Revista Cubana de Ciencias Informáticas (RCCI)
ISSN: 1994-1536 | RNPS: 0547
http://rcci.uci.cu | rcci@uci.cu

Tipo de artículo: Artículo original
Temática: Sistema de bases de datos
Recibido: 5/4/2011 | Aceptado: 13/5/2011| Publicado: 29/9/2011

CRUD-PG

Anthony R. Sotolongo León[1*], Marianela Gutierrez Rodriguez[2] y Beatriz Piñeiro Veitia[3].

[1]* Centro de Tecnologías de Gestión de Datos (DATEC). Universidad de las Ciencias Informáticas (UCI). Carretera a San Antonio de los Baños Km. 2 ½, Boyeros, La Habana, Cuba. asotolongo@uci.cu

[2] Centro de Tecnologías de Gestión de Datos (DATEC). Universidad de las Ciencias Informáticas (UCI). Carretera a San Antonio de los Baños Km. 2 ½, Boyeros, La Habana, Cuba. mgrodriguez@uci.cu

[3] Centro de Tecnologías de Gestión de Datos (DATEC).). Universidad de las Ciencias Informáticas (UCI). Carretera a San Antonio de los Baños Km. 2 ½, Boyeros, La Habana, Cuba. bpineiro@estudiantes.uci.cu

Resumen: en el presente trabajo se desarrolla una herramienta llamada CRUD-PG la cual permite desarrollar parte de la lógica de negocio en la base de datos PostgreSQL a través de funciones (procedimientos almacenados) programados en PL/pgSQL, que realizan operaciones CRUD del inglés Create, Retrieve, Update y Delete, logrando así mayor rapidez en la interacción con la base de datos y de desarrollo de aplicaciones. Esta herramienta está implementada en java.

Palabras claves: desarrollo de aplicaciones; lógica de negocio; procedimientos almacenados; Pl / pgSQL, PostgreSQL

Abstract: *in this work is developed a CRUD-P toll, this allows the development of business logic in the database PostgreSQL via functions (stored procedures) programmed in PL / pgSQL, execute transactions CRUD (in English: Create, Retrieve, Update and Delete) ,thus achieving faster interaction with database development and software implementation, this tool is implemented in Java.*

Keywords: *business logic, stored procedures, Pl / pgSQL, PostgreSQL.*

1. Introducción

La mayor cantidad de las aplicaciones de software que se desarrollan en la actualidad tienen entre sus características la interacción con un sistema de gestión de bases de datos, generalmente estas aplicaciones acceden a los datos para realizar operaciones sobre ellos, principalmente operaciones de tipo CRUD (del inglés Create, Retrieve, Update y Delete). Comúnmente estas operaciones se implementan en la capa de negocio de las aplicaciones. Incluso existen algunos framework que generan estas operaciones, liberando así a los programadores de esta etapa del desarrollo de las aplicaciones, pero son creadas con el mismo código con el que están implementados, es decir que estas operaciones quedan en el negocio de los sistemas, algunos de los más utilizados son symfony, hibertane y kumbiaphp, entre otros. Teniendo así que procesar resultados intermedios en el cliente para obtener el resultado final lo cual ralentiza el proceso de intercambio de datos entre el cliente y el servidor. Además tampoco permite a aplicaciones desarrolladas en diferentes lenguajes de programación acceder a esta lógica de negocio, teniendo así que repetir el código de estas operaciones básicas.

Los principales gestores de base de datos que existen en la actualidad soportan la implementación de procedimientos almacenados y funciones, independientemente de que estos sean gestores comerciales o libres, lo cual posibilita realizar operaciones dentro de los gestores, es decir incluir lógica de negocio dentro de las bases de datos, dentro del grupo de los comerciales se encuentran ORACLE, MS SQL Server, etc. y liderando el grupo de los libres esta PostgreSQL.

Aprovechando las potencialidades de este gestor de código abierto un grupo de especialistas y operadores de PostgreSQL crearon la Comunidad Técnica Cubana de PostgreSQL, que tiene entre sus objetivos: Contribuir al fortalecimiento de la soberanía tecnológica cubana, desde el enfoque del desarrollo de tecnologías de bases de datos tomando como base a PostgreSQL.

Por lo cual se propone como objetivo desarrollar una herramienta (CRUD-PG) la cual permita estandarizar y obtener los procedimientos almacenados y/o funciones que realicen operaciones CRUD sobre bases de datos PostgreSQL, y así proveer a las aplicaciones que usen dichas Bases de Datos de operaciones CRUD centralizadas en el mismo clúster y lograr mejores tiempos de respuestas entre las aplicaciones y el servidor de base de datos PostgreSQL.

2. Materiales y Métodos

Para el desarrollo de la herramienta CRUD-PG se utilizaron las siguientes metodologías y herramientas:

Metodología XP: es una metodología de desarrollo de software ágil con un desarrollo iterativo e incremental. Algunas de las particularidades más significativas de esta metodología es que propone la programación en pares o pareja y que el cliente forma parte del equipo de desarrollo, facilitando esto grandemente la comunicación entre todos los miembros del equipo de desarrollo y el cliente, permitiendo que el mismo esté en todo momento presente y de acuerdo con todos los cambios por los que pase el sistema en desarrollo (Kent, 1999).

PostgreSQL 8.4: Versión del gestor liberada en el 2009 con más de 250 mejoras respecto a sus anteriores versiones, entre las mejoras más populares se encuentra (PostgreSQL Group ,2009):

- Restauración de bases de datos en procesos paralelos, que acelera recuperación de un respaldo hasta 8 veces.
- Privilegios por columna, que permiten un control más granular de datos confidenciales.
- Configuración de ordenamiento configurable por base de datos, lo cual hace a PostgreSQL más útil en entornos con múltiples idiomas.
- Nuevas herramientas de monitoreo de consultas que le otorgan a los administradores mayor información sobre la actividad del sistema.

Pl/pgsql: es un lenguaje estructurado del gestor de base de datos PostgreSQL. Permite ejecutar comandos SQL en el gestor, se pueden realizar cálculos complejos, dispone de estructuras de control repetitivas y condicionales. Posibilita además el trabajo con las funciones (PostgreSQL Group,2009).

Java: lenguaje de programación orientado a objetos, independiente de arquitecturas hardware, sistemas operativos y sistemas de ventanas. Utiliza el concepto de maquina virtual (VM). Un programa nativo: la VM se encarga de traducir este código para que cualquier ordenador pueda ejecutarlo. De esta manera un código generado en Java puede correr en cualquier plataforma, en donde se haya portado la VM (Reges and Stepp, 2010).

Maquina Virtual de Java 1.6: es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (bytecode), el cual es generado por el compilador del lenguaje Java (Reges and Stepp, 2010).

NetBeans 6.8: es un IDE de código abierto ("open source") diseñado para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java. Es una herramienta pensada para escribir, compilar, depurar y ejecutar programas (Boeck, 2009).

3. Resultados y discusión

3.1 Descripción de la API

La herramienta CRUD-PG tiene su tema centrado en las base de datos de servidores PostgreSQL, específicamente en las operaciones básicas que se realizan (escritura, lectura, actualizar y eliminar).

Generando para ello funciones estándares capaces de realizar dichas operaciones, escritas en lenguaje **pl/pgsql**. Y así poder implementar parte de la lógica del negocio en la base de datos. Es decir que en cada base de datos donde se aplique la misma se generarán funciones capaces de operar los datos, permitiendo a los programadores hacer uso de las mismas sin tener que preocuparse por implementarlas en sus aplicaciones, además estas funciones son independiente del el lenguaje en que están programadas dichas aplicaciones.

3.2 Nombre de las funciones

Para que los programadores puedan utilizar las funciones se debe definir los nombres estándares de las mismas. A continuación se detalla el estándar:

3.2.1 Función "*Select*"

La función *Select* se generará una por cada tabla de la base de datos y su nombre estará compuesto por el nombre de la tabla seguido de un "_select" y devuelve los registros de la tabla con todos los atributos de la misma.

Ejemplo: tabla "tbpersona", función "Select" tbpersona_select ().

3.2.2 Función "*Select Condicional*"

La función *Select Condicional* se generará una por cada tabla de la base de datos y su nombre estará compuesto por el nombre de la tabla seguido de un "_select_condicional". Se le pasa como parámetro la condición que se necesita de la misma tabla, devuelve los registros de la tabla con todos los atributos de la misma.

Ejemplo: tabla "tbpersona", función "Select Condicional" tbpersona_select_condicional ('id_centro =1 and edad>65').

3.2.3 Función "*Select General*"

La función *Select General* se generará para uso general, es decir si necesita una consulta donde intervienen varias tablas podrá utilizar esta función. La función se llamará "select_general" .Se le pasa por parámetro la consulta, devuelve un tipo record del catálogo de postgresql. Cuando la ejecuten deben especificar lista de definición de columnas para funciones que retornan «record».

Ejemplo: función *"Select General" select_general ('select * from tbpersona, tbcetro where tbpersona.idcentro=tbcentro.idcentro')*

3.2.4 Función "*Delete*"

La función *Delete* se generarán una por cada tabla de la base de datos y su nombre estará compuesto por el nombre de la tabla seguido de un "_delete". Se le debe pasar como parámetro la condición de eliminación.

Ejemplo: tabla "tbpersona", función "Delete" tbpersona_delete ('id_persona=1').

3.2.5 Función "*Delete Truncate*"

La función *Delete Truncate* se generará para eliminar todos los registros de una tabla. La función se llamará "delete_truncate" Se le pasa por parámetro la tabla de la cual se desea eliminar todos sus registros.

Ejemplo: función *""Delete Truncate"* delete_truncate ('tbpersona').

3.2.6 Función "*Update*"

La función *Update* se generará una por cada tabla de la base de datos y su nombre estará compuesto por el nombre de la tabla seguido de un "_update". Se le pasan dos parámetros, uno dónde van los campos a actualizar con sus nuevos valores y otro especifica las condiciones de actualización.

Ejemplo: tabla "tbpersona", función "Update" tbpersona_update ('id_centro =2', 'id_centro=1').

3.2.7 Función "*Insert*"

La función *"Insert"* se generará una por cada tabla de la base de datos y su nombre estará compuesto por el nombre de la tabla seguido de un "_insert". Se le pasa un parámetro que son los valores separados por coma en el orden en que son definidos en la tabla.

Ejemplo: tabla tbpersona función "Insert" tbpersona_insert ('55, ''Juan'', ''Perez'',1').

```
CREATE TABLE "public"."Tbpersona" (
  "id_persona" INTEGER NOT NULL,
  "nombre" VARCHAR,
  "apellidos" INTEGER,
  "centro" INTEGER,
  CONSTRAINT "tbpersona_pkey" PRIMARY KEY ("id_persona")
) WITH OIDS;
```

3.2 Detalles de las funciones:

La API generara una cantidad de funciones igual a la cantidad de tablas multiplicadas por cinco más dos, es decir:

Total de funciones=cantidad de tablas*5 +2

Tabla 1: Funciones y parámetros

Nombre de funciones	**Detalles de parámetros de la funciones**
nombretabla_select	Nombretabla_select()
nombretabla_select_condicional	Nombretabla_select_condicional(condicional text)
Select_general	selectgeneral(consulta text)
nombretabla_delete	nombretabla_delete(condicional text)
Delete_truncate	deletetruncate(tabla text)
nombretabla_update	nombretabla_update(valorset text, valorwhere text)
nombretabla_insert	nombretabla_insert(valores text)

3.3 Interfaz del CRUD-PG 1.0

Se confeccionó una interfaz capaz de interactuar con las bases de datos y permita implementar las funciones que realizarían las operaciones CRUD en cada una de ellas. Esta interfaz (figura 1) permite la entrada los parámetros de conexión en la parte superior, en el centro muestra un cuadro con las tablas de la base de datos a la cual se ha conectado, y en la parte inferior tiene una barra de progreso la cual indica el estado de la generación de las funciones.

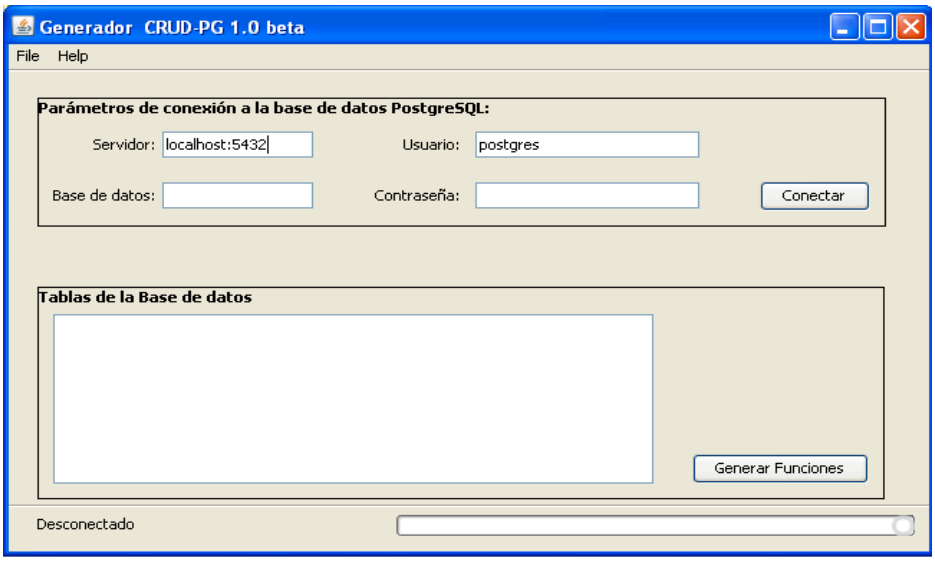

Figura 1. Interfaz visual de CRUD-PG 1.0

4. Conclusiones

Con la herramienta CRUD-PG se logra desarrollar parte de la lógica de negocio (operaciones CRUD) dentro del gestor de base de datos PostgreSQL a través de funciones y/o procedimientos almacenados. Además facilita a los programadores que lo utilicen, el trabajo con la base de datos y su mantenimiento pues se establece un estándar para el nombre de las funciones. Se provee de una interfaz de acceso a los datos dentro de la base de datos postgreSQL independiente del lenguaje de programación de la aplicación que interactúe con la base de datos. Agiliza las operaciones con la base de datos PostgreSQL.

5. Referencias bibliográficas

Boeck Heiko. *The Definitive Guide to NetBeans Platform,*. May 29, 2009

Kent Beck. *Extreme Programming Explained: Embrace Chang*e. Addison Wesley

PostgreSQL Global Development Group. 2009. *PostgreSQL 8.4.1 Documentation.* California: s.n, 2009.

Reges Stuart and Stepp Marty. *Building Java Programs: A Back to Basics Approach (2nd Edition),* Mar 21, 2010