Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 89-103

Tipo de artículo: Artículo original
Temática: Tecnologías de bases de datos
Recibido: 28/08/2014 | Aceptado: 22/06/2015

Técnicas para capturar cambios en los datos y mantener actualizado un almacén de datos

Techniques to capture changes in data and keep updated a data warehouse

Lisandra Díaz-De-la-Paz [1*], Juan Luis García-Mendoza[1], Beatriz Eugenia López-Porrero[1], Luisa Manuela González-González[1], Wilfried Lemahieu[2]

[1] Universidad Central "Marta Abreu" de Las Villas. Carretera a Camajuaní km 5½. Santa Clara, Villa Clara, Cuba. {jgarcias, blopez, luisagon}@uclv.edu.cu.
[2] Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven, Naamsestraat 69, B-3000 Leuven, Belgium. Wilfried.Lemahieu@econ.kuleuven.be

[*] Autor para correspondencia: ldp@uclv.edu.cu

Resumen

Durante los procesos de extracción, transformación y carga es de vital importancia mantener actualizado el almacén y los mercados de datos. En la práctica, varias son las técnicas existentes que se pueden emplear para capturar los cambios en los datos. Aunque desde el punto de vista teórico han sido estudiadas estas técnicas, en la práctica no se visualiza de manera organizada su empleo. Por tanto es necesario caracterizarlas, compararlas y seleccionar la más adecuada. Teniendo en cuenta los aspectos simplicidad, tipos de fuentes en las que se puede utilizar, operaciones que detecta y la no pérdida de información, se eligió la basada en *snapshot* como la más apropiada. Además se propuso un conjunto de pasos a seguir para ser aplicados ante una problemática real, sugiriéndose el uso de pasos pertenecientes a la herramienta de integración de datos *Pentaho Data Integration*.

Palabras clave: almacén de datos, mercados de datos, ETL, *snapshot*, técnicas de CDC.

Abstract

During the extraction, transformation and loading processes it is critical to keep updated the data warehouse and data marts. In practice, several techniques that can be used to capture changes in the data are available. Although from the theoretical point of view these techniques have been studied, in practice its form of use is not visualized in an organized way. Therefore, it is necessary to characterize, compare them and select the most appropriated. Given the

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 89-103

simplicity aspects, types of sources that can be used, operations it can detect and no loss of information, it was chosen the technique based on snapshot as the most appropriated. In addition a set of steps was proposed to follow for application to a real problem, suggesting the use of steps belonging to the data integration tool Pentaho Data Integration.

Keywords: *CDC techniques, data marts, data warehouse, ETL, snapshot.*

Introducción

Hoy en día, los procesos de extracción, transformación y carga (ETL, por sus siglas en inglés) son el centro de cualquier almacén de datos (Jörg and Dessloch, 2008; Jörg and Dessloch, 2009; Kimball and Caserta, 2004; Kimball and Ross, 2002) y reciben una atención considerable en el mercado de integración de datos (Jörg and Dessloch, 2009). El 70% de los recursos necesarios para la implementación y mantenimiento de un almacén de datos [acrónimo del inglés *Data Warehouse* (DWH)] son típicamente consumidos por estos procesos (Kimball and Caserta, 2004).

Los procesos ETL requieren las habilidades de analistas de empresas, diseñadores de bases de datos, desarrolladores de aplicaciones y no son eventos únicos. Como los datos de las fuentes se modifican, el DWH debe actualizarse periódicamente. Además, como el negocio cambia, las necesidades del DWH también suelen cambiar. Los procesos ETL deben ser diseñados para que sean fáciles de modificar; además, un proceso ETL sólido, bien diseñado y documentado es necesario para el éxito de un proyecto de DWH (El-Sappagh et al., 2011).

Según Vassiliadis (2009), las funcionalidades de los procesos ETL pueden estar resumidas en las siguientes tareas prominentes:

- Extraer los datos apropiados de las fuentes de datos.
- Transportarlos para un área de preparación de los datos [acrónimo del inglés *Data Staging Area* (DSA)] donde estos serán procesados.
- Transformar los datos fuentes y el cálculo de los valores nuevos (y, posiblemente los registros) con el propósito de obedecer la estructura de la relación del DWH para el cual son dirigidos.
- Aislar y limpiar los registros problemáticos, para garantizar que las reglas de negocio y las restricciones de la BD sean respetadas.
- Cargar los datos limpios y transformados para la relación apropiada en el DWH.

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 89-103

Kimball and Caserta (2004) crean 38 subsistemas para añadir estructura a las diversas tareas que son parte de un proceso ETL y son requisitos magníficos para validar cualquier solución ETL implementada mediante un software de integración de datos. Luego, Kimball et al. (2008) reestructuran los subsistemas y los reducen finalmente a 34.

El presente trabajo tiene como objetivos: caracterizar las técnicas de captura de cambios en los datos [acrónimo del inglés *Change Data Capture* (CDC)], compararlas y seleccionar la más apropiada al implementar los procesos ETL para proponer un conjunto de pasos que se deben seguir al aplicarse en un problema real.

Materiales y métodos

Durante los procesos ETL, los datos son extraídos de múltiples y heterogéneas fuentes como: bases de datos de Procesamiento de Transacciones en Línea [acrónimo del inglés *OnLine Transaction Processing* (OLTP)], archivos de texto, sistemas legados, hojas de cálculo, entre otras. Luego estos se transforman y limpian con el objetivo de ajustarlos al diseño del DWH con la mayor calidad posible y por último se cargan en dicho DWH, donde quedan accesibles para las aplicaciones de inteligencia de negocios (Berson and Smith, 1997, El-Sappagh et al., 2011).

Proceso de extracción

El primer paso en cualquier proceso ETL es la extracción de los datos (El-Sappagh et al., 2011), los cuales poseen distintas características que necesitan ser manejadas por el proceso ETL para lograr una extracción eficaz. Durante este proceso, el equipo de desarrollo debe estar al tanto de los drivers ODBC\JDBC utilizados para conectarse a las bases de datos, la estructura de los datos y cómo manejar las fuentes con naturaleza heterogénea. La extracción es la parte más difícil de la actualización del DWH, esto es debido a dos hechos: primero, el software de extracción debe provocar afectaciones mínimas en los sistemas fuentes durante la corrida y en segundo lugar, el software de extracción debe ser instalado en el lado fuente con un mínimo efecto en la configuración del software de la fuente (Vassiliadis, 2009). En este proceso los datos son extraídos de sus fuentes y propagados para el DSA (Kimball et al., 1998, Castellanos et al., 2009) y está dividido en dos fases: la inicial y la incremental (Kimball et al., 1998; El-Sappagh et al., 2011; Yuan et al., 2011).

Extracción inicial

En la extracción inicial, se obtienen por primera vez los datos de sus diferentes fuentes para ser cargados dentro del DWH. Este proceso se hace solo una vez después de construir el DWH para poblarlo con un gran volumen de datos

(Kimball et al., 1998; El-Sappagh et al., 2011). La captura de cambios en los datos en sus fuentes no tiene importancia porque en la mayoría de los casos se extrae la fuente de datos entera (Kimball and Caserta, 2004).

Extracción incremental

La aproximación más sencilla para la actualización del DWH es llamada *full reloading*, que consiste en volver a correr el proceso ETL de carga inicial (Jörg and Dessloch, 2009, Jörg and Dessloch, 2010). De este modo, los cambios requeridos para la actualización del DWH pueden ser recuperados. Esta aproximación es ineficiente porque frecuentemente solo una fracción pequeña de los datos fuentes cambia durante los ciclos de carga, y solo se necesita capturar estos cambios y propagarlos para el DWH. Además no es práctico eliminar y volver a cargar los datos del DWH puesto que los datos históricos tienen que ser mantenidos.

Otra aproximación es la incremental, su propósito es capturar solo los datos que cambiaron en las fuentes desde la última extracción, lo que la hace más eficaz que *full reloading* (Jörg and Dessloch, 2008; Jörg and Dessloch, 2009; Jörg and Dessloch, 2010). Los procesos ETL utilizan diversas técnicas de CDC, las cuales constituyen el subsistema dos de Kimball et al. (2008), para capturar los datos modificados, añadidos y eliminados en las fuentes de datos desde la última extracción y actualizar el DWH (Ram and Do, 2000; Kimball and Caserta, 2004; Bouman and Van Dongen, 2009; Casters et al., 2010; Jörg and Dessloch, 2010; Eccles, 2013). Estos procesos son periódicos coincidiendo con el ciclo de actualización del DWH y las necesidades del negocio (El-Sappagh et al., 2011). Cada una de estas técnicas impone uno de dos estilos arquitectónicos en el subsistema CDC.

Las dos arquitecturas son: *Pull* CDC, la cual captura los cambios en los datos en las fuentes de datos y *Push* CDC, que detecta los cambios en los datos en su ruta hacia las fuentes de datos. Las técnicas de CDC que imponen arquitecturas *Pull* son las más frecuentes, debido a la relativa facilidad con que se implementan. Las técnicas *Push* son rara vez usadas, pero tienen la ventaja de estar en mejor posición para permitir la captura de los cambios en los datos en tiempo real.

En el presente trabajo se analiza la arquitectura *Pull* CDC y se utiliza la clasificación de Bouman and Van Dongen (2009) y Casters et al. (2010), porque además de conceptualizarlas ofrecen soluciones en la herramienta *Pentaho Data Integration* (PDI), la cual es considerada una de las más flexibles y potentes en esta área de trabajo. De acuerdo con esta clasificación existen dos categorías principales de técnicas de CDC: intrusivas y poco intrusivas.

Técnicas intrusivas

Las técnicas intrusivas son aquellas que tienen un posible impacto en el desempeño de la fuente donde los datos son recuperados, y por tanto, cualquier operación que requiera ejecutar sentencias SQL en una forma u otra es

considerada una técnica intrusiva (Bouman and Van Dongen, 2009; Casters et al., 2010). Dentro de estas se destacan las basadas en: fuentes de datos, disparadores y *snapshot*.

Técnica de CDC basada en fuentes de datos

La técnica de CDC basada en fuentes de datos emplean columnas de auditoría (Kimball and Caserta, 2004) y depende del hecho de que dichas columnas estén disponibles en las fuentes, de manera que permita al proceso ETL hacer una selección de los registros que cambiaron (Bouman and Van Dongen, 2009; Casters et al., 2010). Estos atributos son usualmente poblados por disparadores de la base de datos y sirven de criterio de selección para capturar los cambios en los datos que ocurrieron desde la última extracción (Jörg and Dessloch, 2008; Jörg and Dessloch, 2009; Jörg and Dessloch, 2010). Existen dos alternativas dentro de esta técnica:

- Lectura directa basada en *timestamps* (valores de fecha y tiempo): Al menos un atributo de tipo *timestamp* se necesita para esta alternativa pero preferentemente se manejan dos: uno con la fecha y tiempo de cuando se creó el registro y otro cuando fue modificado por última vez.
- Secuencias de la base de datos: La mayoría de las bases de datos tienen algún tipo de opción de autoincremento para valores numéricos en una tabla. Cuando tal secuencia es usada, es fácil de identificar cuáles registros se han insertado desde la última vez que se analizó la tabla.

Las alternativas anteriores requieren tablas auxiliares en el DWH para almacenar la información referente a la última fecha en la cual los datos fueron cargados o el último número recuperado de la secuencia (Bouman and Van Dongen, 2009; Casters et al., 2010). Una práctica común es crear estas tablas ya sea en un espacio separado o en el DSA, pero nunca en el almacén o mercado de datos. Esta técnica es considerada por (Bouman and Van Dongen, 2009; Casters et al., 2010) la más simple de implementar; sin embargo, esta simplicidad es discutida, debido a la ausencia de algunas capacidades esenciales que pueden encontrarse en opciones más avanzadas:

- Distinción entre inserciones y actualizaciones: Solamente cuando la fuente de datos contiene dos *timestamps*, uno para inserciones y otro para las actualizaciones, esta diferencia puede ser detectada.
- Detección de registros eliminados: Esto no es posible, a menos que el sistema fuente sólo borre un registro de forma lógica. La columna tiene una fecha de eliminación pero no está físicamente borrado de la tabla.
- Detección de múltiples actualizaciones: Cuando un registro es actualizado múltiples veces durante el período comprendido entre las cargas inicial y actual, estas actualizaciones intermedias se pierden durante el proceso.

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 89-103

- Capacidades de tiempo real: *Timestamp* o extracción basada en secuencias de datos es siempre una operación *batch* y por consiguiente inadecuado para ser usado en tiempo real.

Técnicas de CDC basada en disparadores

Los disparadores de una base de datos pueden usarse para desencadenar acciones ante la ocurrencia de eventos *insert*, *update* o *delete* (Bouman and Van Dongen, 2009; Casters et al., 2010). Además, pueden utilizarse para capturar los cambios en los datos y colocar los registros cambiados en tablas intermedias en las fuentes, para luego extraerlos o ponerlos directamente en el DSA del DWH. Esta técnica no es implementada muy a menudo porque agregar disparadores a una base de datos se prohíbe por seguridad y eficiencia. Esta técnica es considerada la más intrusiva, pero tiene la ventaja de detectar todos los cambios en los datos, permitir cargas en tiempo real (Bouman and Van Dongen, 2009; Casters et al., 2010) y capturar los cambios en los datos a través de la interfaz ODBC sin necesidad de crear o comprar herramientas especializadas (Eccles, 2013).

Las desventajas de esta técnica son: la necesidad de permisos por parte de los administradores de la base de datos [acrónimo del inglés *Database Administrator* (DBA)] para poder modificar la fuente, la sintaxis específica de las declaraciones de los disparadores y esto conlleva a un alto costo de procesamiento y espacio de almacenamiento adicional (Bouman and Van Dongen, 2009; Casters et al., 2010). Una alternativa a utilizar los disparadores directamente en las fuentes de datos es establecer una solución de replicación donde todos los cambios detectados en las tablas seleccionadas son duplicados hacia las tablas receptoras en el lado del DWH.

Técnica de CDC basada en s*napshot*

La técnica *snapshot* guarda una copia exacta de cada extracción previa en el DSA para su uso futuro y durante la siguiente corrida, el proceso lleva la tabla fuente entera al DSA donde se compara con los datos cargados durante el último proceso (Kimball and Caserta, 2004). Si bien no es la técnica más eficiente, es la más confiable de todas las técnicas incrementales de extracción para capturar cambios en los datos porque hace una comparación fila por fila en busca de cambios y es casi imposible la pérdida de datos. Adicionalmente, tiene la ventaja que las filas borradas en la fuente de datos pueden ser detectadas. Esta técnica, conocida también como *snapshot differential,* es además apropiada para todos los tipos de fuentes de datos (Jörg and Dessloch, 2008; Jörg and Dessloch, 2009) Castellanos et al. (2009). Según estos autores, los datos extraídos, después de guardados en el *snapshot* actual *Lnew,* son comparados contra un *snapshot* previo (*Lold),* para discriminar inserciones, eliminaciones y actualizaciones recientes en los registros. Esta comparación es realizada a través de un operador de diferencia, [*Diff* (Δ)], que revisa en busca de igualdad sólo en un cierto subconjunto de atributos de los registros (generalmente la llave primaria). Considerando A, como un conjunto de atributos y B un subconjunto de estos se pueden encontrar los registros recién insertados,

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 89-103

mediante la expresión: ΔB (*Lnew*, *Lold*) = $\{x \in Lnew \mid \neg y \in Lold: x[b1] = y[b1] \wedge .. \wedge x[bn] = y[bn]\}$ donde b1,…,bn ∈ B.

Para encontrar un registro actualizado, se considera que para cada registro de *Lnew* existe un registro en *Lold* con los mismos valores para B y al menos un atributo perteneciente a A con un valor diferente. (Si A=B entonces se puede utilizar el operador de diferencia de relaciones clásico). Invirtiendo el uso del operador de diferencia, se obtienen los registros borrados. El último paso de esta fase es reemplazar la *snapshot Lold* con *Lnew*. Varios métodos pueden servir para eso. Uno de ellos borra la *snapshot* más antigua y simplemente renombra *Lnew* como *Lold* (primero una supresión lógica es realizada para no afectar la carga de trabajo del sistema, y entonces en un punto posterior inactivo, la supresión física se hace). Otro método es actualizar *Lold* con los registros que cambiaron. El uso de *snapshot* tiene un doble propósito. Puede ser considerada como una solución de respaldo cuando se cometen errores o servir de DSA.

Técnicas poco intrusivas

Las técnicas poco intrusivas son aquellas que tienen un bajo impacto en el desempeño de la fuente donde se recuperan los datos.

Técnica de CDC basada en archivos log

La forma más avanzada y menos intrusiva entre las técnicas de CDC es usar una solución basada en archivos *log*; en los cuales puede ponerse cada operación de inserción, actualización y eliminación manejada en una base de datos (Bouman and Van Dongen, 2009; Casters et al., 2010). Esta técnica es típicamente utilizada en conjunto con Sistemas Gestores de Bases de Datos (SGBD). (Jörg and Dessloch, 2008; Jörg and Dessloch, 2009). Cuando un archivo *log* es vaciado, todas las transacciones dentro de él son irrescatables. Para evitar esto se aconseja que el DBA cree un archivo *log* especial específicamente para esta técnica de CDC porque solo se necesitan transacciones para algunas tablas específicas de la base de datos fuente (Kimball and Caserta, 2004). Existen algunas variantes para la implementación de estas técnicas de CDC: *log scraping* o *log sniffing* (Kimball and Caserta, 2004). *Log scraping* analiza gramaticalmente los archivos *log* y recupera los cambios de interés (Labio and García-Molina, 1995; Labio and García-Molina, 1996; Jörg and Dessloch, 2008; Jörg and Dessloch, 2009; Jörg and Dessloch, 2010). *Log sniffing*, en contraste, recorta el archivo *log* y captura los cambios muy de prisa. Mientras estas técnicas tienen poco impacto en la base de datos fuente, implican alguna latencia entre la transacción original y los cambios capturados. Obviamente, esta latencia es más alta para el acercamiento *log scraping* (Jörg and Dessloch, 2010).

Comparación entre las técnicas de CDC

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 89-103

La tabla 1 muestra una comparación entre las técnicas CDC analizadas. En la cual se puede apreciar la eficiencia de las técnicas basadas en archivos *log* y disparadores (*triggers)* cuando las fuentes son bases de datos. Sin embargo las basadas en *snapshot* son independientes del DBMS, lo que implica que se puede aplicar en cualquier fuente de datos, además permite distinguir entre actualizaciones e inserciones y es capaz de detectar las eliminaciones. Por tanto la técnica seleccionada para ilustrar el procedimiento de la implementación de procesos ETL para mantener actualizado un DWH o mercado de datos de manera automática es la basada en *snapshot*.

Tabla 1: Comparación entre las técnicas CDC (Casters et al., 2010).

Aspectos	Timestamps	Snapshot	Triggers	Log
Distinción entre inserción/actualización	NO	SI	SI	SI
Detección de Múltiples Actualizaciones	NO	NO	SI	SI
Identificación de Eliminaciones	NO	SI	SI	SI
Poco Intrusivo	NO	NO	NO	SI
Soporta Tiempo Real	NO	NO	SI	SI
Requiere DBA	NO	NO	SI	SI
Independiente del DBMS	SI	SI	NO	NO

DBMS: Sistema Gestor de Base de Datos

Uno de los algoritmos más usados para la implementación de esta técnica es el *MergeSort* (Labio and García-Molina, 1996). Su idea general es tomar dos *snapshots*, *F1* y *F2*, compararlos y devolver uno nuevo, denominado *Fout*.

La herramienta PDI en el paso *Merge rows (diff)* utiliza un algoritmo similar al *MergeSort*. Este paso toma dos conjuntos ordenados de entrada y los compara en las llaves especificadas (K). Las columnas a ser comparadas pueden ser seleccionadas (*B*) y debe especificarse un nombre para el campo que contiene la bandera de salida la cual puede tomar uno de los siguientes valores: *identical*, *new*, *changed* o *deleted* (Casters et al., 2010).

Proceso de transformación

El segundo paso en cualquier proceso ETL es la transformación de los datos (Vassiliadis, 2009). Este paso realiza la limpieza y conformación de los datos entrantes para obtener datos precisos, correctos, completos, coherentes, y no ambiguos. También incluye depuración de datos, su transformación e integración.

La limpieza es la corrección en los datos de posibles errores, por ejemplo: datos incompletos, duplicados, formatos inconsistentes en cuanto a descripción, abreviaturas y unidades de medidas, falta de datos de entrada o que violen las restricciones de integridad del sistema. Esta etapa es una de las más importantes, ya que garantiza la calidad de los datos en el DWH y en ella se deben corregir las anomalías que se detecten en el proceso de integración de los datos.

La calidad de los datos es un término que abarca el estado de los datos, así como el conjunto de procesos para lograrla. Cuando la información se encuentra limpia y con calidad, ésta es unificada, conformada y normalizada. Los indicadores son calculados de una forma racional, lo mismo que los atributos de las dimensiones, para que estén unificados y en todos los sitios donde aparezcan tengan la misma estructura y el mismo significado (Kimball and Caserta, 2004; Díaz et al., 2013).

Proceso de carga

La carga de los datos para la estructura dimensional seleccionada es el paso final del proceso ETL (Vassiliadis, 2009). En este paso, los datos extraídos y transformados son escritos en las estructuras dimensionales a las que acceden los usuarios finales y aplicaciones de software. El paso de carga incluye las tablas de dimensiones y las de hechos.

Carga inicial

En la carga inicial, los datos extraídos y transformados de las fuentes de datos son cargados en el DWH (Jörg and Dessloch, 2009). Este proceso no es complejo de implementar porque el DWH debe estar vacío.

Carga incremental

Los procesos ETL diseñados para implementar la carga incremental utilizan las técnicas de CDC para capturar los cambios en las fuentes de datos (Jörg and Dessloch, 2008). Durante la misma es necesario tener en cuenta el manejo de las dimensiones lentamente cambiantes [acrónimo del inglés *Slowly Changing Dimension* (SCD)] que se corresponde con el subsistema nueve de los propuestos por (Kimball and Caserta, 2004; Jörg and Dessloch, 2008).

Las dimensiones SCD pueden cambiar de manera ocasional o constante, siendo de gran importancia el registro de los cambios históricos realizados para mantener la veracidad de la información cargada en el DWH (Tellez et al., 2012). Existen varios tipos de SCD, los más conocidos, estudiados y frecuentes son los tipos 1, 2 y 3 propuestos por (Kimball and Caserta, 2004; Tellez et al., 2012), sin embargo (Ross, 2013) corrobora el uso de los tipos 0, 4, 5, 6 y 7.

Resultados y discusión

Los procesos de extracción, transformación y carga comprenden varios aspectos que son determinantes en el proyecto de inteligencia de negocio, por lo que para su diseño e implementación se propone seguir un conjunto de pasos para su correcto desarrollo (Villarreal, 2013) usando la técnica CDC basada en *snapshot*.

Paso 1: Extracción inicial.

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 89-103

El diseño para implementar la extracción inicial se divide en dos fases (ver figura 1): en la primera, se extraen los datos desde las fuentes de datos hacia el DSA y en la segunda, se extraen los datos del DSA para su utilización por parte del paso de transformación (Mundy, 2008).

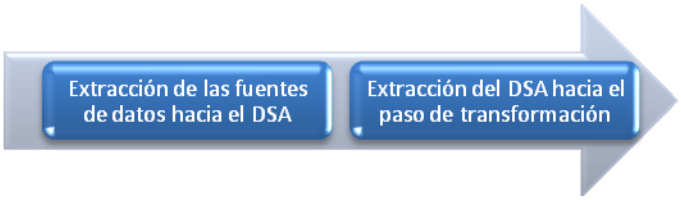


Figura 1. Diseño para la implementación de la extracción inicial

Paso 2: Extracción incremental.

La extracción incremental debe tener en cuenta los datos extraídos de las fuentes y los datos del DSA para luego compararlos mediante el paso *Merge Rows (diff)*, como se muestra en la figura 2.

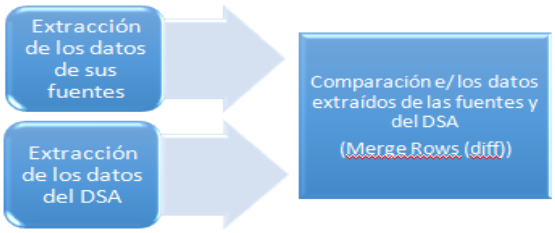


Figura 2. Diseño para la implementación de la extracción incremental

Paso 3: Transformación.

En el paso de transformación se realiza la limpieza y conformación de los datos. Para ello es necesario realizar un análisis profundo de las fuentes y aplicar transformaciones a cada inconsistencia detectada. Esto es, la ejecución del paso de transformación ya sea por la ejecución del flujo de trabajo de ETL para la carga o por la ejecución de las consultas sobre las fuentes. En la tabla 2 se muestran algunas de las anomalías más frecuentes, se proponen posibles soluciones y se sugieren pasos de la herramienta de integración de datos PDI para su implementación.

Tabla 2: Propuesta de solución en el PDI a algunas de las inconsistencias detectadas más frecuentes en las fuentes de datos.

Frecuentes anomalías detectadas	**Propuesta de solución**	**Pasos del PDI**
Existencia de valores nulos	Generar una fila con el valor NO DEFINIDO (A).	*Generate Rows*
	Poner en el campo que corresponda NO DEFINIDO (A).	*Database Lookup* *Set field value*
Valores no	Seleccionar el formato estándar y hacer	*Data validator*

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 89-103

estandarizados	corresponder al resto con este valor.	*Replace in string*
Valores con longitud fuera de rango	Reajustar a la longitud o rango establecido.	*Strings cut* *Java Script*
Redundancia	Eliminar las tuplas repetidas que representan el mismo hecho.	*Sort Rows* *Unique Rows*

Paso 4: Carga inicial.

La implementación de los procesos ETL para poblar un DWH se basa en lo planteado en el paso 2 y 3 como se aprecia en la figura 3.

Figura 3. Diseño de los procesos ETL para poblar un DWH

En los procesos ETL diseñados para poblar un DWH, se recomienda utilizar los pasos del software PDI *Combination Lookup/Update* y *Dimension Lookup/Update* en las tablas de dimensiones e *Insert/Update* para las tablas de hechos. Como se aprecia en la figura 3, primero se implementan las transformaciones que extraen los datos de sus fuentes hacia el DSA. Luego se implementan las transformaciones que realizan la extracción de los datos del DSA, su limpieza y homogeneización y la carga dentro del mercado de datos. En la figura 4 se ilustra la transformación *inicial_dim_tiempo* correspondiente al caso de estudio del mercado de datos Recursos Humanos de la UCLV (García, 2014; Masó and Castellón, 2013) para poblar la tabla de dimensiones Tiempo, que junto a otras, suelen ser orquestadas mediante la implementación de trabajos.

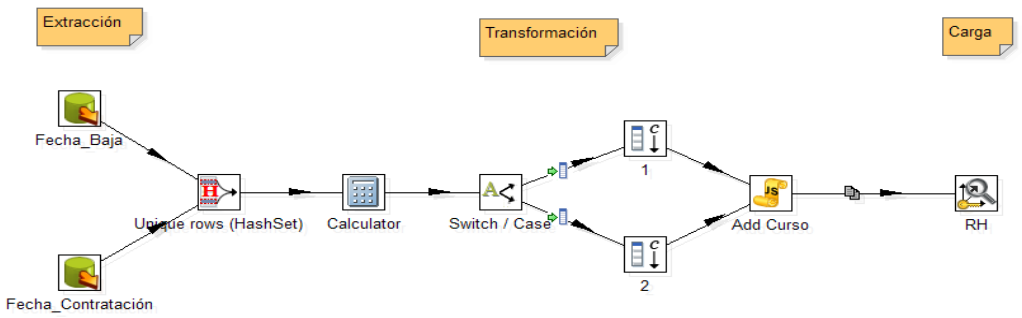


Figura 4. Transformación *inicial_dim_tiempo*

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 89-103

Paso 5: Carga incremental.

El diseño de los procesos ETL encargados de realizar la carga incremental hacia el DWH es más complejo porque se debe tener en cuenta los tipos de SCD, como se muestra en la figura 5. Los procesos ETL diseñados para realizar la carga incremental hacia el DWH utilizan los pasos del software PDI *Combination Lookup/Update*, *Dimension Lookup/Update* y *Update* para el tratamiento de los tipos de SCD.

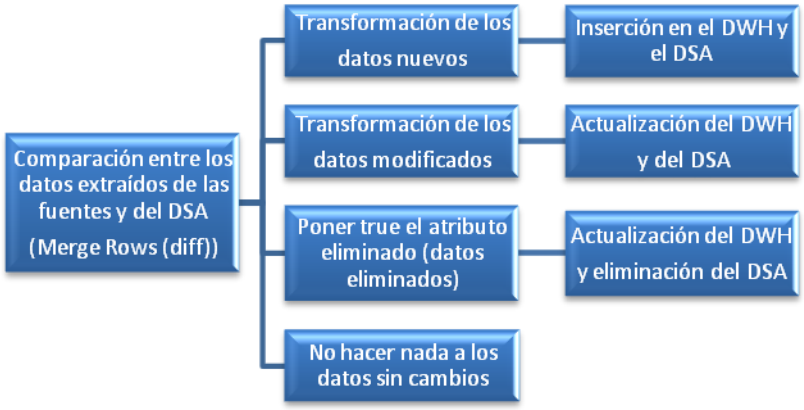
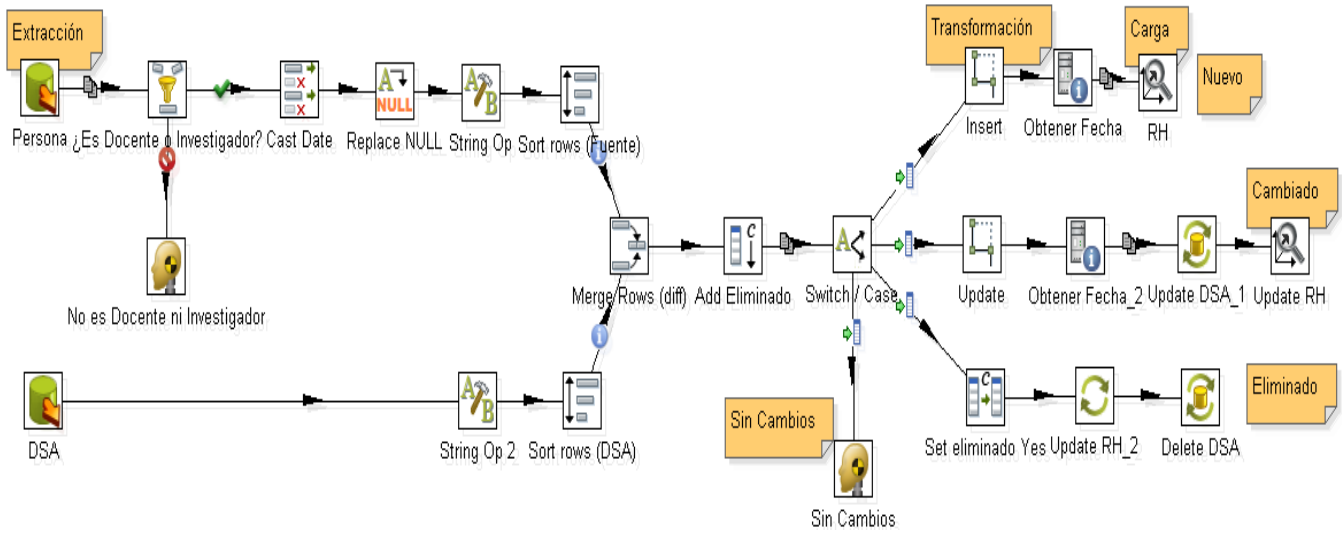


Figura 5. Diseño de las cargas incrementales

En la figura 6 se muestra la transformación *incremental_dim_persona* correspondiente a la tabla de dimensiones Persona perteneciente al mercado de datos Recursos Humanos de la UCLV (García, 2014, Masó and Castellón, 2013). En esta transformación se utilizan los pasos *Dimension Lookup/Update* para manejar los atributos SCD Tipo 2, *Insert/Update* para actualizar el DSA y *Synchronize after Merge* para realizar un borrado físico del DSA. Además, vale destacar el uso de subtransformaciones en los pasos nombrados *Insert* y *Update*.



Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 89-103

Figura 6. Transformación *incremental_dim_persona* utilizando PDI

La figura 7 muestra la implementación de una de las sub-transformaciones implementadas en dicho caso de estudio denominada *subtransformación_dim_persona*.

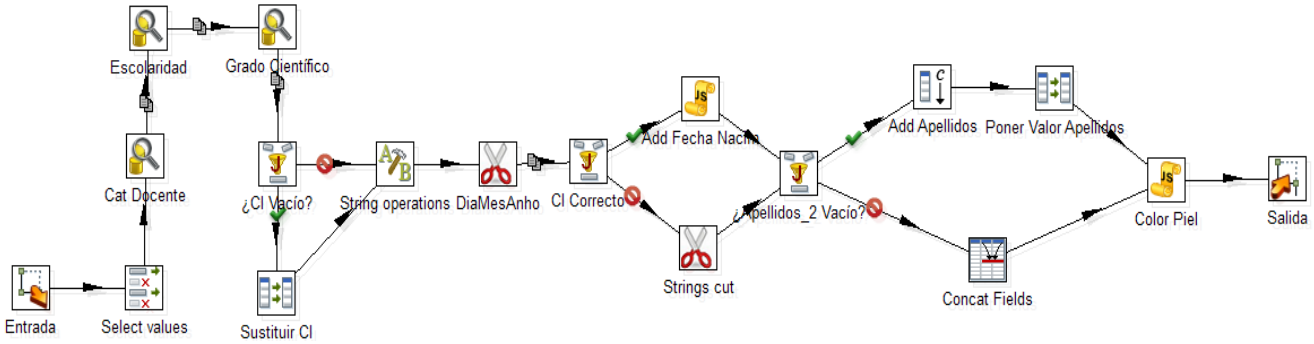


Figura 7. S*ubtransformación_dim_persona*

1.6 Comparación entre las técnicas full reloading y basada en snapshot.

Con el objetivo de evaluar el rendimiento de las técnicas *full reloading* y basada en *snapshot*, se realiza una comparación en cuanto a duración entre las transformaciones *inicial_dim_persona* (*full reloading*) e *incremental_dim_persona* (basada en *snapshot*) ejecutadas para capturar cambios en los datos luego de la carga inicial. En la figura 8 se aprecia como la duración de la técnica basada en *snapshot* muestra una reducción significativa con respecto a *full reloading*. Esta diferencia puede aumentar en la medida que aumenten los registros de las fuentes utilizadas. Los beneficios de la técnica basada en *snapshot* comparada con *full reloading* son dobles: primero, el volumen de datos cambiados en las fuentes es muy pequeño comparado con el volumen global y por otra parte, la mayoría de los datos dentro del DWH permanecen ilesos durante la carga incremental, ya que los cambios son sólo aplicados donde son necesarios.

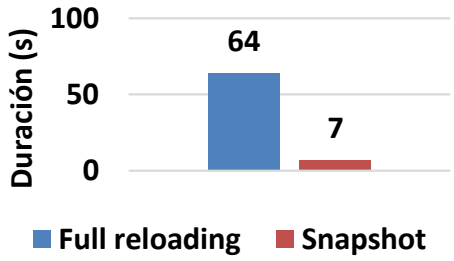


Figura 8. Comparación de la duración entre las técnicas *full reloading* y *basada en snapshot*

Conclusiones

Los procesos ETL son el centro del almacén de datos y su calidad es de importancia significativa para la exactitud, operatividad y usabilidad. Por razones de eficiencia los DWH son típicamente actualizados de forma incremental, es decir, los cambios son capturados en las fuentes y propagados para el DWH regularmente. Es por ello, que se considera de importancia vital las técnicas de CDC. En el presente trabajo, dichas técnicas se caracterizan de acuerdo con la clasificación de intrusivas y poco intrusivas, se comparan y se selecciona como la más apropiada para implementar los procesos ETL la basada en *snapshot.* La técnica escogida permite realizar una comparación fila a fila evitando la pérdida de información, se puede aplicar a cualquier tipo de fuente de datos, distingue entre inserciones y actualizaciones y detecta las eliminaciones. Además se propone un conjunto de pasos a seguir para aplicar dicha técnica a un problema real. No obstante, a la simplicidad teórica que presenta el *snapshot*, la extracción de los datos continúa siendo un problema difícil debido mayormente a la naturaleza heterogénea de las fuentes, lo cual afecta al rendimiento de la integración de los datos y plantea campos de investigación abiertos.

Referencias

BERSON, A. & SMITH, S. J. 1997. *Data Warehousing, Data Mining, and OLAP*, McGraw-Hill, Inc.

BOUMAN, R. & VAN DONGEN, J. 2009. *Pentaho Solutions: Business Intelligence and Data Warehousing with Pentaho and MySQL*, Indianapolis, Indiana, Wiley Publishing, Inc.

CASTELLANOS, M., SIMITSIS, A., WILKINSON, K. & DAYAL, U. 2009. Automating the Loading of Business Process Data Warehouses. *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. ACM.

CASTERS, M., BOUMAN, R. & VAN DONGEN, J. 2010. *Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration*, Indianapolis, Indiana, Wiley Publishing, Inc.

DÍAZ, L., LÓPEZ, B., GONZÁLEZ, L., GALINDO, Y. & LÓPEZ, D. 2013. Integración de Datos. "Universidad Central "Marta Abreu" de las Villas".

ECCLES, M. J. 2013. *Pragmatic Development of Service Based Real-Time Change Data Capture.* Aston University.

EL-SAPPAGH, S. H. A., HENDAWI, A. M. A. & EL BASTAWISSY, A. H. 2011. A proposed model for data warehouse ETL processes. *Journal of King Saud University-Computer and Information Sciences,* 23**,** 91-104.

GARCÍA, J. L. 2014. *Automatización de los procesos de carga en el mercado de datos Recursos Humanos de la UCLV*. Universidad Central "Marta Abreu" de Las Villas.

JÖRG, T. & DESSLOCH, S. 2008. Towards generating ETL processes for incremental loading. *In:* DESAI, B. C. (ed.) *Proceedings of the 2008 international symposium on Database engineering & applications.* Coimbra [Portugal]: ACM.

JÖRG, T. & DESSLOCH, S. 2009. Formalizing ETL Jobs for Incremental Loading of Data Warehouses.

JÖRG, T. & DESSLOCH, S. 2010. Near Real-Time Data Warehousing Using State-of-the-Art ETL Tools. *Enabling Real-Time Business Intelligence, Lecture Notes in Business Information Processing.* Springer-Verlag Heidelberg.

KIMBALL, R. & CASERTA, J. 2004. *The Data Warehouse ETL Toolkit,* Indianapolis, Indiana, Wiley Publishing, Inc.

KIMBALL, R., REEVES, L., ROSS, M. & THORNTHWAITE, W. 1998. *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses,* Indianapolis, Indiana, Wiley Publishing, Inc.

KIMBALL, R. & ROSS, M. 2002. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling,* New York, John Wiley and Sons, Inc.

KIMBALL, R., ROSS, M., THORTHWAITE, W., BECKER, B. & MUNDY, J. 2008. *The Data Warehouse Lifecycle Toolkit,* John Wiley & Sons.

LABIO, W. J. & GARCÍA-MOLINA, H. 1995. Comparing very large database snapshots. Stanford University.

LABIO, W. J. & GARCÍA-MOLINA, H. 1996. Efficient Snapshot Differential Algorithms for Data Warehousing. *Proceedings of VLDB '96.*

MASÓ, A. & CASTELLÓN, Y. 2013. *Mercado de datos en apoyo a la toma de decisiones sobre el personal docente e investigativo en el departamento de Recursos Humanos de la UCLV.* Universidad Central "Marta Abreu" de Las Villas.

MUNDY, J. 2008. Design Tip #99 Staging Areas and ETL Tools. Available: http://www.kimballgroup.com/2008/03/04/design-tip-99-staging-areas-and-etl-tools/.

RAM, P. & DO, L. 2000. Extracting Delta for Incremental Data Warehouse Maintenance. *Proceedings. 16th International Conference on Data Engineering (ICDE).* IEEE.

ROSS, M. 2013. Design Tip #152 Slowly Changing Dimension Types 0, 4, 5, 6 and 7. Available: http://www.kimballgroup.com/2013/02/05/design-tip-152-slowlychanging-dimension-types-0-4-5-6-7/.

TELLEZ, Y., MEDINA, D. & TORRES, R. E. 2012. Propuesta para la Implementación de las Dimensiones Lentamente Cambiantes con Pentaho Data Integration.

VASSILIADIS, P. 2009. A survey of Extract–transform–Load technology. *International Journal of Data Warehousing & Mining,* 5**,** 1-27.

VILLARREAL, R. X. 2013. *Estudio de metodologías de Data Warehouse para la implementación de repositorios de información para la toma de decisiones gerenciales.*, Universidad Técnica del Norte.

YUAN, G., LI, B. & XIAO, T. 2011. Improvement of Snapshot Differential Algorithm Based on Hadoop Platform. *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC).* IEEE.