
1. Presentación de la realidad

1.1. Realidad planteada

En este caso de estudio , vamos a plantear lo que podría ser parte de un diseño de DW en una empresa corriente , que tiene ventas (en este caso en otra moneda distinta a la de su país) y le interesa sacar un reporte de las venta en su moneda por mes. La idea es plantear un diseño de DW muy sencillo ya que el objetivo no es el diseño en sí , sino demostrar la utilidad de la herramienta sobre los posibles cambios que pueden ocurrir.

El caso de estudio trata de una empresa (supongamos una de ventas de artículos de computadoras) que vende en dólares. Cada venta tiene la descripción del producto , la cantidad , el precio unitario en dólares , la fecha de la venta , el identificador de venta y el importe total de la misma. Esta empresa mantiene un histórico de sus ventas, por lo que cada 6 meses necesita saber el resultado en pesos por mes y por producto de las ventas de la empresa (recordemos que acá se está planteando un caso simplificado de la realidad , ya que así planteado no tiene mucho sentido práctico , pero en la realidad esto puede tener sentido si además sacamos los gastos de la empresa por mes y producto , así podríamos calcular la ganancia por mes $Ganancia = Venta - Gastos$). Para cumplir el objetivo pedido , el ingeniero de la empresa decide utilizar la herramienta DWD para diseñar un DW. Los datos que tiene en la base fuente para aplicar el diseño del DW son : ventas , cotización (del dólar por fecha) y a los efectos de este caso de estudio supondremos que se tienen los datos de fechas (esto en realidad no se encuentra en una base fuente , sino que se obtiene en el diseño del DW).

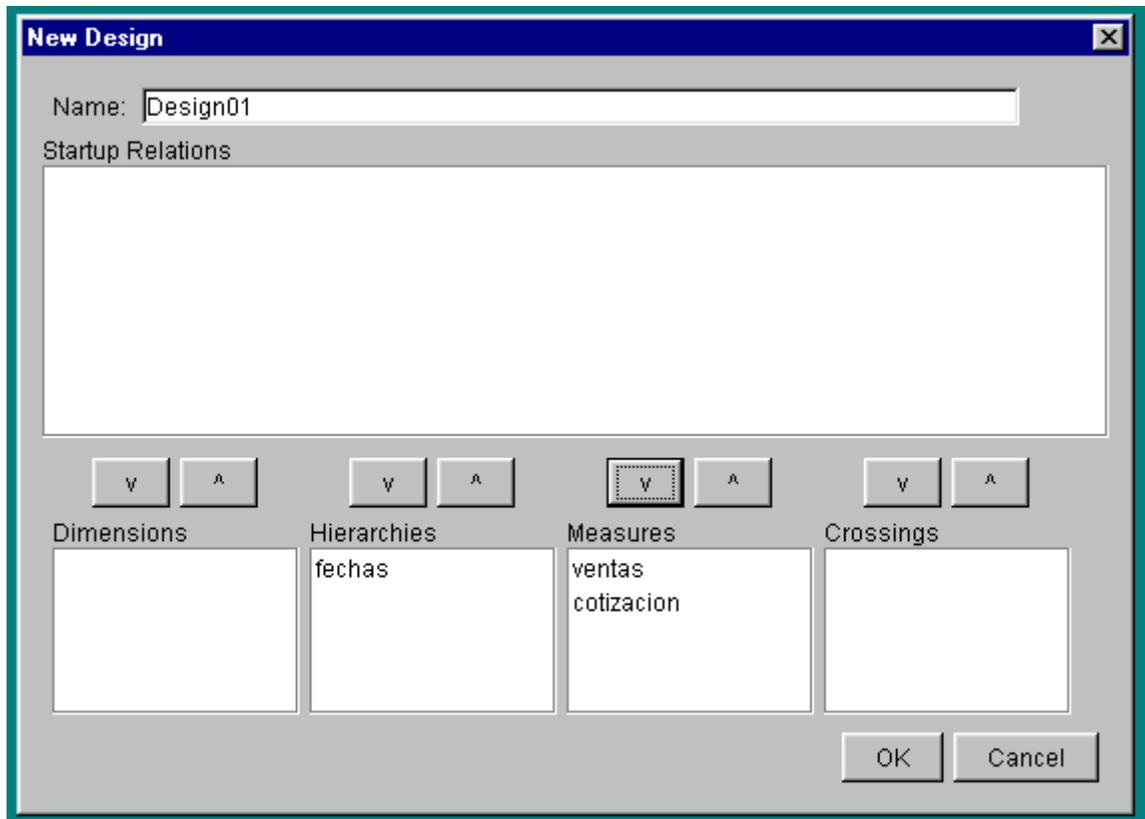
1.2. Pasos seguidos para llegar al diseño del DW

Los relaciones que se disponen en la base fuente son :

- Fechas (fecha,mes,anio)
- Ventas
(id_venta,fecha,importe_venta,cantidad,precio_unitario,id_producto)
- Cotización (fecha,venta_pizarra)

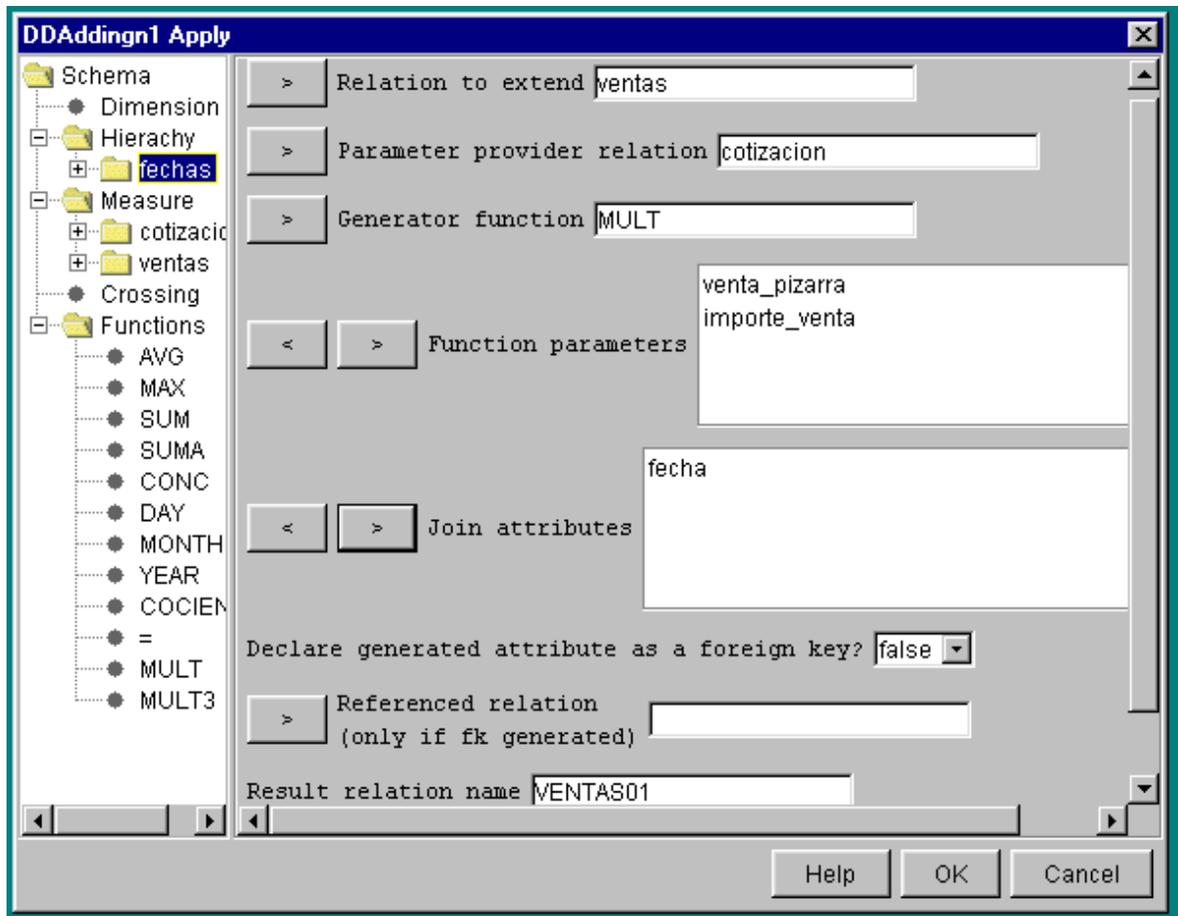


1) Para realizar el diseño , primero se debe inicializar el DW , se inicializa de la siguiente manera



Se ingresa al sistema la relación de fechas como jerarquía y las demás de medidas.

2) Se aplica la primitiva DDadding-N1 entre la tabla cotización y ventas ,para obtener Ventas01 , que tendrá un campo Importe_pesos calculado , que es el importe_venta x venta_pizarra.

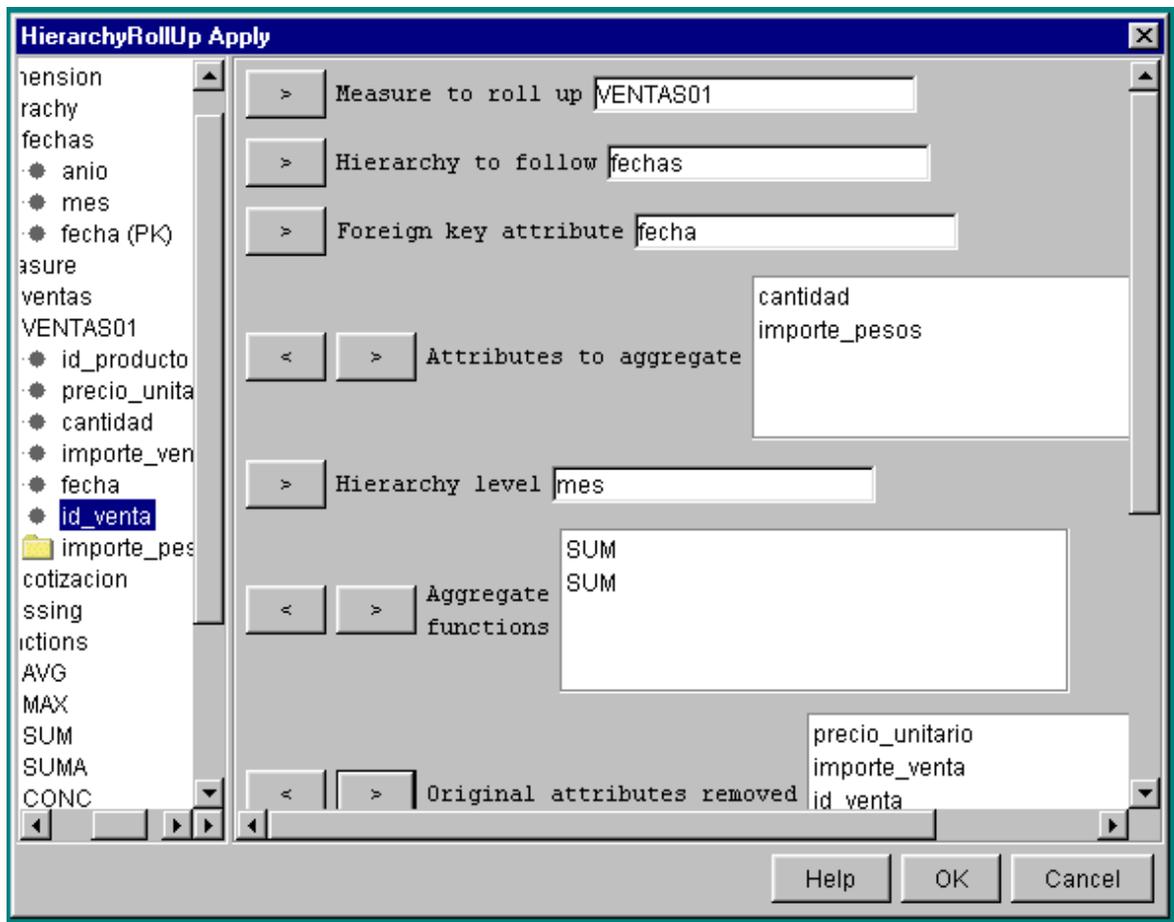


Donde $MULT(x, y) = x * y$.
 Resultado:

- Ventas01 (id_venta, fecha, importe_venta,cantidad, precio_unitario, id_producto, importe_pesos)

Con $importe_pesos = ventas.importe_venta * cotización.venta_pizarra$

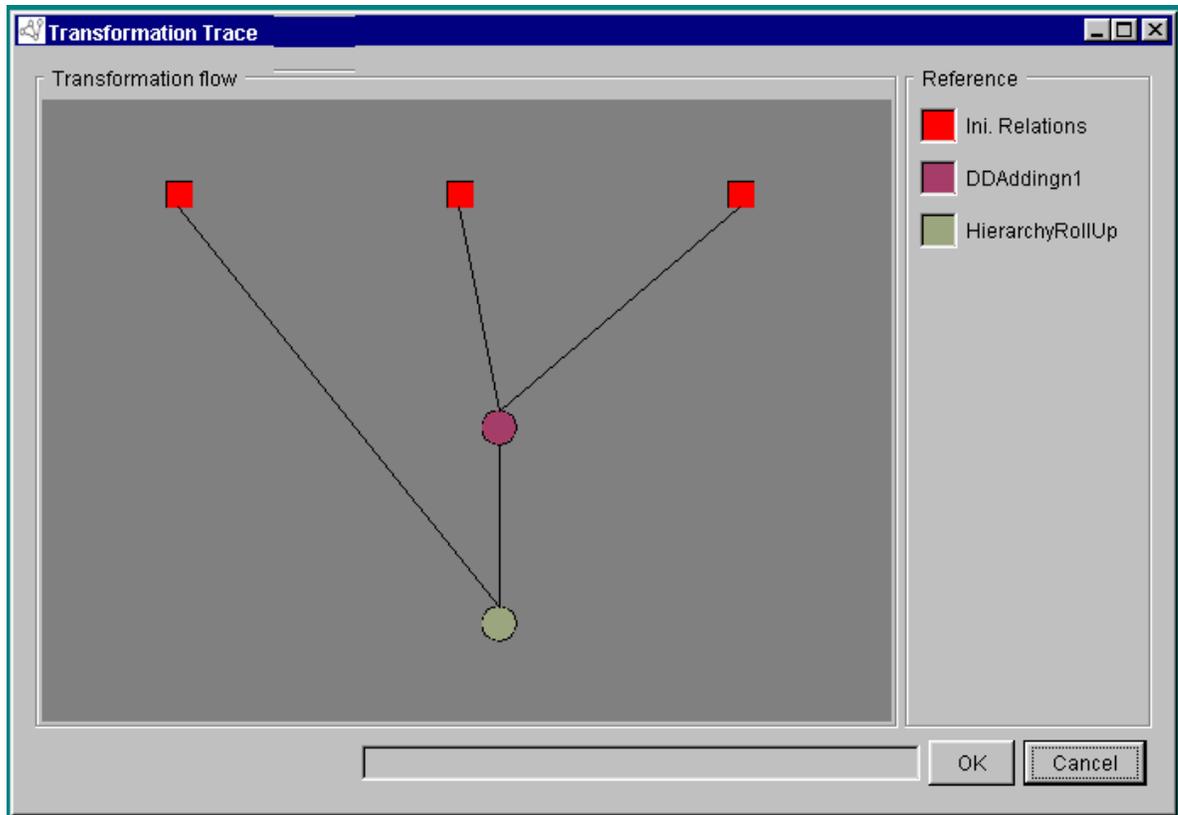
3) Para finalmente obtener las ventas en pesos por mes y por producto se aplica la Herarcy Roll UP de la siguiente forma :



Resultado :

- VentasxMes (Mes,id_producto,Cantidad,importe_pesos)

En este punto se logró el objetivo que se buscaba , de tener las ventas en pesos por mes y por producto. La traza de primitivas aplicadas para este diseño es :



1.3. Análisis de la base fuente

Después de pasado un tiempo en el que se han sacado los reportes pedidos sobre el DW generado a partir del diseño anterior, se decide hacer un análisis sobre el esquema de la base de datos fuentes (ya que se supone que había relaciones no normalizadas) por el ingeniero a cargo. Al evaluar el esquema nota que la relación ventas tiene redundancia , ya que en ella aparece el atributo importe_ventas , que es el precio_unitario * cantidad , así que decide eliminar el atributo importe_ventas.

Al eliminar el atributo , se encuentra con que forma parte del diseño del DW, así que se hace necesario para implementar el cambio una herramienta para la evolución de un DW debido a cambios en el esquema fuente. Sin embargo el diseño del DW no debería cambiar ya que el atributo a eliminar se puede obtener a partir de los otros en la relación.

La opción es utilizar DWE , una herramienta para manejar la evolución de un DW (generado por ejemplo con DWD) a partir de la evolución del esquema fuente.

1.4. Requerimientos

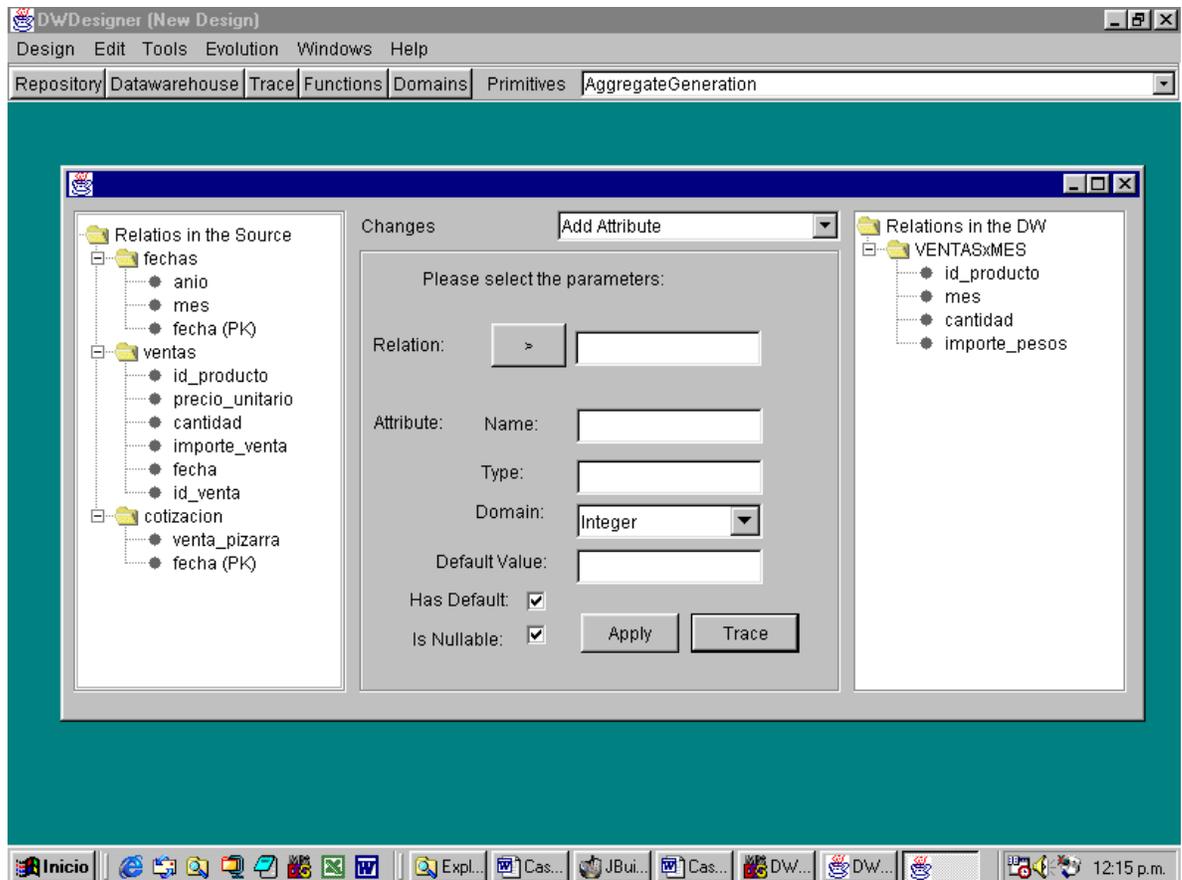
Los nuevos requerimientos son eliminar el atributo redundante de la tabla de ventas (evolución del DW) y obtener un nuevo diseño que permita seguir cumpliendo los requisitos anteriores pedidos por la empresa. Estos son los de obtener los reportes de ventas en pesos por mes y productos.

2. DWE y visualización

2.1. Primer paso

El primer paso es el de cargar el diseño del DW que se le va a aplicar el cambio en el esquema fuente (este diseño lo levantamos de archivo , son los archivos con extensiones dwe) , Para el caso de estudio es el archivo caso1.dwe. Para levantar el diseño hay que ir a File - > Open y seleccionar el archivo desado (caso1.dwe).

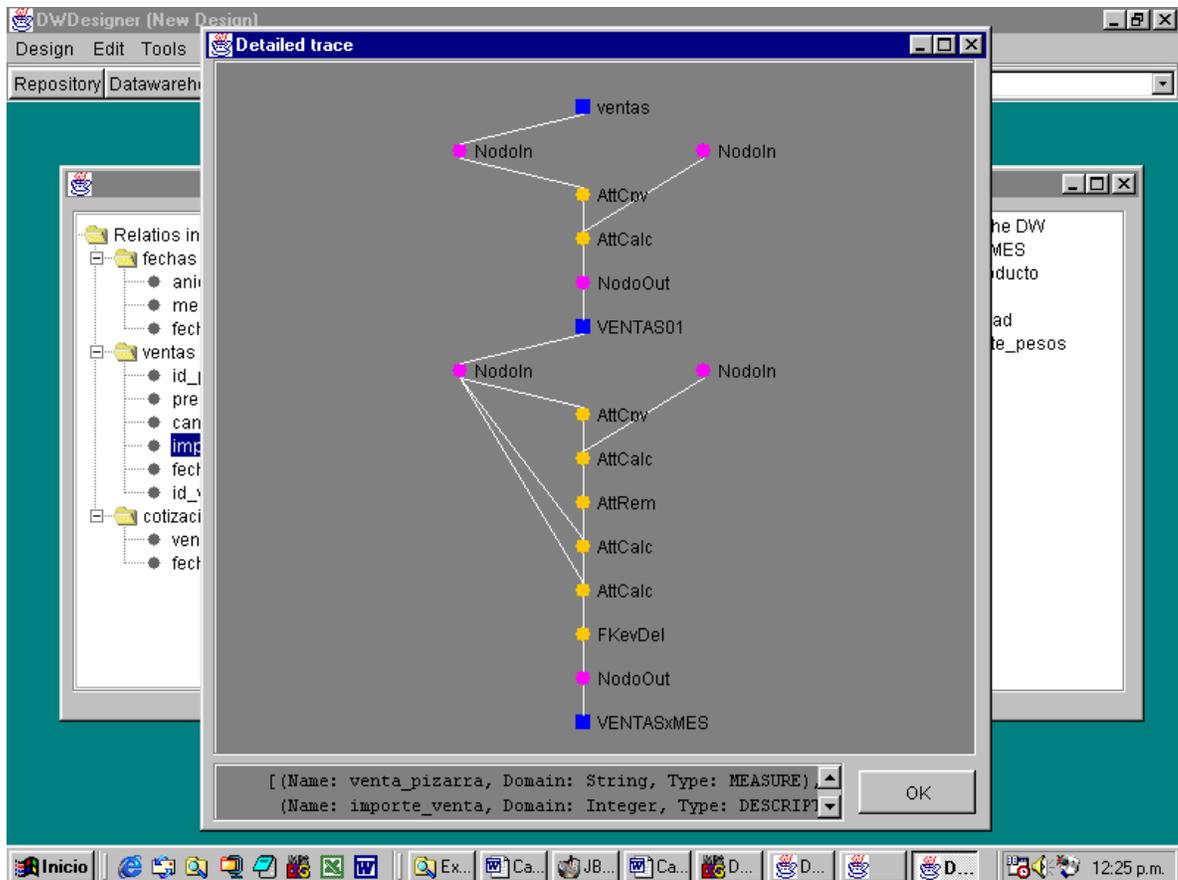
Luego de levantarlo debemos seleccionar la parte de evolución (traer la herramienta de evolución con la cual vamos a trabajar.



2.2. Segundo paso

Ahora vamos a seleccionar la relación de ventas y ver la traza detallada. La traza detallada como se explicó es el grafo de operaciones básicas que tiene cada primitiva según se especifica en la tesis.

La idea de visualizar la traza detallada es para ver en forma gráfica como dependen los atributos de la relación en la cual se van a aplicar los cambios con respecto al DW. A pesar de que el sistema al aplicar un cambio nos avisa de las dependencias es bueno poder visualizarlo para preverlo de antemano.

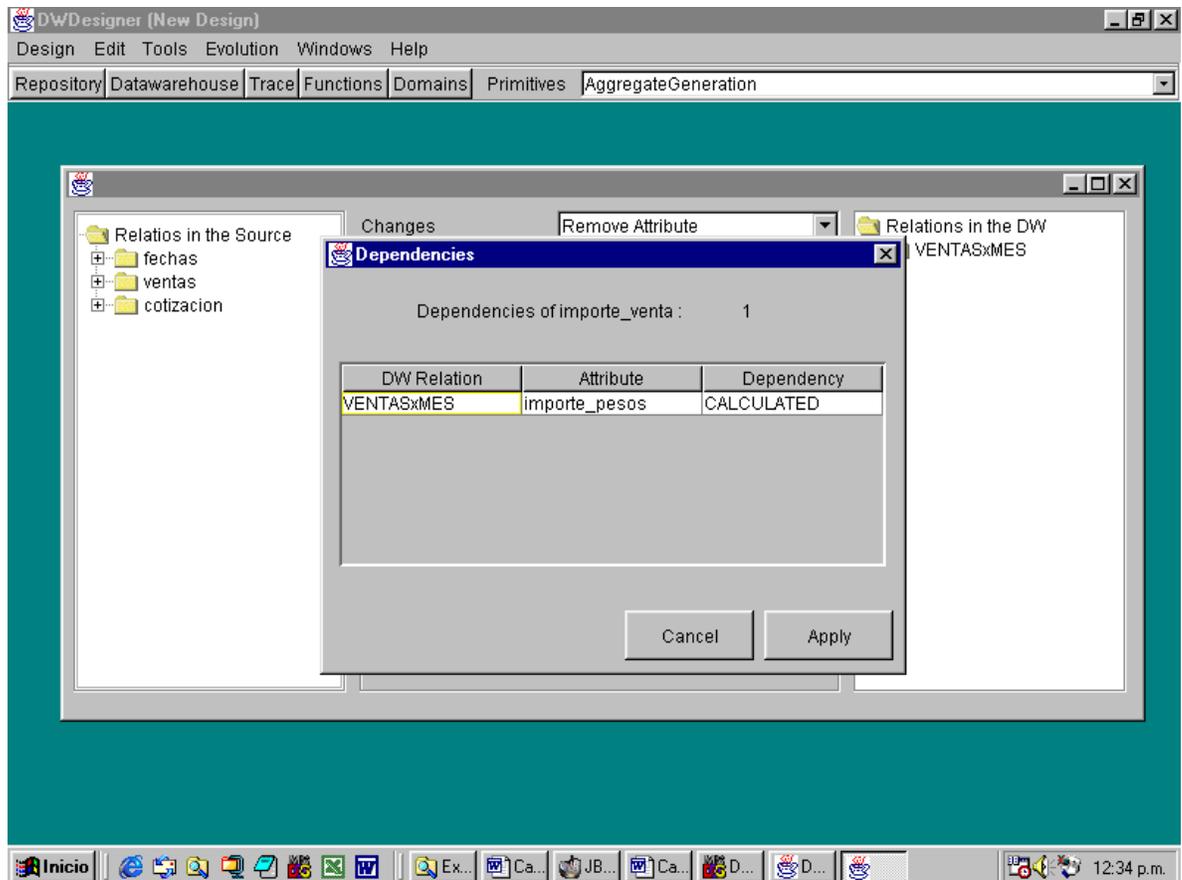


En este gráfico vemos las componentes (op. Básicas) de las primitivas y también podemos ver como de la relación ventas hay atributos que son de dependencia copiados y otros calculados. En particular el que nos interesa el atributo importe_ventas (que es el que se quiere eliminar) . Se observa que este atributo es usado en la función de calculo para obtener el atributo ventas01.imorte_pesos , por lo que se espera que cuando se elimine , nos de cómo dependencia Calculado.

3. Aplicación del cambio

3.1. Eliminando el atributo y viendo dependencia

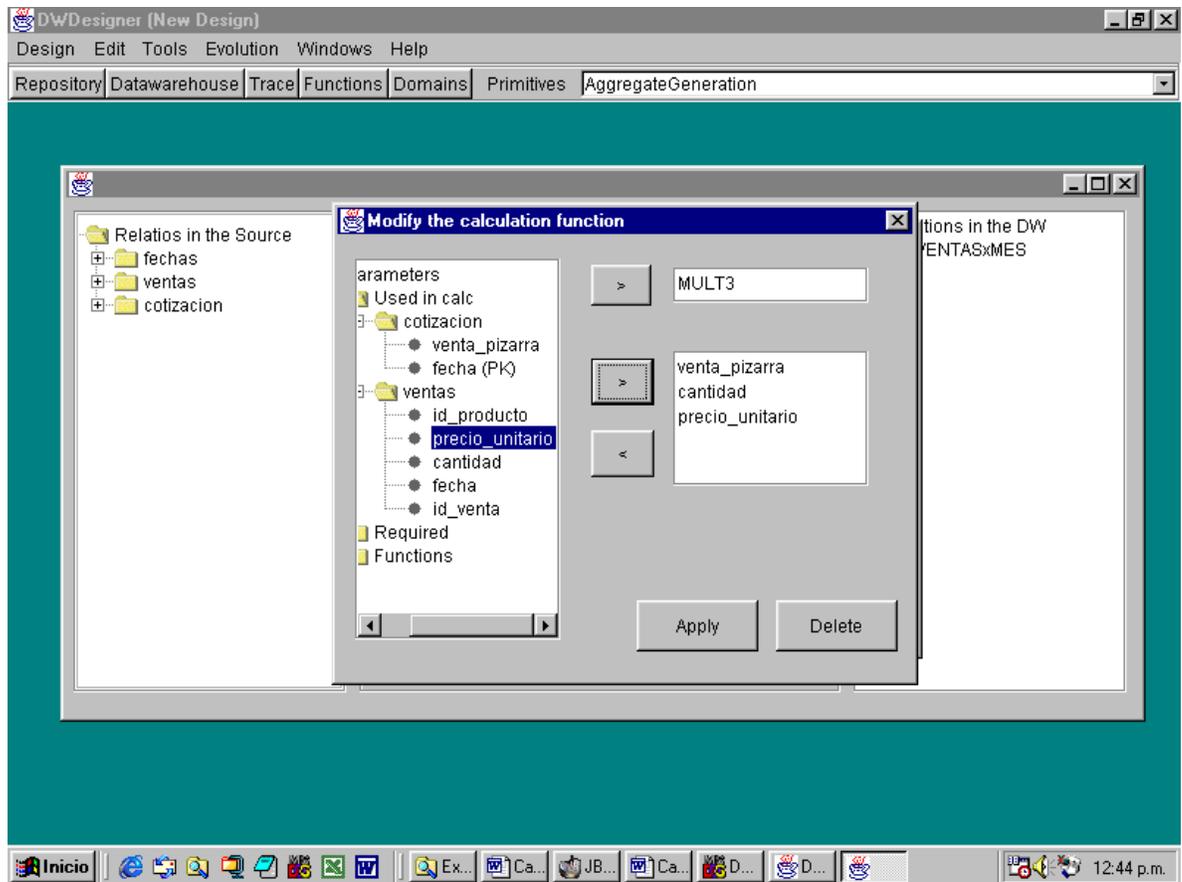
Ahora seleccionamos el cambio Remove Attribute (remover atributo) en la herramienta y seleccionamos el atributo a remover (en este caso importe_venta), luego damos Apply para indicar que efectivamente queremos aplicar el cambio. Al dar Apply el sistema nos avisa de las dependencias de ese atributo respecto al DW.



Acá el sistema nos avisa que hay un atributo en el DW en la relación VentasxMes llamado importe_pesos que depende del atributo que queremos eliminar en forma calculada (que era lo que esperábamos al ver la traza a nivel de operaciones básicas). Ahora volvemos a dar Apply para seguir con el cambio.

3.2. Cambio en la función de cálculo

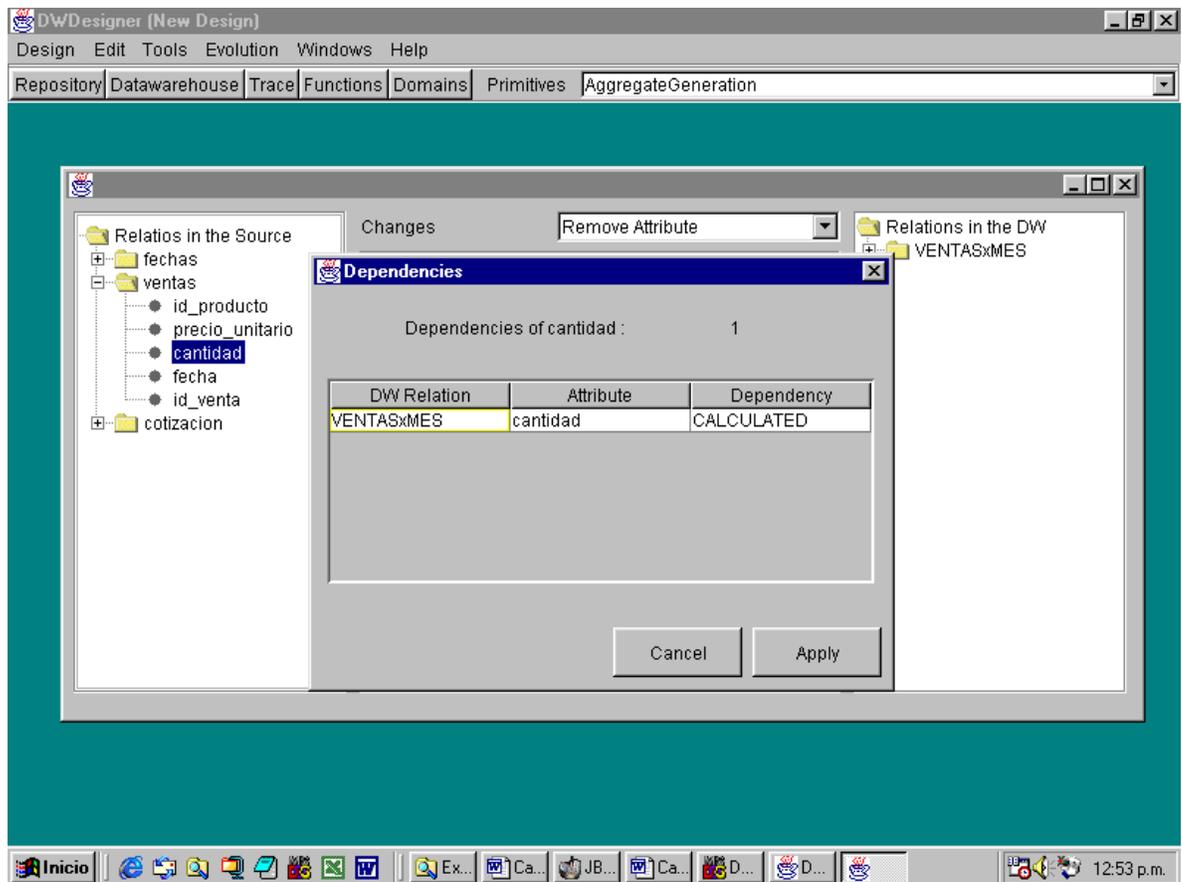
La herramienta cuando va propagando los cambios , detecta que hay un atributo importe_pesos que es calculado a partir de uno de los atributos que estamos eliminando , así que nos pregunta si el atributo importe_pesos lo eliminamos o lo calculamos a partir de otros atributos (cambio en la función de cálculo). La idea acá es cambiar importe_pesos que antes era $MULT(\text{venta_pizarra}, \text{importe_venta})$ cambiarla por $MULT3(\text{venta_pizarra}, \text{cantidad}, \text{precio_Unitario})$. Donde $MULT3(x, y, z) = x * y * z$.



Luego de cambiar la función de cálculo , la herramienta termina de eliminar el atributo según las reglas especificadas en la tesis y cambia la función de cálculo de importe_pesos a la nueva función.

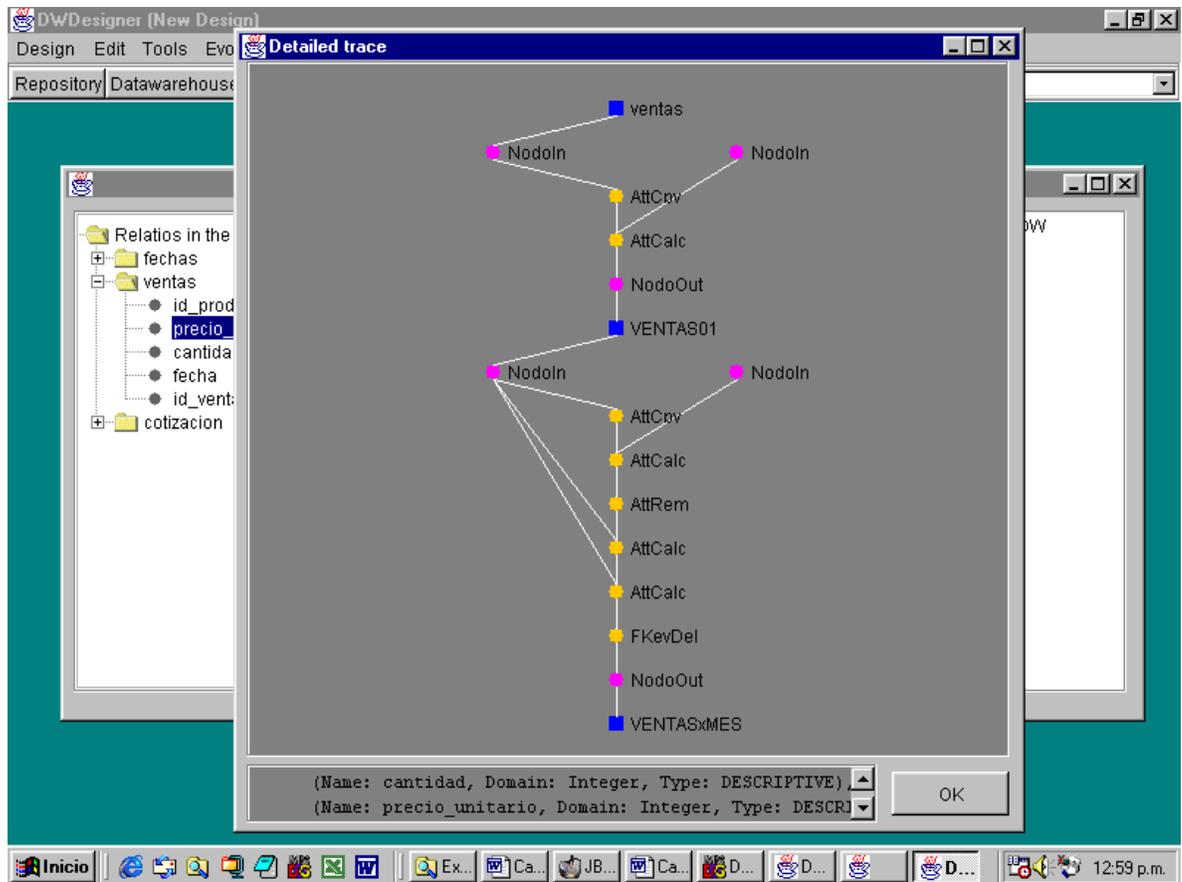
3.3. Verificación de dependencias

Ahora observamos que en la relación fuente el atributo que queríamos eliminar , efectivamente se eliminó , nos queda ver las dependencias de los nuevos atributos de la tabla ventas con respecto a la del DW. Lo que esperamos es que como antes para el atributo importe_venta daba dependencia de calculado , ahora para los atributos cantidad y precio_unitario de calculado (ya que se usan porque se cambio la función de cálculo al remover el atributo para importe_pesos). Como observación podemos decir que las dependencias para el atributo precio_unitario antes era NO DEPENDENCY (ya que no había ninguna con respecto al DW) , y para cantidad era CALCULATED (ya que se usaba en HRUP para la agregación)



3.4. Comparación de traza detallada

Podemos visualizar la nueva traza detallada , vemos que a nivel de las operaciones básicas , esta no cambió, pero si miramos sus atributos , vemos que para el calculo de ventas01.importe_pesos de AttCalc , ahora se calcula a partir de venta_pizarra, cantidad y precio_unitario. Que era lo que efectivamente se quería.



4. Conclusiones

4.1. Evaluación de resultados

Evaluamos ahora los resultados obtenidos, y como estos resuelven los requerimientos planteados al comienzo de este caso de estudio.

- A nivel de Esquema fuente , vemos que el atributo se ha eliminado de la relación fuente
- A nivel de parámetros , acá observamos que el cambio que deseábamos hacer se logró , que era calcular el importe_pesos a partir de cantidad y precio unitario
- A nivel de dependencias, luego de hallar las dependencias nuevas aparecen las esperadas para el caso de ventas.cantidad y ventas.precio_unitario
- A nivel de primitivas , el diseño se mantiene
- A nivel de data warehouse , acá se sigue siendo el mismo al que se tenía antes del cambio en el esquema fuente (esto es justamente lo que se quiere lograr , que tras un cambio en el esquema fuente , este afecte lo menos posible el DW).