ugr | Universidad de Granada

Sistemas Multidimensionales. **Práctica 7:**

Proyecto Final: PDI/Mondrian
Mortalidad en EEUU

Alberto Jesús Durán López

Ingeniería Informática y Matemáticas

2019/2020

INDICE:

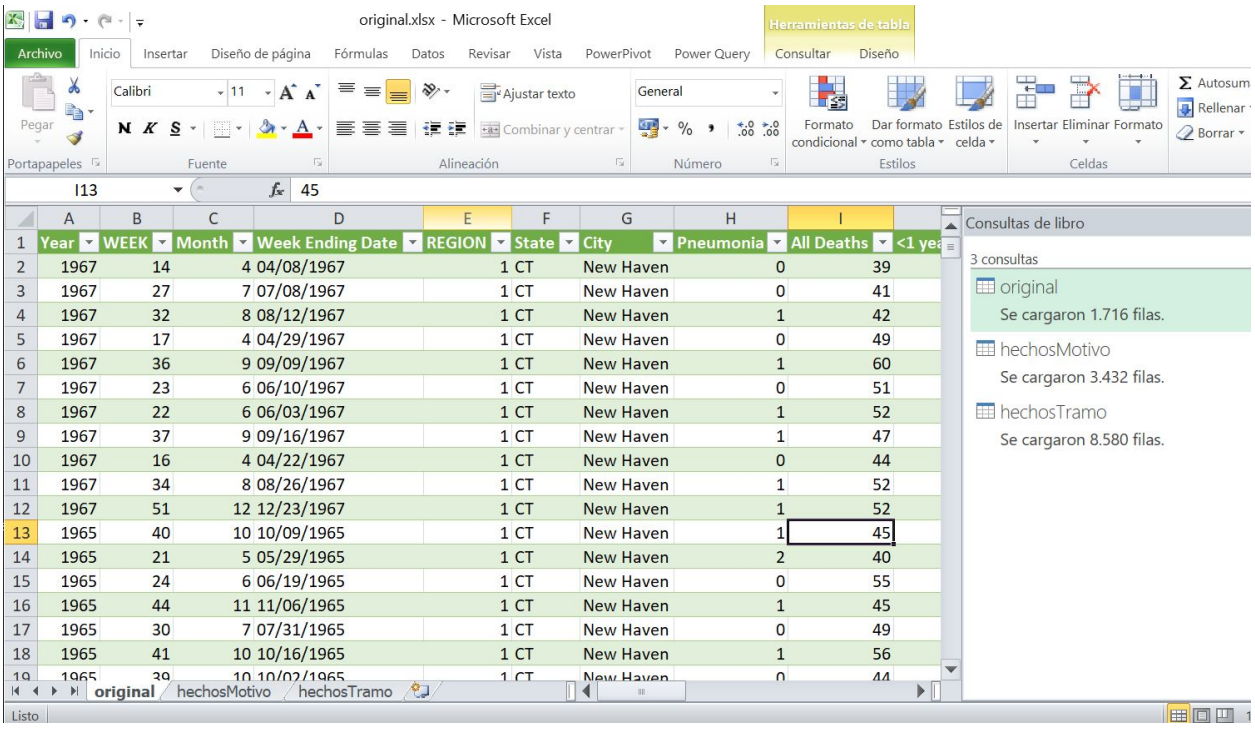
1. Introducción

Nuestro dataset tenía una estructura que podía verse de diferentes puntos de vista. De hecho, el primer modelo conceptual pensado era menos complejo que el implementado finalmente pero más limitado a la hora de hacer consultas. Por ello, se han implementado dos cubos:

1. **Mortalidad_tramo**: Sus dimensiones son Cuándo, Dónde y Tramo. Recoge la información de las muertes pudiendo hacer consultas sobre el tramo de edad establecido (<1año, 1-24 años, 25-44 años, 45-64 años, +65 años)
2. **Mortalidad_motivo**: Sus dimensiones son Cuándo, Dónde y Motivo. Recoge la información de las muertes pudiendo hacer consultas sobre el motivo de la muerte (neumonía u otros)

2. Preprocesado de datos con Excel (Power Query):

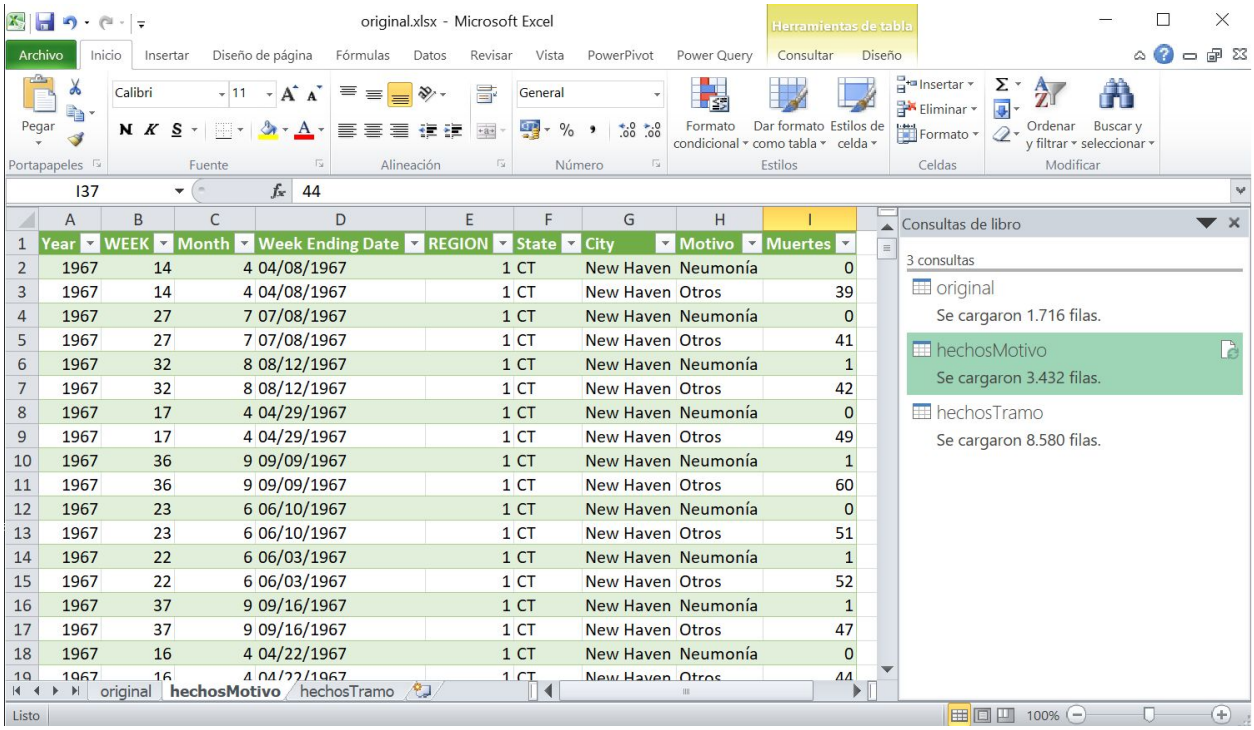
Primeramente se han reducido el número de instancias. Se recogía información sobre 9 regiones de EEUU, a su vez de todos sus estados y de las ciudades de cada uno. Por otro lado, la dimensión Cuándo contaba con datos semanales desde el año 1962 llegando casi al medio millón de instancias. Nuestra máquina virtual prácticamente no respondía al usar todos los datos por lo que nos hemos quedado con información sobre 3 años y 4-5 estados de 2 regiones diferentes.



	A	B	C	D	E	F	G	H	I
1	Year	WEEK	Month	Week Ending Date	REGION	State	City	Pneumonia	All Deaths
2	1967	14	4	04/08/1967	1	CT	New Haven	0	39
3	1967	27	7	07/08/1967	1	CT	New Haven	0	41
4	1967	32	8	08/12/1967	1	CT	New Haven	1	42
5	1967	17	4	04/29/1967	1	CT	New Haven	0	49
6	1967	36	9	09/09/1967	1	CT	New Haven	1	60
7	1967	23	6	06/10/1967	1	CT	New Haven	0	51
8	1967	22	6	06/03/1967	1	CT	New Haven	1	52
9	1967	37	9	09/16/1967	1	CT	New Haven	1	47
10	1967	16	4	04/22/1967	1	CT	New Haven	0	44
11	1967	34	8	08/26/1967	1	CT	New Haven	1	52
12	1967	51	12	12/23/1967	1	CT	New Haven	1	52
13	1965	40	10	10/09/1965	1	CT	New Haven	1	45
14	1965	21	5	05/29/1965	1	CT	New Haven	2	40
15	1965	24	6	06/19/1965	1	CT	New Haven	0	55
16	1965	44	11	11/06/1965	1	CT	New Haven	1	45
17	1965	30	7	07/31/1965	1	CT	New Haven	0	49
18	1965	41	10	10/16/1965	1	CT	New Haven	1	56
19	1965	39	10	10/02/1965	1	CT	New Haven	0	44

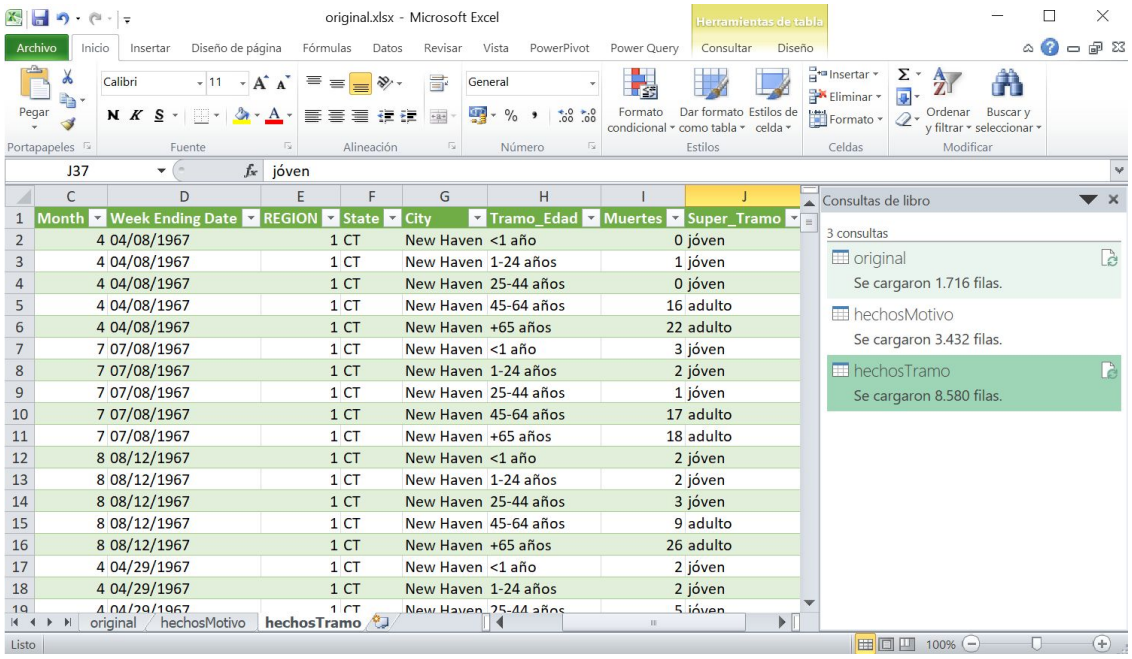
En el archivo *mortalidad-ETL-albduranlopez.xlsx* se han entregado las 3 hojas referentes al archivo original (con instancias reducidas) y a los dos cubos.

Para el cubo **Mortalidad_motivo** se ha anulado la dinamización de las columnas Neumonía y Otros.



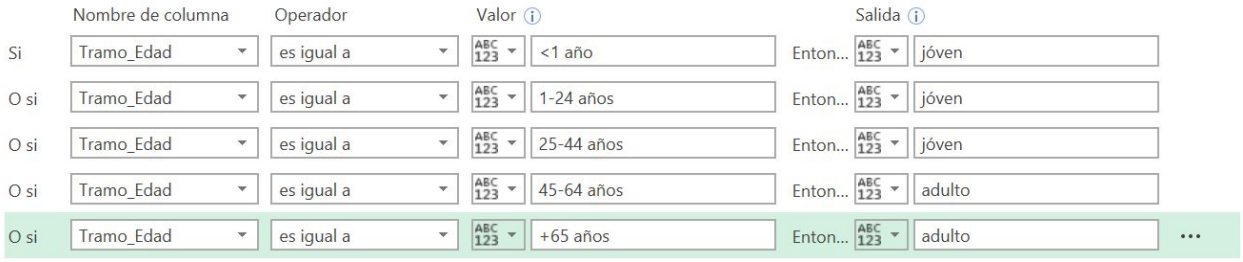
	A	B	C	D	E	F	G	H	I
1	Year	WEEK	Month	Week Ending Date	REGION	State	City	Motivo	Muertes
2	1967	14	4	04/08/1967	1	CT	New Haven	Neumonía	0
3	1967	14	4	04/08/1967	1	CT	New Haven	Otros	39
4	1967	27	7	07/08/1967	1	CT	New Haven	Neumonía	0
5	1967	27	7	07/08/1967	1	CT	New Haven	Otros	41
6	1967	32	8	08/12/1967	1	CT	New Haven	Neumonía	1
7	1967	32	8	08/12/1967	1	CT	New Haven	Otros	42
8	1967	17	4	04/29/1967	1	CT	New Haven	Neumonía	0
9	1967	17	4	04/29/1967	1	CT	New Haven	Otros	49
10	1967	36	9	09/09/1967	1	CT	New Haven	Neumonía	1
11	1967	36	9	09/09/1967	1	CT	New Haven	Otros	60
12	1967	23	6	06/10/1967	1	CT	New Haven	Neumonía	0
13	1967	23	6	06/10/1967	1	CT	New Haven	Otros	51
14	1967	22	6	06/03/1967	1	CT	New Haven	Neumonía	1
15	1967	22	6	06/03/1967	1	CT	New Haven	Otros	52
16	1967	37	9	09/16/1967	1	CT	New Haven	Neumonía	1
17	1967	37	9	09/16/1967	1	CT	New Haven	Otros	47
18	1967	16	4	04/22/1967	1	CT	New Haven	Neumonía	0
19	1967	16	4	04/22/1967	1	CT	New Haven	Otros	44

Para el cubo **Mortalidad_tramo** y, usando Power Query, se ha anulado la dinamización de las columnas que se correspondían con los diferentes tramos de edad.



	C	D	E	F	G	H	I	J
1	Month	Week Ending Date	REGION	State	City	Tramo_Edad	Muertes	Super_Tramo
2	4	04/08/1967	1	CT	New Haven	<1 año	0	jóven
3	4	04/08/1967	1	CT	New Haven	1-24 años	1	jóven
4	4	04/08/1967	1	CT	New Haven	25-44 años	0	jóven
5	4	04/08/1967	1	CT	New Haven	45-64 años	16	adulto
6	4	04/08/1967	1	CT	New Haven	+65 años	22	adulto
7	7	07/08/1967	1	CT	New Haven	<1 año	3	jóven
8	7	07/08/1967	1	CT	New Haven	1-24 años	2	jóven
9	7	07/08/1967	1	CT	New Haven	25-44 años	1	jóven
10	7	07/08/1967	1	CT	New Haven	45-64 años	17	adulto
11	7	07/08/1967	1	CT	New Haven	+65 años	18	adulto
12	8	08/12/1967	1	CT	New Haven	<1 año	2	jóven
13	8	08/12/1967	1	CT	New Haven	1-24 años	2	jóven
14	8	08/12/1967	1	CT	New Haven	25-44 años	3	jóven
15	8	08/12/1967	1	CT	New Haven	45-64 años	9	adulto
16	8	08/12/1967	1	CT	New Haven	+65 años	26	adulto
17	4	04/29/1967	1	CT	New Haven	<1 año	2	jóven
18	4	04/29/1967	1	CT	New Haven	1-24 años	2	jóven
19	4	04/29/1967	1	CT	New Haven	25-44 años	5	jóven

Por último, y para nutrir las dimensiones, se ha añadido el campo *Mes* en ambos cubos para la dimensión *Cuándo* y el campo *Super_tramo* que se ha definido de la siguiente forma:



	Nombre de columna	Operador	Valor		Salida	
Si	Tramo_Edad	es igual a	<1 año	Enton...	jóven	
O si	Tramo_Edad	es igual a	1-24 años	Enton...	jóven	
O si	Tramo_Edad	es igual a	25-44 años	Enton...	jóven	
O si	Tramo_Edad	es igual a	45-64 años	Enton...	adulto	
O si	Tramo_Edad	es igual a	+65 años	Enton...	adulto	•••

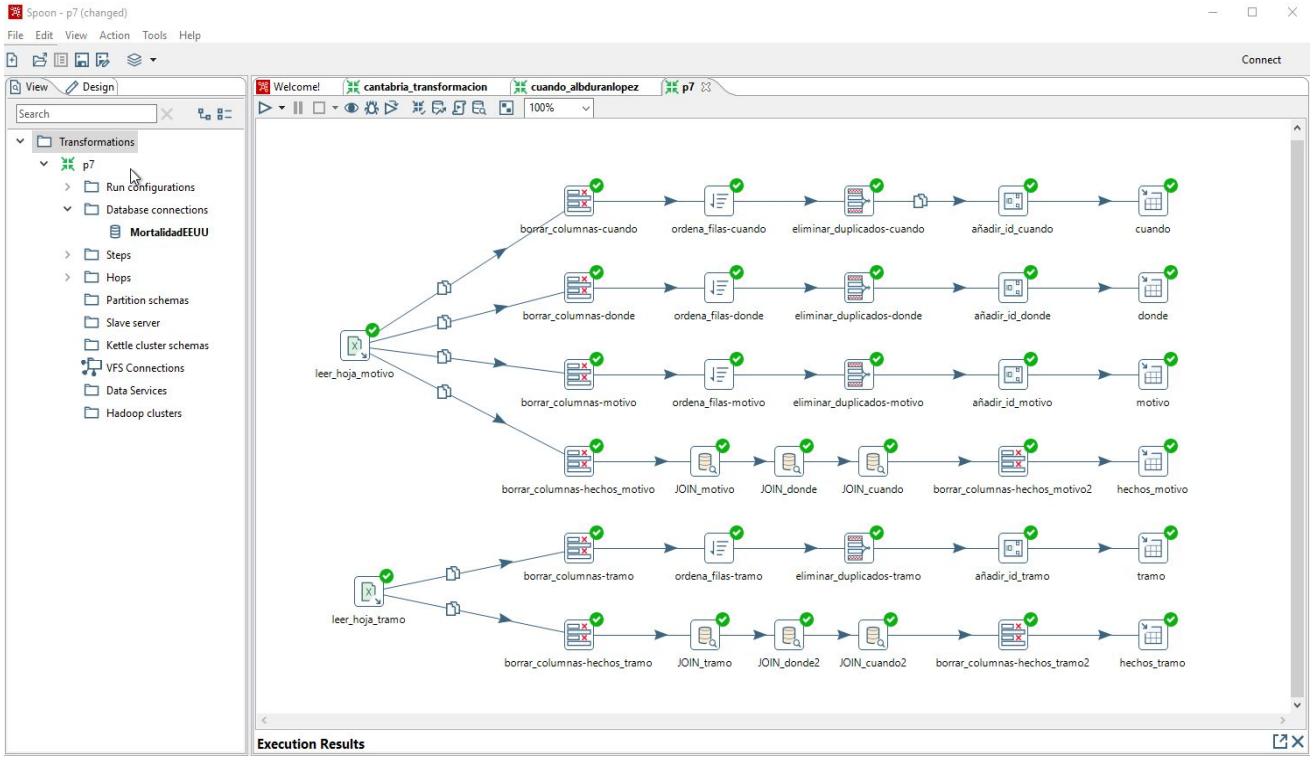
El resto de operaciones para obtener las diferentes dimensiones y los hechos de cada cubo se han realizado con *Spoon*.

3. PDI (Pentaho Data Integration)

Para realizar las distintas operaciones con *Spoon*, es imprescindible haber creado una base de datos para esta nueva práctica. La hemos denominado **MortalidadEEUU** y se ha creado en *pgAdmin*.

Cabe destacar que *Spoon* no almacena bien los enteros, de hecho, cuando intentamos cambiar un tipo a entero, se guarda como *double precision* por lo que aunque en las capturas de abajo se muestran *doubles*, se cambiarán posteriormente a *integer* en *pgAdmin*.
El criterio seguido ha sido **camel_case**, sin tilde, al igual que se ha realizado en las prácticas anteriores.

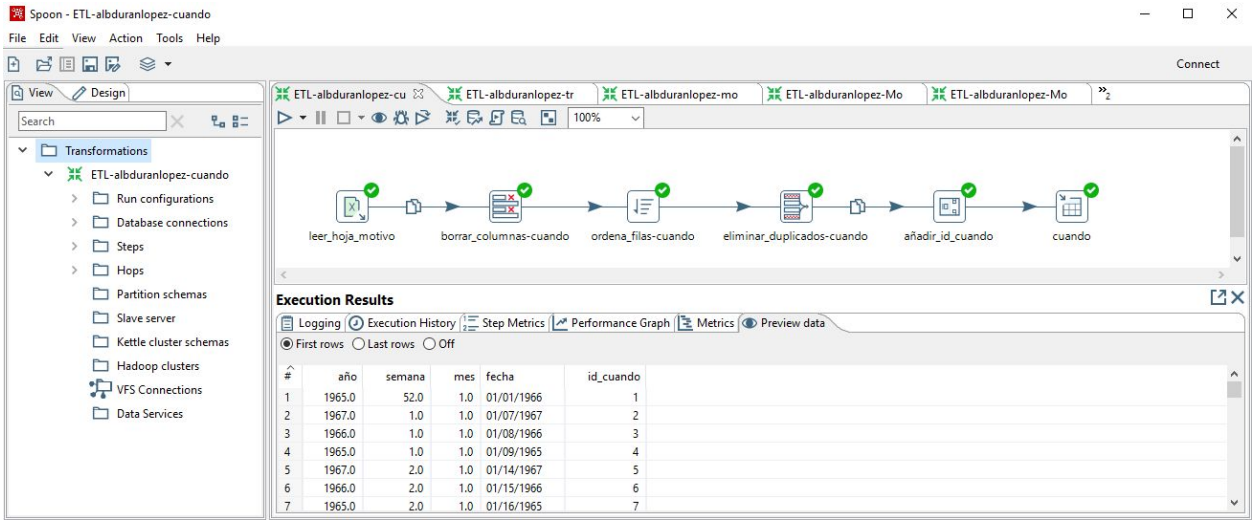
Explicaremos con detalle las operaciones realizadas para cada dimensión y para cada tabla de hechos, pero antes mostramos el resultado general de todas las operaciones.



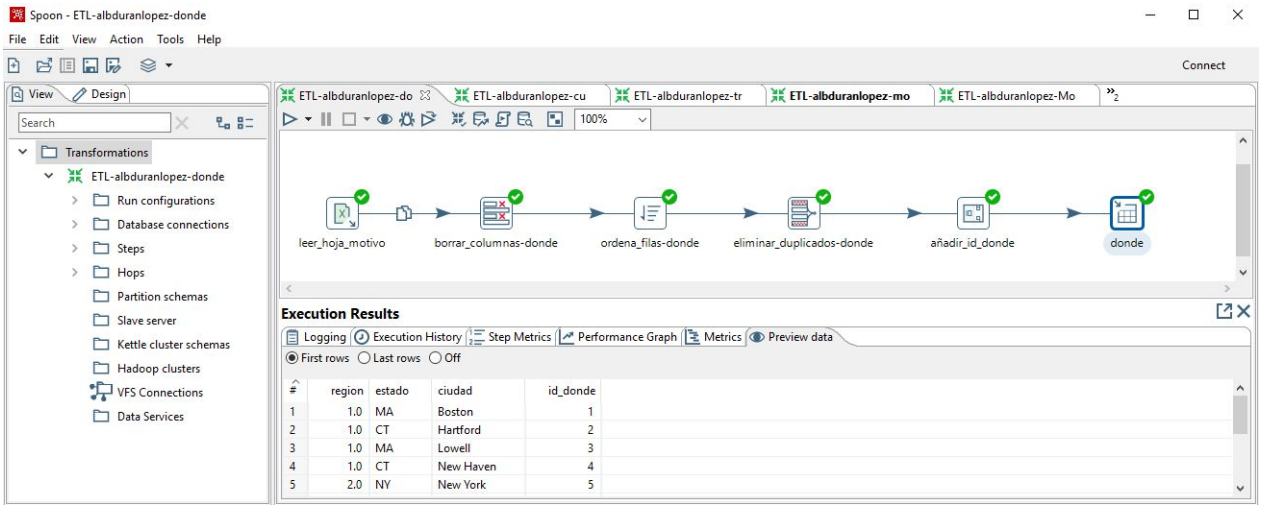
Los últimos nodos reciben el nombre dado a cada dimensión (en total 4) y a cada tabla de hechos (en total 2)

Sin embargo, realizamos la definición de cada dimensión y de cada tabla de hechos en un archivo independiente y, mediante un **job** denominado *ETL-job-mortalidad.kjb,* comprobaremos que se han ejecutado correctamente.

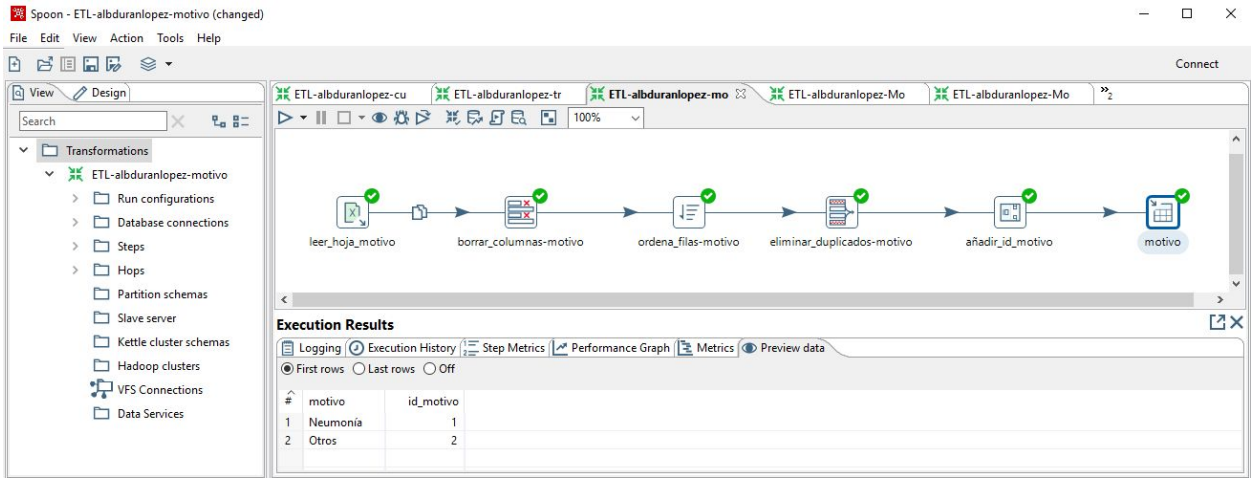
Cuándo: Nos tenemos que quedar con las columnas que hacen referencia al tiempo. Necesitamos borrar todas las demás, ordenar las filas, eliminar los duplicados y, por último, añadir la columna *id_cuando*.



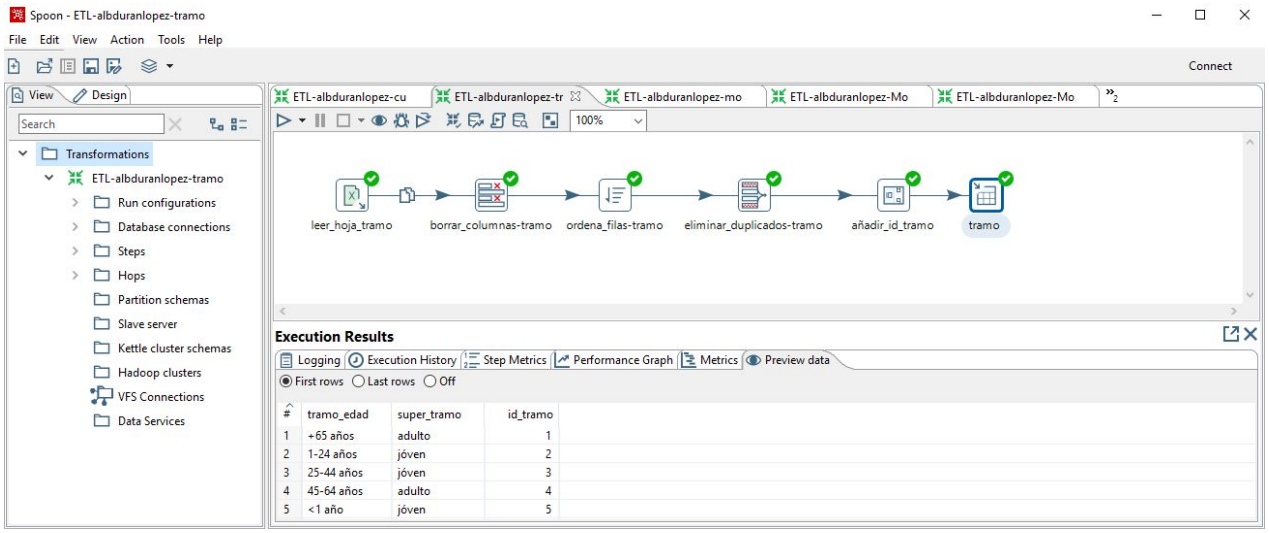
Dónde: Nos quedamos con los campos *ciudad*, *estado* y *región*. Eliminamos el resto y añadimos el identificador *id_donde* que será unívocamente determinado por cada ciudad diferente.



Motivo: Para esta dimensión tenemos que eliminar todos los campos a excepción de la columna *motivo*. Al ordenarlos y eliminar repetidos, nos quedamos únicamente con 2. Añadimos un identificador a cada uno y mostramos el resultado:

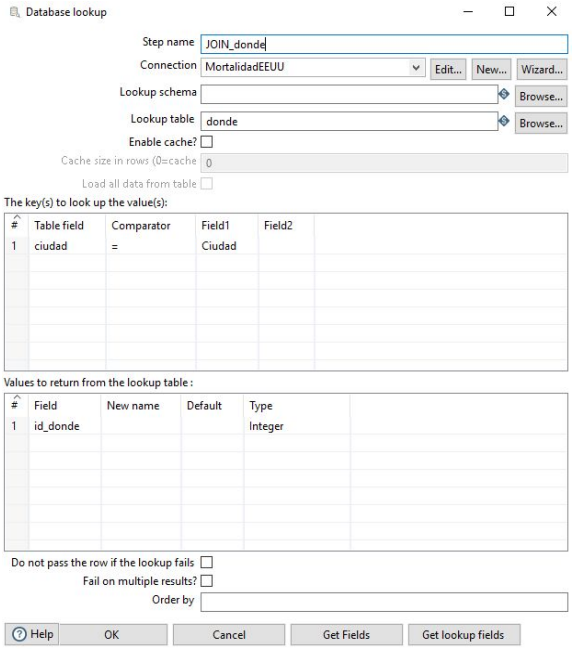


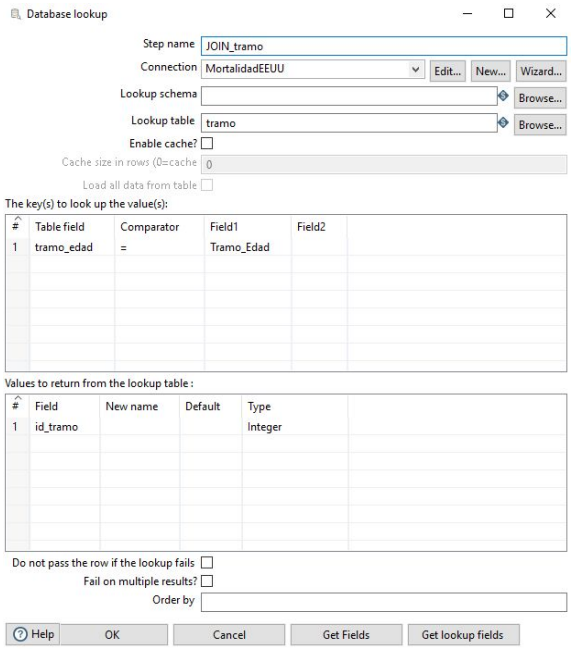
Tramo: Eliminamos todas las columnas menos las dos referentes al tramo de edad. Después le asignamos un identificador *id_tramo* que viene unívocamente determinado por cada valor que toma *tramo_edad*.



Todos los nodos/pasos usados anteriormente no tienen mayor complicación, simplemente leen, ordenan, eliminan, añaden y crean campos nuevos en nuestra base de datos.

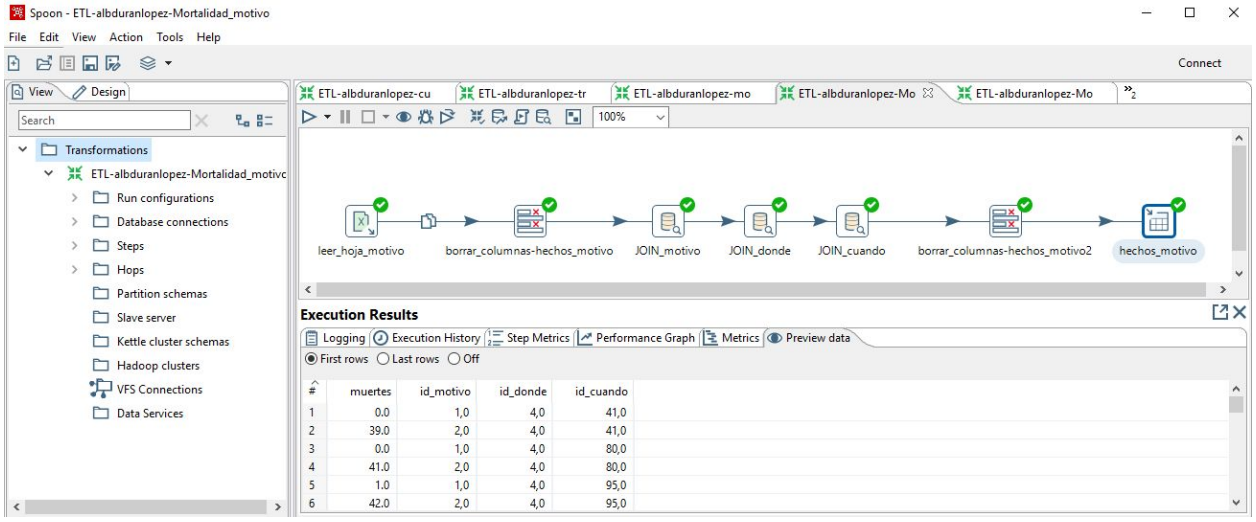
Sin embargo, para realizar cada tabla de hechos se ha requerido del uso de un nuevo nodo que introducimos a continuación. Se trata de **database lookup** y se encarga de realizar la operación JOIN. Se le ha de indicar la conexión de la base de datos (**MortalidadEEUU**), la tabla de la dimensión con la que queremos hacer el JOIN, el campo en común para realizar la operación y, por último, los valores a devolver.



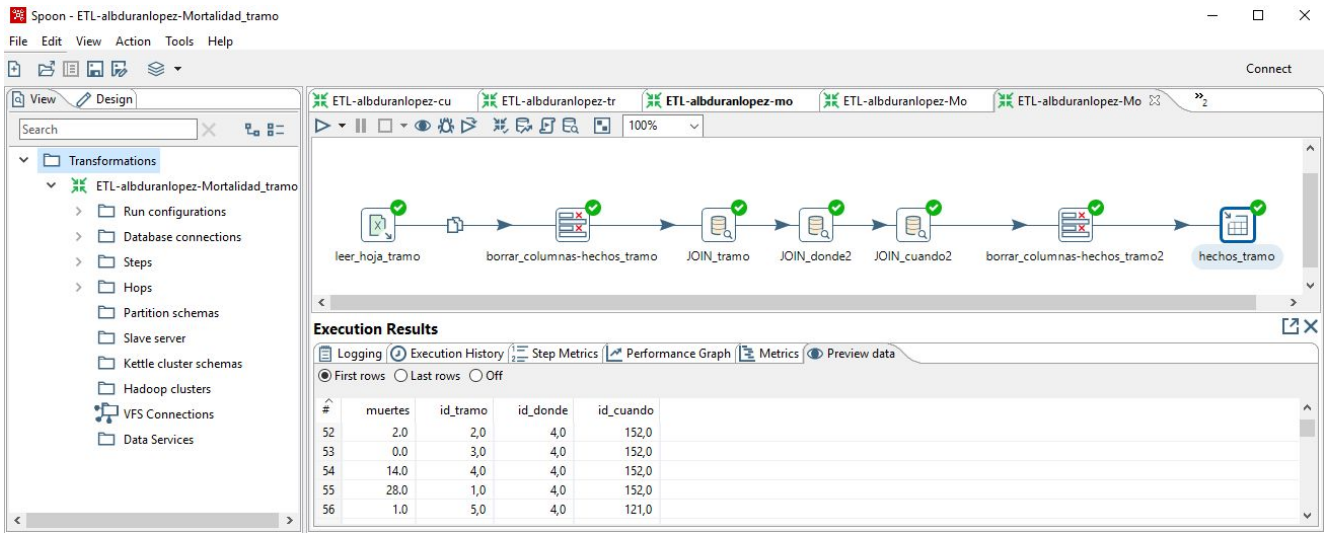


Se ha mostrado la configuración de dos ejemplos distintos donde se ha realizado la operación JOIN. El resto se harían exactamente igual, aunque no mostramos captura de ellos.

Hechos_motivo/Mortalidad_motivo: Para la tabla de hechos, nos tenemos que quedar únicamente con el campo *muertes* y los necesarios para hacer el JOIN por cada dimensión, eliminando el resto que no se necesiten. Ahora bien, utilizamos el paso llamado **database_lookup** que nos permitirá hacer la operación JOIN. Una vez realizado un JOIN para la dimensión *Motivo*, otro para la dimensión *Dónde* y uno último para la dimensión *Cuándo*, eliminamos los campos sobrantes y mostramos el resultado final:



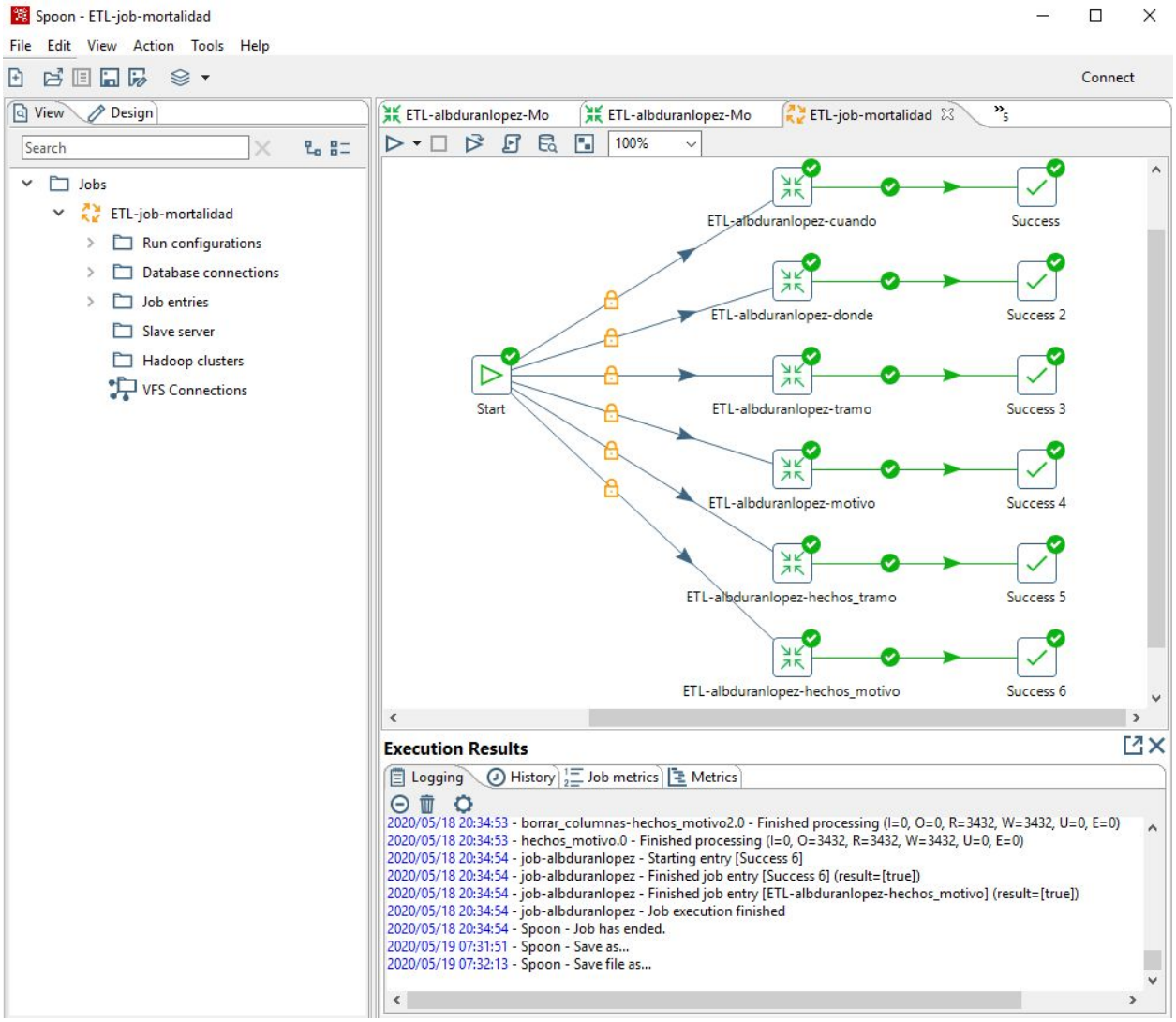
Hechos_tramo/Mortalidad_tramo: Se sigue un procedimiento similar al realizado en la otra tabla de hechos pero con la diferencia de que en vez de realizar un JOIN con la tabla *motivo*, se realiza con la tabla *tramo*, obteniendo como resultado:



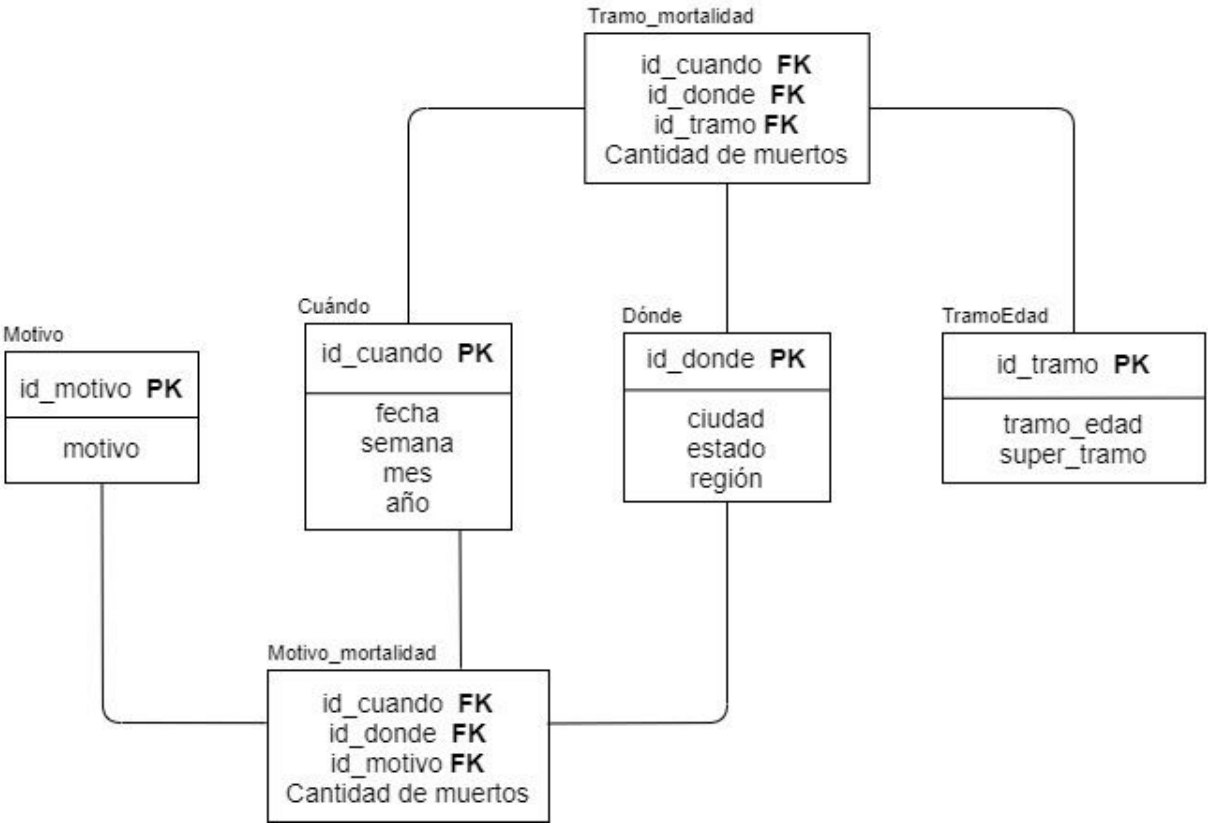
3.1 Automatización mediante un Job

Definimos un trabajo o job que se encargará de controlar la ejecución de las transformaciones definidas previamente. La configuración de cada paso es muy sencilla, simplemente se le ha de indicar la ruta del archivo de la transformación de cada dimensión o tabla de hechos.

Nuestro trabajo o job se ha definido en el archivo *ETL-job-mortalidad.kjb* y el resultado es el que se muestra a continuación:



Para entender mejor todo lo realizado en spoon, mostramos el diseño lógico que previamente se había hecho y se corresponde con lo realizado:



4. pgAdmin

pgAdmin es una herramienta utilizada para gestionar y administrar *PostgreSQL,*
La base de datos de código abierto más avanzada del mundo.

Aunque se ha comentado anteriormente, es necesario haber creado una base de datos previa a las modificaciones realizadas con la herramienta PDI.
En nuestro caso, la base de datos creada y utilizada es **MortalidadEEUU**

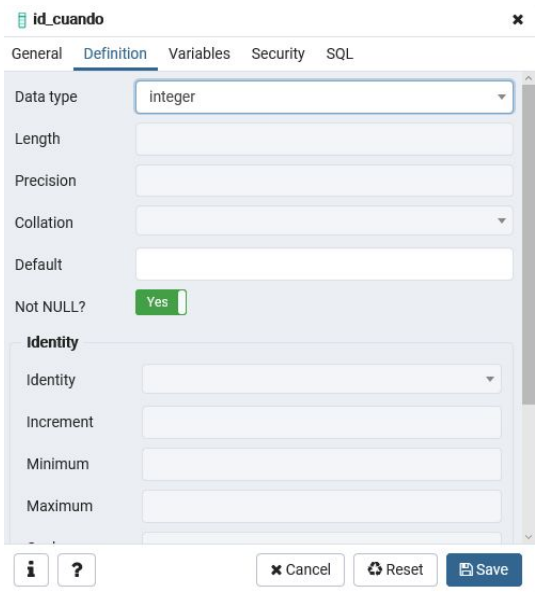
Una vez realizadas y ejecutadas cada una de las transformaciones referentes a cada dimensión o, en su defecto, a cada tabla de hechos, podemos ver en nuestra BD de pgAdmin nuestras tablas.
Para ello, tenemos que desplegar nuestra base de datos y dirigirnos a:
MortalidadEEUU > Schemas > Tables

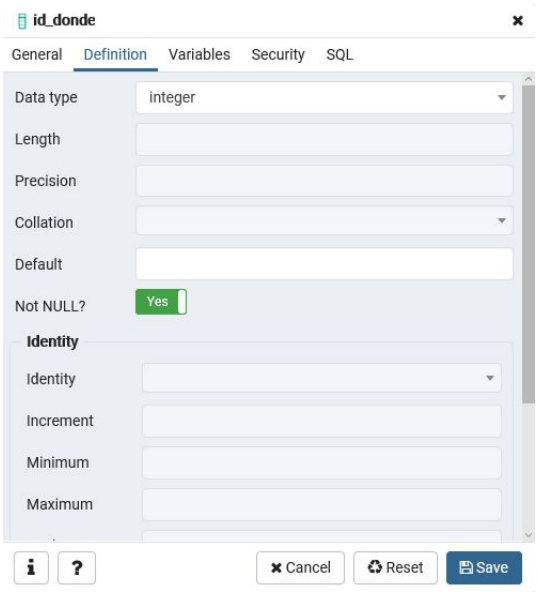
Recordemos los problemas que tuvimos en *Spoon* y cómo se han solucionado:

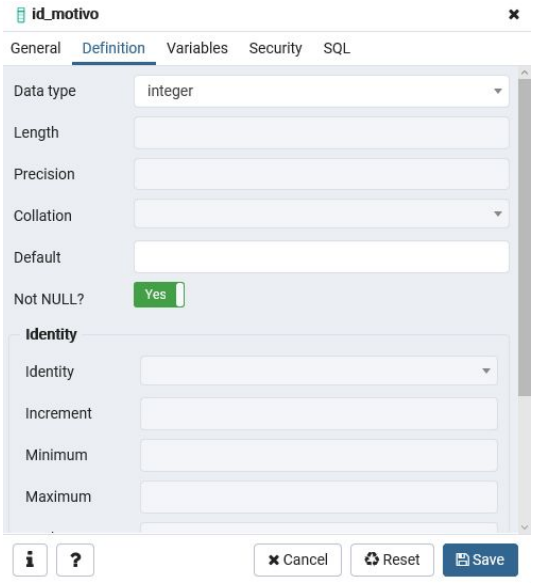
1. *Double precision*: Algunos campos como por ejemplo *Muertes* son de este tipo y no tiene sentido hablar de 35,0 muertes. Otro ejemplo erróneo visto anteriormente ha sido el campo *id_motivo* en la tabla de hechos **Mortalidad_motivo**.
   Solución: Modificar los campos manualmente. Todas las columnas con tipo erróneo *double precision* se han modificado a *integer*. Por otro lado, se ha activado la pestaña *NOT NULL* a los identificadores.

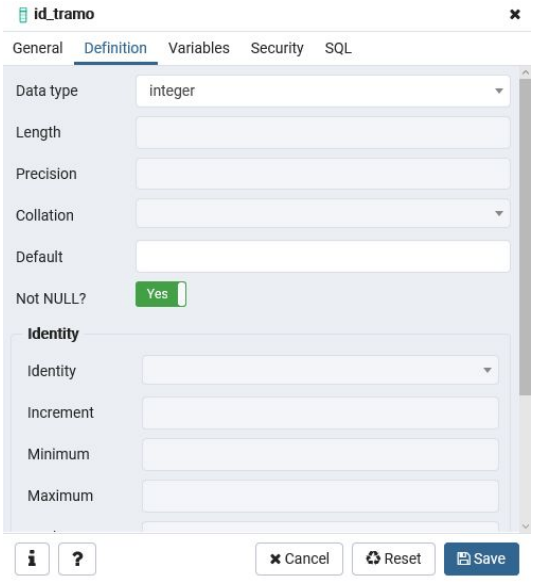
2. No tenemos en cuenta las relaciones de clave primaria y externa en nuestras tablas. Sin embargo, *pgAdmin* nos da la posibilidad de añadir dichas relaciones manualmente. Para cada tabla indicamos la clave primaria y para las tablas de hechos indicamos las claves primarias y externas.

A todos los datos le asociamos el *data type* correcto, modificamos los *double precision* para indicarles el tipo *integer*. Con los identificadores realizamos lo mismo pero además indicamos que no pueden ser nulos. Para ello activamos la pestaña *Not NULL* como se muestra a continuación:

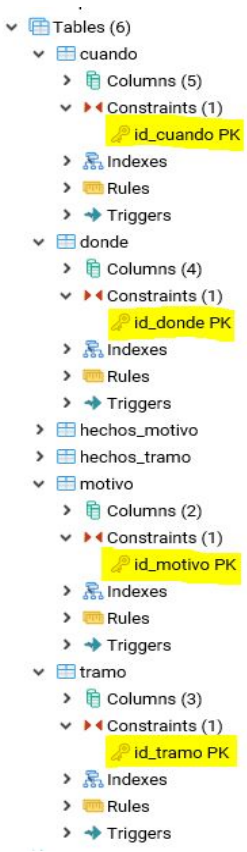








Relaciones de clave primaria y externa:



Mostramos en la captura de la izquierda las claves primarias que se han creado manualmente desde el menú conceptual de cada tabla.

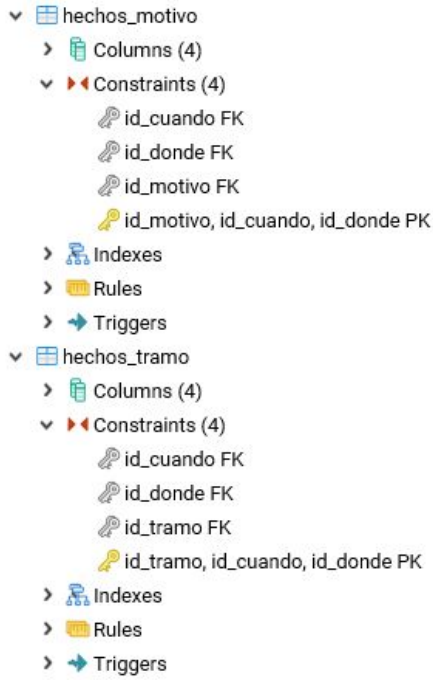
Recordemos que para cada dimensión teníamos un campo identificador por lo que para cada una de ellas le damos a *CREATE > PRIMARY_KEY* y le asociamos el nombre id_ seguido de PK para indicar que es clave primaria.

Por otro lado, para las tablas de hechos debemos crear una clave primaria formada por 3 campos:

-*id_motivo, id_cuando, id_donde* en el caso de la tabla de hechos *motivo*

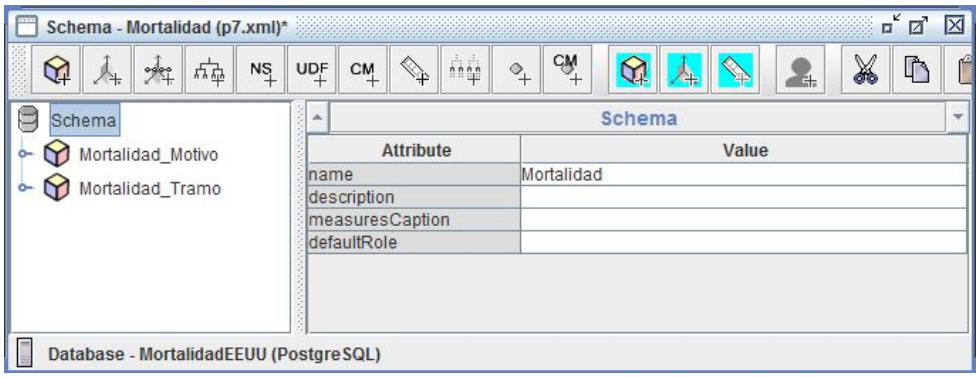
-*id_tramo, id_cuando, id_donde* en el caso de la tabla de hechos *tramo*

A ambas le indicamos el sufijo PK
Por otro lado, tenemos que crear una clave externa por cada dimensión a la que se referencia. En total 3 por cada tabla de hechos.
Le indicamos el sufijo *FK* (foreign key) a cada una de ellas.



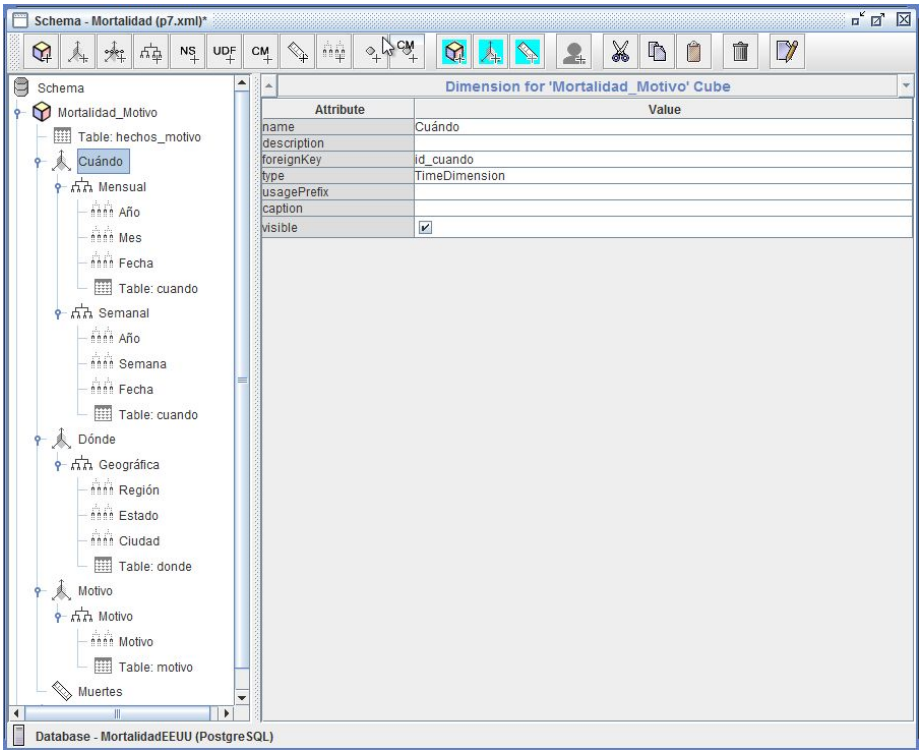
5. Schema Workbench

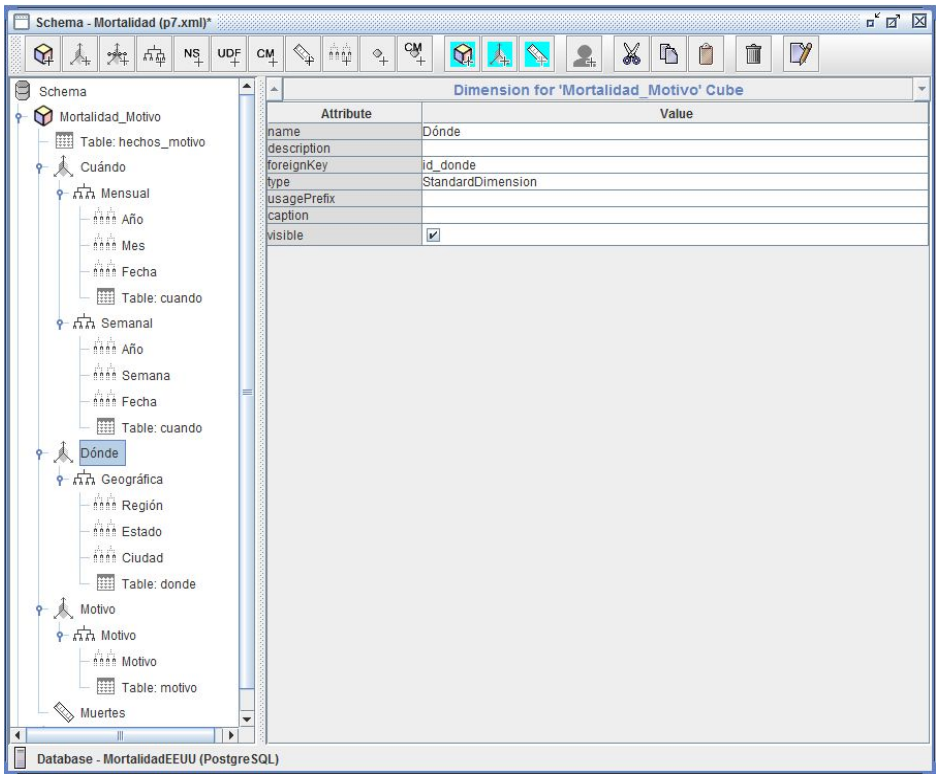
El objetivo del uso de esta herramienta es la obtención del esquema multidimensional (archivo XML). Para ello, tenemos que establecer la conexión con nuestra base de datos y, como tenemos dos cubos, tenemos que definir las distintas jerarquías para cada uno de ellos:



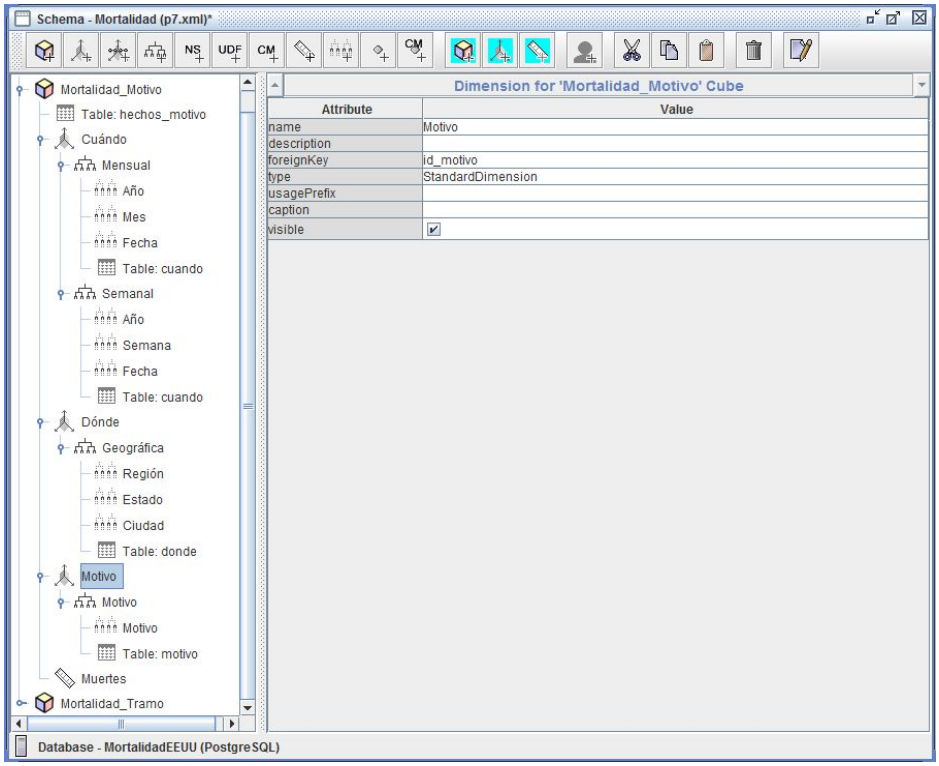
Dimensiones:

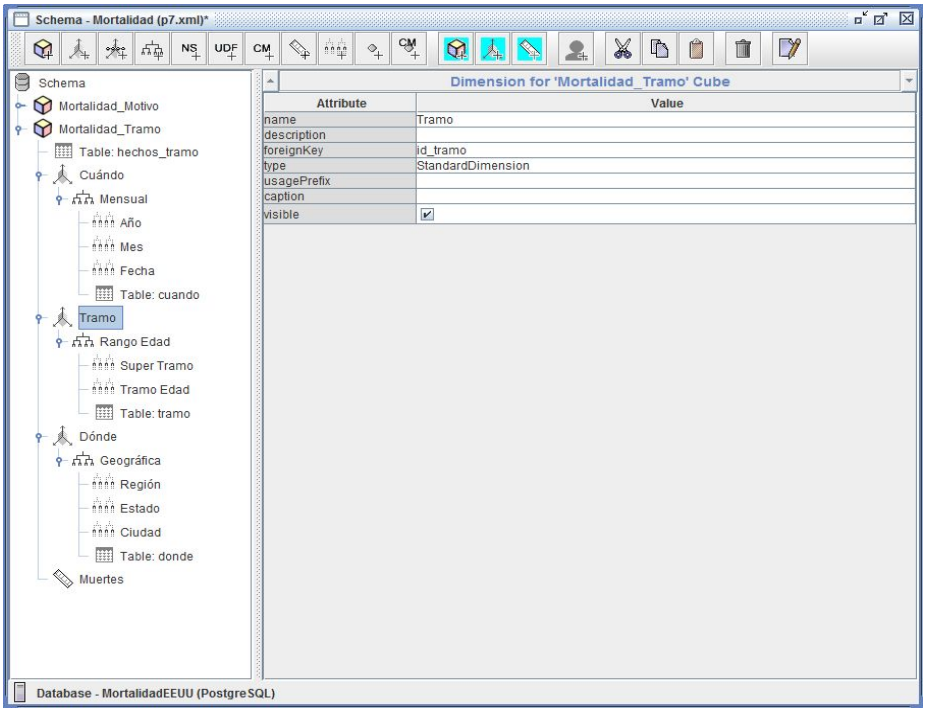
Las configuración dimensiones *Cuándo* y *Dónde* es igual para ambas, teniendo en cuenta que la primera de ellas es de tipo *TimeDimension* y la segunda *StandardDimension*.





El resto de dimensiones (*Motivo* y *Tramo*) se definen de la siguiente forma:





Destacamos que en todas ellas indicamos como *foreignKey* el campo de la tabla de hechos que referencia a la tabla de la dimensión.

Jerarquías y Niveles

Por cada dimensión incluímos una jerarquía mediante la operación <<*Add Hierarchy*>>
A cada jerarquía le asociamos niveles mediante la operación <<*Add Level*>> siempre en orden descendente (de arriba hacia abajo). Las definidas han sido:

Dimensión Cuándo:

1. Jerarquía Mensual:
   Sus niveles son:
   a. Año
   b. Mes
   c. Fecha

2. Jerarquía Semanal
   Sus niveles son:
   d. Año
   e. Semana
   f. Fecha

Dimensión Dónde:

1. Jerarquía Geográfica
   Sus niveles son:
   a. Región
   b. Estado
   c. Ciudad

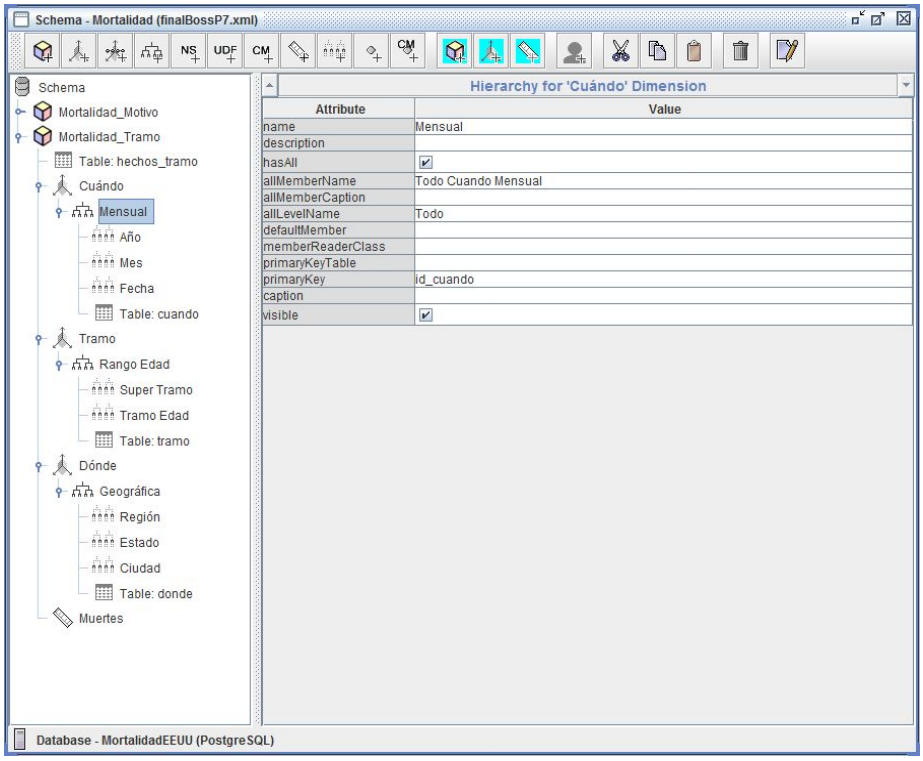
Dimensión Tramo:

2. Jerarquía Tramo
   Sus niveles son:
   a. Super tramo
   b. Tramo edad

Dimensión Motivo:

1. Jerarquía Motivo:
   Su nivel es :
   a. Motivo

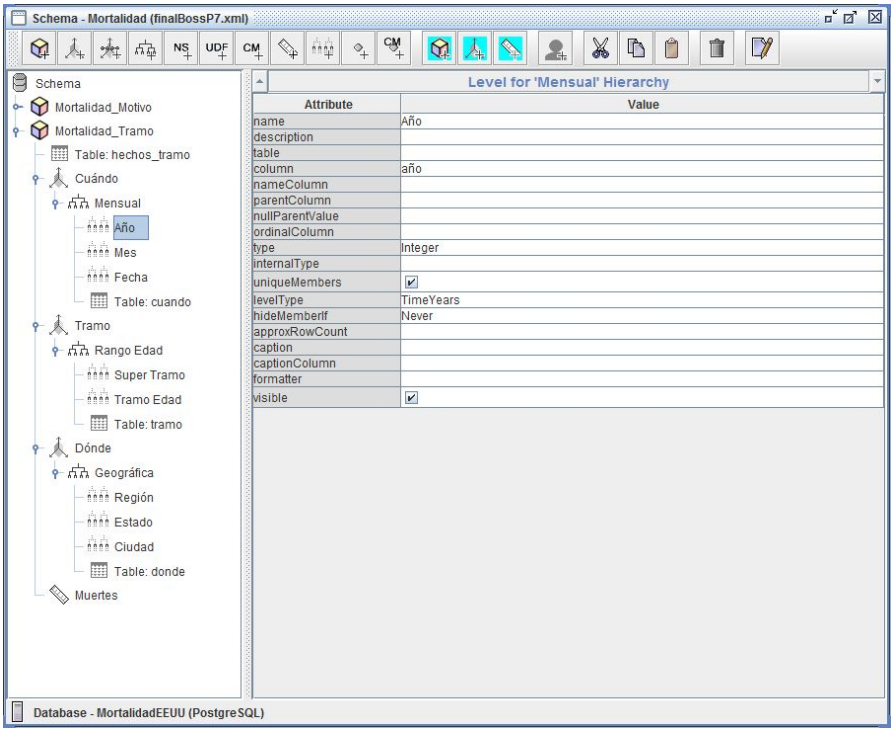
Mostramos algunas capturas a modo de ejemplo:

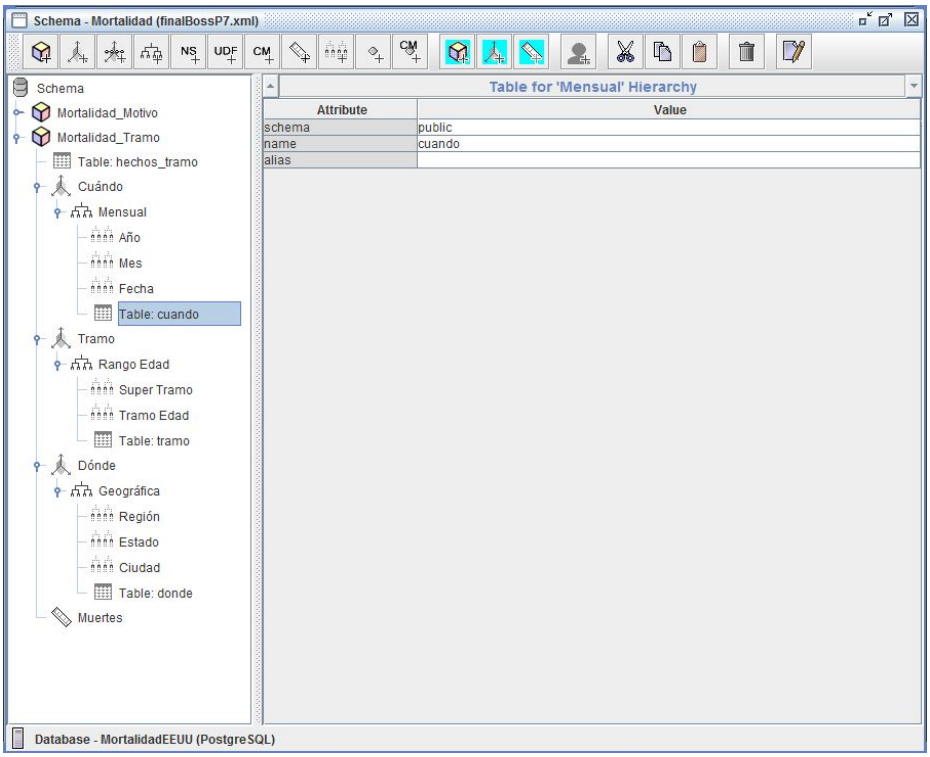


En esta captura mostramos la configuración de la *jerarquía mensual*. Destacamos que en la casilla *primaryKey* se ha de indicar la llave primaria de la tabla de la dimensión.

En esta otra, mostramos la configuración del nivel Año.

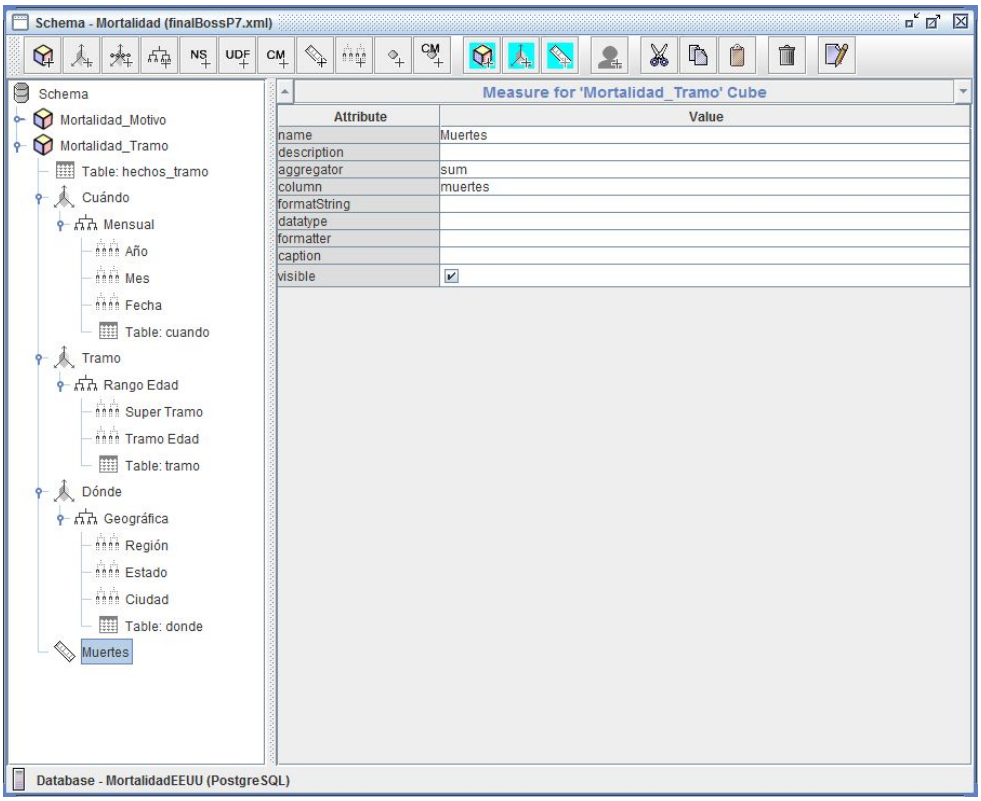
En todas las casillas donde se pueda hacer se ha marcado la opción *uniqueMembers* para así optimizar la generación de SQL. Como se vió en la práctica 5, los miembros son únicos si, considerando todas las instancias de un nivel, los valores no se repiten. En este caso *Año* sí es único, pero por ejemplo *Mes* no lo es ya que se repite para cada *Año*





Asignamos la tabla correspondiente a cada jerarquía. En este caso, la jerarquía es *Mensual,* luego la tabla es *cuando.*

Por último, mostramos la configuración de nuestra medición *Muertes*.

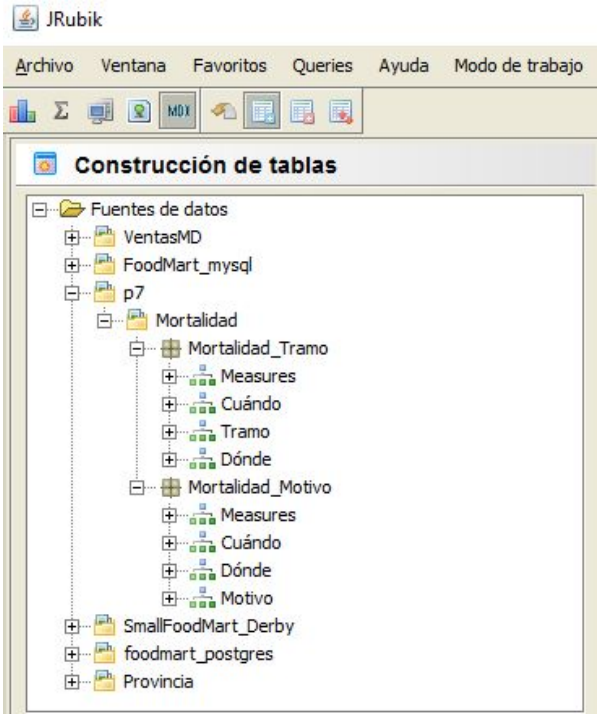


El resto de Jerarquías y niveles se realizan de la misma forma explicada anteriormente.

6. Jrubik

Jrubik es un cliente Olap realizado en java/swing sobre componentes del proyecto JPivot. Este cliente es capaz de conectar a fuentes Olap basadas en el motor relacional Mondrian.
Este cliente nos ofrece numerosas funcionales pero en nuestro caso, nos centraremos en el gestor de consultas.

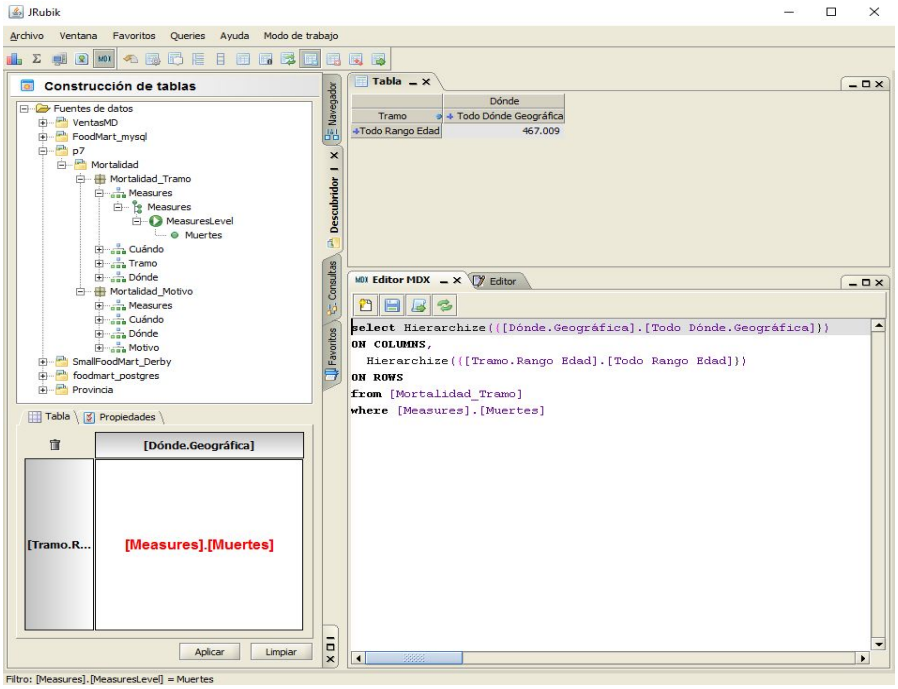
En primer lugar, situamos el archivo *XML* resultante de *Schema Workbench* en la carpeta de catálogos de *Jrubik*. Después, configuramos la fuente de datos para añadir nuestro archivo y guardamos. Desplegando nuestro proyecto, podremos ver nuestros dos cubos.



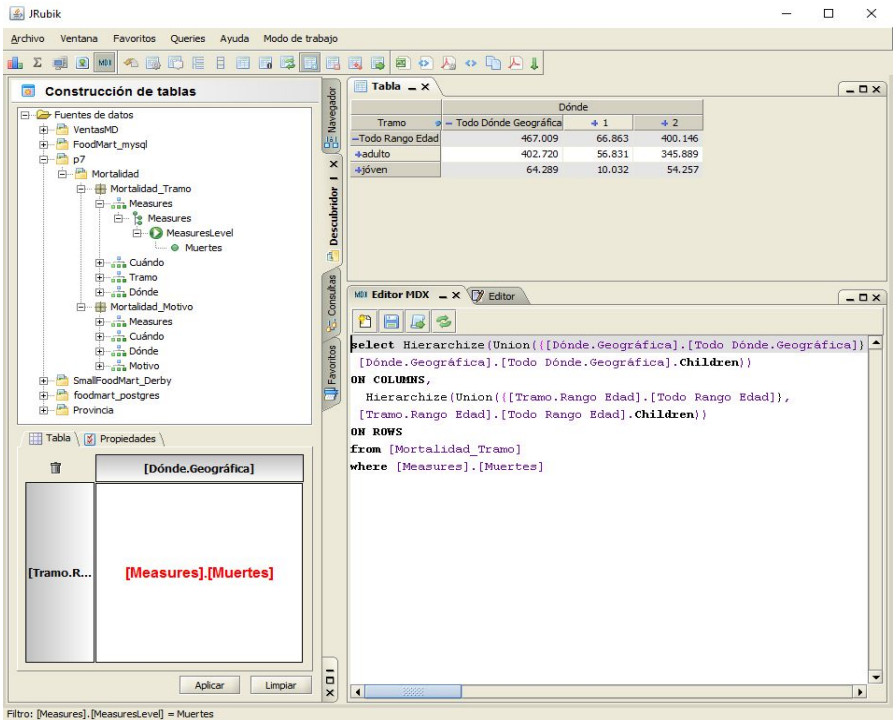
El siguiente paso es el de realizar informes para comprobar la funcionalidad de esta herramienta y mostrar que todo se ha realizado correctamente.

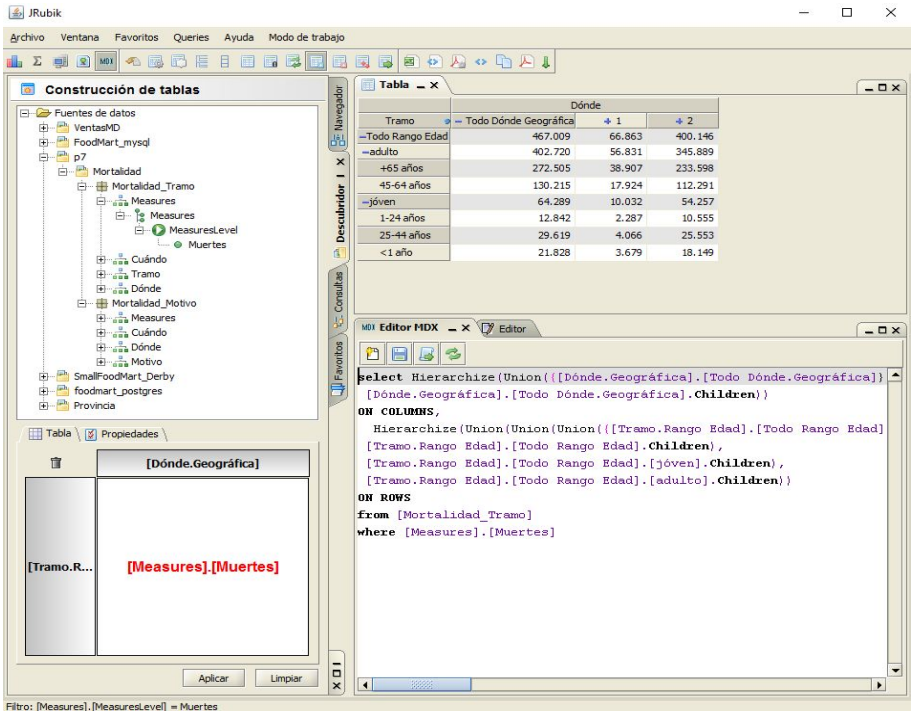
6.1 Informe Mortalidad_tramo:

En primer lugar, mostramos el número de muertes en el nivel todo de las jerarquías *Tramo* y *Dónde*. Como filtro se toma el nivel todo de la dimensión *Cuándo*.



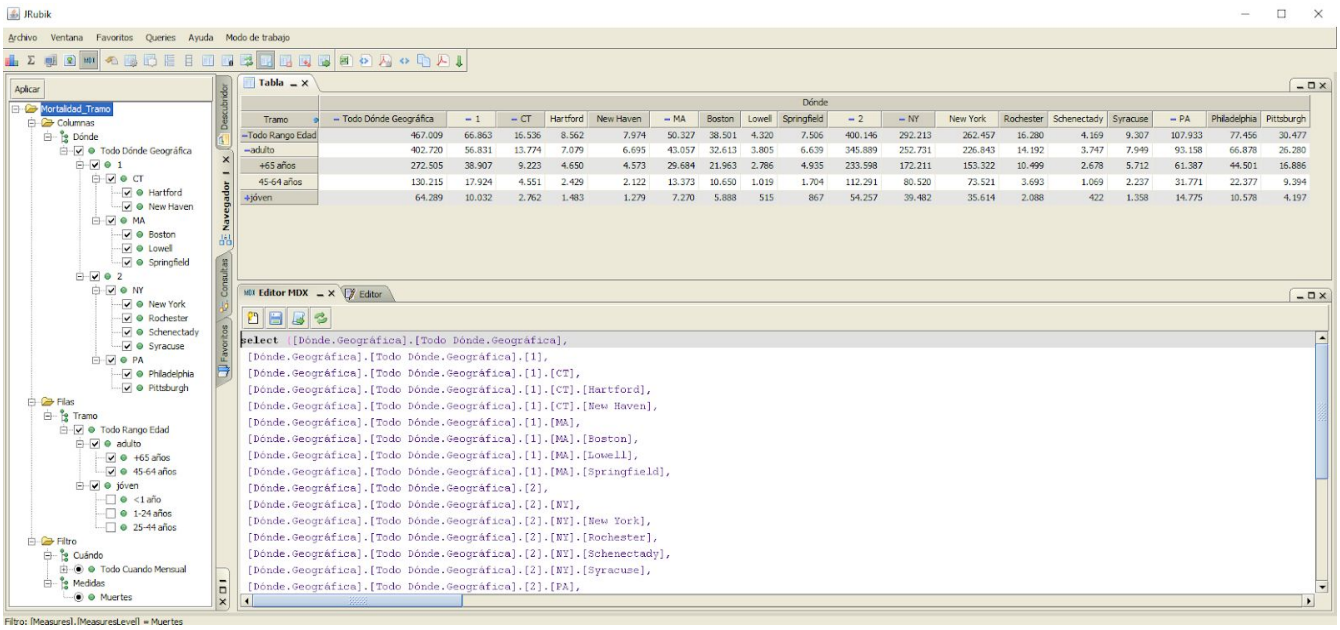
A continuación, realizamos un **Drill-Down** en la dimensión *Tramo* y *Dónde*, aumentando así el nivel de detalle y mostrándose el nº de muertes por *tramo* (adulto, joven) en las regiones 1 y 2.





Realizamos ahora un **Drill-Down** en la dimensión *Tramo* para mostrar el nº de muertes por cada rango de edad, observándose que los mayores de 65 años son los más perjudicados.

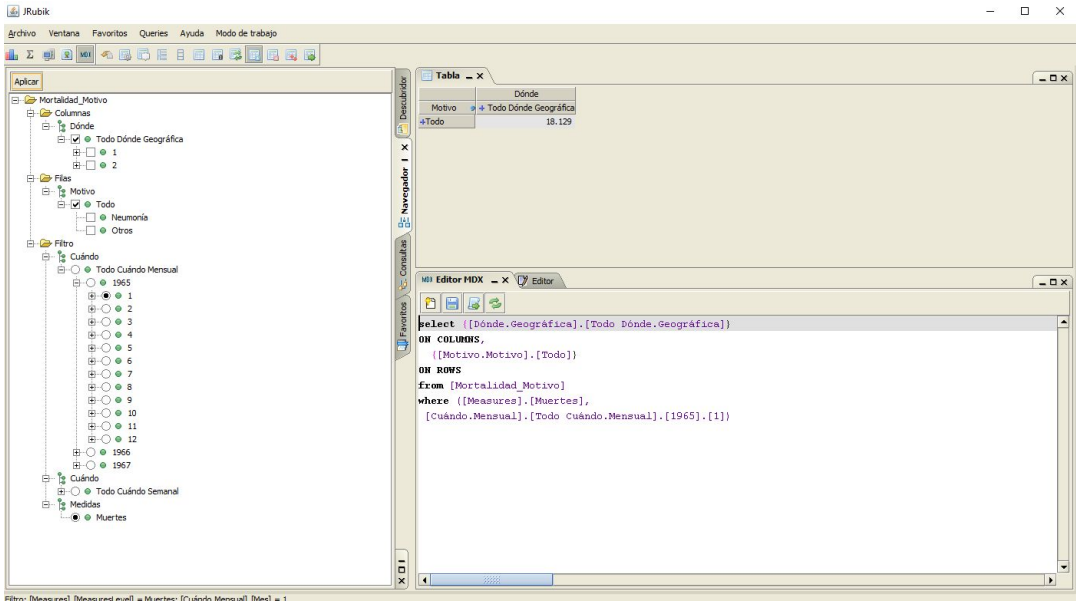
Por último, realizamos un ***Drill-Down*** y mostramos al máximo nivel de detalle (ya que expandimos al máximo todos los datos) en la dimensión *Dónde,* mostrando las ciudades de cada estado de cada región y un **Slice&Dice** en la dimensión *Tramo*, ya que tenemos el mismo nivel de detalle pero únicamente se muestran datos del los tramos *45-64 años* y *+65 años.*



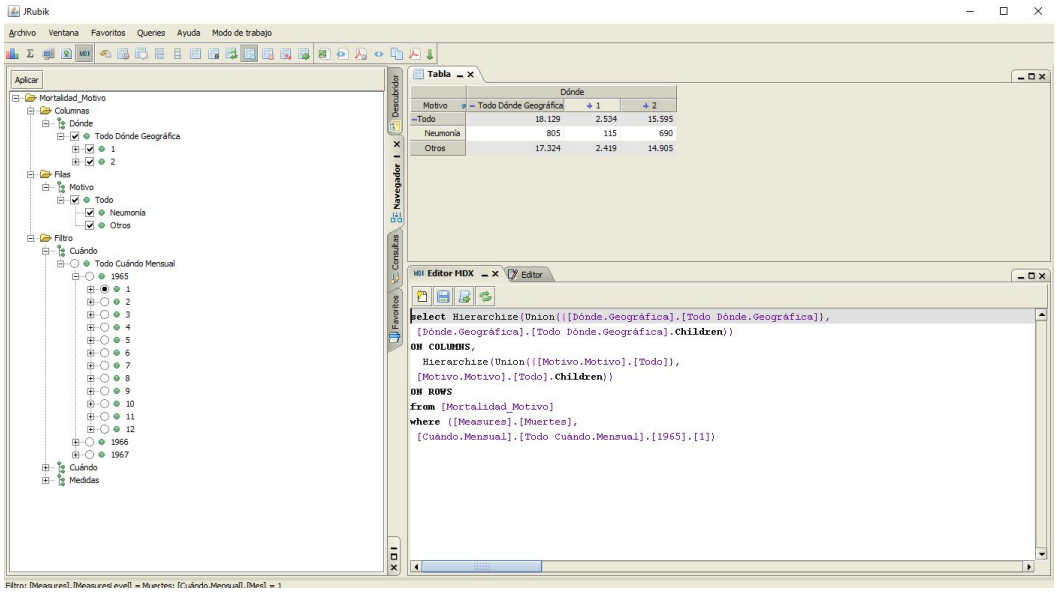
Aunque no se indica en este informe, si leyéramos desde esta captura hasta la primera, las operaciones que se habrían realizado serían **Roll-UP** en vez de **Drill-Down** ya que se agrupan los datos y se oculta detalle (operación de muchos a uno)

6.2 Informe Mortalidad_Motivo:

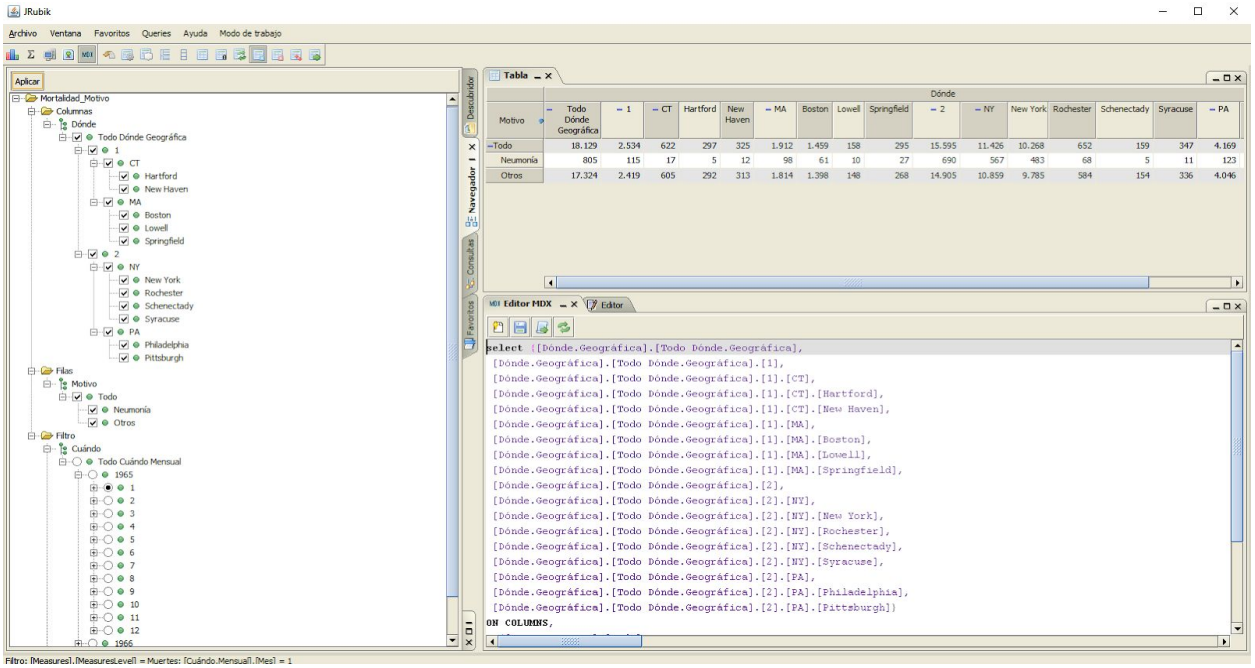
Ya que hemos hecho un cubo específico para poder realizar consultas por motivo de la muerte, usamos la jerarquía *Motivo* y *Dónde* para las filas y columnas, respectivamente.
Adicionalmente, podemos elegir como filtro cualquier nivel de las jerarquías definidas para la dimensión *Cuándo*.
Mostramos un informe inicial reflejando el nivel todo de las dimensiones *Motivo* y *Dónde* y, como filtro, elegimos el mes de Enero (mes 1) del año 1965.



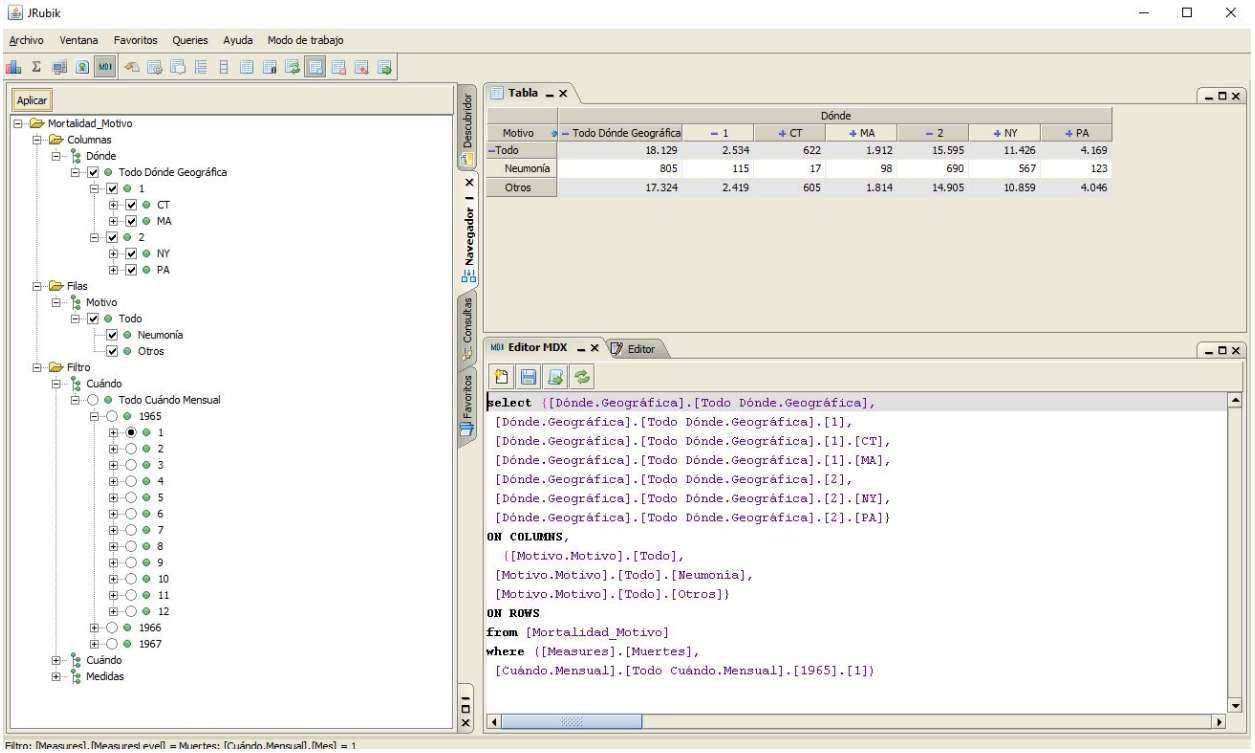
Realizamos un **Drill-Down** en ambas dimensiones, es decir, aumentamos el nivel de detalle ya que expandimos los datos para mostrar los 2 motivos de muerte (Neumonía y otros) y las dos regiones. Se trata de una operación de “uno a muchos”



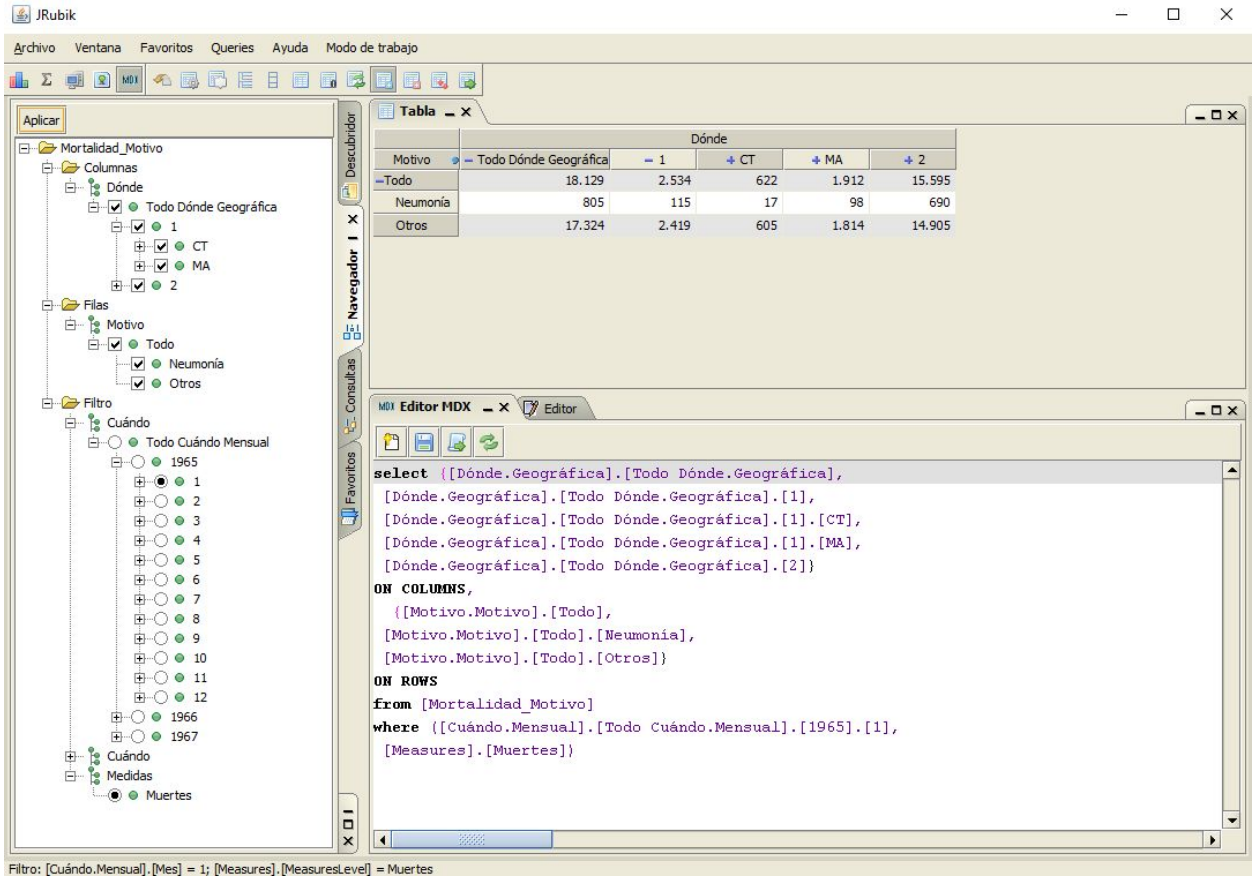
Realizamos un **Drill-Down** para así mostrar el máximo nivel de detalle en ambas dimensiones.



Ahora bien, realizamos un **Roll-UP** en la dimensión *Dónde*. Operación en la que se agrupan los datos y se oculta detalle. En este caso pasamos de mostrar a nivel de detalle *ciudad* a mostrar a nivel de detalle *estado* (CT, MA pertenecientes a la región 1; NY, PA pertenecientes a la 2)



Por último, realizamos un **Slice&Dice** en la dimensión *Dónde*. Tenemos el mismo nivel de detalle pero mostramos únicamente los datos de los estados CT y MA, pertenecientes a la región 1.



7. Referencias

-María Carina Roldan. Learning Pentaho Data Integration 8 CE (Third Edition).

-Maria Carina Roldán. Pentaho Data Integration Quick Start Guide.

-Guión de prácticas 3

-Guión de prácticas 5

-Guión de prácticas 7

-Apuntes de clase [modelo conceptual y lógico]