

CARGA DE DATOS DEL ALMACÉN DE DATOS PARA EL ANÁLISIS DEL IMPACTO AMBIENTAL Y CONSUMO ENERGÉTICO DERIVADOS DE LA ACTIVIDAD ECONÓMICA



EDUARDO MORA GONZÁLEZ

Tabla de contenido

1.	Identificación de los procesos ETL.....	3
1.1.	Bloque IN (de las fuentes a tablas intermedias).....	3
1.2.	Bloque TR (poblar las tablas de nuestro almacén).....	4
2.	Diseño y desarrollo de los procesos ETL.....	5
2.1.	Creación de tablas intermedias (staging area).....	5
2.1.1.	IN_OBJECTIVES	5
2.1.2.	IN_INVESTMENTS	6
2.1.3.	IN_COUNTRIES	6
2.1.4.	IN_PROTECTEDAREAS	7
2.1.5.	IN_ENERGYBALANCES.....	7
2.1.6.	IN_URBANWASTES	8
2.2.	Creación del modelo multidimensional.....	8
2.3.	Base de datos final	8
2.4.	Creación del proceso de extracción, transformación y carga	9
2.4.1.	Crear repositorio de trabajo	9
2.4.2.	Conexión a la base de datos SQL Server	9
2.4.3.	Bloque IN	10
2.4.4.	Bloque TR.....	27
2.4.5.	Bloque TR_DIM	27
2.4.6.	Bloque TR_FACT	37
3.	Implementación de los trabajos con procesos ETL	44
3.1.	JOB_IN	44
3.2.	JOB_TR_DIMS.....	46
3.3.	JOB_TR_FACTS.....	47
3.4.	JOB_CARGA_DW	48

1. Identificación de los procesos ETL

No existe una estrategia única a la hora de diseñar y analizar el proceso de carga de la base de datos. Es muy común construir un proceso ETL basado en las entidades de datos que se van a actualizar, porque existe una diferencia conceptual en la actualización de la tabla de dimensiones en comparación con la tabla de hechos.

Dividir el proceso de carga inicial en diferentes bloques de actualización ayudará al diseño, la secuencia de ejecución y la gestión de dependencias. Cada uno de estos bloques de actualización se dividirá en las correspondientes fases 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).
- **Bloque TR:** procesos de transformación para cargar los datos desde las tablas intermedias hasta nuestro almacén de datos, según el modelo multidimensional diseñado.

1.1. Bloque IN (de las fuentes a tablas intermedias)

NOMBRE DEL ETL	DESCRIPCIÓN	ORÍGENES DE LOS DATOS	TABLA DESTINO
IN_OBJECTIVES	Carga de los datos correspondientes a los Objetivos de Desarrollo Sostenible y su relación con los ámbitos medioambientales	ODS.xlsx	STG_OBJECTIVES STG_OBJECTIVESAREAS
IN_INVESTMENTS	Carga de datos correspondientes a la evolución de la inversión en protección ambiental por tipo de equipo e instalación, ámbito medioambiental y sector de actividad económica.	02002.xlsx	STG_INVESTMENTS
IN_COUNTRIES	Carga de datos correspondientes a los nombres de los distintos países en orden alfabético y los elementos de código ISO 3166-1-alpha-2 y alpha-3	Countries.json	STG_COUNTRIES
IN_PROTECTEDAREAS	Carga de datos correspondientes a las áreas protegidas tanto marinas como terrestres	env_bio1.tsv	STG_PROTECTEDAREAS
IN_ENERGYBALANCES	Carga de datos correspondientes a todos aquellos aspectos relevantes del balance energético mundial	WorldEnergy Balances Highlights_final.xlsx	STG_ENERGYBALANCES
IN_URBANWASTES	Carga de datos correspondientes a generación y tratamiento de residuos urbanos	DataGeneric.xml	STG_URBANWASTES

1.2. Bloque TR (llenar las tablas de nuestro almacén)

El bloque «TR_» de procesos de ETL para llenar el modelo multidimensional del almacén tiene dos partes:

- ❖ Los procesos de carga y transformación de las dimensiones.
- ❖ Los procesos de las tablas de hechos.

El orden de ejecución es importante para que la carga de datos sea correcta. Las dimensiones se cargarán primero y, después, las tablas de hechos, si no ha habido errores.

Los procesos del bloque de carga y transformación de las dimensiones son los siguientes:

NOMBRE DEL ETL	DESCRIPCIÓN	TABLA DE ORIGEN	TABLA DESTINO
TR_DIM_Date	Carga y transformación de la dimensión temporal (fecha en la que se realiza la medición).	SQL	DIM_Date
TR_DIM_SDG	Carga y transformación de la dimensión de Objetivos de Desarrollo Sostenible.	STG_Objectives	DIM_SDG
TR_DIM_Country	Carga de la dimensión de países donde se localiza la medición del balance energético.	STG_Countries	Dim_Country
TR_DIM_EconomicActivitySector	Carga y transformación de la dimensión con datos del sector de la actividad económica al que se clasifica la medición.	STG_Investments	Dim_EconomicActivitySector
TR_DIM_TypeEquipmentInstallation	Carga de la dimensión con los tipos de equipamientos o instalación de la medición.	STG_Investments	Dim_TypeEquipmentInstallation
TR_DIM_Product	Carga de la dimensión con los productos que intervienen en la medición del balance energético.	STG_EnergyBalances	DIM_Product
TR_DIM_Region	Carga de la dimensión de la región se localiza la medición.	STG_Investments STG_ProtectedAreas STG_EnergyBalance STG_Urbanwastes	DIM_Region
TR_DIM_Measurement	Carga de la dimensión con información de las unidades de medida.	STG_ObjectivesAreas STG_Urbanwastes STG_Investments STG_ProtectedAreas	DIM_Measurement

Los procesos del bloque de carga y transformación de las tablas de hechos son:

NOMBRE DEL ETL	DESCRIPCIÓN	TABLA DE ORIGEN
TR_FACT_EnvironmentalMeasurements	Carga y transformación de la tabla de hechos «FACT_EnvironmentalMeasurements»	STG_Investments STG_ProtectedAreas STG_EnergyBalance STG_Urbanwastes
TR_FACT_EnergyBalances	Carga y transformación de la tabla de hechos «FACT_EnergyBalances»	STG_EnergyBalances

2. Diseño y desarrollo de los procesos ETL

En este apartado, combinado con las consideraciones anteriores, usaremos Pentaho Data Integration (PDI) para diseñar e implementar los procesos de carga.

Los procesos de ETL que diseñaremos en PDI consistirán en la definición de trabajos y transformaciones.

2.1. Creación de tablas intermedias (staging area)

El primer paso para la implementación del proceso de ETL consiste en la creación de las tablas intermedias en la staging area. Esta se llevará a cabo una única vez, mediante scripts sobre la base de datos, en nuestro caso SQL Server.

Las tablas intermedias se utilizarán en los procesos IN, que permitirán cargar los datos desde las fuentes de datos.

2.1.1. IN_OBJECTIVES

Inicialmente añadimos los objetivos ODS:

```
CREATE TABLE [dbo].[STG_Objectives] (
    [Objetivo] [float] NULL,
    [Nombre] [varchar](100) NULL,
    [Descripción] [varchar](512) NULL
) ON [PRIMARY]
```

Una vez añadido los objetivos se añaden los ámbitos a áreas que abarcan estos objetivos:

```
CREATE TABLE [dbo].[STG_ObjectivesAreas](
    [Codigo] [varchar](100) NULL,
    [Ambito/VAR/Flow] [varchar](512) NULL,
    [ODS principal] [float] NULL
) ON [PRIMARY]
```

Con estas dos tablas ya se ha completado la estructura donde se va a cargar las dos pestañas del fichero ODS.xlsx

2.1.2. IN_INVESTMENTS

```
CREATE TABLE [dbo].[STG_Investments](
    [Periodo] [int] NOT NULL,
    [Inversion] [float] NULL,
    [sector_economico] [varchar](100) NULL,
    [tipo_instalacion] [varchar](100) NULL,
    [ambito_medioambiental] [varchar](100) NULL,
    [comunidad_autonoma] [varchar](27) NULL
) ON [PRIMARY]
```

Con esta tabla ya se ha completado la estructura donde se va a cargar el fichero 02002.xlsx

2.1.3. IN_COUNTRIES

```
CREATE TABLE [dbo].[STG_Countries](
    [Nombre] [varchar](100) NOT NULL,
    [Name] [varchar](100) NOT NULL,
    [Nom] [varchar](100) NOT NULL,
    [ISO2] [varchar](2) NULL,
    [ISO3] [varchar](3) NULL,
    [phone_code] [varchar](100) NULL
) ON [PRIMARY]
```

A nivel de seguridad en la carga, en la ISO2 e ISO3 se le ha puesto el tamaño máximo que puede tener ambas.

Con esta tabla ya se ha completado la estructura donde se va a cargar el fichero Countries.json

2.1.4. IN_PROTECTEDAREAS

```
CREATE TABLE [dbo].[STG_ProtectedAreas](
    [Year] [varchar](4) NULL,
    [areaprot] [varchar](100) NULL,
    [value] [bigint] NULL,
    [geoTime] [varchar](2) NULL
) ON [PRIMARY]
```

Con esta tabla ya se ha completado la estructura donde se va a cargar el fichero env_bio1.tsv

2.1.5. IN_ENERGYBALANCES

Esta tabla ya ha sido cargada en la BBDD en la PEC2, la sentencia SQL usada fue:

```
CREATE TABLE [dbo].[STG_EnergyBalance](
    [country] [varchar] (255) NULL,
    [product] [varchar] (255) NULL,
    [flow] [varchar] (255) NULL,
    [year] [int] NULL,
    [value] [float] NULL
) ON [PRIMARY]
```

Con esta tabla ya se ha completado la estructura donde se va a cargar el fichero WorldEnergyBalancesHighlights_final.xlsx

2.1.6. IN_URBANWASTES

```
CREATE TABLE [dbo].[STG_Urbanwastes] (
    [Cou] [varchar](100) NULL,
    [Var] [varchar](100) NULL,
    [Time_Format] [varchar](100) NULL,
    [Unit] [varchar](100) NULL,
    [PowerCode] [int] NULL,
    [OBS_Status] [varchar](100) NULL,
    [Time] [int] NULL,
    [ObsValue] [float] NULL
) ON [PRIMARY]
```

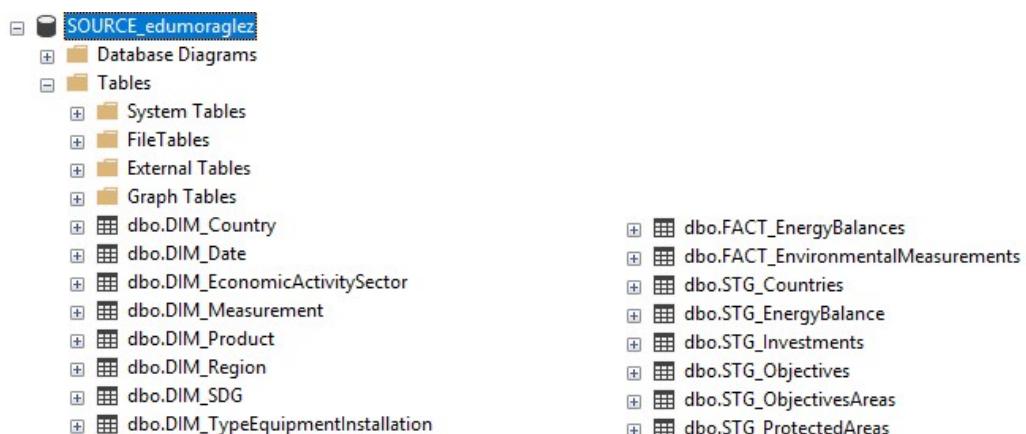
Con esta tabla ya se ha completado la estructura donde se va a cargar el fichero DataGeneric.xml

2.2. Creación del modelo multidimensional

Las tablas relativas a las Dimensiones y las Tablas de hechos se han añadido gracias al script proporcionado en el enunciado de la práctica.

2.3. Base de datos final

Finalmente, y para comprobar que se han añadido todas las tablas a la BBDD, se mostrará una captura en donde aparece el nombre de la BBDD y todas las tablas:

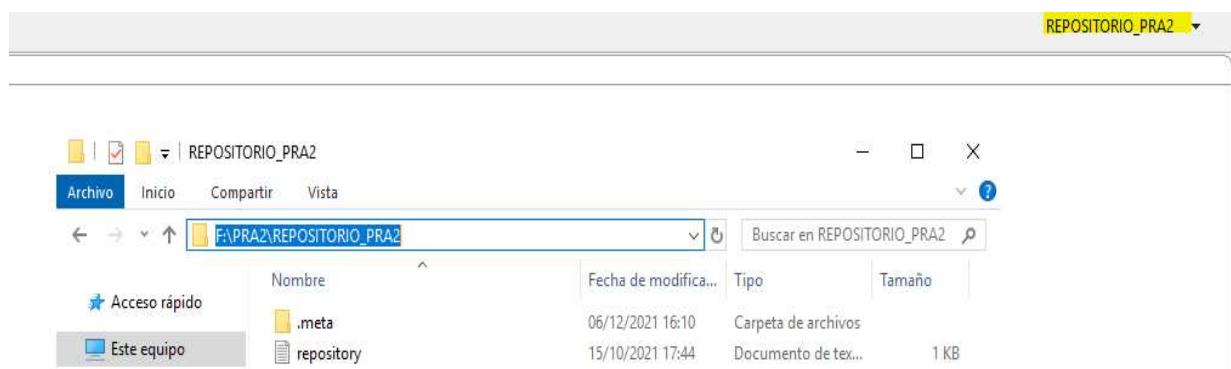


2.4. Creación del proceso de extracción, transformación y carga

Una vez creado el modelo físico del almacén, se va a pasar a crear y diseñar los procesos ETL que permitirán poblar las tablas intermedias del área intermedia (staging area) y las tablas de dimensiones y de hechos del Data Mart.

2.4.1. Crear repositorio de trabajo

La primera cosa que se debe realizar es crear un nuevo repositorio en donde se va a almacenar todas las transformaciones y trabajos de nuestro almacén de datos.



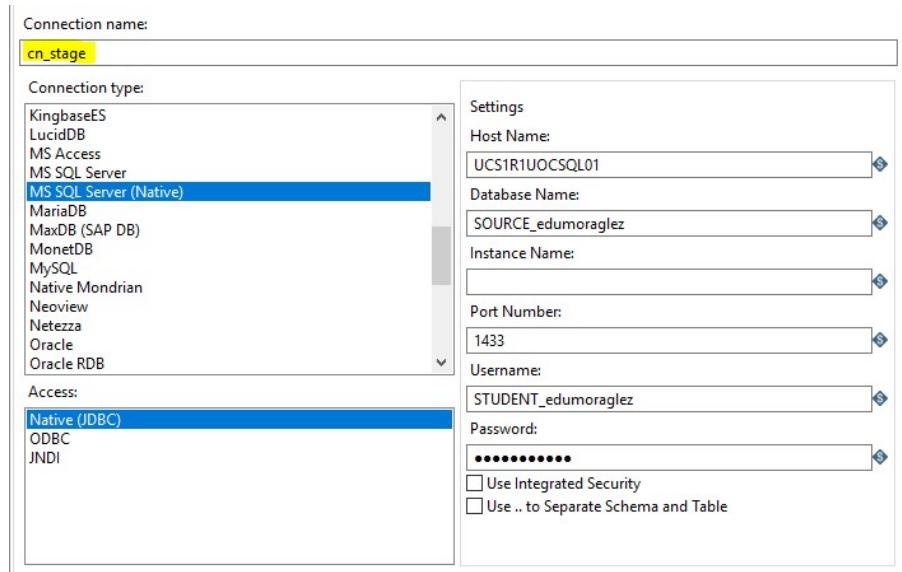
Como se puede comprobar en la imagen, se ha sincronizado el repositorio y se muestra la ruta donde estarán todos los elementos necesarios.

2.4.2. Conexión a la base de datos SQL Server

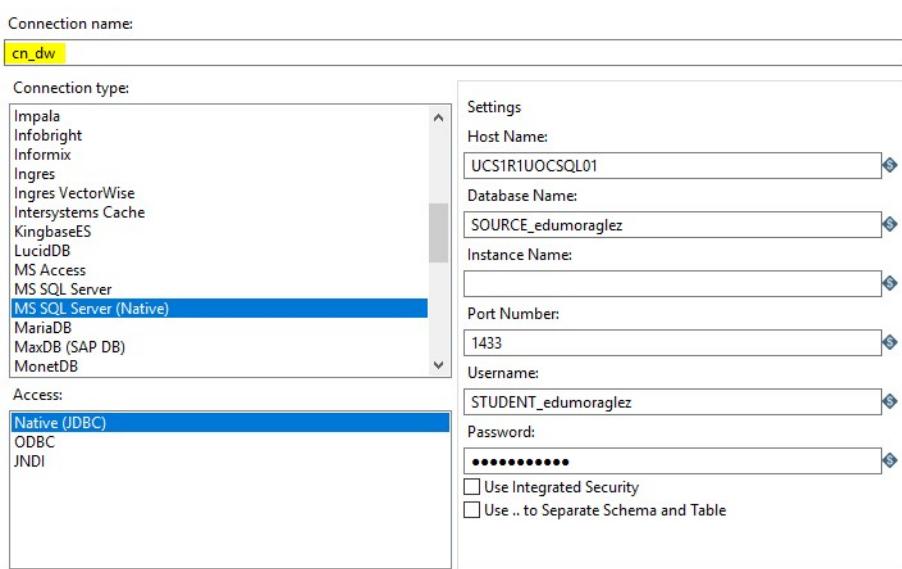
Otro paso previo que se debe realizar es crear las conexiones a las bases de datos que se usan en todas las transformaciones y trabajos de los procesos de carga.

Para facilitar las conexiones, se van a crear dos conexiones diferentes, una para la base de datos del modelo multidimensional y otra para el área intermedia.

En la creación de la conexión al «STAGE», el nombre usado es «cn_stage»:



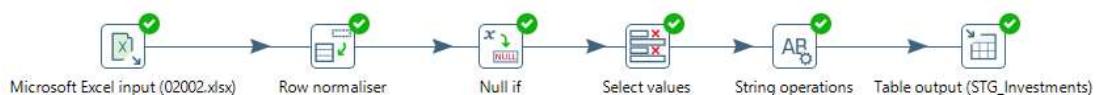
En la creación de la conexión al «DW», el nombre usado es «cn_dw»:



2.4.3. Bloque IN

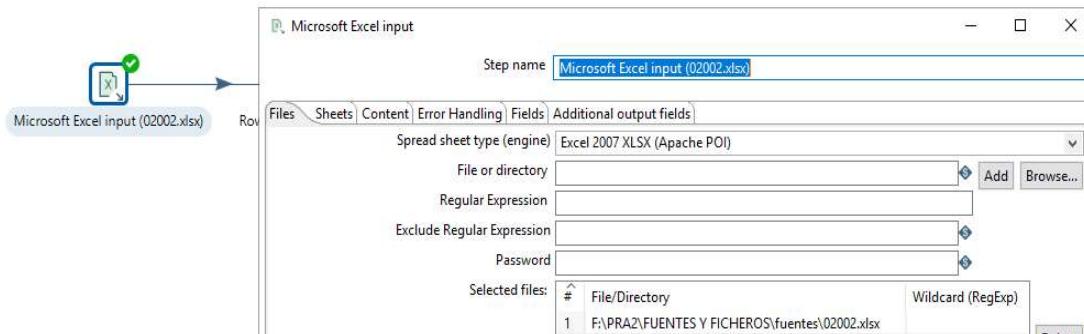
2.4.3.1. Transformación IN_INVESTMENTS

La carga de la fuente 02002.xlsx era una de las que se adjuntaban junto el enunciado, pero se hará un análisis para ver cómo funciona y explicar el paso que se ha añadido. La transformación completa es la siguiente:



Como se puede ver, consta de 6 pasos:

- **Carga del fichero:** en donde se selecciona el fichero a cargar:



Y como se deben cargar los campos:

#	Name	Type	Length	Precision	Trim type	Repeat	Format	Currency
1	Periodo	String	-1	-1	both	N		
2	Sector	String	-1	-1	both	N		
3	Tipo	String	-1	-1	both	N		
4	Ámbito	String	-1	-1	both	N		
5	Andalucía	String	-1	-1	none	N		
6	Aragón	String	-1	-1	none	N		
7	Asturias, Principado de	String	-1	-1	none	N		
8	Balears, Illes	String	-1	-1	none	N		
9	Canarias	String	-1	-1	none	N		
10	Cantabria	String	-1	-1	none	N		
11	Castilla y León	String	-1	-1	none	N		
12	Castilla-La Mancha	String	-1	-1	none	N		
13	Cataluña	String	-1	-1	none	N		
14	Comunitat Valenciana	String	-1	-1	none	N		
15	Extremadura	String	-1	-1	none	N		
16	Galicia	String	-1	-1	none	N		
17	Madrid, Comunidad de	String	-1	-1	none	N		
18	Murcia, Región de	String	-1	-1	none	N		
19	Navarra, Comunidad Foral de	String	-1	-1	none	N		
20	País Vasco	String	-1	-1	none	N		
21	Rioja, La	String	-1	-1	none	N		

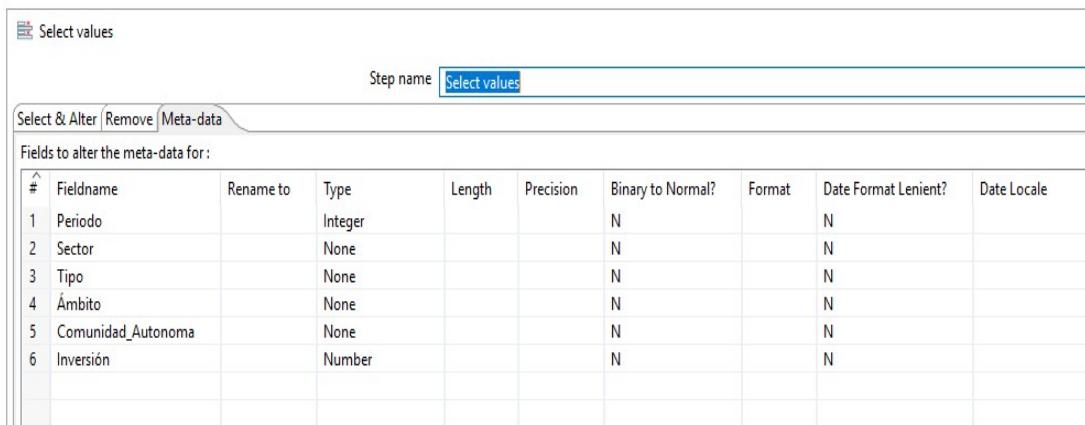
- **Normalización de filas:** se normalizan las filas relativas a las comunidades autónomas:

Fields		
#	Fieldname	Type
1	Andalucía	Andalucía
2	Aragón	Inversión
3	Asturias, Principado de	Inversión
4	Balears, Illes	Inversión
5	Canarias	Inversión
6	Cantabria	Inversión
7	Castilla y León	Inversión
8	Castilla-La Mancha	Inversión
9	Cataluña	Inversión
10	Comunitat Valenciana	Inversión
11	Extremadura	Inversión
12	Galicia	Inversión
13	Madrid, Comunidad de	Inversión
14	Murcia, Región de	Inversión
15	Navarra, Comunidad Foral de	Inversión
16	País Vasco	Inversión
17	Rioja, La	Inversión

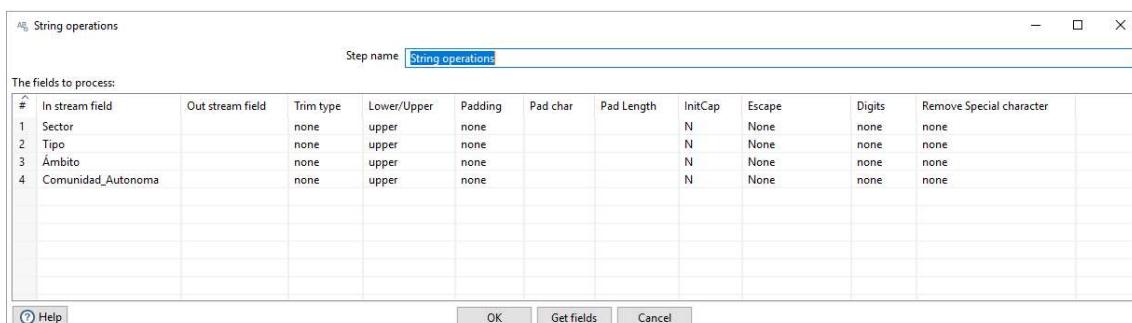
- **Tratamientos de nulos:** como se ha mencionado antes, hay comunidades que no han realizado alguna inversión en un sector o tipo, por eso en el fichero fuente aparecen “..”, y ahora se consideran como nulos:



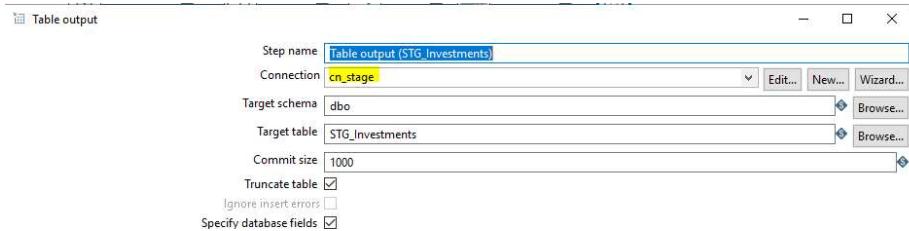
- **Selección de valores:** antes de añadir los valores a la BBDD se formalizan y se preparan:



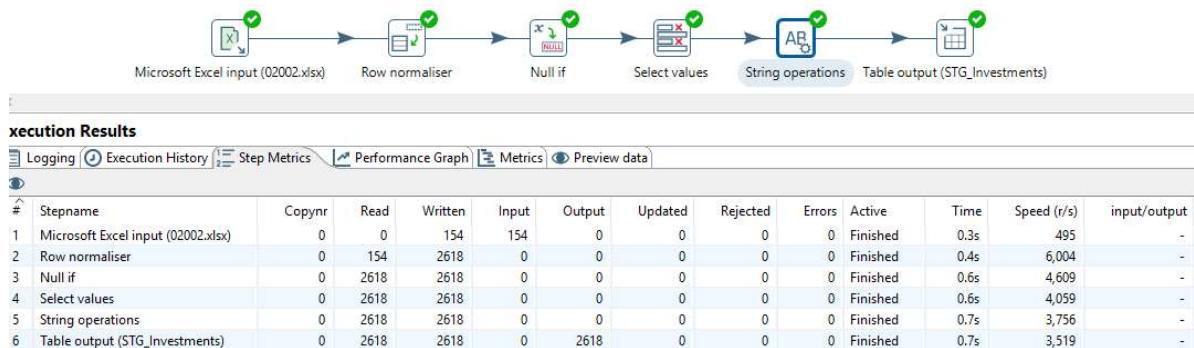
- **Operaciones con cadenas:** El siguiente paso de la transformación es asegurar la calidad de los datos mediante la normalización de los valores de los campos de tipo String que se consideren necesarios, es este caso es poner todas las cadenas en mayúsculas:



- **Carga en la BBDD:** una vez terminado todas las transformaciones, se añaden a la BBDD, para ello se usa la conexión de nombre <>cn_stage<>:



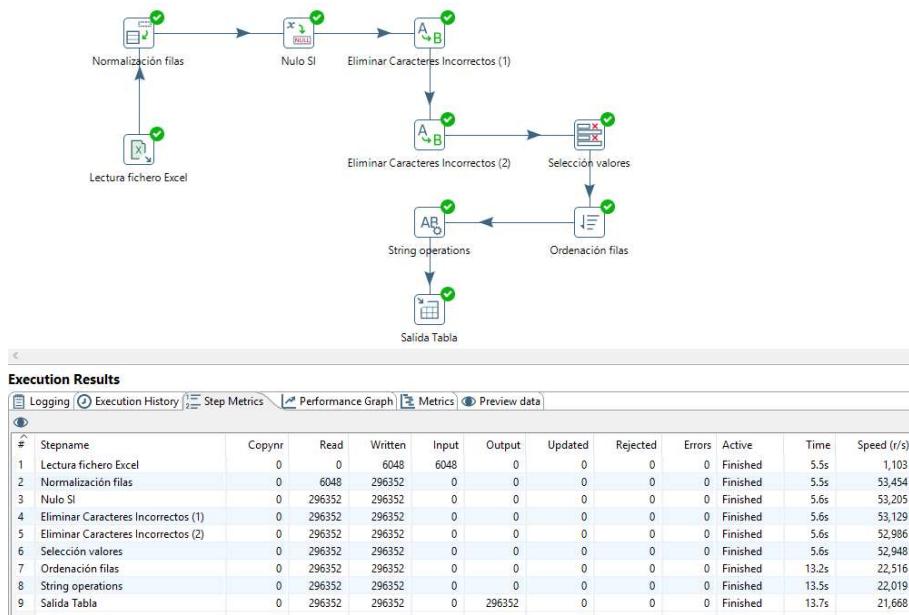
Para comprobar que todo ha funcionado de manera correcta, se mostraran las estadísticas una vez completado todo el proceso:



2.4.3.2. Transformación IN_ENERGYBALANCES

Esta carga se realizó en la PCE2, pero al igual que en la transformación anterior se ha modificado para obtener todos los campos String en mayúsculas.

Como es un paso bastante sencillo, se va a mostrar solamente las estadísticas del proceso terminado:



2.4.3.3. Transformación IN_OBJECTIVES

Como en el modelo de la BBDD, el fichero ODS.xlsx tiene dos pestañas distintas, por lo que se debe realizar dos transformaciones distintas.

STG_OBJECTIVES

La transformación completa es la siguiente:

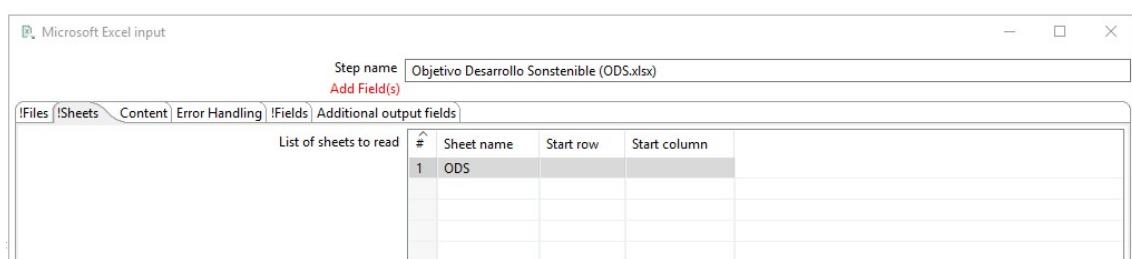


Como se puede observar en la imagen anterior, esta transformación contiene cuatro pasos:

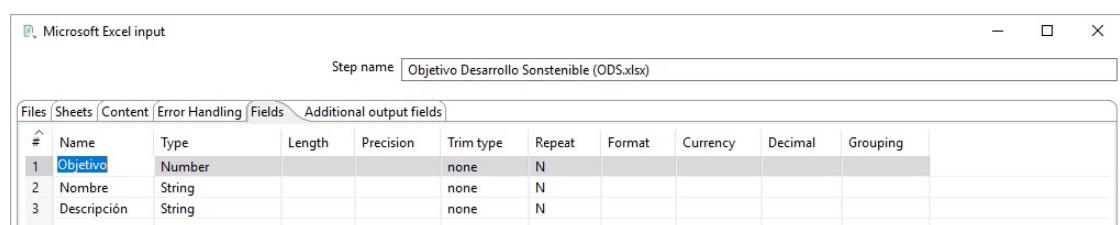
- **Lectura del fichero XLSX:** donde seleccionamos el fichero:



La hoja:



y la forma de importar los campos:



Para comprobar que se han cargado de manera correcta, se hará una visualización previa de los datos:

Rows of step: Objetivo Desarrollo Sostenible (ODS.xlsx) (17 rows)			
#	Objetivo	Nombre	Descripción
1	1.0	Fin de la pobreza	Para lograr este Objetivo de acabar con la pobreza, el crecimiento económico debe ser inclusivo, con el fin de crear empleos sostenibles y de promover la igualdad.
2	2.0	Hambre cero	El sector alimentario y el sector agrícola ofrecen soluciones clave para el desarrollo y son vitales para la eliminación del hambre y la pobreza.
3	3.0	Salud y bienestar	Para lograr los Objetivos de Desarrollo Sostenible, es fundamental garantizar una vida saludable y promover el bienestar universal.
4	4.0	Educación de calidad	La educación es la base para mejorar nuestra vida y el desarrollo sostenible.
5	5.0	Igualdad de género	La igualdad entre los géneros no es solo un derecho humano fundamental, sino la base necesaria para conseguir un mundo pacífico, próspero y sostenible.
6	6.0	Agua limpia y saneamiento	El agua libre de impurezas y accesible para todos es parte esencial del mundo en que queremos vivir.
7	7.0	Energía asequible y no contaminante	La energía es central para casi todos los grandes desafíos y oportunidades a los que se enfrenta el mundo en la actualidad.
8	8.0	Trabajo decente y crecimiento económico	Debemos reflexionar sobre este progreso lento y desigual, y revisar nuestras políticas económicas y sociales destinadas a erradicar la pobreza.
9	9.0	Industria, innovación e infraestructuras	Las inversiones en infraestructura son fundamentales para lograr un desarrollo sostenible.
10	10.0	Reducción de las desigualdades	Reducir la desigualdad en y entre los países.
11	11.0	Ciudades y comunidades sostenibles	Las inversiones en infraestructura son cruciales para lograr el desarrollo sostenible.
12	12.0	Producción y consumo responsables	El objetivo del consumo y la producción sostenibles es hacer más y mejores cosas con menos recursos.
13	13.0	Acción por el clima	El cambio climático es un reto global que no respeta las fronteras nacionales.

Como se han cargado de una manera correcta, podemos pasar al siguiente paso.

- **Operaciones con cadenas:** El siguiente paso de la transformación es asegurar la calidad de los datos mediante la normalización de los valores de los campos de tipo String que se consideren necesarios, que en este caso es poner todos los caracteres en mayúsculas:

String operations											
Step name <input type="text" value="String operations"/>											
The fields to process:											
#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits	Remove Special character
1	Nombre		none	upper	none			N	None	none	none
2	Descripción		none	upper	none			N	None	none	none

- **Ordenación de las filas:** A continuación, hay que ordenar de manera ascendente el campo objetivo:

Sort rows						
Step name <input type="text" value="Sort rows"/>						
Sort directory <input type="text" value="%java.io.tmpdir%"/> <input type="button" value="Browse..."/>						
TMP-file prefix <input type="text" value="out"/>						
Sort size (rows in memory) <input type="text" value="1000000"/>						
Free memory threshold (in %) <input type="text"/>						
Compress TMP Files? <input type="checkbox"/>						
Only pass unique rows? (verifies keys only) <input type="checkbox"/>						
Fields :						
#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	Objetivo	Y	N	N	0	N

- **Carga a la tabla intermedia «STG Objectives»:** El paso de cargar los datos a la tabla intermedia del stage:

Para comprobar que se funciona de manera correcta, se mostrara las estadísticas:



#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Objetivo Desarrollo Sonstensible (ODS.xlsx)	0	0	17	17	0	0	0	0	Finished	0.1s	250	-
2	String operations	0	17	17	0	0	0	0	0	Finished	0.1s	205	-
3	Sort rows	0	17	17	0	0	0	0	0	Finished	0.1s	145	-
4	Table output (STG_Ojetivos)	0	17	17	0	17	0	0	0	Finished	0.1s	115	-

STG_OBJECTIVESAREAS

La transformación completa es la siguiente:



Como se puede observar la transformación contiene los siguientes pasos:

- Lectura del fichero XLSX:** donde seleccionamos el fichero:

La hoja:

#	Sheet name	Start row	Start column
1	Ambito_VAR_Flow-ODS	0	0

Y la forma de importar los campos:

#	Name	Type	Length	Precision	Trim type	Repeat	Format	Currency	Decimal	Grouping
1	Codigo	String			none	N				
2	Ambito/VAR/Flow	String			none	N				
3	ODS principal	Number			none	N				

Para comprobar que se han cargado de manera correcta, se hará una visualización previa de los datos:

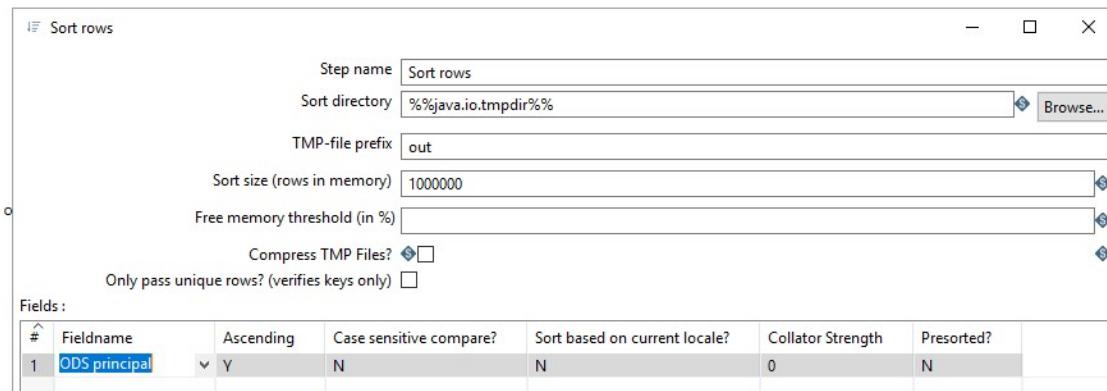
#	Codigo	Ambito/VAR/Flow	ODS principal
1	Protección del aire y el clima	Inversión en Protección del aire y el clima en €	13.0
2	Gestión de aguas residuales	Inversión en Gestión de aguas residuales en €	6.0
3	Gestión de residuos	Inversión en Gestión de residuos en €	12.0
4	Protección y descontaminación de suelos, aguas subterráneas y superficiales	Inversión en Protección y descontaminación de suelos, aguas subterráneas y superficiales en €	14.0
5	Reducción del ruido y las vibraciones	Inversión en Reducción del ruido y las vibraciones en €	3.0
6	Protección de la biodiversidad y los paisajes	Inversión en Protección de la biodiversidad y los paisajes en €	15.0
7	Otras actividades de protección ambiental	Inversión en Otras actividades de protección ambiental en €	15.0
8	BULKY	Bulky waste	9.0
9	COMPOST	Composting	7.0
10	COMPST_SHARE	% Composting	7.0
11	DISP_SHARE	% Disposal	7.0
12	DISPOSAL	Disposal operations	7.0
13	HOUSE_LIKE	Household and similar waste	11.0

Como se han cargado de una manera correcta, podemos pasar al siguiente paso.

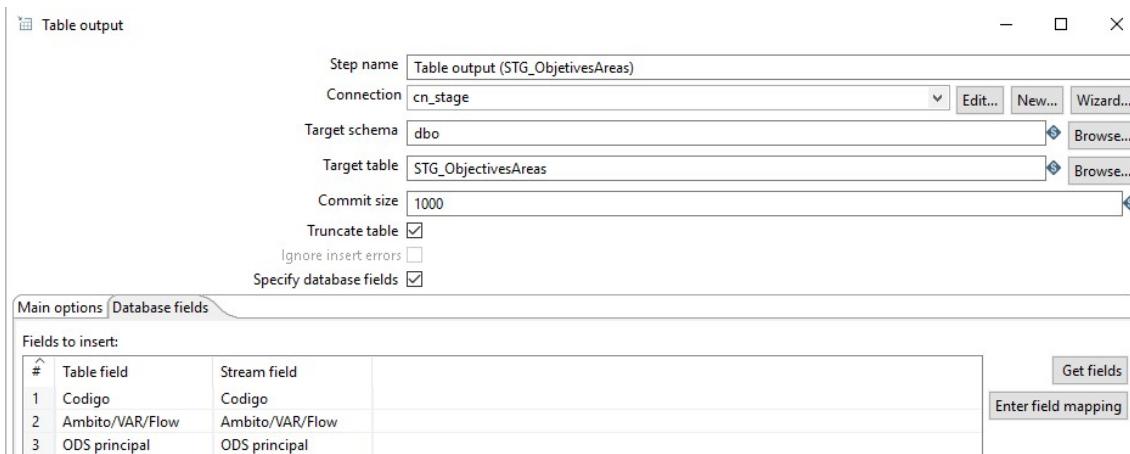
- Operaciones con cadenas:** El siguiente paso de la transformación es asegurar la calidad de los datos mediante la normalización de los valores de los campos de tipo String que se consideren necesarios, que será poner los campos String en mayúsculas:

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits	Remove Special character
1	Codigo		none	upper	none			N	None	none	none
2	Ambito/VAR/Flow		none	upper	none			N	None	none	none

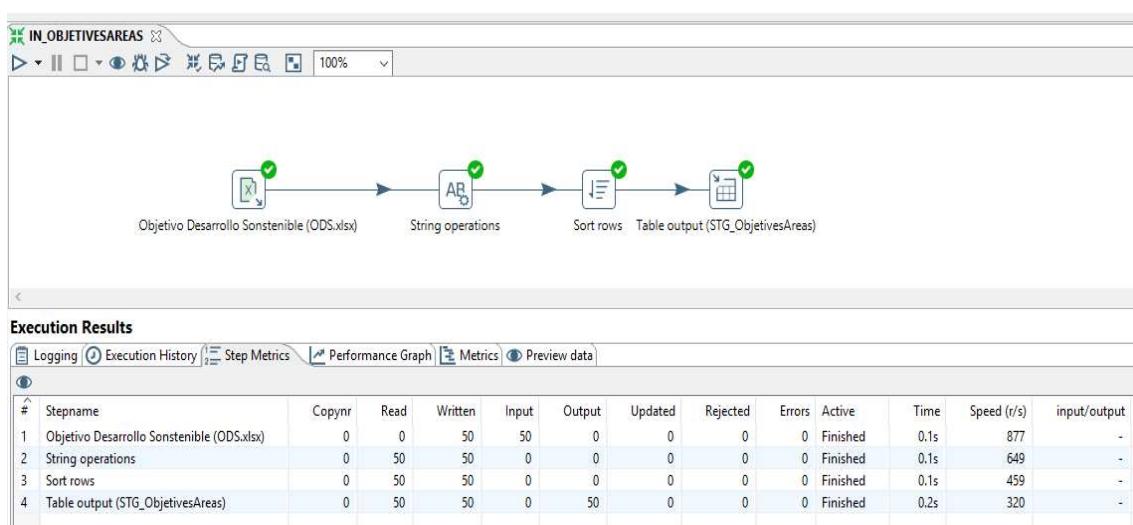
- **Ordenación de las filas:** A continuación, hay que ordenar de manera ascendente el campo referente al ODS principal:



- **Carga a la tabla intermedia STG_ObjectivesAreas:** El paso de cargar los datos a la tabla intermedia del stage:



Para comprobar que se funciona de manera correcta, se mostrara las estadísticas:



2.4.3.4. Transformación IN_COUNTRIES

La transformación completa es la siguiente:



La transformación consta de los siguientes pasos:

- **Lectura del fichero JSON:** donde seleccionamos el fichero:

The screenshot shows the 'JSON input' configuration window with the following details:
Step name: JSON input (Countries.json)
File tab selected.
Source from field section:

- Source is from a previous step:
- Select field:
- Use field as file names:
- Read source as URL:
- Do not pass field downstream:

File or directory: Add Browse
Regular Expression:
Exclude Regular Expression:
Selected files:

#	File/Directory	Wildcard (RegExp)	Exclude wildcard
1	F:\PRA2\FUENTES Y FICHEROS\fuentes\Country.json		

Delete button.

y los campos:

The screenshot shows the 'Fields' tab of the 'JSON input' configuration window with the following details:
Step name: JSON input (Countries.json)
Fields tab selected.

#	Name	Path	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	nombre	\$.nombre	String							
2	name	\$.name	String							
3	nom	\$.nom	String							
4	iso2	\$.iso2	String							
5	iso3	\$.iso3	String							
6	phone_code	\$.phone_code	String							

Para comprobar que se han cargado de manera correcta, se hará una visualización previa de los datos:

The screenshot shows the preview data table with the following data:
Rows of step: JSON input (Countries.json) (246 rows)

#	nombre	name	nom	iso2	iso3	phone_code
1	Afganistán	Afghanistan	Afghanistan	AF	AFG	93
2	Albania	Albania	Albanie	AL	ALB	355
3	Alemania	Germany	Allemagne	DE	DEU	49
4	Algeria	Algeria	Algérie	DZ	DZA	213
5	Andorra	Andorra	Andorra	AD	AND	376
6	Angola	Angola	Angola	AO	AGO	244

Como se han cargado de una manera correcta, podemos pasar al siguiente paso.

- **Operaciones con cadenas:** El siguiente paso de la transformación es asegurar la calidad de los datos mediante la normalización de los valores de los campos de tipo String que se consideren necesarios, que en este caso es poner todos los campos en mayúsculas:

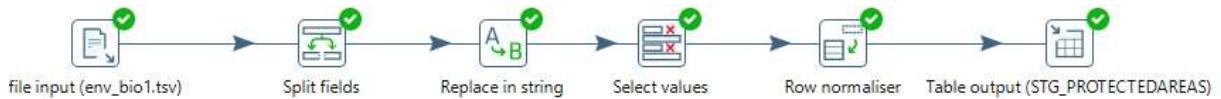
- **Carga a la tabla intermedia «STG_Countries»:** Como el fichero ya venía ordenado alfabéticamente, procedemos a insertar los datos en la tabla intermedia del stage:

Para comprobar que se funciona de manera correcta, se mostrara las estadísticas:

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	JSON input (Countries.json)	0	0	246	246	0	0	0	0	Finished	0.0s	5,591	-
2	String operations	0	246	246	0	0	0	0	0	Finished	0.1s	3,727	-
3	Table output (STG_COUNTRIES)	0	246	246	0	246	0	0	0	Finished	0.1s	2,177	-

2.4.3.5. Transformación IN_PROTECTEDAREAS

La transformación completa es la siguiente:



La transformación consta de los siguientes pasos:

- Lectura del fichero tsv:** donde seleccionamos el fichero y los delimitadores (un espacio en blanco):

CSV file input configuration:

- Step name: file input (env_bio1.tsv)
- Filename: F:\PRA2\FUENTES Y FICHEROS\fuentes\env_bio1.tsv
- Delimiter: (empty)
- Enclosure: (empty)
- NIO buffer size: 50000
- Lazy conversion?
- Header row present?
- Add filename to result
- The row number field name (optional): (empty)
- Running in parallel?
- New line possible in fields?
- Format: mixed
- File encoding: (empty)

Fields table:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	areaprot,geo\time	String		19	0	\$,	.	none
2	2019	Number	#,#	15	0	\$.	,	both
3	2018	Number	#,#	15	0	\$.	,	both
4	2017	String		8		\$,	.	none
5	2016	String		8		\$,	.	none
6	2015	String		8		\$,	.	none
7	2014	String		8		\$,	.	none
8	2013	String		8		\$,	.	none
9	2012	String		7		\$,	.	none
10	2011	String		7		\$,	.	none

Para comprobar que se han cargado de manera correcta, se hará una visualización previa de los datos:

#	areaprot,geo\time	2019	2018	2017	2016	2015	2014	2013	2012	2011
1	AREA_KM2,AT	83944	83944	83944	83944	83944	83944	83944	83944	83944
2	AREA_KM2,BE	30667	30667	30667	30667	30667	30667	30667	30667	30667
3	AREA_KM2,BG	110995	110995	110995	110995	110995	110995	110995	110995	110995
4	AREA_KM2,CY	5736	5736	5736	5736	5736	5736	5736	5736	5736
5	AREA_KM2,CZ	78874	78874	78874	78874	78874	78874	78874	78874	78874

Aunque se han cargado los datos hay que hacer ciertas transformaciones para obtener los datos de manera correcta.

- Dividir campos delimitados por comas:** el campo areaprot,geo\time se dividirá en dos variables separadas:

Split fields configuration:

- Step name: Split fields
- Field to split: areaprot,geo\time
- Delimiter: ,
- Enclosure: (empty)

Fields table:

#	New field	ID	Remove ID?	Type	Length	Precision	Format	Group	Decimal	Currency	Nullif	Default	Trim type
1	areaprot	N		String									none
2	GeoTime	N		String	2								none

Como salida de esta transformación, vemos que tenemos ahora dos campos separados:

#	areaprot	GeoTime	2019	2018	2017	2016	2015	2014	2013	2012	2011
1	AREA_KM2	AT	83944.0	83944	83944	83944	83944	83944	83944	83944	83944
2	AREA_KM2	BE	30667.0	30667.0	30667	30667	30667	30667	30667	30667	30667
3	AREA_KM2	BG	110995.0	110995.0	110995	110995	110995	110995	110995	110995	110995

- **Eliminar caracteres mal ingresados:** hay algunos casos en donde había una “s” después del número, y se ha añadido de la siguiente forma:

TPA_PC	EU	2818	18	18	\$18	\$18	\$18	\$18	\$1	:
TPA_PC	FI	13	13	13	\$13	\$13	\$13	\$13	\$13	\$13

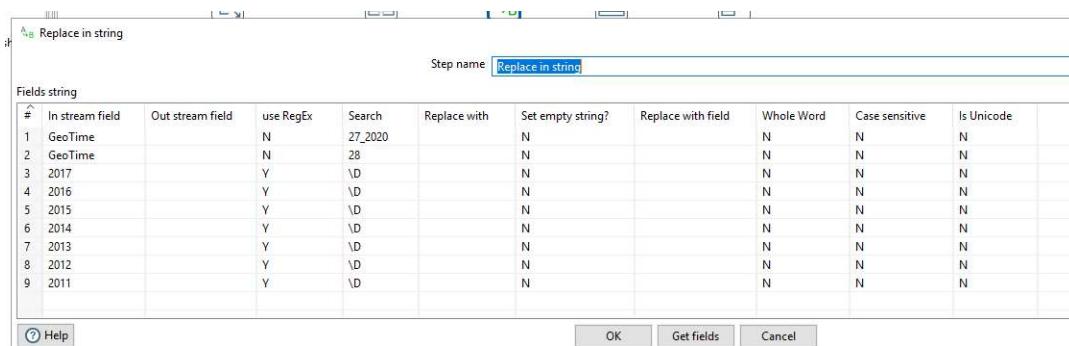
Además, se han añadido valores numéricos en el campo *GeoTime*:

1	AREA_KM2	EU 27_2020	4132405.0	4132405.0	4132405	4132405	4132405	4132405	4132405	:	:
2	AREA_KM2	EU 28	4376978.0	4376978.0	4376978	4376978	4376978	4376978	4376978	:	:

Haciendo un análisis del fichero, nos damos cuenta de que siempre son los mismos valores “27_2020” y “28”.

Para todos estos casos se hará las siguientes operaciones:

- Para los campos referente a los años solamente se permitirán caracteres numéricos.
- Para el campo *GeoTime* solamente nos quedamos con los 2 primeros dígitos.



- **Seleccionar los campos y el formato:** se le cambiaron el formato a los campos para que sean los correctos antes de añadirlos a la BBDD:

#	Fieldname	Rename to	Type	Length	Precision	Binary to Normal?	Format	Date Format Lenient?	Date Locale
1	areaprot		None			N		N	
2	GeoTime		None	2		N		N	
3	2019		Integer	15		N		N	
4	2018		Integer	15		N		N	
5	2017		Integer	15		N		N	
6	2016		Integer	15		N		N	
7	2015		Integer	15		N		N	
8	2014		Integer	15		N		N	
9	2013		Integer	15		N		N	
10	2012		Integer	15		N		N	
11	2011		Integer	15		N		N	

- **Normalización de columnas:** ahora que tenemos todos los datos de una manera correcta, vamos a normalizar los años y sus valores de la siguiente manera:

#	Fieldname	Type	new field
1	2019	2019	Value
2	2018	2018	Value
3	2017	2017	Value
4	2016	2016	Value
5	2015	2015	Value
6	2014	2014	Value
7	2013	2013	Value
8	2012	2012	Value
9	2011	2011	Value

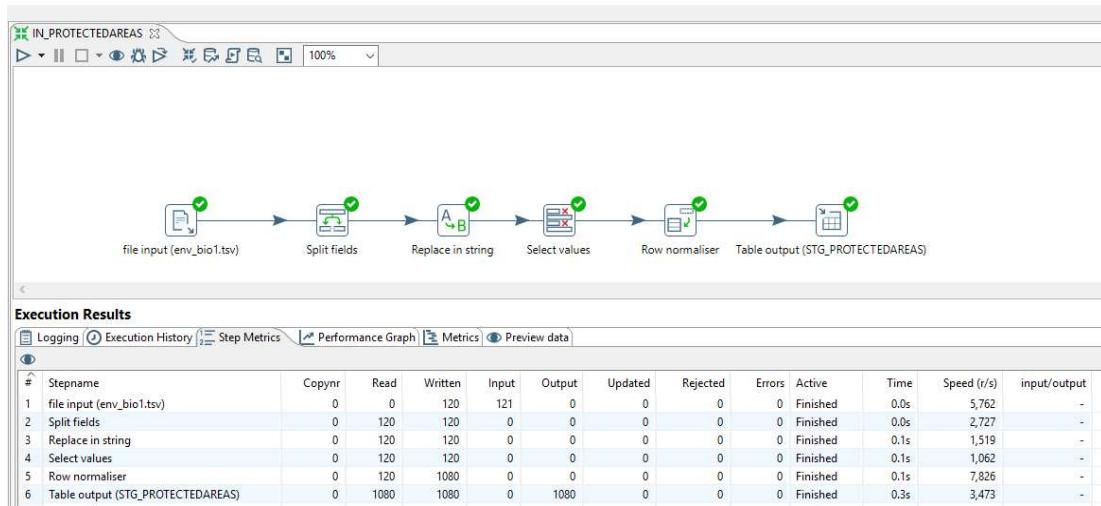
Gracias a esta operación los campos vacíos se han añadido como nulos:

8	AREA_KM2	EU	2012	<null>
9	AREA_KM2	EU	2011	<null>

- **Carga a la tabla intermedia «STG_PROTECTEDAREAS»:** procedemos a insertar los datos en la tabla intermedia del stage:

#	Table field	Stream field
1	areaprot	areaprot
2	GeoTime	GeoTime
3	Year	Year
4	Value	Value

Para comprobar que se funciona de manera correcta, se mostrara las estadísticas:



2.4.3.6. Transformación IN_URBANWASTES

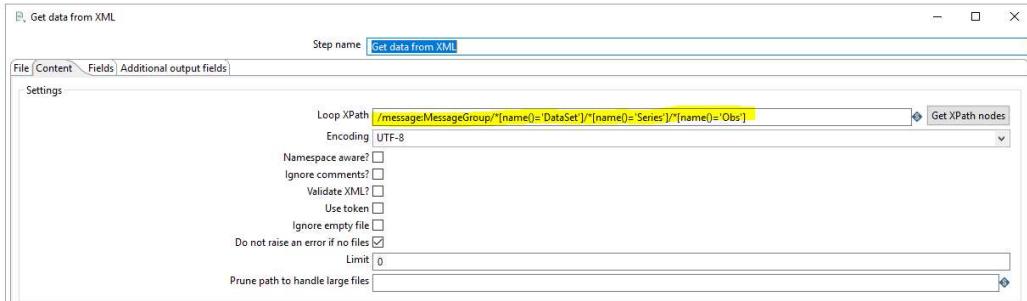
La transformación completa es la siguiente:



La transformación consta de los siguientes pasos:

- **Lectura del fichero XML:** donde seleccionamos el fichero a cargar:

Seleccionamos el *Loop Xpath* para saber dónde iterar:



Seleccionamos los campos a extraer:

#	Name	XPath	Element	Result type	Type
1	Time	*[name()='Time']	Node	Value of	Integer
2	ObsValue	*[name()='ObsValue']/@value	Node	Value of	String
3	COU	./*[name()='SeriesKey']/*[name()='Value' and @concept='COU']/@value	Node	Value of	String
4	VAR	./*[name()='SeriesKey']/*[name()='Value' and @concept='VAR']/@value	Node	Value of	String
5	TIME_FORMAT	./*[name()='Attributes']/*[name()='Value' and @concept='TIME_FORMAT']/@value	Node	Value of	String
6	UNIT	./*[name()='Attributes']/*[name()='Value' and @concept='UNIT']/@value	Node	Value of	String
7	POWERCODE	./*[name()='Attributes']/*[name()='Value' and @concept='POWERCODE']/@value	Node	Value of	Integer
8	OBS_STATUS	*[name()='Attributes']/*[name()='Value' and @concept='OBS_STATUS']/@value	Node	Value of	String

Para comprobar que se han cargado de manera correcta, se hará una visualización previa de los datos:

Rows of step: Get data from XML (1000 rows)									
#	Time	ObsValue	COU	VAR	TIME_FORMAT	UNIT	POWERCODE	OBS_STATUS	
1	1992	12000	AUS	MUNICIPAL	P1Y	TONNE	3	E	
2	2000	13200	AUS	MUNICIPAL	P1Y	TONNE	3	E	
3	2007	12873	AUS	MUNICIPAL	P1Y	TONNE	3		
4	2008	13096.5	AUS	MUNICIPAL	P1Y	TONNE	3		
5	2009	13320	AUS	MUNICIPAL	P1Y	TONNE	3		
6	2010	13534	AUS	MUNICIPAL	P1Y	TONNE	3		
7	2011	13450	AUS	MUNICIPAL	P1Y	TONNE	3		
8	2012	13572.333	AUS	MUNICIPAL	P1Y	TONNE	3		
9	2013	13694.667	AUS	MUNICIPAL	P1Y	TONNE	3		
10	2014	13817	AUS	MUNICIPAL	P1Y	TONNE	3		
11	2015	14003	AUS	MUNICIPAL	P1Y	TONNE	3		
12	2016	13539	AUS	MUNICIPAL	P1Y	TONNE	3		
13	2017	13751	AUS	MUNICIPAL	P1Y	TONNE	3		
14	1992	7000	AUS	HOUSEHOLD	P1Y	TONNE	3	E	

Aunque se han cargado los datos hay que hacer ciertas transformaciones para obtener los datos de manera correcta.

- **Operaciones con cadenas:** El siguiente paso de la transformación es asegurar la calidad de los datos mediante la normalización de los valores de los campos de tipo String que se consideren necesarios:

String operations

Step name **String operations**

The fields to process:

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits	Remove Special character
1	COU		none	upper	none			N	None	none	none
2	VAR		none	upper	none			N	None	none	none
3	TIME_FORMAT		none	upper	none			N	None	none	none
4	UNIT		none	upper	none			N	None	none	none
5	OBS_STATUS		none	upper	none			N	None	none	none

- Reemplazar los caracteres no numéricos:** para preparar los datos antes de transformar el tipo:

Replace in string

Step name **Replace in string**

Fields string

#	In stream field	Out stream field	use RegEx	Search	Replace with	Set empty string?	Replace with field	Whole Word	Case sensitive	Is Unicode
1	ObsValue		Y	\D		N		N	N	N

- Cambio de formato:** transformamos el campo ObsValue de String a numérico.

Select values

Step name **Select values**

Select & Alter Remove Meta-data

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length	Precision	Binary to Normal?	Format	Date Format Lenient?
1	ObsValue		Number			N	#,#	N

- Ordenación de las filas:** A continuación, hay que ordenar de manera ascendente los campos:

Sort rows

Step name **Sort rows**

Sort directory **%%java.io.tmpdir%%**

TMP-file prefix **out**

Sort size (rows in memory) **1000000**

Free memory threshold (in %)

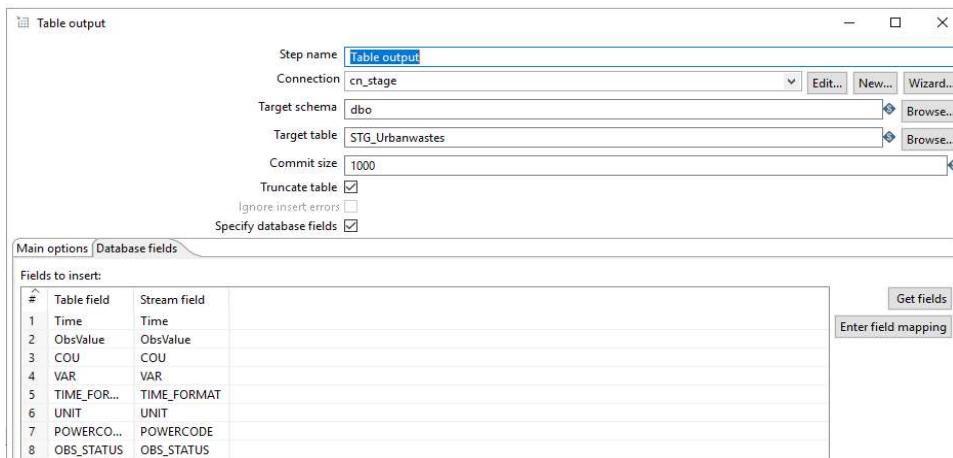
Compress TMP Files?

Only pass unique rows? (verifies keys only)

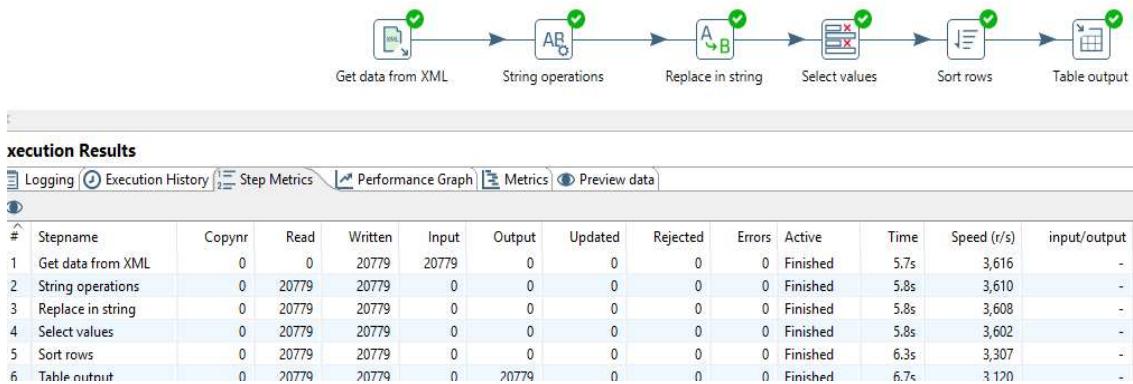
Fields :

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	Time	Y	N	N	0	N
2	COU	Y	N	N	0	N
3	VAR	Y	N	N	0	N
4	TIME_FORMAT	Y	N	N	0	N
5	UNIT	Y	N	N	0	N
6	POWERCODE	Y	N	N	0	N
7	OBS_STATUS	Y	N	N	0	N

- Carga a la tabla intermedia «STG_URBANWASTES»:** procedemos a insertar los datos en la tabla intermedia del stage:



Para comprobar que se funciona de manera correcta, se mostrara las estadísticas:



2.4.4. Bloque TR

El bloque TR contiene los procesos de ETL, que se encargan de la carga inicial de datos, desde las tablas intermedias pobladas con los procesos del bloque IN, al modelo multidimensional del almacén, compuesto por dimensiones y tablas de hechos.

Este bloque se divide, a su vez, en dos bloques: por un lado, los procesos para la carga de dimensiones y, por el otro, los procesos para la carga de tablas de hechos.

2.4.5. Bloque TR_DIM

Las dimensiones:

- ✓ TR_DIM_Country
- ✓ TR_DIM_EconomicActivitySector

- ✓ TR_DIM_TypeEquipmentInstallation

Venían incluidas entre los ficheros del enunciado, por lo que se obviaran su diseño, tratando solamente las nuevas dimensiones.

2.4.5.1. TR_DIM_Date

La carga de esta dimensión es diferente a la carga de las dimensiones de datos, la opción elegida es a través de un script SQL para generar todos los registros necesarios.

```
-- Declaración y establecimiento de fecha inicio y fecha fin (fecha actual)
DECLARE @FechaInicio datetime;
DECLARE @FechaFin datetime;
SET @FechaInicio = '01/01/2020';
```

El script SQL completo es el siguiente:

```
-- Declaración y establecimiento de fecha inicio y fecha fin (fecha actual)
DECLARE @FechaInicio datetime;
DECLARE @FechaFin datetime;
SET @FechaInicio = '01/01/1970';
SET @FechaFin = GETDATE();

-- Declaración y establecimiento de fecha ciclo
DECLARE @FechaCiclo datetime;
SET @FechaCiclo = @FechaInicio;

-- Bucle hasta fecha fin
WHILE @FechaCiclo <= @FechaFin
BEGIN

-- Insertar un registro en la dimensión Fecha
INSERT INTO DIM_Date VALUES (
cast(cast(Year(@FechaCiclo) as
varchar(4))+right('0'+cast(Month(@FechaCiclo) as
varchar(2)),2)+right('0'+cast(Day(@FechaCiclo) as varchar(2)),2)
as int),
```

```

Year(@FechaCiclo),
Month(@FechaCiclo),
Day(@FechaCiclo),
@FechaCiclo
)

-- Incrementar la FechaCiclo en un día
SET @FechaCiclo = DateAdd(d, 1, @FechaCiclo)
END

```

Para comprobar que se funciona de manera correcta, se mostrara las estadísticas:

The screenshot shows the SSIS Execute SQL Script step icon with a green checkmark. Below it is the 'Execute SQL script' task. The 'Execution Results' window displays the following metrics for the step:

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Execute SQL script	0	0	1	0	0	0	0	0	Finished	0.0s	62	-

2.4.5.2. TR_DIM_SDG

La transformación completa es la siguiente:



Esta transformación consta de los siguientes pasos:

- Obtención de los valores:** Mediante una sentencia «SELECT» de SQL
(donde se trata el diseño de los registros NA)

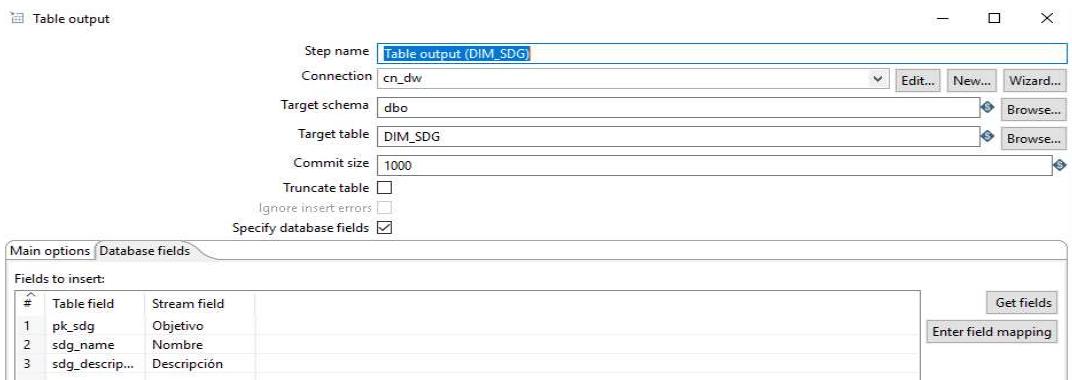
The screenshot shows the 'Table input' configuration dialog. The 'Step name' is set to 'Table input (STG_SDG)' and the 'Connection' is set to 'cn_stage'. The 'SQL' section contains the following query:

```

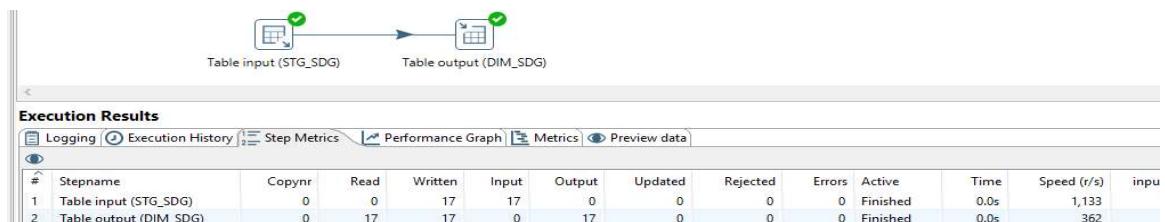
SELECT
    Objetivo,
    Nombre,
    Descripción
FROM dbo.STG_Objectives
UNION
SELECT
    NULL as Objetivo,
    'NA' as Nombre,
    'NA' as Descripción
ORDER BY Objetivo

```

- **Carga los datos:** a la tabla de dimensión DIM_SDG del modelo dimensional en el DataMart:



Para comprobar que se funciona de manera correcta, se mostrara las estadísticas:



Notad que no se añade la secuencia, ya que el código de SDG será el identificador.

2.4.5.3. TR_DIM_Region

La transformación completa es la siguiente:



Esta transformación consta de los siguientes pasos:

- **Obtención de los valores:** Mediante una sentencia «SELECT» de SQL en cada una de las 4 tablas en donde se especifican los países o regiones. La sentencia es la siguiente:

```
-- TABLA STG_Investments
SELECT DISTINCT
    t1.comunidad_autonoma as region,
    t2.country_code as country_code2,
```

```

        t2.country_code3 as country_code3,
        t2.country_name_en as country_name
    FROM [dbo].[STG_Investments] t1, [dbo].DIM_Country t2
    WHERE t2.country_name_en = 'Spain'

UNION

SELECT DISTINCT -- TABLA STG_ProtectedAreas
    'NA' as region,
    t1.geoTime as country_code2,
    t2.country_code3 as country_code3,
    t2.country_name_en as country_name
FROM [dbo].STG_ProtectedAreas t1, [dbo].DIM_Country t2
WHERE t2.country_code = t1.geoTime

UNION

SELECT DISTINCT -- TABLA STG_EnergyBalance
    'NA' as region,
    t2.country_code as country_code2,
    t2.country_code3 as country_code3,
    t2.country_name_en as country_name
FROM [dbo].STG_EnergyBalance t1, [dbo].DIM_Country t2
WHERE t2.country_name_en = t1.country

UNION

SELECT DISTINCT -- TABLA STG_Urbanwastes
    'NA' as region,
    t2.country_code as country_code2,
    t1.Cou as country_code3,
    t2.country_name_en as country_name
FROM [dbo].STG_Urbanwastes t1, [dbo].DIM_Country t2
WHERE t2.country_code3 = t1.Cou

UNION

SELECT -- NULOS
    'NA' as region,
    'NA' as country_code2,
    'NA' as country_code3,
    'NA' as country_name

```

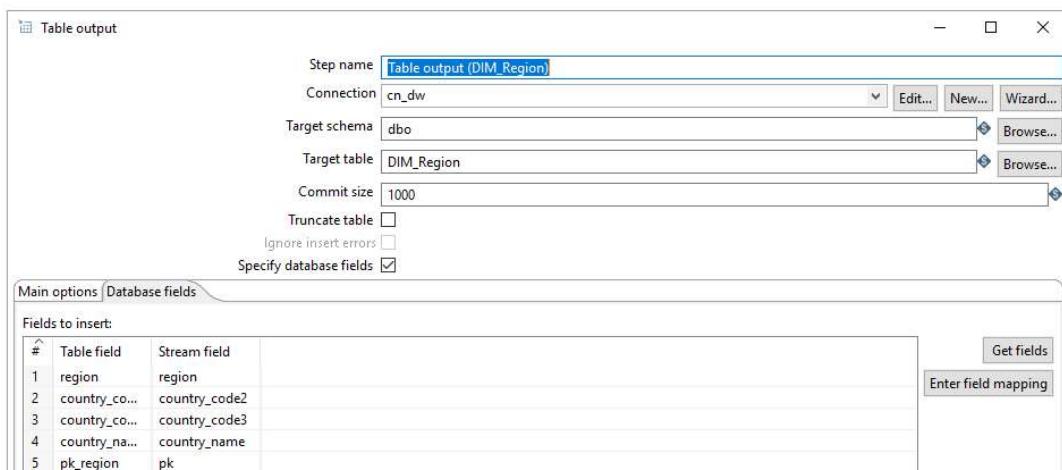
Como se puede observar, solamente en la tabla *STG_Investments* se especifica la comunidad autónoma, en el resto solamente el país, por eso ese valor se pone a nulo en el resto de los casos. Además, solamente se añadirán los distintos valores así evitando añadir repetidos.



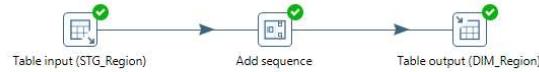
```
-- TABLA STG_Investments
SELECT DISTINCT
    t1.autonomia as region,
    t2.country_code as country_code2,
    t2.country_code3 as country_code3,
    t2.country_name_en as country_name
FROM [dbo].[STG_Investments] t1, [dbo].DIM_Country t2
WHERE t2.country_name_en = 'Spain'
UNION -- TABLA STG_ProtectedAreas
SELECT DISTINCT
```

- **Creación de una secuencia:** que hará las funciones de clave primaria de incremento automático.
- **Carga los datos:** a la tabla de dimensión *DIM_Region* del modelo dimensional en el DataMart. Se ha modificado el campo Región para que permita más caracteres. La sentencia usada ha sido la siguiente:

```
ALTER TABLE dbo.DIM_Region ALTER COLUMN region VARCHAR(27)
```



Para comprobar que se funciona de manera correcta, se mostrara las estadísticas:



Execution Results													
#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Table input (STG_Region)	0	0	63	63	0	0	0	0	Finished	0.2s	293	-
2	Add sequence	0	63	63	0	0	0	0	0	Finished	0.2s	279	-
3	Table output (DIM_Region)	0	63	63	0	63	0	0	0	Finished	0.3s	244	-

2.4.5.4. TR_DIM_Measurement

La transformación completa es la siguiente:



En esta transformación cargaremos los datos de distintas tablas, y siempre comparándolo con las distintas áreas de los objetivos SDG.

- **Obtención de los valores:** Mediante una sentencia «SELECT» de SQL

```

Table input
Step name: Table input
Connection: cn_stage
SQL
SELECT DISTINCT
    t2.Codigo as measurement_code,
    t2.[Ámbito/VAR/Flow] as measurement_name,
    t1.areaprot as Unit,
    t2.[ODS principal] as fk_sdg --Ya que es el Objetivo que trata sobre la proteccion de ecosistema
FROM dbo.STG_ProtectedAreas t1, dbo.STG_ObjectivesAreas t2
WHERE t1.areaprot = t2.Codigo
UNION
SELECT DISTINCT
    t1.Codigo as measurement_code,
    t1.[Ámbito/VAR/Flow] as measurement_name,
    t2.Unit as Unit,
    t1.fk_areaprot as fk_sdg

```

La sentencia SQL es la siguiente:

```

SELECT DISTINCT
    t2.Codigo as measurement_code,
    t2.[Ámbito/VAR/Flow] as measurement_name,
    t1.areaprot as Unit,
    t2.[ODS principal] as fk_sdg

FROM dbo.STG_ProtectedAreas t1, dbo.STG_ObjectivesAreas t2

```

```

WHERE t1.areaprot = t2.Codigo

UNION

SELECT DISTINCT
    t1.Codigo as measurement_code,
    t1.[Ambito/VAR/Flow] as measurement_name,
    t2.Unit as Unit,
    t1.[ODS principal] as fk_sdg

FROM dbo.STG_ObjectivesAreas t1, dbo.STG_Urbanwastes t2

WHERE t1.Codigo = t2.Var

UNION

SELECT DISTINCT
    t1.Codigo as measurement_code,
    t2.tipo_instalacion as measurement_name,
    'euros' as Unit,
    t1.[ODS principal] as fk_sdg

FROM dbo.STG_ObjectivesAreas t1, dbo.STG_Investments t2

WHERE t1.Codigo = t2.ambito_medioambiental

UNION

SELECT
    'NA' as measurement_code,
    'NA' as measurement_name,
    'NA' as Unit,
    0 as fk_sdg --CAMPO NULO DE LA TABLA SDG

ORDER BY fk_sdg

```

- **Creación de una secuencia:** que hará las funciones de clave primaria de incremento automático.
- **Carga los datos:** a la tabla de dimensión *DIM_Measurement* del modelo dimensional en el *DataMart*.

The screenshot shows the configuration of a 'Table output' step in Talend Studio. The 'Step name' is 'Table output (DIM_Measurement)'. The 'Connection' is 'cn_dw'. The 'Target schema' is 'dbo' and the 'Target table' is 'DIM_Measurement'. The 'Commit size' is set to 1000. There are checkboxes for 'Truncate table', 'Ignore insert errors', and 'Specify database fields' (which is checked). Below this, there are tabs for 'Main options' and 'Database fields'. Under 'Database fields', there is a table titled 'Fields to insert' with 5 rows:

#	Table field	Stream field
1	measureme...	measurement...
2	measureme...	measurement...
3	fk_sdg	fk_sdg
4	Unit	Unit
5	pk_measure...	pk

Buttons for 'Get fields' and 'Enter field mapping' are also visible.

Para comprobar que se funciona de manera correcta, se mostrara las estadísticas:

The screenshot shows the execution results of a transformation. The flow consists of three steps: 'Table input' (green checkmark), 'Add sequence' (green checkmark), and 'Table output (DIM_Measurement)' (green checkmark). Below the flow, the 'Execution Results' table provides detailed metrics for each step:

Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1 Table input	0	0	44	44	0	0	0	0	Finished	0.1s	530	-
2 Add sequence	0	44	44	0	0	0	0	0	Finished	0.1s	423	-
3 Table output (DIM_Measurement)	0	44	44	0	44	0	0	0	Finished	0.2s	240	-

2.4.5.5. TR_DIM_Product

La transformación completa es la siguiente:



Esta transformación consta de los siguientes pasos:

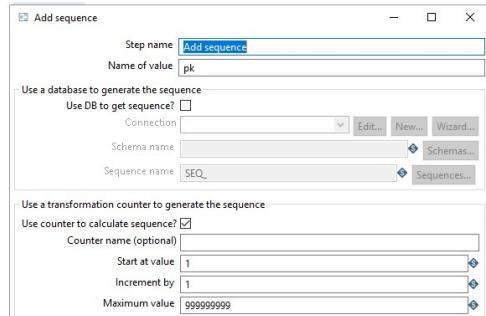
- Obtención de los valores:** Mediante una sentencia «SELECT» de SQL de los productos.

The screenshot shows the configuration of a 'Table input' step for 'STG_Product'. The 'Step name' is 'Table input (STG_Product)' and the 'Connection' is 'cn_stage'. The 'SQL' section contains the following query:

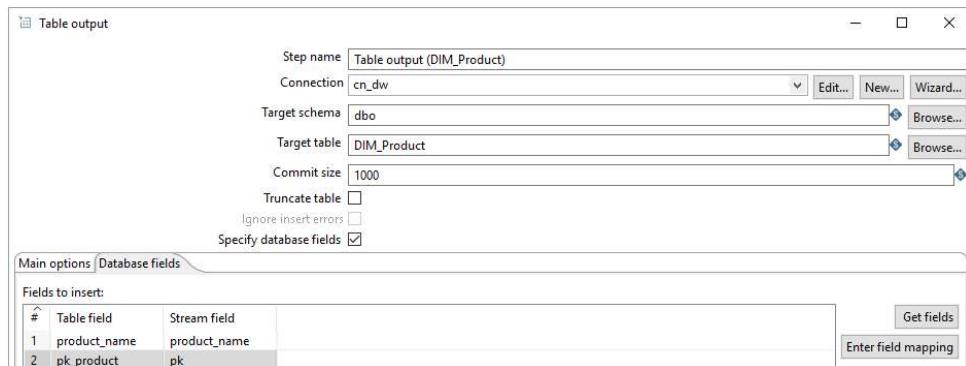
```

SELECT DISTINCT
    product AS product_name
FROM [dbo].[STG_EnergyBalance]
UNION
SELECT
    'NA' AS product_name
  
```

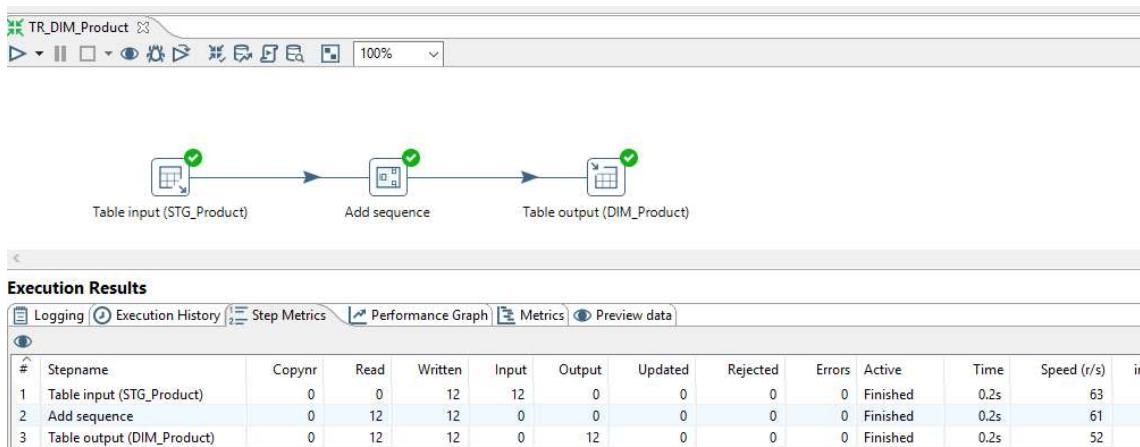
- **Creación de una secuencia:** que hará las funciones de clave primaria de incremento automático.



- **Carga los datos:** a la tabla de dimensión DIM_Product del modelo dimensional en el DataMart.



Para comprobar que se funciona de manera correcta, se mostrara las estadísticas:



2.4.6. Bloque TR_FACT

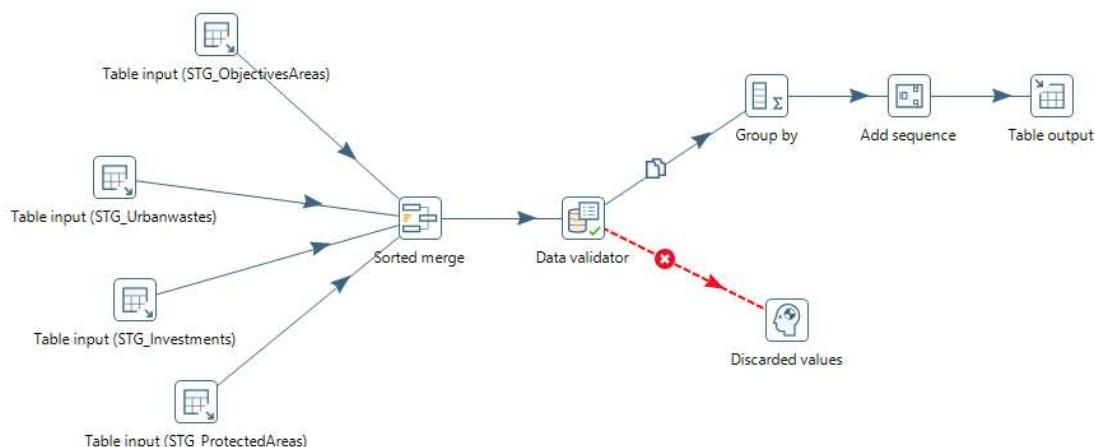
Este bloque contiene las transformaciones para la carga inicial de las tablas de hecho al almacén desde las tablas intermedias «STG_» del staging area.

La parte principal en la carga de las tablas de hechos es la búsqueda de los valores de las claves foráneas en las tablas de dimensiones cargadas anteriormente.

2.4.6.1. TR_FACT_EnvironmentalMeasurements

En esta transformación se integrarán diversas tablas en una única tabla de hechos del almacén de datos.

La transformación completa es la siguiente:



Esta transformación consta de los siguientes pasos:

- **Obtención de los valores:** Mediante sentencias «SELECT» de SQL de distintas tablas:
 - **STG_ObjectivesAreas:** para los campos que no existen valores en esa tabla, se ha asignado el valor nulo por defecto.

A screenshot of the Table input step configuration window. The step name is 'Table input (STG_ObjectivesAreas)'. The connection is set to 'cn_dw'. The SQL query is displayed in the text area:

```
Step name: Table input (STG_ObjectivesAreas)
Connection: cn_dw
SQL
SELECT DISTINCT
    (SELECT dl.pk_measurement WHERE dl.fk_sdg = t1.[ODS principal]) as fk_measurement,
    19700101 as pk_fk_date,
    52 as fk_typearea, --Valor Nulo
    2 as fk_activitysector, --Valor Nulo
    3 as fk_typeequipinstall, --Valor Nulo
    0 as value
FROM
    [dbo].STG_ObjectivesAreas t1,
    [dbo].DIM_Measurement dl
```

- STG_Urbanwastes: para los campos que no existen valores en esa tabla, se ha asignado el valor nulo por defecto.

Table input

Step name: Table input (STG_Urbanwastes)

Connection: cn_dw

Get SQL select statement...

```
SQL
SELECT DISTINCT
    (SELECT d4.pk_measurement WHERE d4.measurement_code = t1.Var) as fk_measurement,
    (SELECT DISTINCT pk_date FROM DIM_Date WHERE date_year = t1.Year and date_day = 1 and date_month = 1) as pk_fk_date,
    (SELECT d1.pk_region WHERE d1.country_code3 = t1.Cou) as fk_region,
    2 as fk_activitysector, --Valor Nulo
    3 as fk_typeequipinstall,--Valor Nulo
    0 as value
FROM
    [dbo].STG_Urbanwastes t1,
    [dbo].DIM_Region d1,
    [dbo].DIM_Measurement d4
```

- STG_Investments: para los campos que no existen valores en esa tabla, se ha asignado el valor nulo por defecto.

Table input

Step name: Table input (STG_Investments)

Connection: cn_dw

Get SQL select statement...

```
SQL
SELECT DISTINCT
    (SELECT d1.pk_measurement WHERE d1.measurement_code = t1.areaprot) as fk_measurement,
    (SELECT DISTINCT pk_date FROM DIM_Date WHERE date_year = t1.Year and date_day = 1 and date_month = 1) as pk_fk_date,
    (SELECT d2.pk_region WHERE d2.country_code2 = t1.geoTime) as fk_region,
    2 as fk_activitysector, --Valor Nulo
    3 as fk_typeequipinstall,--Valor Nulo
    0 as value
FROM
    [dbo].STG_ProtectedAreas t1,
    [dbo].DIM_Region d2,
    [dbo].DIM_Measurement d1
```

- STG_ProtectedAreas: para los campos que no existen valores en esa tabla, se ha asignado el valor nulo por defecto.

Table input

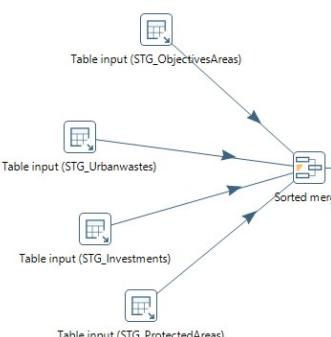
Step name: Table input (STG_ProtectedAreas)

Connection: cn_stage

Get SQL select statement...

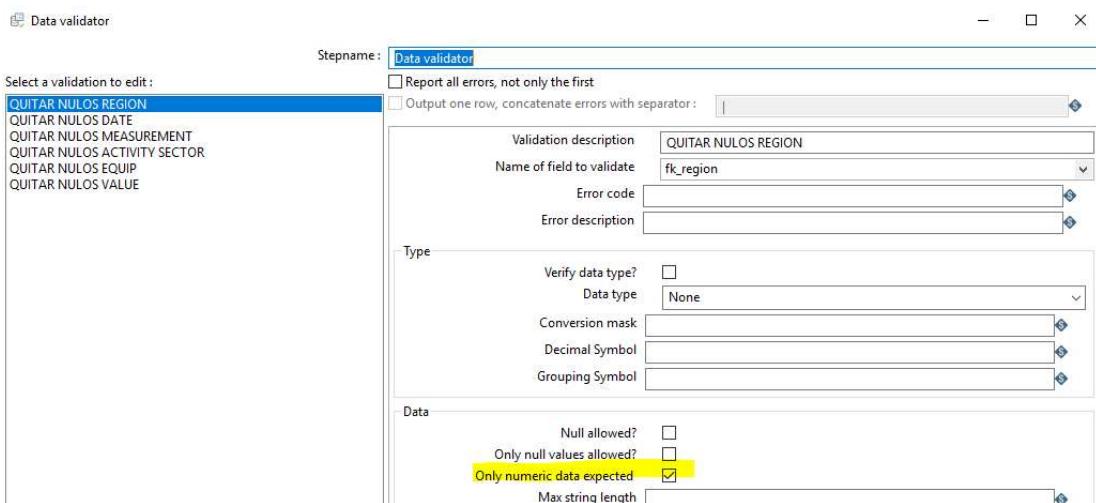
```
SQL
SELECT DISTINCT
    (SELECT d1.pk_measurement WHERE d1.measurement_name = t1.areaprot) as fk_measurement,
    (SELECT DISTINCT pk_date FROM DIM_Date WHERE date_year = t1.Year and date_day = 1 and date_month = 1) as pk_fk_date,
    (SELECT d2.pk_region WHERE d2.country_code2 = t1.geoTime) as fk_region,
    2 as fk_activitysector, --Valor Nulo
    3 as fk_typeequipinstall,--Valor Nulo
    0 as value
FROM
    [dbo].STG_ProtectedAreas t1,
    [dbo].DIM_Region d2,
    [dbo].DIM_Measurement d1
```

- Unión ordenada: una vez obtenido los valores, a través de esta funcionalidad unimos los valores de todas las tablas:



- **Validador de datos:** haciendo un análisis exploratorio de los datos obtenidos, veo que hay muchos campos nulos, por lo que he tomado la decisión de quedarme solamente con los datos completos, ya que no hay ninguna regla que especifique que se debe hacer en estos casos, si llenar los campos o descartarlos.

Las validaciones son por todos los campos y para todos se sigue la misma forma comprobar de que el campo tenga un valor numérico:



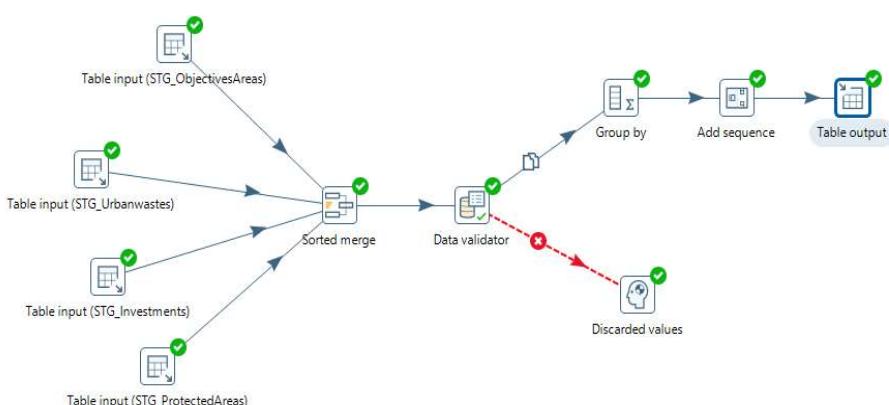
Para ver los valores descartados se ha añadido un dummy.

- **Agrupar campos:** la otra opción del validador es la funcionalidad que nos permite agrupar los campos, esto se hace para evitar tener valores repetidos, para ello he decidido agruparlos por todas las claves foráneas y sumar los valores de cada una de ellas:

The screenshot shows the 'Group by' configuration window. The 'Step name' is 'Group by'. Under 'The fields that make up the group:', there is a table with columns '#', 'Group field', and 'Get Fields'. The rows are: 1 fk_measurement, 2 pk_fk_date, 3 fk_region, 4 fk_activitysector, and 5 fk_typeequipinstall. Under 'Aggregates:', there is a table with columns '#', 'Name', 'Subject', 'Type', 'Value', and 'Get lookup fields'. The row is: 1 value value Sum.

- **Creación de una secuencia:** que hará las funciones de clave primaria de incremento automático.
- **Carga los datos:** a la tabla de hechos FACT_EnvironmentalMeasurements:

Para comprobar que se funciona de manera correcta, se mostraran las estadísticas:



Execution Results													
	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Table input (STG_ObjectivesAreas)	0	0	45	45	0	0	0	0	Finished	0.0s	938	-
2	Table input (STG_ProtectedAreas)	0	0	396	396	0	0	0	0	Finished	38.0s	10	-
3	Table input (STG_Investments)	0	0	1584	1584	0	0	0	0	Finished	38.6s	41	-
4	Table input (STG_Urbanwastes)	0	0	25097	25097	0	0	0	0	Finished	2mn 45s	151	-
5	Sorted merge	0	27122	27122	0	0	0	0	0	Finished	2mn 46s	163	-
6	Data validator	0	27122	24029	0	0	0	3093	0	Finished	2mn 46s	163	-
7	Group by	0	24029	24029	0	0	0	0	0	Finished	2mn 46s	145	-
8	Add sequence	0	24029	24029	0	0	0	0	0	Finished	2mn 46s	144	-
9	Discarded values	0	3093	3093	0	0	0	0	0	Finished	2mn 46s	19	-
10	Table output	0	24029	24029	0	24029	0	0	0	Finished	2mn 46s	144	-

Además, se van a mostrar el resultado de la carga de datos en la BBDD:

```

-SELECT TOP (1000) [pk_id]
    ,[fk_date]
    ,[fk_region]
    ,[fk_activitysector]
    ,[fk_typeequipinstall]
    ,[fk_measurement]
    ,[value]
FROM [SOURCE_edumoraglez].[dbo].[FACT_EnvironmentalMeasurements]

```

pk_id	fk_date	fk_region	fk_activitysector	fk_typeequipinstall	fk_measurement	value
1	19900101	15	2	3	6	0
2	19900101	17	2	3	6	0
3	19900101	21	2	3	6	0
4	19900101	37	2	3	6	0
5	19900101	66	2	3	6	0

2.4.6.2. TR_FACT_EnergyBalances

En esta transformación se integrarán diversas tablas en una única tabla de hechos del almacén de datos.

La transformación completa es la siguiente:



Esta transformación consta de los siguientes pasos:

- Obtención de los valores:** Mediante sentencias «SELECT» de SQL de distintas tablas, al tratar solamente de una tabla IN, la consulta queda mas sencilla:

Table input

Step name: Table input (STG_EnergyBalances)

Connection: cn_stage

SQL

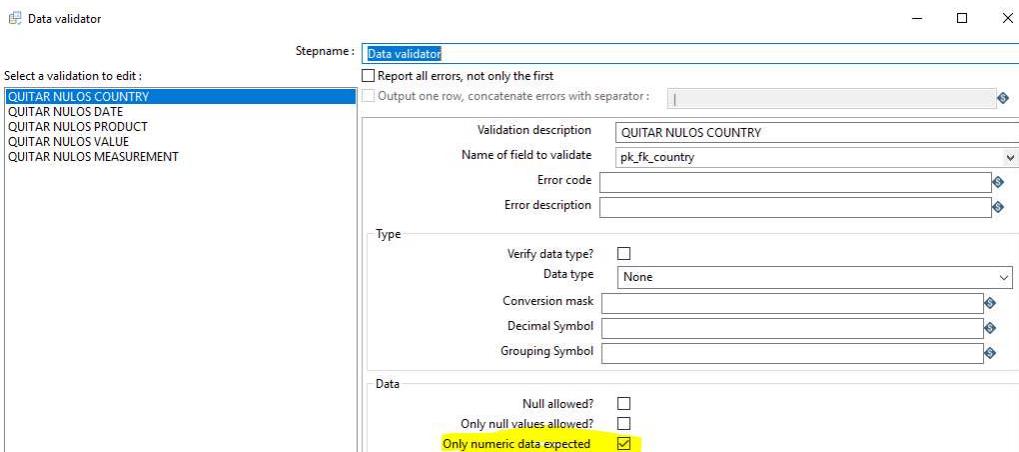
```

SELECT
    (SELECT pk_product FROM DIM_Product WHERE product_name = t1.product) as pk_fk_product,
    (SELECT DISTINCT pk_date FROM DIM_Date WHERE date_year = t1.year AND date_day = 1 AND date_month = 1) as pk_fk_date,
    (SELECT pk_country FROM DIM_Country WHERE country_name_sp = t1.country ) as pk_fk_country,
    t1.value,
    1 as pk_fk_measurement
FROM [dbo].STG_EnergyBalance t1

```

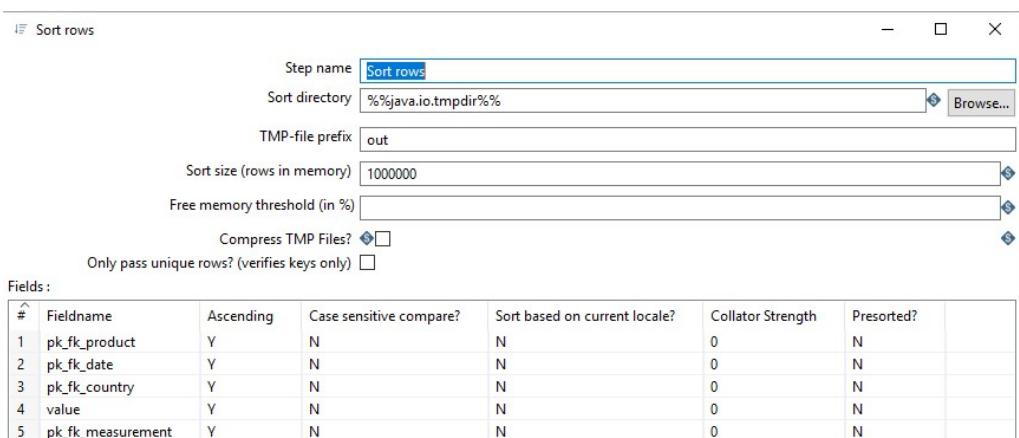
- **Validador de datos:** haciendo un análisis exploratorio de los datos obtenidos, veo que hay muchos campos nulos, por lo que he tomado la decisión de quedarme solamente con los datos completos, ya que no hay ninguna regla que especifique que se debe hacer en estos casos, si llenar los campos o descartarlos.

Las validaciones son por todos los campos y para todos se sigue la misma forma comprobar de que el campo tenga un valor numérico:

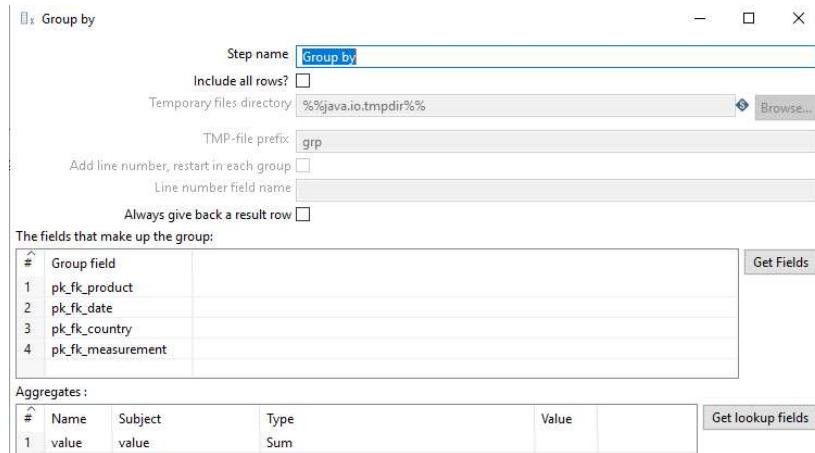


Para ver los valores descartados se ha añadido un dummy como en el caso anterior.

- **Ordenar los datos:** de manera ascendente todos los campos:

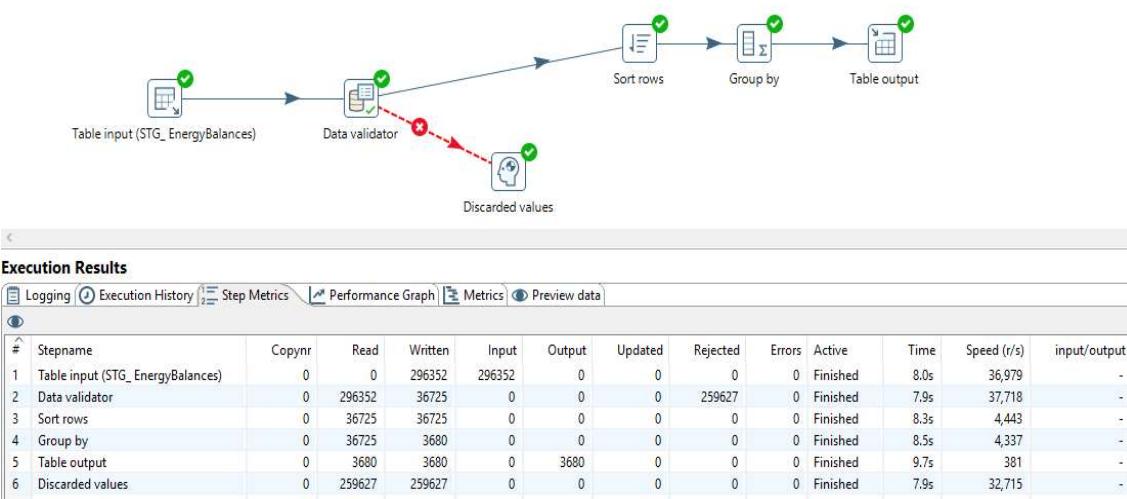


- **Agrupar campos:** la otra opción del validador es la funcionalidad que nos permite agrupar los campos, esto se hace para evitar tener valores repetidos, para ello he decidido agruparlos por todas las claves foráneas y sumar los valores de cada una de ellas:

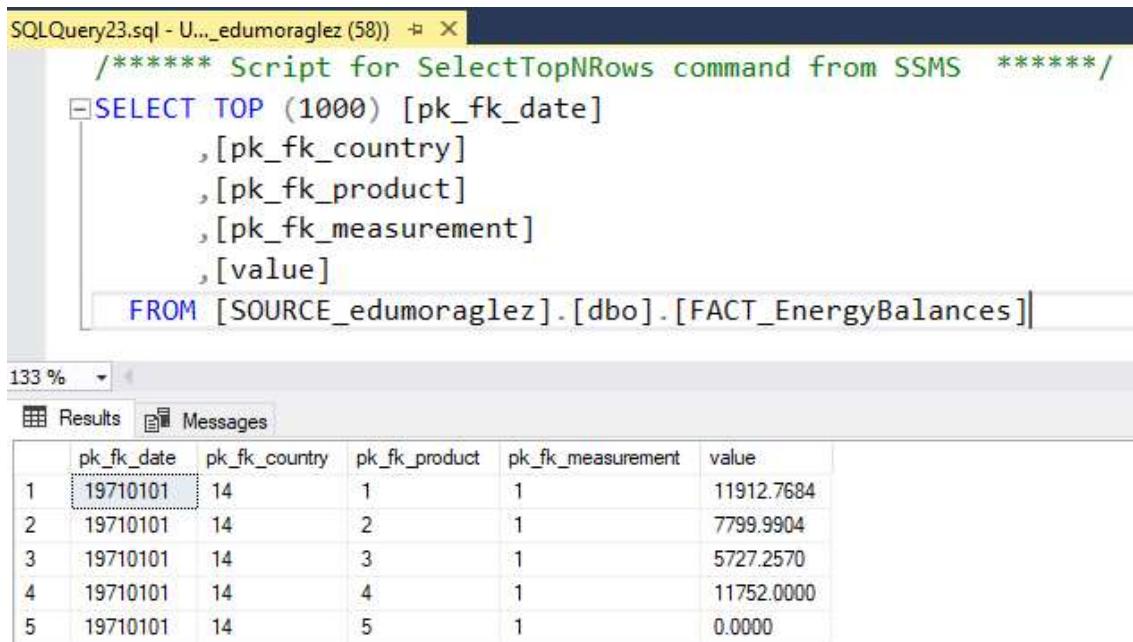


- **Carga los datos:** a la tabla de hechos:

Para comprobar que se funciona de manera correcta, se mostraran las estadísticas:



Además, se van a mostrar el resultado de la carga de datos en la BBDD:



The screenshot shows a SQL query in the SSMS interface. The query is:

```
SQLQuery23.sql - U..._edumoraglez (58)  X
/*
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [pk_fk_date]
    ,[pk_fk_country]
    ,[pk_fk_product]
    ,[pk_fk_measurement]
    ,[value]
FROM [SOURCE_edumoraglez].[dbo].[FACT_EnergyBalances]
```

The results grid displays five rows of data:

	pk_fk_date	pk_fk_country	pk_fk_product	pk_fk_measurement	value
1	19710101	14	1	1	11912.7684
2	19710101	14	2	1	7799.9904
3	19710101	14	3	1	5727.2570
4	19710101	14	4	1	11752.0000
5	19710101	14	5	1	0.0000

3. Implementación de los trabajos con procesos ETL

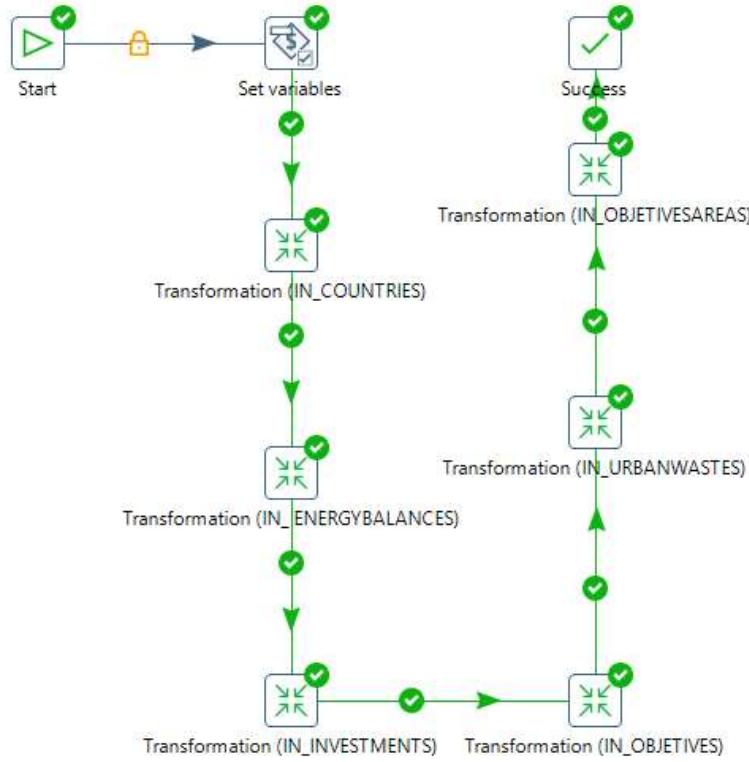
Habrá que tener en cuenta los siguientes bloques de procesos implementados:

- ✓ Bloque «IN_»: procesos de ETL de transformación y carga al área intermedia.
- ✓ Bloque «TR_DIM»: procesos de ETL de transformación y carga de dimensiones.
- ✓ Bloque «TR_FACT»: procesos de ETL de transformación y carga de hechos.

3.1. JOB_IN

El trabajo (job) «JOB_IN» procesa todas las transformaciones del bloque «IN_» para la carga de datos desde las fuentes de datos proporcionadas al área intermedia (staging area).

El diseño completo del trabajo (job) «JOB_IN» es el siguiente:



Los pasos incluidos en el trabajo «JOB_IN» son:

- Inicio del job.
- Configuración de las variables de entorno.
- Ejecución de las transformaciones «IN_» de carga del staging area.
- Finalización del job.

El resultado de la ejecución de la transformación completa es el siguiente:

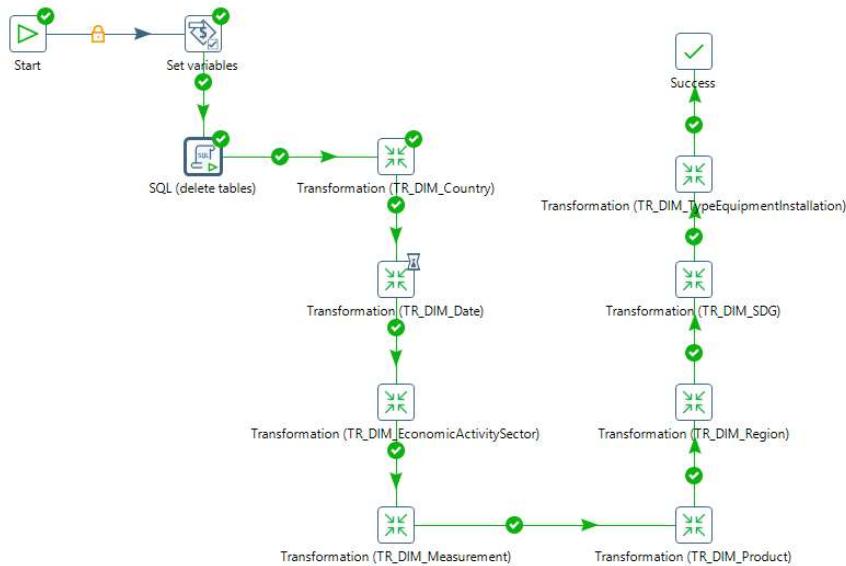
Execution Results						
	Logging	History	Job metrics	Metrics		
Job / Job Entry	Comment		Result	Reason	Filename	
Job: JOB_IN						
Start	Start of job execution		start			2021/12/09 17:27:04
Start	Start of job execution		start			2021/12/09 17:27:04
Set variables	Start of job execution		Success	Followed unconditional link		2021/12/09 17:27:04
Set variables	Job execution finished		Success			0 2021/12/09 17:27:04
Transformation (IN_COUNTRI)	Start of job execution			Followed link after success		2021/12/09 17:27:04
Transformation (IN_COUNTRI)	Job execution finished		Success			2 2021/12/09 17:27:05
Transformation (IN_ENERGYVB)	Start of job execution			Followed link after success		2021/12/09 17:27:05
Transformation (IN_ENERGYVB)	Job execution finished		Success			3 2021/12/09 17:27:19
Transformation (IN_INVESTMI)	Start of job execution			Followed link after success		2021/12/09 17:27:19
Transformation (IN_INVESTMI)	Job execution finished		Success			4 2021/12/09 17:27:20
Transformation (IN_OBJETIVE!)	Start of job execution			Followed link after success		2021/12/09 17:27:20
Transformation (IN_OBJETIVE!)	Job execution finished		Success			5 2021/12/09 17:27:20
Transformation (IN_URBANW)	Start of job execution			Followed link after success		2021/12/09 17:27:20
Transformation (IN_URBANW)	Job execution finished		Success			6 2021/12/09 17:27:27
Transformation (IN_OBJETIVE!)	Start of job execution			Followed link after success		2021/12/09 17:27:27
Transformation (IN_OBJETIVE!)	Job execution finished		Success			7 2021/12/09 17:27:28
Success	Start of job execution			Followed link after success		2021/12/09 17:27:28
Success	Job execution finished		Success			7 2021/12/09 17:27:28
Job: JOB_IN	Job execution finished		Success	finished		7 2021/12/09 17:27:28

Se observa el procesamiento con éxito de todos los pasos del «JOB_IN» correspondientes a la ejecución de todas las transformaciones que están incluidas en el trabajo.

3.2. JOB_TR_DIMS

El trabajo (job) «JOB_TR_DIMS» procesa todas las transformaciones del bloque «TR_DIMS» para la carga de datos, desde las tablas intermedias hasta las tablas de dimensiones del almacén.

El diseño completo del trabajo (job) «JOB_TR_DIMS» es el siguiente:



Los pasos incluidos en el trabajo «JOB_TR_DIMS» son:

- Inicio del job.
- Carga de variables de entorno (path de orígenes de datos y conexiones).
- Borrado de todas las tablas. Esto permite la recarga inicial en caso de ser necesario. Aunque es importante respetar el orden de borrado, según las relaciones definidas entre tablas, en este caso en particular, dado que no hay relaciones entre tablas de dimensión, no influirá el orden.
- Ejecución secuencial de todas las transformaciones «TR_DIM» (extracción, transformación y carga de dimensiones).
- Finalización del job.

El resultado de la ejecución de la transformación completa es el siguiente:

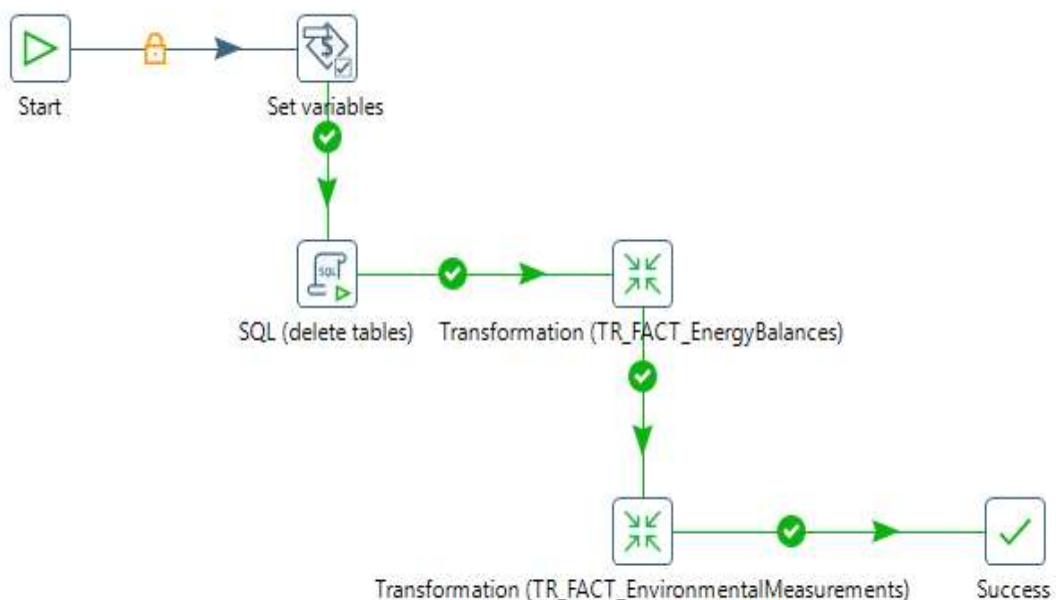
Execution Results						
Job / Job Entry	Comment	Result	Reason	Filename	Nr	Log date
Job: JOB_TR_DIMS						
Start	Start of job execution	start	start		2021/12/09 17:51:03	
Set variables	Job execution finished	Success	Followed unconditional link		0	2021/12/09 17:51:03
SQL (delete tables)	Start of job execution	Success	Followed link after success		0	2021/12/09 17:51:03
SQL (delete tables)	Job execution finished	Success	Followed link after success		0	2021/12/09 17:51:03
Transformation (TR_DIM_Cou)	Start of job execution	Success	Followed link after success		3	2021/12/09 17:51:03
Transformation (TR_DIM_Cou)	Job execution finished	Success	Followed link after success		3	2021/12/09 17:51:03
Transformation (TR_DIM_Date)	Start of job execution	Success	Followed link after success		4	2021/12/09 17:51:03
Transformation (TR_DIM_Date)	Job execution finished	Success	Followed link after success		4	2021/12/09 17:51:03
Transformation (TR_DIM_Ecor)	Start of job execution	Success	Followed link after success		5	2021/12/09 17:51:03
Transformation (TR_DIM_Ecor)	Job execution finished	Success	Followed link after success		5	2021/12/09 17:51:03
Transformation (TR_DIM_Mea)	Start of job execution	Success	Followed link after success		6	2021/12/09 17:51:03
Transformation (TR_DIM_Mea)	Job execution finished	Success	Followed link after success		6	2021/12/09 17:51:03
Transformation (TR_DIM_Proc)	Start of job execution	Success	Followed link after success		7	2021/12/09 17:51:03
Transformation (TR_DIM_Proc)	Job execution finished	Success	Followed link after success		7	2021/12/09 17:51:03
Transformation (TR_DIM_Regi)	Start of job execution	Success	Followed link after success		8	2021/12/09 17:51:03
Transformation (TR_DIM_Regi)	Job execution finished	Success	Followed link after success		8	2021/12/09 17:51:03
Transformation (TR_DIM_Mea)	Start of job execution	Success	Followed link after success		9	2021/12/09 17:51:03
Transformation (TR_DIM_Mea)	Job execution finished	Success	Followed link after success		9	2021/12/09 17:51:03
Transformation (TR_DIM_Type)	Start of job execution	Success	Followed link after success		10	2021/12/09 17:51:03
Transformation (TR_DIM_Type)	Job execution finished	Success	Followed link after success		10	2021/12/09 17:51:03
Success	Start of job execution	Success	Followed link after success		10	2021/12/09 17:51:03
Success	Job execution finished	Success	Followed link after success		10	2021/12/09 17:51:03
Job: JOB_TR_DIMS	Job execution finished	Success	finished		10	2021/12/09 17:51:03

Se observa el procesamiento con éxito de todos los pasos del «JOB_TR_DIMS», correspondientes a la ejecución de todas las transformaciones que están incluidas en el trabajo.

3.3. JOB_TR_FACTS

El trabajo (job) «JOB_TR_FACTS» procesa todas las transformaciones del bloque «TR_FACT» para la carga de datos desde las tablas intermedias a las tablas de hechos del almacén.

El diseño completo del trabajo (job) «JOB_TR_FACTS» es el siguiente:



Los pasos incluidos en el trabajo «JOB_TR_FACTS» son:

- Inicio del job.
- Carga de variables de entorno.
- Eliminación de tablas.
- Ejecución de las transformaciones «TR_FACT».
- Finalización del job.

El resultado de la ejecución de la transformación completa es el siguiente:

Execution Results						
Job / Job Entry	Comment	Result	Reason	Filename	Nr	Log date
Job: JOB_TR_FACT						
Start	Start of job execution	start	start			2021/12/10 09:58:29
Start	Job execution finished	Success	Followed unconditional link		0	2021/12/10 09:58:29
Set variables	Start of job execution	Success	Followed unconditional link		0	2021/12/10 09:58:29
Set variables	Job execution finished	Success	Followed unconditional link		0	2021/12/10 09:58:29
SQL (delete tables)	Start of job execution	Success	Followed link after success		0	2021/12/10 09:58:29
SQL (delete tables)	Job execution finished	Success	Followed link after success		0	2021/12/10 09:58:29
Transformation (TR_FACT_EnergyBalances)	Start of job execution	Success	Followed link after success		3	2021/12/10 09:58:29
Transformation (TR_FACT_EnergyBalances)	Job execution finished	Success	Followed link after success		3	2021/12/10 09:58:34
Transformation (TR_FACT_EnvironmentalMeasures)	Start of job execution	Success	Followed link after success		4	2021/12/10 09:58:34
Transformation (TR_FACT_EnvironmentalMeasures)	Job execution finished	Success	Followed link after success		4	2021/12/10 10:01:00
Success	Start of job execution	Success	Followed link after success		4	2021/12/10 10:01:00
Success	Job execution finished	Success	Followed link after success		4	2021/12/10 10:01:00
Job: JOB_TR_FACT	Job execution finished	Success	finished		4	2021/12/10 10:01:00

3.4. JOB_CARGA_DW

El trabajo «JOB_CARGA_DW» orquesta todos los trabajos anteriores en un único proceso.

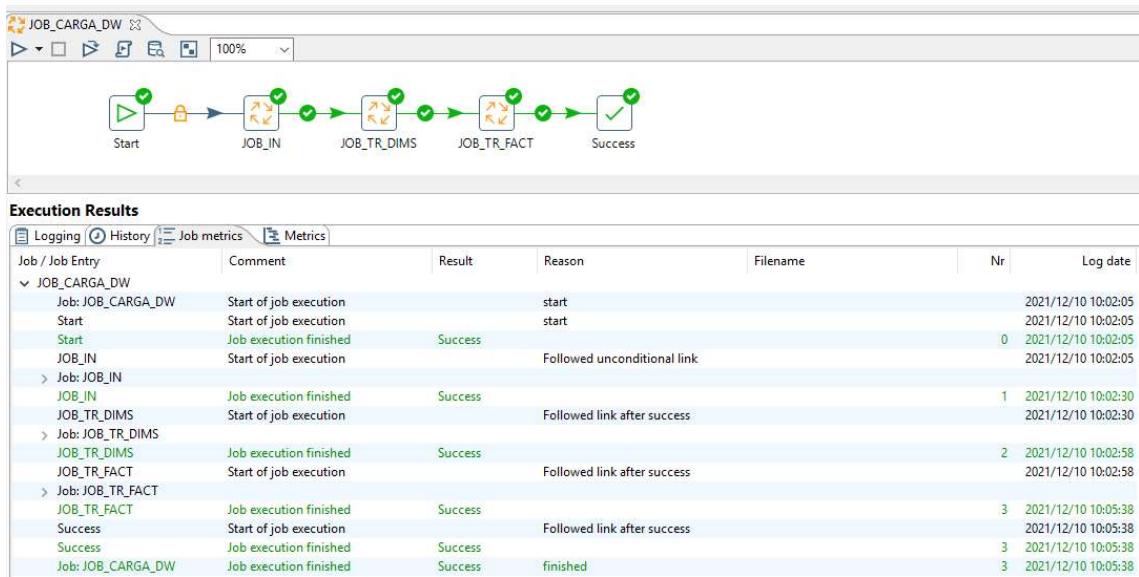
El diseño completo del trabajo «JOB_CARGA_DW» es el siguiente:



Los pasos incluidos en el trabajo «JOB_CARGA_DW» son:

- Inicio del job.
- Ejecución orquestada de los jobs de carga de todas las transformaciones.
- Finalización del job.

El resultado de la ejecución de la transformación completa es el siguiente:



Se observa el procesamiento con éxito de todos los pasos del «JOB_CARGA_DW», correspondientes a la ejecución de todas las transformaciones que están incluidas en el trabajo.

El tiempo total de la carga inicial del data Warehouse es de aproximadamente de 3 minutos y medio:

Job: JOB_CARGA_DW	Start of job execution	start		2021/12/10 10:02:05
Start	Start of job execution	start		2021/12/10 10:02:05
Start	Job execution finished	Success		0 2021/12/10 10:02:05
JOB_IN	Start of job execution		Followed unconditional link	2021/12/10 10:02:05
> Job: JOB_IN				
JOB_IN	Job execution finished	Success		1 2021/12/10 10:02:30
JOB_TR_DIMS	Start of job execution		Followed link after success	2021/12/10 10:02:30
> Job: JOB_TR_DIMS				
JOB_TR_DIMS	Job execution finished	Success		2 2021/12/10 10:02:58
JOB_TR_DIMS	Start of job execution		Followed link after success	2021/12/10 10:02:58
> Job: JOB_TR_FACT				
JOB_TR_FACT	Job execution finished	Success		3 2021/12/10 10:05:38
Success	Start of job execution		Followed link after success	2021/12/10 10:05:38
Success	Job execution finished	Success		3 2021/12/10 10:05:38
Job: JOB_CARGA_DW	Job execution finished	Success	finished	3 2021/12/10 10:05:38