

**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](#_Toc72002905)

[Índice de tablas 5](#_Toc72002906)

[Índice de ilustraciones 6](#_Toc72002907)

[1. Introducción 11](#_Toc72002908)

[1.1. Presentación 11](#_Toc72002909)

[1.2. Descripción 11](#_Toc72002910)

[2. Identificación de los procesos ETL 13](#_Toc72002911)

[2.1. Bloque IN 13](#_Toc72002912)

[2.2. Bloque TR 15](#_Toc72002913)

[2.2.1. Dimensiones 15](#_Toc72002914)

[2.2.2. Hechos 16](#_Toc72002915)

[3. Diseño y desarrollo de los procesos ETL 17](#_Toc72002916)

[3.1. Creación de tablas 17](#_Toc72002917)

[3.1.1. Tablas del área intermedia (*staging area*) 17](#_Toc72002918)

[3.1.2. Tablas de las dimensiones 21](#_Toc72002919)

[3.1.3. Tablas de hechos 23](#_Toc72002920)

[3.2. Bloque IN 25](#_Toc72002921)

[3.2.1. Definición de variables de entorno 25](#_Toc72002922)

[3.2.2. Conexión base de datos SQL Server 26](#_Toc72002923)

[3.2.3. Transformación IN\_DENUNCIAS\_INFRACCIONES 27](#_Toc72002924)

[3.2.4. Transformación IN\_POBLACION 32](#_Toc72002925)

[3.2.5. Transformación IN\_MOVILIDAD 35](#_Toc72002926)

[3.2.6. Transformación IN\_AGLOMERACION 40](#_Toc72002927)

[3.2.7. Transformación IN\_LLAMADAS112 47](#_Toc72002928)

[3.2.8. Transformación IN\_FECHAS 53](#_Toc72002929)

[3.3. Bloque TR Dimensiones 69](#_Toc72002930)

[3.3.1. Transformación TR\_DIM\_GRUPO\_EDAD 69](#_Toc72002931)

[3.3.2. Transformación TR\_DIM\_Medicion 72](#_Toc72002932)

[3.3.3. Transformación TR\_DIM\_TIPOLOGIA 76](#_Toc72002933)

[3.3.4. Transformación TR\_DIM\_AMBITO\_GEOGRAFICO 80](#_Toc72002934)

[3.3.5. Transformación TR\_DIM\_FECHA 84](#_Toc72002935)

[4. Bibliografía **¡Error! Marcador no definido.**](#_Toc72002936)

|  |
| --- |
| Índice de tablas |

[Tabla 1 - Procesos ETL Bloque IN. 15](#_Toc72002157)

[Tabla 2 - Procesos ETL Bloque TR Dimensiones. 15](#_Toc72002158)

[Tabla 3 - Procesos ETL Bloque TR Hechos. 16](#_Toc72002159)

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

[Ilustración 1 - STG\_Denuncias\_Infracciones. 18](#_Toc72002780)

[Ilustración 2 - STG\_Poblacion. 18](#_Toc72002781)

[Ilustración 3 - STG\_Llamadas112. 19](#_Toc72002782)

[Ilustración 4 - STG\_Movilidad. 19](#_Toc72002783)

[Ilustración 5 - STG\_Evitar\_Aglomeracion. 20](#_Toc72002784)

[Ilustración 6 - STG\_Fechas. 20](#_Toc72002785)

[Ilustración 7 - Tablas de staging area. **¡Error! Marcador no definido.**](#_Toc72002786)

[Ilustración 8 - DIM\_Ambito\_Geografico. 22](#_Toc72002787)

[Ilustración 9 - DIM\_Fecha. 22](#_Toc72002788)

[Ilustración 10 - DIM\_Grupo\_Edad. 22](#_Toc72002789)

[Ilustración 11 - DIM\_Medicion. 23](#_Toc72002790)

[Ilustración 12 - DIM\_Tipologia. 23](#_Toc72002791)

[Ilustración 13 - Tablas de dimensiones. 23](#_Toc72002792)

[Ilustración 14 - FACT\_Llamadas112. 24](#_Toc72002793)

[Ilustración 15 - FACT\_Mediciones. 24](#_Toc72002794)

[Ilustración 16 - Alter table hechos. 25](#_Toc72002795)

[Ilustración 17 - Tablas de hechos. 25](#_Toc72002796)

[Ilustración 18 - Variables de entorno. 26](#_Toc72002797)

[Ilustración 19 - Conexión a la base de datos. 27](#_Toc72002798)

[Ilustración 20 - IN\_DENUNCIAS\_INFRACCIONES. 27](#_Toc72002799)

[Ilustración 21 - Lectura IN\_DENUNCIAS\_INFRACCIONES. 28](#_Toc72002800)

[Ilustración 22 - Lectura IN\_DENUNCIAS\_INFRACCIONES. 28](#_Toc72002801)

[Ilustración 23 - Lectura IN\_DENUNCIAS\_INFRACCIONES. 29](#_Toc72002802)

[Ilustración 24 - Mapeo Valores IN\_DENUNCIAS\_INFRACCIONES. 29](#_Toc72002803)

[Ilustración 25 - Normalización Strings IN\_DENUNCIAS\_INFRACCIONES. 30](#_Toc72002804)

[Ilustración 26 - Ordenación IN\_DENUNCIAS\_INFRACCIONES. 30](#_Toc72002805)

[Ilustración 27 - Guardado IN\_DENUNCIAS\_INFRACCIONES. 31](#_Toc72002806)

[Ilustración 28 - Métricas IN\_DENUNCIAS\_INFRACCIONES. 31](#_Toc72002807)

[Ilustración 29 - IN\_POBLACION. 32](#_Toc72002808)

[Ilustración 30 - Lectura IN\_POBLACION. 32](#_Toc72002809)

[Ilustración 31 - Separación Campos IN\_POBLACION. 33](#_Toc72002810)

[Ilustración 32 - Mapeo Valores IN\_POBLACION. 33](#_Toc72002811)

[Ilustración 33 - Normalización Strings IN\_POBLACION. 34](#_Toc72002812)

[Ilustración 34 - Guardado IN\_POBLACION. 34](#_Toc72002813)

[Ilustración 35 - Guardado IN\_POBLACION. 35](#_Toc72002814)

[Ilustración 36 - Métricas IN\_POBLACION. 35](#_Toc72002815)

[Ilustración 37 - IN\_MOVILIDAD. 36](#_Toc72002816)

[Ilustración 38 - Lectura IN\_MOVILIDAD. 36](#_Toc72002817)

[Ilustración 39 - Mapeo Valores IN\_MOVILIDAD. 37](#_Toc72002818)

[Ilustración 40 - Normalización IN\_MOVILIDAD. 37](#_Toc72002819)

[Ilustración 41 - Replace IN\_MOVILIDAD. 38](#_Toc72002820)

[Ilustración 42 - Select Values IN\_MOVILIDAD. 38](#_Toc72002821)

[Ilustración 43 - Guardado IN\_MOVILIDAD. 39](#_Toc72002822)

[Ilustración 44 - Guardado IN\_MOVILIDAD. 40](#_Toc72002823)

[Ilustración 45 - Métricas IN\_MOVILIDAD. 40](#_Toc72002824)

[Ilustración 46 - IN\_AGLOMERACION. 41](#_Toc72002825)

[Ilustración 47 - Lectura IN\_AGLOMERACION. 41](#_Toc72002826)

[Ilustración 48 - Lectura IN\_AGLOMERACION. 42](#_Toc72002827)

[Ilustración 49 - Lectura IN\_AGLOMERACIONES. 42](#_Toc72002828)

[Ilustración 50 - Mapeo Valores IN\_AGLOMERACION. 43](#_Toc72002829)

[Ilustración 51 - Replace IN\_AGLOMERACION. 43](#_Toc72002830)

[Ilustración 52 - Split IN\_AGLOMERACION. 44](#_Toc72002831)

[Ilustración 53 - Normalización Strings IN\_AGLOMERACION. 44](#_Toc72002832)

[Ilustración 54 - Replace IN\_AGLOMERACION. 45](#_Toc72002833)

[Ilustración 55 - Normalización Filas IN\_AGLOMERACION. 45](#_Toc72002834)

[Ilustración 56 - Guardado IN\_AGLOMERACIONES. 46](#_Toc72002835)

[Ilustración 57 - Guardado IN\_AGLOMERACIONES. 47](#_Toc72002836)

[Ilustración 58 - Métricas IN\_AGLOMERACION. 47](#_Toc72002837)

[Ilustración 59 - IN\_LLAMADAS112. 48](#_Toc72002838)

[Ilustración 60 - Lectura IN\_LLAMADAS112. 48](#_Toc72002839)

[Ilustración 61 - Lectura IN\_LLAMADAS112. 49](#_Toc72002840)

[Ilustración 62 - Lectura IN\_LLAMADAS112. 49](#_Toc72002841)

[Ilustración 63 - Mapeo Valores IN\_LLAMADAS112. 50](#_Toc72002842)

[Ilustración 64 - Mapeo Valores IN\_LLAMADAS112. 51](#_Toc72002843)

[Ilustración 65 - Normalización IN\_LLAMADAS112. 51](#_Toc72002844)

[Ilustración 66 - Guardado IN\_LLAMADAS112. 52](#_Toc72002845)

[Ilustración 67 - Guardado IN\_LLAMADAS112. 53](#_Toc72002846)

[Ilustración 68 - Métricas IN\_LLAMADAS112. 53](#_Toc72002847)

[Ilustración 69 - IN\_FECHAS. 54](#_Toc72002848)

[Ilustración 70 - Borrado IN\_FECHAS. 54](#_Toc72002849)

[Ilustración 71 - Lectura IN\_FECHAS. 55](#_Toc72002850)

[Ilustración 72 - Añadimos Constante IN\_FECHAS. 55](#_Toc72002851)

[Ilustración 73 - Concatenación IN\_FECHAS. 56](#_Toc72002852)

[Ilustración 74 - Conversión IN\_FECHAS. 56](#_Toc72002853)

[Ilustración 75 - Guardado IN\_FECHAS. 57](#_Toc72002854)

[Ilustración 76 - Lectura IN\_FECHAS. 58](#_Toc72002855)

[Ilustración 77 - Guardado IN\_FECHAS. 59](#_Toc72002856)

[Ilustración 78 - Lectura IN\_FECHAS. 60](#_Toc72002857)

[Ilustración 79 - Guardado IN\_FECHAS. 61](#_Toc72002858)

[Ilustración 80 - Métricas IN\_FECHAS. 61](#_Toc72002859)

[Ilustración 81 - TR\_DIM\_GRUPO\_EDAD. 69](#_Toc72002860)

[Ilustración 82 - Borrado TR\_DIM\_GRUPO\_EDAD. 69](#_Toc72002861)

[Ilustración 83 - Grid TR\_DIM\_GRUPO\_EDAD. 70](#_Toc72002862)

[Ilustración 84 - Grid TR\_DIM\_GRUPO\_EDAD. 70](#_Toc72002863)

[Ilustración 85 - Normalización TR\_DIM\_GRUPO\_EDAD. 70](#_Toc72002864)

[Ilustración 86 - Secuenciación TR\_DIM\_GRUPO\_EDAD. 71](#_Toc72002865)

[Ilustración 87 - Guardado TR\_DIM\_GRUPO\_EDAD. 71](#_Toc72002866)

[Ilustración 88 - Guardado TR\_DIM\_GRUPO\_EDAD. 72](#_Toc72002867)

[Ilustración 89 - Métricas TR\_DIM\_GRUPO\_EDAD. 72](#_Toc72002868)

[Ilustración 90 - TR\_DIM\_MEDICION. 73](#_Toc72002869)

[Ilustración 91 - Borrado TR\_DIM\_MEDICION. 73](#_Toc72002870)

[Ilustración 92 - Grid TR\_DIM\_DIM\_MEDICION. 73](#_Toc72002871)

[Ilustración 93 - Grid TR\_DIM\_MEDICION. 74](#_Toc72002872)

[Ilustración 94 - Normalización TR\_DIM\_MEDICION. 74](#_Toc72002873)

[Ilustración 95 - Secuenciación TR\_DIM\_MEDICION. 75](#_Toc72002874)

[Ilustración 96 - Guardado TR\_DIM\_MEDICION. 75](#_Toc72002875)

[Ilustración 97 - Guardado TR\_DIM\_MEDICION. 76](#_Toc72002876)

[Ilustración 98 - Métricas TR\_DIM\_MEDICIONES. 76](#_Toc72002877)

[Ilustración 99 - TR\_DIM\_TIPOLOGIA. 77](#_Toc72002878)

[Ilustración 100 - Borrado TR\_DIM\_TIPOLOGIA. 77](#_Toc72002879)

[Ilustración 101 - Lectura TR\_DIM\_TIPOLOGIA. 78](#_Toc72002880)

[Ilustración 102 - Secuenciación TR\_DIM\_TIPOLOGIA. 79](#_Toc72002881)

[Ilustración 103 - Guardado TR\_DIM\_TIPOLOGIA. 79](#_Toc72002882)

[Ilustración 104 - Guardado TR\_DIM\_TIPOLOGIA. 80](#_Toc72002883)

[Ilustración 105 - Métricas TR\_DIM\_TIPOLOGIA. 80](#_Toc72002884)

[Ilustración 106 - TR\_DIM\_AMBITO\_GEOGRAFICO. 81](#_Toc72002885)

[Ilustración 107 - Borrado TR\_DIM\_AMBITO\_GEOGRAFICO. 81](#_Toc72002886)

[Ilustración 108 - Lectura TR\_DIM\_AMBITO\_GEOGRAFICO. **¡Error! Marcador no definido.**](#_Toc72002887)

[Ilustración 109 - Lectura TR\_DIM\_AMBITO\_GEOGRAFICO. **¡Error! Marcador no definido.**](#_Toc72002888)

[Ilustración 110 - Nulos TR\_DIM\_AMBITO\_GEOGRAFICO. **¡Error! Marcador no definido.**](#_Toc72002889)

[Ilustración 111 - Secuenciación TR\_DIM\_AMBITO\_GEOGRAFICO. 83](#_Toc72002890)

[Ilustración 112 - Guardado TR\_DIM\_AMBITO\_GEOGRAFICO. 83](#_Toc72002891)

[Ilustración 113 - Guardado TR\_DIM\_AMBITO\_GEOGRAFICO. 84](#_Toc72002892)

[Ilustración 114 - Métricas TR\_DIM\_AMBITO\_GEOGRAFICO. 84](#_Toc72002893)

[Ilustración 115 - TR\_DIM\_FECHA. 85](#_Toc72002894)

[Ilustración 116 - Borrado TR\_DIM\_FECHA. 85](#_Toc72002895)

[Ilustración 117 - Lectura TR\_DIM\_FECHA. 86](#_Toc72002896)

[Ilustración 118 - Conversión String TR\_DIM\_FECHA. 86](#_Toc72002897)

[Ilustración 119 - Split TR\_DIM\_FECHA. 87](#_Toc72002898)

[Ilustración 120 - Split TR\_DIM\_FECHA. 87](#_Toc72002899)

[Ilustración 121 - Concatenación TR\_DIM\_FECHA. 88](#_Toc72002900)

[Ilustración 122 - Conversión TR\_DIM\_FECHA. 88](#_Toc72002901)

[Ilustración 123 - Secuenciación TR\_DIM\_FECHA. 89](#_Toc72002902)

[Ilustración 124 - Guardado TR\_DIM\_FECHA. 89](#_Toc72002903)

[Ilustración 125 - Métricas TR\_DIM\_FECHA. 90](#_Toc72002904)

|  |
| --- |
| 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 |
| IN\_AMBITO\_GEOGRAFICO | Almacenamos todas las localización de todos los ficheros de datos proporcionados. | STG\_Evitar\_Aglomeraciones  STG\_Poblacion  STG\_Llamadas112 | STG\_Ambito\_Geografico |

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\_Ambito\_Geografico | 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

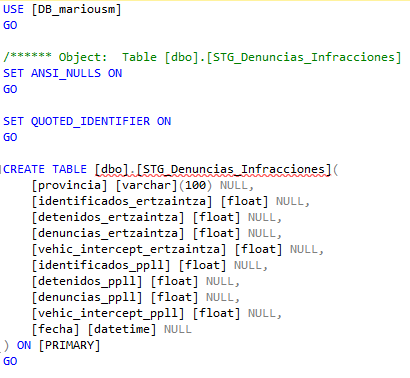


Ilustración 1 - STG\_Denuncias\_Infracciones.

#### Tabla intermedia STG\_Poblacion

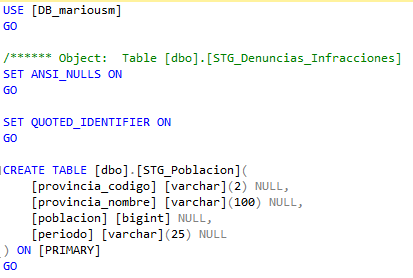


Ilustración 2 - STG\_Poblacion.

#### Tabla intermedia STG\_Llamadas112

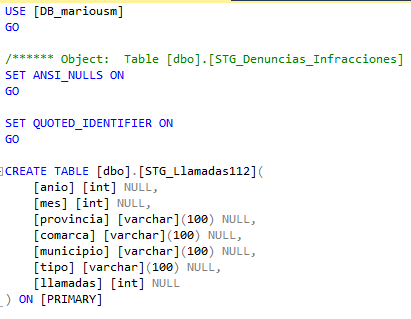


Ilustración 3 - STG\_Llamadas112.

#### Tabla intermedia STG\_Movilidad

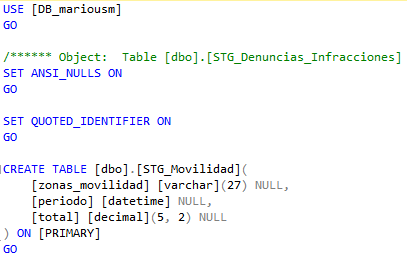


Ilustración 4 - STG\_Movilidad.

#### Tabla intermedia STG\_Evitar\_Aglomeracion

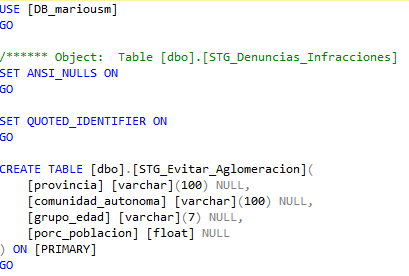


Ilustración 5 - STG\_Evitar\_Aglomeracion.

#### Tabla intermedia STG\_Fechas

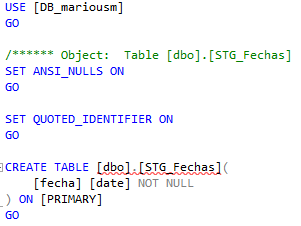


Ilustración - STG\_Fechas.

#### Tabla intermedia STG\_Ambito\_Geografico

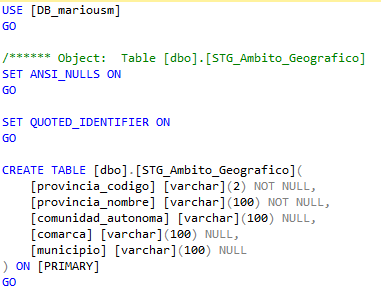


Ilustración - STG\_Ambito\_Geografico.

Comprobamos que todas las tablas intermedias se han creado de forma correcta:

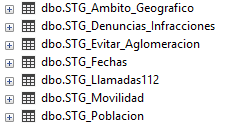


Ilustración - Tablas intermedias STG.

### Tablas de las dimensiones

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

#### Tabla dimensión DIM\_Ambito\_Geografico

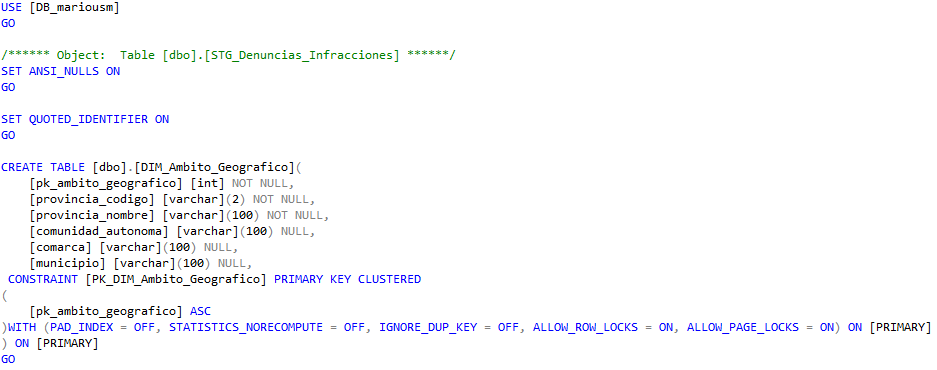


Ilustración 9 - DIM\_Ambito\_Geografico.

#### Tabla dimensión DIM\_Fecha

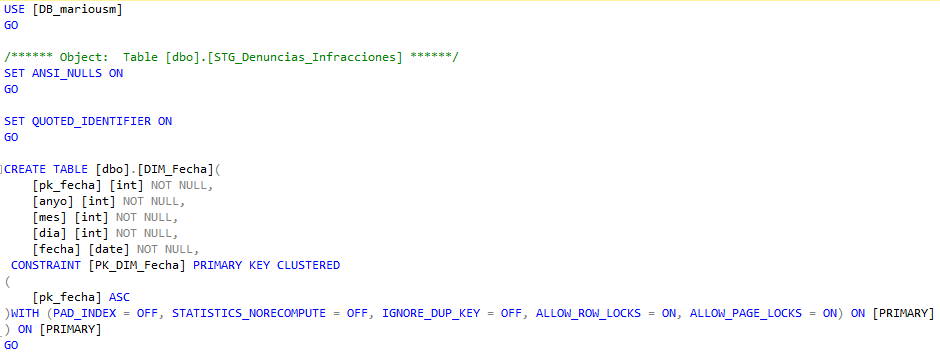


Ilustración 10 - DIM\_Fecha.

#### Tabla dimensión DIM\_Grupo\_Edad

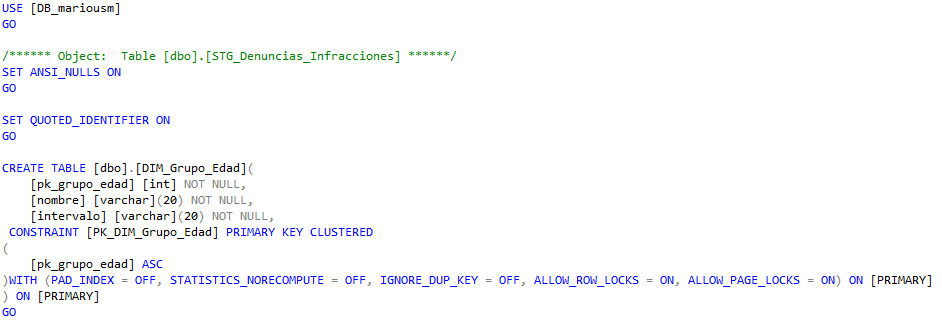


Ilustración 11 - DIM\_Grupo\_Edad.

#### Tabla dimensión DIM\_Medicion

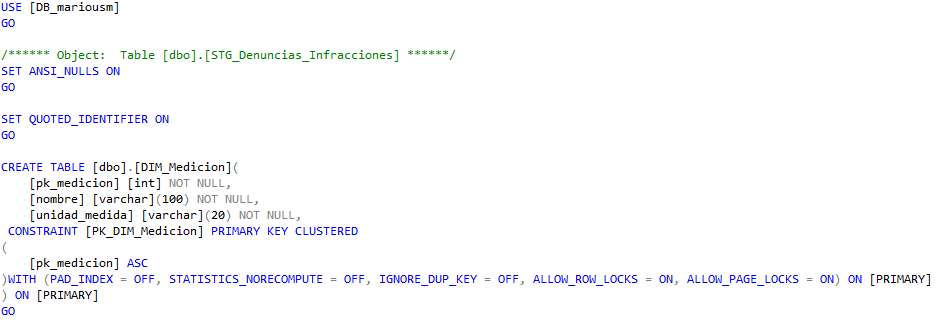


Ilustración 12 - DIM\_Medicion.

#### Tabla dimensión DIM\_Tipologia



Ilustración 13 - DIM\_Tipologia.

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

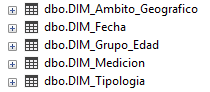


Ilustración 14 - Tablas de dimensiones.

### Tablas de hechos

Finalmente creamos las diferentes tablas de los hechos:

#### Tabla hecho FACT\_Llamadas112

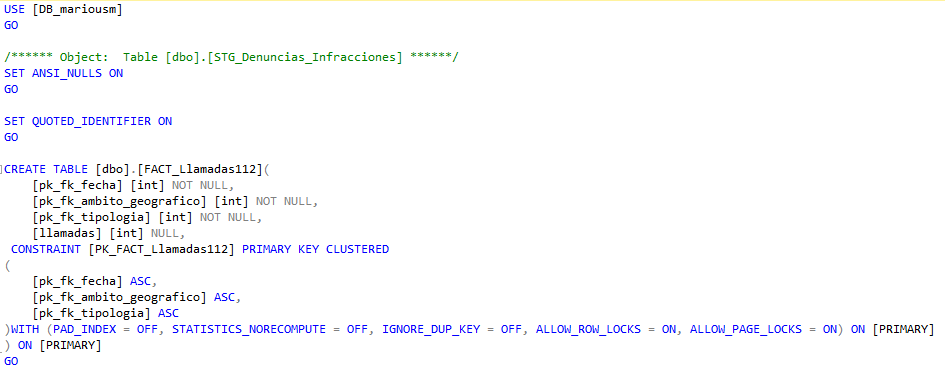


Ilustración 15 - FACT\_Llamadas112.

#### Tabla hecho FACT\_Mediciones

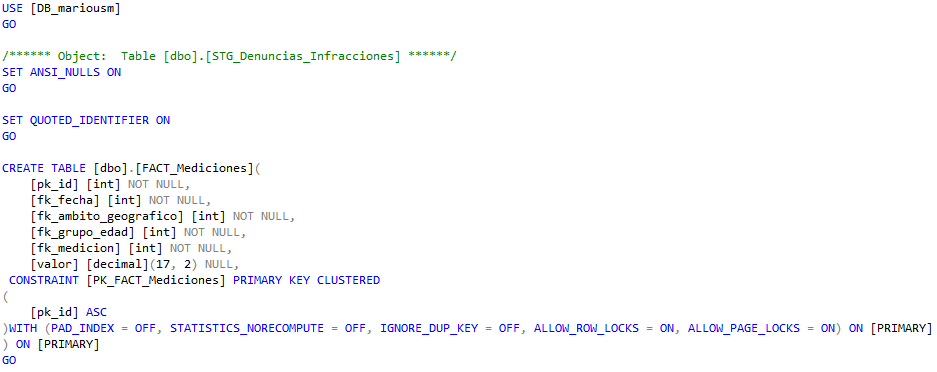


Ilustración 16 - FACT\_Mediciones.

Realizamos los alter table de las tablas de hechos:

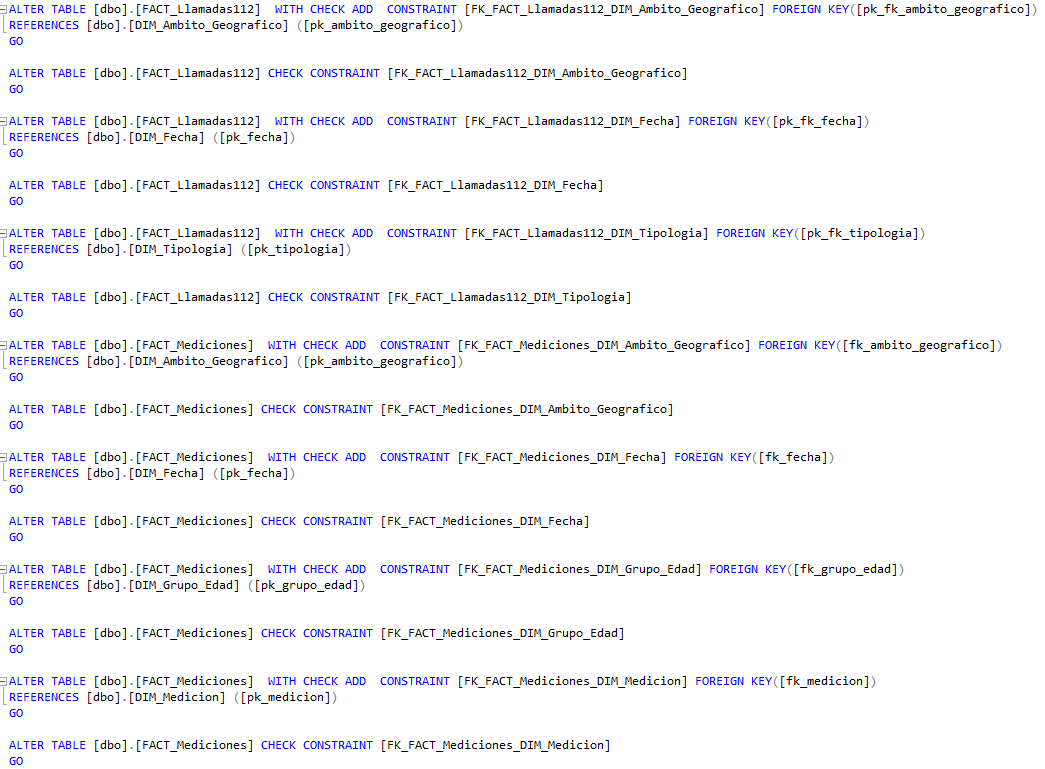


Ilustración 17 - Alter table hechos.

Comprobamos que se han creado todas las tablas correspondientes:



Ilustración 18 - 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

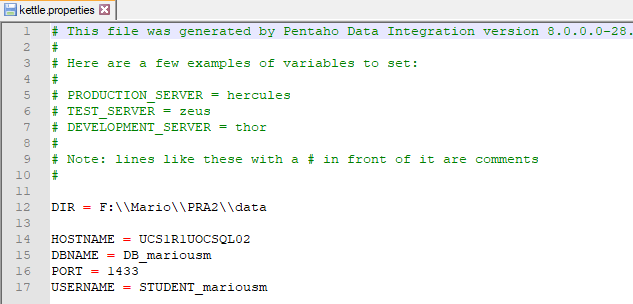


Ilustración 19 - Variables de entorno.

### 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:

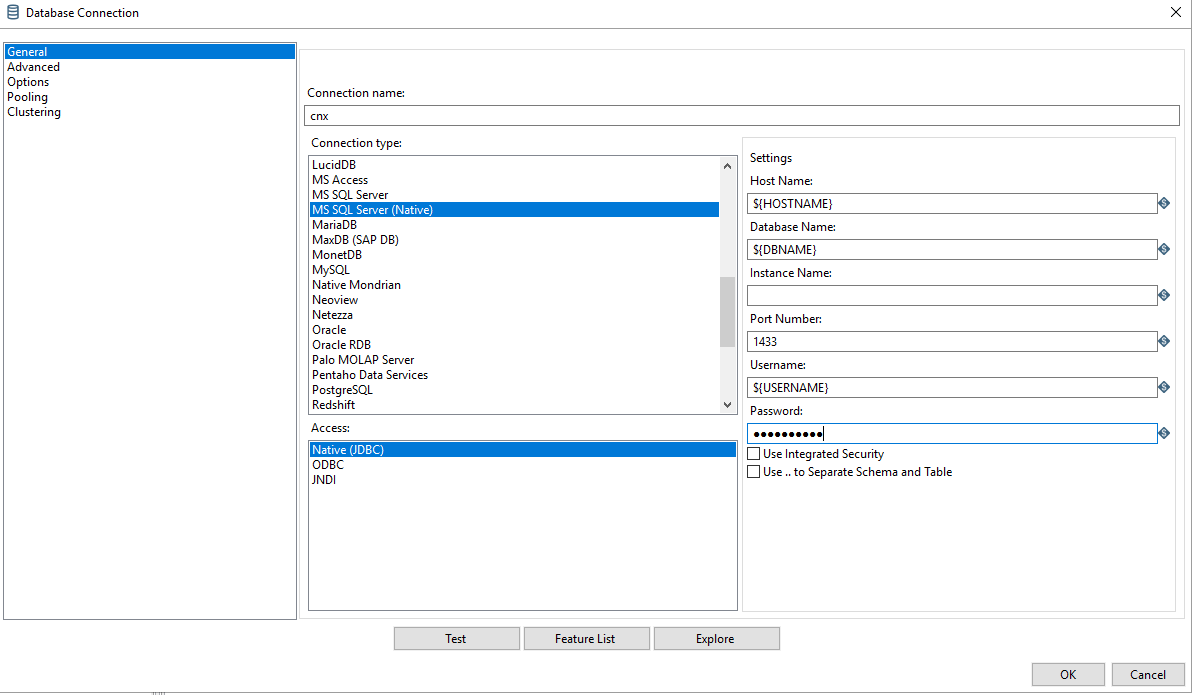


Ilustración 20 - Conexión a la base de datos.

### 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:

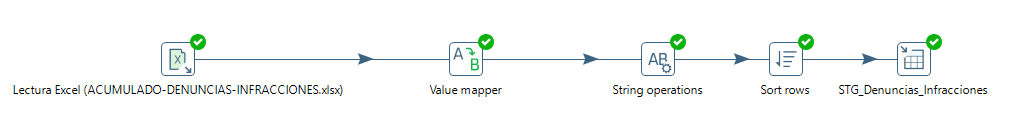


Ilustración 21 - IN\_DENUNCIAS\_INFRACCIONES.

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 XLSX:

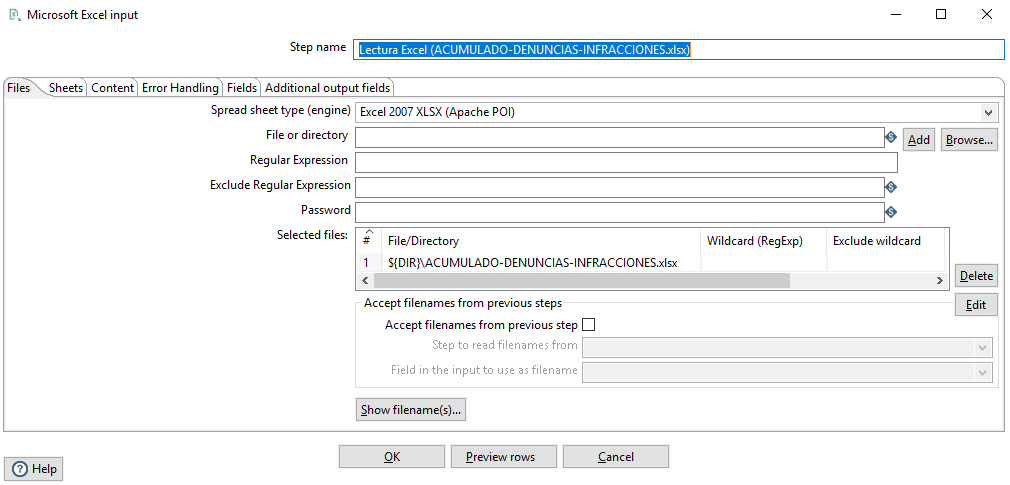


Ilustración 22 - Lectura IN\_DENUNCIAS\_INFRACCIONES.

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:

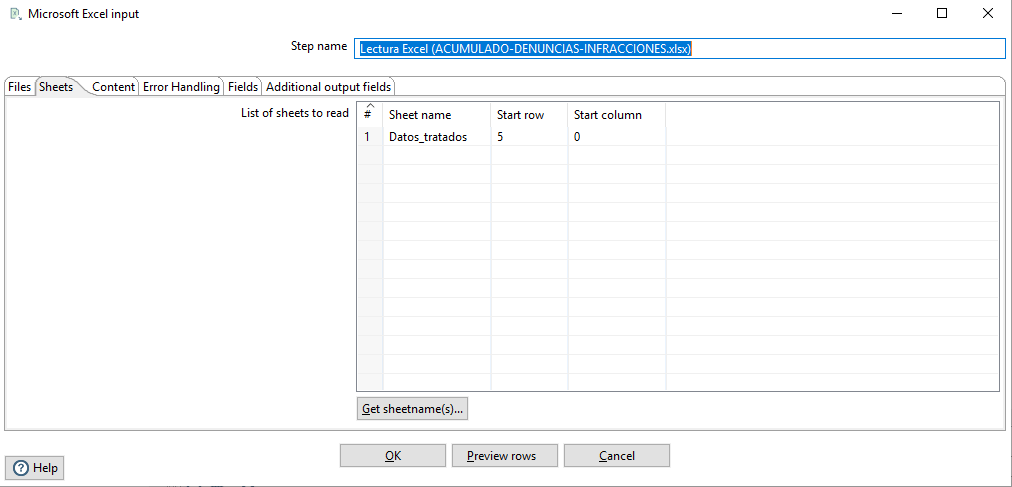


Ilustración 23 - Lectura IN\_DENUNCIAS\_INFRACCIONES.

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

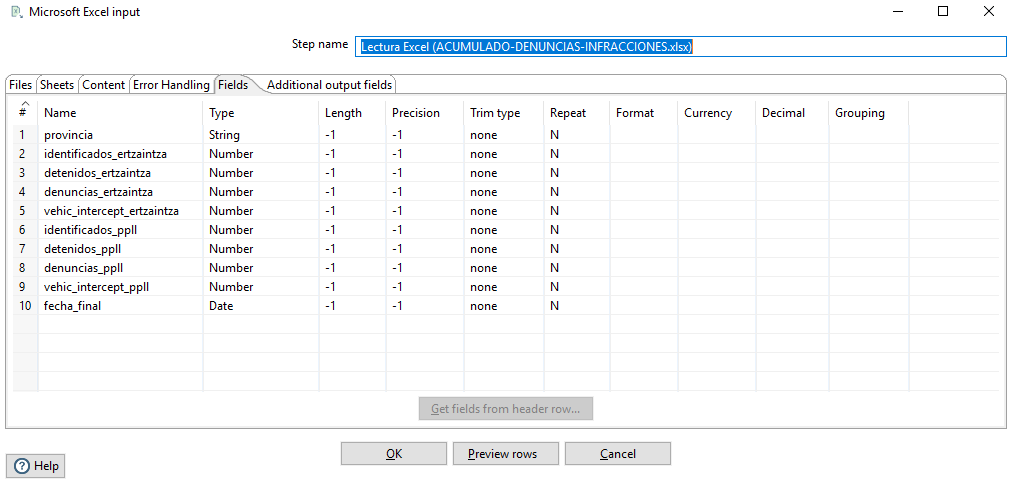


Ilustración 24 - Lectura IN\_DENUNCIAS\_INFRACCIONES.

#### 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:

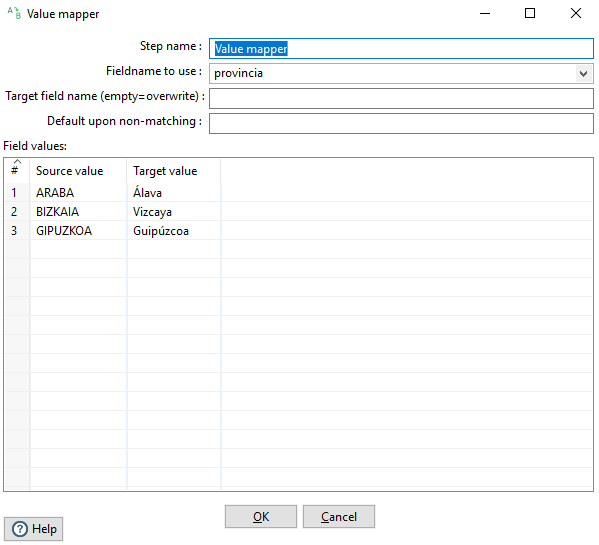


Ilustración 25 - Mapeo Valores IN\_DENUNCIAS\_INFRACCIONES.

#### Normalización

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

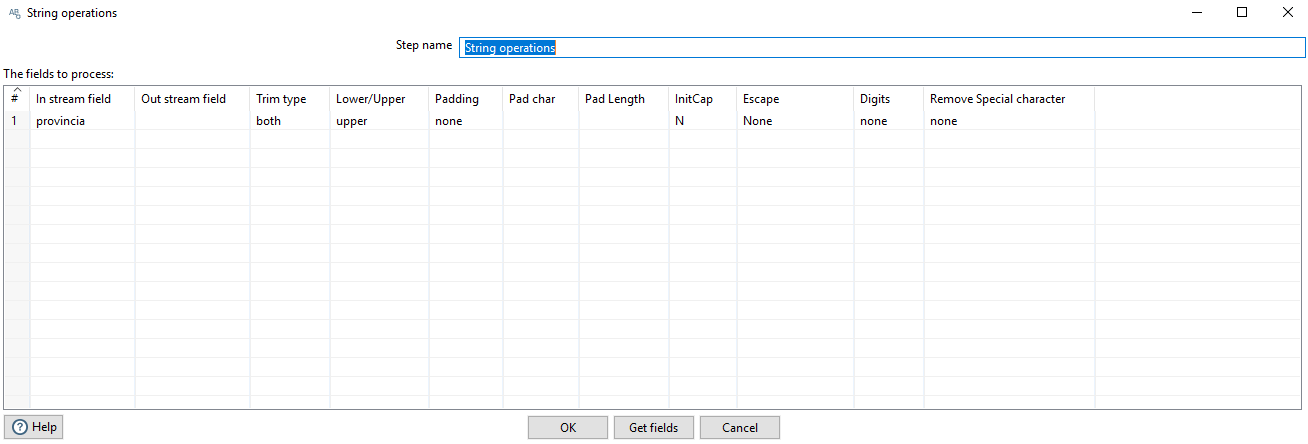


Ilustración 26 - Normalización Strings IN\_DENUNCIAS\_INFRACCIONES.

#### Ordenación

Posteriormente, ordenamos todos los campos de forma ascendente:

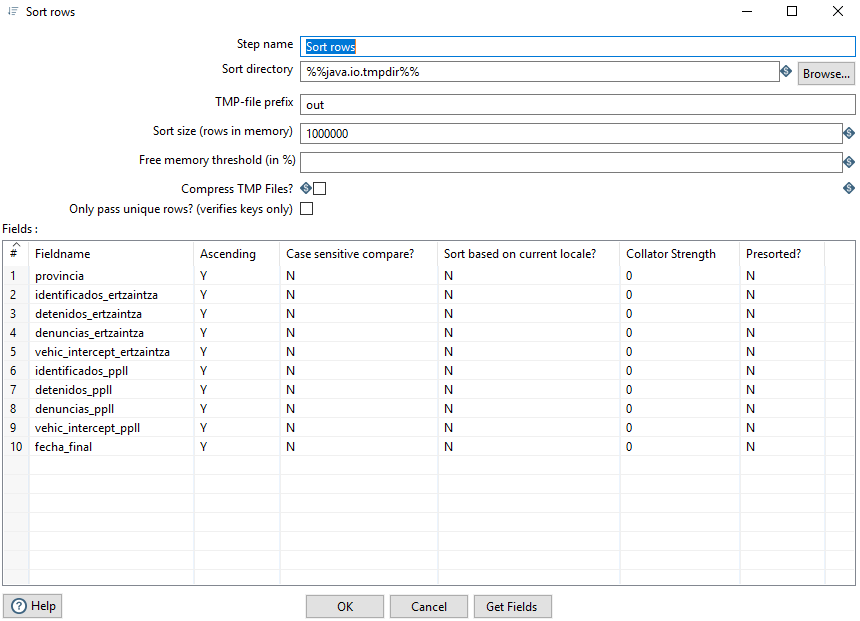


Ilustración 27 - Ordenación IN\_DENUNCIAS\_INFRACCIONES.

#### 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:

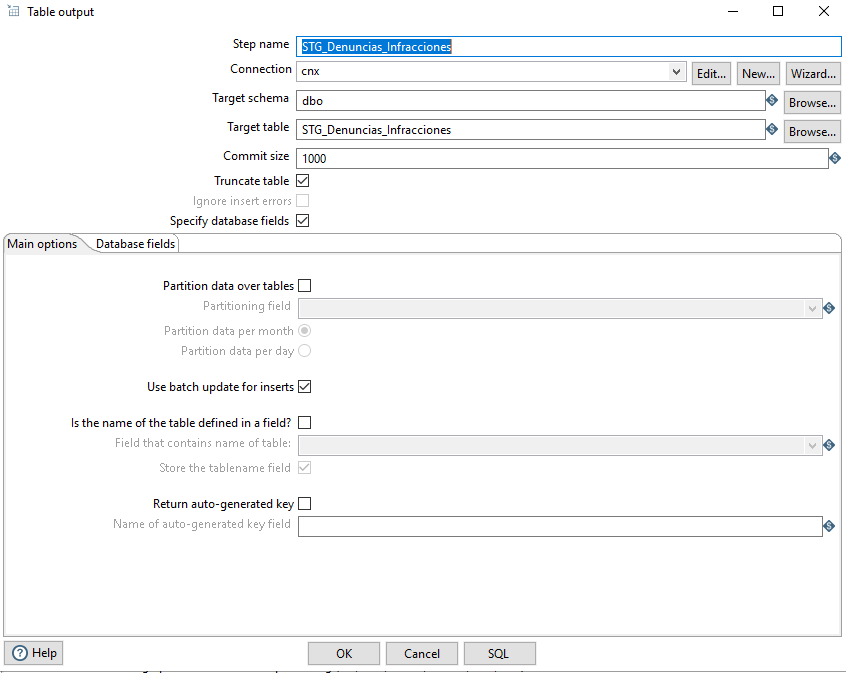


Ilustración 28 - Guardado IN\_DENUNCIAS\_INFRACCIONES.

Al ejecutar la anterior transformación obtenemos las siguiente métricas:

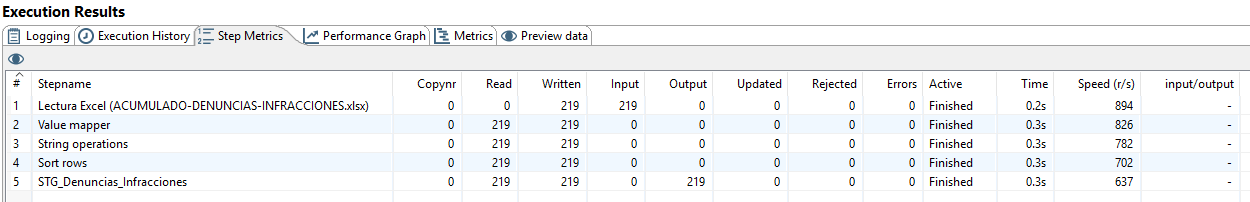


Ilustración 29 - Métricas IN\_DENUNCIAS\_INFRACCIONES.

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:

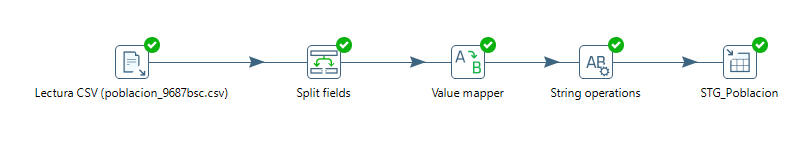


Ilustración 30 - IN\_POBLACION.

#### 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 “;”:

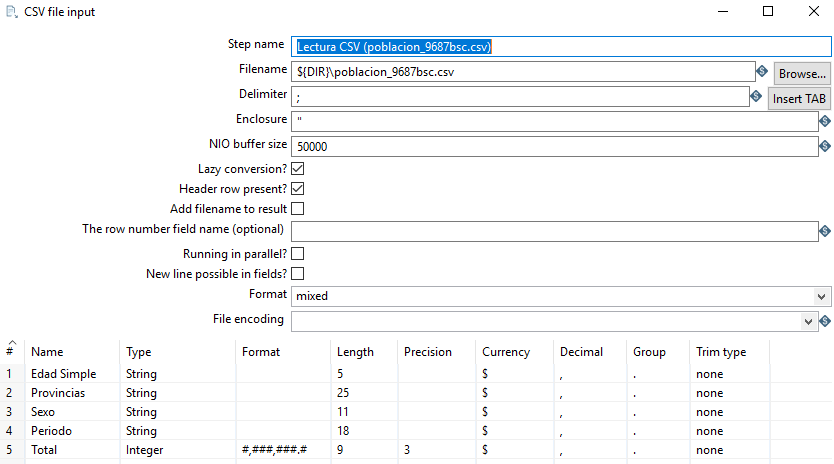


Ilustración 31 - Lectura IN\_POBLACION.

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:

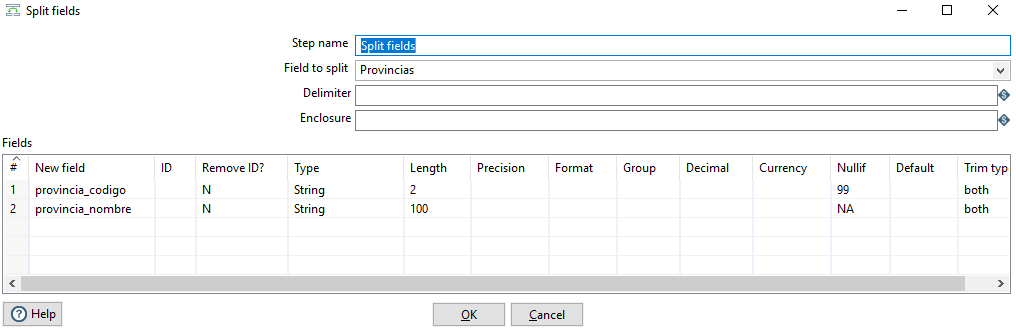


Ilustración 32 - Separación Campos IN\_POBLACION.

#### 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:

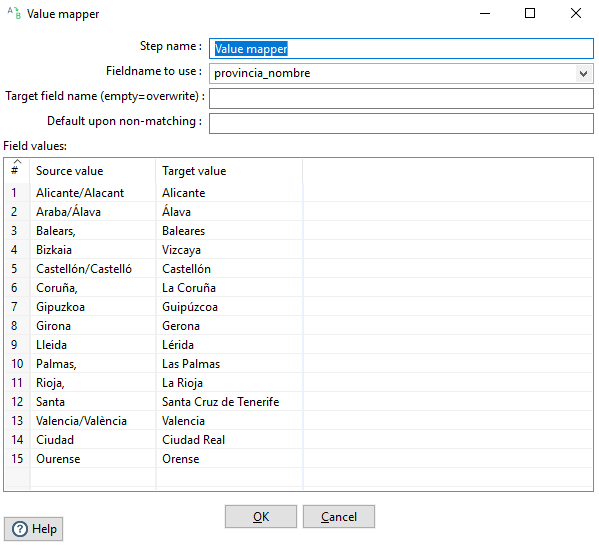


Ilustración 33 - Mapeo Valores IN\_POBLACION.

#### 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:

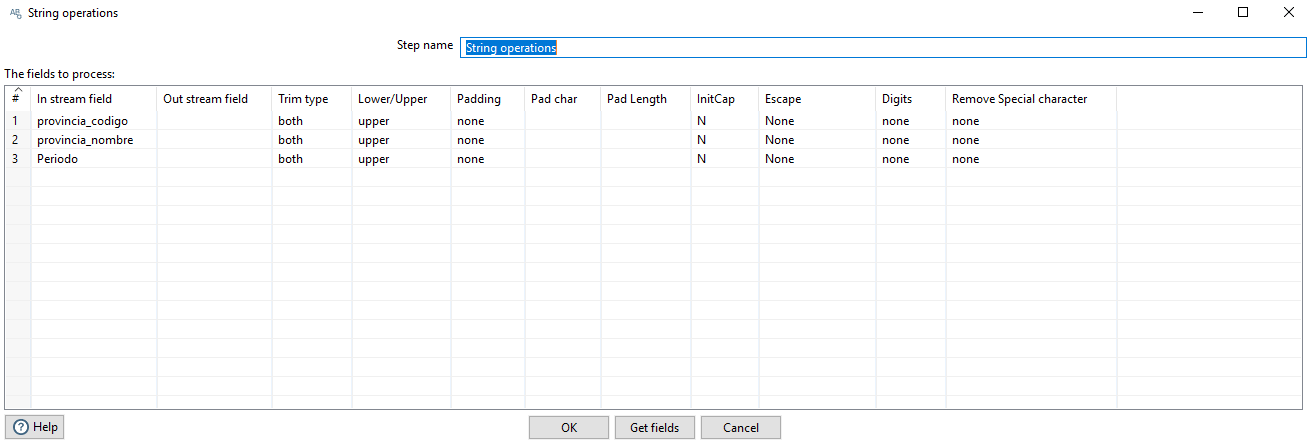


Ilustración 34 - Normalización Strings IN\_POBLACION.

#### Guardado

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

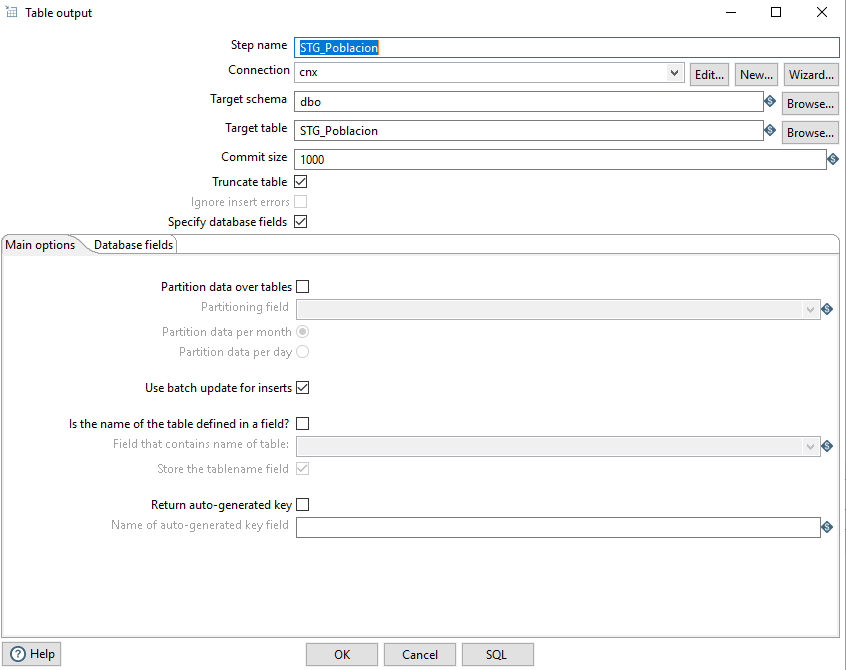


Ilustración 35 - Guardado IN\_POBLACION.

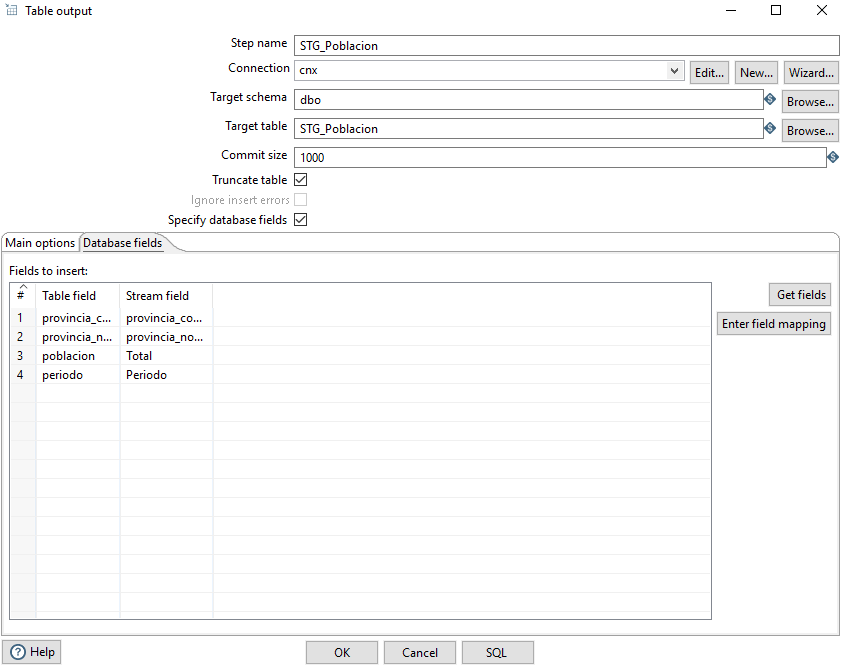


Ilustración 36 - Guardado IN\_POBLACION.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

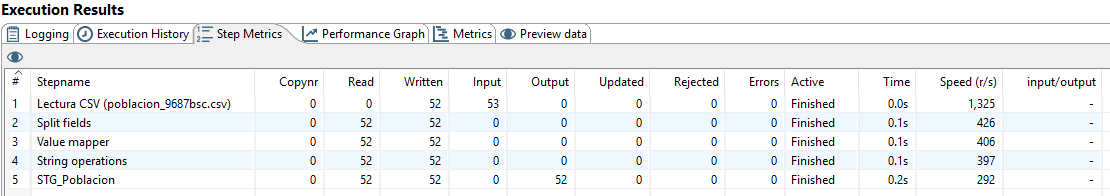


Ilustración 37 - Métricas IN\_POBLACION.

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:

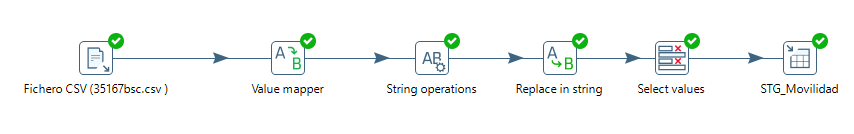


Ilustración 38 - IN\_MOVILIDAD.

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 “,”:

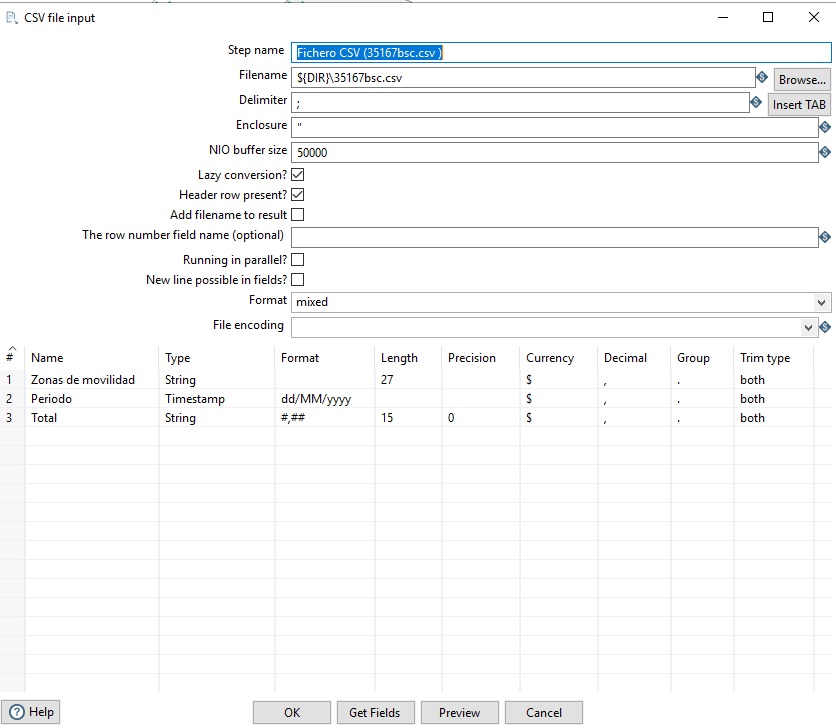


Ilustración 39 - Lectura IN\_MOVILIDAD.

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.

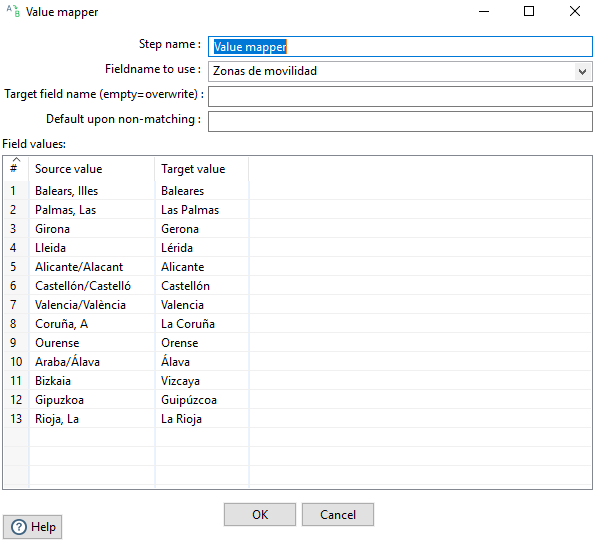


Ilustración 40 - Mapeo Valores IN\_MOVILIDAD.

#### 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:

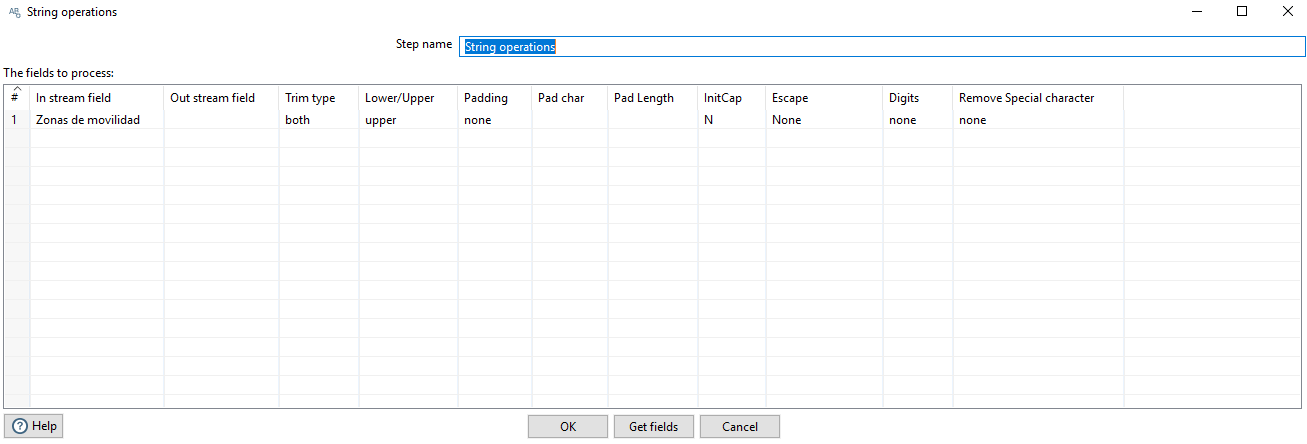


Ilustración 41 - Normalización IN\_MOVILIDAD.

#### 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:

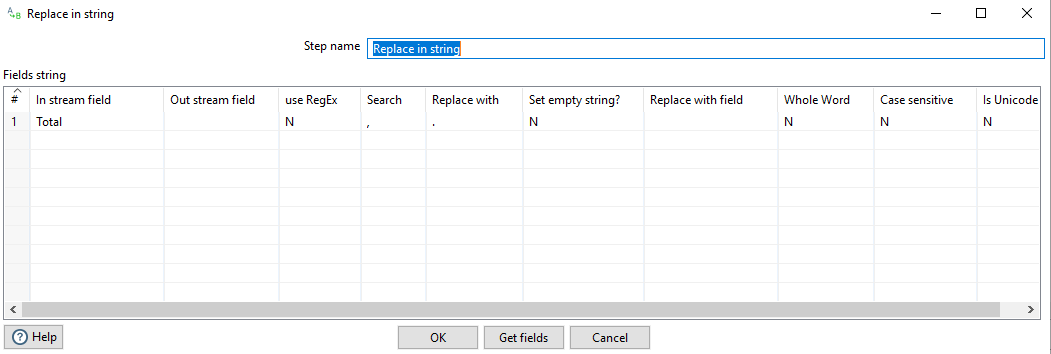


Ilustración 42 - Replace IN\_MOVILIDAD.

#### 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”:

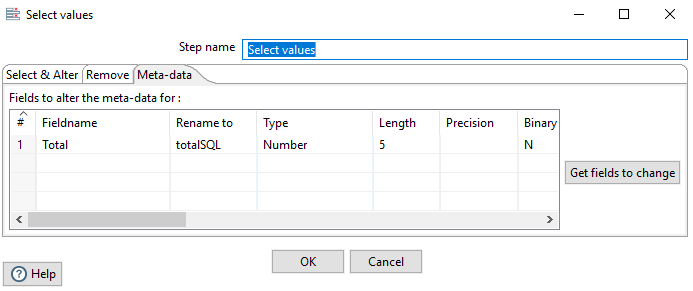


Ilustración 43 - Select Values IN\_MOVILIDAD.

#### 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:

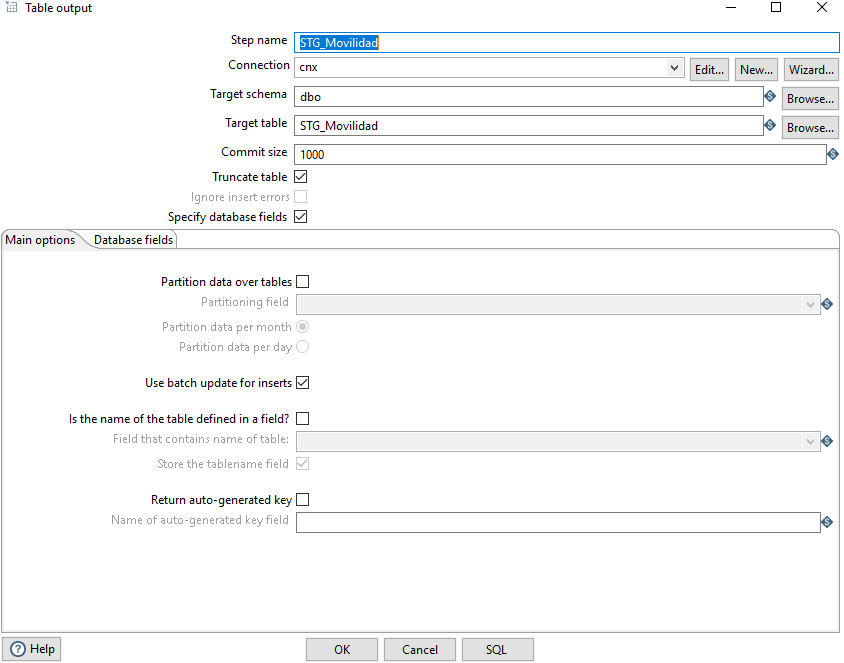


Ilustración 44 - Guardado IN\_MOVILIDAD.

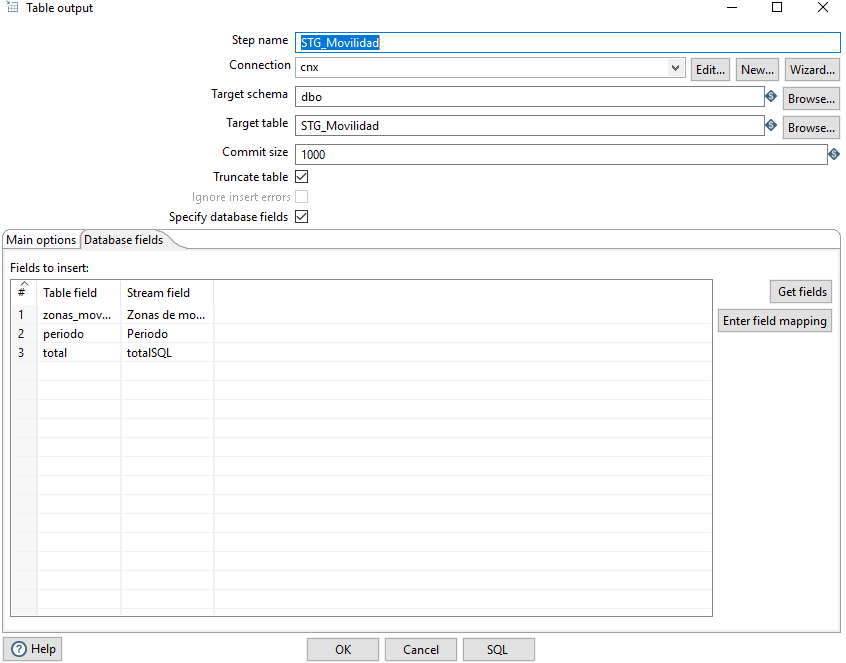


Ilustración 45 - Guardado IN\_MOVILIDAD.

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

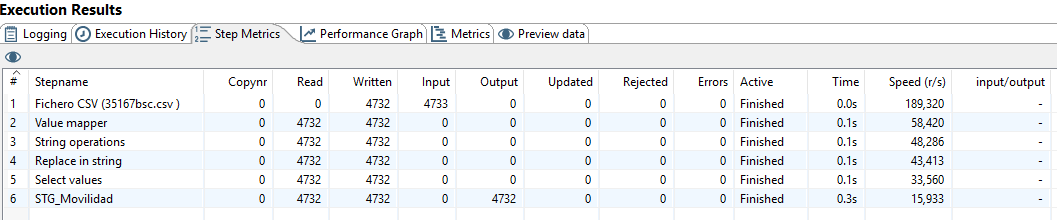


Ilustración 46 - Métricas IN\_MOVILIDAD.

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:

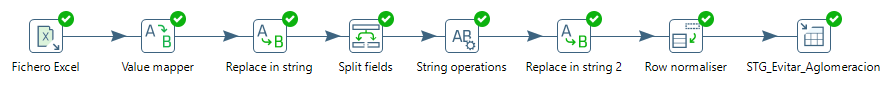


Ilustración 47 - IN\_AGLOMERACION.

#### 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:

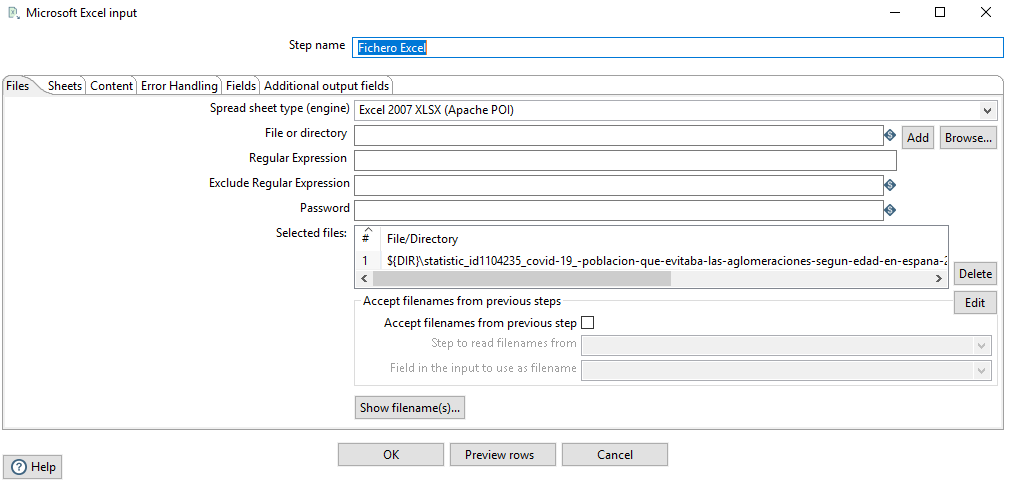


Ilustración 48 - Lectura IN\_AGLOMERACION.

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:

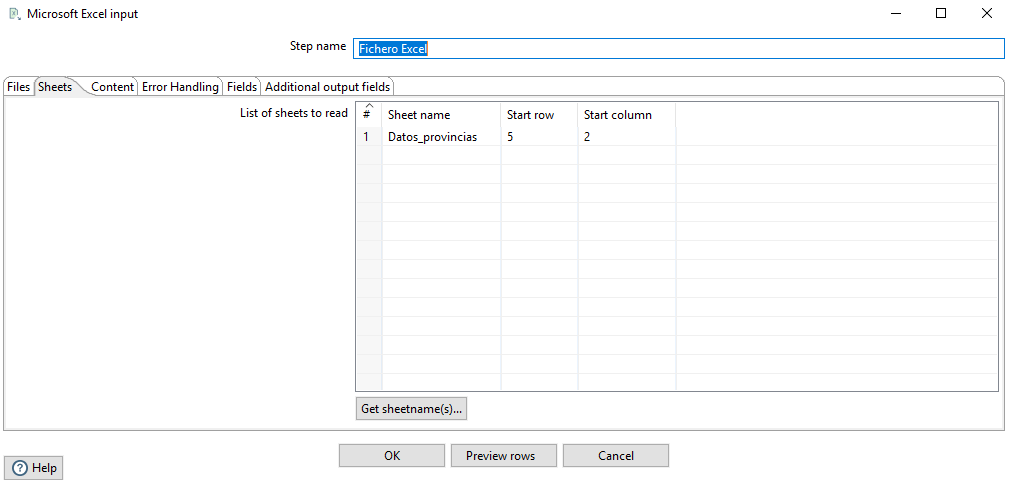


Ilustración 49 - Lectura IN\_AGLOMERACION.

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

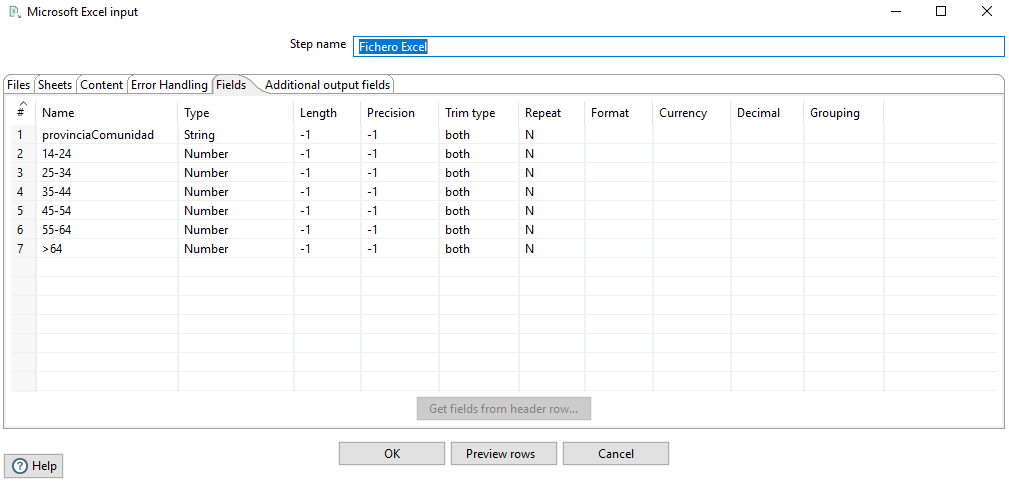


Ilustración 50 - Lectura IN\_AGLOMERACIONES.

#### 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:

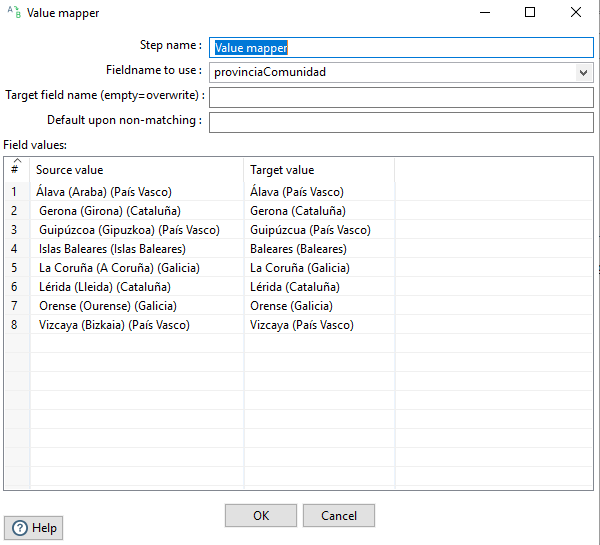


Ilustración 51 - Mapeo Valores IN\_AGLOMERACION.

#### 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:

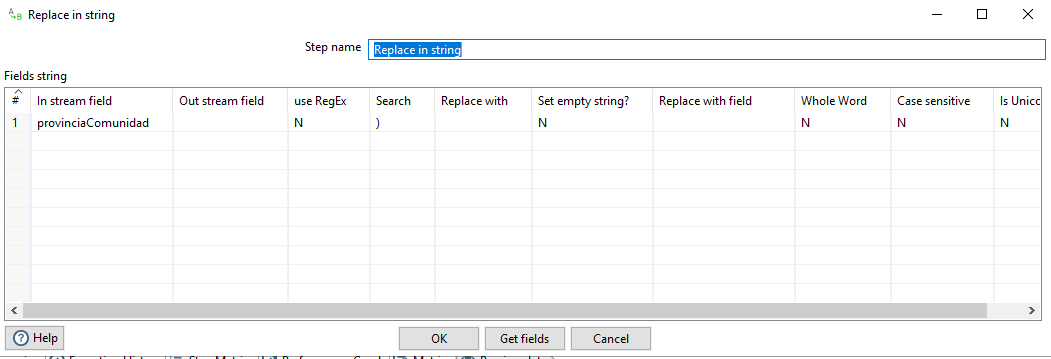


Ilustración 52 - Replace IN\_AGLOMERACION.

#### 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.

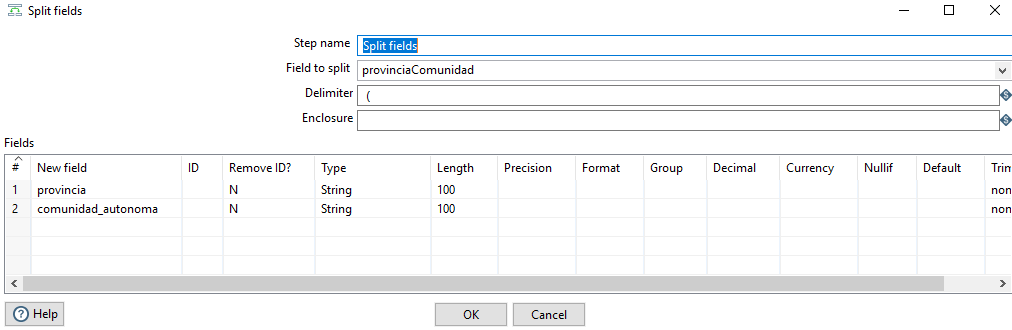


Ilustración 53 - Split IN\_AGLOMERACION.

#### 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:

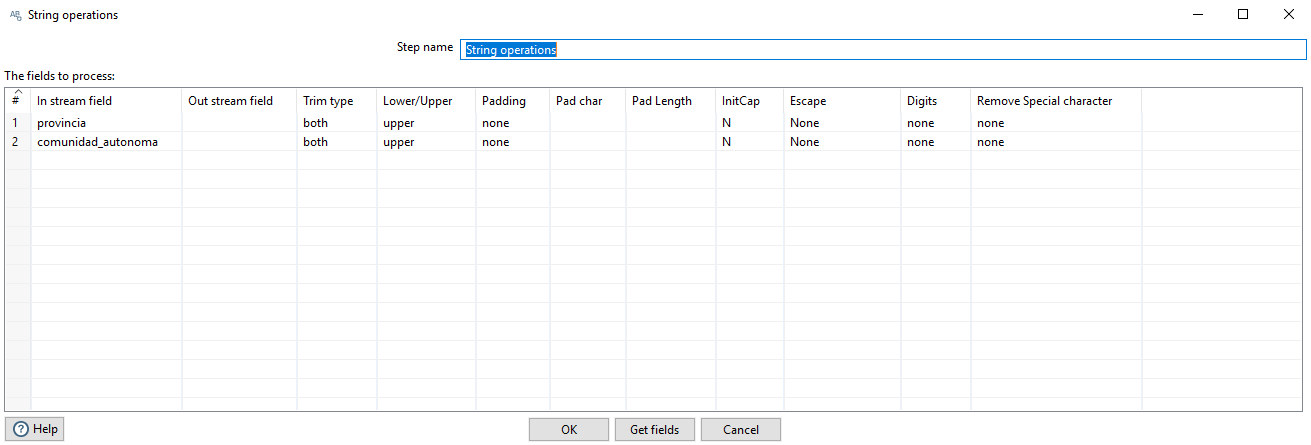


Ilustración 54 - Normalización Strings IN\_AGLOMERACION.

#### 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:

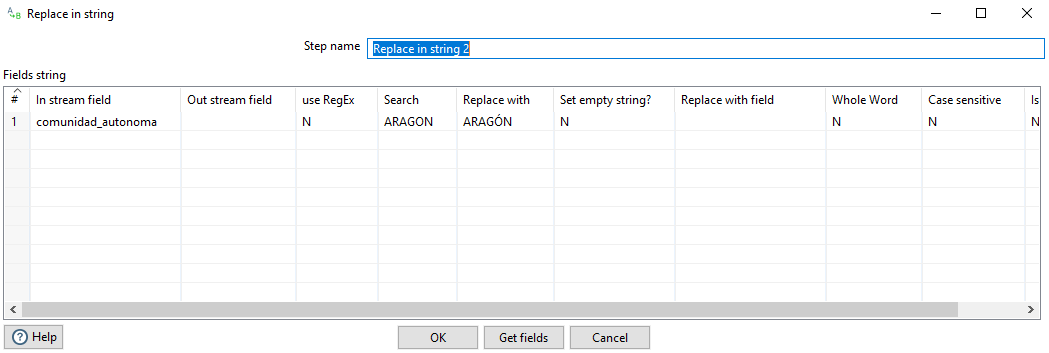


Ilustración 55 - Replace IN\_AGLOMERACION.

#### 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:

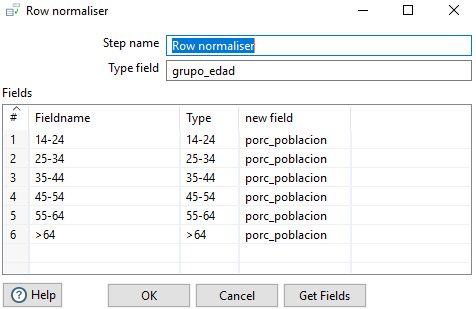


Ilustración 56 - Normalización Filas IN\_AGLOMERACION.

#### 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:

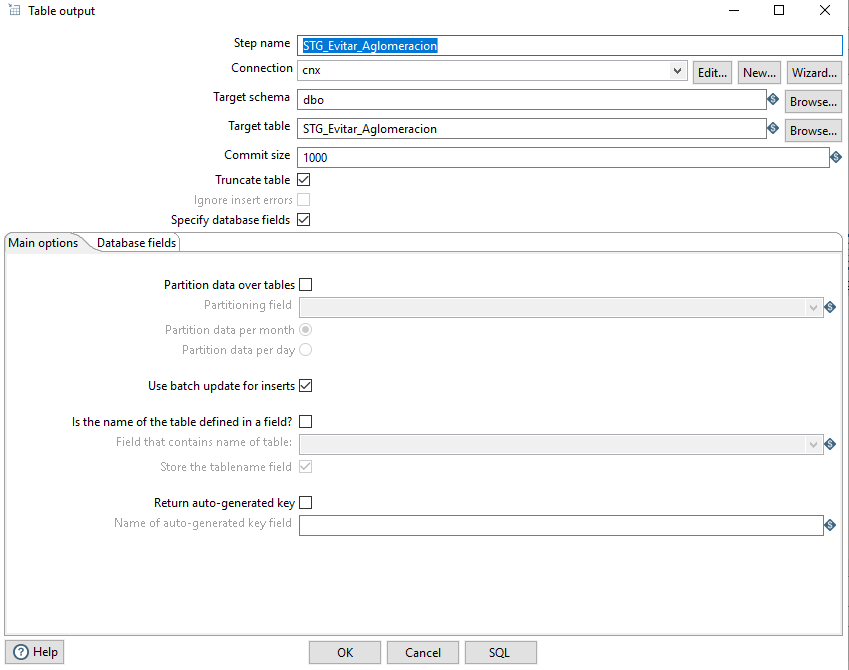


Ilustración 57 - Guardado IN\_AGLOMERACIONES.

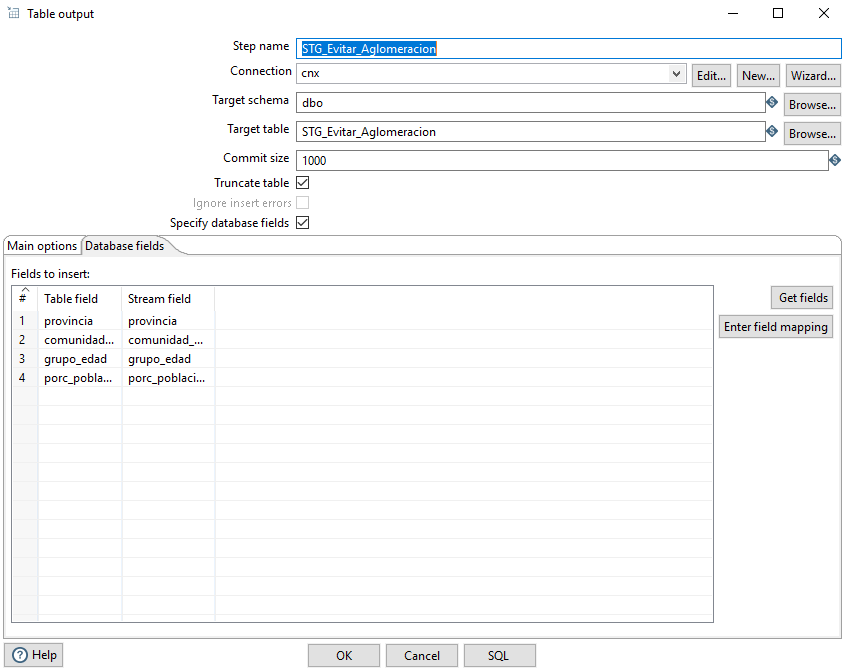


Ilustración 58 - Guardado IN\_AGLOMERACIONES.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

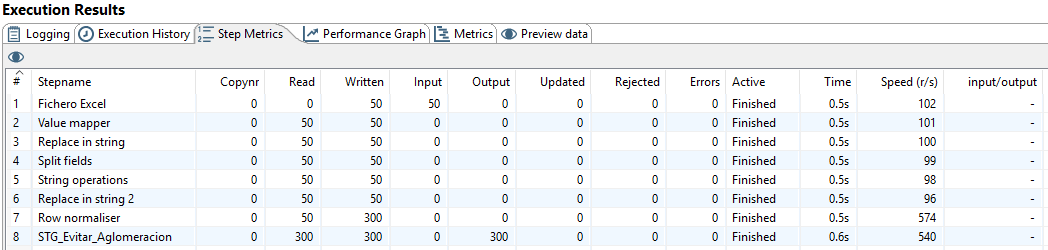


Ilustración 59 - Métricas IN\_AGLOMERACION.

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 quinta 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:

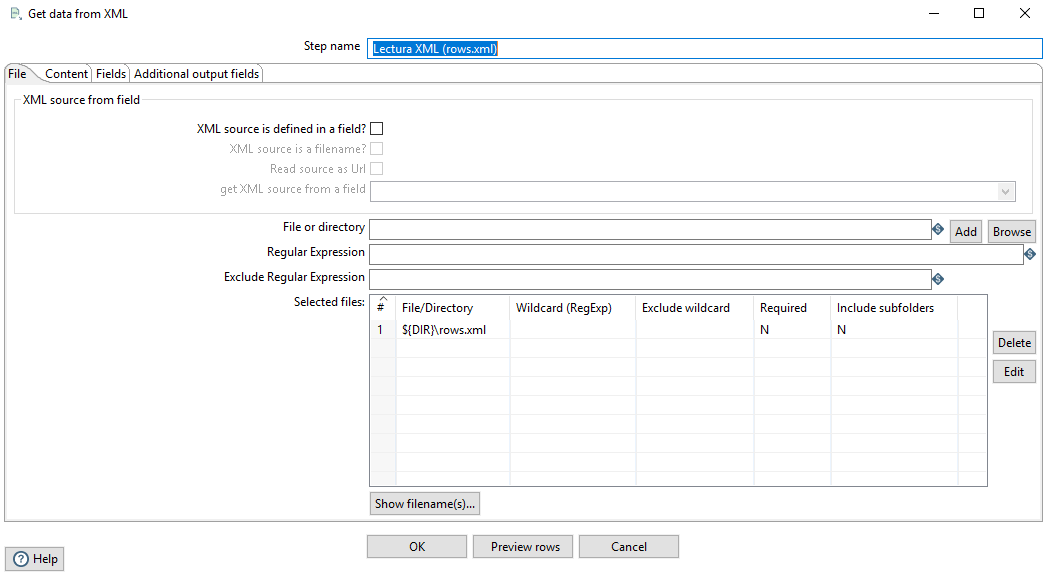


Ilustración - Lectura IN\_LLAMADAS112.

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

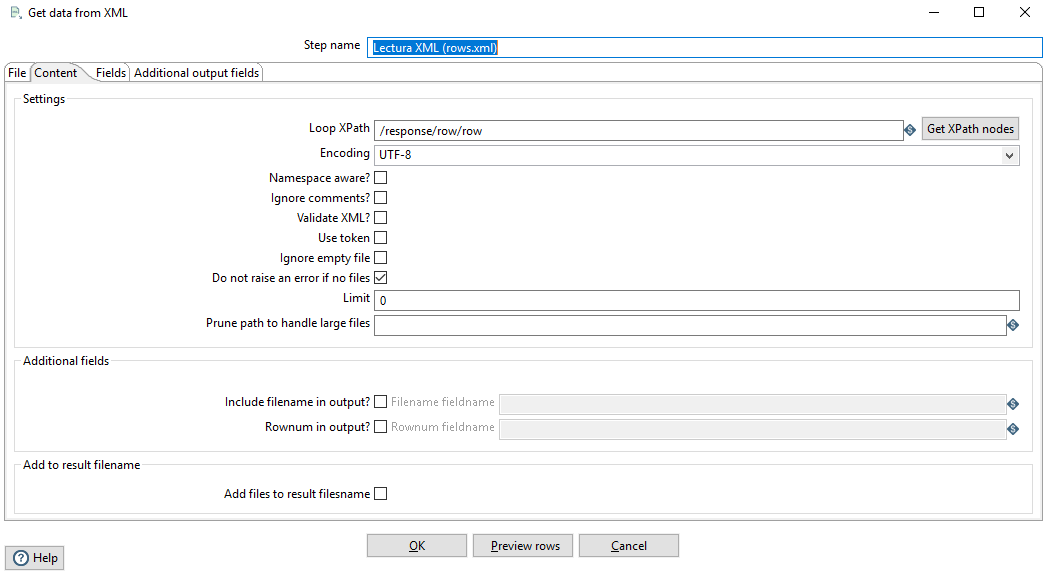


Ilustración - Lectura IN\_LLAMADAS112.

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

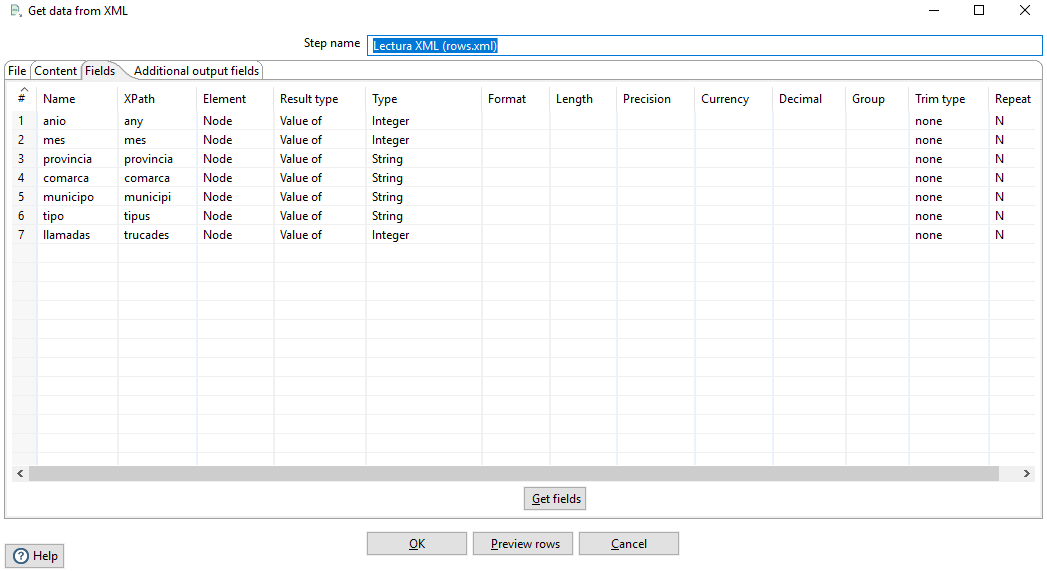


Ilustración - Lectura IN\_LLAMADAS112.

#### 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:

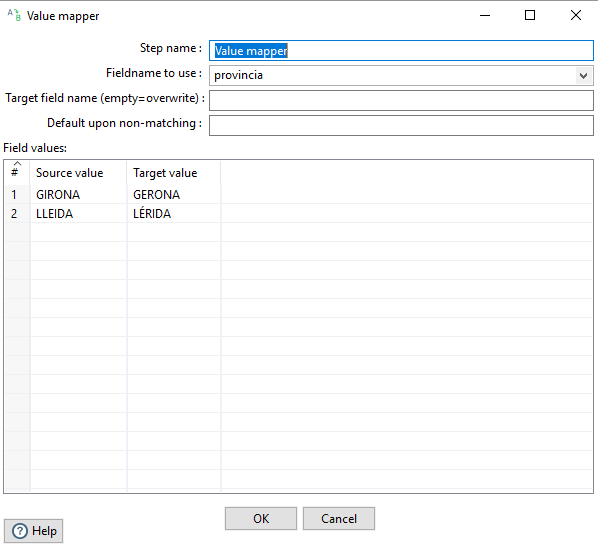


Ilustración - Mapeo Valores IN\_LLAMADAS112.

#### 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:

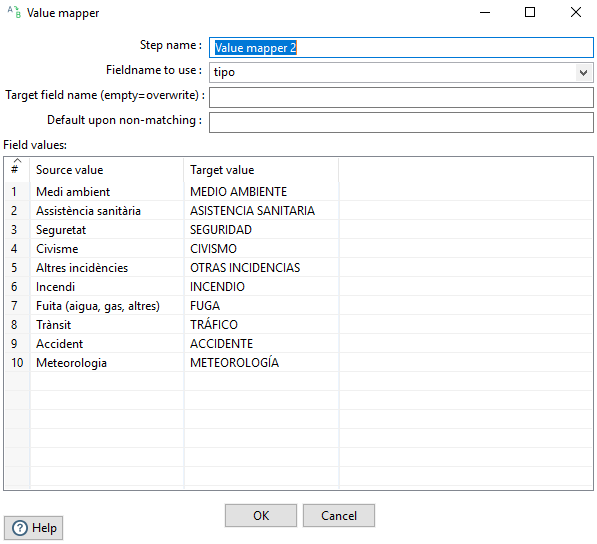


Ilustración - Mapeo Valores IN\_LLAMADAS112.

#### 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:

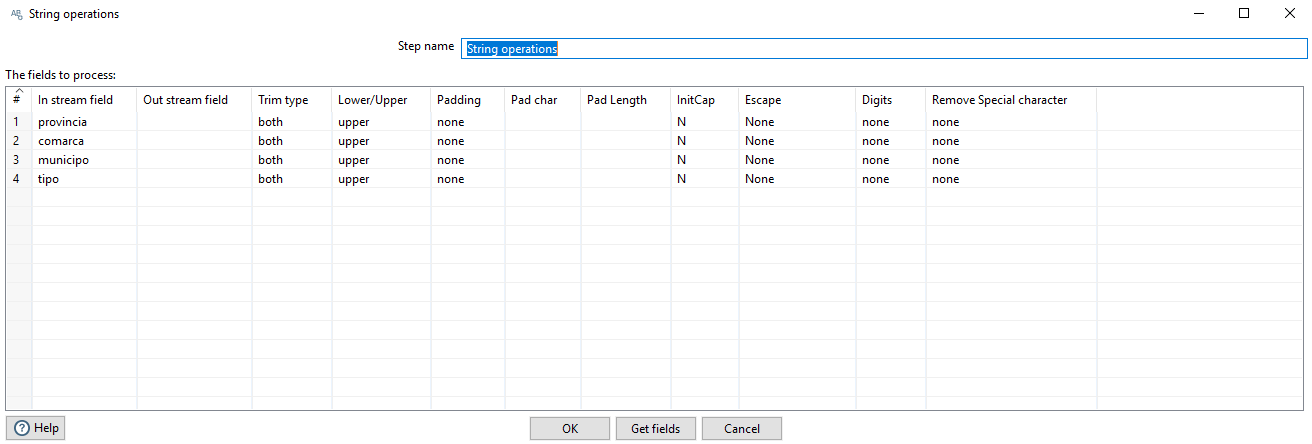


Ilustración - Normalización IN\_LLAMADAS112.

#### Guardado

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

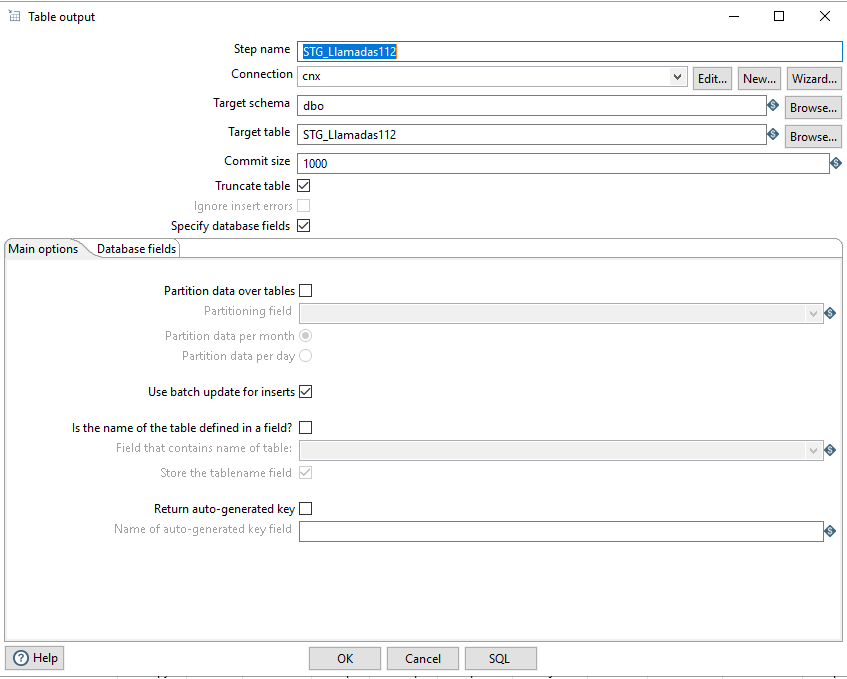


Ilustración - Guardado IN\_LLAMADAS112.

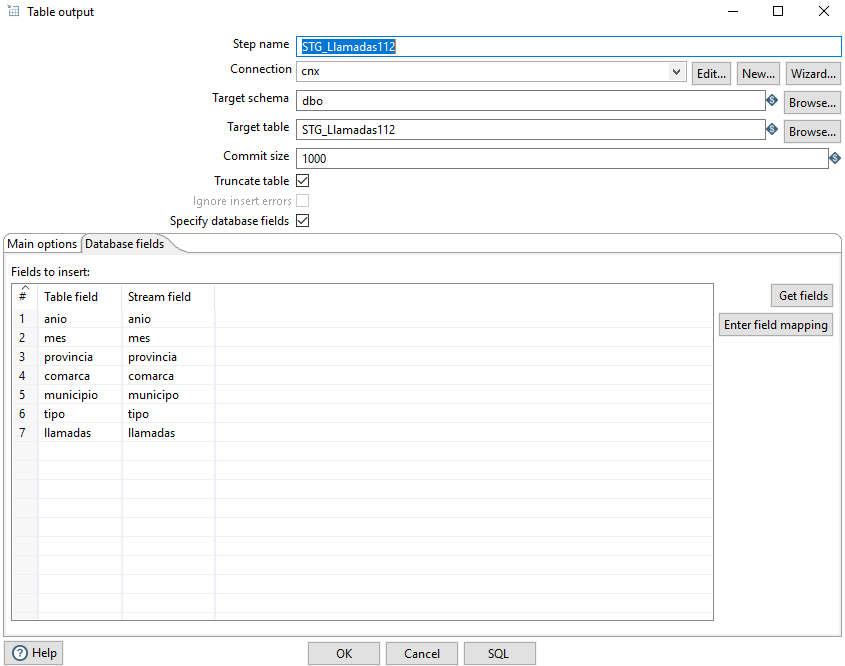


Ilustración - Guardado IN\_LLAMADAS112.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

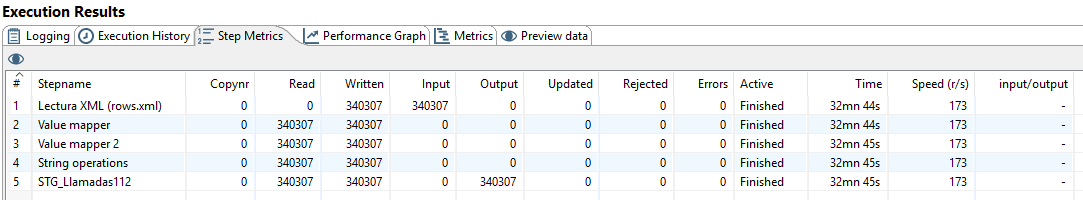


Ilustración - Métricas IN\_LLAMADAS112.

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

### Transformación IN\_FECHAS

La penúltima transformación que vamos a realizar respecto a este bloque es “IN\_FECHAS”, su objetivo es leer todas las fechas que hay en todos los ficheros fuente y almacenarlas en una tabla intermedia llamada “STG\_Fechas”.

La transformación nos ha quedado de la siguiente forma:

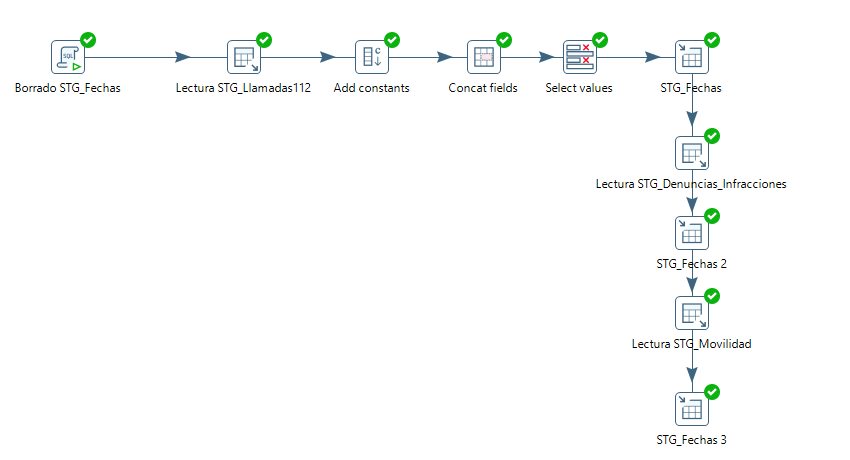


Ilustración 70 - IN\_FECHAS.

#### Borrado

Lo primero de todo es hacer un borrado de la tabla, ya que al no obtener la información directamente de los ficheros puede darse el caso de que ya tengamos información en dicha tabla, por lo tanto borramos todos los registros de forma manual a partir de una sentencia SQL:

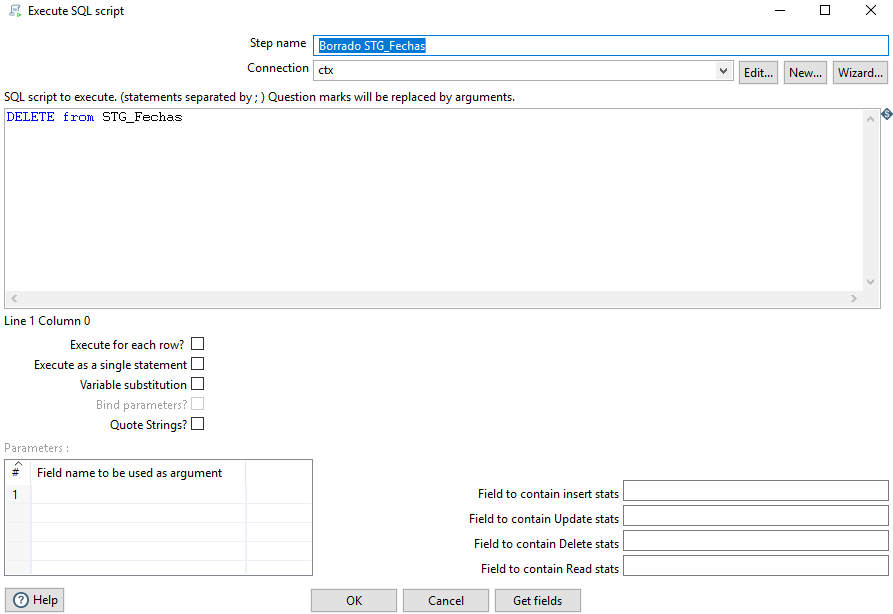


Ilustración 71 - Borrado IN\_FECHAS.

#### Lectura

Una vez eliminados todos los registros leemos todas las fechas que se encuentran en la tabla intermedia “STG\_Llamadas112”, para ello cargamos la información del mes y año tal y como se muestra en la siguiente ilustración:

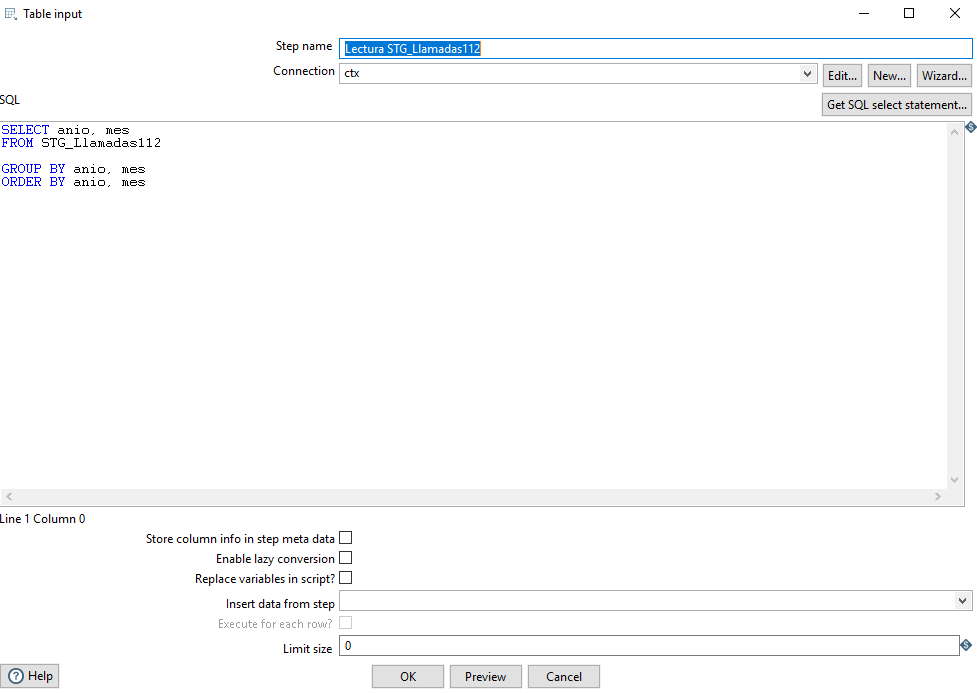


Ilustración 72 - Lectura IN\_FECHAS.

#### Añadimos el día

Una característica de la información de “STG\_Llamadas112” es que sí que se nos proporciona el año y mes pero no el día, es por ello que creamos un nuevo campo para el día cuyo valor va a ser siempre 1:

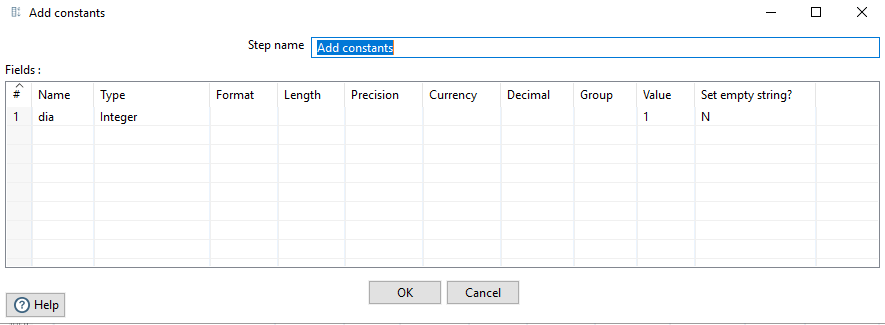


Ilustración 73 - Añadimos Constante IN\_FECHAS.

#### Concatenación

Un aspecto a tener en cuenta es que respecto a “STG\_Llamadas112” no tenemos una fecha como tal, sino que tenemos tres campos de tipo entero que nos indican el año, mes y día. Por lo tanto, lo primero que debemos de hacer es concatenar estos campos en un string:

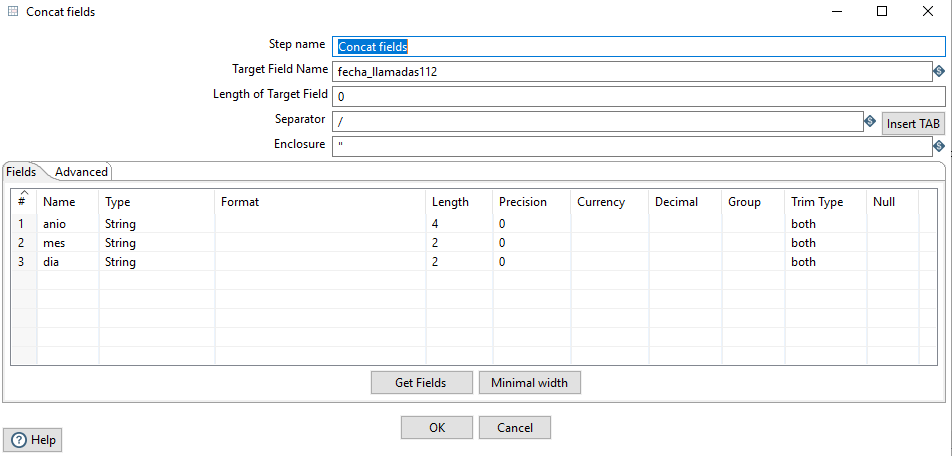


Ilustración 74 - Concatenación IN\_FECHAS.

#### Conversión

Una vez que tenemos el string con el formato de la fecha, tenemos que convertir dicho campo a tipo date, para así poder almacenarlo en la base de datos:

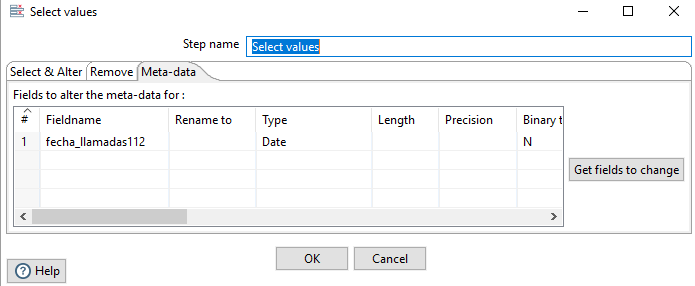


Ilustración 75 - Conversión IN\_FECHAS.

#### Guardado

Finalmente, guardamos todas las fechas de la tabla “STG\_Llamadas112” en la tabla intermedia “STG\_Fechas”, para ello asociamos el campo de la transformación con el de la tabla de la base de datos:

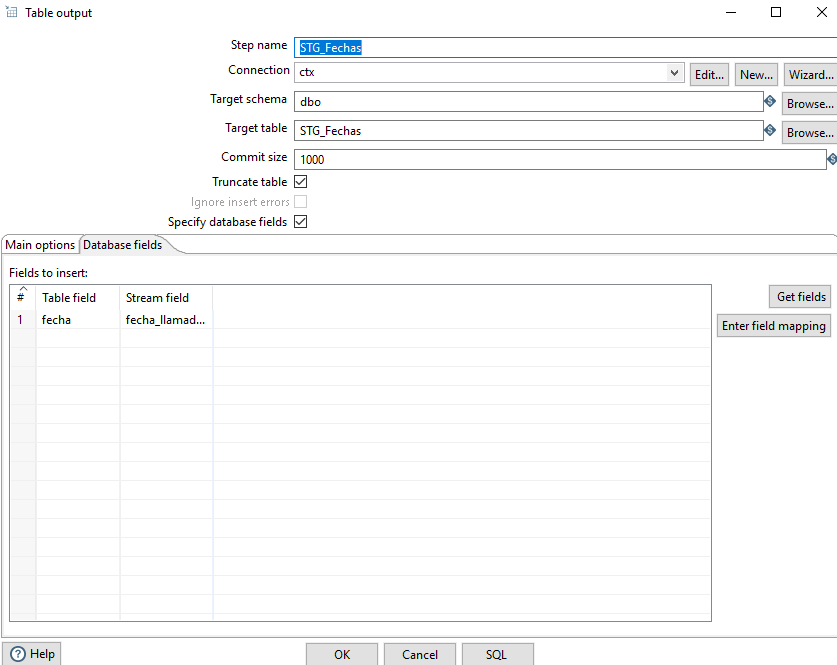


Ilustración 76 - Guardado IN\_FECHAS.

#### Lectura

Al igual que hicimos con las llamadas al 112, tenemos que leer todas las fechas que hay en la tabla “STG\_Denuncias\_Infracciones”, para ello cargamos los datos a partir de la siguiente secuencia SQL:

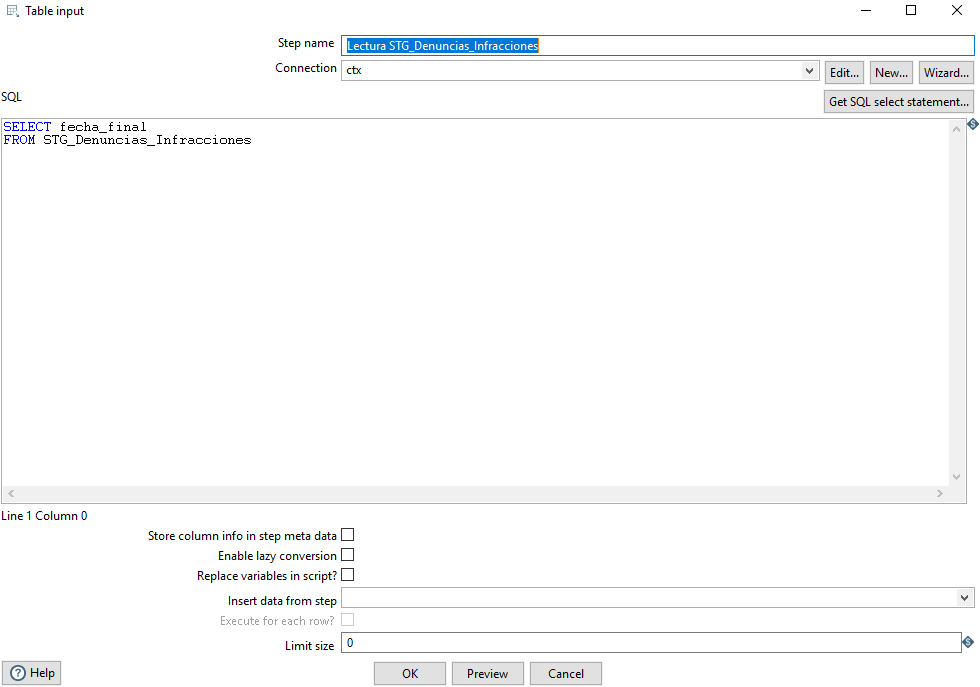


Ilustración 77 - Lectura IN\_FECHAS.

#### Guardado

Como en este caso la tabla “STG\_Denuncias\_Infracciones” ya contiene todas las fechas de forma correcta las podemos guardar directamente en la base de datos, para ello asociamos los campos:

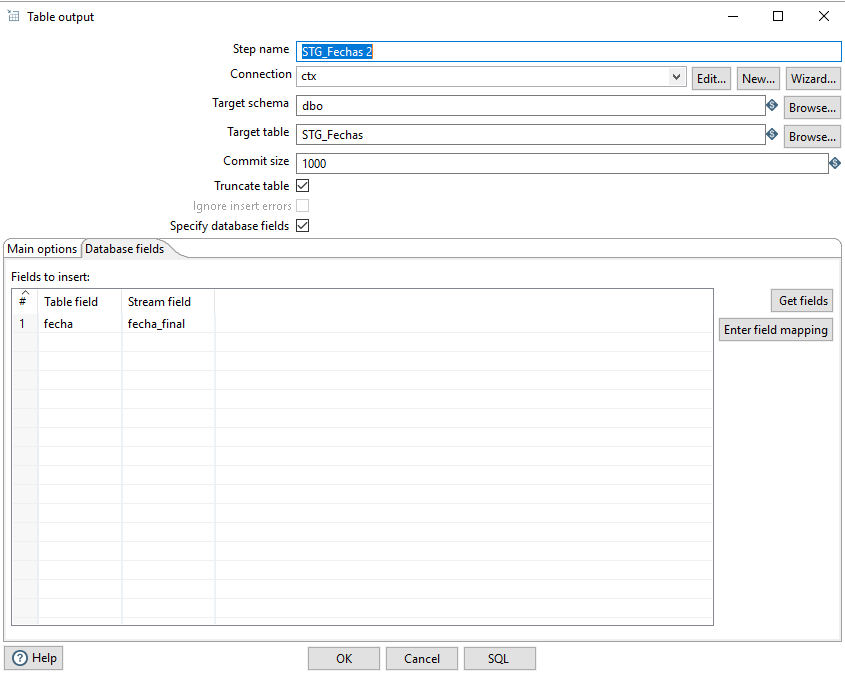


Ilustración 78 - Guardado IN\_FECHAS.

#### Lectura

Por último, tenemos las fechas que se encuentran en “STG\_Movilidad”, éstas las tenemos que leer y lo hacemos al igual que en los casos anterior con una sentencia SQL:

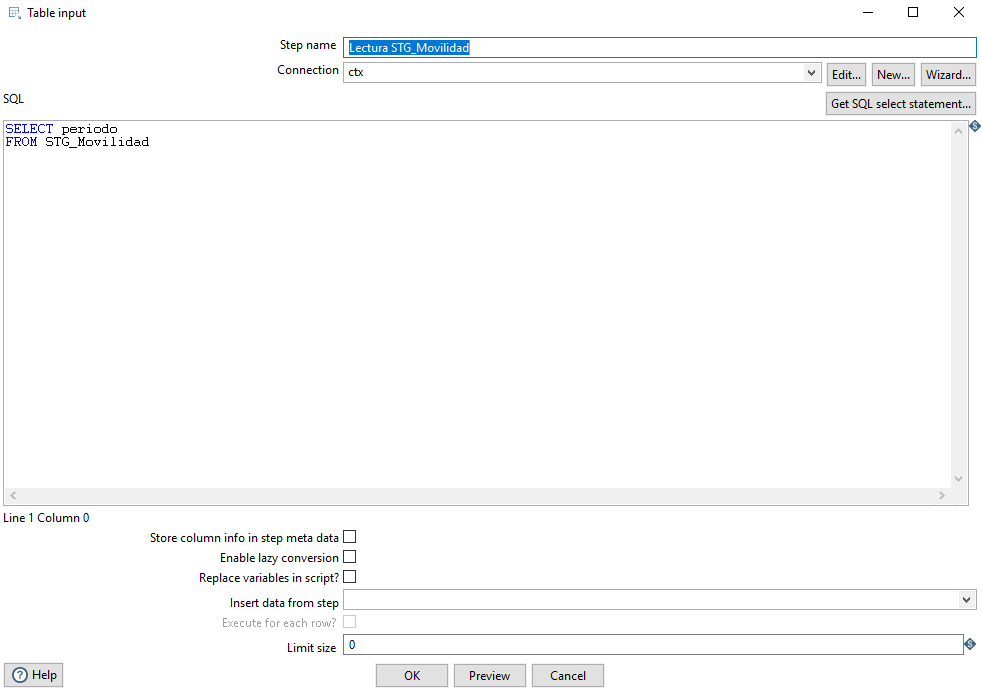


Ilustración 79 - Lectura IN\_FECHAS.

#### Guardado

Al igual que en “STG\_Denuncias\_Infracciones” en “STG\_Movilidad” no necesitamos realizar ninguna transformación, ya que todos los datos están de forma correcta. Debido a esto podemos almacenarlos directamente en la base de datos, y para ello asociamos los campos:

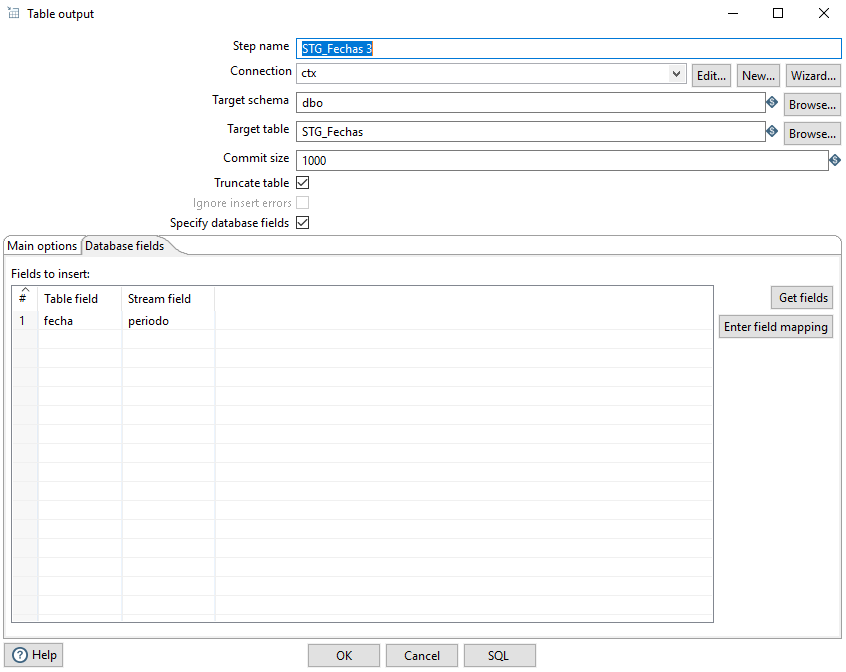


Ilustración 80 - Guardado IN\_FECHAS.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

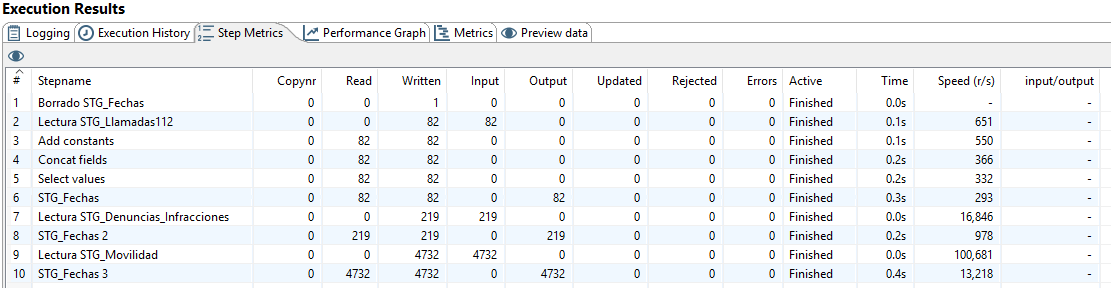


Ilustración 81 - Métricas IN\_FECHAS.

De la anterior ejecución vemos que en “STG\_Llamadas112” tenemos 82 fechas, en “STG\_Denuncias\_Infracciones” hay 219 fechas y en “STG\_Movilidad” 4732. Estas fechas no significan que sean únicas, de hecho todo lo contrario, como veremos más adelante solo 170 fechas son diferentes.

### Transformación IN\_AMBITO\_GEOGRAFICO

La última transformación que vamos a realizar respecto a este bloque es “IN\_AMBITO\_GEOGRAFICO”, su objetivo es leer y almacenar en “STG\_Ambito\_Geografico” todas las localizaciones que existen en los ficheros proporcionados.

La transformación nos ha quedado de la siguiente forma:

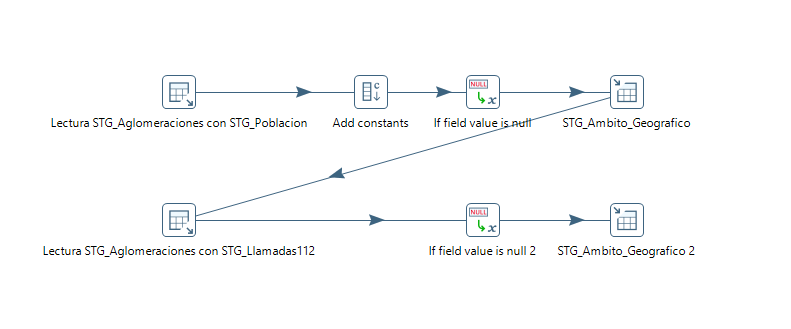


Ilustración - IN\_AMBITO\_GEOGRAFICO.

#### Lectura

Lo primero que vamos a hacer es un join entre las tablas “STG\_Evitar\_Aglomeracion” y “STG\_Poblacion”, de esta forma conseguimos obtener el código de la provincia, su nombre y su comunidad respecto a todas las provincias españolas. Como el campo en común entre las dos tablas es el nombre de la provincia, usamos dicho campos para hacer el join:

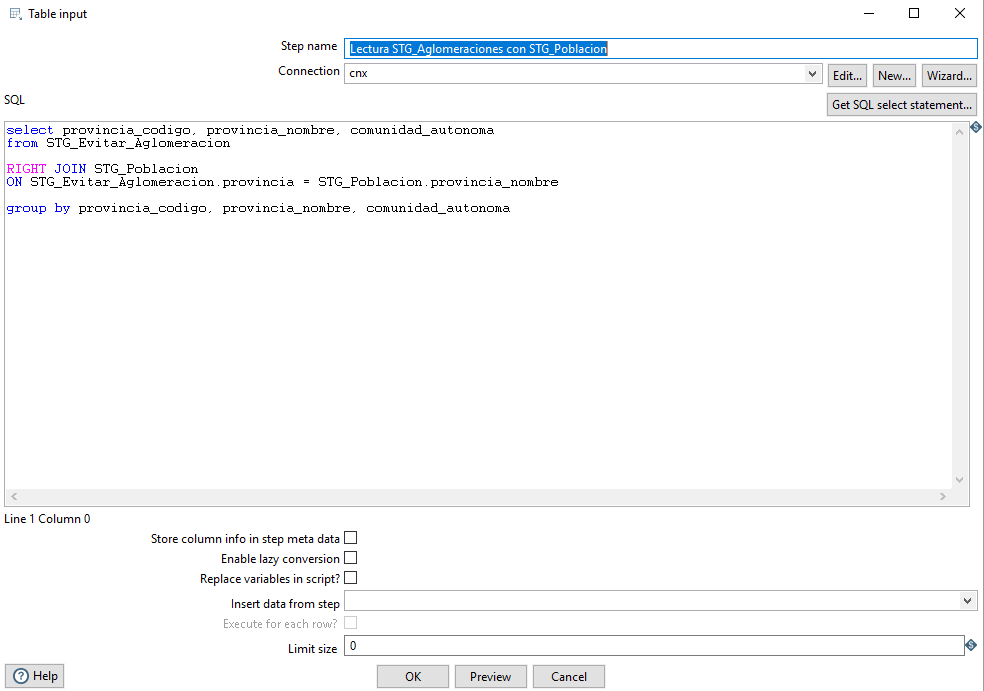


Ilustración - Lectura IN\_AMBITO\_GEOGRAFICO.

#### Añadimos la comarca y el municipio

Debido a que en la dimensión ámbito geográfico tenemos que almacenar tanto la comarca y el municipio, debemos de crear dos atributos cuyos valores van a ser constantes “NA”. Esto se debe a que al hacer el join de las dos tablas anteriores carecemos de dichos campos, por lo tanto en Spoon nos queda de la siguiente forma:

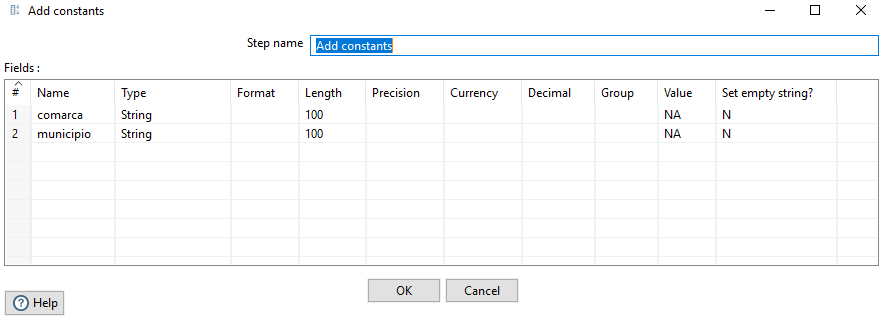


Ilustración - Constatnes Comarca Municipio IN\_AMBITO\_GEOGRAFICO.

#### Replace

Una vez que ya tenemos todos los datos puede darse el caso por ejemplo de que haya provincias sin comunidades, Ceuta y Melilla son ciudades autónomas pero no comunidades, por lo que vamos a reemplazar por “NA” aquellos valores que son nulos:

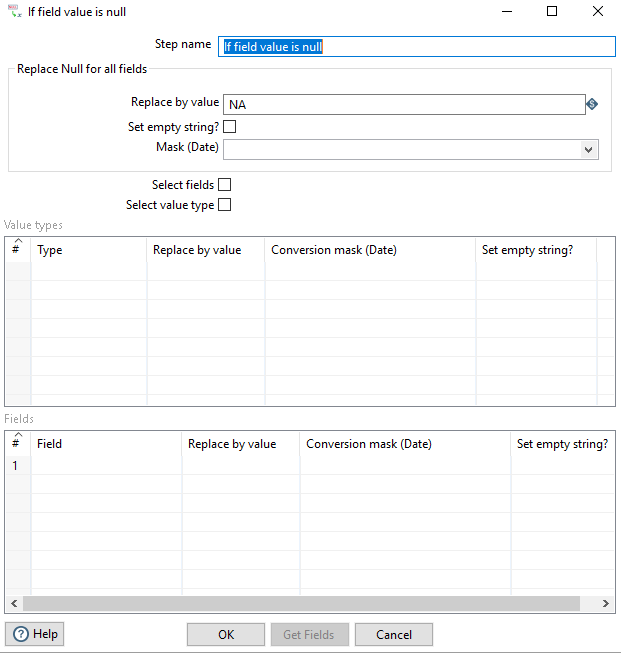


Ilustración - Replace IN\_AMBITO\_GEOGRAFICO.

#### Guardado

Finalmente, respecto a los datos de estas dos tablas los podemos guardar directamente a la base de datos, para ello indicamos la tabla “STG\_Ambito\_Geografico” y asociamos los valores:

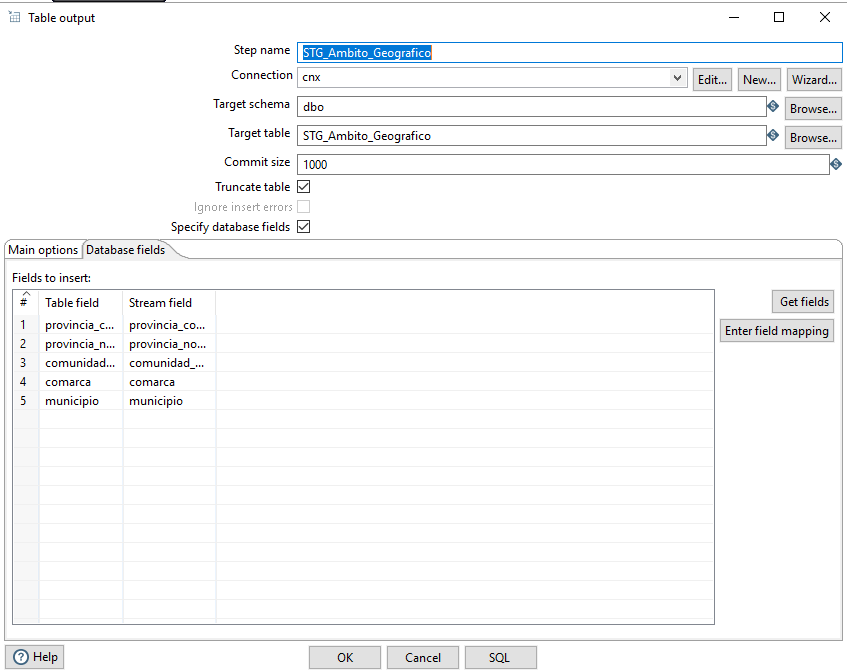


Ilustración - Guardado IN\_AMBITO\_GEOGRAFICO.

#### Lectura

Con los procesos anteriores conseguimos almacenar las provincias de España de forma general, sin embargo hay más localizaciones que se encuentran en las llamadas al 112, ya que en este fichero tenemos los campos comarca y municipio. Por lo tanto, hacemos un join entre “STG\_Evitar\_Aglomeracion”, “STG\_Llamadas112” y “STG\_Población”, de esta forma conseguimos el código de la provincia, su nombre, su comunidad autónoma, su comarca y su municipio:

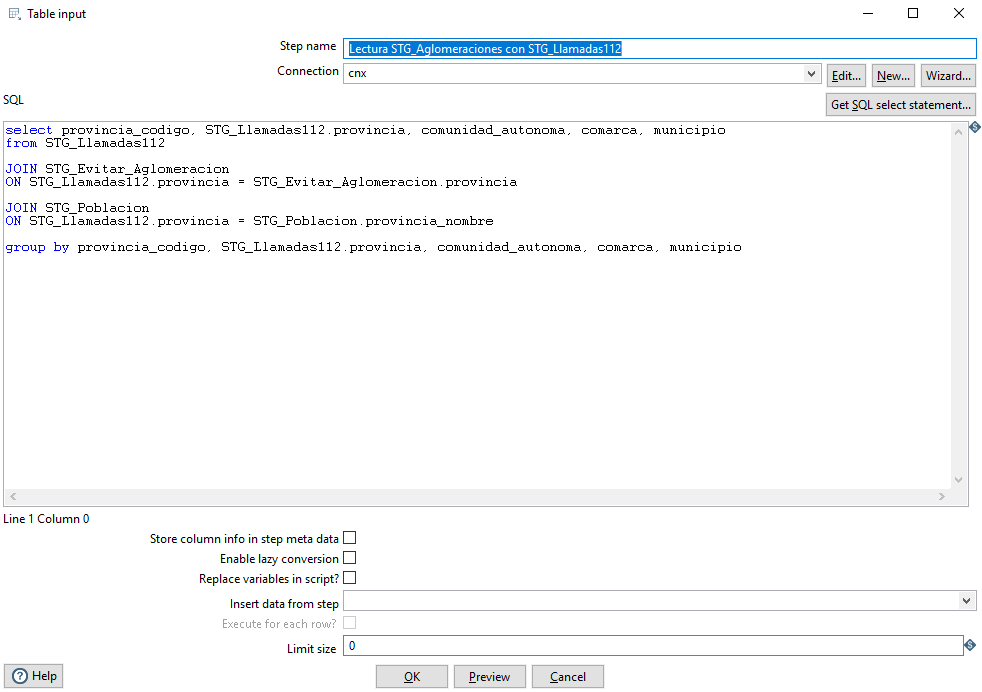


Ilustración - Lectura IN\_AMBITO\_GEOGRAFICO.

#### Replace

Al igual que antes puede darse el caso de que haya valores nulos en determinados atributos, esto lo solucionamos reemplazando los nulos por “NA”:

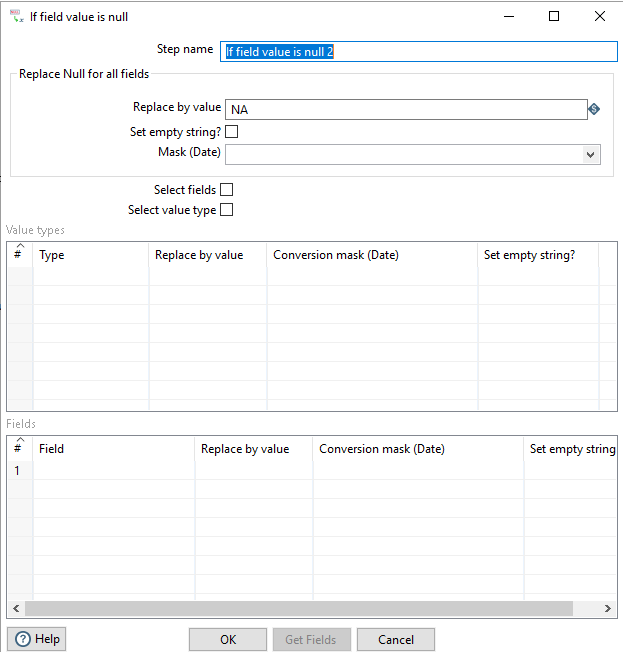


Ilustración - Replace IN\_AMBITO\_GEOGRAFICO.

#### Guardado

Finalmente, ya tenemos todos los datos de forma correcta y los podemos guardar en la tabla “STG\_Ambito\_Geografico”. Por lo tanto, asociamos los atributos y ejecutamos la transformación:

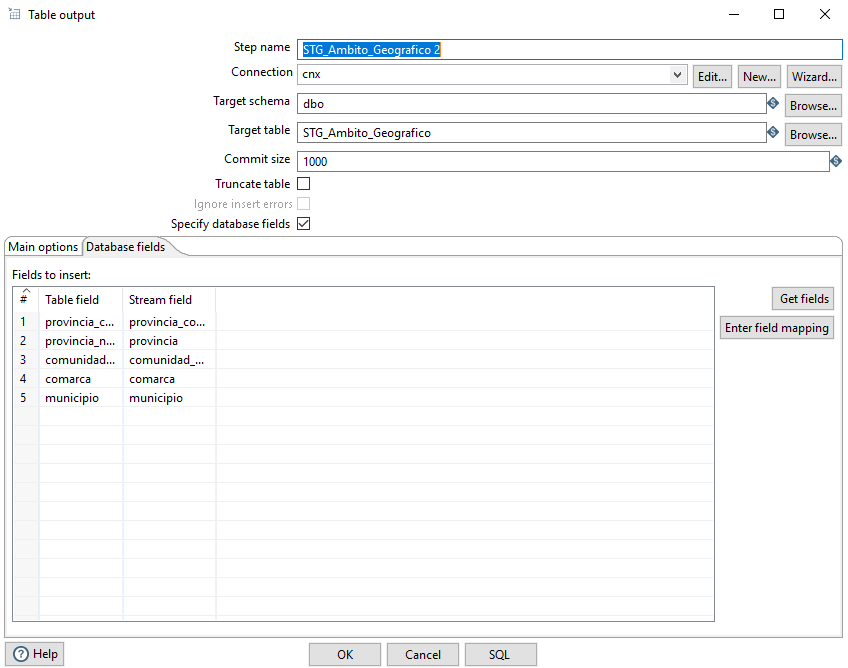


Ilustración - Guardado IN\_AMBITO\_GEOGRAFICO.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

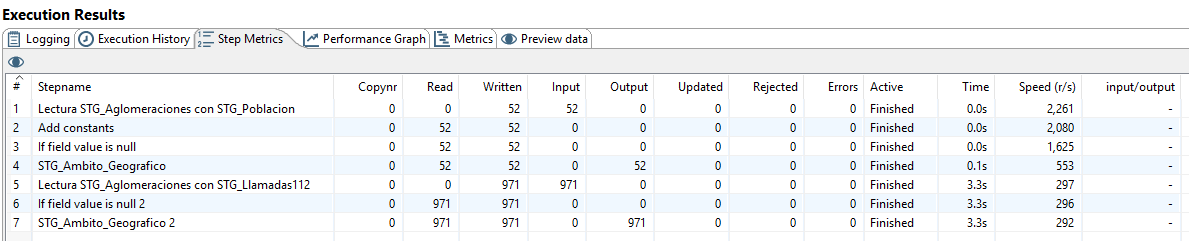


Ilustración - Métricas IN\_AMBITO\_GEOGRAFICO.

Como podemos apreciar de la primera lectura leemos 52 registros, uno por cada provincia española, y de la segunda 971 registros.

## 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:

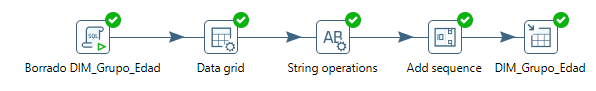


Ilustración - TR\_DIM\_GRUPO\_EDAD.

#### 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:

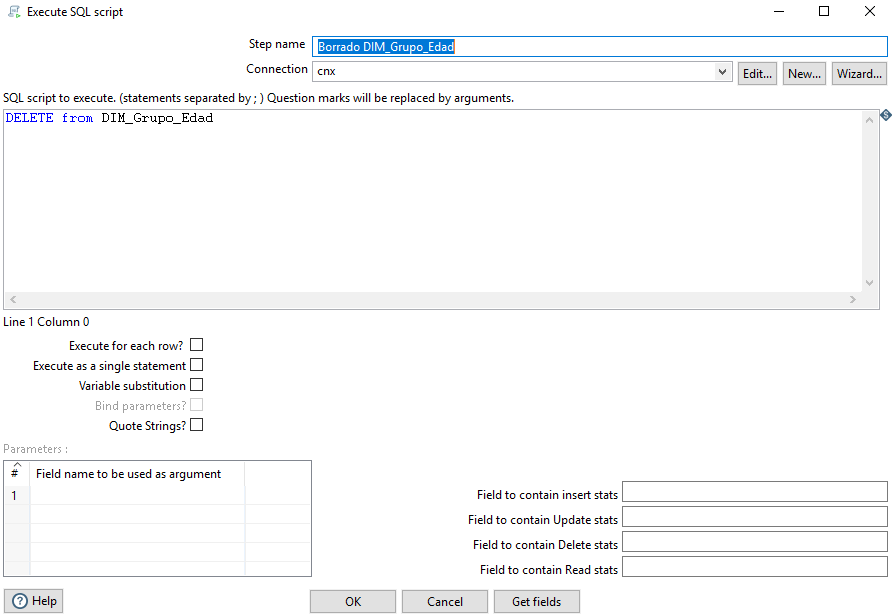


Ilustración - Borrado TR\_DIM\_GRUPO\_EDAD.

#### 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:

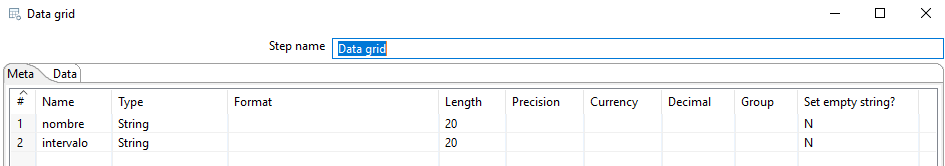


Ilustración - Grid TR\_DIM\_GRUPO\_EDAD.

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:

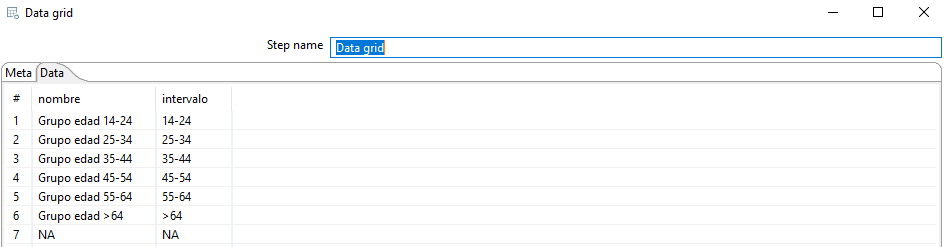


Ilustración - Grid TR\_DIM\_GRUPO\_EDAD.

#### Normalización

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

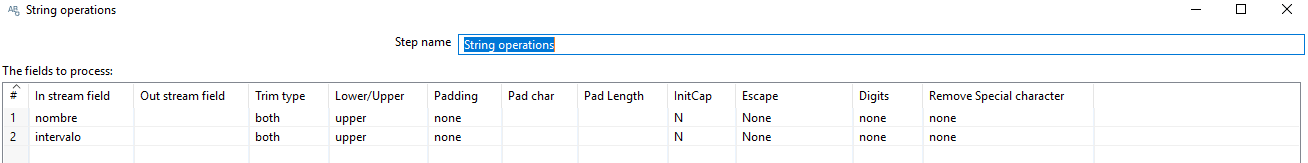


Ilustración - Normalización TR\_DIM\_GRUPO\_EDAD.

#### 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:

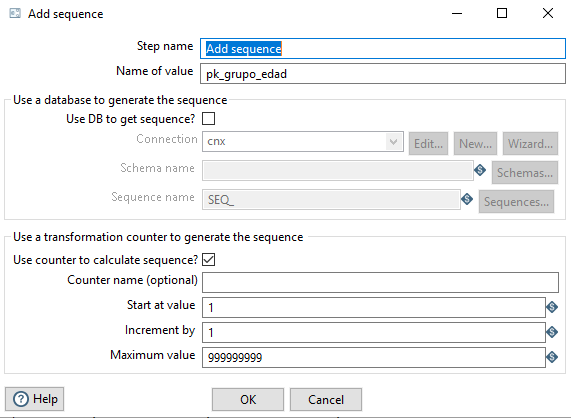


Ilustración - Secuenciación TR\_DIM\_GRUPO\_EDAD.

#### Guardado

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

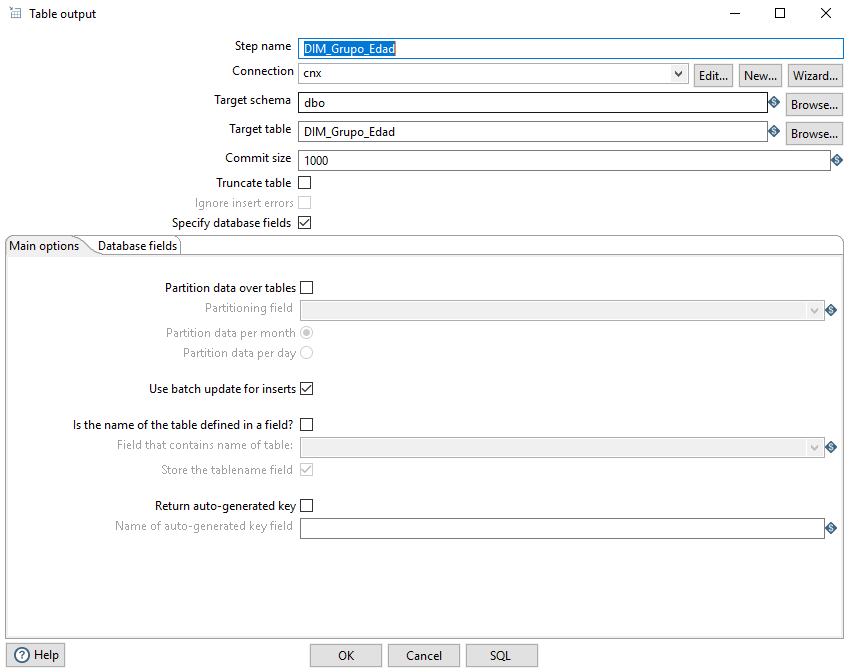


Ilustración - Guardado TR\_DIM\_GRUPO\_EDAD.

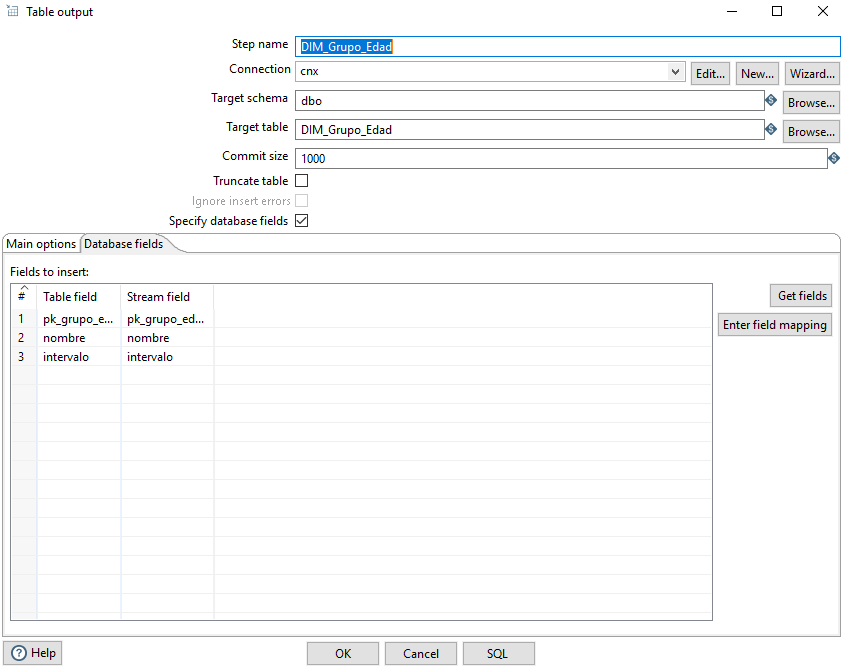


Ilustración - Guardado TR\_DIM\_GRUPO\_EDAD.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

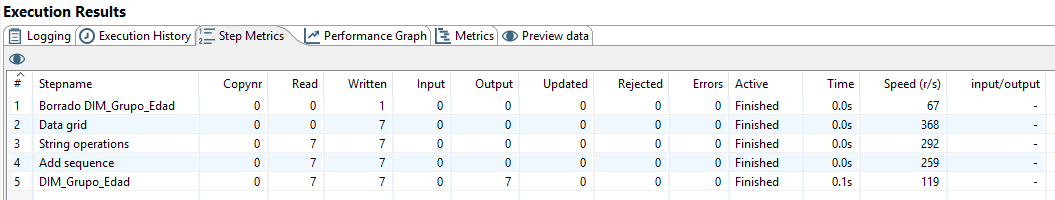


Ilustración - Métricas TR\_DIM\_GRUPO\_EDAD.

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:

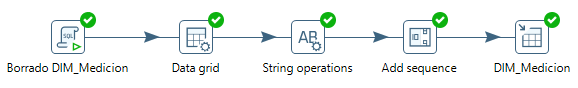


Ilustración - TR\_DIM\_MEDICION.

#### Borrado

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

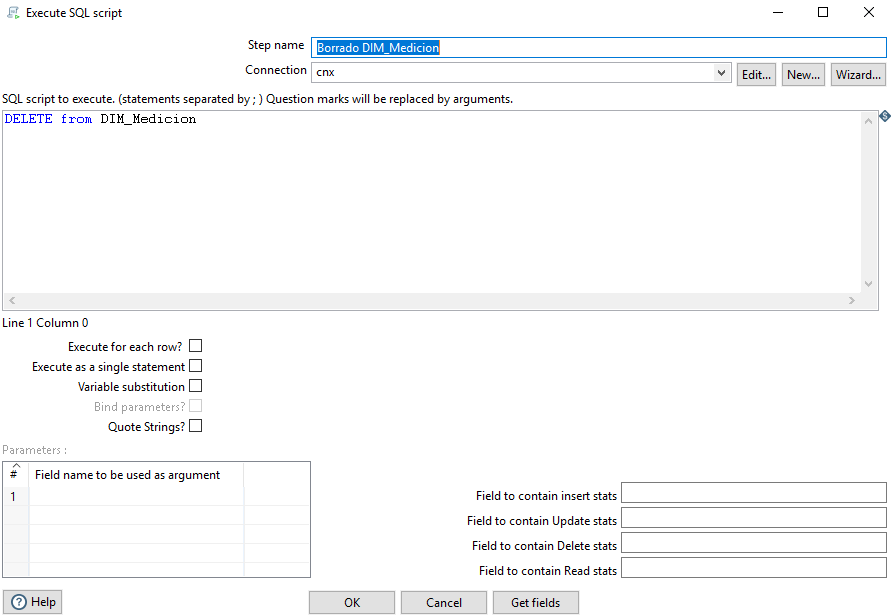


Ilustración - Borrado TR\_DIM\_MEDICION.

#### 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:

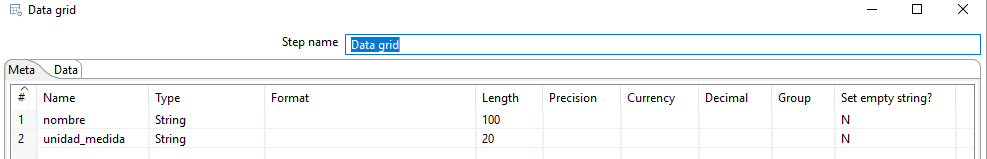


Ilustración - Grid TR\_DIM\_DIM\_MEDICION.

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

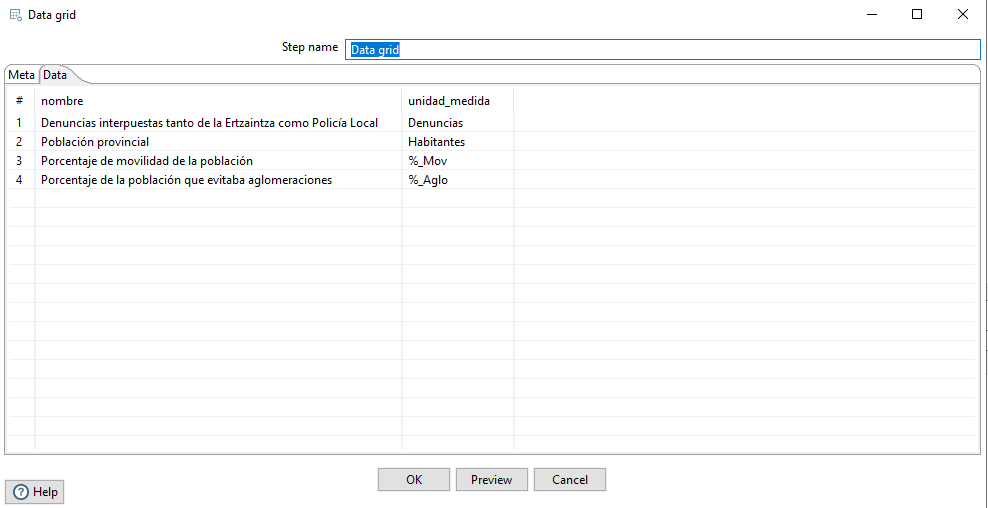


Ilustración - Grid TR\_DIM\_MEDICION.

#### Normalización

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

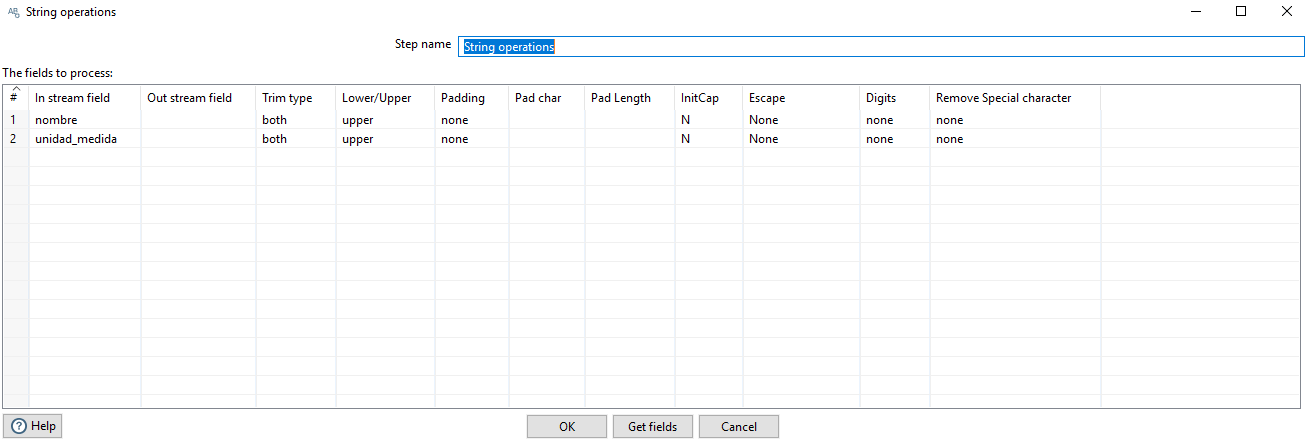


Ilustración - Normalización TR\_DIM\_MEDICION.

#### Secuenciación

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

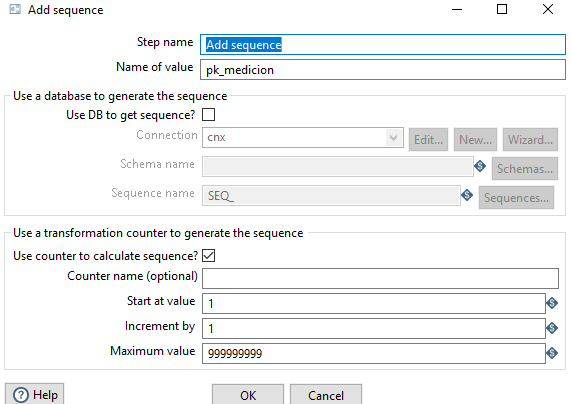


Ilustración - Secuenciación TR\_DIM\_MEDICION.

#### 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”:

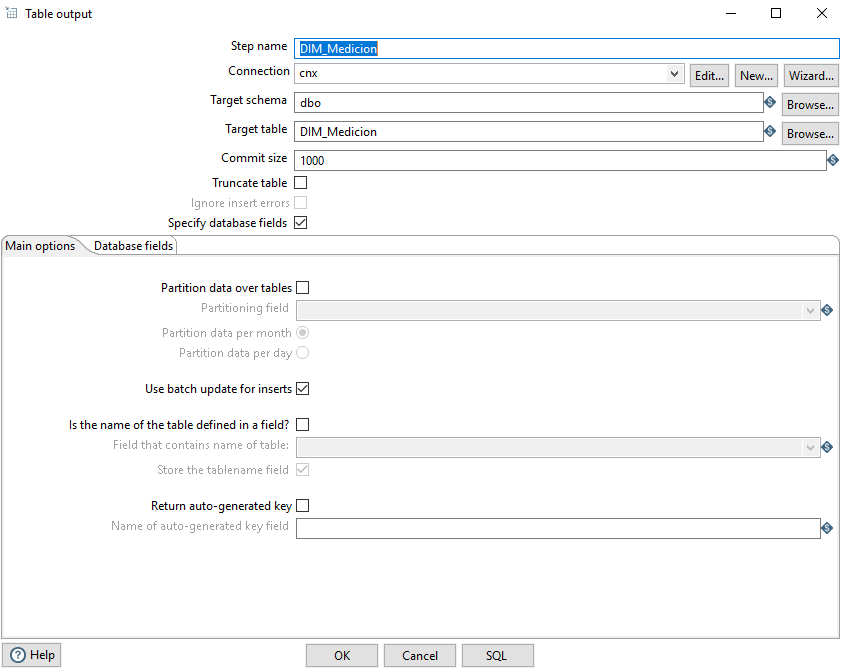


Ilustración - Guardado TR\_DIM\_MEDICION.

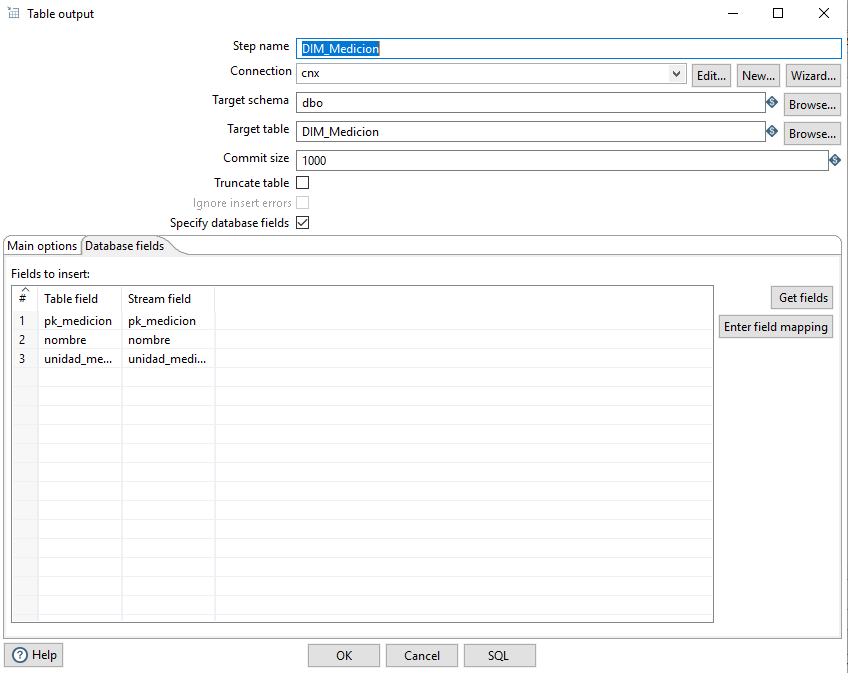


Ilustración - Guardado TR\_DIM\_MEDICION.

Al ejecutar la anterior transformación nos proporciona las siguientes métricas:

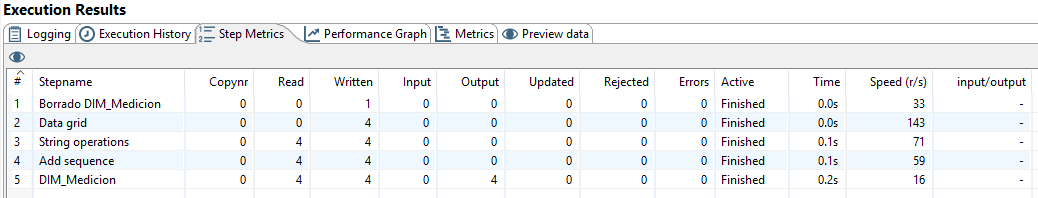


Ilustración - Métricas TR\_DIM\_MEDICIONES.

Como podemos observar generamos los 4 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:

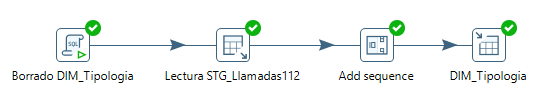


Ilustración - TR\_DIM\_TIPOLOGIA.

#### 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:

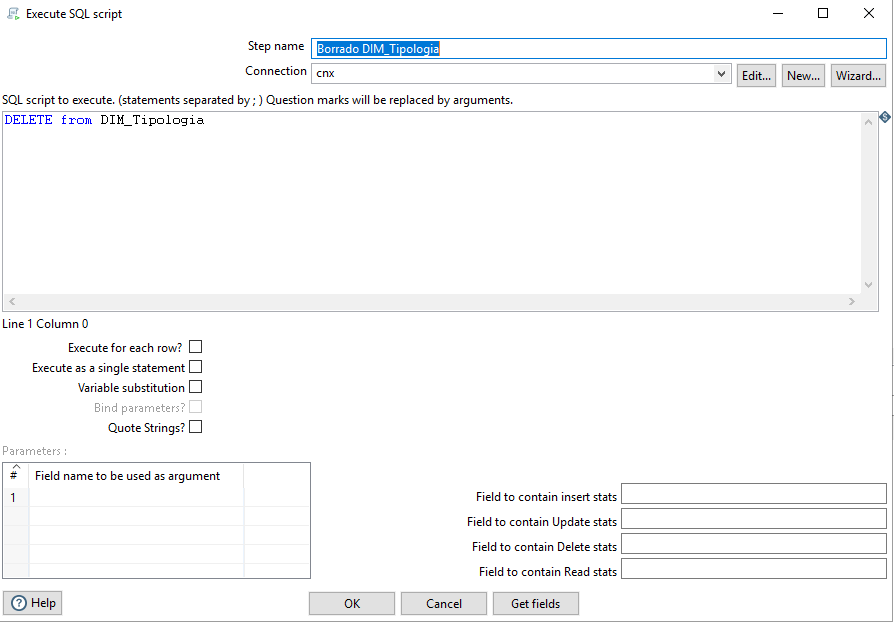


Ilustración - Borrado TR\_DIM\_TIPOLOGIA.

#### Lectura

Aunque en este caso hay también pocas tipologías, para ser más exactos hay 10, podríamos 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:

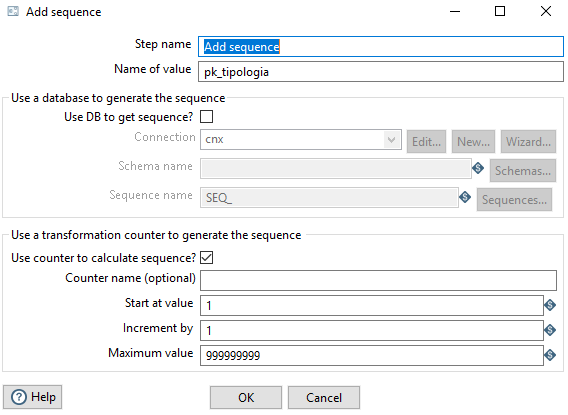


Ilustración - Secuenciación TR\_DIM\_TIPOLOGIA.

#### Guardado

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

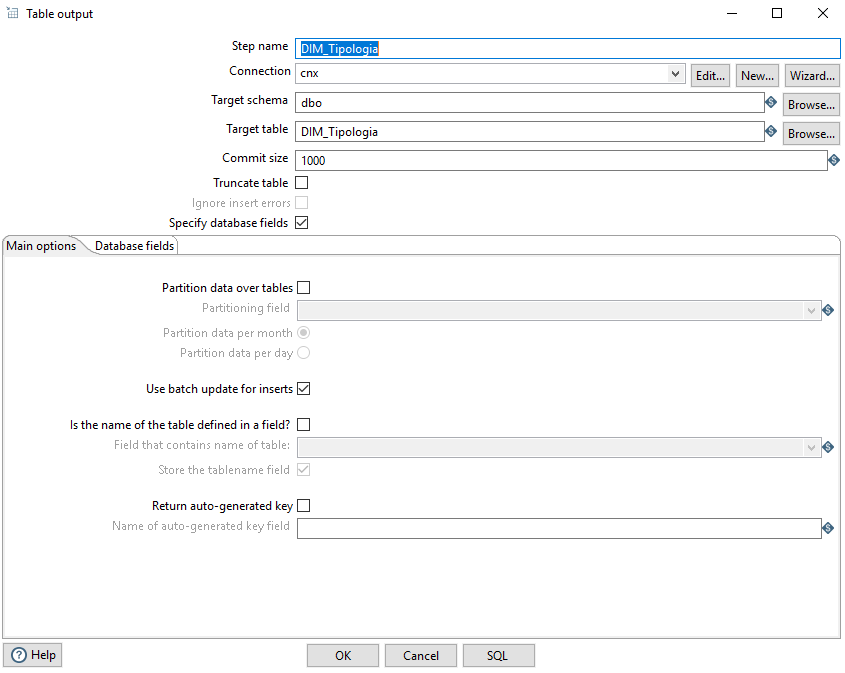


Ilustración - Guardado TR\_DIM\_TIPOLOGIA.

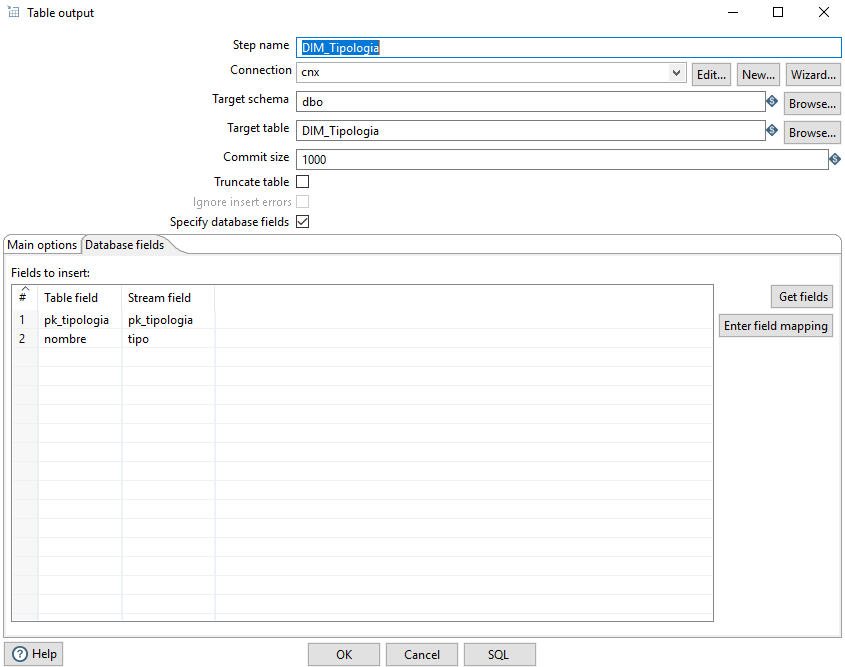


Ilustración - Guardado TR\_DIM\_TIPOLOGIA.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

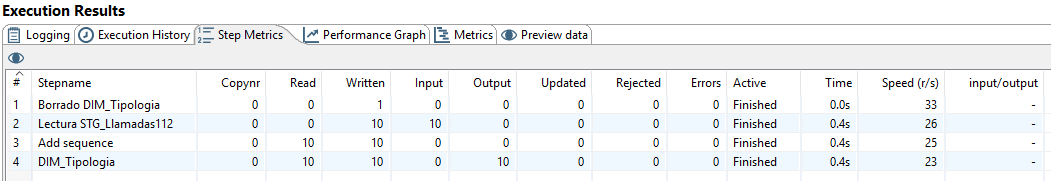


Ilustración - Métricas TR\_DIM\_TIPOLOGIA.

### 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, cabe destacar que al estar relacionado con los hechos debemos primero eliminar los registros de los hechos:

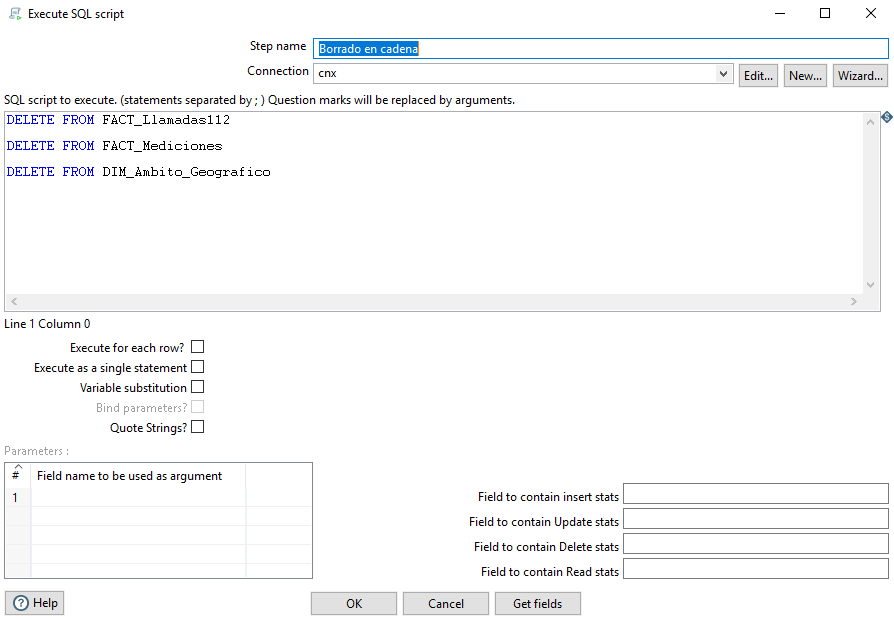


Ilustración - Borrado TR\_DIM\_AMBITO\_GEOGRAFICO.

#### Lectura

Una vez que hemos eliminado todos los registros los podemos volver a cargar, para ello tenemos que hacer una consulta a la tabla “STG\_Ambito\_Geografico”, y agrupamos para que no haya registros duplicados:

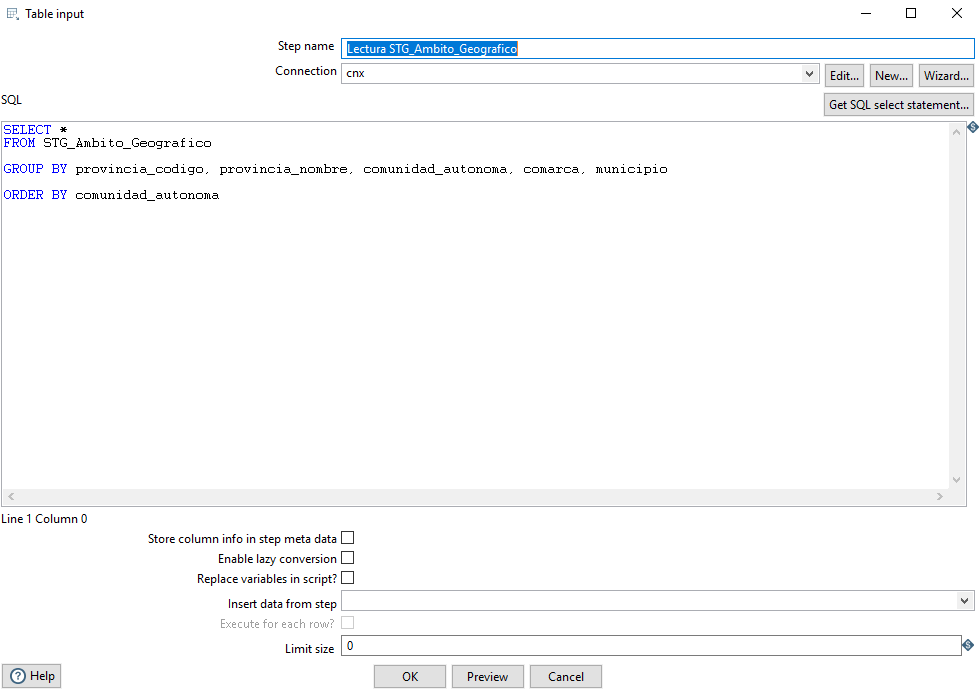


Ilustración - Lectura TR\_DIM\_AMBITO\_GEOGRAFICO.

#### 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:

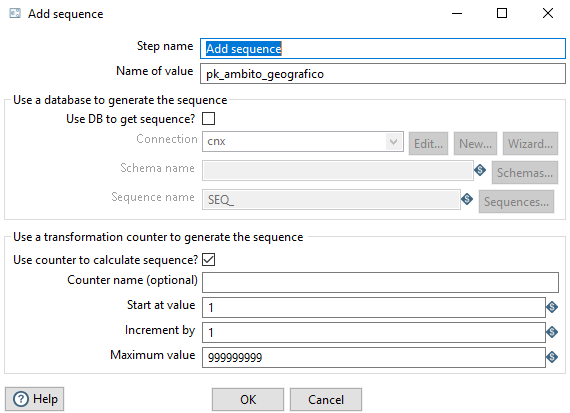


Ilustración - Secuenciación TR\_DIM\_AMBITO\_GEOGRAFICO.

#### Guardado

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

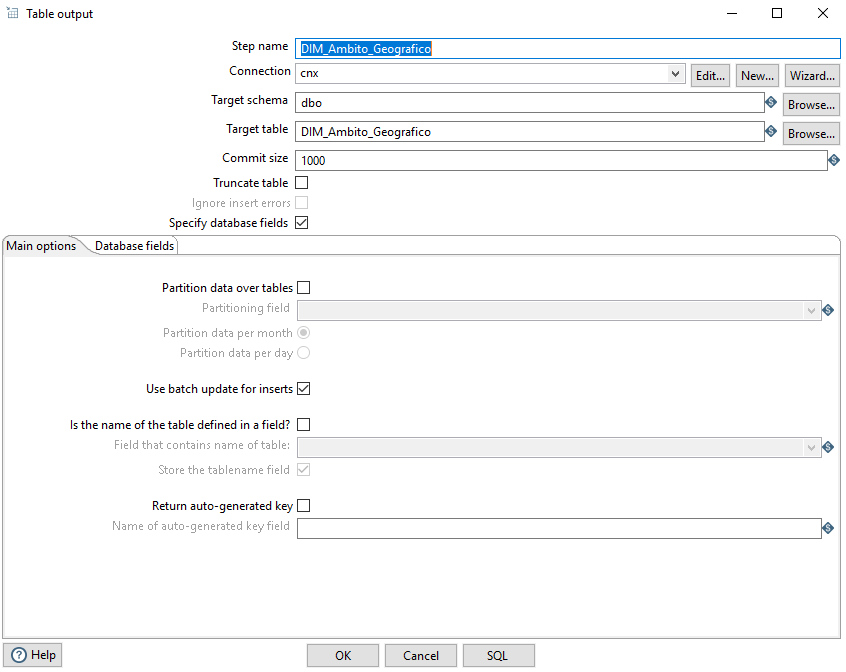


Ilustración - Guardado TR\_DIM\_AMBITO\_GEOGRAFICO.

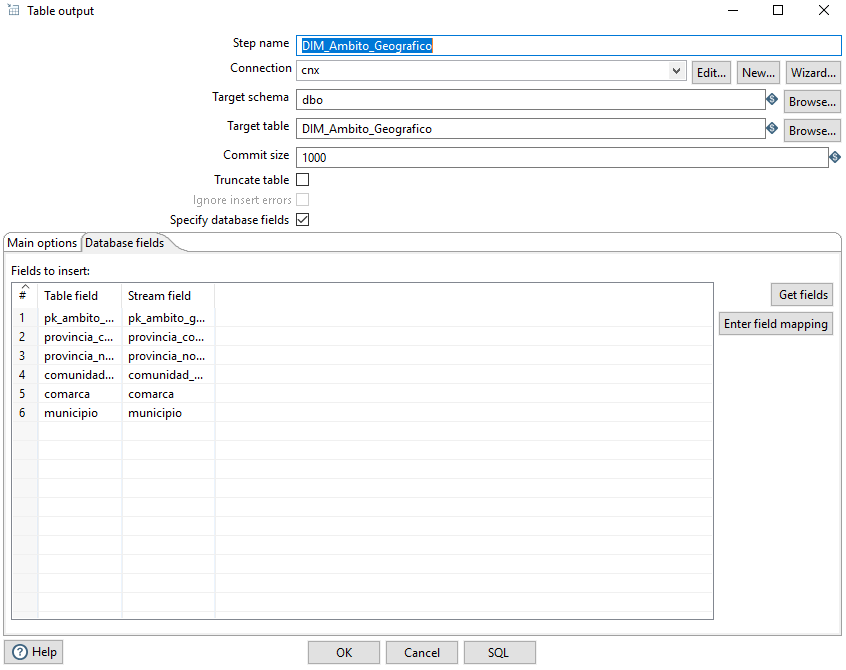


Ilustración - Guardado TR\_DIM\_AMBITO\_GEOGRAFICO.

El ejecutar la anterior transformación obtenemos las siguientes métricas:

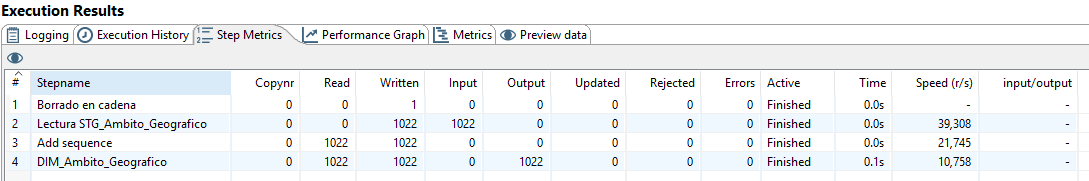


Ilustración - Métricas TR\_DIM\_AMBITO\_GEOGRAFICO.

Como podemos observar leemos 1022 registros y almacenamos en la dimensión los mismos 1022 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:

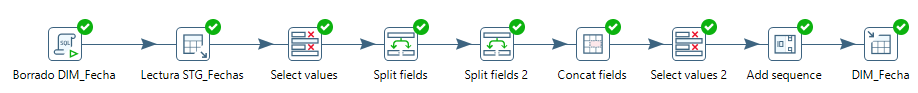


Ilustración 123 - TR\_DIM\_FECHA.

#### Borrado

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

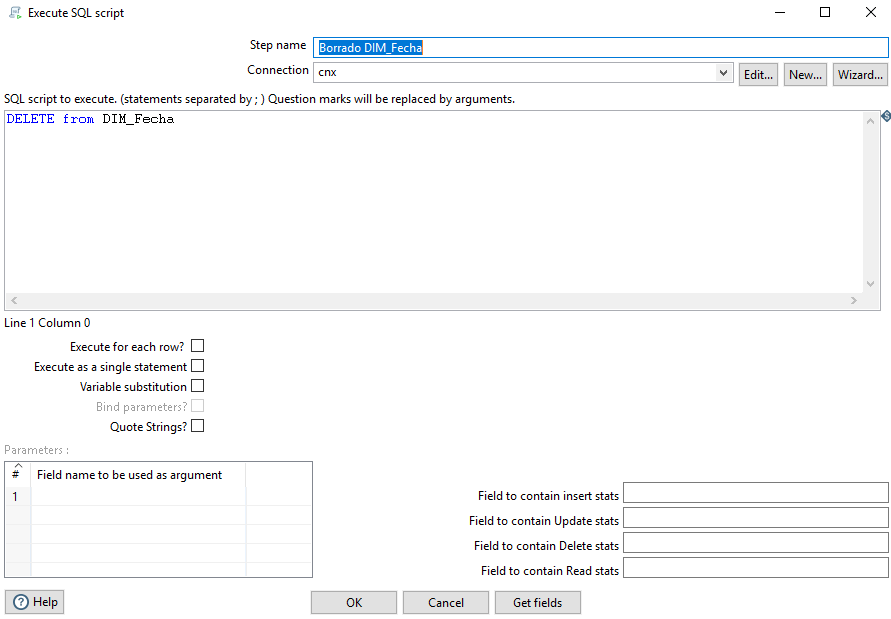


Ilustración 124 - Borrado TR\_DIM\_FECHA.

#### Lectura

Para hacer la lectura de las fechas tenemos que cargar todos los registros distintos de “STG\_Fechas”, ya que es en esta tabla intermedia donde están almacenadas todas las fechas de todos los ficheros fuentes que nos han proporcionado.

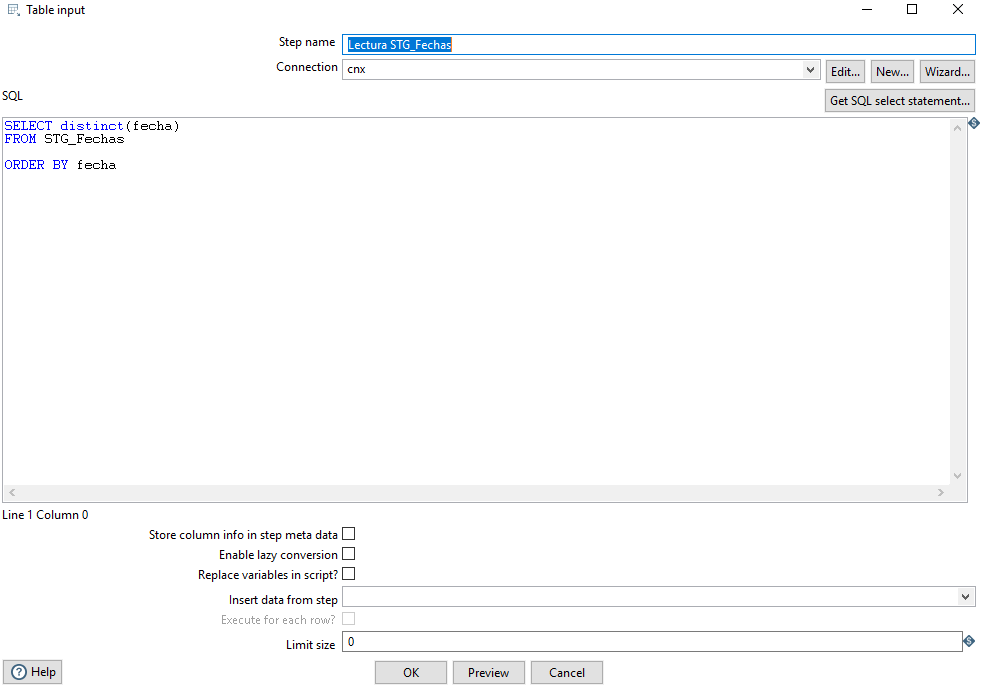


Ilustración 125 - Lectura TR\_DIM\_FECHA.

#### Conversión

Lo siguiente que debemos hacer es convertir el campo fecha que hemos leído en el paso anterior a string, ya que para esta dimensión no solo tenemos que guardar la fecha como tal, sino que también el día, mes y año por separado. Por lo tanto, convertimos a string la fecha:

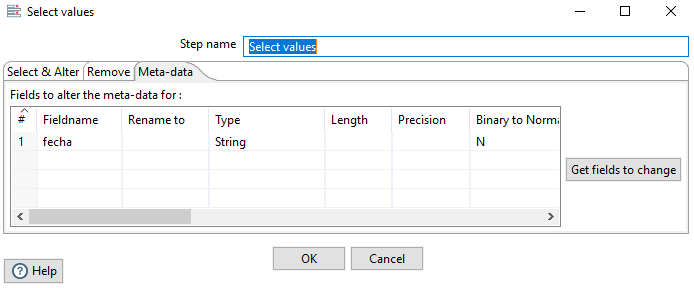


Ilustración 126 - Conversión String TR\_DIM\_FECHA.

#### Split

La fecha que tenemos en la base de datos nos proporciona también la hora, pero estos datos no nos interesan, es por ello que usamos split, le indicamos que divida la fecha a partir de un espacio en blanco:

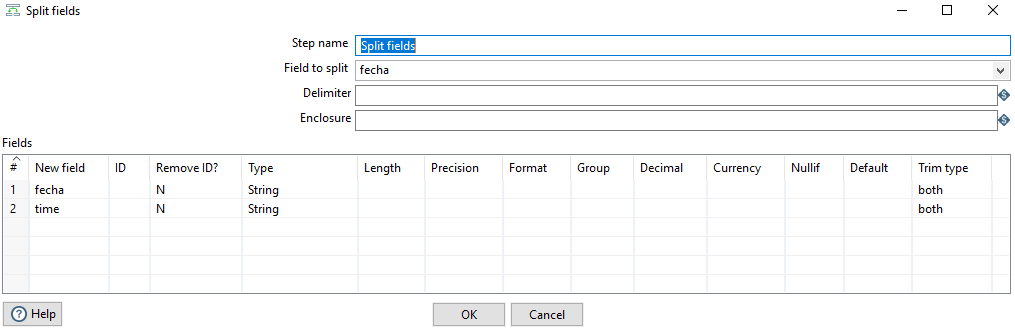


Ilustración 127 - Split TR\_DIM\_FECHA.

#### Split

Una vez que tenemos solamente la fecha en formato string, volvemos a hacer un split de los campos para obtener el día, mes y año en formato numérico:

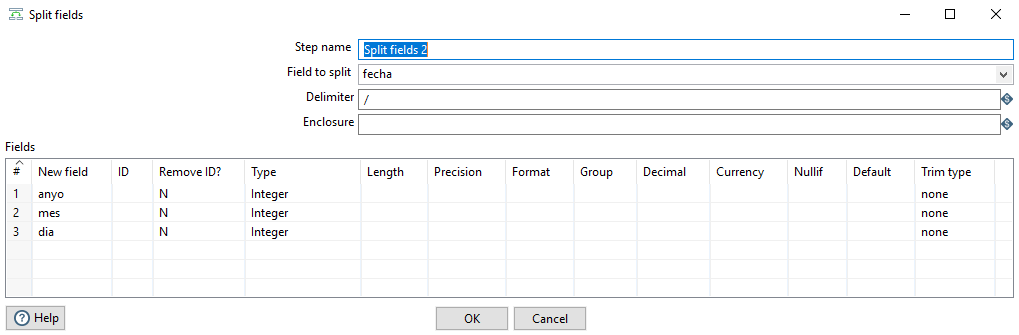


Ilustración 128 - Split TR\_DIM\_FECHA.

#### Concatenación

Al hacer split sobre el campo fecha hemos perdido ese campo como tal, es decir, hemos perdido el campo que tenía tanto el mes, día y año con el formato “yyyy/MM/dd”, es por ello que a partir de los campos creados en el paso anterior volvemos a crear la fecha:



Ilustración 129 - Concatenación TR\_DIM\_FECHA.

#### Conversión

Ese nuevo campo fecha está en formato string, pero en la dimensión necesitamos que sea de tipo date, es por ello que la convertimos a dicho tipo:

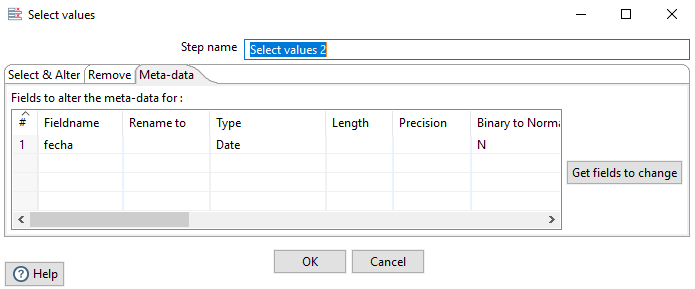


Ilustración 130 - Conversión TR\_DIM\_FECHA.

#### Secuenciación

Al igual que sucedía con las dimensiones anteriores, necesitamos definir una clave primaria, es por ello que creamos un nuevo campo autonumérico que se va incrementado de uno en uno:

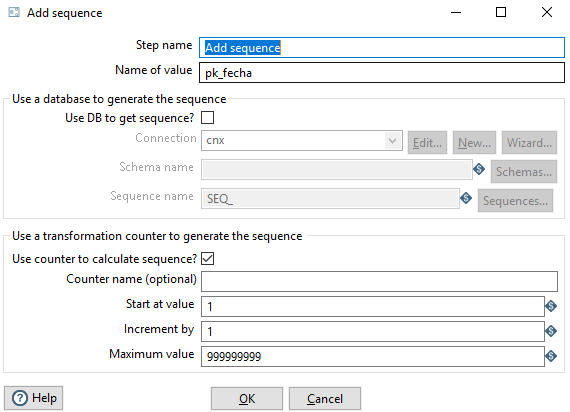


Ilustración 131 - Secuenciación TR\_DIM\_FECHA.

#### Guardado

Finalmente realizamos el guardado de todo este proceso en la “DIM\_Fecha”, para ello asociamos los campos con la tabla de la base de datos:

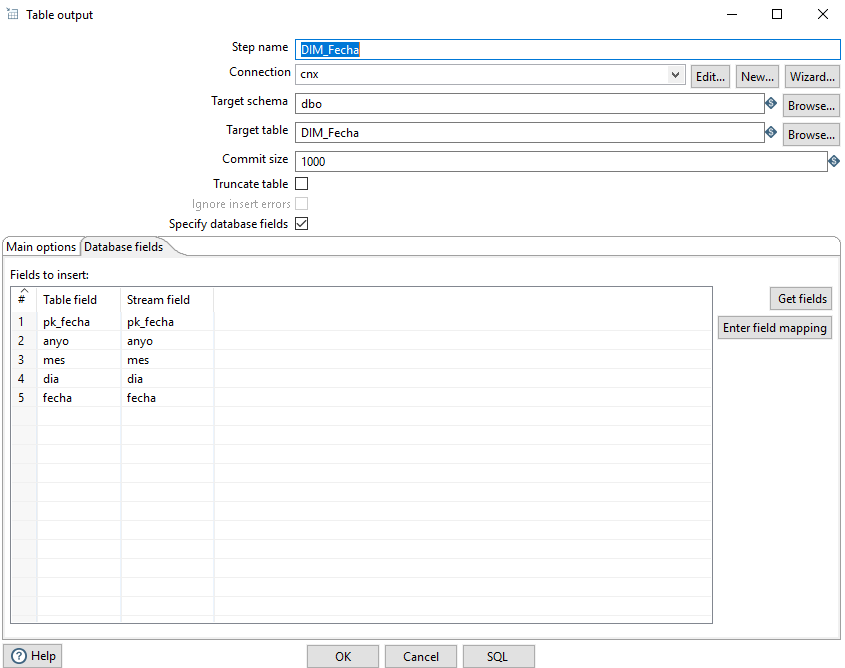


Ilustración 132 - Guardado TR\_DIM\_FECHA.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:



Ilustración 133 - Métricas TR\_DIM\_FECHA.

Como podemos apreciar de la anterior ilustración, tenemos 170 fechas únicas entre todos los ficheros proporcionados, y éstas mismas 170 se almacenan en la dimensión de forma correcta.

## Bloque TR Hechos

Una vez que hemos almacenado toda la información respectiva a las dimensiones, podemos hacer uso de estos datos junto con los de las tablas intermedias para crear las tablas de hechos.

### Transformación TR\_FACT\_LLAMADAS112

La primera transformación que vamos a realizar se llama “TR\_FACT\_LLAMADAS112”, su objetivo es almacenar las métricas relativas a dicho hecho, el resultado de esta transformación va a ser los datos almacenados en “FACT\_Llamadas112”.

La transformación nos ha quedado de la siguiente forma:

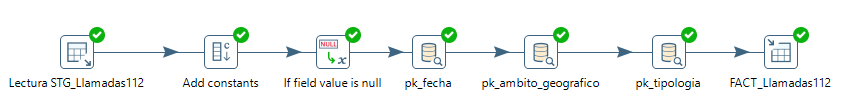


Ilustración - TR\_FACT\_LLAMADAS112.

#### Lectura

Lo primero de todo es cargar la información que tenemos en la tabla intermedia “STG\_Llamadas112”, en dicha tabla se encuentra las métricas de llamadas. Para cargar dicha información hacemos una consulta directamente a la base de datos:

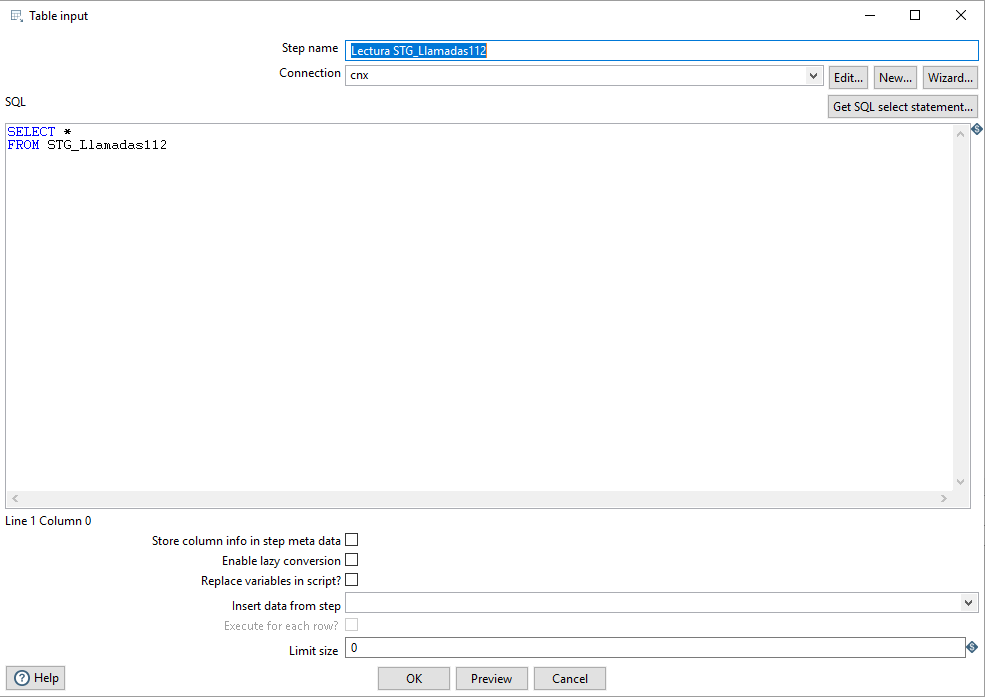


Ilustración - Lectura TR\_FACT\_LLAMADAS112.

#### Añadimos el día

Aunque a priori no es necesario el día para obtener la fecha, ya que las llamadas se registran con el año y el mes, para evitar problemas en el futuro añadimos el día como una constantes con valor 1:

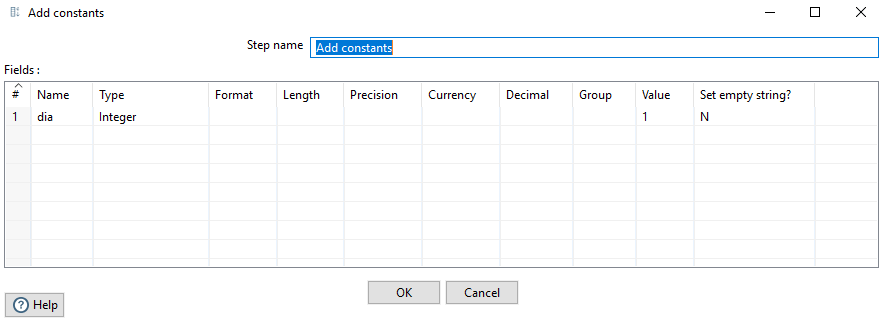


Ilustración - Añadimos Día TR\_FACT\_LLAMADAS112.

#### Replace

El siguiente paso es hacer un replace de los valores nulos por “NA”, esto se debe a que en la tabla intermedia “STG\_Llamadas112” almacenamos la información sin hacer esta transformación (sustituir los valores nulos por “NA”), por ello indicamos que en los campos (provincia, comarca y municipio ) si hay un valor nulo lo reemplace por “NA”:

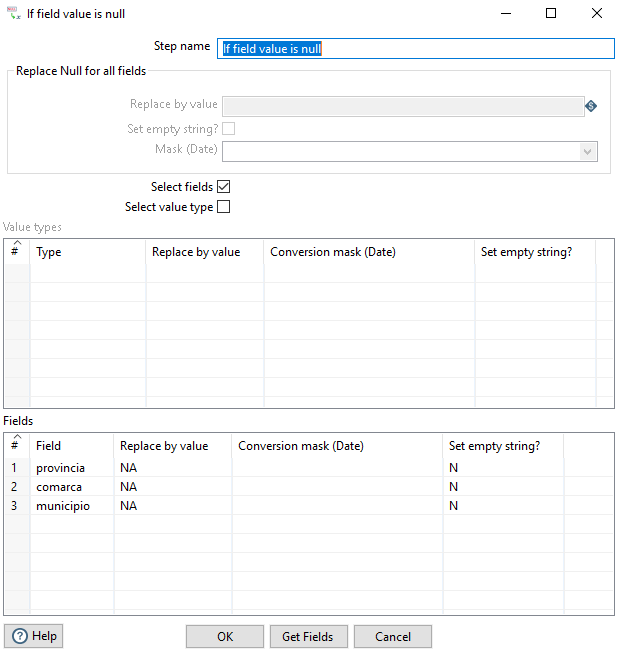


Ilustración - Replace TR\_FACT\_LLAMADAS112.

#### Búsqueda PK\_Fecha

Una vez que ya tenemos todos los datos de forma correcta, pasamos a obtener las claves primarias que serán a su vez claves foráneas de la tabla del hecho. La primera clave primaria a buscar es la fecha, para ello la buscamos según su año, mes y día:

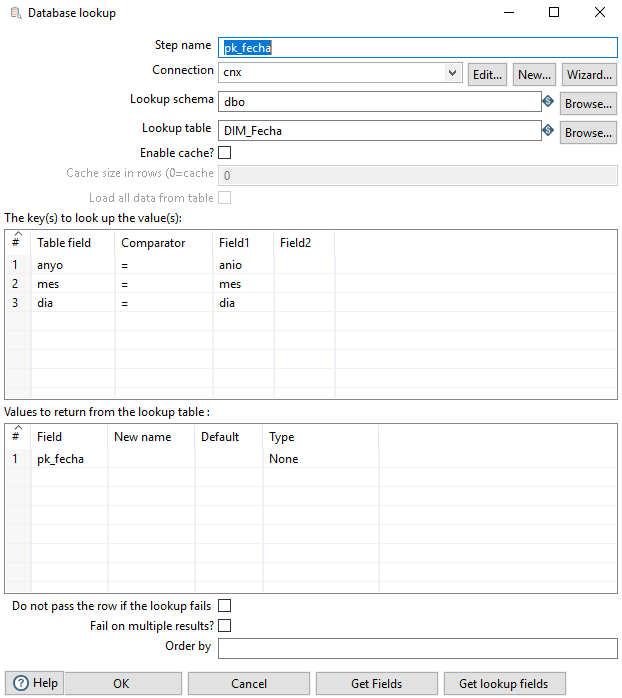


Ilustración - Búsqueda PK Fecha TR\_FACT\_LLAMADAS112.

#### Búsqueda PK\_Ambito\_Geografico

La segunda clave que tenemos que obtener está relacionada con la dimensión ámbito geográfico, para ello la buscamos a partir de su provincia, comarca y municipio:

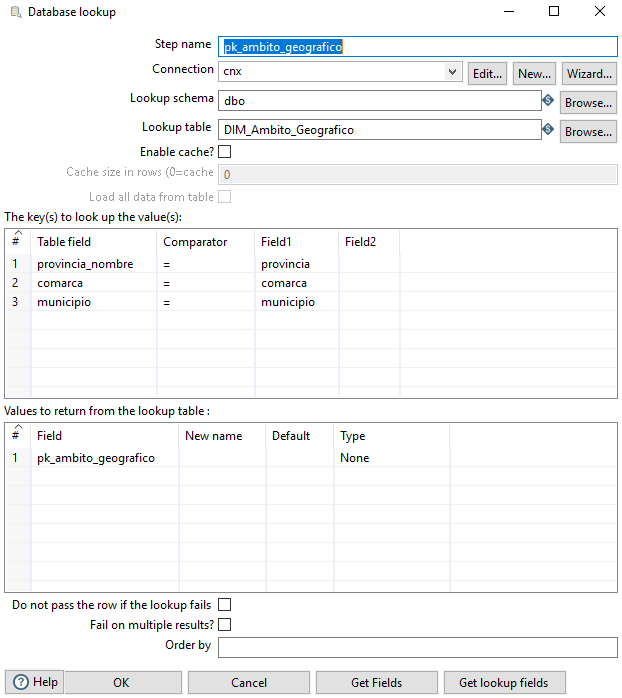


Ilustración - Búsqueda PK\_Ambito\_Geografico TR\_FACT\_LLAMADAS112.

#### Búsqueda PK\_Tipología

La última clave foránea que tenemos en la tabla del hecho está relacionada con la clave primaria de la dimensión de tipologías. Para obtener dicha clave buscamos por el nombre de la tipología:

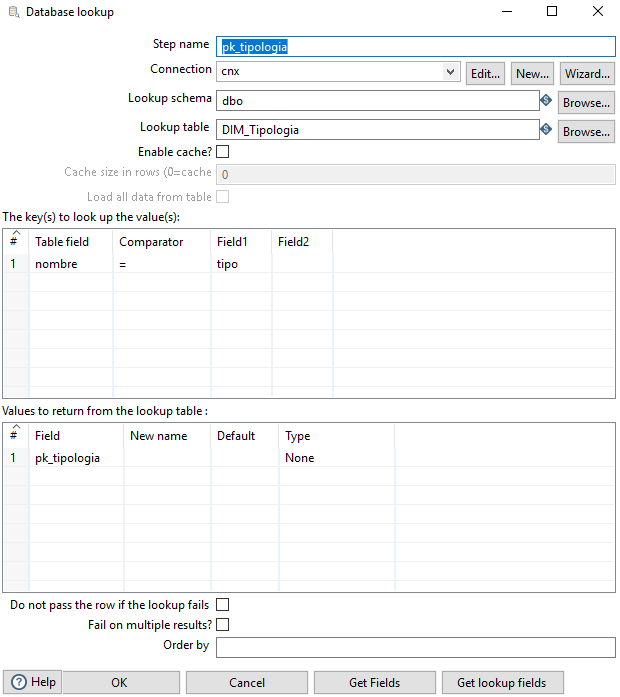


Ilustración - Obtención PK\_Tipología TR\_FACT\_LLAMADAS112.

#### Guardado

Finalmente podemos guardar los datos del hecho, es decir, las claves foráneas junto con su métrica “llamadas”, para ello asociamos los campos:

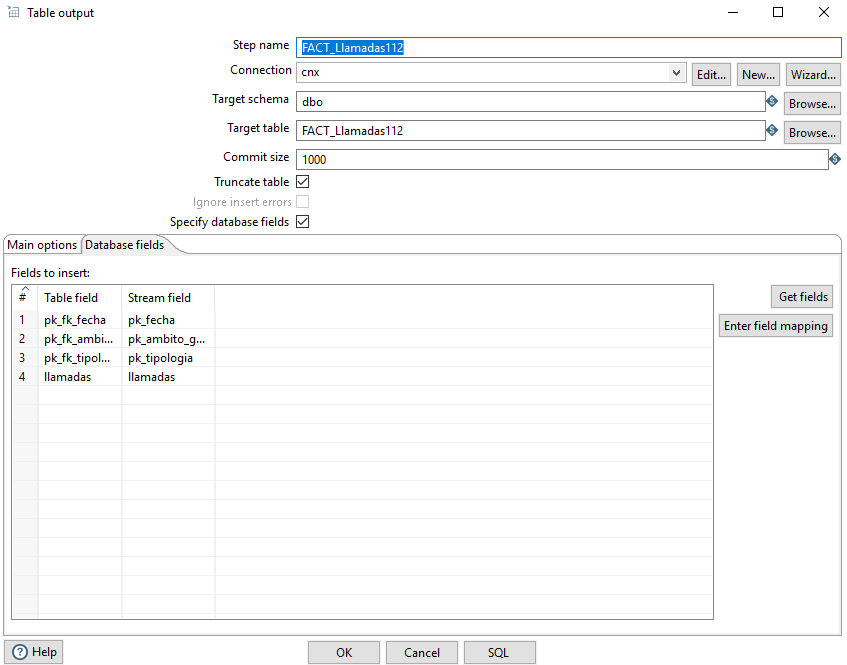


Ilustración - Guardado TR\_FACT\_LLAMADAS112.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

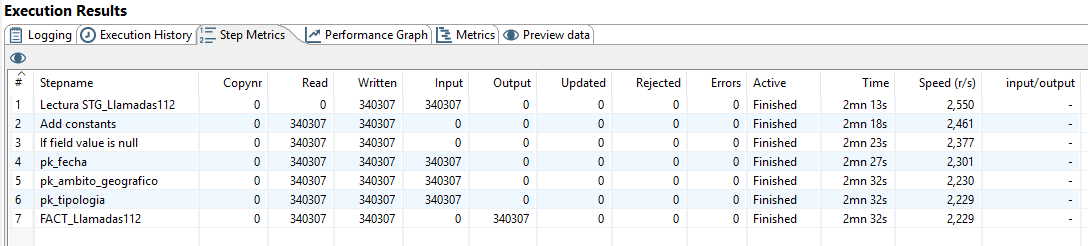


Ilustración - Métricas TR\_FACT\_LLAMADAS112.

De la ilustración anterior podemos ver que almacenamos 340307 registros.

### Transformación TR\_FACT\_MEDICIONES

La última transformación respecto a esta práctica es “TR\_FACT\_MEDICIONES”, su objetivo es almacenar ciertas mediciones (la movilidad, los datos de la población, los expedientes sancionadores aplicados y el grado de consciencia de los ciudadanos para evitar aglomeraciones).

La transformación nos ha quedado de la siguiente forma:

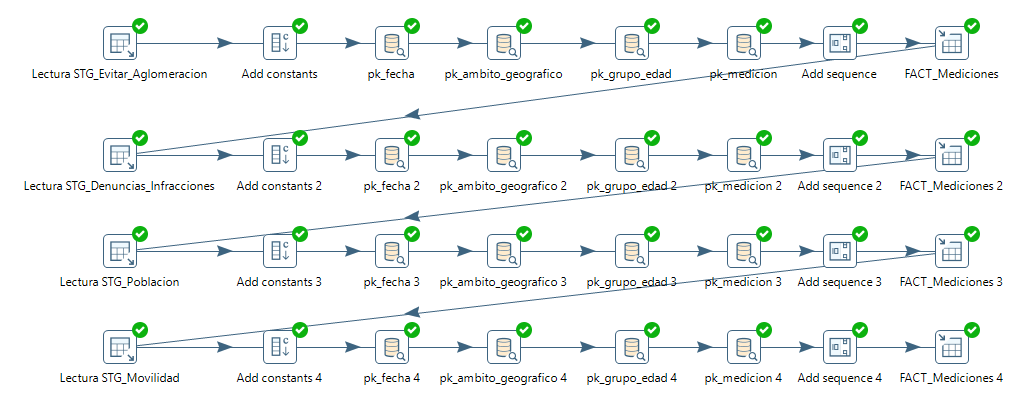


Ilustración - TR\_FACT\_MEDICIONES.

Como podemos apreciar esta transformación a nivel visual es la más compleja, no por su complejidad en sí, sino por la cantidad de elementos que hemos tenidos que introducir debido a los diferentes flujos de entrada.

Cabe destacar que la metodología para poder almacenar de forma correcta cada flujo de entrada es igual, es decir, leemos la tabla intermedia correspondiente, añadimos las constantes necesarias, buscamos las diferentes claves foráneas, añadimos la clave primaria y lo almacenamos en la base de datos. Para que esta transformación siguiera un orden hemos decidido que la salida de una fuera la entrada de otra.

Por lo tanto, vamos a explicar que hemos hecho en esta transformación en cada uno de sus componente, pero al ser todos los componentes comunes no vamos a entrar en mucho detalle para no explicar todo el rato lo mismo.

#### Lectura

En esta fase hemos realizado la lectura de todas las tablas intermedias que contenían la información necesaria para cumplir con el hecho.

Respecto a “STG\_Evitar\_Aglomeracion”, hemos leído todos los campos ya que éstos nos iban a hacer falta:

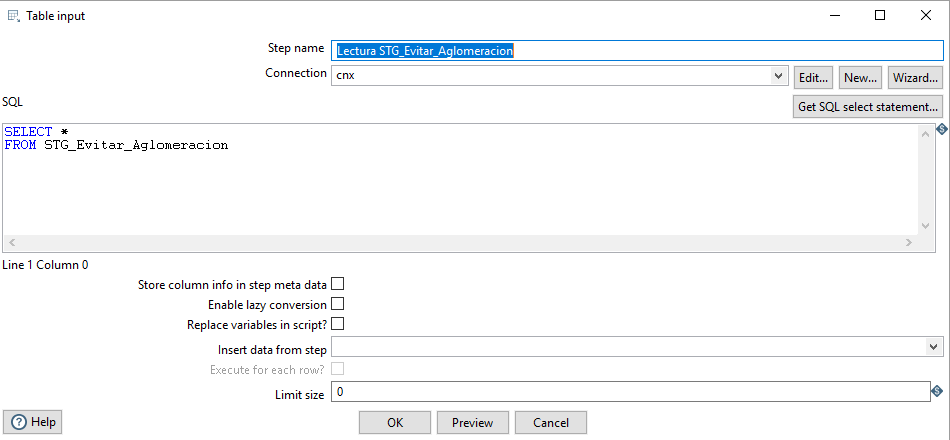


Ilustración - Lectura TR\_FACT\_MEDICIONES.

Respecto a “STG\_Denuncias\_Infracciones”, también hemos leído todos los campos, aunque en este caso solo son necesarios: la provincia, la fecha y las denuncias (tanto de la Ertzaintza como de la Policía Local).

Cabe destacar que según la solución de la práctica uno, el valor de denuncias tiene que ser la suma de las denuncias de la Ertzaintza más las denuncias de la policía local, es por ello que la lectura respecto a esta tabla nos ha quedado de la siguiente forma:

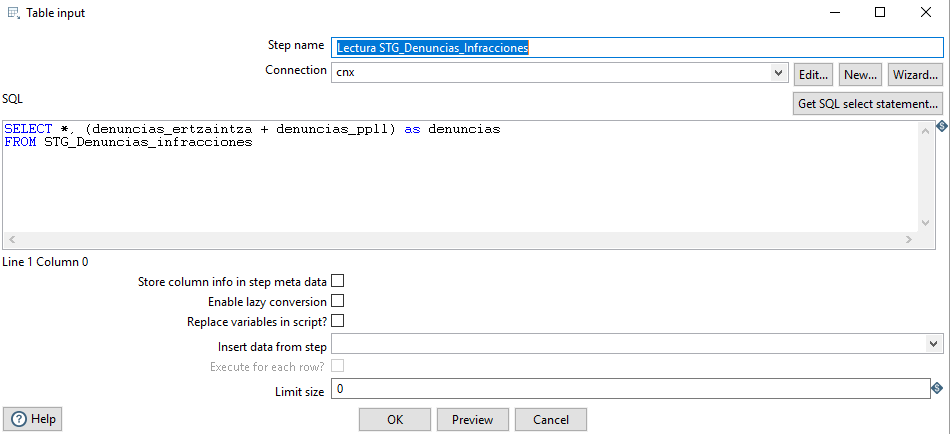


Ilustración - Lectura TR\_FACT\_MEDICIONES.

En cuanto a “STG\_Poblacion” leemos todos los campos:

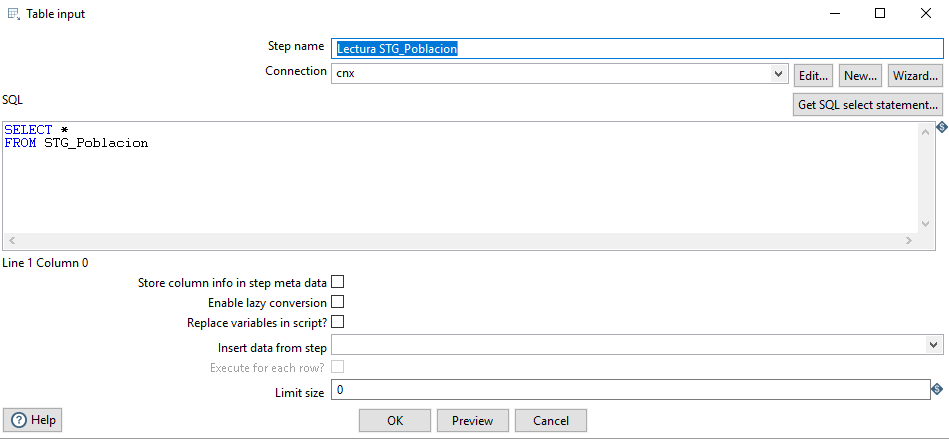


Ilustración - Lectura TR\_FACT\_MEDICIONES.

Finalmente hacemos la lectura de “STG\_Movilidad”, al igual que antes leemos todos los campos:

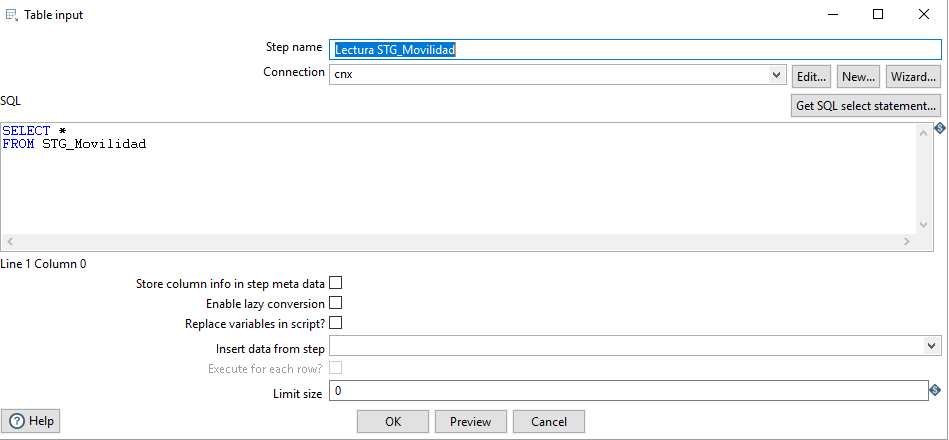


Ilustración - Lectura TR\_FACT\_MEDICIONES.

#### Añadimos los campos necesarios

Una vez que hemos hecho todas las lecturas, debemos de añadir los campos necesarios para buscar las claves foráneas de nuestra tabla del hecho.

Respecto a “STG\_Evitar\_Aglomeracion”, creamos el campo fecha cuyo valor va a ser siempre el “2020-01-01”, creamos el campo medida con el valor “%\_AGLO”, e indicamos que la comarca y el municipio son “NA” debido a que no tenemos dichos valores:

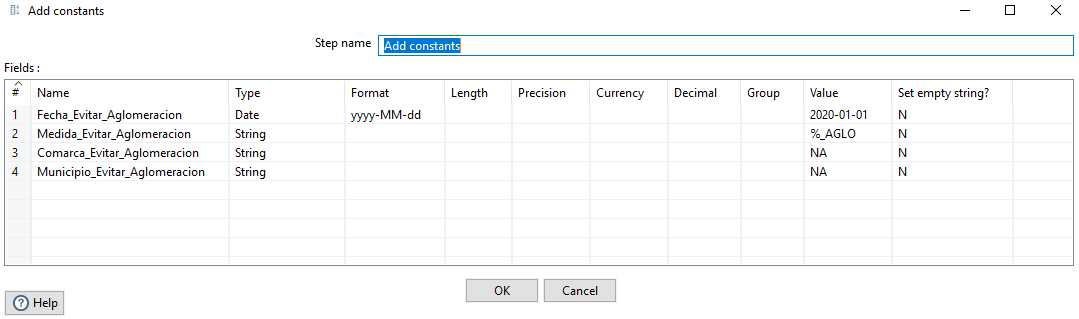


Ilustración - Añadimos Campos TR\_FACT\_MEDICIONES.

Respecto a “STG\_Denuncias\_Infracciones”, indicamos que la medida es “DENUNCIAS”, su comarca/municipio y grupo de edad es “NA” (carecemos de dicha información y el grupo de edad no aplica según estos datos) e indicamos que la comunidad autónoma es “PAÍS VASCO”:

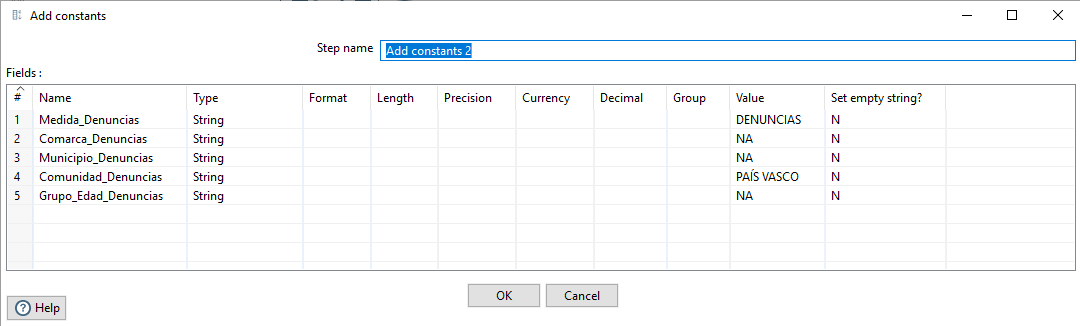


Ilustración - Añadimos Campos TR\_FACT\_MEDICIONES.

Respecto a “STG\_Poblacion”, indicamos que la medida es “HABITANTES”, su comarca/municipio y grupo de edad es “NA” (por el mismo motivo que en el caso anterior) y la fecha va a ser “2020-01-01” (en dicha tabla la fecha viene dada como “1 DE ENERO DE 2020” para todos los valores):

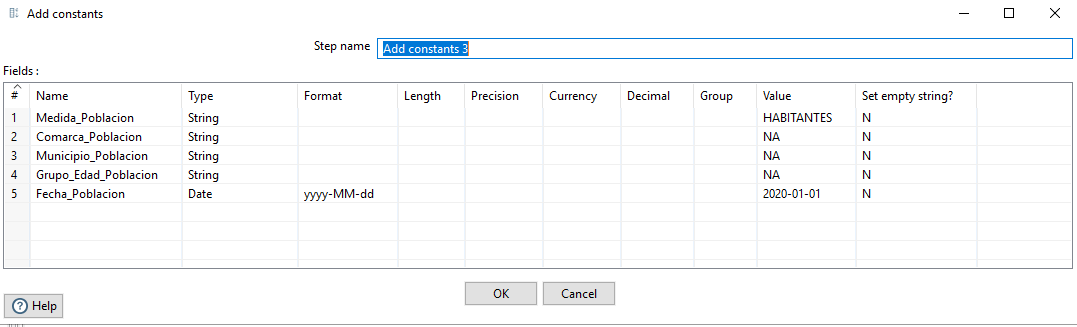


Ilustración - Añadimos Campos TR\_FACT\_MEDICIONES.

En cuanto a “STG\_Movilidad”, indicamos que la medida es “%\_MOV”, e indicamos que la comarca/municipio y grupo de edad es “NA” por los mismos motivos que antes:



Ilustración - Añadimos Campos TR\_FACT\_MEDICIONES.

#### Búsqueda PK\_Fecha

El siguiente paso es obtener cada una de las claves foráneas, en nuestro caso, la primera es la fecha.

Respecto a “STG\_Evitar\_Aglomeracion”, buscamos la fecha como tal y devolvemos la clave primaria asociada a dicha fecha, el procedimiento va a ser el mismo en todos los flujos de entrada:

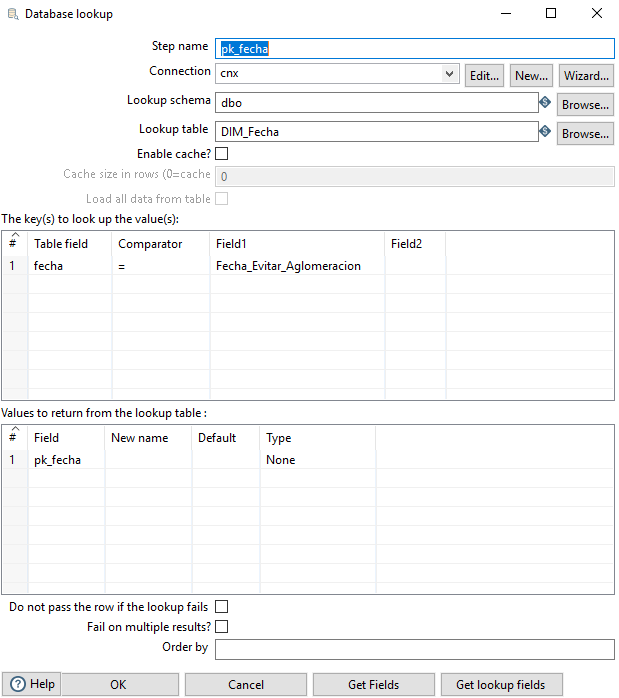


Ilustración - Búsqueda PK\_Fecha TR\_FACT\_MEDICIONES.

Respecto a “STG\_Denuncias\_Infracciones”, realizamos lo mimo que en caso anterior pero con el campo fecha correspondiente:

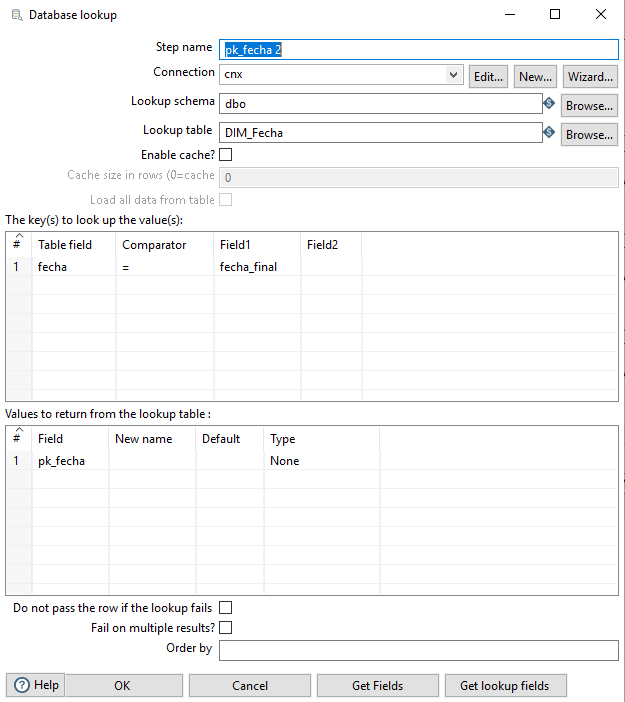


Ilustración - Búsqueda PK\_Fecha TR\_FACT\_MEDICIONES.

En cuanto a “STG\_Poblacion” realizamos lo mismo pero con su atributo correspondiente:

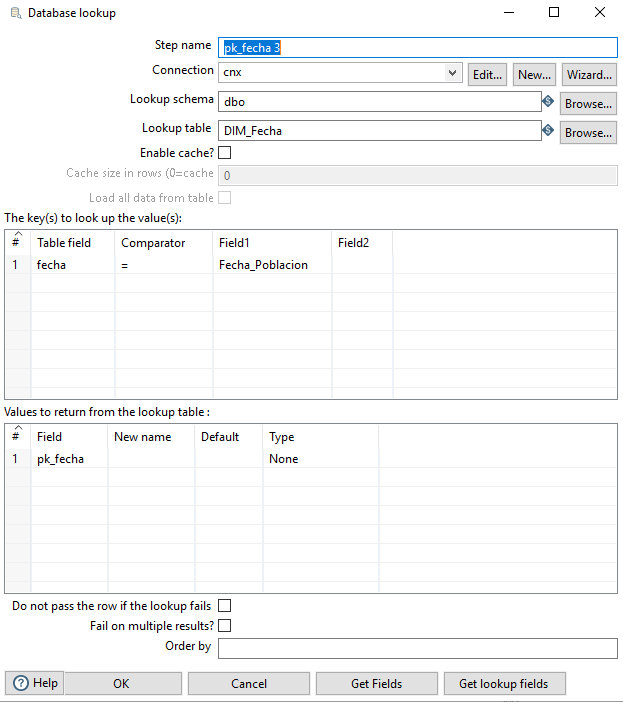


Ilustración - Búsqueda PK\_Fecha TR\_FACT\_MEDICIONES.

Finalmente en “STG\_Movilidad” hacemos lo mismo:

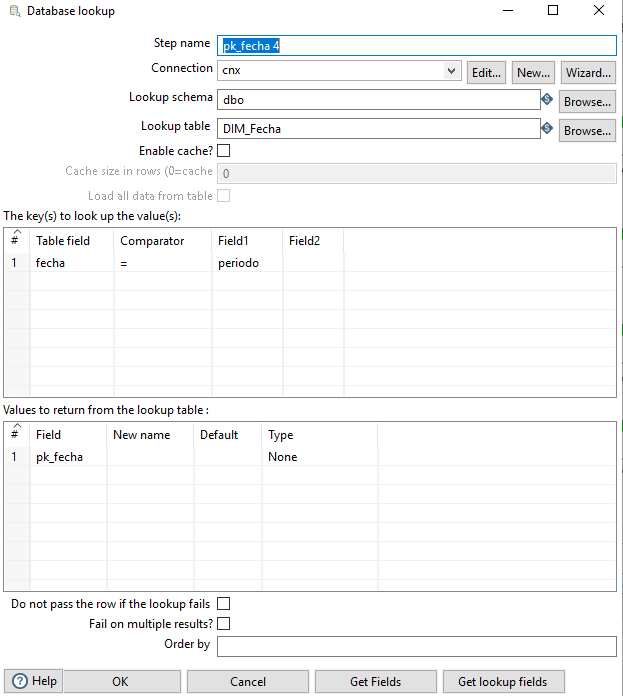


Ilustración - Búsqueda PK\_Fecha TR\_FACT\_MEDICIONES.

#### Búsqueda PK\_Ambito\_Geografico

Lo siguiente es obtener la clave primaria respecto al ámbito geográfico de queda flujo de entrada.

En cuanto a “STG\_Evitar\_Aglomeracion”, buscamos tanto por el nombre de la provincia, su comunidad autónoma, comarca y municipio:

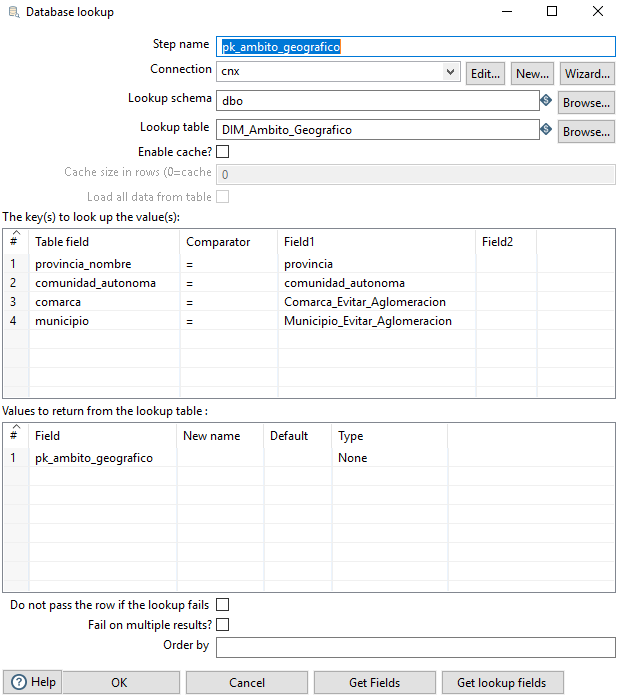


Ilustración - Búsqueda PK\_Ambito\_Geografico TR\_FACT\_MEDICIONES.

Respecto a “STG\_Denuncias\_Infracciones” buscamos la clave primaria a partir de los mismos datos que en caso anterior, es decir, de la provincia, comunidad autónoma, comarca y municipio:

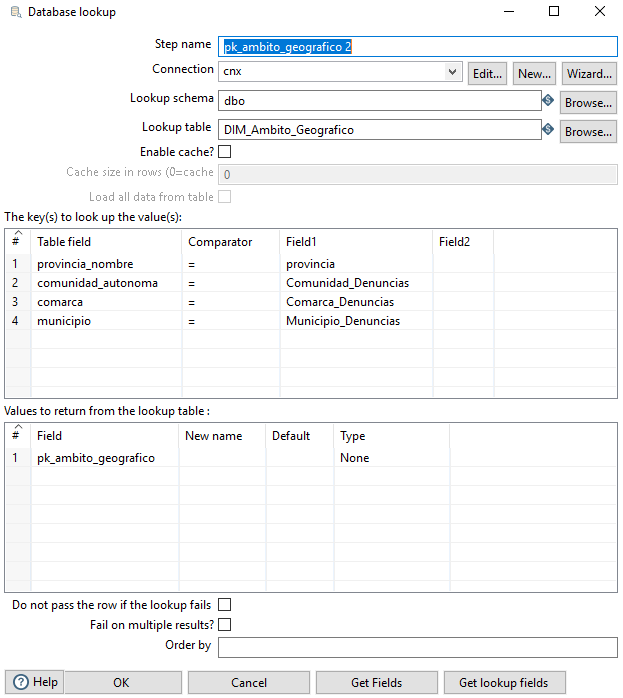


Ilustración - Búsqueda PK\_Ambito\_Geografico TR\_FACT\_MEDICIONES.

En cuanto a “STG\_Poblacion”, realizamos un cambio respecto a los casos anteriores, ahora no tenemos la comunidad autónoma, sin embargo sí que tenemos el código de la provincia, por lo tanto buscamos la clave primaria a partir de la provincia, su código, comarca y municipio:

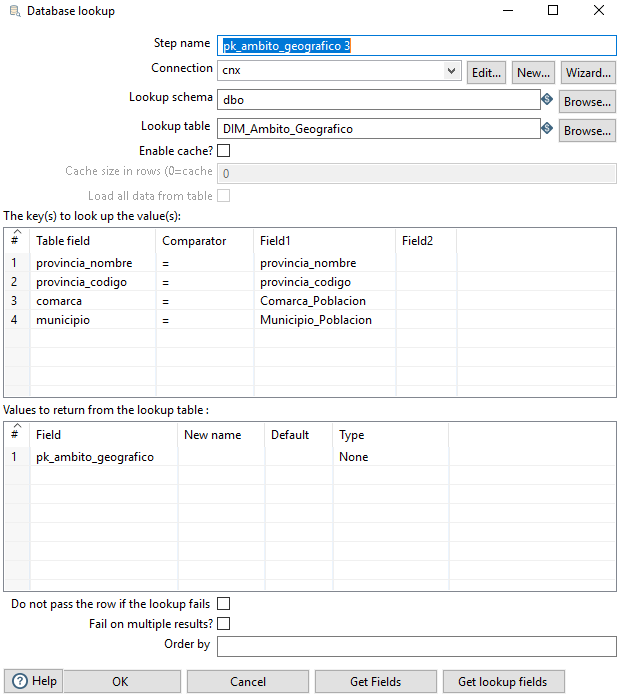


Ilustración - Búsqueda PK\_Ambito\_Geografico TR\_FACT\_MEDICIONES.

Finalmente “STG\_Movilidad”, solo tenemos información sobre el nombre de la provincia, su comarca y municipio, por lo que buscamos por esos campos:

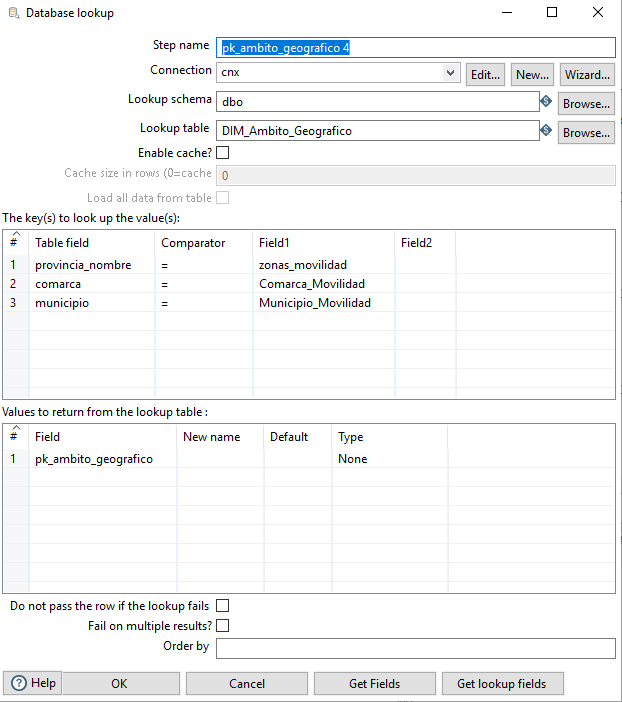


Ilustración - Búsqueda PK\_Ambito\_Geografico TR\_FACT\_MEDICIONES.

#### Búsqueda PK\_Grupo\_Edad

La siguiente clave foránea que tenemos que buscar es el grupo de edad.

Respecto a “STG\_Evitar\_Aglomeracion”, vamos a buscar que el intervalo de grupo de edad coincida para así obtener la clave, este proceso es el mismo para todos los flujos de entrada:

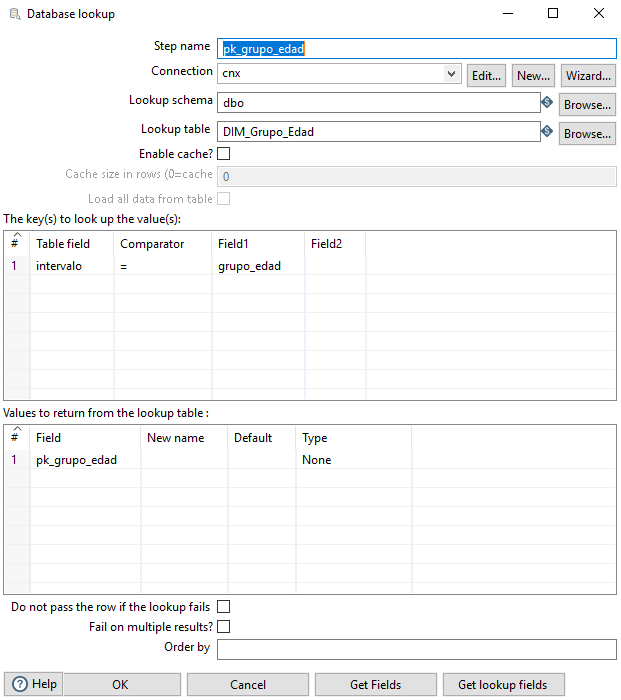


Ilustración - Búsqueda PK\_Grupo\_Edad TR\_FACT\_MEDICIONES.

Respecto a “STG\_Denuncias\_infracciones”:

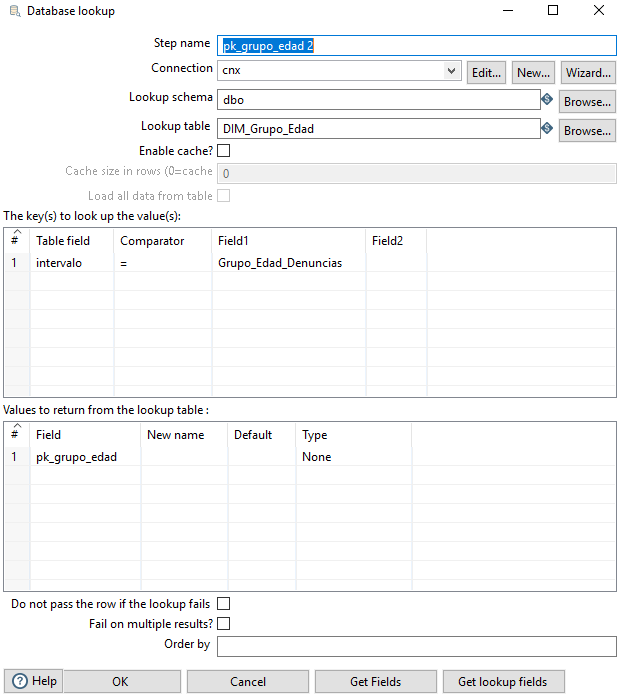


Ilustración - Búsqueda PK\_Grupo\_Edad TR\_FACT\_MEDICIONES.

En cuanto a “STG\_Poblacion”:

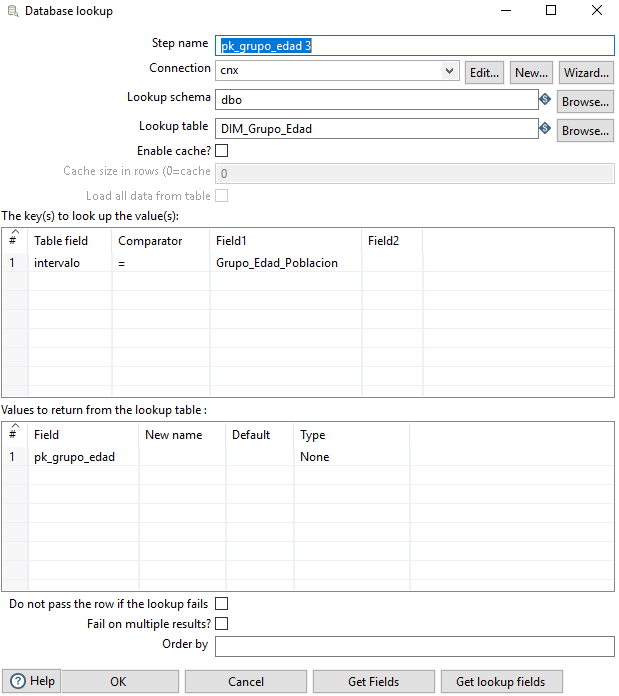


Ilustración - Búsqueda PK\_Grupo\_Edad TR\_FACT\_MEDICIONES.

En cuanto a “STG\_Movilidad”:

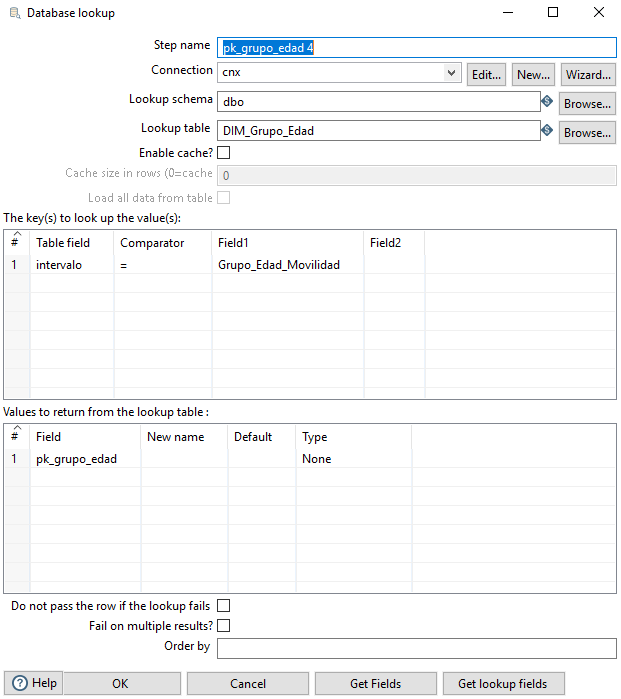


Ilustración - Búsqueda PK\_Grupo\_Edad TR\_FACT\_MEDICIONES.

#### Búsqueda PK\_Medicion

Finalmente, la última clava foránea que tenemos que buscar es la clave primaria respecto a la medición.

En cuanto a “STG\_Evitar\_Aglomeracion”, indicamos que busque la clave primaria a partir del nombre de la medición, este proceso es igual para todos los flujos de entrada:

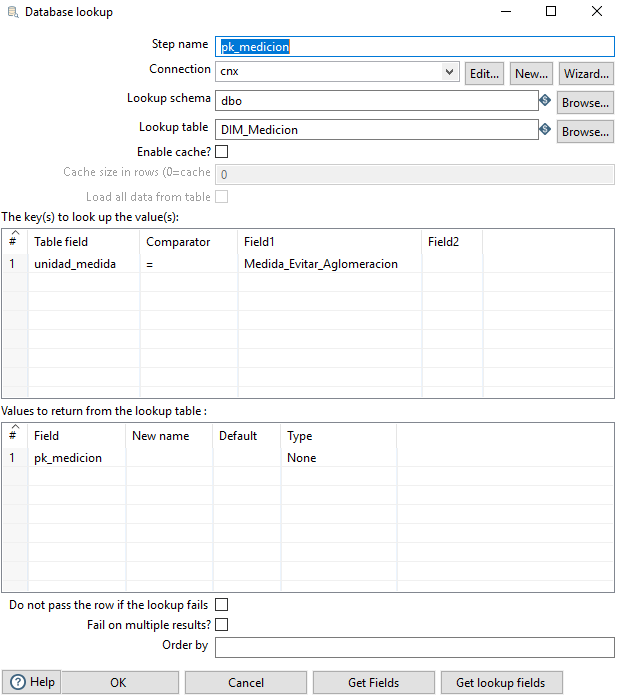


Ilustración - Búsquedas PK\_Medicion TR\_FACT\_MEDICIONES.

Respecto a “STG\_Denuncias\_Infracciones”:

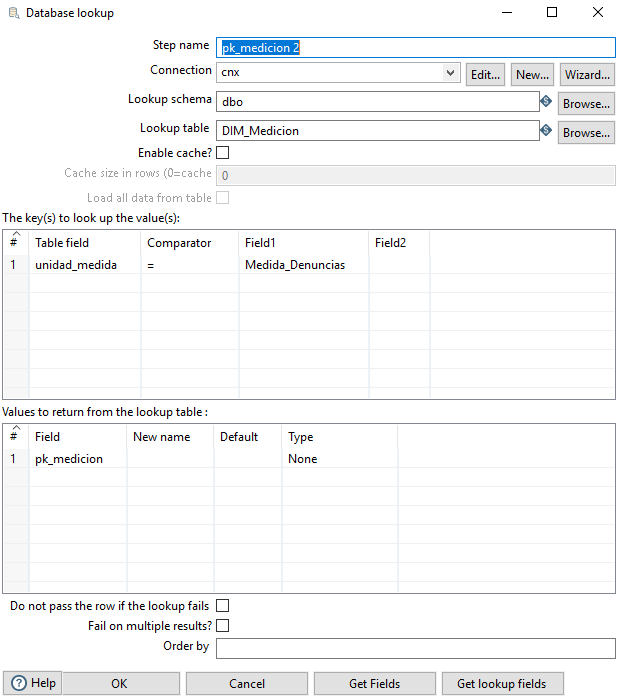


Ilustración - Búsqueda PK\_Medicion TR\_FACT\_MEDICIONES.

Respecto a “STG\_Poblacion”:

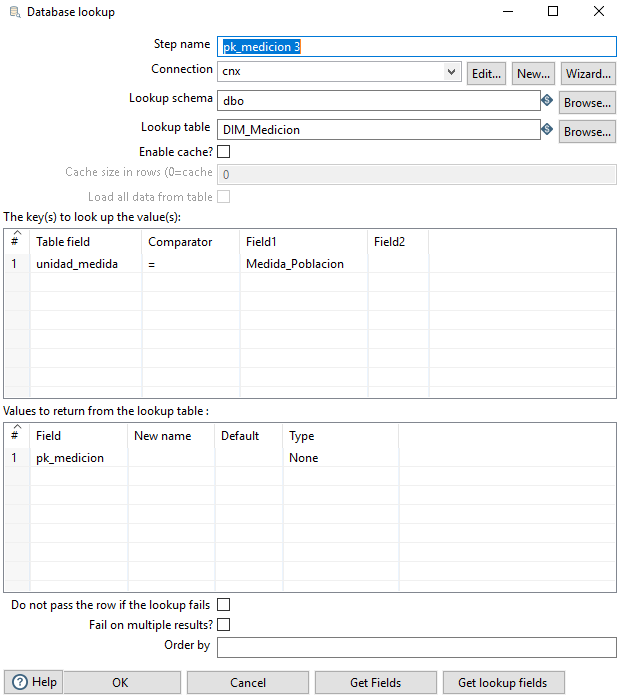


Ilustración - Búsqueda PK\_Medicion TR\_FACT\_MEDICIONES.

Respecto a “STG\_Movilidad”:

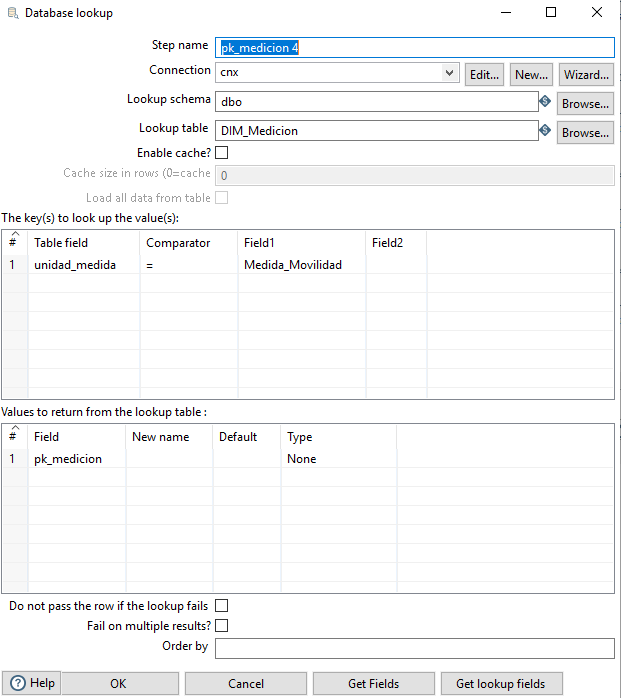


Ilustración - Búsqueda PK\_Medicion TR\_FACT\_MEDICIONES.

#### Secuenciación

Una vez que ya tenemos todas las claves foráneas, tenemos que establecer la clave primaria de nuestra tabla del hecho, en este caso es una secuencia que se va autoincrementado de uno en uno.

Este componente es exactamente el mismo para todos los flujos de entrada, es por ello que solo lo vamos a indicar una vez el cómo está configurado:

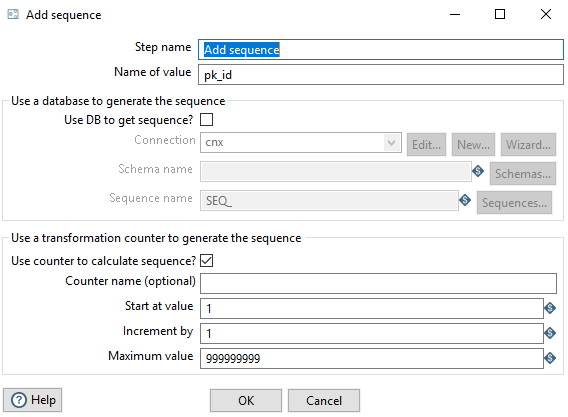


Ilustración - Secuenciación TR\_FACT\_MEDICIONES.

#### Guardado

Cuando ya tenemos todos los datos podemos realizar el guardado en la tabla “FACT\_Mediciones”, en ella vamos a almacenar toda la información necesaria respecto a las métricas.

En cuanto a “STG\_Evitar\_Aglomeracion”, al ser la primera transformación hacemos un truncate de la tabla por si hay algún registro. Por otro lado, indicamos el id, la fecha, el ámbito geográfico, el grupo de edad, la medición y la métrica (el porcentaje de la población que evitaba aglomeraciones):

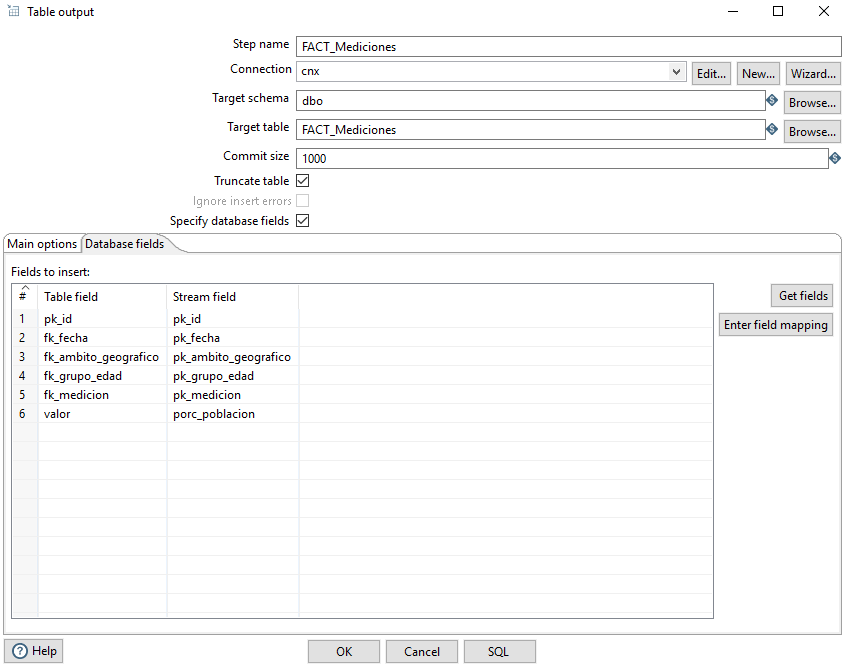


Ilustración - Guardado TR\_FACT\_MEDICIONES.

Respecto a “STG\_Denuncias\_Infracciones”, todos los campos son comunes con el caso anterior, a excepción de la métricas (en este caso el número de denuncias):

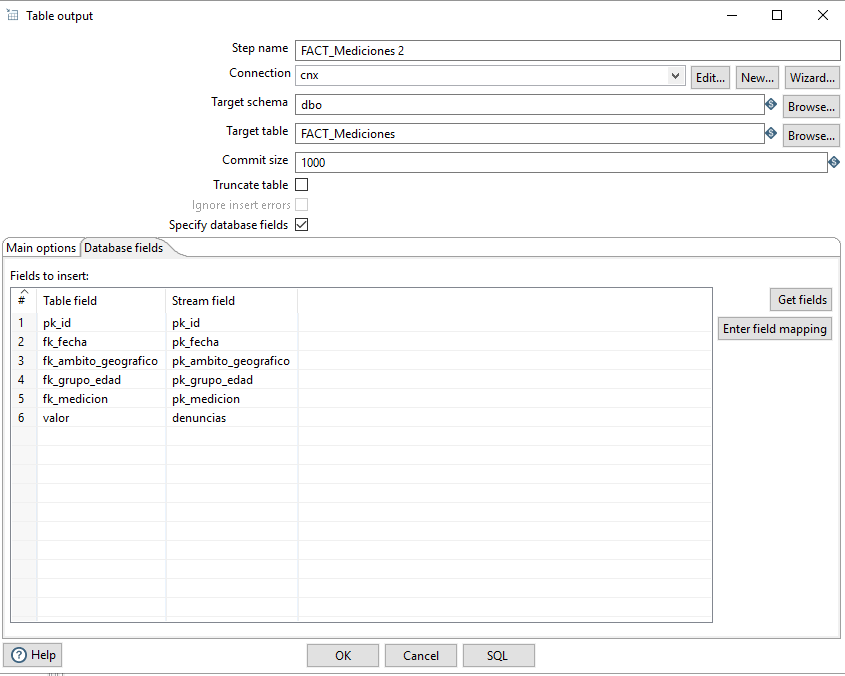


Ilustración - Guardado TR\_FACT\_MEDICIONES.

En cuanto a “STG\_Poblacion”, se difiere de los procesos anteriores por la métrica, en este caso la población por provincia que hay:

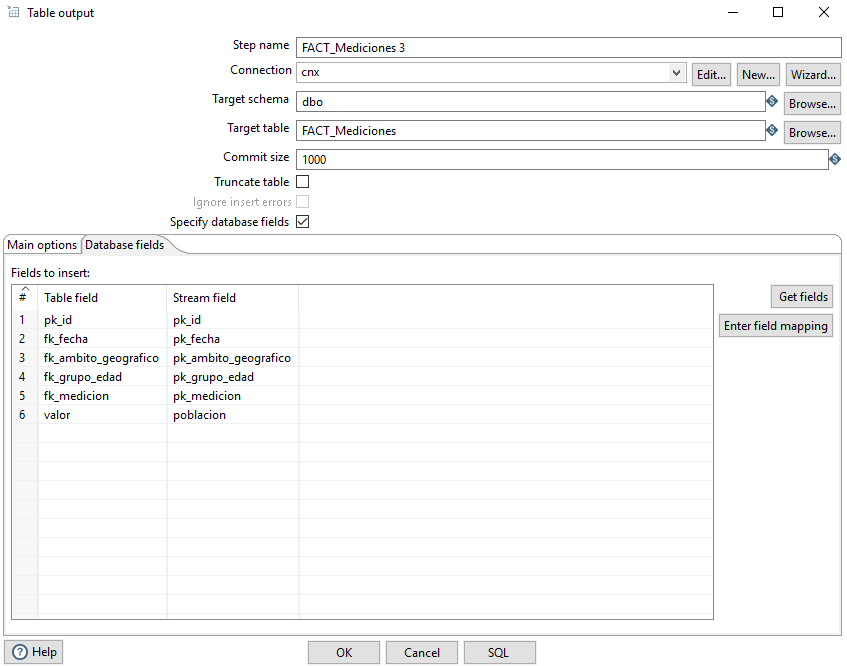


Ilustración - Guardado TR\_FACT\_MEDICIONES.

Finalmente respecto a “STG\_Movilidad”, indicamos el porcentaje de la población que se trasladó en el estado de alarma:

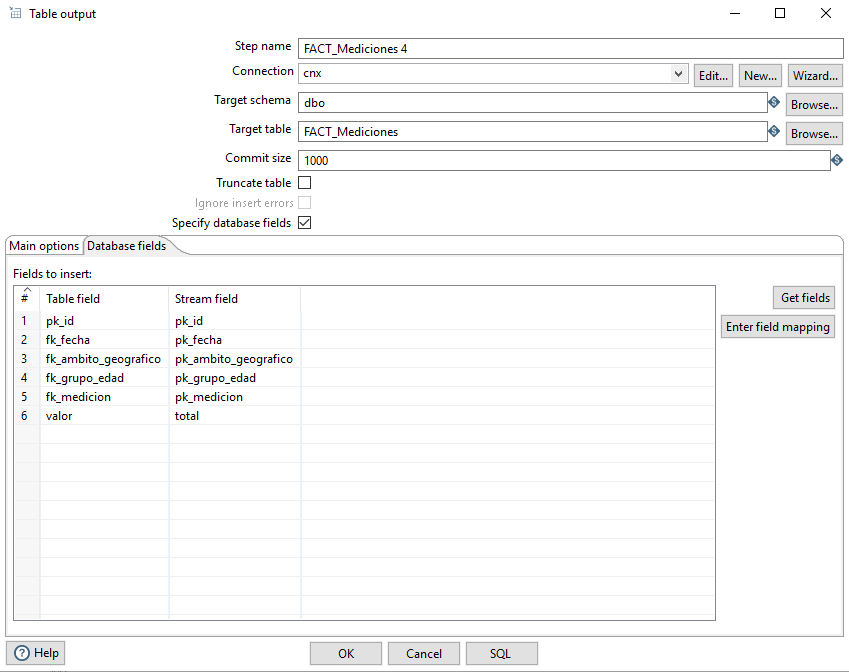


Ilustración - Guardado TR\_FACT\_MEDICIONES.

Al ejecutar la anterior transformación obtenemos las siguientes métricas:

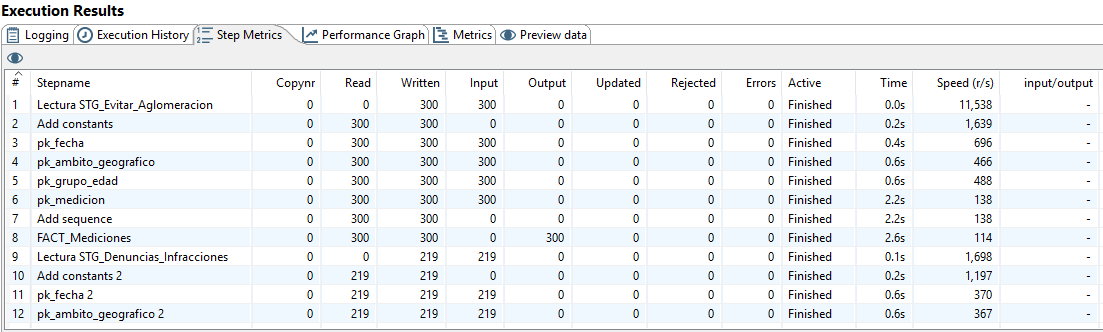


Ilustración - Métricas TR\_FACT\_MEDICIONES.

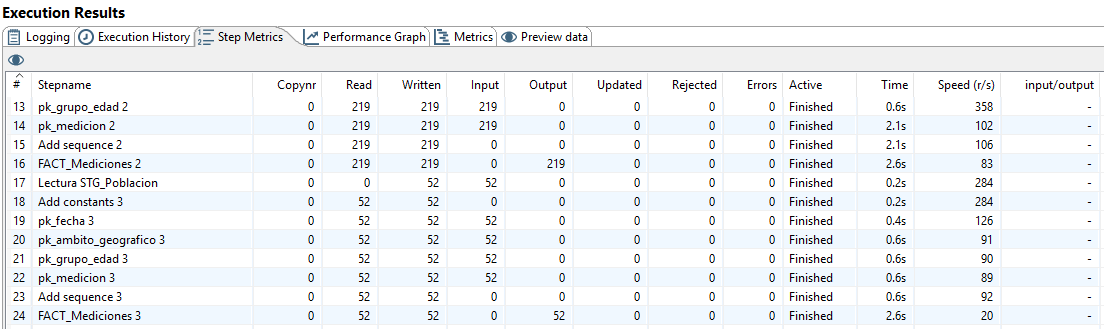


Ilustración - Métricas TR\_FACT\_MEDICIONES.

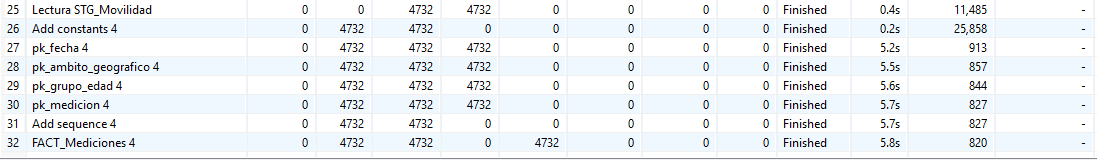


Ilustración - Métricas TR\_FACT\_MEDICIONES.

De la anterior ejecución podemos ver que de “STG\_Evitar\_Aglomeracion” almacenamos 300 registros, de “STG\_Denuncias\_Infracciones” 219 registros, de “STG\_Poblacion” 52 registros y de “STG\_Movilidad” 4732 registros. Esto hace un total de 5303 registros en la tabla “FACT\_Mediciones”.

|  |
| --- |
| Implementación de jobs |

a