

# Hacia la aplicación de consultas recursivas en bases de datos deductivas

Héctor P. Liberatori<sup>1</sup> & José H. Paganini<sup>2</sup> & Sebastián M. Figueroa<sup>3</sup>

(1) *Cátedra de Base de Datos, Facultad de Ingeniería, Universidad Nacional de Jujuy.*  
*hliberatori@fi.unju.edu.ar*

(2) *Cátedra de Lógica Computacional, Facultad de Ingeniería, Universidad Nacional Jujuy.*  
*jhpaganini@fi.unju.edu.ar*

(3) *Cátedra de Lógica Computacional, Facultad de Ingeniería, Universidad Nacional Jujuy.*  
*tkdfigueroa@fi.unju.edu.ar*

**RESUMEN:** desde hace más de tres décadas que se desarrollan de forma paralela las teorías que sustentan las bases de datos relacionales y las bases de datos deductivas. El poder expresivo de la programación lógica, sobre la que se sustentan las bases de datos deductivas, supera al que se puede tener de forma nativa en una base de datos relacional; y puede ser utilizado para brindar soluciones novedosas a problemas de las bases de datos relacionales. El presente trabajo le da un tratamiento teórico a las bases de datos deductivas y a la forma de obtener consultas recursivas en las mismas, empleando además el lenguaje SQL y el lenguaje Datalog. Las consultas son optimizadas utilizando los métodos de Conjunto Mágico y Filtrado Estático.

## 1 INTRODUCCIÓN

La conexión entre “Lógica y Bases de Datos” comienza en la reunión celebrada en Toulouse en 1977 Gallaire, H.; Minker J. (1978). Allí se establece la equivalencia entre una base de datos relacional y la teoría lógica de primer orden, con lo que se plantea la definición de relaciones en función de otras, es decir, con definiciones intencionales en vez de extensionales.

Los sistemas basados en el conocimiento han sido un área de investigación importante desde más de treinta años, aunque recientemente la perspectiva ha cambiado ligeramente y se llaman bases de conocimiento Levesque, H. J.; Lakemeyer, G. (2001). Los sistemas expertos de los años setenta y ochenta han dado lugar a las bases de conocimiento, pero con una perspectiva más informacional, a veces llamadas bases de datos expertas, se trata de sistemas que almacenan reglas, además de datos.

En realidad los modernos decision support systems empresariales no son más que sistemas expertos sobre un dominio bastante particular (el entorno del negocio), que además tienen una interfaz muy fluida con la fuente de información organizacional o el almacén de datos. Este tipo de

herramientas pueden significar un paso intermedio hacia las bases de conocimiento del futuro Leondes, C. T. (2000).

El mayor inconveniente de las bases de conocimiento (al igual que los sistemas expertos) es que el conocimiento (el conjunto de reglas) se ha de incorporar manualmente. Esto hace que la creación de una base de conocimiento sea un proceso lento y costoso.

El hecho de que esta área se esté revitalizando recientemente se debe a la combinación de las bases de conocimiento con las bases de datos deductivas, bases de datos activas, bases de datos inductivas y las bases de datos temporales Tansel, A. (1993); junto con la importación y exportación de ontologías Bouguettaya, Athman (1999) (posiblemente en XML) para potenciar las posibilidades de los sistemas y facilitar su desarrollo. También es muy importante la incorporación de metainformación en estos sistemas, es decir, reglas que versen sobre el grado de veracidad de otros datos, su aplicabilidad, su contexto, etc. Por lo que en el futuro se verá un gran avance de las bases de conocimiento, incorporando diferentes procesos de adquisición, extracción, recuperación e intercambio de información extensional (factual) o intensional (en forma de reglas, conceptos o entidades).

Finalmente, el salto a la industria y comercialización masiva de estos sistemas vendrá cuando se puedan amalgamar de forma eficiente con los sistemas de gestión de bases de datos. El objetivo es superar las limitaciones de los sistemas de gestión de bases de datos (dificultades para tratar reglas declarativas, metainformación e inconsistencias) y de los sistemas de conocimiento (problemas de actualización de la información, de la asimilación de conocimiento, mantenimiento de la integridad, robustez y consulta). El interés está por lo tanto en combinar la teoría de los modelos relacionales y objetuales en su versión más amplia (incluyendo reglas deductivas y reactivas) junto con los sistemas de conocimientos generales, capaces de tratar información difusa, temporal, no fiable e inconsistente Wagner G. (1998).

### 1.1 Propuesta

La propuesta en este trabajo es analizar mediante ejemplos sencillos la posibilidad de utilizar estos métodos en una base de datos relacional de educación, para solucionar problemas de consultas complejas en tablas autorreferenciales, con estructura de árbol, etc.

## 2 ESTRUCTURA DE UNA BASE DE DATOS DEDUCTIVA

Una base de datos deductiva se compone de 3 conjuntos finitos: un conjunto de hechos, un conjunto de reglas deductivas y un conjunto de restricciones de integridad:

- Hechos: se especifican de manera similar a las relaciones, excepto que no es necesario incluir los nombres de los atributos. Una tupla en una relación describe algún hecho del mundo real cuyo significado queda determinado en parte por los nombres de los atributos. En una base de datos deductiva, el significado del valor de un atributo en una tupla queda determinado exclusivamente por su posición dentro de la tupla.
- Reglas: presentan un parecido a las vistas relacionales, especifican relaciones virtuales que no están almacenadas, pero que se pueden formar a partir de los hechos, aplicando mecanismos de inferencia basados en las especificaciones de las reglas y bajo el control de las restricciones de integridad. La principal diferencia entre las reglas y las vistas, es que en las primeras puede haber recursión y por lo tanto, pueden producir relaciones

virtuales que no es posible definir en términos de las vistas relacionales.

- Restricciones de integridad: son las condiciones que cada estado de la base de datos debe satisfacer, por lo que si un determinado estado se pudiese derivar a un estado nuevo que no cumpliera dichas restricciones de integridad, no se produciría la transición entre estados.

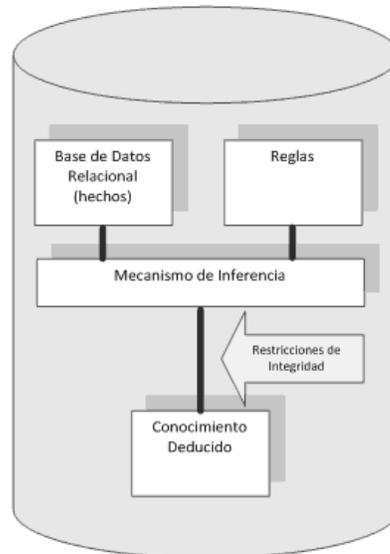


Figura 1. Estructura de una base de datos deductiva

### 2.1 Modelo

El modelo empleado en las bases de datos deductivas está muy relacionado con:

- el modelo de datos relacional,
- el formalismo del cálculo relacional,
- la programación lógica,
- el lenguaje Prolog y
- el Datalog.

La evaluación de los programas en Prolog se basa en una técnica llamada encadenamiento hacia atrás, que implica una evaluación descendente de los objetivos. En las bases de datos deductivas que emplean Datalog, el enfoque es hacia el manejo de grandes volúmenes de datos almacenados en una base de datos relacional; para ello se han desarrollado técnicas de evaluación denominadas encadenamiento hacia adelante.

Al ser Datalog un lenguaje basado en la lógica, los programas se construyen a partir de objetos básicos llamados fórmulas atómicas.

Las fórmulas atómicas son literales de la forma:

$$p(a_1, a_2, \dots, a_n) \quad (1)$$

donde:

p: es el nombre del predicado

n: es el número de argumentos del predicado.

### 3 LENGUAJES: SQL, PROLOG, DATALOG

Desde 1996 con la inclusión de la opción PSM (Persistent Stored Modules) al SQL, éste se vuelve computacionalmente completo gracias a las estructuras clásicas de programación: secuencia, decisión e iteración. Se puede distinguir entre el SQL “puro” y el SQL + PSM. Los PSM se han incorporado en diversos sistemas de gestión de bases de datos. En Oracle por ejemplo la opción procedimental se llama PL/SQL y en SQL Server TSQL.

Con la incorporación de la opción procedimental es posible resolver prácticamente cualquier consulta por medio de SQL debido a que las funciones, que se programan en PSM, pueden ser invocadas desde las consultas. Estas funciones pueden ser tan complejas como se requiera, incluso pueden ser recursivas. Por otro lado el aspecto de la optimización de consultas en los sistemas de gestión de bases de datos es un problema prioritario y en constante investigación. Aunque la recursión es una técnica poderosa para dar solución a determinados problemas tiene como contraparte su enorme costo computacional. Varios sistemas gestores de bases de datos poseen operadores especializados para lograr la recursión en SQL puro, por ejemplo:

- Oracle, CONNECT BY
- DB2 WITH

Muchos autores acuden al concepto de cierre transitivo para abordar consultas recursivas en bases de datos pero orientadas a lenguajes tipo Prolog. Prolog es un lenguaje basado en reglas utilizado en el campo de la inteligencia artificial. En este tipo de lenguajes el planteamiento de expresiones recursivas es natural. Prolog se ha extendido para consultas en bases de datos, dicha extensión se conoce como Datalog.

Rosenthal (1986) presenta algoritmos eficientes para la implementación de la operación cierre transitivo y como éstos pueden incorporarse en el optimizador de un sistema de gestión de base de datos. Introduce el concepto de transversal recursión el cual es una generalización del cierre transitivo y que permite mayor flexibilidad para el planteamiento de consultas recursivas.

James (1982) propone operadores especializados para “tablas con árboles” y Biskup (1990) propone extender SQL con un conjunto de

operadores que faciliten la concepción de consultas para tablas “basadas en grafos”. Tal como lo exponen los autores, su lenguaje no pretende ser una alternativa para lenguajes de consulta relacionales, sino un lenguaje complementario en el cual las consultas recursivas son el objetivo.

Partiendo de la propuesta del SQL-99 para la concepción de consultas recursivas, Ordonez (2005) trata los aspectos relativos a la optimización de tales consultas.

Celko (2005) propone un método denominado “anidado de árboles”, sin embargo es complejo manejar ciertas consultas en él, por ejemplo para hallar la altura del árbol (el cual se genera a raíz de una tabla que exhibe una relación recursiva, se requiere una reunión de la tabla consigo misma más una operación de grupo (GROUP BY)

Todas las propuestas mencionadas conciben un lenguaje especializado para tratar consultas recursivas o proponen extender el SQL o el álgebra relacional con operadores especializados

### 4 INTERPRETACION DE REGLAS

Existen dos alternativas para interpretar el significado teórico de las reglas:

- Teoría de la Demostración
- Teoría de Modelos

#### 4.1 Teoría de la demostración

En esta interpretación se consideran los hechos y las reglas como enunciados verdaderos o axiomas. Los axiomas base no contienen variables, los hechos son axiomas base que se dan por ciertos.

Las reglas se denominan axiomas deductivos, ya que pueden servir para deducir hechos nuevos. Con estos axiomas se pueden construir demostraciones que deriven hechos nuevos a partir de los ya existentes.

La interpretación por la teoría de la demostración ofrece un enfoque procedimental o computacional para calcular una respuesta a una consulta Datalog. Al proceso de demostrar si un determinado hecho (teorema) se cumple, se lo conoce también como demostración de teoremas.

#### 4.2 Teoría de modelos

En esta teoría se considera un dominio finito o infinito de valores constantes, se le asignan a un predicado todas las combinaciones posibles de valores como argumentos. Luego se especifican las combinaciones de argumentos que hacen que el predicado sea verdadero y todas las demás combinaciones que hacen que sea falso.

Si esto se realiza con todos los predicados, se habla de una interpretación del conjunto de predicados.

A una interpretación se le llama modelo de un conjunto específico de reglas, si esas reglas siempre se cumplen en esa interpretación.

En el enfoque según la teoría de modelos, el significado de las reglas se establece proporcionando un modelo para dichas reglas.

Un modelo es mínimo para un conjunto de reglas si no es posible cambiar ningún hecho de verdadero a falso y seguir teniendo un modelo para esas reglas.

## 5 MECANISMOS DE INFERENCIA

Otra forma para interpretar el significado de las reglas implica definir un mecanismo de inferencia que el sistema utilice para deducir hechos a partir de reglas. Este mecanismo define una interpretación computacional del significado de las reglas.

Los mecanismos de inferencia se fundamentan en principios de la inteligencia artificial, basándose en mecanismos de inferencia tales como: inferencia ascendente (encadenamiento hacia adelante), inferencia descendente (encadenamiento hacia atrás).

### 5.1 Mecanismo de inferencia ascendente (encadenamiento hacia adelante)

En este mecanismo el motor de inferencia parte de los hechos y les aplica las reglas para generar hechos nuevos, que se comparan con el predicado que es el objetivo de la consulta y verifica si coinciden. El término encadenamiento hacia adelante indica que la inferencia avanza de los hechos hacia el objetivo. A esta estrategia de búsqueda también se la denomina Bottom-Up.

### 5.2 Mecanismo de inferencia descendente (encadenamiento hacia atrás)

En este mecanismo el motor de inferencia parte del predicado que es el objetivo de la consulta, e intenta encontrar coincidencias con las variables que conduzcan a hechos válidos de la base de datos. El término encadenamiento hacia atrás indica que la inferencia retrocede desde el objetivo buscado para determinar hechos que lo satisfagan. En este enfoque no se generan explícitamente hechos, como en el enfoque anterior. A esta estrategia de búsqueda también se la denomina Top-Down.

## 6 EVALUACIÓN DE CONSULTAS RECURSIVAS

Una de las características más notables de los sistemas de bases de datos deductivas es su soporte para la recursión, mediante definiciones de reglas recursivas y por lo tanto, también de consultas recursivas. Una regla es recursiva cuando el predicado de la cabeza aparece también en el cuerpo. Existen diferentes tipos de reglas recursivas:

### 6.1 Linealmente recursiva

Cuando el predicado recursivo aparece una y sólo una vez en la parte RHS de la regla. Por ejemplo:  $mg(X,Y):-padre(X,XP),padre(Y,YP),mg(XP,YP)$  es una regla lineal en la que el predicado  $mg$  (primos de la misma generación), se utiliza sólo una vez en la parte RHS. La regla dice que  $X$  e  $Y$  son primos de la misma generación si sus padres son primos de la misma generación.

### 6.2 Linealmente recursiva a izquierda

Se presenta de la siguiente manera  $antepasado(X,Y):-antepasado(X,Z),padre(Z,Y)$

### 6.3 Linealmente recursiva a derecha

Presenta la siguiente forma:  $antepasado(X,Y):-padre(X,Z),antepasado(Z,Y)$

### 6.4 No linealmente recursiva

Cuando el predicado recursivo no aparece una y solo una vez en la parte RHS de la regla. Por ejemplo:  $antepasado(X,Y):-antepasado(X,Z),antepasado(Z,Y)$

## 7 PROCESAMIENTO DE CONSULTAS RECURSIVAS

Las reglas recursivas presentan el problema de caer en bucles infinitos. Se han propuesto muchos métodos para resolver este problema, los cuales dieron como resultado una gran variedad de algoritmos que agregan la dificultad de elegir el adecuado que se deba aplicar.

El enfoque del procesamiento de consultas recursivas se clasifican de acuerdo a:

- Objetivo de la consulta, que se puede descomponer en dos métodos:
  - Evaluación: se crea un esquema de evaluación de consultas que dé respuesta a la consulta y queda definida por: el dominio de aplicación y por un algoritmo que resuelva las

- consultas dadas por un conjunto de reglas.
- Reescritura: se optimiza el esquema en una estrategia más eficiente, reescribiendo las reglas en función del argumento instanciado que proporciona la consulta.
  - Mecanismo de inferencia que se utiliza para deducir hechos a partir de reglas. Los dos tipos principales de mecanismos de inferencia computacional se basan en la interpretación de reglas:
    - Encadenamiento hacia adelante (Bottom Up): el motor de inferencia parte de los hechos y aplica las reglas para generar hechos nuevos, éstos se comparan con el predicado que es el objetivo de la consulta para comprobar si coinciden.
    - Encadenamiento hacia atrás (Top Down): parte del predicado que es el objetivo de la consulta e intenta encontrar coincidencia con las variables que conduzcan a los hechos válidos en la base de datos.

La siguiente tabla muestra los aspectos definidos con anterioridad para la clasificación de las técnicas de consulta con ejemplos de los algoritmos:

MECANISMO DE INFERENCIA \ OBJETIVO	Evaluación	Reescritura
Bottom Up	Naive	Magic Sets
	Semi-Naive	Counting Sets (Conjuntos de conteo)
		Static Filtering
Top Down	Query Sub Query	Reducción de Variables
		Reducción de Constantes

Tabla 1. Clasificación de las consultas recursivas

A continuación se describen los métodos de Conjunto mágico y Filtrado estático mencionados en la Tabla 1.

### 7.1 Magic Sets (Conjunto Mágico):

Esta técnica intenta resolver el problema que a veces se presenta cuando una consulta no solicita toda la relación que corresponde a un predicado intencional, sino un subconjunto pequeño de dicha relación. Dadas las siguientes reglas:

$mg(X, Y) :- plana(X, Y)$

$mg(X, Y) :- arriba(X, U), mg(U, V), abajo(V, Y)$

El predicado  $mg$  significa “primo de la misma generación”, y la cabeza de ambas reglas es la fórmula atómica  $mg(X, Y)$ .

Los demás predicados de las reglas son plana, arriba y abajo; que están almacenados extensionalmente como hechos. El predicado  $mg$  es intencional, o sea que está definido por reglas.

En la consulta  $mg(John, Z)$  “quienes son los primos de John de su misma generación”, la respuesta se obtiene examinando la parte de la base de datos que sea relevante; en este caso, la parte relacionada con los individuos que están emparentados con John.

Una búsqueda descendente o de encadenamiento hacia atrás, partiría de la consulta como objetivo y utilizaría las reglas de la cabeza hacia el cuerpo para crear más objetivos. Ninguno de estos objetivos sería irrelevante para la consulta, aunque algunos podrían hacer explorar caminos equivocados.

Una búsqueda ascendente o de encadenamiento hacia adelante, procede desde los cuerpos de las reglas hacia las cabezas, haría inferir hechos  $mg$  que jamás se considerarían en la búsqueda descendente. No obstante, la evaluación ascendente es deseable porque evita problemas de ciclos y cálculos repetidos que son inherentes al enfoque descendente.

Además el enfoque ascendente permite utilizar operaciones que obtienen un conjunto de valores a la vez, como las reuniones relacionales.

La técnica de reescritura de reglas, permite armar las mismas en función exclusiva de la forma de la consulta. Considera que argumentos del predicado están enlazados a constantes y cuáles son variables, de modo que se combinen las ventajas de los métodos descendente y ascendente.

Esta técnica se centra en el objetivo inherente de la evaluación descendente, pero la combina con la ausencia de ciclos, la facilidad de establecer la terminación y la eficiencia de la evaluación ascendente.

### 7.2 Método de filtrado estático (static filtering)

El método de Filtrado estático, que introduce una optimización a los Métodos de Evaluación.

El método construye un grafo de axioma\_relación a partir de un programa y una meta.

La optimización se logra imponiendo restricciones, las cuales deben ser cumplidas por las tuplas de la solución, es decir, los filtros se propagan desde el puerto de salida del nodo meta hacia todos los nodos relación, quedando las variables ligadas.

Las tuplas que no cumplan la condición son cortadas, tan pronto como sea posible, en la primera etapa de su flujo hacia el nodo meta. El

método finaliza cuando un nodo de axioma no genera nuevas tuplas.

La estrategia asume un proceso de evaluación "Bottom\_Up" a partir de un programa y una meta, construye un grafo llamado grafo axioma\_relación, para ello se efectúa una reescritura de los argumentos para poder diferenciar el mismo argumento, en diferentes ocurrencias del mismo predicado.

## 8 CONCLUSIONES

Los Sistemas de bases de datos deductivas combinan un estilo de programación lógica para formular consultas y restricciones con la tecnología de base de datos para almacenar datos de una manera eficiente y confiable en almacenamiento masivo. La contribución principal de estos sistemas es la capacidad de especificar reglas recursivas y de proporcionar un marco de referencia para inferir información nueva con base a las reglas especificadas; es decir, estos sistemas buscan derivar nuevos conocimientos a partir de datos existentes proporcionando interrelaciones del mundo real en forma de reglas, procesándolas dentro de un sistema de gestión de base de datos en vez de un subsistema separado.

Los Sistemas de bases de datos deductivas utilizan Modelos de Datos que soportan valores compuestos en los atributos, además de aparecer nuevos constructores que operan sobre datos complejos así como también en nuevas operaciones como recursión y negación.

Los Modelos de Datos se amplían aceptando datos complejos Orientados a Objetos y las técnicas de evaluación se mejoran con la aparición de nuevos lenguajes que hacen posible su implementación, estos sistemas también evitan el problema de incompatibilidad de impedancia que se presenta cuando se trata de realizar una interfase para lenguajes declarativos y procedurales, diseñando e implementado nuevas arquitecturas de sistemas de bases de datos.

Los sistemas de bases de datos deductivas pueden ser utilizados en diversas áreas de aplicación como: áreas de la inteligencia artificial, sistemas expertos, representación del conocimiento, tecnología de agentes, sistemas de información, por nombrar algunas. En el caso de la propuesta realizada en este trabajo, se realizaron pruebas sobre ejemplos sencillos aplicando reglas para producir un conocimiento nuevo. De esta forma se genera la base de datos intensional para satisfacer consultas complejas. También se aplicaron los métodos del Conjunto mágico y

Filtrado estático para optimizar a las consultas realizadas. Los resultados de este trabajo nos permite asegurar que las técnicas estudiadas serán aplicadas con éxito sobre la base de datos relacional de educación en la cual ya se ha diseñado los modelos conceptual, lógico y físico.

## 9 BIBLIOGRAFÍA

- Gallaire, H.; Minker, J. "Logic and Databases" Plenum Press 1978.
- Levesque, H.J.; Lakemeyer, G. "The Logic of Knowledge Bases" MIT Press, 2001.
- Leondes, C.T. (ed). "Knowledge-Based Systems. Vols. I-IV" Harcourt International, Academic Press 2000.
- Tansel, A.; Clifford, J.; Gadia, S. et al. (eds.). "Temporal Databases" Benjamin Cummings, 1993.
- Bouguettaya, Athman (Editor). "Ontologies and Databases", Kluwer Academic Publishers, 1999.
- Wagner, G. "Foundations of Knowledge Systems - with Applications to Databases and Agents", Kluwer Academic Publishers, 1998.
- Rosenthal Arnon, Heiler Sandra, Umeshwar Dayal y Frank Manola. "Traversal Recursion: A Practical Approach to Supporting Recursive Applications", SIGMOD Conference, 1986, pp. 166-176.
- James G. y Stoeller W. "Operations on Tree-structured Tables", X3H2-26-15 Standards Communication, 1982.
- Biskup Joachim, Uwe Räsch y Holger Stiefeling. "An Extension of SQL for Querying Graph Relations", Computer Languages, Volumen 15, Número 2, 1990, pp. 65-82.
- Ordonez Carlos. "Optimizing Recursive Queries in SQL", SIGMOD Conference, 2005, pp. 834-839.
- Celko Joe. "SQL for Smarties", Tercera Edición, Morgan-Kaufmann, San Francisco, California, 2005, pp. 623-640.