DESARROLLO DE HERRAMIENTA DE SOFTWARE PARA DETERMINAR LA CALIDAD DE LOS DATOS DE UNA BASE DE DATOS RELACIONAL

David Santiago Palacio Colorado
Universidad Pontificia Bolivariana – Ingeniería de Sistemas e Informática
dasapaco@hotmail.com

Daniel Rodríguez González
Universidad Pontificia Bolivariana – Ingeniería de Sistemas e Informática
danyrg89@hotmail.com

Resumen — En este trabajo se dará primeramente una explicación al lector sobre qué es la calidad de datos y porque es importante para las empresas durante la toma de decisiones y generación de estrategias.

Se plantea al lector el problema al que se pretende dar solución y el enfoque que se usó para lograrlo. Se realiza una breve descripción de la arquitectura de la aplicación y que problemas soporta y finalmente unas conclusiones sobre el trabajo desarrollado.

***Palabras claves* — calidad de datos, problemas de calidad de datos, bases de datos, escalabilidad, módulos, perfilamiento de datos, métricas.**

I. INTRODUCCIÓN

La calidad de los datos almacenados por las empresas constituye un factor fundamental para la toma de sus decisiones ya que puede afectar en gran medida la implementación de proyectos que dependen de este factor, por ejemplo, los proyectos de bodegas de datos o proyectos de inteligencia de negocio.

Es por esto que la calidad de datos debe ser monitoreada con el fin de tomar medidas correctivas y servir de estimador del esfuerzo a invertir en la tarea, labor que en bases de datos de gran tamaño puede ser dispendiosa y poco eficiente realizarla sin ayuda de una herramienta especializada. Es aquí donde entra en escena nuestra aplicación, con el fin de permitir a un usuario con un nivel intermedio de conocimiento en la materia realizar la configuración de un perfilamiento completo a múltiples bases de datos en busca de problemas de calidad a través de una interfaz amigable e intuitiva que lo guiará por los pasos requeridos, desde la conexión a la base de datos hasta la presentación de los resultados. De este modo el usuario sólo tendrá que sentarse y esperar para poder observar un reporte con los resultados del análisis para los campos, tablas o bases de datos seleccionadas.

II. CALIDAD DE DATOS

Los datos a procesar adquieren un valor agregado, lo que impone como requisito que su contenido sea fiable y corresponda a la realidad. Este valor agregado se conoce como calidad [1], la cual genera un nivel de confianza sobre la información arrojada, permitiendo una mejor toma de decisiones y colaborando en el cumplimiento de los objetivos de las organizaciones que hacen uso de ella.

III. IMPORTANCIA DE LA CALIDAD DE LOS DATOS

Según Harrington [2], la calidad de los datos es importante ya que es necesario tener confianza de que lo que se obtiene de una base de datos es fiable. Las organizaciones toman decisiones tanto operacionales como estratégicas basadas en lo que se tiene almacenado en las bases de datos. La calidad de estas decisiones está directamente relacionada con la calidad de los datos que subyace bajo estos.

La falta de conocimiento de la naturaleza de los datos puede acarrear sobrecostos o llevar a fracasar procesos de bodegaje de datos como el presentado en [3] donde una compañía diseñó su modelo de datos y sus scripts de extracción, transformación y carga (ETL por sus siglas en ingles) alrededor de una base de datos de 11.5 millones de clientes solo para darse cuenta posteriormente que de hecho contenía únicamente 1.5 millones de registros de clientes distintos. En [4] se presenta una serie de casos reales donde la falta de la calidad de la información ha llevado a multimillonarias pérdidas que según cálculos del autor llegan a totalizar USD 44.589’450.000.

IV. PERFILAMIENTO DE DATOS

Siguiendo esta línea es necesario realizar un estudio y análisis de los datos para determinar su calidad. Este proceso se llama perfilamiento de datos (*data profiling*), el cual consiste en el uso de una serie de algoritmos para el análisis y la evaluación de la calidad de un conjunto de datos [5].

Este análisis se puede realizar de dos formas: manual o automática. El método manual requiere que los administradores generen sentencias SQL para verificar así su calidad. El problema de este método es que requiere conocimientos técnicos y es muy dispendioso. Por lo tanto se plantea el método automático, en el cual se usan aplicaciones que analizan la base de datos en su totalidad y generan reportes que ofrecen un mejor entendimiento del estado de los datos [3].

V. DEFINICIÓN DEL PROBLEMA

Comúnmente, la depuración de los datos de una base de datos es llevada a cabo por personal del área de tecnología informática como DBAs (administradores de bases de datos) o analistas, quienes con frecuencia no están preparados para realizar esa labor. Saber qué pasos seguir para determinar la calidad de los datos de una base de datos, unido a la variedad de posibles problemas de calidad, la multitud de técnicas existentes para cada uno de ellos y las condiciones que deben cumplirse para que las técnicas puedan aplicarse exitosamente, hacen que diagnosticar la situación de calidad de los datos de una base de datos sea una tarea difícil que, para realizarla exitosamente, requiere conocimientos tanto estadísticos como de metodologías y técnicas para limpieza de datos.

En el mercado existen herramientas de software, tanto estadísticas como de perfilamiento de datos (*Data profiling*), con diferentes tipos de licenciamiento que determinan la calidad de los datos almacenados en una base de datos. En general, estas herramientas adolecen de lo siguiente:

- No incorporan una metodología que oriente al usuario para seguir un proceso ordenado en el diagnóstico.
- La selección de la técnica durante la elaboración del perfil de la calidad de los datos requiere tener en cuenta la naturaleza de los mismos, es decir, las características específicas que presentan los datos a evaluar, ya que una técnica incorrecta puede no encontrar ningún problema o encontrar más de los existentes. Esto hace que sea necesario determinar la naturaleza de los datos para seleccionar una técnica acorde. A modo de ejemplo, para realizar una detección de cadenas de texto duplicadas, si los datos presentan problemas como abreviaturas (Juan Alberto López vs Juan A López), la técnica más adecuada es Brecha Afín. Si en este caso se utiliza Distancia de Edición, las diferencias entre las cadenas son muy grandes y no detectará que se trata de un duplicado de la cadena en cuestión.
- El análisis consolidado del estado de la base de datos es difícil de realizar ya que las herramientas entregan resultados atributo por atributo. Aunque tratar los atributos individualmente es necesario para poder tomar acción sobre los problemas encontrados, es altamente deseable poder contar con informes resumidos del estado general de la base de datos para darse una idea del esfuerzo que conllevará su limpieza y los efectos sobre un posible proyecto de Inteligencia de Negocios. Por ejemplo, no se tiene conocimiento de alguna herramienta que informe el porcentaje total de datos con problemas en una base de datos lo cual sería un indicador útil.

Se pretende entonces, realizar una investigación aplicada cuyos resultados sean plasmados en una herramienta de software que supere las dificultades enunciadas. Esta aplicación tendrá la finalidad de determinar el estado de una base de datos en cuanto a calidad de datos se refiere, esto es, se construirá un software que realice en forma semiautomática la detección de los posibles problemas de calidad de los datos incorporando una metodología de evaluación, elaborando el perfil de múltiples campos simultáneamente sin necesidad de contar con expertos en estadística o en calidad de datos y facilitando el análisis global de los resultados.

VI. ARQUITECTURA

Para el desarrollo de la aplicación se planteó una arquitectura base que permitiera el crecimiento modular y de poco impacto a medida que se necesitara agregar nuevas funcionalidades. Esta arquitectura facilita además la modificación de las funcionalidades ya implementadas sin afectar los demás componentes de la aplicación. A continuación se describen los módulos implementados.

A. Acceso a base de datos

Este módulo se definió para dar abstracción a los detalles propios de cada motor de base de datos ya que uno de los requisitos de la aplicación es permitir la conexión a múltiples fuentes. En este se definen tareas como la construcción del esquema de la base de datos, conexión desconexión y ejecución de queries.

B. Problemas de calidad de datos

Este componente agrupa los problemas soportados por la aplicación y los algoritmos de cada uno. Es el encargado de realizar el perfilamiento y análisis de los datos contenidos en las bases de datos, tablas y campos configurados por el usuario, generando los queries y ejecutándolos a través del módulo de acceso a base de datos.

C. Reportes

Una vez realizado el perfilamiento a los datos se hace necesario presentar al usuario los resultados de dicho análisis. Las disimilitudes propias de la naturaleza de cada problema y de las métricas generadas por estos, llevaron a que la implementación de un reporte genérico fuese compleja debido a que la interpretación de los resultados está fuertemente relacionada con el algoritmo que los generó. Esta situación se abordó creando una clase abstracta implementada por cada algoritmo en una clase concreta que está en capacidad de entender las métricas generadas. Cada subclase está encargada de consultar los resultados y definir los tipos de gráficas a utilizar (gráficos de torta, tablas o texto) con el fin de presentar los resultados del análisis.

D. Interfaz de usuario

La interfaz de usuario es el medio por el cual los componentes lógicos obtienen sus parámetros de funcionamiento y presentan los resultados del análisis. En este aspecto se dio a la aplicación un funcionamiento tipo wizard que brinda un manejo intuitivo y requiere una pequeña curva de aprendizaje para el usuario final debido a su extendido uso en las aplicaciones actuales. Este enfoque también permite a los desarrolladores agregar nuevos problemas a la aplicación de una manera sencilla y modular que genera poco impacto para lo construido hasta el momento.

E. Configuración

El módulo de configuración permite a los componentes de la aplicación brindar un contexto global a las variables locales que así lo requieran. Luego de que una variable alcanza este nivel, las demás páginas del wizard podrán leer, eliminar o editar su valor sin necesidad de estar acopladas con la entidad que la generó, brindando así a este módulo la capacidad de servir como medio de "comunicación" entre los componentes de la aplicación.

F. Modelo de datos de la aplicación

Para la aplicación se diseñó un modelo de datos flexible que permitiera el almacenamiento de los resultados generados durante el perfilamiento a las bases de datos. La flexibilidad radica en brindar la capacidad de almacenar los valores generados por cada tipo de problema independientemente de la naturaleza o significado del dato. Este funcionamiento permitió una fácil integración de nuevos problemas a medida que se realizaba su construcción.

VII. DESARROLLO

Para la construcción de la aplicación se utilizo el .NET Framework 3.5 y el lenguaje C# debido a la experiencia en el uso de estas y a que brindan agilidad y rapidez en la construcción de aplicaciones. Para el acceso a datos, se utilizó la tecnología Linq To SQL debido a que brinda una abstracción del modelo entidad-relación al permitir al desarrollador interactuar con los dato como si fueran objetos ordinarios. Esta tecnología se utilizó en conjunto con el motor de base de datos SQL Server 2008 Express incluido en el ambiente de desarrollo utilizado, Visual Studio 2010, y para la manipulación del esquema de la base de datos se empleó SQL Server Management Studio 2008.

VIII. PROBLEMAS SOPORTADOS

El adicionar un nuevo problema a la aplicación consiste en agregar un nuevo módulo, realizar la implementación de las clases necesarias e integrarlo al wizard para mostrar la pantalla de configuración y las gráficas de resultados en los reportes.

A. Nulos

Este problema es el encargado de buscar en las columnas seleccionadas por el usuario los campos cuyo valor sea igual a NULL. La ausencia de un valor, dependiendo de las definiciones del negocio, pude ser considerado un error, por ejemplo para campos definidos como llave primaria o cuyo valor es requerido para la entidad.

B. Integridad referencial

Este problema es el encargado de identificar los registros "huérfanos" y los "padres sin hijos" para las tablas seleccionadas por el usuario. Un análisis para este problema permitirá encontrar errores tales como ciudades que no estén asociadas a una región o facturas sin detalle.

C. Violación de llave primaria

Este problema es el encargado de identificar los registros en una entidad donde los valores en los campos definidos como llave primaria están repetidos en otra túpla de la tabla, por ejemplo, si se tienen dos números de factura idénticos o dos o más cédulas iguales.

IX. CONCLUSIONES

En el trabajo elaborado es posible apreciar que la calidad de datos puede llegar a ser un factor determinante para las empresas debido a su influencia directa en la toma de decisiones y en el esfuerzo y tiempo que es necesario dedicar cuando se emprenden proyectos de bodegas de datos o de inteligencia de negocios.

Se logró desarrollar una herramienta con una arquitectura modular .que realiza satisfactoriamente el análisis de varios problemas de calidad de datos a múltiples fuentes, entregando resultados a nivel de campo, tabla o base de datos. Esto es de gran importancia ya que puede hacer más eficiente un proceso de análisis de calidad, sobre todo cuando las fuentes poseen un gran número de entidades y atributos.

Cuando se está manejando temas que requieren del criterio de varios de los integrantes del equipo resulta más ágil que ellos se encuentren en el mismo espacio, ya que de esta forma se pueden compartir ideas, resolver inquietudes y presentar ejemplos de una manera más eficiente que al utilizar medios electrónicos sujetos a fallas técnicas, retardos, lentitud u otros inconvenientes. Adicionalmente el uso de herramientas de CVS, bug traking, servidores para hosting de base de datos facilitan el desarrollo, dado que tener centralizado el código, la base de datos y el listado de bugs permite a los desarrolladores trabajar sobre la misma versión de la aplicación y tener mayor claridad sobre los errores que esta tiene, ayudando a la solución de los mismos.

REFERENCIA

[1] Heiko Müller and Johann-Christoph Freytag, "Problems, Methods, and Challenges in Comprehensive Data Cleansing.".

[2] Jan L. Harrington, *Relational Database Design and Implementation: Clearly Explained*.: Morgan Kaufmann, 2009.

[3] W. Eckerson, "Data Profiling: A Tool Worth Buying (Really!).," 2004.

[4] Larry P. English, *Information quality applied : best practices for improving business information, processes, and systems.*, 2009.

[5] David Loshin, *Master Data Management*.: Morgan Kaufmann, 2009.

[6] Codeplex. [Online]. http://dbschemareader.codeplex.com/