



UNIVERSIDAD NACIONAL AUTONÓMOMA DE NICARAGUA, MANAGUA
UNAN-MANAGUA
CONGRESO DE LA RED DE COMPUTACIÓN PARA EL DESARROLLO,
COMPDES 2014

Del 23 al 25 de julio de 2014

“Investigando e Innovando para el Desarrollo Tecnológico”

LAS TABLAS RECURSIVAS EN SQL SERVER 2008

Eddy Antonio Arias Silva (ariassilva01@yahoo.com), Erick Gustavo Cruz Pérez (ercruz@unan.edu.ni), Harold Ramiro Gutiérrez Marcenaro (haroldgut@yahoo.com), María Catalina Tapia López (kathytapia7@hotmail.com)

RESUMEN

Se presenta una revisión teórica de la recursividad o recursión como una de las técnicas más potentes de la programación pura, que genera códigos altamente compactos y eficientes. Se describe también como este concepto se aplica al diseño de bases de datos relacionales, particularmente en SQL Server, en aquellos casos en que existe una relación jerárquica implícita entre los registros de una tabla.

PALABRAS CLAVE

Recursividad, Normalización, Bases de Datos.

I. INTRODUCCIÓN

El concepto de recursividad se aplica en la programación de computadoras en aquellos casos en que la solución a un problema determinado está expresada en términos del mismo problema. El caso más representativo de la recursión es el

cálculo del factorial de un número entero N. Si queremos calcular el factorial de 5, este puede expresarse de la siguiente manera:

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$

o lo que es igual

$$5! = 5 \times 4!$$

Esto genera un nuevo nivel en términos de la misma problemática. Puede seguirse esta línea de planteamientos de la siguiente manera:

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 5 \times 4!$$

$$4! = 4 \times 3 \times 2 \times 1 = 4 \times 3!$$

$$3! = 3 \times 2 \times 1 = 3 \times 2!$$

$$2! = 2 \times 1 = 2 \times 1!$$

$$1! = 1 \times 0!$$

$$0! = 1$$

Se llega a un nivel en donde se encuentra un estado básico, es decir un valor para el que la solución al problema es un número concreto, en este caso el factorial de cero, que por definición matemática es igual a uno.

La ejecución de una función recursiva lleva al compilador a generar una pila que guarda todo el estado del programa en cada momento en que se invoca recursivamente a la función. Esto permite, una vez encontrado el estado básico, ir recuperando valores de la pila, en el orden inverso al que fueron ingresados, hasta dar solución al problema original. Esto genera un proceso en espiral de doble vía, primero desde el planteamiento original adentrándose hasta el estado básico, luego desde el estado básico, consiguiendo soluciones parciales, hasta lograr la solución del planteamiento original. Para el caso del factorial tendríamos:

$$0! = 1$$

$$1! = 1 \times 0! = 1 \times 1 = 1$$

$$2! = 2 \times 1! = 2 \times 1 = 2$$

$$3! = 3 \times 2! = 3 \times 2 = 6$$

$$4! = 4 \times 3! = 4 \times 6 = 24$$

$$5! = 5 \times 4! = 5 \times 24 = 120$$

A continuación se muestra el código C para resolver recursivamente el factorial de un entero N.

```
int factorial(int N)
{
    if(N==0)
        return(1);
    else
        return(N*factorial(N-1));
}
```

Hay muchos otros problemas que están definidos recursivamente, entre ellos los más conocidos son: Los números de Fibonacci, Las Torres de Hanoi, El cálculo del Máximo Común Divisor, entre otros. También hay estructuras de datos que por su naturaleza están definidas recursivamente, entre ellas están las listas, y especialmente los árboles. De hecho tanto las listas como los árboles están basadas en estructuras auto referenciadas.

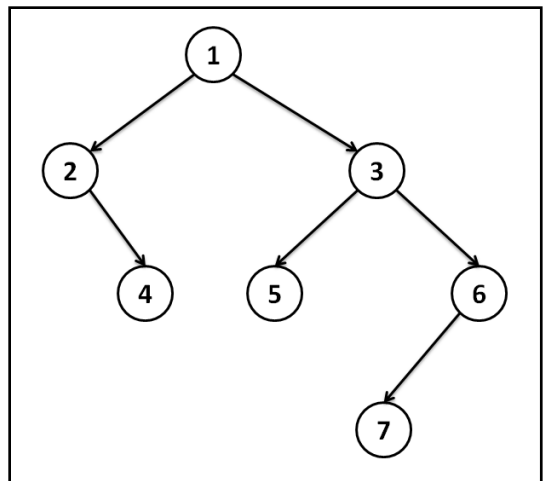


Ilustración 1. Árbol Binario de Números Enteros

La recursión es una técnica ampliamente utilizada, altamente valiosa, que requiere gran destreza y pensamiento lógico por parte del programador. Sin embargo su aplicación en el área del diseño de bases de datos es relativamente nueva y su referencia bibliográfica es escasa.

Una tabla recursiva, es en términos generales, una tabla en donde un campo (no llave) está relacionado con el campo llave de la misma. Esto se aplica a relaciones de jerarquía, como el organigrama de una organización o una relación genealógica, entre otros casos.

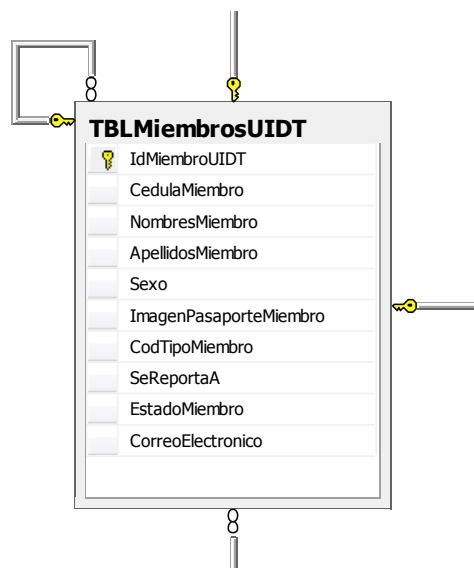


Ilustración 2. Tabla recursiva en SQL Server 2008

Para el caso de los registros de esta tabla tenemos los siguientes datos:

	IdMiembroUIDT	NombresMiembro	ApellidosMiembro	SeReportaA
▶	1	Pedro Alberto	Aburto Jarquín	NULL
	2	Sergio Antonio	Vado Conrado	1
	3	Harold Ramiro	Gutiérrez Marce...	2
	4	Erick Gustavo	Cruz Pérez	3
	5	Oscar Ramón	Fletes Calderón	3
	6	Eddy Antonio	Arias Silva	3
	7	María Catalina	Tapia López	3
	8	Darío Benjamín	Rodríguez Martínez	3
*	NULL	NULL	NULL	NULL

Ilustración 3. Datos de la Tabla Recursiva.

Lo cual representa el siguiente organigrama.

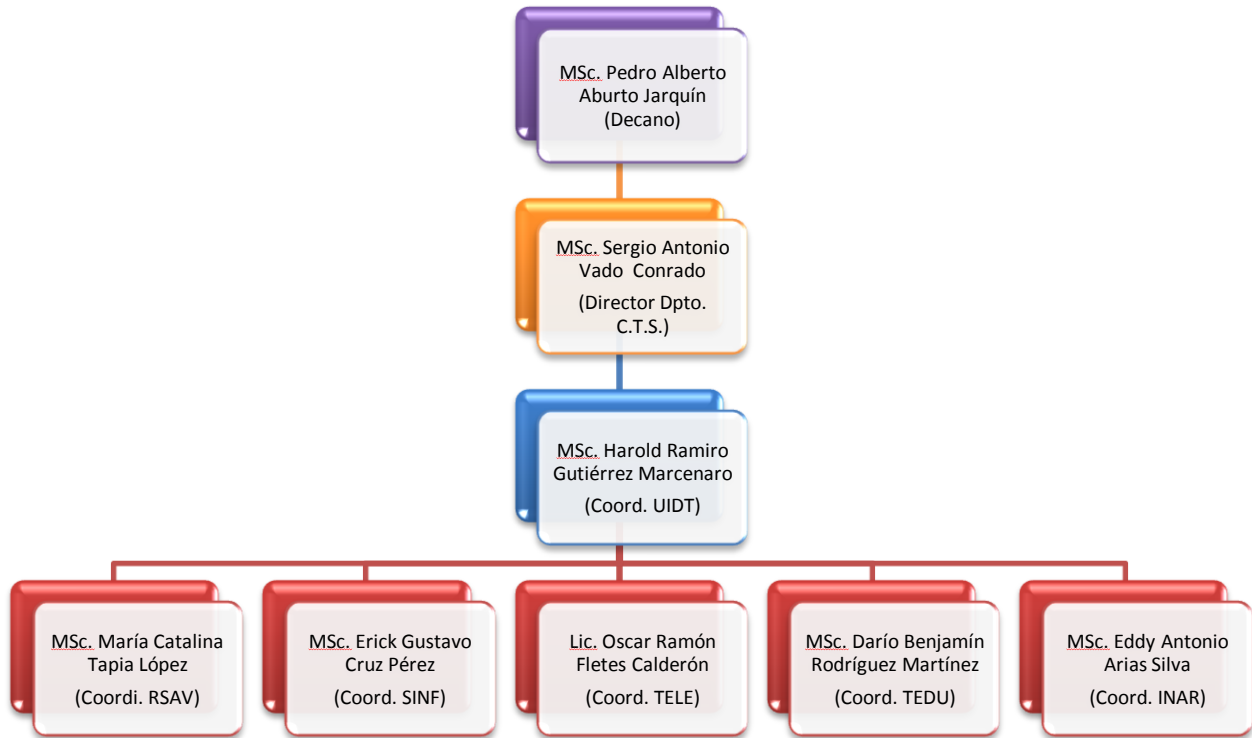


Ilustración 4. Organigrama derivado de la Tabla Recursiva.

El presente trabajo consiste en la revisión práctica de diversos casos de aplicación de las tablas recursivas en Microsoft SQL Server 2008, y la creación de una interfaz gráfica en Visual C Sharp para la representación jerárquica de los datos de las tablas recursivas.

II. OBJETIVOS

1. Realizar una revisión teórica y práctica de las tablas recursivas en SQL Server 2008 y sus casos de aplicación.

2. Desarrollar una interfaz sencilla de visualización jerárquica de tablas recursivas por medio de Visual C Sharp.

III. MARCO TEÓRICO

La recursión o recursividad, nos explican (Cairó & Guardati, 2006) : “es un concepto amplio, con muchas variantes, y difícil de precisar con pocas palabras. Aparece en numerosas actividades de la vida diaria; por ejemplo, en una fotografía donde se observa otra fotografía. Otro caso ilustrativo es el que se presenta en los programas de televisión, en los cuales un periodista transfiere el control de la noticia a otro periodista que se encuentra en otra ciudad, y éste a su vez, hace lo mismo con un tercero. Cuando este último termina su participación regresa el control al segundo, y cuando éste también finaliza su intervención regresa el control al primero”. (pág. 110).

La recursión puede ser directa, cuando una función se llama a si misma desde su cuerpo; o indirecta, cuando la función se llama indirectamente a través de otra función o una cadena de funciones.

Existen tres condiciones que debe cumplir un problema para considerarse e implementarse recursivamente:

1. Que exista una forma general o “paso recursivo” en el cual la solución del problema está expresada en términos de sí mismo.
2. Que exista un estado básico en el cual la solución es directa.
3. Que exista un acercamiento paulatino hacia el estado básico.

En cuanto a la aplicación de la recursión o recursividad en el ámbito del diseño de las bases de datos, (Sánchez, 2004, pág. 19) nos explica: “Las relaciones recursivas se tratan de la misma forma que las otras, sólo que un mismo atributo puede figurar dos veces en una tabla como resultado de la transformación”.

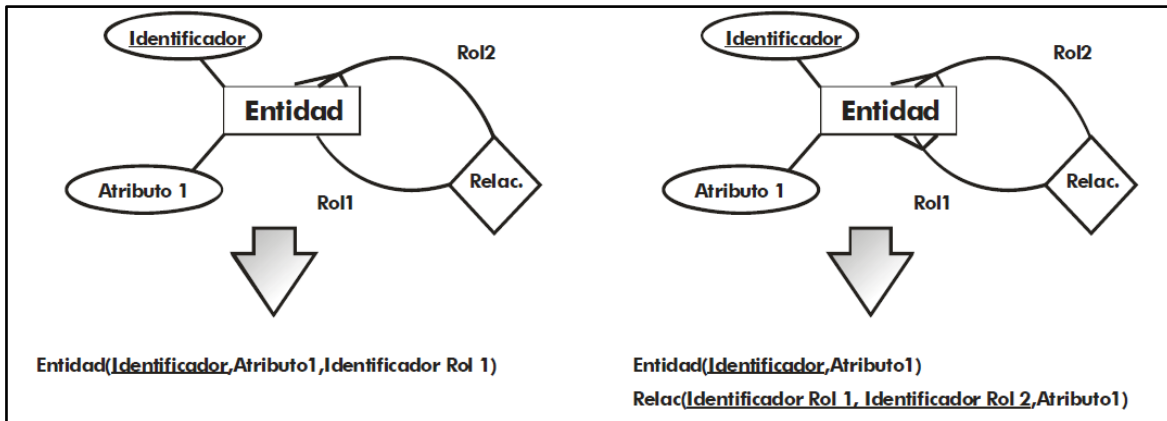


Ilustración 5. Representación gráfica de una relación recursiva. Fuente: (Sánchez, 2004)

Particularmente en SQL Server, la recursividad en tablas se maneja de la siguiente manera:

“Una expresión de tabla común (CTE) ofrece la gran ventaja de poder hacer referencia a sí misma, creando así una CTE recursiva. Una CTE recursiva es aquella en la que una CTE inicial se ejecuta varias veces para devolver subconjuntos de datos hasta que se obtenga el conjunto de resultados completo.

Se considera que una consulta es recursiva cuando hace referencia a un CTE recursiva. La devolución de datos jerárquicos es un uso frecuente de las consultas recursivas; por ejemplo, mostrar los empleados en un organigrama o los datos en un escenario de lista de materiales en donde un producto primario tiene uno o varios componentes que, a su vez, tienen subcomponentes o son componentes de otros elementos primarios.

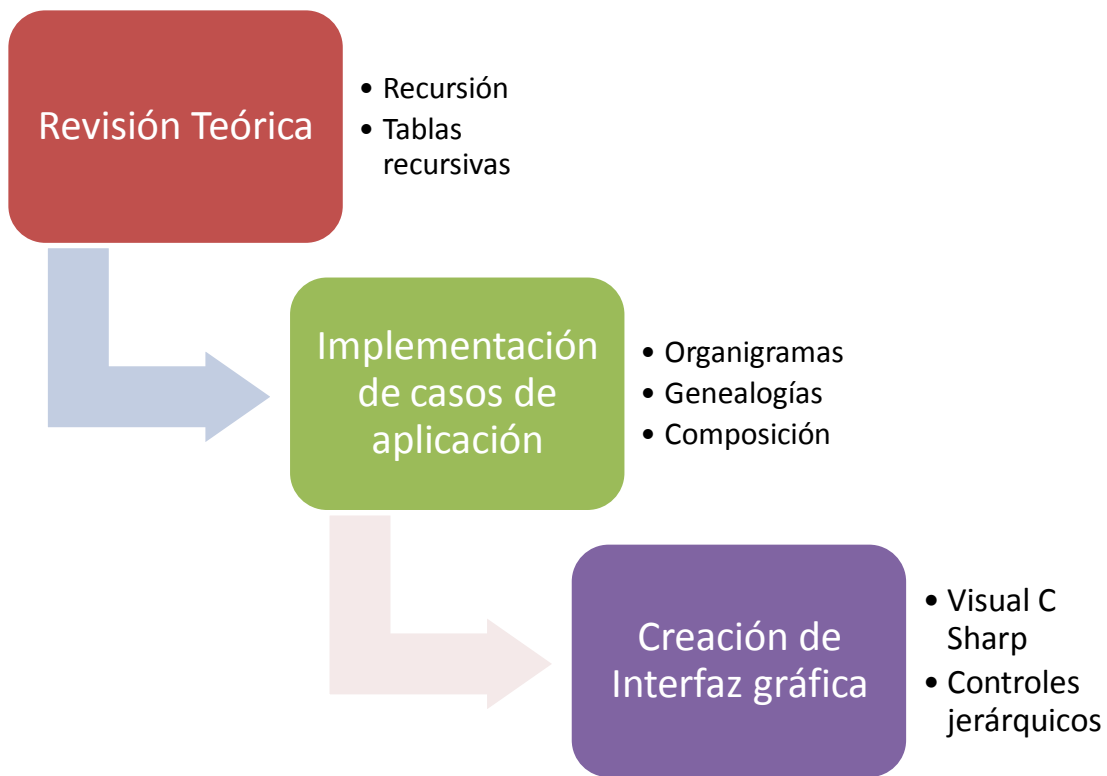
Una CTE recursiva puede simplificar en gran medida el código necesario para ejecutar una consulta recursiva en una instrucción SELECT, INSERT, UPDATE, DELETE o CREATE VIEW. En versiones anteriores de SQL Server, suele ser necesario que una consulta recursiva utilice tablas temporales, cursores y lógica para controlar el flujo de los pasos recursivos.”(SQL Server)

Una vez creada la tabla recursiva, pueden realizarse consultas recursivas como la siguiente:

```
--mostrar el jefe superior de cada empleado
withcte
as(selecte.IdEmpleado,e.NombreNombre,e.IdEmpleadojefe
fromEmpleadosewheree.Jefeisnull
unionallselecte.IdEmpleado,e.Nombrenombre,j.jefefromEmpleados
ase
innerjoincteasjone.Jefe=j.IdEmpleado)
selectc.Nombreempleado,j.NombrejefefromctecinnerjoinEmpleados
junc.jefe=j.IdEmpleado
```

IV. METODOLOGÍA DE TRABAJO

La metodología de trabajo ha contemplado las siguientes fases:



V. ANÁLISIS DE RESULTADOS

- a) Se realizó una revisión bibliográfica tanto en lo relativo a la recursión en sí misma como técnica de programación, como en lo referente a las tablas recursivas en SQL Server 2008, y programación en Visual C Sharp.
- b) Se implementaron y experimentaron diferentes casos de aplicación de tablas recursivas en SQL Server 2008. Entre estos casos los relativos a organigramas, genealogías y composición.
- c) Se programó una interfaz gráfica sencilla desde Visual C Sharp para el manejo de tablas recursivas en SQL Server 2008.

VI. CONCLUSIONES

- a) La recursión es una técnica poderosa que ha trascendido el ámbito de la programación pura, aplicándose ahora al diseño de bases de datos relacionales, facilitando así la creación de tablas relacionadas con estructuras jerárquicas de diferentes tipos.
- b) Las tablas recursivas son una forma altamente eficiente de evitar diseños de bases de datos con repetición de tablas básicamente similares, relacionadas entre sí.
- c) El manejo desde Visual C Sharp de bases de datos relacionales, con tablas recursivas, creadas en SQL Server 2008, utilizando controles gráficos de tipo jerárquico, permite la creación de aplicaciones sencillas y prácticas para una adecuada visualización de las estructuras jerárquicas.

VII. BIBLIOGRAFÍA

Cairó, O., & Guardati, S. (2006). *Estructuras de datos*. México D.F.: Mc. Graw Hill.

Sánchez, J. (2004). *Principios sobre bases de datos relacionales*. Recuperado el 19 de Junio de 2014, de <http://www.jorgesanchez.net/bd/bdrelacional.pdf>

SQL Server. (s.f.). Recuperado el 19 de Junio de 2014, de <http://technet.microsoft.com/es-es/library/ms186243%28v=sql.105%29.aspx>