

TecnoDB una Base de Datos Relacional y Prometeo un metodo de aprovechamiento de Metadatos y Generador de Consultas

Iris Gastañaga
Erica Hintermeister
Calixto Maldonado
Cesar Martinez Spessot

Universidad Tecnológica Nacional
Departamento de Sistemas de Información
Laboratorio de Investigación de Software

Abstract

El presente trabajo describe brevemente el proyecto TecnoDB, iniciado en el año 2002, con acreditación de las Secretaria de Ciencia y Técnica de la Utn desde 2004 y finalizado en Diciembre de 2005. De ese proyecto se generó un grupo de trabajo para definir una heurística de explotación de metadatos y un software capaz de implementar esta metodología. Este conjunto de método y software ha sido denominado Prometeo y forma parte principal de este trabajo. Con el mismo nombre fue presentado y aprobado a la Agencia Córdoba Ciencia dentro del programa de Grupos de Reciente Formación año 2005. Contiene una síntesis del proyecto madre TecnoDB, que fue enviado y aprobado en la Wicc realizada en Neuquén, ahora con este apoyo de la Wicc y de la UTN Fac. Regional Córdoba, estamos difundiendo nuestra actividad y el producto de la misma. Este trabajo presentara luego del resumen de TecnoDB una síntesis de la Heurística y la descripción de la herramienta que esta siendo desarrollada, denominada Prometeo. Es para destacar la importancia del apoyo brindado por la Agencia Cordoba Ciencia y la Univ. Tecnologica Nacional a este grupo de trabajo para la continuidad de la tarea iniciada.

Palabras Clave

Bases de Datos relacionales, RDBMS, Sql generado, Metadatos.

1-TecnoDB

TecnoDB es un motor de bases de datos relacional experimental que actualmente atiende la administración básica de una base de datos, que permite conocer en forma práctica la resolución de consultas (select, insert, update, delete, etc.) de una tabla específica de la base de datos. Además esta actualmente en desarrollo, un modelo de datos que almacena Metadatos (información sobre las tablas, columnas, constraints o relaciones, índices existentes en la Base de datos),.

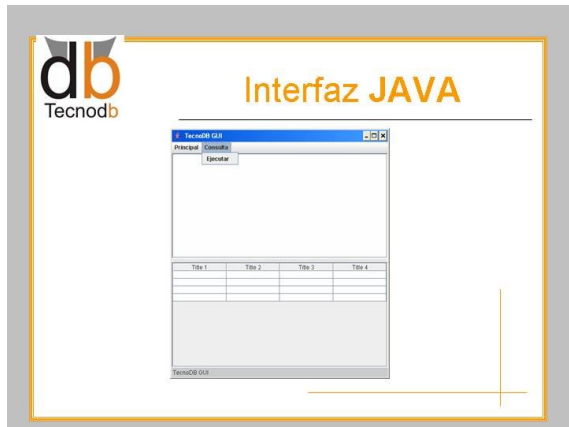
Una de las principales características para extender el motor es la investigación y desarrollo del módulo de Log de Operaciones y Transacciones en la cual se proporcionará información sobre tiempos de búsqueda, inserción, borrado y actualización de tuplas en las tablas de la Base de Datos, conjuntamente con comparaciones de los mismos utilizando diferentes índices y sin valerse de éstos a los efectos de que pueda realizarse comparaciones.

Con el objetivo de probar diferentes algoritmos de mejoras de performance se esta tratando de definir arquitectónicamente una capa de abstracción de algoritmos de indexación a fin de poder tener una interfaz común para la implementación e investigación de nuevas formas de ordenamiento de claves.

El motor cuenta con servicios de red mediante el protocolo de TCP/IP y permite la transmisión de consultas y sus respectivos resultados con XML. Este módulo se extenderá a fin de proveer Web Services. Otro aspecto planeado para este proyecto es posibilitar el acceso al motor desde aplicaciones se desarrollará un driver JDBC.

Se esta desarrollando una interfaz gráfica en Java para facilitar la creación de consultas en lenguaje SQL, trabajándose actualmente para que use el socket programado, que traduzca a XML, la consulta ingresada y que traduzca la respuesta de TecnoDB en XML y ubique los datos en la grilla de respuesta de la interfaz.

En la figura siguiente se muestra el boceto de la interfaz grafica:



Durante el proyecto, creció la percepción sobre la falta de herramientas que asistan al usuario de nuestro motor para escribir sentencias SQL en forma automática, basadas en la información residente en el diccionario de Datos de la Base de Datos objetivo. Este es el origen del sub proyecto Prometeo, descrito a continuación.

2-El subproyecto Prometeo:

Introducción:

Trabajando con otras bases de datos comerciales se observo que existen muchas herramientas que auxilian al desarrollador y usuarios avanzados, a construir las consultas requeridas en una aplicación. El conocimiento de los diccionarios de datos disponibles en esa base de Datos nos ayudo a desarrollar un método que avance mas allá de las herramientas conocidas, que consiste en definir que consultas por defecto pueden construirse automáticamente y guardarse en un modelo de tablas simple, para su posterior aprovechamiento.

Hasta que TecnoDB se establezca como producto y desarrolle su diccionario donde guardar los metadatos, trabajamos sobre PostgreSQL y vamos a avanzar sobre Oracle

Este sub-proyecto deberá cumplir los siguientes objetivos.

3-Objetivos Generales

Desarrollar el software que permita realizar lo indicado en la Heurística, es decir, leer los metadatos de la base de datos elegida y como propuesta de mínima, ayudar al desarrollador a construir código de sentencias de consulta y de actualización sin errores y guardar esa sentencia en el mismo diccionario. Como propuesta de máxima que ante el pedido del desarrollador, repase las tablas elegidas por este y genere las consultas que pueden derivarse de la información en los metadatos sobre tipos de datos, de claves foráneas y construya vistas con las sentencias SELECT generadas y grupos de sentencias DML de carga de datos generados automáticamente para pruebas de aplicaciones.

Otro objetivo general es avanzar en desarrollar aun más las definiciones de la heurística, buscando más oportunidades de extraer el mayor beneficio de los metadatos almacenados en el diccionario de la Base de Datos objetivo.

4-Avances Logrados

Heurística

El desarrollo de la Heurística se inicio en mayo del 2005, y su presentación fue realizada en la 34 Jaiio, a finales de Agosto en Rosario, en el Simposio AST 2005.

La heurística propone que como el lenguaje SQL puede ser escrito por la sentencia SELECT, utilizando literales alfabéticos para las palabras reservadas y las columnas del diccionario como contenedores de nombres de columnas y de tablas podemos lograr que un generador de consultas al ejecutar la sentencia:

(Ejemplo usando Oracle)

```
SELECT 'select * '||table_name||';' FROM user_tables;
```

Produzca, si el usuario tiene dos tablas a su nombre (Tabla1 y Tabla2):

```
Select * from tabla1;  
Select * from tabla2;
```

Este es el principio inicial, que permite automatizar la creación de código SQL, usando sentencias Select del mismo SQL.

Se avanzó entonces en el enunciado de que consultas se podrían generar automáticamente con esta forma de escribir sql y se definieron 21 pautas para extraer consultas interesantes.

Una de ellas define que las sentencias select generadas, se deben guardar como vistas en el mismo diccionario, agregando un nombre también derivado que describa sintéticamente, de que se trata cada vista, usando el prefijo 'get_' la columna y la tabla usada.

Esto está tomado de la regla de estilo en JAVA de los métodos asociados a los atributos privados de un objeto.

Desarrollo del Software

Luego de la presentación se inició el desarrollo del software en lenguaje PHP, y sobre la base de datos PostgreSQL. Las siguientes figuras nos mostrarán parte del trabajo realizado en la primera versión.

IMPLEMENTACIÓN

- **Lenguaje de Programación:**
 - Php 5.0.4
 - HTML
- **Motor de base de datos:**
 - PostgreSQL 8.0
- **Servidor Web:**
 - Apache 2.0.53
- **Herramienta de desarrollo:**
 - Dreamweaver MX 2004

Fig.1: Presentación de las características técnicas del Generador de Consultas

INTERFACES

Motor: PostgreSQL
 Base de Datos: prometeo
 Esquema: public
 Usuario: erica
 Contraseña: *****

Aceptar Limpiar

Fig. 2: Pantalla de selección de motor de bases de datos, nombre, esquema y la identificación del usuario para acceder a las tablas de ese esquema.

INTERFACES

Selección de tabla y tipo de consulta:

USUARIO: ERICA - BASE DE DATOS: PROMETEO
 TABLAS (4): lineas

ELIJA SU CONSULTA:
 SELECCIONAR COLUMNAS
 GENERAR TODAS LAS CONSULTAS POSIBLES

Seguir Volver

Fig. 3: Una vez identificado puedo elegir la tabla principal sobre la que realizaremos las consultas, y también seleccionar el modo de trabajo, eligiendo o generando todas las consultas posibles.

INTERFACES

Selección de campos y personalización:

ESTRUCTURA DE LA TABLA
 LINEAS

SELECCIONAR/DESSELECCIONAR TODOS

| COLUMNA | TIPO DATO | ORDEN | DESCENDENTE | ALIAS |
|--|-----------|-------|---|------------|
| <input checked="" type="checkbox"/> project_id | int4 | 1 | <input checked="" type="checkbox"/> Descendente | NoProyecto |
| <input checked="" type="checkbox"/> query_id | int4 | 2 | <input checked="" type="checkbox"/> Descendente | NoConsulta |
| <input checked="" type="checkbox"/> line_id | int2 | 3 | <input checked="" type="checkbox"/> Descendente | NoLinea |
| <input checked="" type="checkbox"/> type | varchar | 4 | <input checked="" type="checkbox"/> Descendente | TipoDato |
| <input checked="" type="checkbox"/> text | varchar | 5 | <input checked="" type="checkbox"/> Descendente | TextoSql |

Generar Consulta Volver

Fig. 4: Por ahora la herramienta esta permitiendo que el usuario elija las columnas a mostrar y en que orden se mostraran las filas, es decir la cláusula 'Order by' y por ultimo se pueden definir los alias de columnas

INTERFACES

Consulta formulada y ejecutada:

```
SELECT PROJECT_ID AS "NroProyecto", QUERY_ID AS "NroConsulta", LINE_ID AS "NroLinea", TYPE AS "TipoDato", TEXT AS "TextoSql" FROM LINEAS ORDER BY PROJECT_ID ASC, QUERY_ID ASC, LINE_ID ASC, TYPE ASC, TEXT ASC
```

RESULTADOS DE LA CONSULTA

| NroProyecto | NroConsulta | NroLinea | TipoDato | TextoSql |
|-------------|-------------|----------|----------|------------|
| 2 | 4 | 2 | SELECT | query_id |
| 2 | 4 | 2 | SELECT | line_id |
| 2 | 4 | 3 | SELECT | type |
| 2 | 1 | 1 | SELECT | text |
| 2 | 1 | 5 | SELECT | project_id |
| 2 | 1 | 6 | FROM | linea |
| 3 | 1 | 7 | SELECT | query_id |
| 3 | 1 | 8 | SELECT | line_id |
| 3 | 1 | 9 | SELECT | type |
| 3 | 1 | 10 | SELECT | text |

Fig. 5: El resultado, en amarillo, es la sentencia generada por Prometeo y en la tabla de Resultados las filas obtenidas al ejecutarla contra los datos reales

Estamos trabajando para mejorar el recupero de las consultas guardadas y para ejecución de las vistas creadas por el generador.

Hasta el momento Prometeo permite cumplir algunos puntos de la Heurística planteada y el avance del proyecto apunta a cumplimentar todos los definidos hasta el momento.

Los próximos pasos son cubrir los puntos de la Heurística, construir la funcionalidad sobre la base de datos Oracle, estudiar la factibilidad de hacerlo sobre SQLServer 2005. En otro sentido y debido al origen de este proyecto, se aportará al diseño del diccionario de TecnoDB para enriquecer el proyecto con una herramienta que facilite el trabajo a sus usuarios.

5-Conclusiones

Desde la definición de la Heurística nos esforzamos en la concreción de la interfase y que interactúe con un diccionario de datos ya funcionando y los resultados logrados nos estimulan a seguir con otros diccionarios de datos de bases de datos comerciales, sin olvidar que se deberá atender en el momento que el proyecto madre, TecnoDB inicie el diseño del almacén de datos de sus metadatos.

Es necesario seguir agregando pautas a la heurística para aumentar las posibilidades de uso de los metadatos almacenados en los diccionarios de las bases de datos.

En el software se esta documentando lo realizado y buscando la mayor modularizacion de la programación para poder atender a otros motores de bases de datos y aumentar el reuso de lo construido.

6-Bibliografía:

[Ullman-1997] ULLMAN, J. D. - Widom, J. (1997) "A first course in database System", Prentice Hall,

[Leonidovich-1996] Leonidovich, K. M. (1996) "Research and Development of a Languaje Subsystem SQL", Russian Academy of Sciences, Institute of System Programming, Moscow

[Date-1990] DATE, C. J. (1990) "An Introduction to Database Systems", Addison-Wesley Publishing Company, Vol I.

[Silberchatz-1998] Silberschatz, A. – Korth, H.F. – Sudarshan, S. (1998) "Database System Concepts", Mc Graw-Hill, 3 Edition.

[Aho-1990] Aho, A. V. – Sethi, R. – Ullman, J. (1990) "Compiladores Principios, Técnicas y Herramientas", Addison-Wesley Iberoamericana

[Langsam-1997] Langsam, Y. – Augentein, M. J. - Tennenbaum, A. M. (1997) "Estructura de Datos con C y C++", Prentice Hall, 2 Edición.

[Chorafas-1983] Chorafas, D. N. (1983) "DBMS for Distributed Computers and Networks", Petrocelli Books

[Elmasri-2002] Elmasri, R.- Navathe, S. (2002) "Fundamentals of Database Systems", Addison-Wesley, 3rd edition.

[Ramakrishnan-2000] Ramakrishnan, R. – Gehrke, J. (2000) "Database Management System", McGraw-Hill, , 2nd edition

Novática Número 140 Bases de Datos. Julio/Agosto 1999

ACM Queue. Numero

PAginas web consultadas:
Página web visitadas período agosto-diciembre de 2005. Enero- Mayo2006

<http://www.ucse.edu.ar/fma/compiladores/>)

<http://www.ispras.ru/~kml/gss/>

<http://sal.kachinatech.com/H/1/>

<http://leap.sourceforge.net/>

<http://www.jepstone.net/tinySQL>

<http://amssoft.ru/easysql>

<http://macine.epublish.cl/tesis/index.html>

<http://www.lsi.us.es/colabora/fieware/programas/sbase.html#basica>

<http://www.cindoc.csic.es/isis/indice.htm>

<http://members.tripod.com/~MoisesRBB/sql.html>

<http://www3.uji.es/~mmarques/f47/apun/>

http://www.cicese.mx/~orodrigu/database/resumen_02.htm

http://www.cicese.mx/~orodrigu/database/resumen_01.htm

<http://espanol.geocities.com/michelsoftpe/bdatos.htm>

<http://www.dsic.upv.es/~jorallo/docent/BDA/DisiplinaBD.pdf>

<http://www.monografias.com/trabajos13/trsqlinf/trsqlinf.shtml#BREVE>

<http://es.tldp.org/Postgresql-es/web/navegable/tutorial/sql.html>

<http://www.odmg.org/>

<http://java.sun.com/products/jdbc/index.html>

<http://www.w3.org/XML/Query>