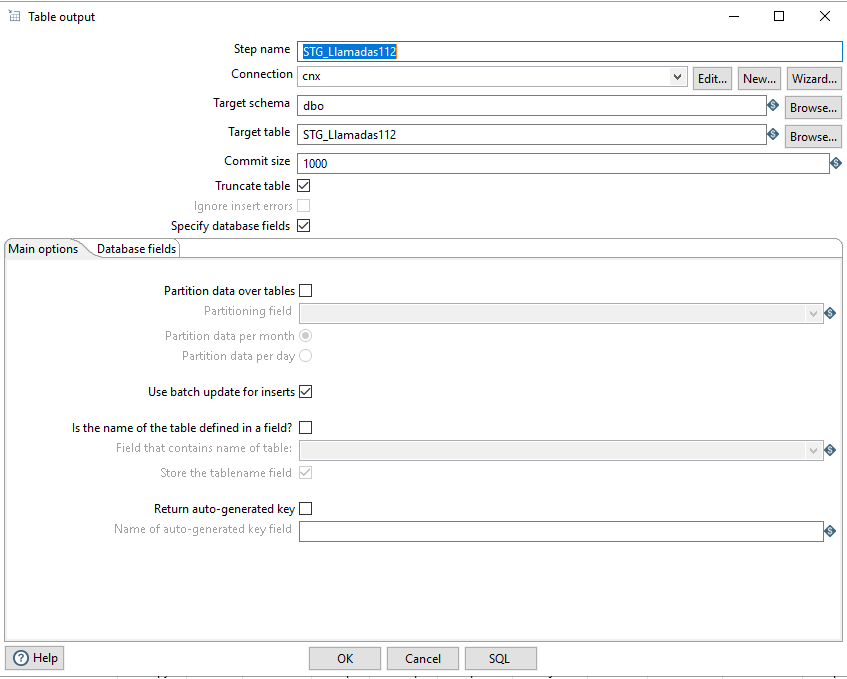


Ilustración - Guardado IN\_LLAMADAS112.

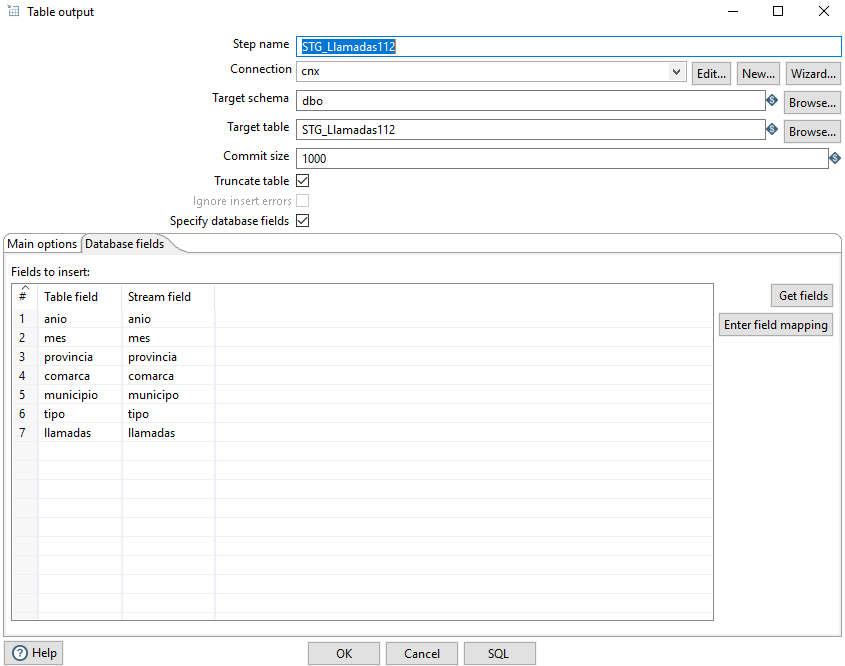


Ilustración - Guardado IN\_LLAMADAS112.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

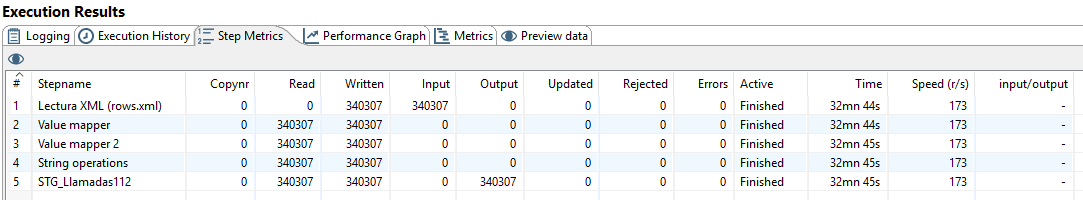


Ilustración - Métricas IN\_LLAMADAS112.

Observamos que tenemos 340307 registros leídos y almacenamos el mismo número de registros, por lo que la información es correcta.

### Transformación IN\_FECHAS

## Bloque TR Dimensiones

Una vez que hemos almacenado toda la información en la base de datos gracias a las tablas intermedias, ahora vamos a hacer uso de estos datos para crear las diferentes dimensiones de nuestro modelo.

### Transformación TR\_DIM\_GRUPO\_EDAD

La primera transformación que vamos a realizar se llama “TR\_DIM\_GRUPO\_EDAD”, su objetivo es almacenar los diferentes grupos de edad para así hacer uso de ellos en el hecho de mediciones, el resultado de esta transformación va a ser los datos almacenados en “DIM\_Grupo\_Edad”.

La transformación nos ha quedado de la siguiente forma:

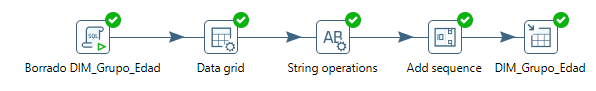


Ilustración - TR\_DIM\_GRUPO\_EDAD.

#### Borrado

Lo primero que debemos de hacer es el borrado de los registros que contenía la dimensión, para ello escribimos directamente la sentencia SQL y la ejecutamos:

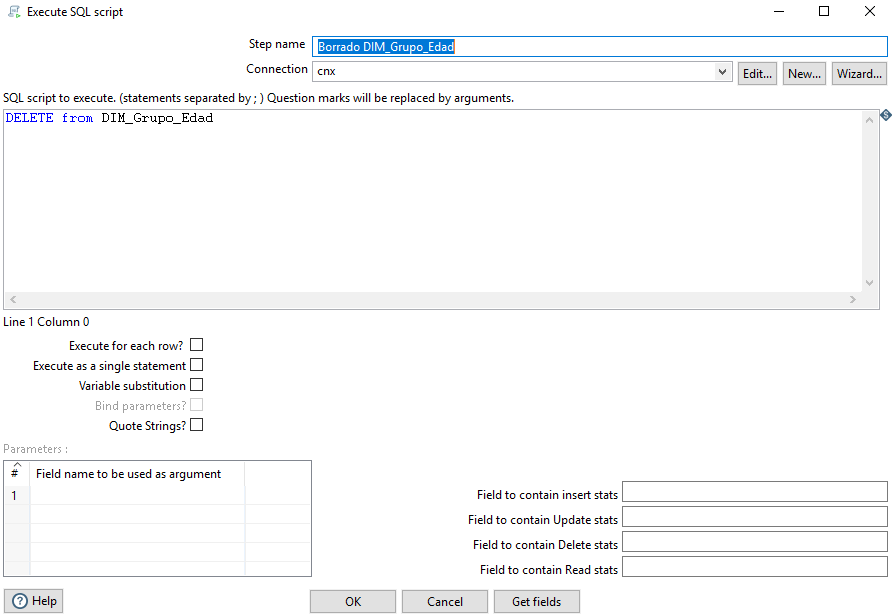


Ilustración - Borrado TR\_DIM\_GRUPO\_EDAD.

#### Grid

Puesto que la información de esta dimensión es fija y tiene tan solo 7 registros, nos resulta más fácil almacenar la información a partir de un grid (ya que en el enunciado de la práctica no se indica que no se pueda hacer uso de ellos), es por ello que hemos definido el siguiente grid:

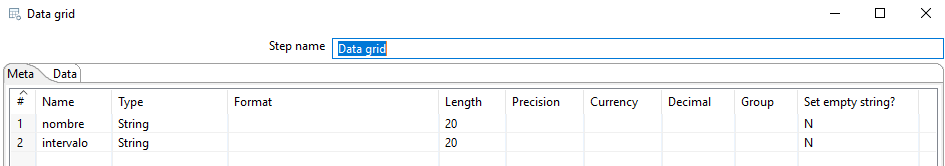


Ilustración - Grid TR\_DIM\_GRUPO\_EDAD.

Una vez definidos los campos, introducimos los registros de forma manual. Cabe destacar que vamos a tener un registro con valores “NA”, esto significa que está dimensión no va a aplicar para calcular ciertas medidas:

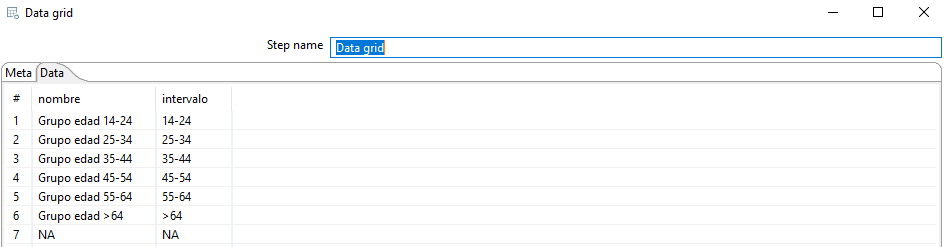


Ilustración - Grid TR\_DIM\_GRUPO\_EDAD.

#### Normalización

Normalizamos tanto el nombre como el intervalo para que estén en mayúsculas y no tengan espacios ni al principio ni al final:

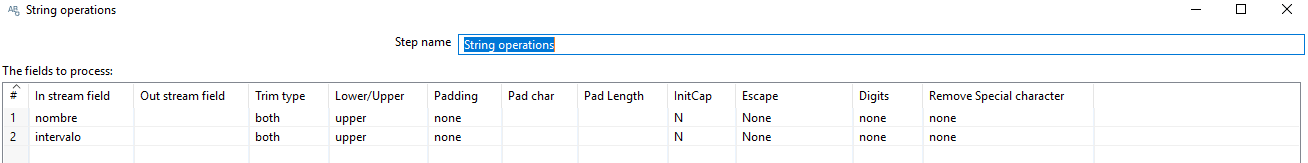


Ilustración - Normalización TR\_DIM\_GRUPO\_EDAD.

#### Secuenciación

Otro aspecto a destacar es que las dimensiones ya tienen claves primarias, por lo tanto vamos a definir la misma como un autonumérico incrementándose de uno en uno:

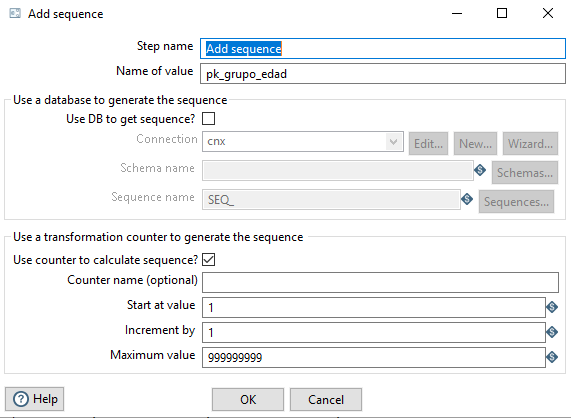


Ilustración - Secuenciación TR\_DIM\_GRUPO\_EDAD.

#### Guardado

Finalmente, realizamos el guardado en la dimensión indicando la tabla destino como “DIM\_Grupo\_Edad” y asociamos los atributos:

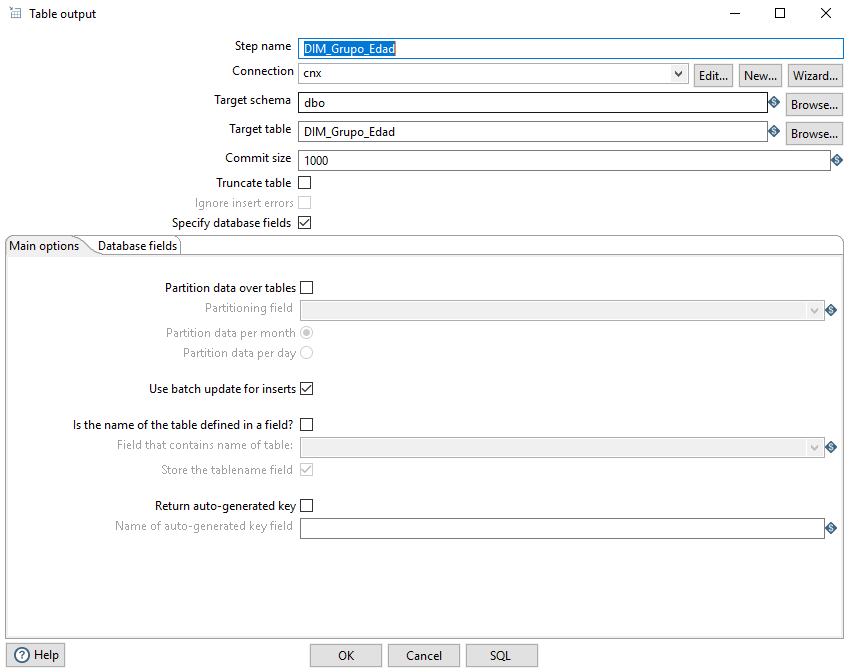


Ilustración - Guardado TR\_DIM\_GRUPO\_EDAD.

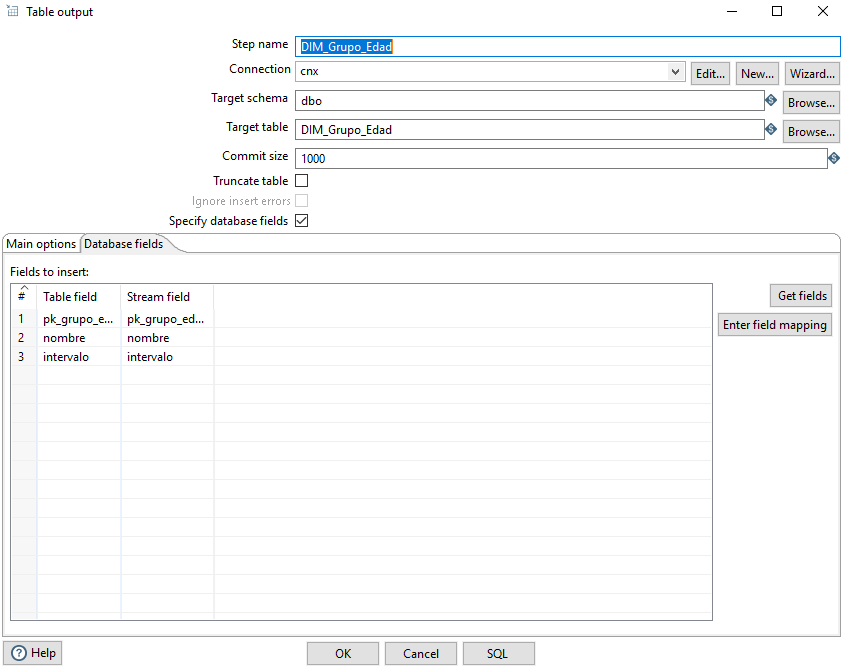


Ilustración - Guardado TR\_DIM\_GRUPO\_EDAD.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

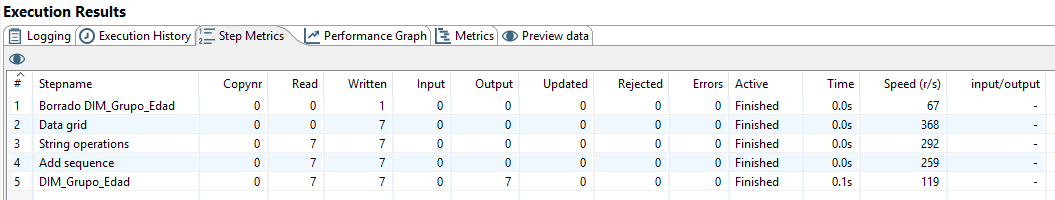


Ilustración - Métricas TR\_DIM\_GRUPO\_EDAD.

Como podemos observar generamos los 7 registros creados manualmente y guardamos todos en la base de datos.

### Transformación TR\_DIM\_Medicion

La segunda transformación que vamos a realizar se llama “TR\_DIM\_GRUPO\_EDAD”, su objetivo es almacenar las diferentes medidas que vamos a usar en la tala de hechos mediciones, el resultado de esta transformación va a ser los datos almacenados en “DIM\_Medicion”.

La transformación nos ha quedado de la siguiente forma:

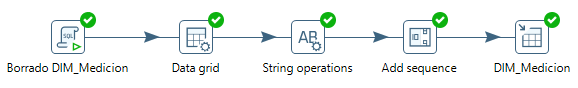


Ilustración - TR\_DIM\_MEDICION.

#### Borrado

Lo primero que debemos de hacer es borrar todos los registros que hay en la tabla, por si había previos:

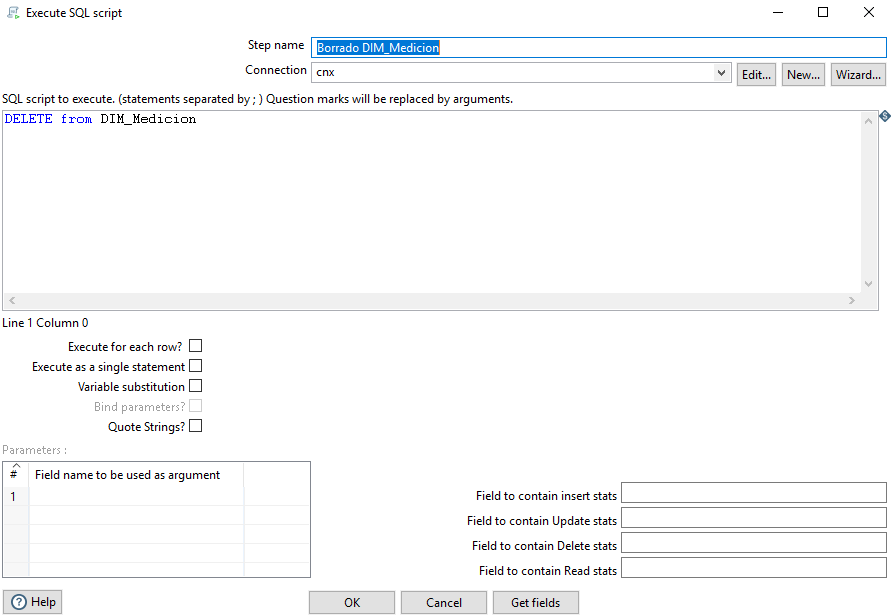


Ilustración - Borrado TR\_DIM\_MEDICION.

#### Grid

Como los datos de esta dimensión no se encuentran en ningún fichero, la única solución que tenemos es introducirlos de forma manual, es por ello que hemos creado el siguiente grid:

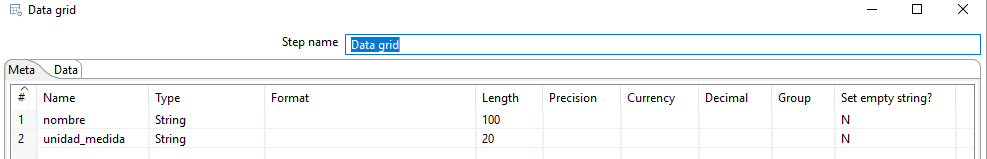


Ilustración - Grid TR\_DIM\_DIM\_MEDICION.

Una vez definidos los campos, introducimos los registros de forma manual:

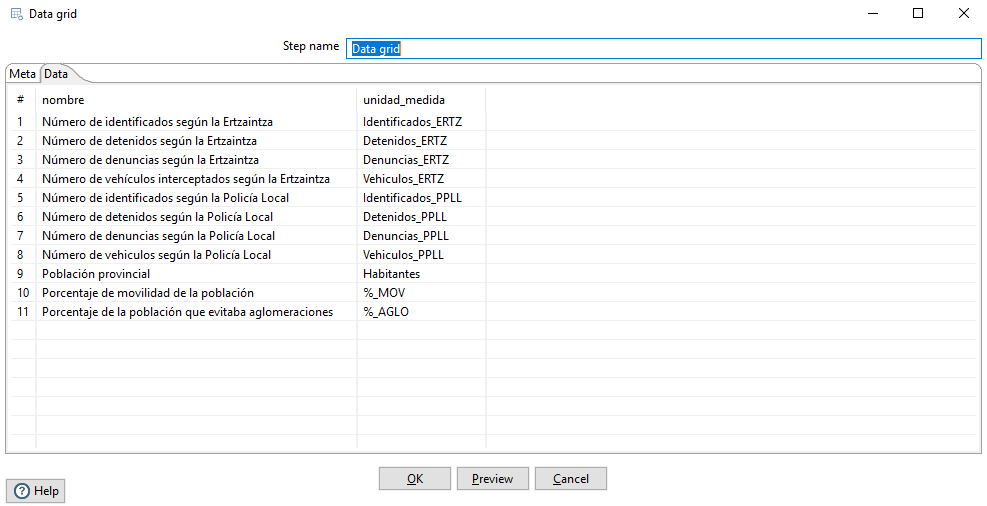


Ilustración - Grid TR\_DIM\_MEDICION.

#### Normalización

Normalizamos tanto el nombre como la unidad de mediada, para que así todo esté en mayúsculas y sin espacios:

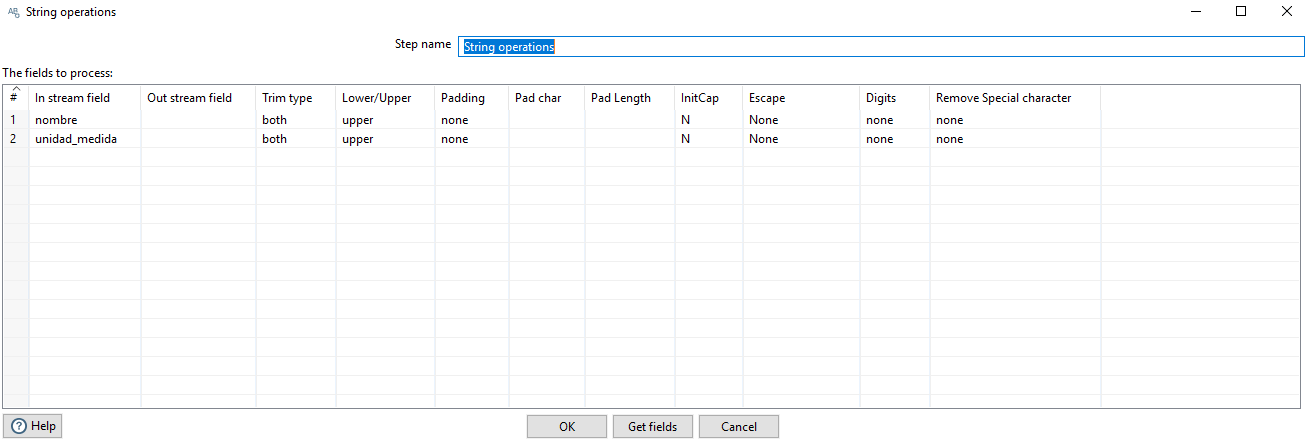


Ilustración - Normalización TR\_DIM\_MEDICION.

#### Secuenciación

Al igual que sucedía antes, las tablas dimensiones ya tienen claves primarias, por lo que tenemos que definir la misma para esta dimensión:

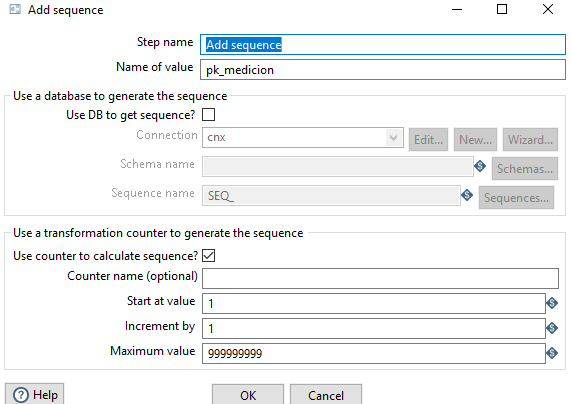


Ilustración - Secuenciación TR\_DIM\_MEDICION.

#### Guardado

Una vez que ya tenemos todos los datos de forma correcta, procedemos a realizar el guardado en la tabla correspondiente, en nuestro caso “DIM\_Medicion”:

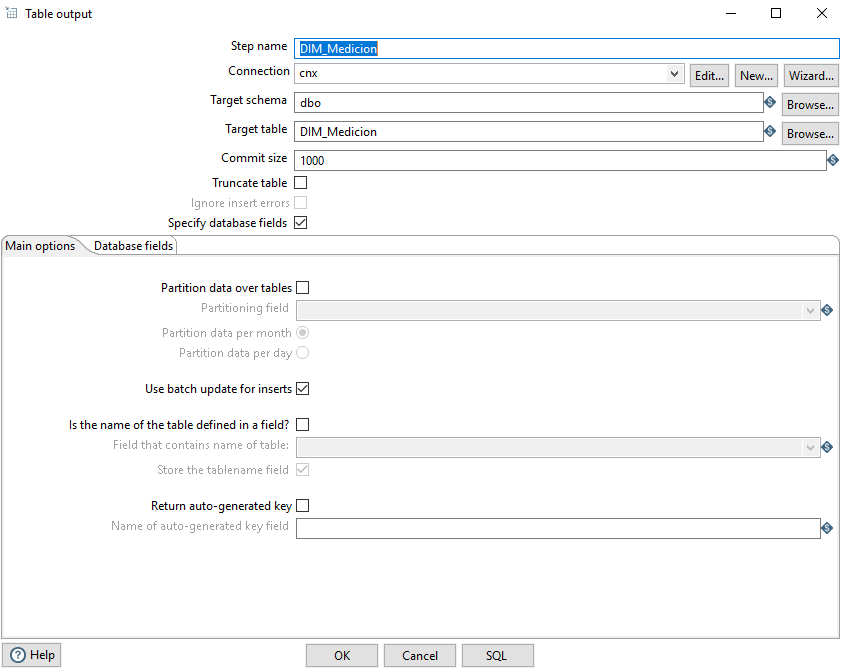


Ilustración - Guardado TR\_DIM\_MEDICION.

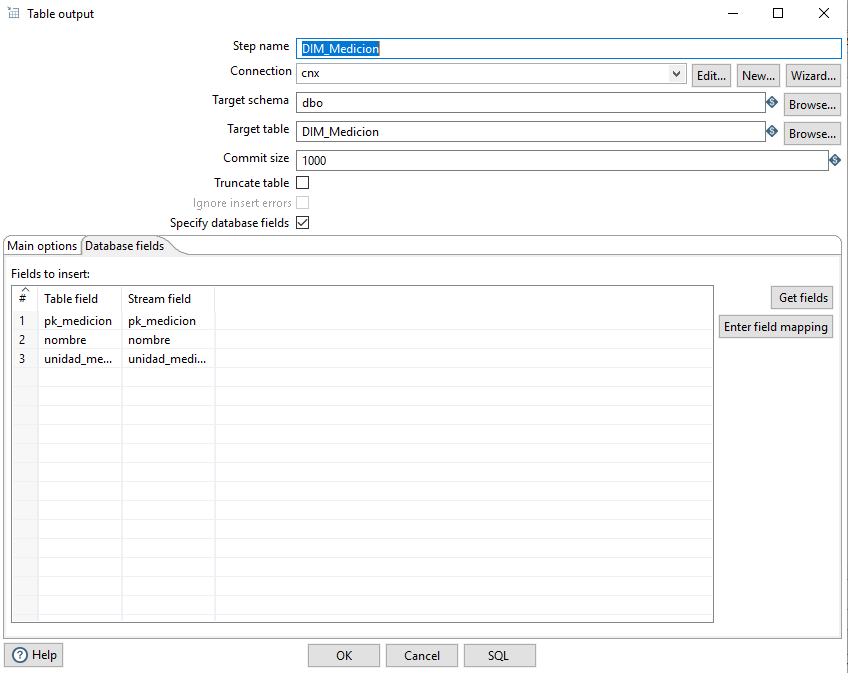


Ilustración - Guardado TR\_DIM\_MEDICION.

Al ejecutar la anterior transformación nos proporciona las siguientes métricas:

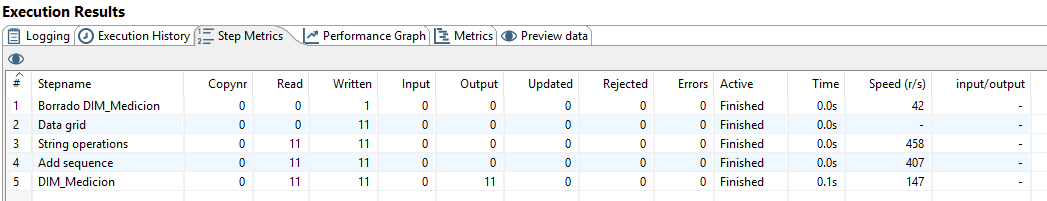


Ilustración - Métricas TR\_DIM\_MEDICIONES.

Como podemos observar generamos los 11 registros de forma manual y los guardamos perfectamente en la base de datos.

### Transformación TR\_DIM\_TIPOLOGIA

La tercera transformación de este bloque se corresponde con “TR\_DIM\_TIPOLOGIA”, su objetivo es almacenar las diferentes tipologías en las llamadas al 112 en Cataluña, el resultado de esta transformación va a ser los datos almacenados en “DIM\_Tipologia”.

La transformación nos ha quedado de la siguiente forma:

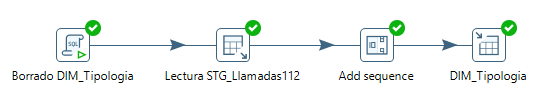


Ilustración - TR\_DIM\_TIPOLOGIA.

#### Borrado

Al igual que en las transformaciones anteriores lo primero que debemos de hacer es el borrado de los registros que tenemos en la dimensión:

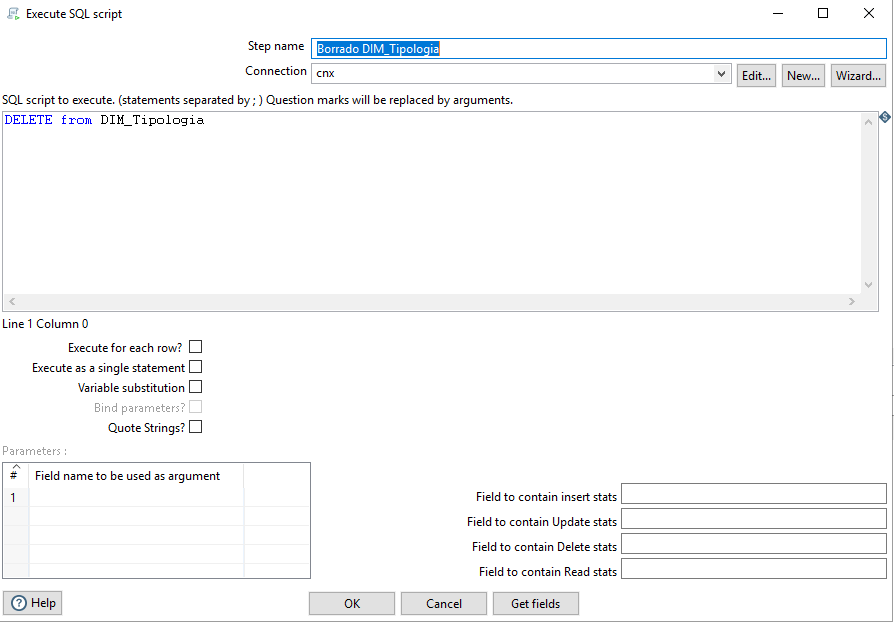


Ilustración - Borrado TR\_DIM\_TIPOLOGIA.

#### Lectura

Aunque en este caso hay también pocas tipologías, para ser más exactos hay 10, podría haber usado un grid pero hemos considerado que lo mejor es hacer la lectura de la tabla intermedia “SGT\_Llamadas112” porque la información no es fija, es decir, en un futuro pueden pasarnos tipologías nuevas y de no hacerlo así tendríamos que modificar la transformación.

Al hacer la lectura indicamos el campo “tipo”, la tabla “STG\_Llamadas112” y con la función distinct nos quedamos con todas las tipologías diferentes:



Ilustración - Lectura TR\_DIM\_TIPOLOGIA.

#### Secuenciación

Al igual que en las transformaciones anteriores, definimos la clave primaria de “DIM\_Tipologia” a partir de una secuencia numérica:

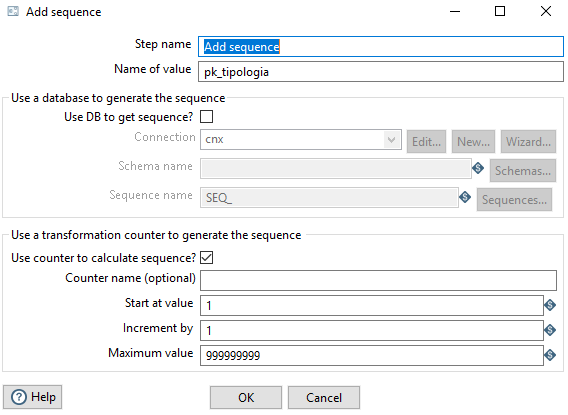


Ilustración - Secuenciación TR\_DIM\_TIPOLOGIA.

#### Guardado

Finalmente, realizamos el guardado en la dimensión indicando la tabla destino como “DIM\_Tipologia” y asociamos los campos:

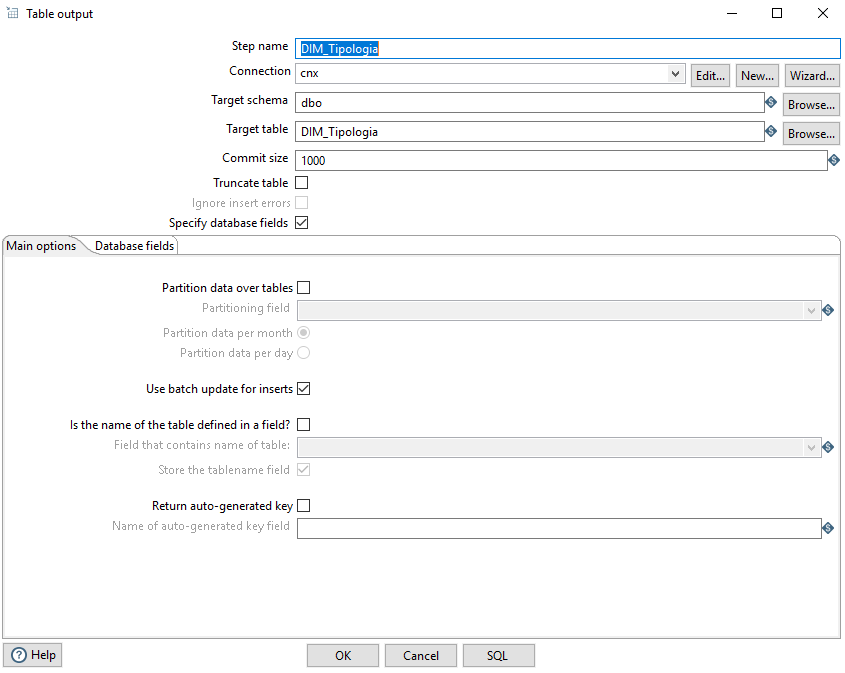


Ilustración - Guardado TR\_DIM\_TIPOLOGIA.

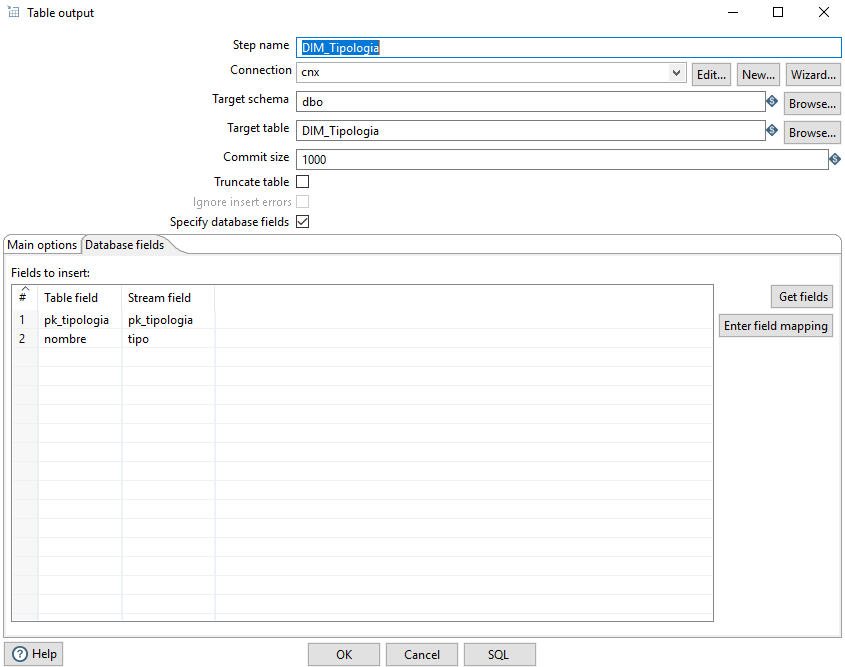


Ilustración - Guardado TR\_DIM\_TIPOLOGIA.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

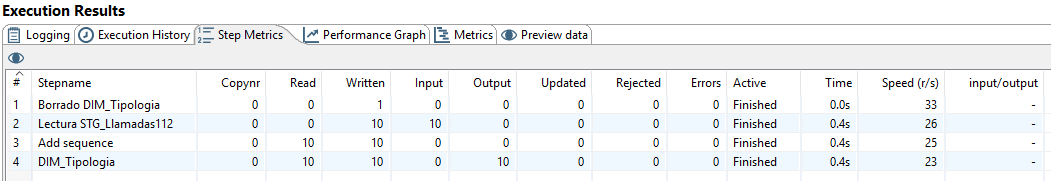


Ilustración - Métricas TR\_DIM\_TIPOLOGIA.

### Transformación TR\_DIM\_AMBITO\_GEOGRAFICO

La cuarta transformación se corresponde con una dimensión compartida por ambos hechos, esta transformación se llama “TR\_DIM\_AMBITO\_GEOGRAFICO” y se encarga de almacenar todos los datos geográficos recogidos de la fuentes proporcionadas, es decir, datos que se encuentran en las tablas intermedias.

Una vez que hemos leído todos los datos los vamos a almacenar a la tabla “DIM\_Ambito\_Geografico”, ya que es ésta la que se corresponde con la dimensión.

La transformación nos ha quedado de la siguiente forma:

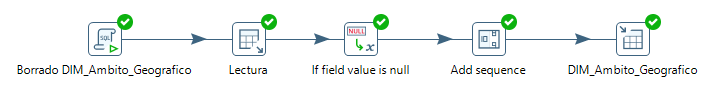


Ilustración - TR\_DIM\_AMBITO\_GEOGRAFICO.

#### Borrado

Lo primero que debemos de hacer es un borrado de los registros (si hay) de “DIM\_Ambito\_Geografico”, para ello escribimos directamente la sentencia SQL y la ejecutamos:

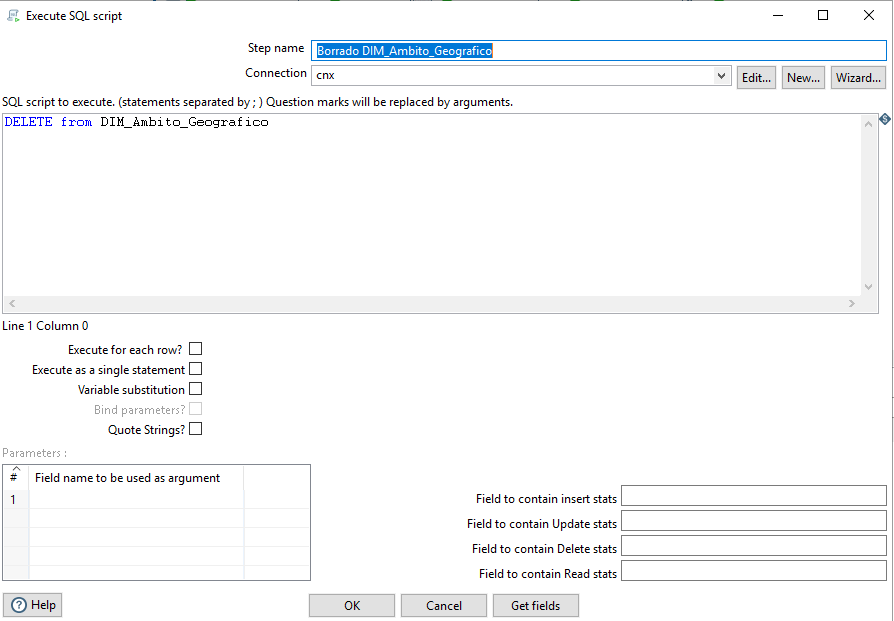


Ilustración - Borrado TR\_DIM\_AMBITO\_GEOGRAFICO.

#### Lectura

En este caso tenemos que introducir en la dimensión todos los datos relativos al ámbito geográfico que tenemos en las tablas intermedias. Tal y como está definida la dimensión los atributos “provincia\_codigo” y “provincia\_nombre” son obligatorios (no pueden ser nulos), es por ello que hacemos diferentes joins entre las tablas implicadas (STG\_Evitar\_Aglomeracion, STG\_Poblacion, STG\_Llamadas112) ya que no todas tienen el atributo “provincia\_codigo”. Además aprovechando los joins, establecemos el nombre de la comunidad a cada provincia.

El script necesario para realizar la operación comentada en el párrafo anterior es el siguiente:

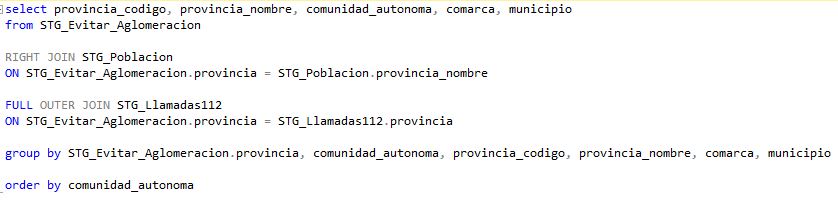


Ilustración - Lectura TR\_DIM\_AMBITO\_GEOGRAFICO.

De tal forma, al agrupar por los campos que apreciamos en la imagen anterior, obtenemos todos los datos geográficos de forma única, es decir, no tenemos duplicados. Finalmente, ordenamos dichos valores por su comunidad para que sea más legible, una vez hecho todo esto usamos Spoon para realizar la carga:

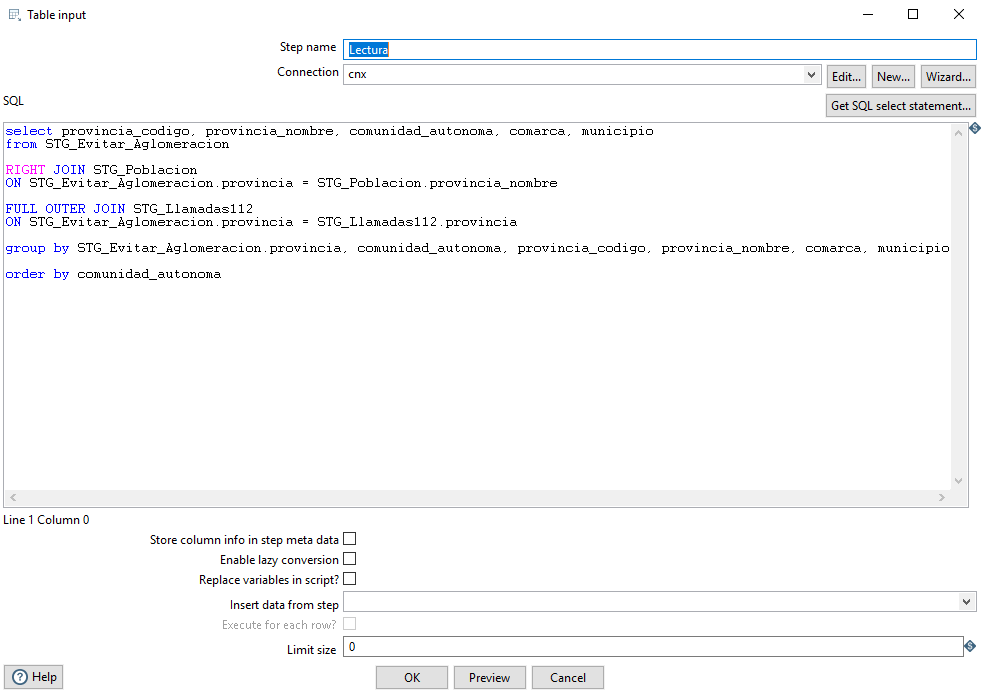


Ilustración - Lectura TR\_DIM\_AMBITO\_GEOGRAFICO.

#### Nulos

Al hacer los joins anteriores en determinados atributos (comunidad autónoma, comarca y municipio) no siempre tienen valor, por ejemplo, la ciudad Ceuta o Melilla no tienen una comunidad como tal, porque son ciudades autónomas pero no comunidades.

Para solventar estos problemas sustituimos los valores nulos de todos los campos por “NA”, esto significa que no es aplicable, de tal forma en Spoon nos quedaría la siguiente configuración:

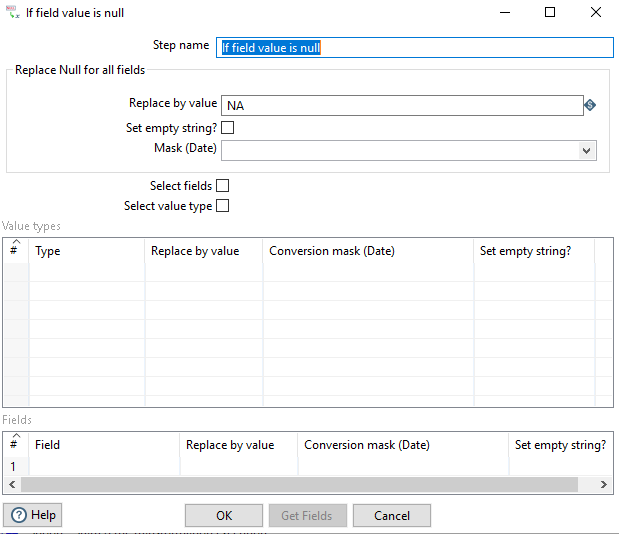


Ilustración - Nulos TR\_DIM\_AMBITO\_GEOGRAFICO.

#### Secuenciación

En este caso no normalizamos los datos porque ya lo hicimos al crear las tablas STG, de tal forma que todos los datos están en mayúsculas y sin espacios.

Otro aspecto a tener en cuenta es la creación de la clave primaria para esta dimensión, por lo que vamos a definir la misma como un autonumérico incrementándose de uno en uno:

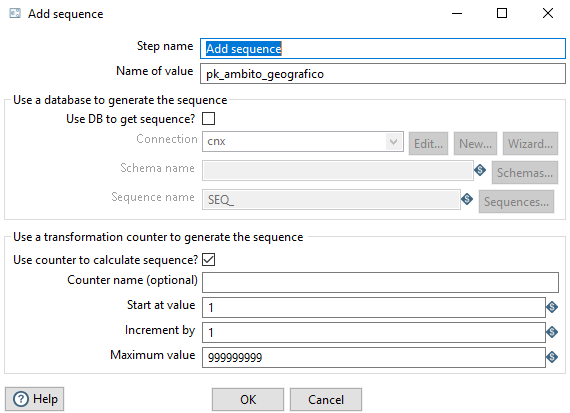


Ilustración - Secuenciación TR\_DIM\_AMBITO\_GEOGRAFICO.

#### Guardado

Finalmente, realizamos el guardado en la dimensión indicando la tabla destino como “DIM\_Ambito\_Geografico” y asociamos los atributos:

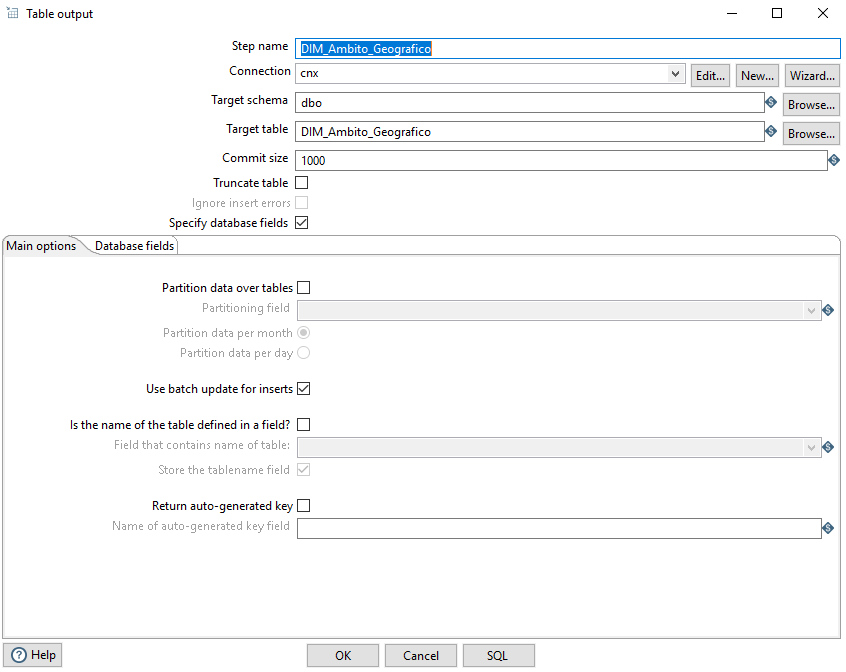


Ilustración - Guardado TR\_DIM\_AMBITO\_GEOGRAFICO.

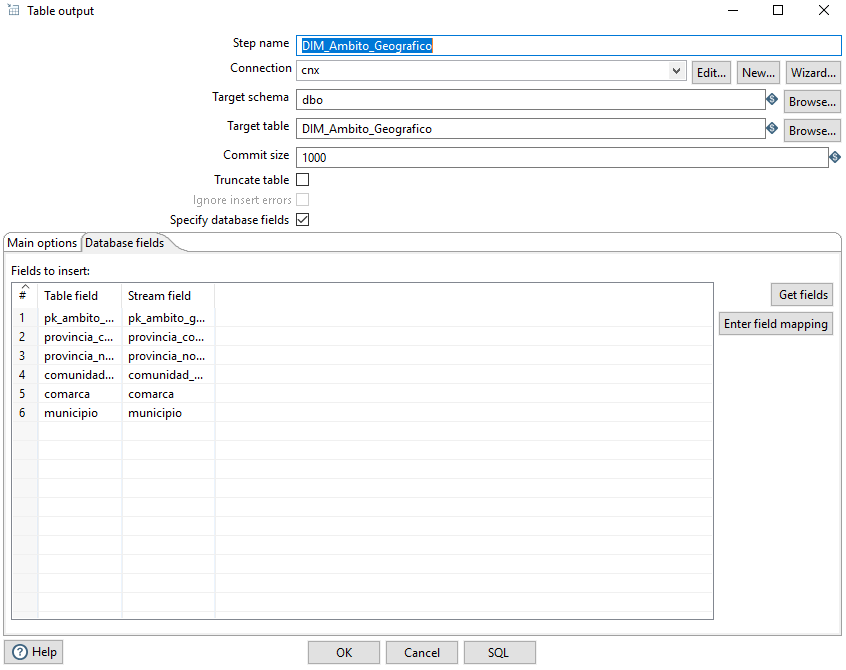


Ilustración - Guardado TR\_DIM\_AMBITO\_GEOGRAFICO.

El ejecutar la anterior transformación obtenemos las siguientes métricas:

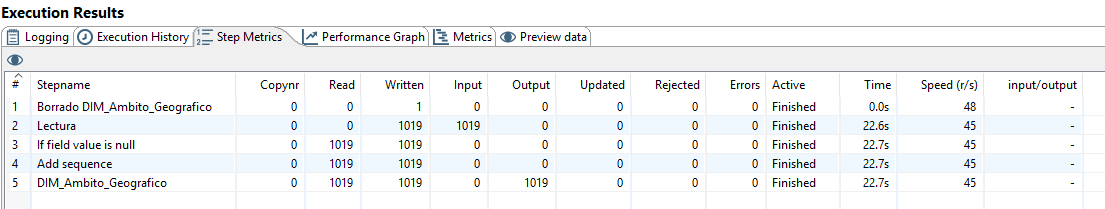


Ilustración - Métricas TR\_DIM\_AMBITO\_GEOGRAFICO.

Como podemos observar leemos 1019 registros y almacenamos en la dimensión los mismos 1019 registros.

### Transformación TR\_DIM\_FECHA

La última transformación respecto a las dimensiones es “TR\_DIM\_FECHA”, su objetivo es almacenar todas las fecha que se encuentran en las tablas intermedias y almacenarlas en “DIM\_Fecha”.

La transformación nos ha quedado de la siguiente forma:

|  |
| --- |
| Bibliografía |

a