

# SMD. Práctica 5. Herramientas OLAP

## *Mondrian*

José Samos Jiménez

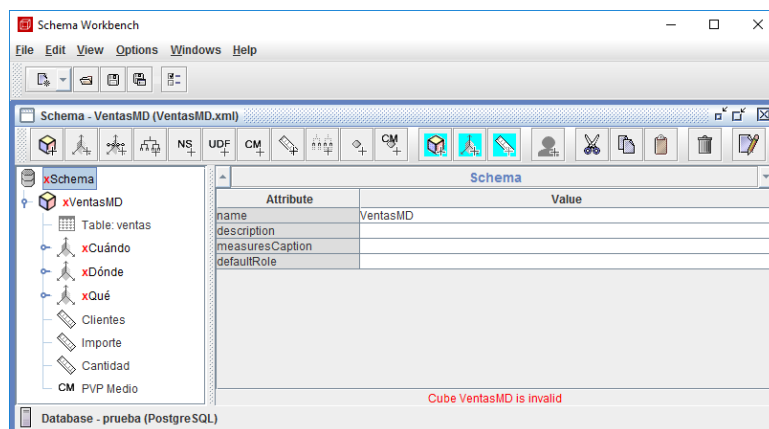
Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Granada

2020 jsamos (LSI-UGR)

Curso 2019-2020

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Definición del esquema multidimensional</b>	<b>3</b>
2.1. Conexión con la BD . . . . .	4
2.2. Definición del esquema y el cubo . . . . .	5
2.2.1. Esquema . . . . .	5
2.2.2. Cubo . . . . .	5
2.3. Definición de dimensiones, jerarquías, niveles y propiedades . . . . .	5
2.3.1. Dimensiones . . . . .	5
2.3.2. Jerarquías . . . . .	6
2.3.3. Niveles . . . . .	6
2.3.4. Propiedades . . . . .	7
2.4. Definición de mediciones y mediciones calculadas . . . . .	7
2.4.1. Mediciones . . . . .	8
2.4.2. Mediciones calculadas . . . . .	8
2.5. Operaciones sobre el archivo XML . . . . .	9
<b>3. Definición de consultas OLAP</b>	<b>9</b>
3.1. Configuración de <i>JRubik</i> . . . . .	9
3.2. Construcción de tablas . . . . .	10
3.3. Refinar la construcción de tablas . . . . .	11
<b>4. Operaciones a realizar</b>	<b>11</b>
4.1. Definición del esquema multidimensional . . . . .	11
4.2. Definición de consultas OLAP . . . . .	11
<b>Bibliografía</b>	<b>11</b>

Figura 1: Esquema abierto con *Schema Workbench*.

Los objetivos de esta actividad son:

- Entender el desarrollo de esquemas multidimensionales.
- Definir los elementos de un esquema multidimensional para un caso sencillo.
- Usar una herramienta OLAP profesional.
- Aprender el funcionamiento básico de la herramienta OLAP *Mondrian*.

A continuación, después de una introducción sobre las herramientas que usaremos, se irán presentando algunas de sus funcionalidades utilizando los datos que se facilitan. Posteriormente, deberás realizar operaciones similares sobre tus propios datos.

Descarga el archivo `ventas_md.zip` que contiene los archivos `ventas_md` y `VentasMD.xml`, el backup de una BD *PostgreSQL* y el esquema de *Mondrian* sobre esa BD, respectivamente, que se usarán en los apartados siguientes. Extrae dichos archivos en una carpeta temporal. En *PostgreSQL*, crea una BD llamada `ventas_md` y restáurala con el backup del archivo descargado (en el menú contextual de la BD, la opción «Restore»).

## 1. Introducción

*Mondrian* es un motor OLAP escrito en *Java*: ejecuta consultas MDX sobre datos en una BD relacional y devuelve los resultados en formato multidimensional ([Hyd09, BGH14]). Como BD relacional usaremos *PostgreSQL*.

Adicionalmente, necesitamos una herramienta que permita generar consultas MDX, enviárselas a *Mondrian* y presentar adecuadamente los resultados devueltos en formato multidimensional. Vamos a utilizar *JRubik*<sup>1</sup>: es un cliente OLAP sobre *Mondrian* desarrollado en *Java/Swing* utilizando componentes del proyecto *JPivot*.

*Mondrian* ofrece una visión multidimensional sobre los datos en la BD relacional mediante un esquema multidimensional definido en XML. Para definir el esquema podemos utilizar directamente un editor de XML o bien utilizar *Schema Workbench* que permite definir los elementos del esquema basándonos en la BD relacional que contiene los datos.

## 2. Definición del esquema multidimensional

El esquema multidimensional es un archivo XML. El que hemos descargado es `VentasMD.xml`. Podemos abrirlo con *Schema Workbench* (figura 1), a la izquierda se representan de los elementos del esquema y, en la ventana central, se muestran los atributos del elemento seleccionado. Los iconos de

<sup>1</sup>[http://rubik.sourceforge.net/spanish/jrubik\\_es.html](http://rubik.sourceforge.net/spanish/jrubik_es.html)

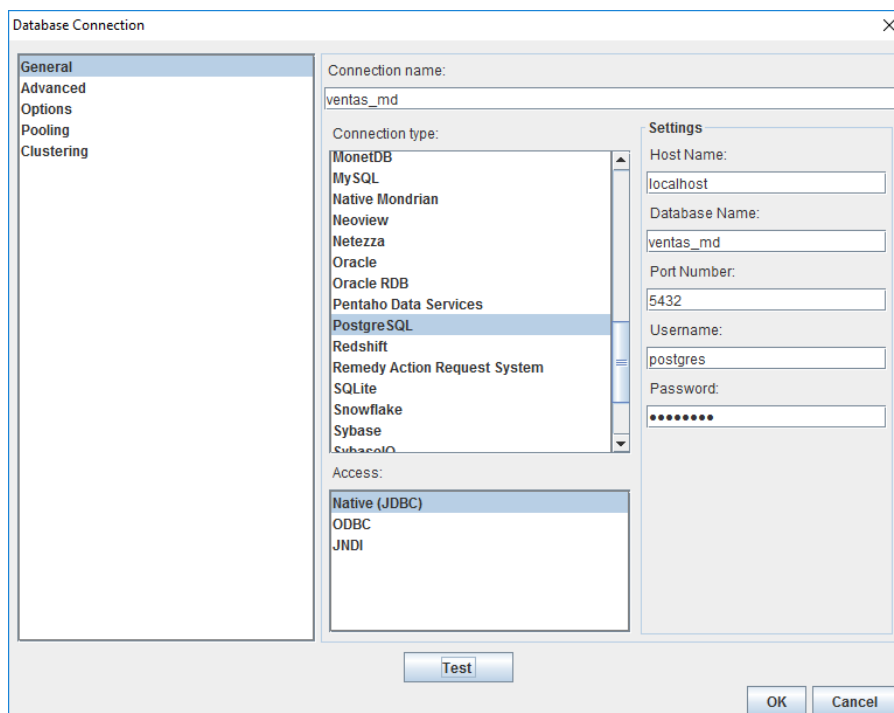


Figura 2: Conexión con la BD.

la barra de herramientas nos permiten añadir nuevos elementos al elemento seleccionado; también se pueden añadir desde el menú contextual del elemento correspondiente.

En la línea de estado de la parte inferior, avisa de los errores que ha detectado para el elemento seleccionado, y marca con una «x» roja los elementos con errores.

El error que muestra para el esquema no es de gran ayuda («Cube VentasMD is invalid») pero, si pulsamos sobre el cubo el error es «Fact table ventas does not exist in database schema public»: es decir, busca en una BD una tabla llamada **ventas** pero no la encuentra.

En el esquema se asocian los elementos del modelo de datos multidimensional a nivel conceptual con los elementos del esquema de la BD, el esquema a nivel lógico. Para evitar errores en el momento de la ejecución de las consultas, *Schema Workbench* permite definir el esquema multidimensional accediendo a los elementos de la BD relacional para establecer las correspondencias.

Por este motivo, antes de definir el esquema (o para comprobar la definición) hemos de definir la conexión con la BD. Posteriormente definiremos los componentes del esquema.

Respecto a la definición de los componentes del esquema, *Schema Workbench* ofrece muchas posibilidades de definición y *Mondrian* soporta todavía más posibilidades en los esquemas ([Hyd09, BGH14]). Aquí nos vamos a focalizar en las más básicas para comenzar en este primer contacto con las herramientas.

## 2.1. Conexión con la BD

La conexión con la BD se define pulsando sobre [«Options», «Connection»]. Se abre la ventana donde podemos seleccionar el tipo de BD y definir los parámetros de la conexión (figura 2). En este caso, es suficiente con definir los parámetros del apartado «General»: para *PostgreSQL* seleccionamos el acceso mediante JDBC e indicamos los valores habituales de conexión, además del nombre de la BD que hemos restaurado.

Al definir correctamente la conexión, podemos observar que desaparecen los errores del esquema (figura 3). Adicionalmente, en la parte izquierda de la línea de estado se muestra la BD con la que se ha establecido la conexión (en la figura 1 se mostraba una BD de una prueba previa).

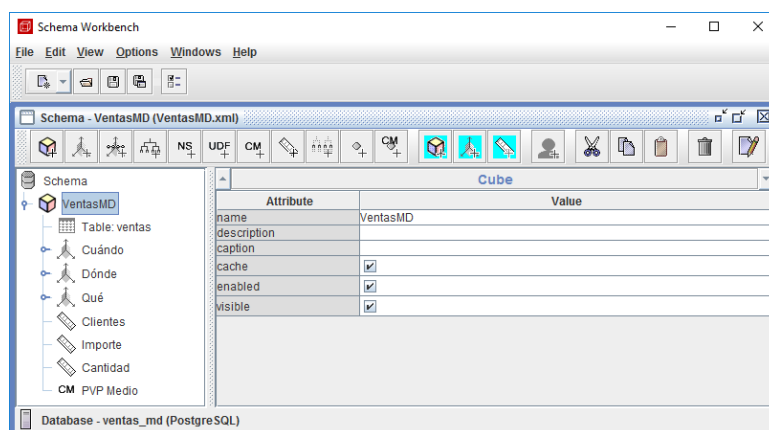


Figura 3: Esquema abierto sin errores.

## 2.2. Definición del esquema y el cubo

Un esquema puede contener varios cubos. Nosotros vamos a definir un solo cubo en el esquema.

### 2.2.1. Esquema

Para definir un nuevo esquema pulsamos sobre el icono «New» de la ventana de la aplicación y seleccionamos «Schema» o bien pulsamos sobre [«File», «New», «Schema»]. Para el esquema es suficiente con definir el nombre en el campo «name» (figura 1).

### 2.2.2. Cubo

Una vez definido el esquema, podemos definir un cubo en él pulsando sobre el icono «Add cube» de la barra de herramientas de la ventana del esquema o bien desde el menú contextual del esquema, seleccionando esa misma opción. El único atributo que necesitamos definir del cubo también es su nombre (figura 3).

Cuando crea el cubo nos avisa del error «Fact name must be set»: debemos indicar los hechos. Esto se hace asociando la tabla de hechos al cubo, seleccionando la opción «Add table» en su menú contextual. Para cada tabla que indiquemos, debemos indicar el nombre del esquema y el de la tabla, en este caso la tabla es **ventas**.

Al definir la tabla, el error asociado al cubo cambia a «Cube must contain dimensions».

## 2.3. Definición de dimensiones, jerarquías, niveles y propiedades

Las dimensiones contienen jerarquías con niveles que pueden tener definidas propiedades.

### 2.3.1. Dimensiones

Estando situados en el cubo, definimos una nueva dimensión mediante la operación «Add dimension».

Para definir una dimensión debemos dar valores a los atributos (figura 4):

- «name»: el nombre de la dimensión,
- «foreignKey»: el campo de la tabla de hechos que referencia a la tabla de la dimensión, nos permite seleccionarlo si tenemos definida la conexión con la BD, y
- «type»: el tipo de la dimensión, la dimensión tiempo es de tipo «TimeDimension» el resto de dimensiones son del tipo «StandardDimension».

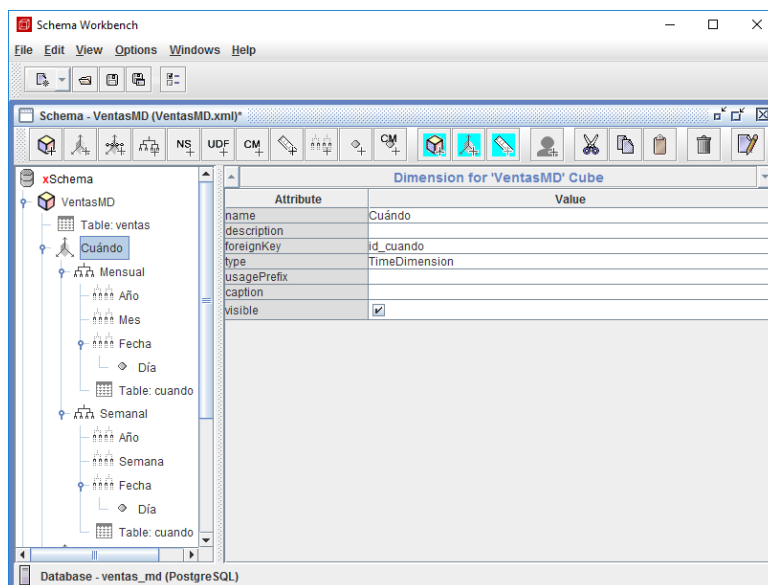


Figura 4: Definición de una dimensión.

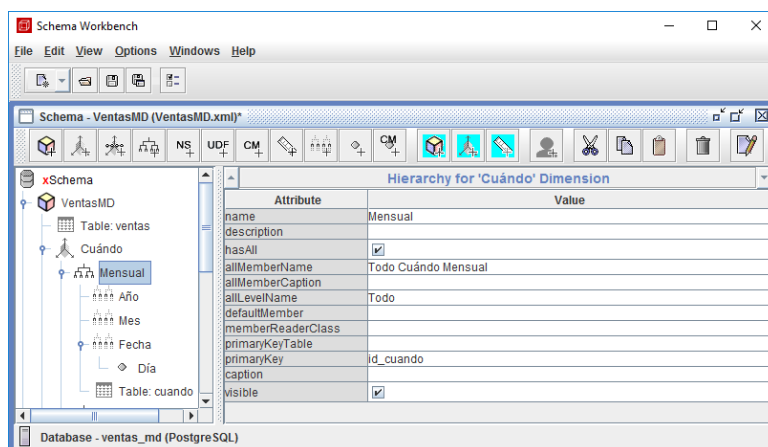


Figura 5: Definición de una jerarquía.

### 2.3.2. Jerarquías

Cada dimensión puede incluir varias jerarquías. Se definen mediante la operación «Add Hierarchy» desde la dimensión correspondiente. En lugar de definir una tabla asociada a la dimensión, cada jerarquía tiene asociada una tabla de la BD.

Para definir una jerarquía debemos dar valores a los atributos (figura 5):

- «name»: el nombre de la jerarquía,
- «hasAll»: el nivel *Todo* es opcional, indicamos si tiene nivel *Todo*. En caso de tenerlo, podemos indicar el nombre del nivel y de su instancia:
  - «allMemberName»: valor de la única instancia del nivel *Todo* y
  - «allLevelName»: nombre del nivel *Todo*.
- «primaryKey»: llave primaria de la tabla de la dimensión.

### 2.3.3. Niveles

Cada dimensión puede tener varios niveles. Se definen mediante la operación «Add Level» desde la jerarquía correspondiente. Existe un orden entre los niveles definidos, de arriba hacia abajo; se puede cambiar el orden de los niveles desde su menú contextual.

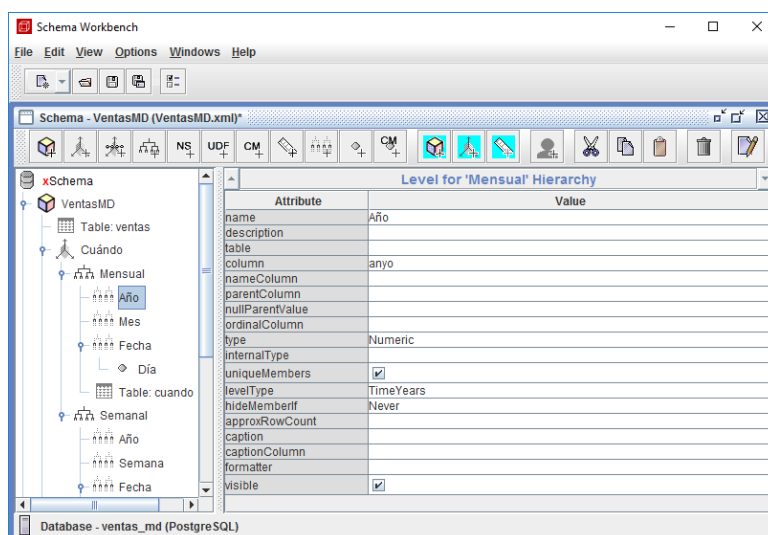


Figura 6: Definición de un nivel.

Para definir un nivel debemos dar valores a los atributos (figura 6):

- «name»: el nombre del nivel,
- «column»: columna de la tabla asociada al nivel<sup>2</sup>, se puede seleccionar de la lista obtenida si tenemos definida la conexión con la BD,
- «type»: seleccionamos el tipo del campo,
- «uniqueMembers»: este atributo se usa para optimizar la generación de SQL. Los miembros son únicos si, considerando todas las instancias de un nivel, los valores no se repiten. Así, para el nivel más alto de la jerarquía siempre será cierto; en el caso particular del nivel **Mes** no será cierto porque los nombres de los meses se repiten para cada año; para el nivel **Fecha** será cierto, porque las fechas no se repiten (la fecha incluye el día, mes y año). Si se tiene duda, se recomienda indicar el valor falso ([Hyd09]),
- «levelType»: para la dimensión temporal tiene predefinidos niveles que se pueden asociar a los niveles correspondientes.

#### 2.3.4. Propiedades

Cada nivel puede tener varias propiedades. Se definen mediante la operación «Add Property» desde el nivel correspondiente.

Para definir una propiedad debemos dar valores a los atributos (figura 7):

- «name»: el nombre de la propiedad,
- «column»: columna de la tabla asociada a la propiedad, tenemos que escribir su nombre (no conecta con la BD),
- «type»: seleccionamos el tipo del campo.

#### 2.4. Definición de mediciones y mediciones calculadas

Definimos mediciones y mediciones calculadas asociadas a cada cubo mediante las operaciones «Add Measure» y «Add Calculated Member», respectivamente.

<sup>2</sup>Se puede definir adicionalmente «nameColumn» y «ordinalColumn» para definir el nombre del nivel y el campo por el cual se ordenan sus valores. En las pruebas que he hecho con la herramienta OLAP no he conseguido que funcionen bien estos aspectos y no los vamos a usar, aunque están disponibles los campos para definirlos.

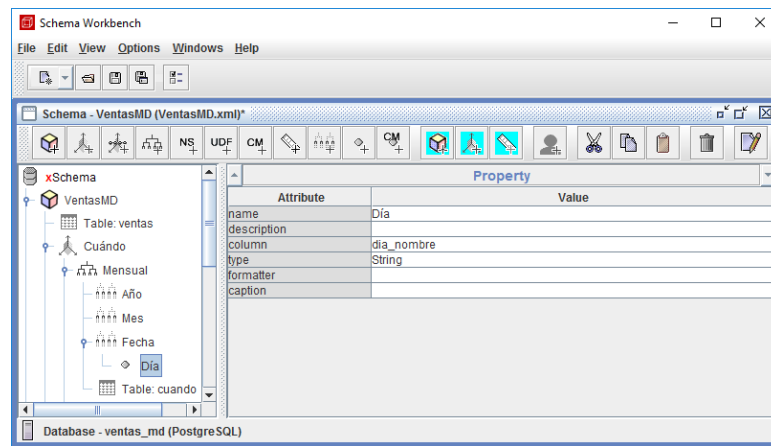


Figura 7: Definición de una propiedad.

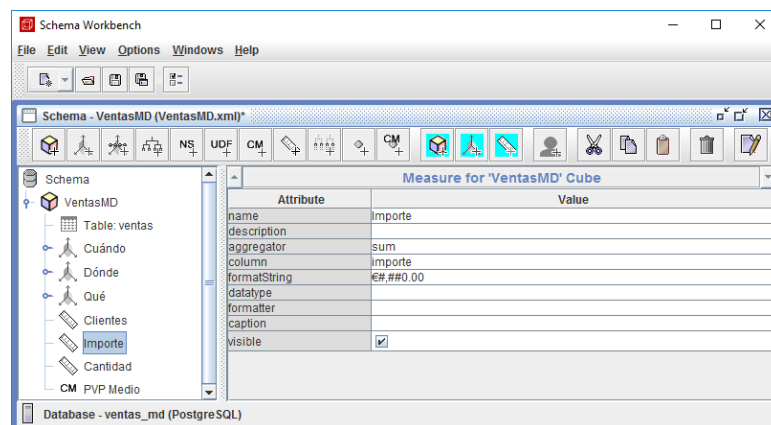


Figura 8: Definición de una medición.

### 2.4.1. Mediciones

Para definir una medición debemos dar valores a los atributos (figura 8):

- «name»: el nombre de la medición,
- «aggregator»: función de agregación, dependiendo de la aditividad de la medición,
- «column»: columna de la tabla de hechos asociada,
- «formatString»: opcionalmente podemos definir el formato del número, en particular el número de decimales o el indicador de moneda.

### 2.4.2. Mediciones calculadas

Para definir una medición calculada debemos dar valores a los atributos (figura 9):

- «name»: el nombre de la medición calculada,
- «dimension»: se pueden definir miembros calculados para cualquier dimensión, las mediciones se tratan como una dimensión especial, para definir una medición calculada deberemos seleccionar «Measures»,
- «formula»: expresión MDX para obtener el resultado,
- «formatString»: opcionalmente podemos definir el formato del resultado.



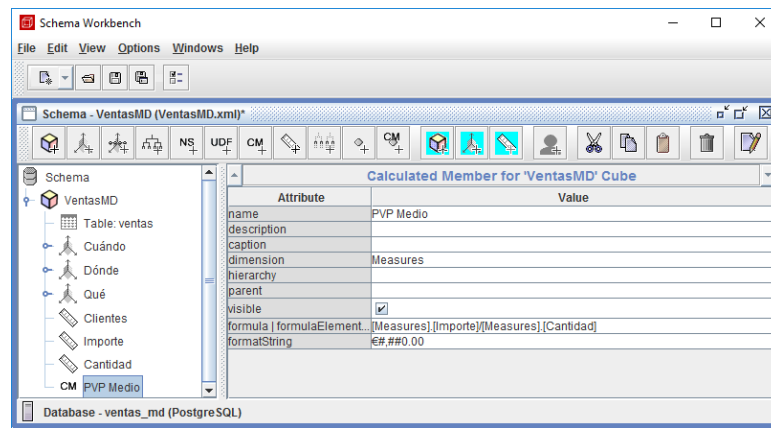
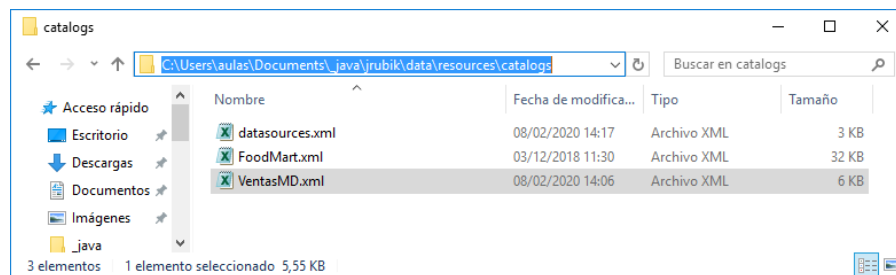


Figura 9: Definición de una medición calculada.

Figura 10: Carpeta de catálogos de *JRubik*.

## 2.5. Operaciones sobre el archivo XML

Pulsando sobre el icono «Edit Mode» nos muestra el fragmento XML generado mediante el proceso de definición llevado a cabo para el elemento actualmente seleccionado. El resultado en XML no podemos editarlo desde la ventana que lo muestra.

Adicionalmente incorpora iconos para las operaciones «Copy» y «Paste» pero son aplicables solo a valores de los atributos definidos, no para elementos completos, ni para el archivo XML.

Se puede dar la situación en la que tengamos jerarquías muy parecidas dentro de una dimensión o incluso dimensiones similares. Para agilizar su definición, podemos editar el archivo XML con uno de los editores disponibles (por ejemplo, *Microsoft Visual Studio 2010*, *XML Copy Editor* o *XML Notepad 2007*). Para ello, deberemos salvar y cerrar el esquema en *Schema Workbench*, editarlo con el editor que elijamos y volverlo a abrir con *Schema Workbench* para comprobar que está todo correcto.

## 3. Definición de consultas OLAP

Vamos a definir consultas OLAP mediante la herramienta *JRubik*. Una vez tenemos los datos en la BD y hemos definido el esquema multidimensional en el archivo XML, *JRubik* puede configurar *Mondrian* con el archivo del esquema para acceder a la BD. *JRubik* se encarga de obtener la estructura de datos de acuerdo al esquema, generar las consultas MDX y presentar los resultados.

Esta herramienta ofrece muchas posibilidades; aquí nos vamos a centrar en las básicas para mostrar el funcionamiento de *Mondrian* con nuestro esquema multidimensional sobre la BD relacional.

### 3.1. Configuración de *JRubik*

En primer lugar, deberemos situar el archivo XML del esquema multidimensional en la carpeta de catálogos de *JRubik*, `_java/jrubik/data/resources/catalogs` (figura 10).

Pulsando sobre [«Archivo», «Configuración»] podemos configurar diversos aspectos de la aplicación, en particular, las fuentes de datos. Pulsamos sobre «Configuración de Datasources», nos aparecen

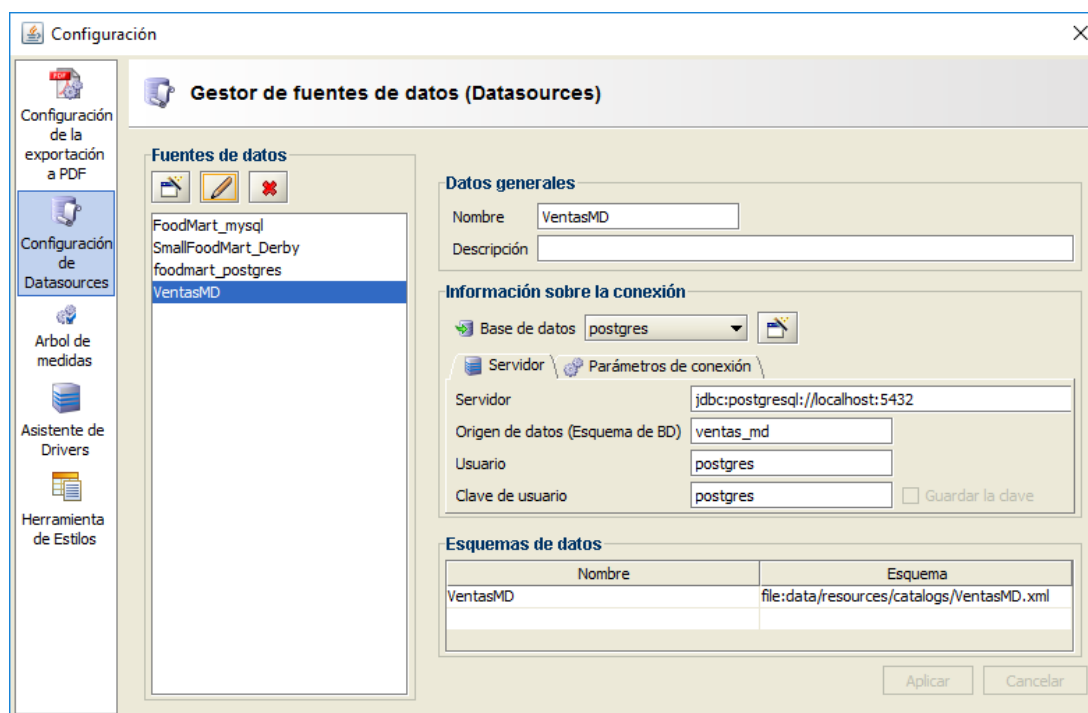


Figura 11: Definición de una fuente de datos.

las fuentes de datos definidas. Para definir una nueva fuente de datos, en el apartado «Fuentes de datos» (figura 11), pulsamos sobre el icono «Crea una nueva fuente de datos» y, en la zona de la derecha de la ventana de la nueva fuente, completamos adecuadamente los datos (siguiendo el ejemplo de otras fuentes que incluye).

Le asignamos un nombre a la fuente. Para la conexión, elegimos el tipo de servidor de la BD y el servidor (para *PostgreSQL* es `jdbc:postgresql://localhost:5432`); adicionalmente indicamos el nombre de la BD el usuario y la clave. Para indicar el esquema, le asignamos un nombre e indicamos su ubicación (`file:data/resources/catalogs/VentasMD.xml`).

Con estos elementos ya hemos definido la conexión con *Mondrian* y la BD utilizando el esquema XML definido.

### 3.2. Construcción de tablas

Para definir una consulta OLAP, primero deberemos construir una tabla multidimensional desde la pestaña «Descubridor»<sup>3</sup> (figura 12).

Desplegamos el árbol de la conexión con la que queramos trabajar y «pulsamos-arrastramos-soltamos» el elemento que queramos representar sobre la tabla de la parte inferior: columnas, filas o filtro.

Sobre las filas o columnas se pueden situar dimensiones o jerarquías (las mediciones están definidas como una dimensión más, con una jerarquía con un solo nivel). Sobre el filtro sólo se pueden situar instancias de las mediciones o de las dimensiones (las mediciones están definidas como instancias de un nivel llamado «MeasuresLevel»). Para obtener el resultado pulsamos sobre el botón «Aplicar» (debajo de la tabla de definición); para definir una nueva tabla pulsamos sobre el botón «Limpiar».

El resultado obtenido se muestra en la parte de la derecha de la ventana: por una parte, en la zona superior, la tabla de resultados cuyos elementos se pueden desplegar y, por otra, en la zona inferior, la sentencia MDX generada mediante la definición de la tabla (esta sentencia es la que ha utilizado *Mondrian* para obtener el resultado). Al interaccionar con los datos de la tabla, la sentencia MDX se adapta adecuadamente. Lo incluido en el filtro se corresponde con la parte **where** de la sentencia generada.

<sup>3</sup>Las pestañas están situadas verticalmente en la parte central de la ventana

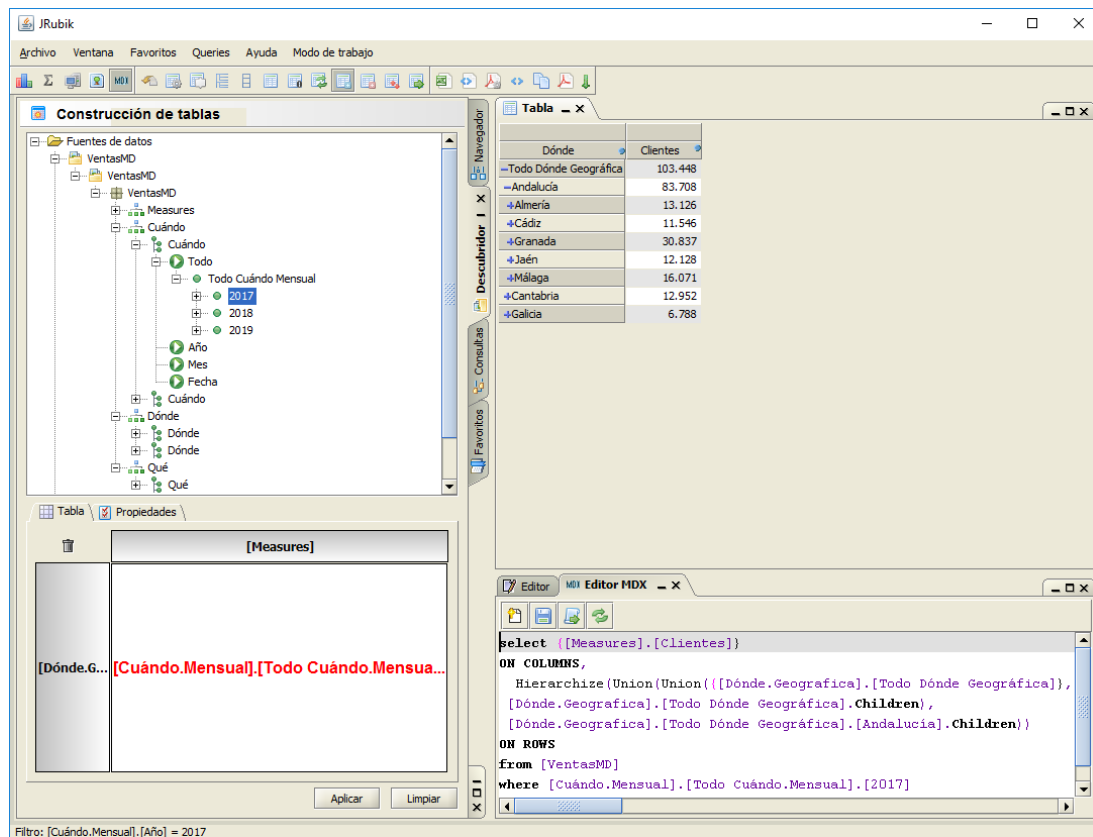


Figura 12: Construcción de tablas.

Adicionalmente, podemos ver el resultado en forma gráfica pulsando sobre el icono de la barra de herramientas «Muestra/oculta el Gráfico» (el primero de la izquierda).

### 3.3. Refinar la construcción de tablas

Podemos refinar la construcción de la tabla desde la pestaña «Navegador». Permite seleccionar elementos adicionales o bien elementos distintos de los componentes incluidos en las filas, columnas y filtro (figura 13). Para llevar a cabo los cambios definidos debemos pulsar sobre el botón «Aplicar» (en la barra de herramientas de la zona izquierda). Los cambios se reflejan tanto en la tabla como en la sentencia MDX de la parte derecha.

## 4. Operaciones a realizar

### 4.1. Definición del esquema multidimensional

1. Define el esquema multidimensional para *Mondrian* asociado a la BD *PostgreSQL* creada en la actividad **Herramientas ETL: PDI (Pentaho Data Integration)**.

### 4.2. Definición de consultas OLAP

2. Sobre *Mondrian*, utilizando el esquema multidimensional:
  - Obtén un informe inicial libre y explica su contenido.
  - Mediante Drill-down obtén un nuevo informe y explica su contenido.
  - Mediante Slice&Dice obtén un nuevo informe y explica su contenido.
  - Mediante Roll-up obtén un nuevo informe y explica su contenido.

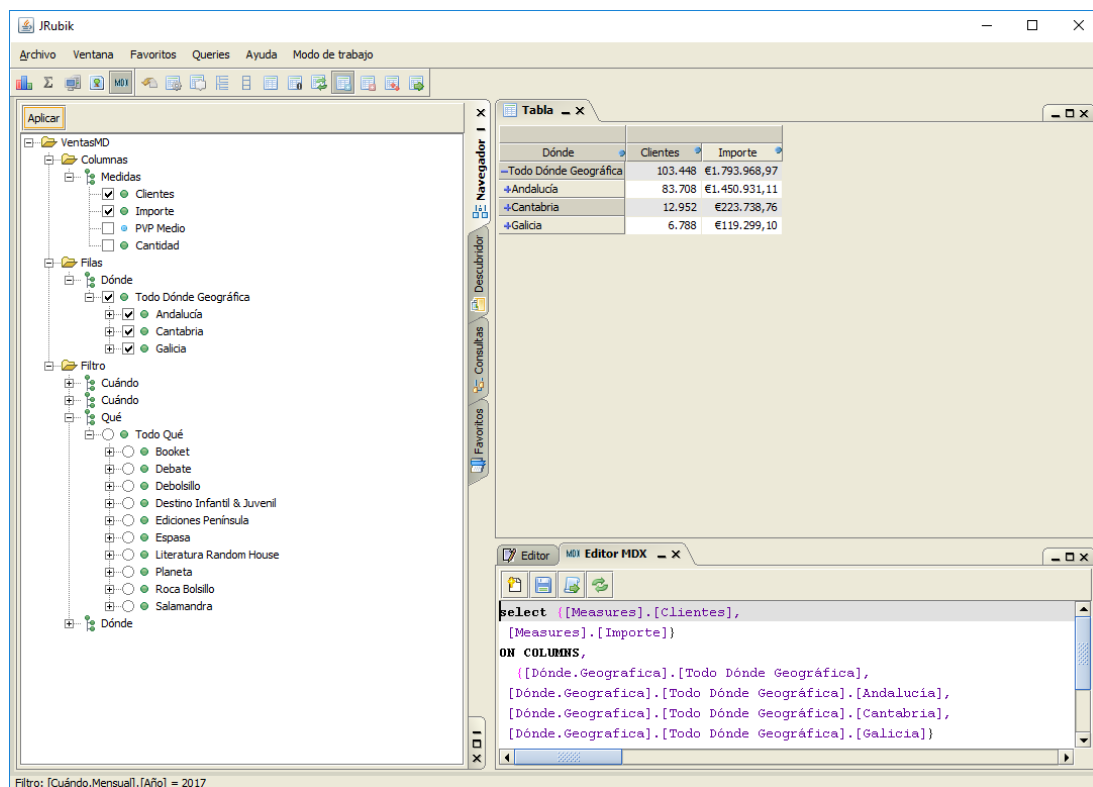


Figura 13: Refinar la construcción de tablas.

## Bibliografía

- [BGH14] William D. Back, Nicholas Goodman, and Julian Hyde. *Mondrian in Action. Open Source Business Analytics*. Manning, 2014.
- [Hyd09] Julian Hyde. *Mondrian 3.0.4 Technical Guide. Developing OLAP solutions with Mondrian/JasperAnalysis*. Mondrian, 2009.