

HERRAMIENTA ASISTIDA POR COMPUTADORA PARA LA ENSEÑANZA DEL ÁLGEBRA RELACIONAL EN BASES DE DATOS

COMPUTER-AIDED TOOL FOR THE TEACHING OF RELATIONAL ALGEBRA IN DATA BASE COURSES

Johnny Villalobos Murillo¹
Steven Brenes Chavarría²
Sonia Mora Rivera³

RESUMEN

En este artículo se describe el diseño, la implementación y el uso de una herramienta asistida por computadora para la enseñanza del álgebra relacional en cursos de bases de datos, la herramienta ha sido llamada Relational Algebra Traductor (RAT).

En el curso EIF 211 Diseño e implementación de bases de datos de la carrera de Ingeniería en Sistemas de Información de la Universidad Nacional de Costa Rica se tiene la problemática de introducir el tema del álgebra relacional y siendo los conocimientos matemáticos del estudiante no muy profundos en el área, se daba una gran dificultad en el aprendizaje del tema, el cual es muy importante para entender bien el trabajo que realizan las bases de datos, de esta forma el RAT ha sido conceptualizado como una herramienta que viene a apoyar el proceso de enseñanza-aprendizaje.

Se presentan los principios de diseño y arquitectura que requiere su implementación tales como: la tabla de símbolos del lenguaje, las reglas gramaticales y los algoritmos básicos que utiliza el Relational Algebra Traductor, para realizar las traducciones de álgebra relacional al lenguaje SQL⁴. La herramienta se ha puesto en práctica durante un ciclo lectivo y ha demostrado tener efectividad en el proceso de enseñanza aprendizaje; esto ha motivado a los investigadores a publicarlo en el sitio

www.slinfo.una.ac.cr con el propósito de que esta herramienta pueda ser utilizada en diversos cursos en otras universidades.

Palabras clave: compilador, enseñanza-aprendizaje, software educativo, SQL, álgebra relacional, traductor.

ABSTRACT.

This article describes the design and implementation of computer-aided tool called Relational Algebra Translator (RAT) in data base courses, for the teaching of relational algebra. There was a problem when introducing the relational algebra topic in the course EIF 211 Design and Implementation of Databases, which belongs to the career of Engineering in Information Systems of the National University of Costa Rica, because students attending this course were lacking profound mathematical knowledge, which led to a learning problem, being this an important subject to understand what the data bases search and request do RAT comes along to enhance the teaching-learning process.

It introduces the architectural and design principles required for its implementation, such as: the language symbol table, the gramatical rules and the basic algorithms that RAT uses to translate from relational algebra to SQL language. This tool has been used for one periods and has demonstrated to be effective in the learning-teaching process. This

1 Universidad Nacional de Costa Rica, Escuela de Informática, jvillalo@una.ac.cr

2 Universidad Nacional de Costa Rica, Escuela de Informática, steven.brenes.chavarría@est.una.ac.cr,

3 Universidad Nacional de Costa Rica, Escuela de Informática, smora@una.ac.cr

4 SQL es un lenguaje de consulta estructurado que permite realizar consultas en bases de datos relacionales.

urged investigators to publish it in the web site: www.slinfo.una.ac.cr in order for this tool to be used in other university courses.

Keywords: compiler, teaching-learning, educational software, SQL relational algebra, translator.

INTRODUCCIÓN

Las carreras en ingeniería de sistemas de información, ciencias de la computación y carreras afines, en general, tienen una problemática al contemplar cursos con temas que tienen un alto nivel de abstracción para los estudiantes, como por ejemplo el álgebra relacional, y se podrían citar muchos ejemplos más de esto, pero no es tema del artículo, sino más bien enfocarse en la dificultad que presentan los cursos de bases de datos cuando se enseña el tema del álgebra relacional. En los contenidos del curso EIF 211 Diseño e implementación de bases de datos, que se imparte en la carrera Ingeniería en Sistemas de Información de la Universidad Nacional de Costa Rica se contempla el tema del álgebra relacional, partiendo de la importancia de este tema en el área de bases de datos y lenguajes de consulta (Cood, 1990). El alto nivel de abstracción que exige del estudiantado dificulta mucho su enseñanza-aprendizaje, pues prácticamente es un lenguaje de consulta muy teórico; mientras que otros lenguajes de consulta como el SQL, sí permiten realizar pruebas y obtener resultados de forma natural en las bases de datos y por lo tanto son menos abstractos.

El álgebra relacional sin embargo, no dispone de un mecanismo que permita interactuar con las bases de datos de la actualidad directamente; por esta razón se gesta la idea de crear una herramienta que facilitara a los estudiantes el aprendizaje en este tema, se parte del supuesto que los estudiantes logran un aprendizaje significativo en el tema si prueban las soluciones a los problemas que se les presentan contra datos reales, esto disminuye el nivel de abstracción y reafirma los prin-

cipios teóricos del modelo relacional que es uno de los ejes principales del curso.

1. La herramienta

1.1 Conceptualización

Estando clara la necesidad de contar con una herramienta que facilite al estudiante el modelado y puesta en práctica del álgebra relacional para comprender los procesos que se llevan a cabo en la generación de consultas en bases de datos, se procede a diseñar la herramienta RAT.

Tomando en cuenta la tecnología actual la herramienta debe ser capaz de traducir el álgebra relacional a un lenguaje que sea entendido no sólo por los estudiantes sino por las bases de datos utilizadas en el curso (Oracle, MySQL, Postgress y SQL server), dadas estas condiciones se establece por lo tanto el lenguaje SQL como el lenguaje destino.

De esta manera, se enuncian los siguientes requerimientos de la herramienta, como básicos por ser considerados en el diseño inicial:

- Debe contar con controles para asegurar que la escritura de las sentencias sea la correcta, es decir debe garantizar la sintaxis correcta de las sentencias.
- Debe poder traducir sentencias del álgebra relacional a SQL.
- Debe poder conectarse a diversas bases de datos, para probar las sentencias.
- Debe poder graficar las consultas para facilitar la visualización de la solución y construcción de las consultas.

Estos requerimientos también establecen que el formalismo y la simbología de los operadores del álgebra relacional se deben garantizar, por lo que la herramienta necesariamente tendrá que contar con la implementación de los operadores oficiales: *sigma*, *pi*, *ro*, *uniones*, *intersecciones* y *diferencia de conjuntos*

(Silberschatz, 2006) que representan los operadores de relaciones y teoría de conjuntos del álgebra relacional.

Además, al considerar la representación gráfica de las operaciones en el álgebra relacional como un requerimiento importante para la comprensión de las consultas, el sistema debe ser capaz de representar mediante un árbol de parser⁵ (Aho, 2008) la construcción de las sentencias durante la traducción.

1.2 Diseño y construcción de la herramienta

De esta forma la herramienta propuesta es una aplicación computacional que debe combinar las técnicas de compiladores, matemáticas discretas, estructura de datos y técnicas de programación de algoritmos como recursividad sobre árboles binarios para lograr el objetivo final. Las técnicas de compiladores, estudian el diseño de aquellos programas que

permiten traducir un lenguaje origen a un lenguaje destino (Hopcroft, 2001), normalmente cuando se habla de compiladores el lenguaje destino suele ser el lenguaje máquina llamado también código binario, sin embargo la definición no limita la existencia de un lenguaje diferente a este último, para ser precisos se van utilizar técnicas de traductores, las cuales a su vez se basan en técnicas de compiladores.

Se plantea entonces un *software* con características de compilador que permita traducir sentencias de álgebra relacional a un lenguaje que las bases de datos entiendan (SQL), un ejemplo de esta traducción puede ser: $\Pi_{nombre, cedula, edad} (\sigma_{sexo=masculino} (personas))$ como sentencia de álgebra relacional, su correspondiente traducción a SQL es:

```
SELECT nombre, cedula, edad FROM personas WHERE sexo = 'Masculino'
```

Cuadro 1. Ejemplo de traducción del álgebra relacional al SQL.

Componente	Álgebra relacional	SQL
Proyección	$\Pi_a(t1)$	SELECT a FROM t1;
Selección	$\sigma_{a=12}(t1)$	SELECT * FROM t1 WHERE a = 12;
Producto cartesiano	$t1 \times t2$	SELECT * FROM t1 JOIN t2
Producto natural	$t1 \bowtie t2$	SELECT * FROM t1 NATURAL JOIN t2
Sobrenombre	$\rho_{alias} t1$	(SELECT * FROM t1) alias
Intersección	$t1 \cap t2$	(SELECT * FROM t1) INTERSECT (SELECT * FROM t2)
Unión	$t1 \cup t2$	(SELECT * FROM t1) UNION (SELECT * FROM t2)
Diferencia	$t1 - t2$	(SELECT * FROM t1) DIFFERENCE (SELECT * FROM t2)

Fuente: Desarrollo propio de los autores.

5 Un árbol de parser o árbol de derivación es una estructura de datos llamada árbol binario, en bases de datos es una estructura compuesta por operadores y variables que modelan una consulta en álgebra relacional. Se usará para efectos del artículo, por ser más conocido el término de árbol de parser.

Cabe rescatar que las bases de datos que usan el SQL son sistemas basados en la teoría de relaciones enarias, funciones y teoría de conjuntos (Silberschatz, 2006), motivo por el cual la traducción entre el álgebra relacional al SQL evita la pérdida de funcionalidad, ejemplo de ello es que existe el equivalente en SQL de los operadores del álgebra relacional. En el siguiente cuadro se muestran algunos ejemplos de las equivalencias antes mencionadas:

De esta forma, se establecen los siguientes componentes como factores principales en el diseño e implementación de la herramienta:

1. analizador lexicográfico
2. analizador sintáctico
3. analizador semántico
4. traductor de álgebra a SQL
5. graficador del árbol de parser

Este diseño está orientado a disminuir la posibilidad de errores de escritura por parte de los estudiantes, esto porque los tres primeros analizadores garantizan que la sentencia de álgebra relacional ingresada al sistema sea válida y por lo tanto la consulta pueda ser realizada.

Para comprender mejor el desarrollo de la aplicación es importante retomar algunas definiciones importantes que permitan una mejor comprensión del tema:

Existen dos tipos básicos y reconocidos de lenguajes: los lenguajes naturales y los lenguajes formales (Chomsky, 1975); los primeros se fueron construyendo con el paso del tiempo mediante la conversación hablada y escrita, estos son por ejemplo el español o el inglés. Además, los lenguajes formales se basan en normas matemáticas tales como la lógica y teoría de conjuntos para su modelaje y entendimiento.

Definición 1. Se define un lenguaje como un conjunto de palabras. Cada lenguaje está

formado por secuencias de símbolos (palabras) tomados de alguna colección finita llamada alfabeto.

Definición 2. Se define un alfabeto como un conjunto no vacío y finito de símbolos, entonces escribimos un alfabeto y Una secuencia finita de símbolos de un alfabeto, se llama palabra, expresión o cadena.

Definición 3. Una palabra es un conjunto finito de símbolos , con la concatenación de palabras formaremos un conjunto denotado por , luego el conjunto de palabras que tengan significado se llama diccionario.

Desde el punto de vista técnico, un compilador además de ser un traductor es un programa que verifica que la sentencia de entrada sea válida. Una sentencia válida es aquella expresión lexicográfica, semántica y sintácticamente correcta. De esta manera, la herramienta propuesta debe cumplir con estas validaciones para asegurar sentencias correctas que puedan ser traducidas a sentencias en un lenguaje SQL.

Definición 4. Una expresión es lexicográficamente correcta si, y solo , con *exp* una palabra.

La utilidad del analizador lexicográfico es la detección temprana de errores de tipo alfabético (Allen, 2002); por ejemplo la expresión no es una expresión léxico-gráficamente válida, pues , en particular .

Desde estas premisas se define en el RAT el alfabeto del álgebra relacional como: $= \{0, 1, 2, \dots, 8, 9, _ , a, b, c, \dots, A, B, C, \dots, Z, <, >, \leq, \geq, \neq, =, \vee, \circ, \square, \cup, -, \square, \times, \sigma, \Pi, \text{"}, \text{"}\}$, donde $_$ representa un carácter nulo.

Definición 6. Se establece una gramática usando la definición de Noam Chomsky (Chomsky, 1975), compuesta por la siguiente cuaterna:

1. Un conjunto finito de símbolos, que representa la totalidad del alfabeto de un lenguaje. Los componentes reciben el nombre de *terminales* o *símbolos terminales*.
2. Un conjunto de variables, a estas variables se les denomina *no terminales* o *categorías sintácticas*, estos no terminales se forman por otros símbolos terminales o no terminales.
3. Existe un símbolo inicial, esta variable representa el inicio de la gramática, además debe ser un símbolo no terminal.
4. Un conjunto finito de *producciones* o *reglas*, que representa una definición recursiva de lenguaje. Cada regla está conformada de:
 - a. Una variable llamada cabeza, es un identificador de la regla en particular.
 - b. El símbolo de producción ().
 - c. Una cadena de cero o más símbolos terminales o no terminales, a este bloque se le llama cuerpo de la producción, en la cual cada elemento se sustituye recursivamente para formar una derivación.

bién recibe el nombre de *gramática* (Chomsky, 1975) de tal forma que se puede designar a $G = (V,T,P,S)$ donde V es el conjunto de los símbolos no terminales, T el de los símbolos terminales, P son las reglas de producción y la S es el símbolo inicial de la gramática.

Se define entonces la gramática para el RAT como la tupla con $V = \{\text{literal, numeral, símbolos numéricos, símbolos lógicos, símbolos conjuntos, variable, número, parámetro, proyección, condicional, renombramiento, expresión, expresión lógica, expresión numérica}\}$. El conjunto T o alfabeto del lenguaje se define como $T = \{0, 1, 2, \dots, 8, 9, _ , a, b, c, \dots, A, B, C, \dots, Z, <, >, \leq, \geq, \neq, =, \vee, \circ, \square, \cup, -, \sqcap, \times, \sigma, \Pi, \text{“,”}\}$ donde $_$ representa un carácter nulo. Se cuenta también el símbolo inicial $S = \text{expresión}$, definido en las reglas de producción de P .

Para el conjunto P se definen las siguiente reglas de producción, para efectos de simplificar la notación se usará el formato Backus-Naur form o BNF (Hoperoft, 2001), donde los símbolos de producción () y sus respectivos cuerpos se agrupan con el símbolo ::= y separamos los cuerpos de las producciones por el símbolo (|) para identificar que son de la misma regla.

Estas cuatro componentes definen una tupla llamada *gramática libre de contexto*, tam-

A continuación se detalla el conjunto de producciones:

```

<literal> ::= _|a|b|c|...|A|B|C|...|Z

<numeral> ::= ,|-|0|1|2|...|8|9

<simbolosNumericos> ::= +|-|*|/

<simbolosLogicos> ::= =\vee|\circ|<|>|\leq|\geq|\neq|=

<simbolosConjuntos> ::= \cap|\cup|-|\times

<variable> ::= <literal><variable>|<variable>
    
```

```

<número> ::= <literal> <variable> | <variable>

<parámetros> ::= <variable>, <parámetros> | <variable>

<proyección> ::=  $\Pi$ (<parámetros>) (<expresión>)

<renombramiento> ::= (<literal>) (<expresión>)

<condicional> ::= [<proyección>]  $\sigma$ (<expresiónLogica>) (<expresión>)

<expresiónLogica> ::= <expresiónNumerica> <simboloLogicos>
| <expresiónNumerica>

<expresiónNumerica> ::= <variable> <simbolosNumericos> <número> | <número>
<simboloNumericos> <variable>

<expresión> ::=
[ (<expresión> ) ] <simbolosConjuntos> [ (<expresión> ) ] |
<proyección> | <condicional> | <variable>

```

Definición 8. Una expresión es semánticamente correcta si, y solo si la expresión es producida por alguna regla de producción.

La importancia del analizador semántico es la detección de errores estructurales, de esta forma se sabe si la expresión que introduce el usuario está correctamente formada,

suponga la expresión: , esta expresión es lexicográficamente válida, pues todos los símbolos de la expresión son elementos del alfabeto, pero es semánticamente incorrecta, el error se encuentra en el "" porque la regla dentro del PI es un parámetro definido por las siguientes reglas:

```

<parámetros> ::= <variable>, <parámetros> | <variable>

<variable> ::= <literal> <variable> | <simbolosNumericos>

<simbolosNumericos> ::= +|-|*|/

```

Se esperaba un número o una variable después del primer signo de más, pero lo que se leyó fue el signo “+”, y es aquí donde se presenta un error de semántica.

Definición 9. Se define una matriz gramatical o matriz de adyacencia, como una matriz cuadrada que permite representar una relación binaria⁶.

Para el RAT se desarrolla una **matriz gramatical del álgebra relacional**, para el verificador sintáctico, para crear y evaluar todos los posibles escenarios. Esta matriz booleana está formada por todos los símbolos no terminales. La matriz completa del álgebra relacional al ser tan grande no se adjuntará, sin embargo se puede ejemplificar con los símbolos Pi, Sigma, paréntesis, literal y número (constante) mostrados en el cuadro 2.

La función de esta matriz dentro del programa es verificar la gramática del álgebra relacional, funciona con los estados de la siguiente regla, por ejemplo en la fila de “constante” todas las columnas tienen ceros

menos la columna llamada “paréntesis cerrado” con lo que se interpreta como después de un número puede ir un paréntesis cerrado. Observe que también puede ir un operador matemático pero por efectos del ejemplo queda excluido este caso.

Definición 10. Un árbol consta de un conjunto finito de elementos, llamados nodos y un conjunto también finito de líneas dirigidas o vértices que conectan los nodos (Joyanes, 2007). También, se puede definir como

1. Una estructura de datos vacía.
2. Una conjunto de nodos tales que
 - a. Hay un nodo llamado raíz
 - b. Todos los demás son sub-árboles

Observe que la definición de árbol es recursiva, y que la mayoría de los algoritmos para recorrer árboles son recursivos, pero lo importante de los árboles binarios es su capacidad para modelar las expresiones de álgebra relacional de forma prácticamente natural. En la figura 1. Árboles binarios y el álgebra

Cuadro 2. Matriz gramatical (adyacencia) parcial.

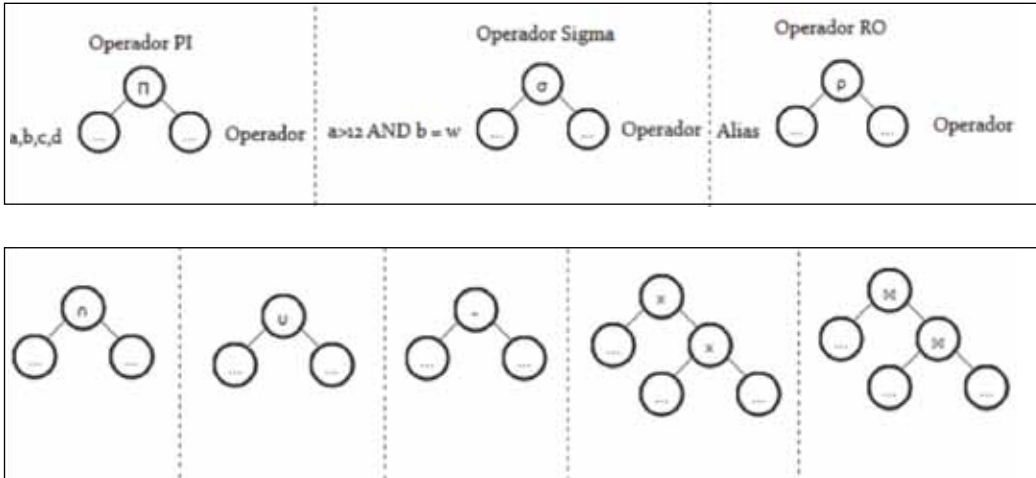
	SIGMA	LITERAL	CONSTANTE	PARENTESIS ABIERTO	PARENTESIS CERRADO
PI	0	0	0	1	0
SIGMA	0	0	0	1	0
LITERAL	0	0	0	0	1
CONSTANTE	0	0	0	0	1
PARENTESIS ABIERTO	1	1	1	1	0
PARENTESIS CERRADO	0	0	0	1	1

Fuente: desarrollo propio de los autores

⁶ Una relación binaria es una asociación que se realiza entre dos conjuntos.

relacional se aprecia cómo se modelaron en el RAT las estructuras de árboles.

Figura 1. Árboles binarios y el álgebra relacional



Fuente: Desarrollo de los autores.

Se presenta a continuación el algoritmo generador de árboles de parser, la función tiene

como entrada una lista de Tokens⁷ pre procesada de una expresión de álgebra relacional:

```

OPERADOR_APERTURA = {PI, SIGMA, RO, CARTESIANO, NATURAL, INTERSECCIÓN, UNION, DIFERENCIA}
FUNCION crearArbol [TOKENS tokens, NODO actual]
INICIO
SI tokens.vacio() == falso ENTONCES
    TOKEN tmp = tokens.siguiete();
    SI tmp.dato == OPERADOR_APERTURA ENTONCES
        actual.dato = tmp;
        tones.siguiete();
        crearArbol(tokens, actual.izquierda);
        tones.siguiete();
        crearArbol(tokens, actual.derecha);
    EN_CASO_CONTRARIO
        MIENTRAS tmp <> CERRAR_PARENTESIS ENTONCES
            actual.agregar(tmp);
        tmp = tokens.siguiete();
        FIN MIENTRAS;
    FIN SI;
FIN FUNCION;

```

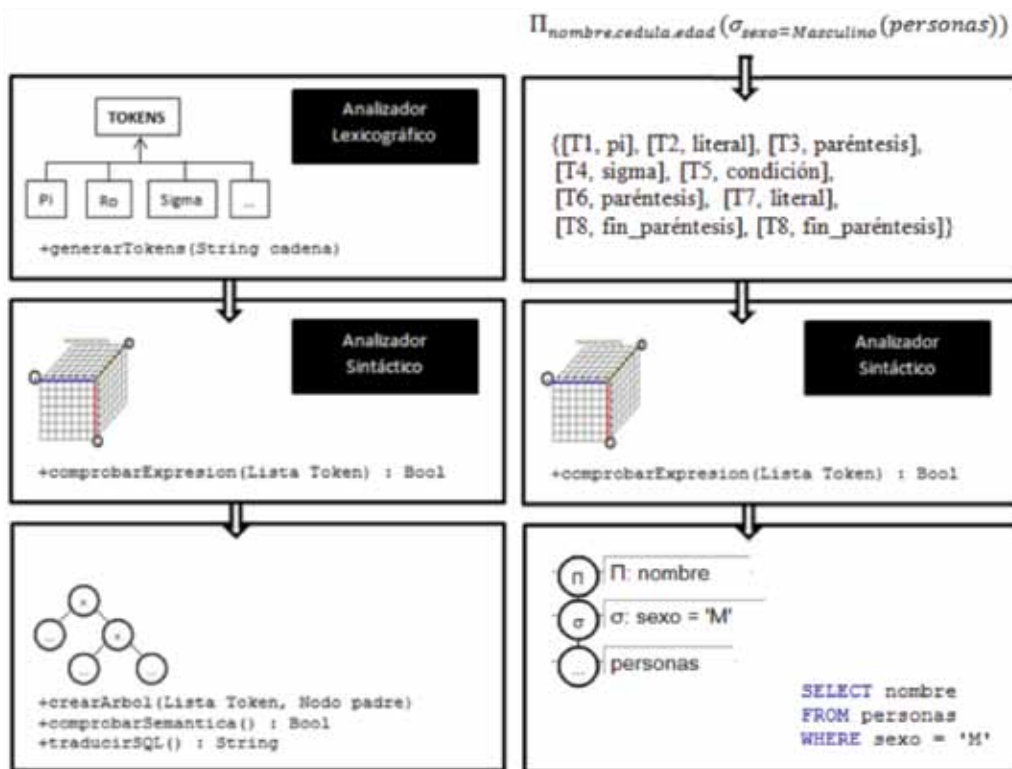
⁷ Un *token* es una unidad léxica que tiene significado en un ámbito.

Este algoritmo tiene el propósito de generar un árbol de parser a partir de una lista de *tokens*, el objetivo es distribuir recursivamente los *tokens* según su tipo, los árboles de parser se bifurcan en presencia de un operador de conjuntos (unión, intersección o diferencia), un operador de producto cartesiano o un operador de producto natural, de esta forma se crean dos nuevos hijos en el algoritmo. Para los demás casos se acumulan los tokens en los nodos.

Para traducir una sentencia de álgebra relacional a SQL, el programa está dividido en tres componentes principales. Inicialmente las sentencias ingresan como cadenas de ca-

racteres, en ese momento, el analizador lexicográfico revisa que no ingresen símbolos del alfabeto que no pertenecen al mismo, si la verificación léxica es correcta el analizador retornará una lista de tokens que luego son alimentados al analizador sintáctico el cual procesa las reglas de la matriz gramatical para detectar errores tempranos, si el proceso fue correcto entonces los *tokens* prosiguen a la etapa final donde el RAT genera el árbol de *parser* con el algoritmo ejemplificado anteriormente. Posteriormente, para realizar la traducción a SQL es tan simple como recorrer el árbol en inorden⁸ e ir concatenando los resultados de los nodos.

Figura 2. Proceso de la traducción. Consulta



Fuente: desarrollo de los autores

8 El recorrido en inorden se basa en extraer el dato del nodo actual, luego irse recursivamente al nodo izquierdo y nodo derecho.

De esta forma, cuando se ingresa una consulta en el RAT, este se encarga de traducir a lenguaje SQL con su correspondiente árbol de parser, como se muestra en la Figura 2. Traducción de la sentencia , donde en el primer cuadro esta la lista de *tokens* generado por el analizador lexicográfico, estos *tokens* son validados contra las reglas de la gramática para finalmente aplicar el algoritmo generador del árbol, con lo que se obtiene el árbol final con su respectiva sentencia SQL.

En la figura 3. Software Relational Algebra Translator (RAT) v4.1.0.0 se muestra propiamente la herramienta traduciendo la sentencia:

- A $\leftarrow \pi$ {nombre, edad, cédula} (personas)
- B $\leftarrow \pi$ {nota, curso, estudiante} (notas)
- C $\leftarrow \sigma$ {cédula = estudiante} (A x B)
- D $\leftarrow \pi$ {nombre, edad, cédula, nota, curso} (C)

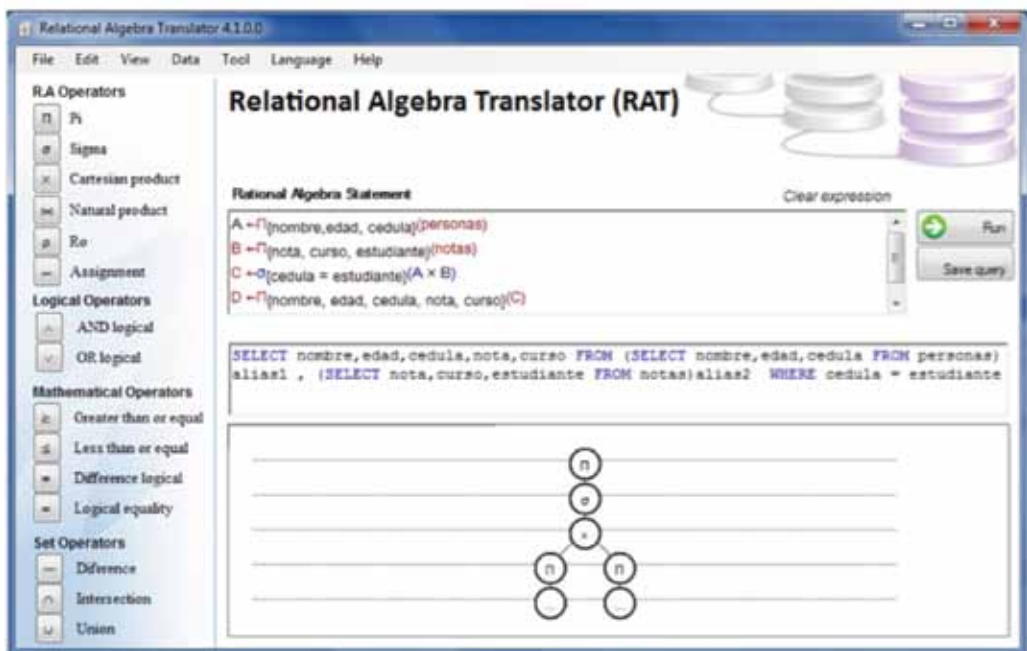
Note que los nodos de las hojas del árbol generado están siempre equilibrados y representan el crecimiento de la consulta, desde

las tablas originales hasta llegar al último nodo de la consulta.

Este proceso les permite a los estudiantes comprender mejor el proceso que realizan las bases de datos cuando reciben una consulta a ser procesada, de ahí la importancia de generar el árbol, pues la herramienta se convierte en un recurso didáctico más en el curso.

Las consultas en álgebra relacional fácilmente se pueden complicar, por lo que se implementó un operador denominado *operador de asignación*, el cual tiene la función de asignar a variables locales los valores de las relaciones que se encuentren a su derecha, por ejemplo: asigna el resultado del sigma a la variable A. Se presenta a continuación un cuadro comparativo que ejemplifica el operador de asignación. Es claramente más simple escribir sentencias usando el operador de asignación, lo que disminuye la complejidad del tema; además el RAT identifica cuando se declaran operaciones recursivas inválidas como por ejemplo: .

Figura 3. Software Relational Algebra Translator (RAT) v4.1.0.0



Cuadro 3. Ejemplo del operador de asignación

Sin operador de asignación	Con operador de asignación
$\Pi_{\{\text{nombre, edad, cedula, nota, curso}\}}(\sigma_{\{\text{cedula} = \text{estudiante}\}}(\Pi_{\{\text{nombre, edad, cedula}\}}(\text{personas}) \times \Pi_{\{\text{nota, curso, estudiante}\}}(\text{notas})))$	$A \leftarrow \Pi_{\{\text{nombre, edad, cedula}\}}(\text{personas})$ $B \leftarrow \Pi_{\{\text{nota, curso, estudiante}\}}(\text{notas})$ $C \leftarrow \sigma_{\{\text{cedula} = \text{estudiante}\}}(A \times B)$ $D \leftarrow \Pi_{\{\text{nombre, edad, cedula, nota, curso}\}}(C)$

Fuente: desarrollo propio de los autores.

Y pensando en las necesidades de los estudiantes y la facilidad de uso de la herramienta, se implementaron algunos otros módulos como son:

1. Una biblioteca donde el estudiante puede guardar las consultas que ha ido construyendo y probando. En este proceso, las consultas pueden ser exportadas y enviadas como proceso de verificación del trabajo, documentación posterior de proyectos, y eventualmente como comprobación de tareas o quices.
2. Un analizador de rendimiento que compara la cantidad de productos cartesianos entre dos consultas, para efecto de optimizarlas.
3. La herramienta puede ser visualizada en diferentes idiomas: español, inglés, italiano y alemán, además se da la facilidad de que cada persona la adapte a su propio idioma.
4. La herramienta permite la conexión a cualquier base de datos relacional, mediante ODBC⁹

Además de estas facilidades, se puede decir que el uso del RAT va mucho más allá de las expectativas de los diseñadores, pues se puede utilizar como herramienta para enseñar matemáticas, vea el siguiente ejemplo: *Demuestre que la siguiente afirmación es válida para conjuntos finitos predefinidos; utilice en su demostración el siguiente teorema:* . De esta forma el estudiante de matemática

podría coger dos conjuntos A y B de una base de datos y aplicar las siguientes consultas, que debe dar un conjunto vacío.

$$\begin{aligned}
 X &\leftarrow [[A]-[[A]-[B]]]-[[A]\cap[B]] & A-B &= \phi \\
 Y &\leftarrow [[B]-[[B]-[A]]]-[[B]\cap[A]] & B-A &= \phi \\
 W &\leftarrow [X]\cup[Y] & & \rightarrow A=B \blacksquare
 \end{aligned}$$

También, podría ser una herramienta muy útil para formar nuevos conjuntos a partir de los ya existentes, y se aplica productos cartesianos o proyecciones; para que de esta forma sea más claro el proceso, y en esta línea el operador *sigma* facilita la escritura de conjuntos mediante el axioma de comprensión¹⁰. Por ejemplo: *Genere un conjunto formado solo por número pares, a partir de un subconjunto de los naturales*. El estudiante podría llegar a generar el siguiente conjunto , ya sea mediante la consulta de proyección: o la siguiente consulta equivalente , esta última es la forma más natural de exponer los conjuntos por comprensión, la primera forma en que se presentó sería un ejemplo típico de función (de naturales a pares). Estos aspectos de apoyo en áreas específicas de la enseñanza de la matemática aún no se han experimentado con estudiantes, pero serán trabajos para futuras investigaciones.

1.3 Etapa de pruebas

Para probar la utilidad de la herramienta RAT se diseñaron una serie de prácticas de

⁹ ODBC es un estándar abierto para comunicación entre bases de datos.

¹⁰ El axioma de comprensión afirma que se puede declarar un conjunto primero exponiendo una propiedad que todos sus elementos la cumplan (Halmos, 2000).

laboratorio constituidas por problemas de diseño de consultas en álgebra relacional. Las consultas se clasificaron en consultas simples, complejas y optimizadas. Se tomó el siguiente escenario formado por las tablas: artículos, vendedores, detalles, facturas, clientes, zonas, inventarios y bodegas. Para plantear consultas simples como son:

1. Obtener todos los vendedores en el sistema.
2. Obtener todos los clientes en el sistema.

Así también se utilizaron consultas complejas tales como

1. Mostrar todos los clientes que han comprado productos ubicados en bodegas de la zona Sur.
2. Mostrar todas las facturas de aquellos productos que vengan un artículo "TV Plasma" de inventarios ubicados en la zona Norte.

Para las consultas optimizadas, se solicitó al estudiante que aplicara la heurística sigma-pi¹¹ sobre las consultas complejas con su respectivo árbol canónico de parser.

Para esta etapa se consideraron cuatro grupos de estudiantes del curso EIF 211 Diseño e implementación de base de datos, cada grupo estaba formado por 25 estudiantes, la herramienta se le proporcionó a dos de los grupos y se les capacitó en su uso.

El cálculo de consultas optimizadas, el cual representaba uno de los mayores retos para este tema en particular, según los estudiantes dejó de ser un proceso complejo debido al

11 La heurística sigma-pi (Silberschatz, 2006) disminuye la cantidad de productos cartesianos, se basa en distribuir los sigmas por todo el árbol lo más cerca de las tablas y los productos cartesianos, la heurística de pi establece el uso explícito de las columnas que se ocupan de la consulta en vez de tomar todas columnas a la vez (situación que se da por defecto en las bases de datos).

uso del RAT, y lo más importante es que los estudiantes pueden comprobar sus optimizaciones, que generalmente son estructuras complejas de varios niveles, de forma ágil y rápida.

Según las opiniones de los estudiantes la prueba fue superada con éxito, la diferencia fundamental radica en la facilidad que la herramienta les proporciona para disminuir los requerimientos de abstracción y la facilidad de realizar las pruebas de las posibles soluciones que calculan.

Además, los estudiantes expresaron que la herramienta es fácil de usar y les pareció muy útil la ayuda que ofrece cuando se les presenta un error de escritura y reconocieron que los árboles de parser, generados para la consulta, son de gran valor agregado debido a que el proceso de dibujar el árbol era un proceso meramente manual y tedioso.

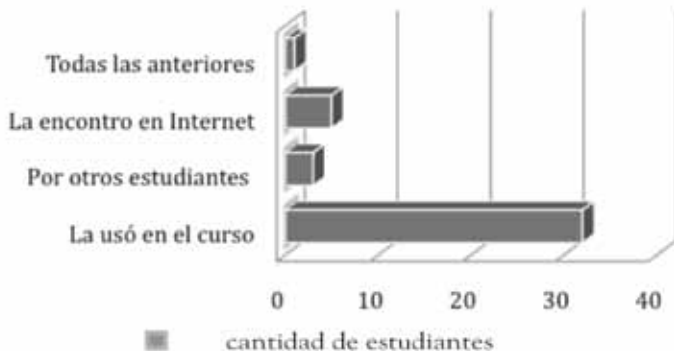
2. Aplicación y uso de la herramienta

2.1 A nivel nacional

La herramienta ha estado en uso y en este ciclo se decide hacer una encuesta anónima que permita validarla, se entrega a estudiantes de bases de datos del curso EIF 402 Administración de bases de datos, que es el siguiente curso de bases de datos que llevan los estudiantes del plan de estudios, se logran 36 encuestas, se valida inicialmente el conocimiento que los estudiantes tienen de la herramienta, pues en este momento no todos los profesores del primer curso la usan, sin embargo de los 36 estudiantes encuestados el 100% conoce la herramienta, y de distintas maneras se enteran de su existencia, como se ve en el siguiente gráfico, Figura 4. Conocimiento del RAT, donde la mayoría de los estudiantes usan el RAT en el curso, otros la usan por que sus compañeros se lo dan a conocer y otros porque la encuentran en Internet.

Figura 4. Conocimiento del RAT

¿Cómo se enteró de la herramienta RAT?



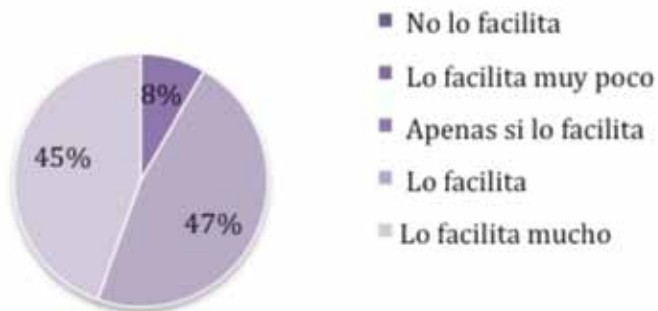
Fuente: encuesta realizada a estudiantes, 2011.

Para conocer si la herramienta les había facilitado a los estudiantes la comprensión del álgebra relacional aplicada a las bases de datos, en la Figura 5. El Rat facilita el proceso de consulta, se muestran los resultados, en este aspecto los estudiantes en su mayoría expre-

saron que con la herramienta y sobre todo con las facilidades que presenta y la usabilidad de la misma, se había mejorado el proceso de entender la generación de consultas:

Figura 5. El RAT facilita el proceso de consulta.

¿Facilita el RAT el proceso de consulta?



Fuente: Encuesta realizada a estudiantes, 2011.

Figura 6. Aporte del RAT al aprendizaje del estudiante.

Fuente: encuesta realizada a estudiantes, 2011.

En general los estudiantes opinaron que efectivamente el RAT facilita los procesos de construcción de consultas, un 92 % expresó que lo facilita o lo facilita mucho, y según Figura 6. Aportes del RAT al aprendizaje, los estudiantes consideran que la herramienta ha sido un valioso aporte a su proceso de aprendizaje.

Por último se les pidió a los estudiantes que expresaran su opinión acerca de las facilidades que presenta el RAT, y algunas de las opiniones son:

- “se obtiene una optimización eficiente de las consultas sobre una base de datos”
- “ayuda a comprender mejor la elaboración de las consultas”
- “permite conectarnos a la base de datos y ver si nuestra consulta es correcta”
- “permite diseñar consultas por separado para luego unir las en una sola consulta general lo cual promueve la optimización de las mismas”
- “muestra mucho más claramente la generación de sentencias de SQL y álgebra relacional”

- “integra el álgebra relacional, la consulta SQL y el árbol, lo cual lo hace más comprensible”
- “es muy básico y útil, sin botones ni cosas que compliquen su usabilidad”

De esta forma la mayoría de los estudiantes reconoce que el RAT ha hecho aportes en su aprendizaje y ha mejorado el nivel de comprensión, mediante una interfaz sencilla, agradable que de manera gráfica muestra la construcción de una consulta usando álgebra relacional en bases de datos.

2.3 A nivel internacional

En el año 2010 con el fin de ampliar la difusión de la investigación y que otros estudiantes y docentes pudieran conocer y evaluar el RAT, se decidió ponerlo a disposición desde el sitio de investigaciones en bases de datos de la Escuela de Informática llamado Laboratorio Virtual de Bases de Datos, en el siguiente enlace: www.slinfo.una.ac.cr).

Desde entonces, el sitio ha recibido 2763 visitas de diferentes países del mundo, y de estas visitas 1566 se han convertido

en descargas de la herramienta, como se muestra en la Figura 7. Descargas del RAT, en el cual se observa un reporte de utilización y descargas obtenido de Google Analytics, el 30 de octubre del 2011.

Figura 7. Descargas del RAT

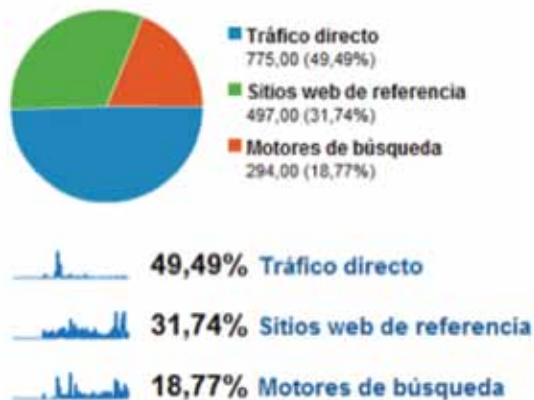
Perfiles del sitio web		
Nombre	Visitas	Promedio de tiempo en el sitio
http://www.sleifo.una.ac.cr/ratidesIA-21078955-8		
★ Sibo de descargas del RAT	1.566	00:01:17
http://www.sleifo.una.ac.cr/manual955-6		
★ Metodología para la Administración de Bases de Datos Relacionales	934	00:01:31
http://www.sleifo.una.ac.cr/ratvafL		
★ Relational Algebra Translator	2.763	00:01:50

Fuente: Google Analytics

El RAT se ha popularizado a nivel mundial y diferentes sitios de descarga lo recomiendan como herramienta, los usuarios ingresan a obtener una copia desde estos sitios o mediante buscadores directos, en la Figura 8. Formas de acceder al RAT se muestran las diferentes formas de acceso al sitio de descarga del RAT, desde Google Analytics.

Además, se tiene información de usuarios de diversos países de todos los continentes que han visitado y descargado el RAT, en la Figura 9. Descargas por país se muestra la lista de los 10 países que han presentado más descargas, obviamente Costa Rica se encuentra entre los primeros, al ser una herramienta de estudio en los cursos, luego siguen España, Estados Unidos, Perú, Brasil, Colombia, Argentina, México, etc. En este momento se están haciendo los contactos de tal manera que se empiecen a tener comentarios acerca del uso de la herramienta, desde estas universidades y sitios.

Figura 8. Formas de acceder al RAT



Fuente: Google Analytics 10/30/2011

Figura 9. Descargas por país

	Visitas ↓	Páginas/visita	Promedio de tiempo en el sitio	Porcentaje de visitas nuevas
1. Costa Rica	732	1,39	00:01:11	54,76%
2. Spain	613	1,25	00:01:53	81,42%
3. United States	76	1,05	00:00:14	42,11%
4. Peru	67	1,21	00:01:10	77,81%
5. Brazil	67	1,24	00:02:15	85,07%
6. Colombia	65	1,23	00:01:24	66,15%
7. Argentina	62	1,23	00:02:11	91,94%
8. Mexico	52	1,35	00:01:48	84,62%
9. Italy	32	1,09	00:00:01	96,50%
10. El Salvador	25	1,16	00:02:14	84,00%

Fuente: Google Analytics 10/30/2011

Muchas de esas descargas vienen de universidades, el Cuadro 3. Universidades que hacen uso del RAT. Muestra algunas de las universidades donde el RAT está siendo usada, de esta forma se puede decir que la Universidad Nacional está haciendo un gran aporte a nivel internacional en el área de bases de datos.

Cuadro 3. Universidades que hacen uso del RAT

Nombre de Universidad	País
Nicolai Copernicus	Poland
Nueva Granada	Colombia
Instituto Politécnico Nacional México	
Columbia	Paraguay
Stanford	Estados Unidos
University for Computer Science and Technology	Macedonia
Leibniz University Hanover	Hanover
Universidad Nacional de Colombia	Colombia
Instituto Superior Técnico	Portugal
Universidad Nacional de Asunción	Paraguay
Universidad Nacional Autónoma de Honduras	Honduras
Universidad de Buenos Aires	Argentina
Unicatólica	Colombia
Cordoba	Colombia
University of Wisconsin-Parkside	Estados Unidos
Unipanamericana	Colombia
Instituto de Matemática y estadística	Brasil
San pedro	Perú
Universidad complutense	España
Unicatólica Lumen Gemtium	Colombia

Fuente: Formularios de registro de descarga.

Así, se puede afirmar que este trabajo ha venido a mejorar los procesos de enseñanza aprendizaje del curso EIF 211 Diseño e implementación de bases de datos, con respecto a la implementación del álgebra relacional a procesos de consulta y generación de senten-

cias en SQL, y que con sobrada razón empieza a darse a conocer nacional e internacionalmente, de ahí la importancia de comunicar a la sociedad académica y a la sociedad en general de este tipo de trabajos.

3. Conclusiones

El desarrollo del RAT inicia como la idea de un profesor, como parte de sus investigaciones, poco a poco va tomando forma gracias a estudiantes de la carrera, hasta convertirse en una realidad: una herramienta que facilita la introducción y comprensión del álgebra relacional en los cursos de bases de datos de la Escuela de Informática de la Universidad Nacional, y mejora la comprensión y entendimiento de la consultas hechas en SQL, según los mismos estudiantes lo confirman.

El desarrollo del RAT fue un desafío, pues incorporaba, muchos conceptos teóricos que incluyen las carreras de computación e informática, y al llevarlos a la práctica, realmente se convierten en una práctica muy enriquecedora para los desarrolladores.

Además el RAT incluye múltiples aplicaciones y ha sido traducida a varios, idiomas: español, alemán, inglés e italiano, ha tomado relevancia mundial y está siendo utilizada en otras universidades del mundo, y es descargada desde el sitio del Laboratorio Virtual de Bases de Datos de la Escuela de Informática.. Como herramienta ya ha sido utilizada en los cursos de bases de datos de la Escuela de Informática de la UNA para enseñar álgebra relacional de forma muy especializada, como se ha mostrado, pero podría ser utilizada como una valiosa herramienta para enseñar teoría de conjuntos, en cursos de matemática, pues facilita realizar operaciones en conjuntos como la intersección, unión o diferencia de conjuntos o combinaciones de ellas.

Los resultados de este proceso de investigación y desarrollo, han sido muy halagadores, realmente se ha generado, a partir del

esfuerzo de profesores y el apoyo estudiantil una herramienta que mejora los procesos de enseñanza aprendizaje del álgebra relacional, aplicada a las bases de datos, esperamos en el futuro seguir aportando al conocimiento de los estudiantes.

4. Bibliografía

- Aho, Alfred V. (2008). *Compiladores, principios, técnicas y herramientas*. México: Pearson Educación
- B. Tucker Allen, Noonan Robert E. (2002). *Programming languages: principles and paradigms*. Estados Unidos: McGrawHill
- Chomsky, N. (1975) *The Logical Structure of Linguistic Theory*. Estados Unidos: Facsimile.
- E.F. Cood.(1990). *The Relational Model for Database Management: version 2*. Estados Unidos: Addison Wesley.
- Gamma Erich. (1994). *Design Patterns Elements of reusable Object-Oriented Software*. Estados Unidos: Addison Wesley.
- Hopcroft, Jonh E. (2001). *Introduction to Automata Theory, Lenguajes, and Computation*. Estados Unidos: Addison Wesley.
- Joyanes Aguilar, Luis. (2007). *Estructura de datos en C++*. España: McGrawHill.
- Paul R. Halmos, (2000). *Naive Set Theory*. Estados Unidos: Springer.
- Silberschatz, Korth, Sudarshan. (2006). *Fundamentos de bases de datos*. Estados Unidos: McGrawHill.