4

Toma de Decisiones

¿Pan o cereal?, ¿Jugo o leche?, ¿Corbata azul o verde?, ¿Carro o taxi? Son sólo algunas de las miles de decisiones que como ser humanos tomamos cada día en nuestra vida cotidiana. Muchas de estas las hacemos de manera irracional y otras tantas de manera analítica, pero lo que es común en todas es que, sin importar lo bueno o malo de nuestra elección, ésta influirá a lo largo del día, de la semana o quizá de nuestra vida.

Si lo anterior fuera extrapolado a una empresa, en la que día a día se toman decisiones que trazan el éxito o fracaso de esta en el mercado, la manera "irracional" queda inmediatamente descartada para este proceso, por el contrario, se requiere de un análisis basado en información detallada que ayude a justificar esa *última palabra* y quedar con la certeza de que fue la mejor decisión.

Para el caso de estudio de este trabajo, la mayoría de las decisiones se basan en el área de ventas: Se produce bajo demanda. Sin embargo, debido a que se trata de una pequeña empresa, con un número reducido de trabajadores dedicados a producción y una amplia demanda de uniformes, la producción para la temporada alta (julio-septiembre) se comienza hasta con medio año de anticipación, y debido a esto, no se tiene información precisa acerca de la demanda real de cierto tipo de uniforme por lo que algunos uniformes se agotan rápidamente mientras otros no son vendidos con la misma rapidez. Además de que, en los demás meses del año los uniformes siguen siendo requeridos. Algunas de las cuestiones importantes para la toma de decisiones son las siguientes:

- ¿Qué tallas son las más vendidas?
- ¿Qué tipo de uniforme es el más demandado?
- ¿En qué épocas del año crece y/o decrece la demanda de cierto uniforme?
- ¿En qué escuela se tiene un mayor consumismo?

4.1 Necesidad de un Cubo OLAP

Los sistemas OLTP se encuentran orientados a transacciones que soporten las principales tareas de negocio que se requieren a diario en donde las principales operaciones que se usan son: `INSERT`, `DELETE`, `UPDATE` y son concretadas una vez que se hace `commit` o bien abortadas cuando se hace `rollback`. Este tipo de sistemas no son muy útiles para llevar a cabo las tareas de análisis de datos, debido a que el modelo de datos se encuentra altamente normalizado, generalmente no

cuentan con un registro histórico y la ejecución de consultas que involucren grandes volúmenes de datos podrían entorpecer las transacciones.

Los sistemas OLAP (On-Line Analytical Processing) se encuentran orientados al procesamiento analítico, esto es, que su objetivo principal es la lectura de grandes volúmenes de datos que resulten útiles al usuario y así, con la información recabada, encaminarlo hacia la toma de decisiones inteligente.

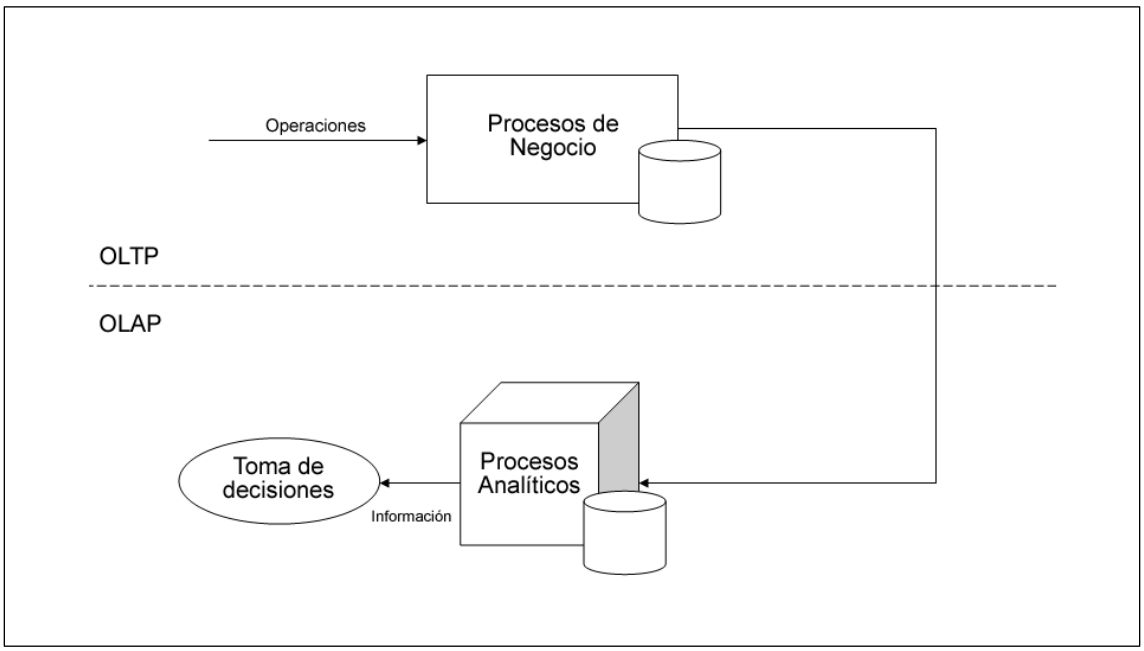

Figura 4.1: Sistemas OLTP y OLAP

Este tipo de sistemas, se basan en cubos, llamados así por la característica multidimensional de la base de datos que es utilizada para su construcción.

Un cubo OLAP, representa un conjunto de hechos relacionados con un área específica de negocio y cuya información se encuentra ordenada en vectores de `n` dimensiones (generalmente tres). La dimensión es la forma en que el tema objetivo es separado para su análisis, y generalmente responde a las preguntas de ¿Cuándo?, ¿Qué?, ¿Dónde?, etcétera. La intersección de las dimensiones constituyen un hecho.

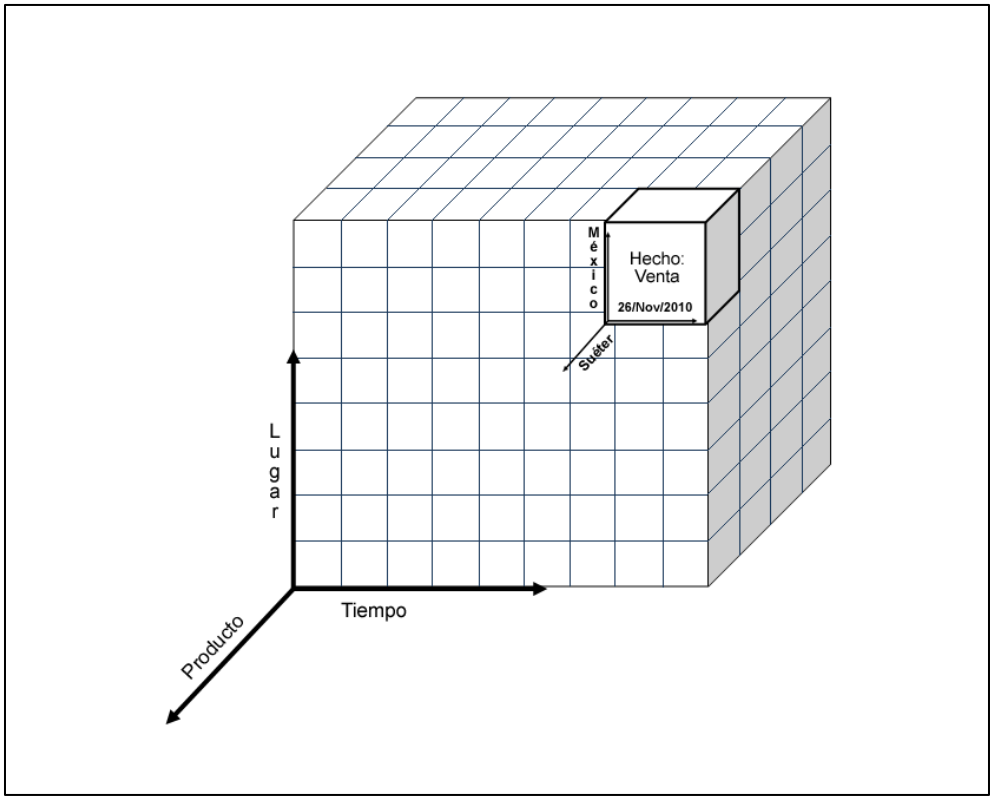

Figura 4.2 Visualización multidimensional del hecho venta

4.2 Componentes y diseño

Antes de diseñar un cubo es importante considerar los diferentes componentes que forman parte de éste:

- Tabla de hechos: Es la tabla central que recoge la información de las dimensiones por medio de las claves foráneas que formarán la llave primaria de esta. Adicionalmente deberá contener atributos, idealmente numéricos, que representen la información de cada hecho. Estos atributos son denominados medidas.
- Tablas de dimensiones: Estas tablas contienen el detalle de cada dimensión útiles para describir los hechos almacenados en el cubo.

Existen 2 maneras de diseñar una base de datos OLAP:

1. Basado en un esquema estrella: Se le llama así debido a que su estructura forma una estrella, con la tabla de hechos en el centro, y cada una de las dimensiones conectada a esta. Existe sólo una conexión por dimensión, no existen extensiones ni caminos alternativos entre estas.
2. Basado en un esquema copo de nieve: Es una variación del esquema anterior en el que las tablas dimensionales se encuentran normalizadas con la finalidad de eliminar la redundancia.

La decisión entre elegir uno y otro se debe basar en la cantidad de datos con que se trabajará y el diseño original de la aplicación OLTP tomando en cuenta las siguientes comparativas entre ambos esquemas:

Esquema estrella	**Esquema copo de Nieve**
Cada dimensión tiene un solo *join* hacia la tabla de hechos, por lo que las consultas se llevan a cabo de una manera más eficiente.	Incrementa el tiempo de ejecución de las consultas debido a la normalización de las dimensiones.
Este tipo de diagrama es más comprensible por su simplicidad en el diseño.	Generalmente un sistema OLAP nace del diseño OLTP en el que sus tablas se encuentran normalizadas, por lo que su implementación resulta más sencilla.
La migración de datos de OLTP a OLAP podría afectar el rendimiento del sistema OLTP si la transformación de datos resulta ser excesiva.	Las tablas dimensionales son pobladas de manera similar a como se encuentran en OLTP por lo que el rendimiento de este último no se vería muy afectado.

Tabla 4.1: Comparativa entre esquema estrella y copo de nieve

4.3 Implementación

Se pueden identificar 4 etapas principales en la construcción del almacén de datos:

1. Elegir el área de negocio para el cuál será construido
2. Decidir el hecho central
3. Identificar las distintas dimensiones

4. Elegir las medidas de negocio para la tabla de hechos.

Este cubo OLAP estará orientado al área de ventas, cuyo hecho principal es la venta de un uniforme formado por las dimensiones: Escuelas (¿Dónde?), Uniformes (¿Qué?) y Tiempo (¿Cuándo?)

Está diseñado bajo un modelo estrella para la simplificación de consultas OLAP, para obtener un tiempo óptimo respuesta y para que su diseño intuitivo sea más fácil de comprender para el usuario final.

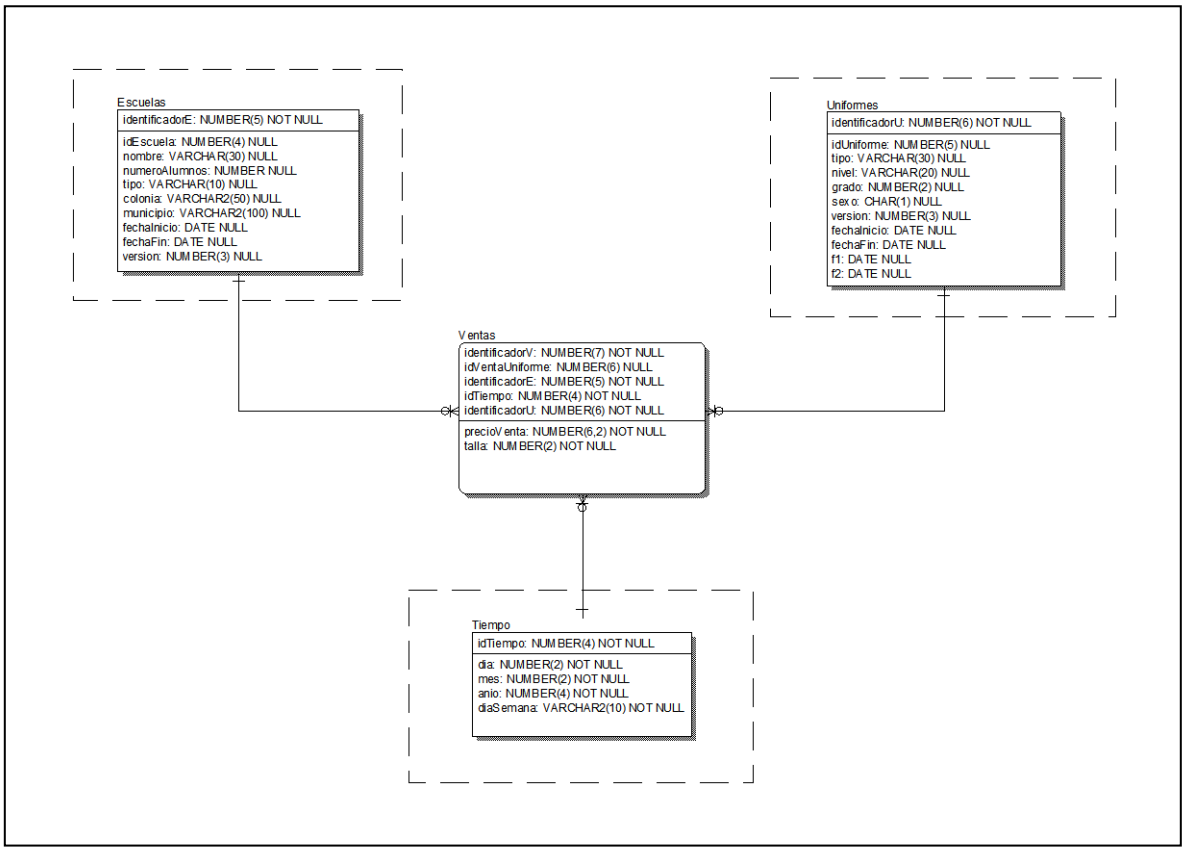

Figura 4.3: Diseño de cubo OLAP orientado a ventas

4.3.1 ETL (Extracción Transformación y carga)

Hasta el momento, sólo se ha diseñado la base de datos OLAP, sin embargo, esta no ha sido alimentada con los datos del negocio. Este proceso de migración de datos se le conoce como ETL (Extract, Transform, Load por sus siglas en inglés) y se divide en cada una de las actividades que conforma su nombre:

- Extracción: Comprende las tareas de obtener la información necesaria de las diferentes fuentes de datos que será integrada en el sistema OLAP, generalmente el origen de datos son sistemas OLTP, sin embargo, pueden provenir también de archivos de texto plano, hojas de cálculo e incluso otros sistemas OLAP.

- Transformación: En esta etapa se realizan las tareas que, independientemente del origen y formato de los datos, se convierten a un estado uniforme con un mismo formato defino. Algunas de estas tareas son:
  - Validación de datos: Se verifica que los datos extraídos cumplan con las restricciones correspondientes.
  - Limpieza de datos: Se realiza la corrección de datos erróneos o incompletos, y en caso de no contar con suficiente información de estos, puedes ser descartados y clasificados para darles tratamiento posterior.
  - Normalización de datos: Convierte los datos extraídos a valores más descriptivos y uniformes. La discretización (convertir un valor numérico a nominal) y numerización (convertir un valor nominal a numérico) son técnicas muy utilizadas en esta fase.
  - Agregación: En ocasiones, debido a las necesidades de negocio, es conveniente almacenar información agregada de ciertos valores numéricos, principalmente útil para la tabla de hechos.

- Carga: En esta etapa los datos que ya han sido extraídos y transformados de acuerdo a los requerimientos, son cargados en las diferentes tablas que forman las dimensiones y el hecho central.

4.3.2 Dimensiones de lenta variación (SCD)

En un sistema OLAP las dimensiones sufren cambios ocasionales a través del tiempo que deben ser actualizados en el sistema para una traducción real de hechos.

Para manejar esta actualización de datos existen tres formas principales de registrarlos en la base de datos de acuerdo a las necesidades de negocio:

1. SCD Tipo 1 – Sobreescritura: Cuando se detecta un cambio en alguno de los atributos de la dimensión, este se sobreescribe y no se guardan sus valores históricos.

id_subrogado	id_negocio	columna_1	columna_2
2	5	valor_1	valor_2

id_subrogado	id_negocio	columna_1	columna_2
2	5	valor_3	valor_2

   Para que se puedan gestionar de manera correcta las actualizaciones de los datos, se requiere de dos claves en la tabla de la dimensión: El valor original de la clave primaria ubicada en el origen de datos y una clave subrogada que identifique la fila en el sistema OLAP. Esto con el fin de que se logren detectar los cambios por medio del identificador original y a través de la clave sustituta poblar los hechos de la tabla central.

2. SCD Tipo 2 – Nueva fila: Si se detecta un cambio en algún valor se crea una nueva fila con la actualización del atributo y un nuevo id subrogado, conservándose el valor anterior en una fila diferente.

id_subrogado	id_negocio	columna_1	columna_2	fecha_1	fecha_2	version
2	5	valor_1	valor_2	26/11/2010	NULL	1

id_subrogado	id_negocio	columna_1	columna_2	fecha_1	fecha_2	version
2	5	valor_1	valor_2	26/11/2010	24/12/2010	1
3	5	valor_3	valor_2	24/12/2010	NULL	2

Es importante considerar que el histórico se lleva a cabo por medio de 3 columnas extra: la fecha en que se dio de alta el registro, la fecha en que uno nuevo lo sustituyó, y el número de versión de ese registro.

3. SCD Tipo 3 – Nueva Columna: En esta estrategia se requiere una columna extra por cada atributo que se desee mantener su historial. La nueva columna almacenará el valor anterior del atributo en caso de ser actualizado.

id_subrogado	id_negocio	columna_1	columna_1_anterior	columna_2
2	5	-----	valor_1	valor_2

id_subrogado	id_negocio	columna_1	columna_1_anterior	columna_2
2	5	valor_1	valor_3	valor_2

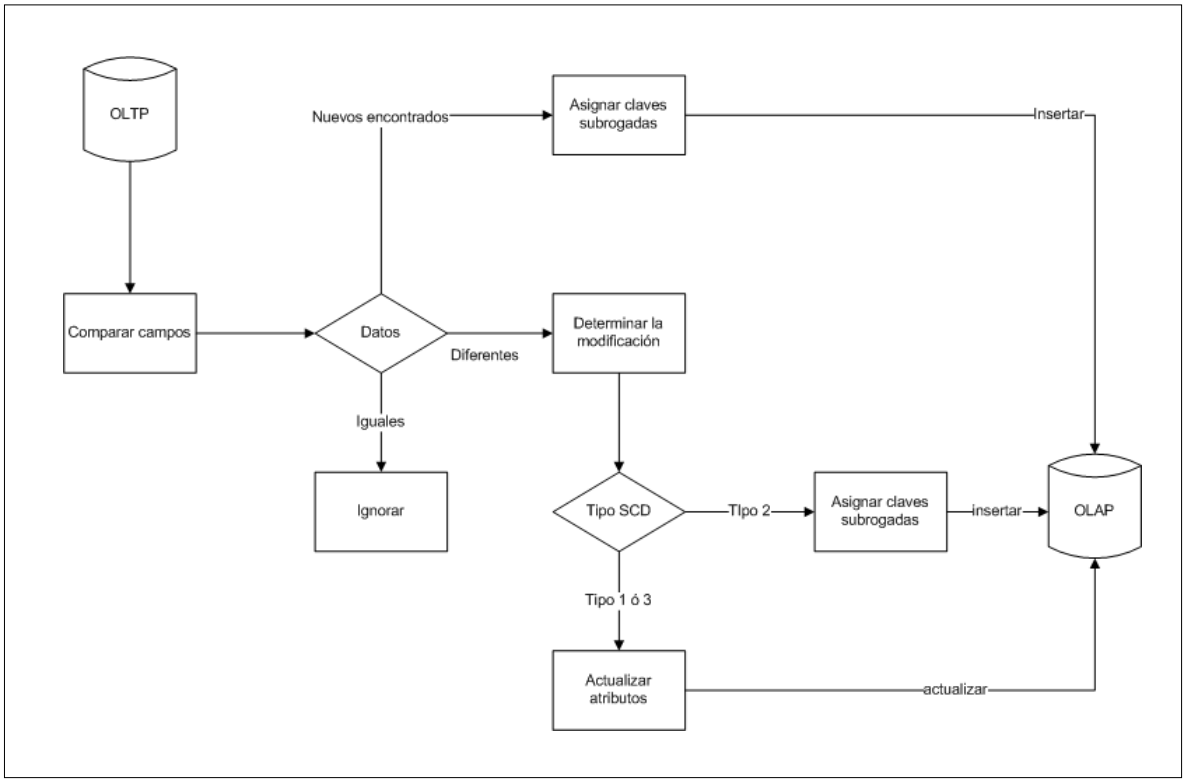

Figura 4.4: Algoritmo para el tratamiento de las SCD

4.3.3 Implementación en Kettle

Kettle (K Extraction, Transformation, Transportation & Load Environment) es una herramienta de código abierto que contiene una amplia gama de utilidades diseñada para ayudar en los procesos ETL. Está dividida en pasos a través de los cuáles se implementan las diferentes tareas. A continuación se muestran los más importantes:

- Input: Contiene las diferentes fuentes de datos con las que se realizará la extracción de la información, tales como: archivos CSV, archivo XML, archivo XLS, archivos genéricos o bien información directa de una tabla por medio de una secuencia SQL.
- Output: Contiene los diferentes destinos que pueden tener los datos una vez que han pasado por el proceso de transformación, tales como: archivos CSV, archivos de propiedades, archivos XML o bien una tabla en la base de datos OLAP.
- Transform: Contiene las tareas básicas de transformación de datos como filtrado de filas, mapeo de valores, agrupación, calculadora, normalización y desnormalización, pivoteo, validación de datos, entre otras.
- Scripting: Puede ser considerado parte del paso anterior debido a que contiene un editor que soporta javascript para que a través de este los datos sean manipulados y transformados por medio de las diferentes funciones del lenguaje de script.
- Data warehouse: Es este paso el que contiene las tareas para la implementación del llenado de las diferentes dimensiones y la tabla de hechos, Kettle se encargará de crear las claves subrogadas y administrarlas, junto con las primarias originales, para la gestión de las tablas.

Cada proceso creado para la manipulación de datos se hace de manera gráfica indicando la interacción de los elementos.

Implementación de Dimensión Escuelas:

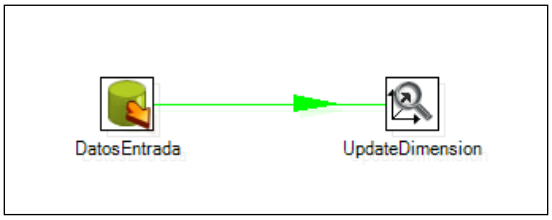

Figura 4.5: ETL de dimensión Escuelas

Esta tarea cuenta con 2 procesos principales, la extracción de datos de una tabla de la base de datos OLTP y la carga hacia una de las dimensiones del cubo OLAP.

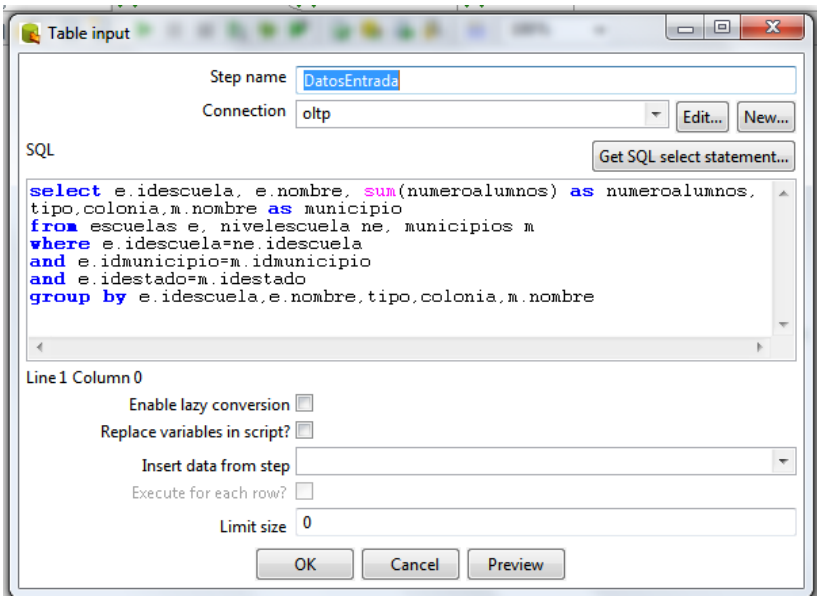

Figura 4.5 (*continuación*)

En el primer proceso se seleccionan, por medio de una sentencia SQL, los atributos del sistema operacional que serán utilizados para alimentar el sistema OLAP así como la conexión del origen de datos.

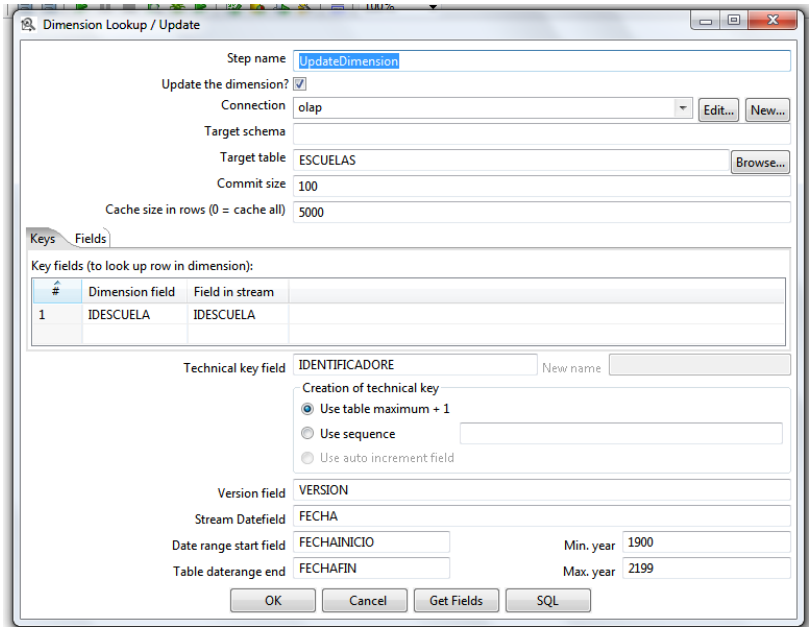

Figura 4.5 (*continuación*)

En el siguiente paso se configuran la conexión hacia la base de datos destino, la tabla a la que llegarán los datos extraídos, la columna que contiene el identificador original de cada fila, la columna que contendrá el identificador subrogado de los registros (*Technical Key Field*), la estrategia que se utilizará para su creación (*máximo + 1, secuencia o autoincremental*), así como las columnas que indicarán el periodo de validez del registro (*Date range start field-Table daterange end*) y la versión de este (*Version Field*) para que se haga uso del algoritmo SCD.

Mediante el check button *Update Dimension?* se le indica si se tratará de una operación de sólo lectura (únicamente obtiene la clave subrogada mediante el identificador original) o de lectura/escritura (obtiene la clave subrogada y en caso de que se detecte una variación en los datos originales se actualizará la dimensión).

La implementación de la dimensión Uniformes es bastante similar a la anterior, por lo que sólo resta implementar la dimensión Tiempo y la conjunción de todas con la tabla de hechos.

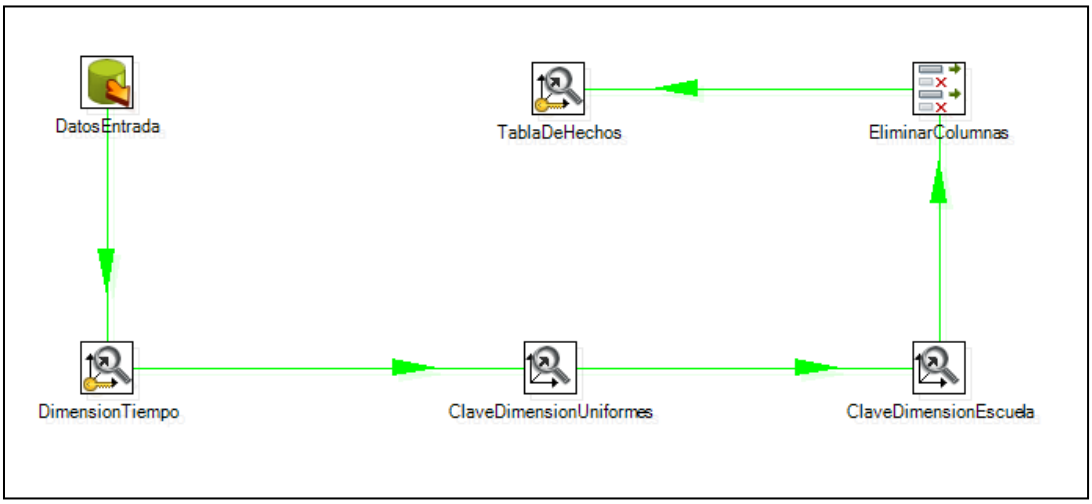

Figura 4.6: ETL de la tabla de Hechos y dimensión Tiempo

Es conveniente unir en una misma tarea la manipulación de la dimensión tiempo y la tabla de hechos debido a que cada hecho sucede en una fecha determinada, partiendo de ese dato se puede poblar de manera automática la dimensión de tiempo.

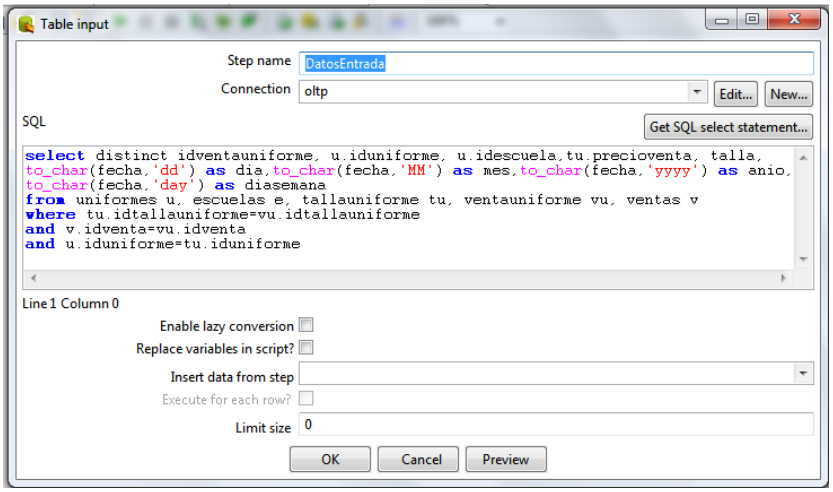

Figura 4.6 (*continuación*)

En el paso inicial se obtienen las columnas requeridas por la tabla de hechos (principalmente identificadores) así como la fecha en que ocurrió tal evento.

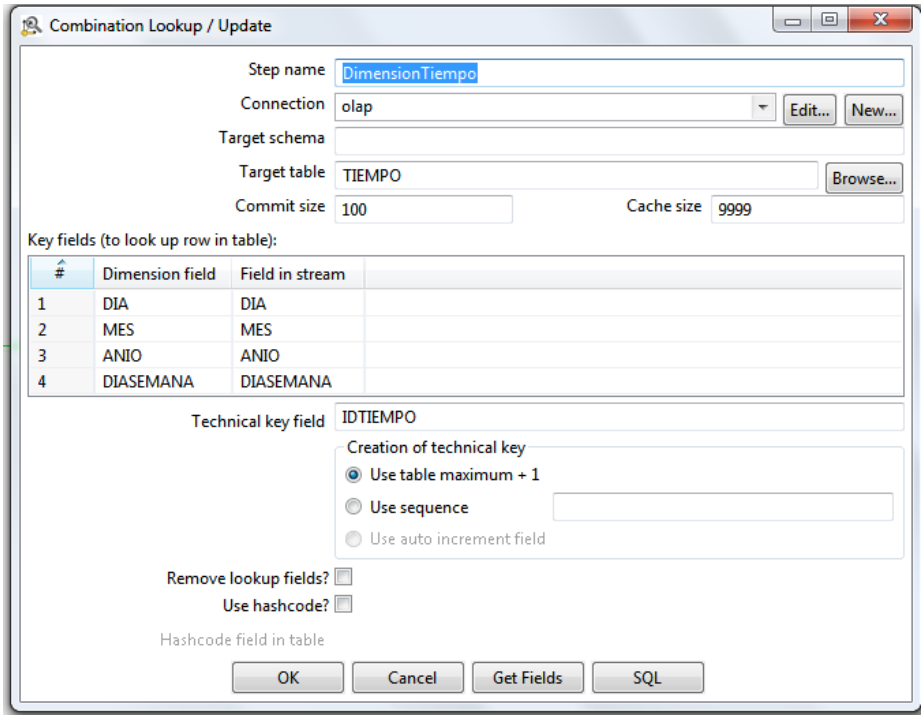

Figura 4.6 (*continuación*)

Se utiliza el paso *Combination Lookup/Update* para llenar la tabla del tiempo. El funcionamiento de este paso es el siguiente: Mediante los atributos seleccionados (dia, mes, anio, diasemana, en este caso) realiza una búsqueda en cada una de las filas de la tabla indicada. Si la combinación de valores de estos atributos (provenientes del paso anterior) existe, entonces se obtiene la clave subrogada que se creó para dicha combinación, de lo contrario se inserta una nueva fila con dicha combinación de valores y retorna la clave subrogada creada.

En los siguientes dos pasos se lleva a cabo la configuración de la obtención de las claves subrogadas de las dimensiones escuelas y uniformes, por lo que deberán ser marcadas como sólo lectura usando el identificador original obtenido en el primer paso.

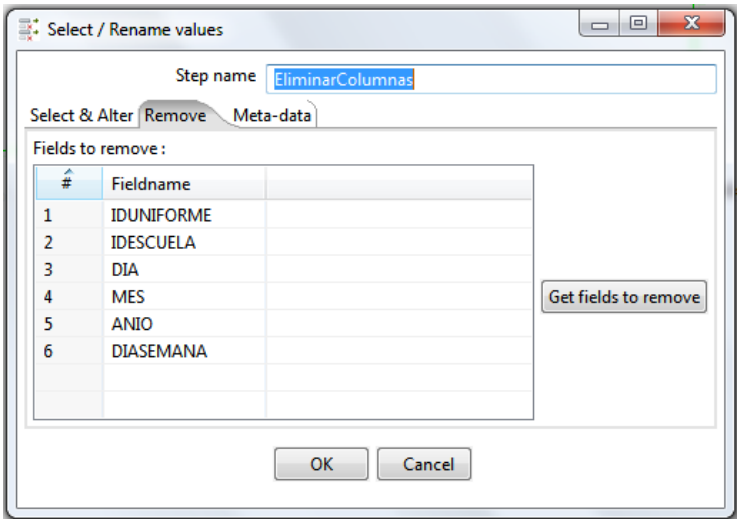

Figura 4.6 (*continuación*)

Debido a que el flujo de datos obtenidos desde el paso 1 hasta el paso actual se conserva, existen atributos que no son necesarios para poblar la tabla de hechos (tales como los identificadores originales útiles para la recuperación de las claves subrogadas y la combinación de valores de la fecha útiles para la dimensión tiempo) por lo que deben ser removidos del flujo y únicamente mantener los que son requeridos por la tabla central.

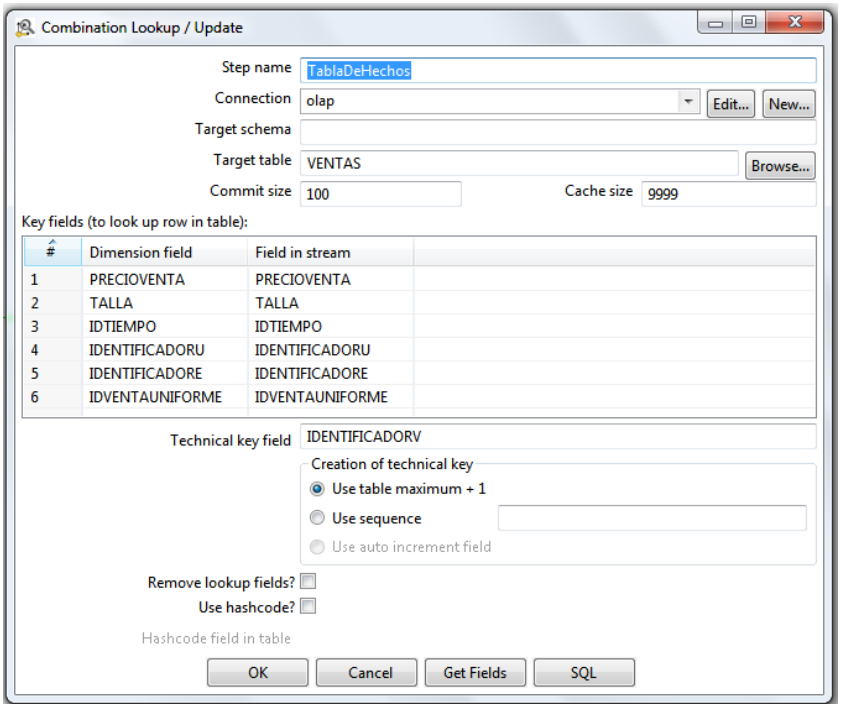

Figura 4.6 (*continuación*)

Finalmente, con la combinación de identificadores obtenidos y las medidas requeridas, se inserta un nuevo registro en la tabla de hechos con su propio identificador sustituto.

Todo este proceso debe realizarse de manera automática en intervalos de tiempo regulares. Kettle cuenta con un programa, llamado Kitchen, que realiza la ejecución automática de los trabajos programados.

Kitchen permite la ejecución de un trabajo por medio de dos opciones:

1. A través de un archivo: Los trabajos realizados en Kettle pueden ser guardados en archivos de extensión .kjb en el que se almacena toda la configuración del trabajo. Para ser ejecutado basta con llamar al programa y la ubicación del archivo por medio de su opción file en línea de comandos.

```
Kitchen.bat /file:"C:\ActualizarCubo.kjb"
```

Código 4.1: Programación automatic de trabajos

2. A través del repositorio: Si se opta por esta opción, se debe especificar el nombre del repositorio, el nombre del trabajo a ejecutar, el nombre de usuario y su contraseña.

```
Kitchen.bat /rep:"OLAP" /job:"TablaHechos" /user:usuario /pass:pass
```

Código 4.1 (*continuación*)

Es de vital importancia conocer lo que sucede con cada una de las ejecuciones automáticas que se llevan a cabo para así comprobar que el sistema OLAP está recibiendo los datos de manera correcta o de lo contrario, diseñar una estrategia para trabajar con los datos erróneos. El resultado de cada una de estas ejecuciones puede ser redirigido hacia un archivo de bitácoras incremental, en el que se indicará la hora de ejecución, el trabajo realizado y si fue finalizado con éxito o no. Para esto, después de especificar el tipo de ejecución (por archivo o repositorio), se indica el archivo al que se mandarán todos estos resultados.

```
Kitchen.bat /file:"C:\ActualizarCubo.kjb" > C:\LOG\trans.log
```

Código 4.1 (*continuación*)

Para la programación automática de estos procesos, las instrucciones deben ser guardadas en un archivo por lotes e indicar el momento en el que deben ser ejecutadas con el comando `at` en Windows o con `crontab` en Linux.

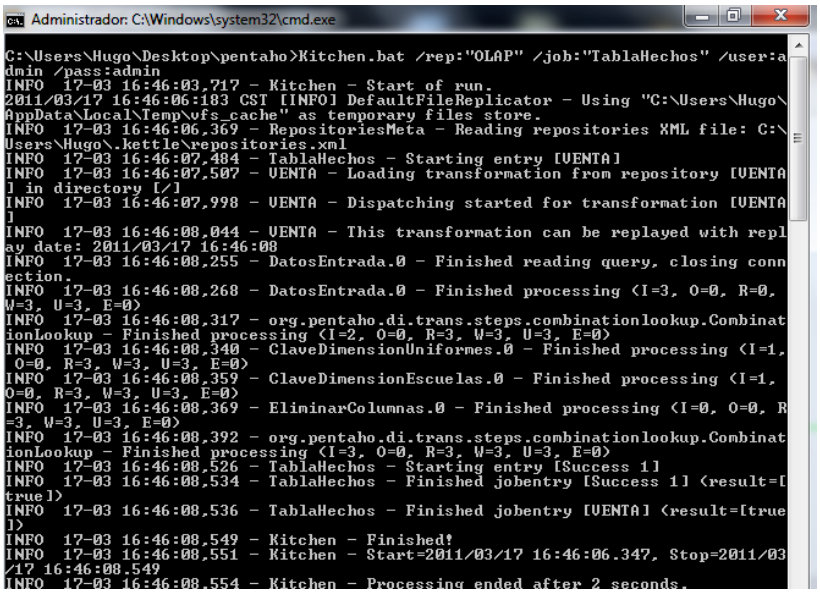

Figura 4.7: Ejecución de Trabajo mediante Kitchen.bat