

EL TRATAMIENTO DE LOS TÉRMINOS BORROSOS EN MANEJADORES DE BASES DE DATOS RELACIONALES

CARLOS ZAPATA

Escuela de Sistemas. Facultad de Minas. Universidad Nacional de Colombia, Medellín.

CLAUDIA JIMÉNEZ

Postgrado en Ingeniería de Sistemas. Facultad de Minas. Universidad Nacional de Colombia, Medellín.

JUAN VELÁSQUEZ

Postgrado en Ingeniería de Sistemas. Facultad de Minas. Universidad Nacional de Colombia, Medellín.

Recibido para revisar 4 de Septiembre de 2002, aceptado 15 de Diciembre de 2002, versión final 20 de Enero de 2003.

RESUMEN: Los Manejadores de Bases de Datos Relacionales (MBDR) convencionales presentan dificultades cuando se ven abocados a entregar información solicitada en forma vaga, imprecisa o borrosa y los investigadores en el mundo se han percatado de esta situación. En este artículo se resumen dos vertientes de solución a este problema en el mundo: las bases de datos borrosas y la incorporación de características borrosas en los MBDR convencionales. Además, se presenta una nueva propuesta metodológica que incorpora operadores borrosos en MBDR convencionales.

PALABRAS CLAVES: Bases de Datos Relacionales Borrosas, Búsquedas Borrosas, Operadores Borrosos, Lenguaje Estructurado de Búsqueda.

ABSTRACT: Conventional Relational Database Management Systems (RDMS) presents difficulties when they try to give information which is requested in a vague, imprecise or fuzzy way, and world researchers have seen this situation. Some trends about the solution of this problem in the world: Fuzzy Databases and Fuzzy Features Incorporation in conventional RDBMS are summarized. Furthermore, in this paper a new methodological proposal, which incorporates fuzzy operators in conventional RDBMS, is presented.

KEYWORDS: Fuzzy Relational Databases, Fuzzy Queries, Fuzzy Operators, Structured Query Language.

1. INTRODUCCIÓN

La información es ahora uno de los más importantes activos que poseen las organizaciones de cualquier tipo en el mundo; sin ella no se podrían adquirir los conocimientos suficientes para generar los avances en todas las áreas del saber. En especial la recopilación de los diferentes tipos de información en grandes Bases de Datos se ha constituido en uno de las principales ventajas competitivas de los negocios

en el mundo puesto que, dados los grandes avances que han alcanzado los Manejadores de Bases de Datos en aspectos claves de este tema como la seguridad, confiabilidad, y accesibilidad, se han especializado cada vez más los recursos disponibles para su manejo.

Sin embargo, y pese a estos avances, se ha descuidado a lo largo de la evolución de dichos manejadores un aspecto que incide en la recuperación de la información contenida en las bases de datos actuales y su relación con los usuarios de las mismas: el carácter vago o impreciso que puede tener una búsqueda de datos

realizada por un usuario desprevenido y carente de los conocimientos necesarios para dominar los lenguajes empleados en la elaboración de dichas búsquedas. Hasta ahora, el usuario que se aproxima a las Bases de Datos debe ser alguien capacitado para ello, con una serie de conocimientos previos que le permitan extraer de la información allí contenida conocimientos que puedan ser valiosos para lo que pretende analizar de su organización. El usuario que no posea tales características está condenado a depender de otros para la extracción de la información relevante de su negocio. Además, los sistemas Manejadores de Bases de Datos Relacionales (MBDR) de la actualidad presentan inconvenientes en presencia de solicitudes de búsqueda vagas, imprecisas o simplemente "borrosas", y por ello, si se quiere obtener alguna respuesta válida de dichos manejadores, se deben "precisar" los términos, de forma que el lenguaje que los soporta pueda ser capaz de interpretar la necesidad para convertirla en un cierto número de tuplas que cumplan con el requerimiento inicial. En síntesis, el sistema puede no entender la búsqueda solicitada porque no soporta "lenguaje natural".

Desde hace algunos años, los investigadores han formulado posibles soluciones a este inconveniente y se han propuesto algunos trabajos. En este artículo se hará una revisión de la literatura existente y se presentará un trabajo que pretende suministrar un nuevo enfoque en relación con la adición de términos borrosos en manejadores de bases de datos relacionales. Para ello, el artículo consta de las siguientes secciones, además de esta introducción: Una segunda sección donde se introducen algunos de los términos básicos en el manejo de la lógica borrosa y las bases de datos relacionales; una tercera sección donde se expone la problemática asociada con la incorporación de términos borrosos en manejadores de bases de datos relacionales; una cuarta sección que presenta el estado del arte en relación con los términos borrosos en manejadores de bases de datos relacionales; una quinta sección donde se presenta una nueva metodología de solución del problema. Finalmente en la sexta sección se presentan las conclusiones que se desprenden de la nueva metodología presentada.

2. LÓGICA Y SISTEMAS MBDR

2.1. Lógica borrosa

Zadeh (1965) presentó un trabajo inicial en relación con la lógica borrosa como una forma de generalización de la lógica Booleana tradicional que es completamente excluyente en relación con la pertenencia de un elemento a un determinado conjunto. Cuando se habla de la teoría de conjuntos tradicional, un elemento puede pertenecer o no pertenecer a ese conjunto; en su trabajo, Zadeh propone una generalización de esta teoría cuando presenta un ejemplo con animales de la siguiente manera: "Es claro para la lógica tradicional que caballos, perros y aves pertenecen completamente al conjunto de los animales; igualmente claro es el hecho de que rocas, plantas y fluidos no pertenecen a ese conjunto. Sin embargo, ¿qué ocurre con otras especies como las estrellas de mar y las bacterias?". En opinión de Zadeh, sí pertenecen al conjunto de los animales pero *parcialmente*, lo que permite introducir el concepto de *Grado de Pertenencia* a un conjunto borroso, que se suele denotar en la literatura por la letra griega μ .

En la lógica Booleana sólo existen dos valores para el grado de pertenencia a un conjunto: el valor cero (*0*) para denotar aquellos elementos que no pertenecen al conjunto definido y el valor uno (*1*) para denotar aquellos elementos que definitivamente pertenecen al conjunto. Es claro en estas condiciones que un elemento no puede pertenecer en forma simultánea a un conjunto y a su complemento. En el caso de la lógica borrosa sí se puede dar el hecho de que los elementos pertenezcan a un conjunto y simultáneamente a su complemento *en algún grado*. Esto implica la definición de una determinada función de pertenencia que permita especificar la variación de la pertenencia de dicho elemento al conjunto que estamos tratando de analizar. La lógica borrosa se ha aplicado especialmente a términos lingüísticos que no poseen unos límites claramente definidos, como es el caso de estatura, peso, lejanía, cercanía, etc. Esta situación se puede ilustrar más claramente con un ejemplo: Cuando se habla de personas ALTAS alguien en forma subjetiva podría afirmar que una persona de 140 cm es

realmente baja, mientras que una persona de 180 cm es realmente alta; para quien está haciendo esta definición no existen dudas en relación con el hecho de que la persona de 140 cm tiene un grado de pertenencia de cero (0) al conjunto de las personas ALTAS, en tanto que la persona de 180 cm tiene un grado de pertenencia de uno (1) al conjunto de las personas ALTAS. Sin embargo, ¿qué ocurre con aquellas personas que tienen una estatura mayor de 140 cm y menor de 180 cm? En ese caso no se puede afirmar que completamente sean personas ALTAS ni que sean completamente BAJAS. Estas personas tendrán un grado de pertenencia diferente de cero y uno al conjunto de las personas ALTAS. La función de pertenencia de esas personas al conjunto de las ALTAS podría tener muchas formas diversas; una de las más sencillas se presenta en la Figura 1, en la cual la variación del grado de pertenencia μ en mención se realiza mediante una línea recta que une los dos puntos extremos. Esa función podría tener diferentes formas dependiendo de la interpretación y la forma de cálculo de la función de pertenencia respectiva, para lo cual está disponible una serie de métodos empíricos especialmente y que se pueden consultar en (Li, 1995).

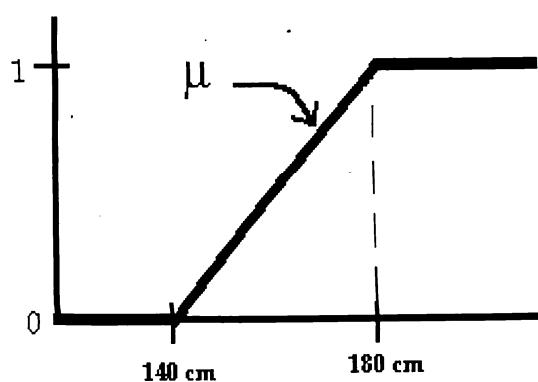


Figura 1. Una interpretación de la función de pertenencia de una persona al conjunto de las ALTAS.

Esta característica de la lógica borrosa ha posibilitado su uso, entre muchas otras aplicaciones, en problemas que presentan variables de tipo lingüístico, o para solucionar problemas con valores matemáticos precisos que

no posean una delimitación clara entre los diferentes conceptos que involucran.

2.2. Bases de datos relacionales

En los comienzos del manejo del software, no existía una necesidad clara de almacenamiento de información, puesto que cada vez que se requería la ejecución de un programa, por lo general aquellos programas de tipo científico, se digitaban directamente algunos pocos datos y con ello se obtenían los resultados esperados de ejecución.

A medida que los programas fueron evolucionando, se fueron requiriendo progresivamente algunos tipos de almacenamiento, básicamente debido a la necesidad de reiteración de los datos de entrada de los diferentes programas. Con ello se dio inicio al manejo de los denominados "Sistemas de Archivos". En dichos sistemas, la información se comenzó a almacenar físicamente en archivos mediante diferentes estructuras, tales como los arrays, las listas enlazadas, los árboles y los grafos. Este tipo de estructuras se pueden consultar en libros de estructuras de datos como (Lipschutz, 1986).

Sin embargo, el surgimiento del modelo relacional (Codd, 1970) revolucionó el almacenamiento de datos al definir una serie de estructuras en forma de arreglos bidimensionales (tablas), en las cuales se podían almacenar los datos sin tener que preocuparse el programador por punteros y elementos de direccionamiento. Una buena descripción de las bases de datos relacionales se puede encontrar en (Date, 1990).

Las bases de datos relacionales permitieron el surgimiento de una serie de lenguajes de consultas basados en dos lenguajes teóricos denominados Álgebra relacional y Cálculo relacional. El más popular y generalizado de esos lenguajes ha sido el SQL (Structured Query Language), un lenguaje declarativo que permite realizar consultas, modificaciones y actualizaciones sobre los datos contenidos en una base de datos de tipo relacional.

El SQL tiene la siguiente sintaxis:

`SELECT atributos`

FROM relaciones
WHERE condiciones lógicas.

Una descripción completa del SQL puede ser consultada también en (*Date, 1990*).

3. PROBLEMÁTICA ASOCIADA CON LA INCORPORACIÓN DE TÉRMINOS BORROSOS EN MBDR

El SQL ha demostrado ser un lenguaje bastante versátil cuando se trata de la recuperación de información de tipo preciso sobre bases de datos relacionales existentes. Su aplicación ha posibilitado el desarrollo de bases de datos de gran tamaño de las cuales se puede extraer bastante información en forma relativamente sencilla. Sin embargo, y pese a que los diferentes Manejadores de Bases de Datos Relacionales (MBDR) de la actualidad poseen una serie de ayudas gráficas y software de apoyo que traduce las consultas necesarias al SQL convencional, aún se requiere de algún conocimiento para extraer la información requerida de las bases de datos.

Además, si el usuario no ha definido en forma precisa su búsqueda, sino que tan sólo tiene una vaga idea de lo que está buscando, el SQL no será su mejor opción y, desafortunadamente, ningún MBDR posee algún mecanismo que le pueda ayudar en la solución a su búsqueda particular. Si el usuario desea emplear SQL para la solución de su inquietud, deberá hacer uso de su capacidad de precisión y restringir el ámbito de su búsqueda; términos de búsqueda como la edad, la estatura, la temperatura o el peso pueden ser términos fácilmente determinables para un elemento perteneciente a una base de datos convencional, pero existen ciertas "categorías" a las cuales no es tan fácil acceder por cuanto requieren un cierto nivel de abstracción y de precisión en la consulta que se está emitiendo.

El siguiente ejemplo puede contribuir a clarificar los alcances del problema:

Un asesor requiere analizar una base de datos para determinar la cantidad de personas jóvenes que hacen parte de la misma y, para emplear el SQL

de un manejador de bases de datos convencional, decide precisar la información afirmando que va a buscar entre los individuos pertenecientes a la base de datos aquellos que sean menores de 25 años, porque considera en forma SUBJETIVA que una persona con una edad superior a esa no puede ser considerada como joven. Sin embargo, si la base de datos pertenece a los habitantes registrados en hogares geriátricos de la ciudad, es posible que su consulta no devuelva ninguna tupla, que no exista coincidencia entre la consulta realizada y los individuos pertenecientes a la base de datos. Sin embargo, lo que probablemente esté buscando el asesor puede pertenecer a las características propias de la base de datos, porque pese a que los individuos no cumplan con la condición que debió precisar el asesor, es posible que aún existan personas JOVENES dentro de esa base de datos; es probable que la mayoría de las personas recluidas en hogares geriátricos de la ciudad excedan los sesenta años, pero que haya un pequeño porcentaje de ellos que se encuentren entre cuarenta y cinco y cincuenta años. En ese caso, y pese a que no se encuentran por debajo de los veinticinco años, que es el parámetro inicialmente precisado por el asesor, aún es posible recuperar algunas tuplas pertenecientes al conjunto de los JOVENES de la base de datos.

En ocasiones no es sólo la imprecisión de la búsqueda sino también la incertidumbre de la información la que puede conducir al uso de términos borrosos. Por ejemplo, es posible que en los archivos correspondientes a los NN de la morgue no se tengan muchas de las características pertinentes a los integrantes de la base de datos; en general, y a no ser por la realización de un examen especial sobre los cuerpos, no se sabrá con exactitud la edad. Probablemente sería mucho más fácil almacenar un término borroso como JOVEN y no un intervalo como "entre los veinte o los veinticinco años" o algo similar. En este caso existen muchos más elementos de imprecisión que pueden dificultar las consultas realizadas.

4. ESTADO DEL ARTE EN RELACIÓN CON LOS TÉRMINOS BORROSOS EN MBDR

Existen básicamente dos tendencias en lo que respecta a la resolución del problema de realización de Búsquedas Borrosas sobre Manejadores de Bases de Datos Relacionales (MBDR):

- Mediante bases de datos borrosas, las cuales pueden poseer atributos de tipo intervalo o cualidad, que permiten en cierta forma manejar la imprecisión del lenguaje natural.
- Mediante la incorporación de características borrosas en los MBDR convencionales.

En lo que respecta al primer enfoque, en (*Chiang et al.*, 1997) se entrega un trabajo teórico con las ecuaciones fundamentales para el estudio de las Bases de datos Relacionales Borrosas, en tanto que en (*Medina et al.*, 1994a y 1994b; *Pons et al.*, 1995; *Galindo*, 1998) se da la definición de nuevos atributos para extender las bases de datos a bases de datos relacionales borrosas. Estos investigadores pertenecen al grupo ARAI (Aproximate Reasoning and Artificial Intelligence) de la Universidad de Granada y sus trabajos han pasado por el planteamiento teórico, llegando incluso a instancias de definición de conceptos y modelamiento en este campo. En (*Damiani et al.*, 1997) se analiza la Programación Orientada a Objetos en bases de datos relacionales borrosas y en (*Hale et al.*, 1995) la Minería de Datos y Seguridad en este tipo de bases, desde el punto de vista de la realización de búsquedas que puedan vulnerar la seguridad de las mismas. Finalmente, en (*Kumar et al.*, 2000) se realiza una caracterización teórica de los modelos de Manejadores de Bases de Datos Relacionales Borrosos de otros autores.

Una de las críticas que se han realizado al manejo de bases de datos que puedan almacenar información de tipo borroso, obedece a que existe en este momento en el mundo una gran cantidad de Bases de Datos que incluyen información "precisa", y con un planteamiento tal se tendrían que adicionar unos tipos "borrosos" a la información preexistente en la Base de Datos, con

lo cual deberían coexistir dos tipos de datos para hacer la recuperación de la información contenida en ella, que podrían tergiversar de una u otra manera la información. Igualmente, otra opción sería la actualización de la información para que tuviera características "borrosas" para efectos de recuperación de la información mediante búsquedas; la principal dificultad se presentaría en las bases de datos de gran tamaño, en las cuales la información recopilada es de tipo preciso y requeriría un esfuerzo adicional considerable para la conversión de la información a las características "borrosas" requeridas.

En lo relativo al segundo enfoque, la incorporación de características borrosas en los MBDR convencionales, (*Bosc y Pivert*, 1995 y 1996) han trabajado ampliamente en la extensión del lenguaje SQL para incorporar características borrosas que permitan el manejo de la imprecisión. (*Kraft y Petry*, 1997), han realizado una síntesis de las dos tendencias en recuperación de información imprecisa, que han denominado precisamente "recuperación de información borrosa y sistemas de bases de datos borrosas", correspondientes con las tendencias mencionadas al inicio de esta sección como las estudiadas a nivel mundial para la realización de búsquedas que involucren términos borrosos. Estas dos tendencias se basan siempre en la preexistencia de una función de pertenencia que permite la realización de las búsquedas, por lo cual la subjetividad de quien ejecuta la búsqueda se encuentra sesgada por el punto de vista de quien programa el modelo, quien de antemano ha debido suponer la función de pertenencia correspondiente. Un ejemplo práctico interesante lo constituye el *SmartRanker* de (*Wang*, 1998), un trabajo sencillo programado en Java y disponible en Internet para realizar simulaciones de ordenamiento sobre bases de datos sencillas con el empleo de operadores borrosos. De nuevo, el autor presupone la función de pertenencia de las variables involucradas, las cuales nunca son preguntadas al usuario.

En cuanto al aspecto comercial, se debe resaltar el hecho de que *Oracle Corporation* (1997) ha incorporado una serie de operadores para manejar las aproximaciones en las búsquedas, tales como:

- CONTAINS: Permite la identificación de una palabra o grupo de palabras dentro de un archivo de texto.
- ACCUMULATE: Permite la identificación de los "mejores" textos que contienen dos cadenas de texto, asignando puntajes a los textos candidatos y acumulando esos puntajes para la determinación del mejor de ellos.
- STEM: Emplea el signo "\$" para expandir la búsqueda que se está realizando a las palabras que tienen una raíz igual a la palabra que se está buscando.
- SOUNDEX: Emplea el signo "!" para expandir la búsqueda que se está realizando a palabras que tienen un sonido similar (en inglés únicamente) a la palabra que se está buscando.
- FUZZY: Emplea el signo "?" para generar expansiones borrosas sobre las palabras buscadas, de modo que se encuentren palabras que se escriban en forma similar a la que se está buscando.

Sin embargo, estos operadores constituyen tan sólo una aproximación de menor cuantía a lo que se puede hacer con la incorporación de búsquedas aproximadas empleando operadores borrosos más consolidados. También comercialmente se han realizado lanzamientos como el de *Sonalysts Inc.* (2000) (ver también *Internet Wire*, 1999), que prometen solucionar el problema de las búsquedas aproximadas mediante un paquete de software que ha resuelto esta situación en forma cómoda para su desarrollo particular. En su software *Fuzzy Query*, cada término lingüístico debe ser definido por el usuario en forma cuidadosa, no sólo en los límites superior e inferior del término buscado, sino también en la forma de la función de pertenencia asociada con el término. Sin embargo, ¿qué pasa si el usuario, como es de suponer en la generalidad de los casos, desconoce los principios fundamentales de la lógica borrosa? ¿Si el usuario no conoce cómo debe ser la función de pertenencia para un caso particular (recta, curva, trapezoidal, triangular, etc.) qué tan confiable puede ser la selección que haga de dicha función en el término borroso buscado?

En la Figura 2, se puede apreciar cómo en el programa *Fuzzy Query de Sonalysts*, la búsqueda correspondiente a los mejores alumnos de una institución se debe calcular mediante una variable

GPA (promedio general del alumno), en la cual se incluye el modificador GOOD ("bueno"). Ese modificador para el promedio GPA debe ser definido mediante una ardua selección de la función de pertenencia más ajustada a la pregunta subjetiva que se está realizando y los valores máximo y mínimo de esa función, que corresponden a los valores que el usuario cataloga como el máximo a partir del cual un promedio es siempre bueno y el mínimo antes del cual el promedio es definitivamente malo. Entre máximo y mínimo, no se puede dar un calificativo específico al promedio, es decir no se puede decir que es bueno completamente ni malo completamente. Esas características en el software de Sonalysts deben ser definidas por el usuario y de su elección dependerá cuáles tuplas van a ser devueltas por el sistema en la búsqueda que se plantea de los estudiantes con los mejores promedios.

Por una u otra vertiente de la solución de este problema se pueden encontrar dificultades de implementación o de conceptualización que harían poco viable el manejo. Sin embargo, como vemos, ya se están adelantando trabajos que podrían en un corto lapso dar un viraje total al manejo de la información para acercarlo un poco al lenguaje natural que suelen emplear algunos usuarios de este tipo de Bases de Datos.

5. UNA NUEVA METODOLOGÍA DE SOLUCIÓN AL PROBLEMA

En (Zapata, 2002) se ha propuesto una nueva sintaxis para el manejo de las búsquedas borrosas sobre MBDR convencionales, que trata de resolver uno de los vacíos existentes en torno de este problema y que se relaciona con la asignación de la función de pertenencia a la búsqueda particular y concreta que se está realizando. En la sección 3 se presentaron algunos ejemplos de la problemática que se enfrenta con la utilización de términos borrosos en el planteamiento de consultas sobre bases de datos convencionales y se argüía la subjetividad de quien está realizando la consulta como uno de los factores que más incidencia tiene en la recuperación de la información a partir de la búsqueda.

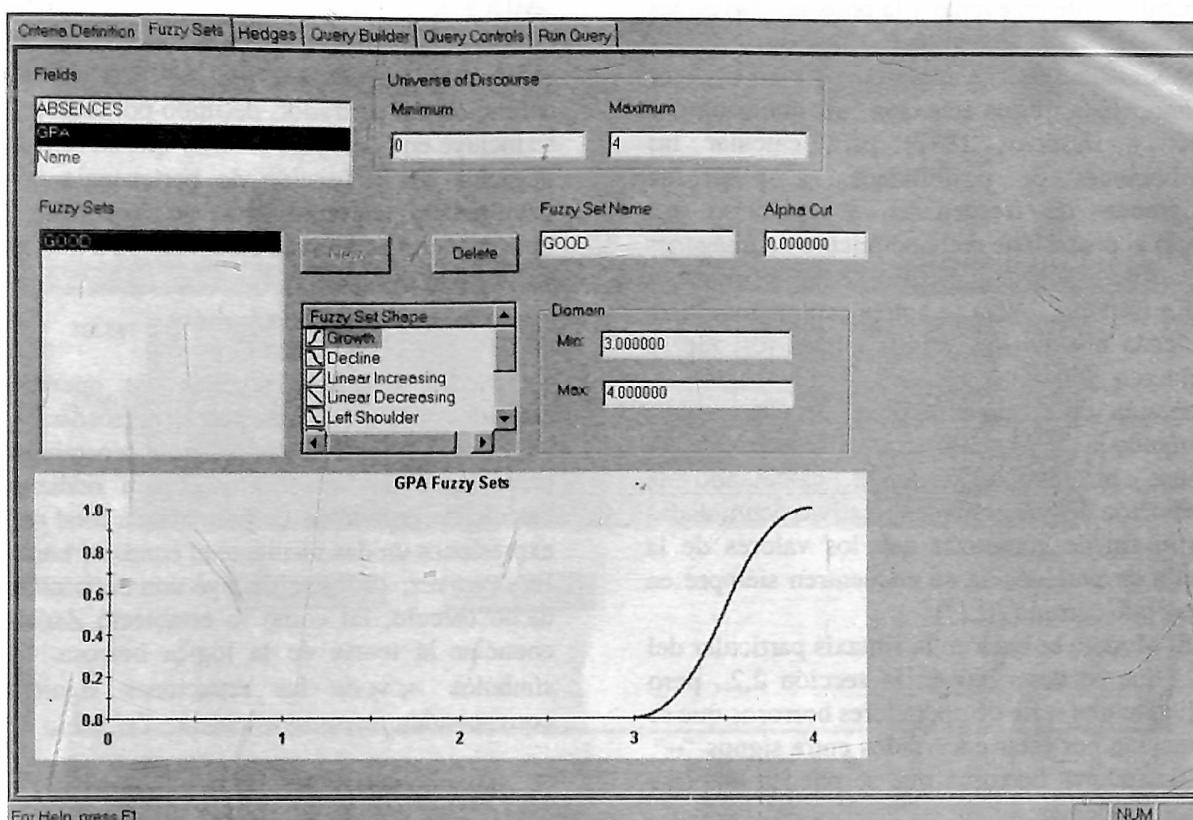


Figura 2. Imagen del Programa Fuzzy Query, de Sonalysts.

Con esta nueva metodología se reconoce la existencia de ciertas tendencias o características en la información que minimizan la subjetividad de quien emite la consulta y le facilitan la labor de búsqueda, pues no requiere que se precisen los términos de la búsqueda. El modelo en cuestión permite al usuario la comodidad de lanzar la búsqueda en un lenguaje un poco más cercano al natural, pero también puede tomar algunas decisiones en relación con dicha búsqueda mediante el análisis minucioso de la información contenida en la base de datos, la cual le sirve de soporte para calcular, con base en la teoría de las posibilidades, la función de pertenencia a la búsqueda particular emitida por el usuario. En (*Dubois y Prade, 1988*) se realiza un completo tratado sobre la Teoría de Posibilidades y su relación con las funciones de pertenencia de la lógica borrosa. Allí se introducen los conceptos de posibilidad y necesidad como complemento a la teoría de probabilidad, con las siguientes ecuaciones fundamentales:

$$\begin{aligned} P(A) + P(\tilde{A}) &= I \\ N(A) + N(\tilde{A}) &\leq I \\ \Pi(A) + \Pi(\tilde{A}) &\geq I \end{aligned}$$

En las cuales $P(A)$ es la probabilidad de A , $N(A)$ es la necesidad de A y $\Pi(A)$ es la posibilidad de A . Para los tres casos, \tilde{A} es el complemento de A . Igualmente, llegan a una clase de medidas de probabilidad tal que:

$$P = \{P / \forall A, N(A) \leq P(A) \leq \Pi(A)\}$$

Finalmente, llegan a demostrar que a partir de las distribuciones de probabilidad se pueden calcular las correspondientes distribuciones de posibilidades, con la ecuación siguiente:

$$\pi(\omega i) = \sum_{j=1}^n \min(p(\omega i), p(\omega j))$$

allí se establece que la posibilidad de un evento resulta de la sumatoria de la comparación entre la

probabilidad de un evento y la probabilidad de los demás eventos de una distribución de probabilidades, escogiendo en cada caso el valor mínimo. Esta última ecuación fue empleada por (Roux y Desachy, 1997) para calcular las distribuciones de posibilidades a partir de histogramas de frecuencias, y aplicaron ese cálculo al proceso de reconocimiento de imágenes por satélite.

En (Zapata, 2002) se aplica esta metodología para cada uno de los términos borrosos que se involucren en una búsqueda, calculando un histograma de frecuencias con diez intervalos, calculando la distribución de posibilidades a partir de ese histograma y luego calculando la distribución de posibilidades relativas acumuladas (con el fin de garantizar que los valores de la función de pertenencia se encuentren siempre en el intervalo cerrado $[0,1]$).

El Modelo se basa en la sintaxis particular del SQL que se describió en la sección 2.2., pero añadiendo una serie de operadores borrosos que se diferencian por estar encerrados entre signos "~~". Los operadores borrosos que se pueden emplear son los siguientes:

~~MUY GRANDE~, ~~MUY PEQUEÑO~, ~~GRANDE~, ~~PEQUEÑO~, ~~CERCANO A~, ~~ALTO~, ~~ALTA~, ~~RAPIDO~, ~~RAPIDA~, ~~MUCHO~, ~~MUCHA~, ~~PEQUEÑA~, ~~BAJO~, ~~BAJA~, ~~LENTO~, ~~LENTA~, ~~POCO~, ~~POCA~, ~~MUY ALTO~, ~~MUY ALTA~, ~~MUY RAPIDO~, ~~MUY RAPIDA~, ~~MUY PEQUEÑA~, ~~MUY BAJO~, ~~MUY BAJA~, ~~MUY LENTO~, ~~MUY LENTA~, ~~MUY POCO~, ~~MUY POCA~, ~~CERCANA A~, ~~ALREDEDOR DE~.

Los operadores asociados con el modificador MUY se calculan normalmente en su función de pertenencia y se modifican sus valores con el concepto de concentración analizado en (Pedrycz, 1993):

$$CON \mu(x) = \mu^2(x)$$

Con el método descrito se construye la distribución de frecuencias relativas de posibilidades acumuladas para los operadores borrosos ~~GRANDE~, ~~ALTO~, ~~ALTA~, ~~RAPIDO~, ~~RAPIDA~, ~~MUCHO~,

~~MUCHA~. Para los operadores ~~PEQUEÑO~, ~~PEQUEÑA~, ~~BAJO~, ~~BAJA~, ~~LENTO~, ~~LENTA~, ~~POCO~, ~~POCA~ se usa la definición del operador definido por Zadeh, que se incluye en (Pedrycz, 1993) y que sirve para la negación de la función de pertenencia de un determinado término borroso (pequeño en oposición a grande, bajo en oposición a alto, etc):

$$\neg \mu_A(x) = 1 - \mu_A(x)$$

Se pueden combinar también los operadores definidos con los conectores borrosos (~Y~, ~O~), cuya función principal es la separación de expresiones de tipo borroso, para realizar el cálculo de la función de pertenencia total de las expresiones unidas mediante el conector borroso. En (Pedrycz, 1993) se incluye una definición de dicho cálculo, tal como lo estableció Zadeh al enunciar la teoría de la lógica borrosa. Los símbolos \wedge , \vee , de las ecuaciones siguientes, corresponden respectivamente a ~Y~, ~O~.

$$\begin{aligned} \mu_{A \cup B}(x) &= \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x) \\ \mu_{A \cap B}(x) &= \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \end{aligned}$$

Además, los operadores borrosos se pueden enlazar con búsquedas precisas mediante los operadores AND y OR del SQL convencional.

De esta manera, una consulta emitida con esta nueva metodología tiene la apariencia de la Figura 3, donde además, se muestra la interfaz del modelo, realizada en Fox Pro 2.6 bajo Windows. Allí se puede apreciar la similitud con el SQL convencional, con la diferencia de que se han añadido algunos operadores entre el símbolo "~~", los cuales corresponden a operadores borrosos, según la descripción que se realizó del modelo. En el ejemplo particular que allí se muestra se está realizando la siguiente pregunta: "Muéstreme la cédula y el nombre de los empleados de Medellín con sueldos altos y jóvenes", a lo cual el usuario debe realizar la siguiente precisión lingüística previa: "Muéstreme la cédula y el nombre de los empleados de la ciudad de Medellín con sueldo grande y edad tiene pequeña". Finalmente, la búsqueda se escribe como se muestra en la Figura 3.

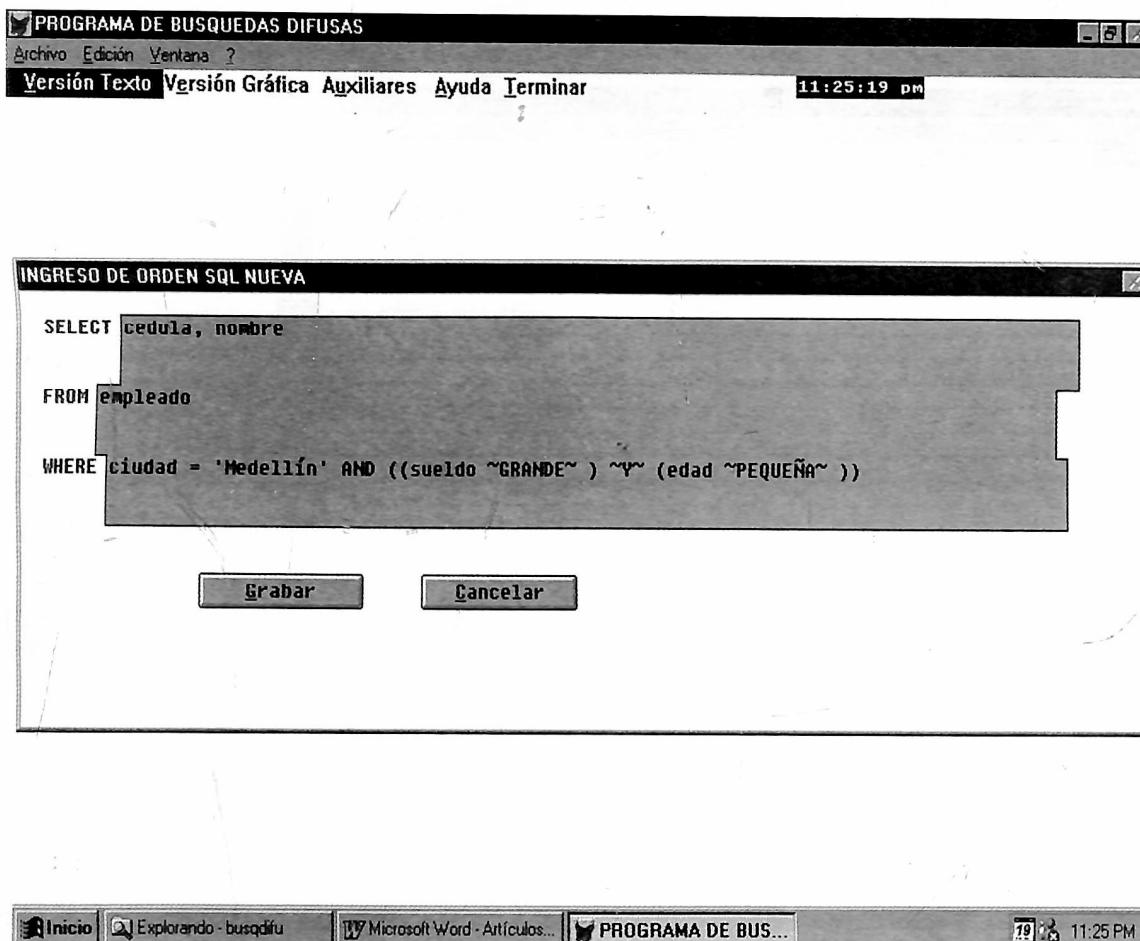


Figura 3. Imagen de una búsqueda borrosa. Modelo de Incorporación de operadores borrosos en MBDR.

En la Figura 4 se muestra la respuesta del Sistema a la búsqueda que planteó el usuario empleando los nuevos operadores borrosos que se adicionaron. Se destaca allí la presencia de un campo denominado "pertotal", que indica el grado de pertenencia total de cada una de las tuplas a la búsqueda planteada, conjuntamente con los demás campos solicitados.

Si bien el usuario aún debe realizar una traducción de sus requerimientos en relación con la búsqueda a un lenguaje tipo SQL, los operadores borrosos definidos permiten abstraer fácilmente su representación, facilitando un poco la escritura de la consulta y permitiendo de paso tomar ciertas conclusiones en relación con los datos incluidos en la base. Una evolución futura de este trabajo podría ser la adecuación de una interfaz en lenguaje natural que permitiese la

codificación interna de la consulta para llegar al resultado anotado.

4. CONCLUSIONES

Los Manejadores de Bases de Datos Relacionales convencionales presentan problemas en el momento de abordar la información de tipo borroso que le sea requerida por un usuario. El tema se ha abordado con dos tendencias principales en el mundo: Bases de Datos Borrosas e Incorporación de Características borrosas en MBDR convencionales.

Ambas soluciones presentan todavía inconvenientes tales como el hecho de que quien las programa debe suponer ciertas características que hacen que finalmente la búsqueda sea completamente sesgada.

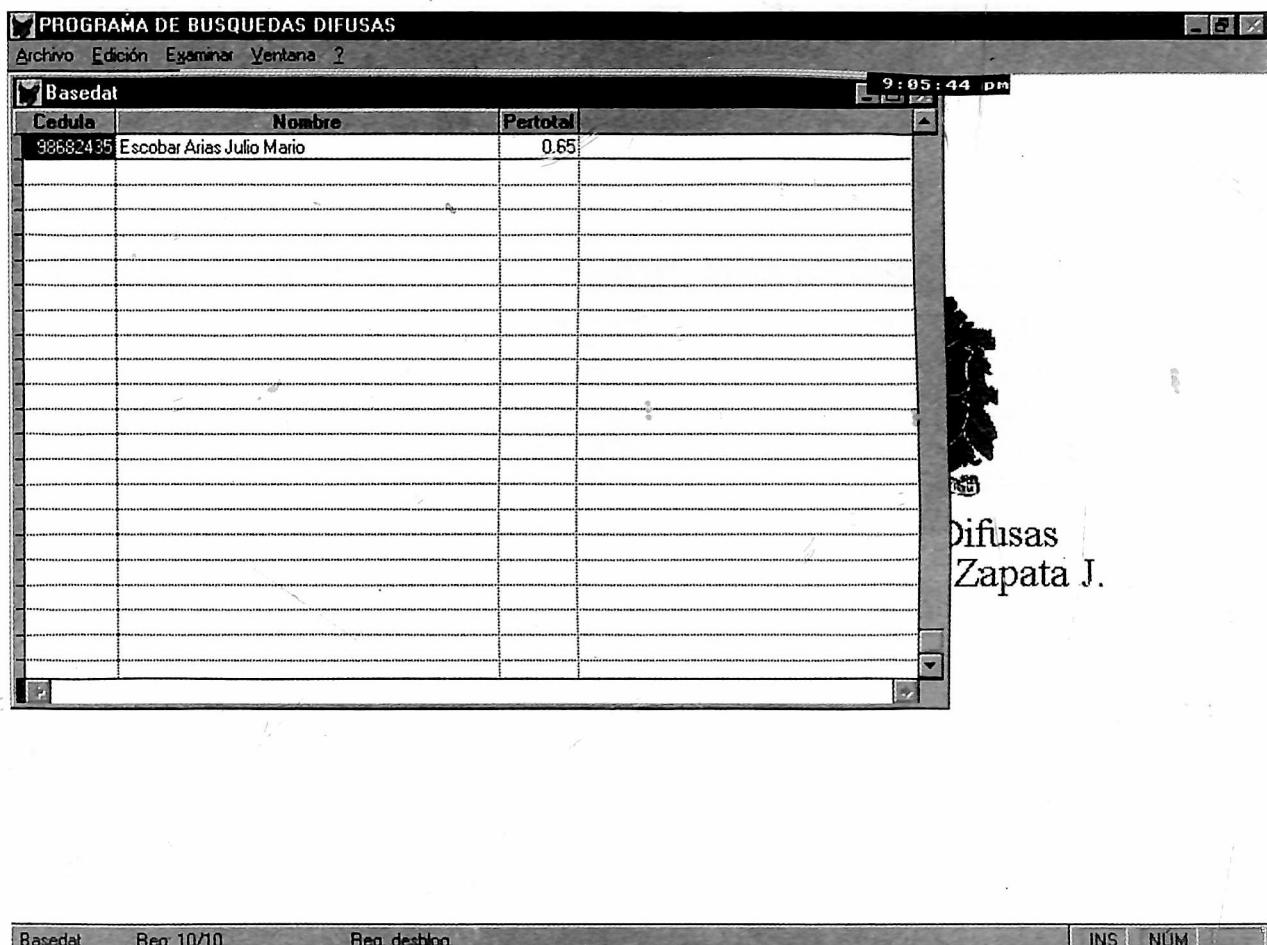


Figura No. 4. Respuesta del sistema a la búsqueda realizada.

Se presentó en este trabajo una nueva propuesta que aborda el tema desde la información que está contenida en la base de datos, de forma que mediante esta información se puedan deducir ciertas características que permitan el empleo de términos borrosos dentro de las búsquedas. El modelo se basa en SQL convencional, al cual se le adicionan operadores de tipo borroso.

Los cálculos de las funciones de pertenencia que realiza el modelo se basan en la teoría de lógica borrosa y en la teoría de posibilidades, a partir de la información que está consignada en la base de datos convencional de la cual se origina.

Futuras líneas de investigación que continúen este trabajo pueden ser: la definición de nuevos tipos de operadores borrosos que puedan complementar las búsquedas definidas, la definición de operadores borrosos sobre datos

alfanuméricicos, de fechas, objetos, etc., y la implementación de modelos basados en la metodología propuesta que se acerquen mucho más al lenguaje natural.

REFERENCIAS

- Bosc, P. y Pivert, O. *SQLF: A relational database language for fuzzy querying*. IEEE TFS, 1, (3), 1-17, 1995.
- Bosc, P. y Pivert, O. *SQLF Query Functionality on Top of a Regular Relational Database Management System*. 6th International Conference on Information Processing and Management of Uncertainty in Knowledge - Based Systems (IPMU'96), 1-6, 1996.

- Chiang, D., et al., Fuzzy Information in extended fuzzy relational Databases. *Fuzzy Sets and Systems*, 92, 1-20, 1997.
- Codd, E. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 6, (13), 1970.
- Damiani, E. et al., *A descriptor - based approach to OO code reuse*. IEEE Computer, 10, (30), 1997.
- Date, C. J. *An Introduction to Database Systems*, Reading, Mass: Addison-Wesley Publishing, Volume 1, Fifth Edition, 1990.
- Dubois, D. y Prade, H. *Possibility Theory: An approach to Computerized Processing of Uncertainty*. Primera Edición, Plenum Press, Nueva York, 1988.
- Galindo, J. et al. *FSQL (Fuzzy SQL): A Fuzzy Query Language*, 1998.
<http://www.lcc.uma.es/~ppgg/FSQL.html>.
- Grupo ARAI. No se conocen fechas de actualización. Último ingreso: Mayo 25 de 2000.
- Hale, J. y Shenoi, S. Analyzing FD Inference in Relational Databases. CORA. Artículo remitido a *Data and Knowledge Engineering Journal*, 1995.
- INTERNET WIRE. *Sonalysts releases of fuzzy query version 1.1*.
<http://www.internetwire.com/technews/tn/tn981665.htm>, 1999.
- Kraft, D. y Petry, F. Fuzzy Information Systems: managing uncertainty in databases and information retrieval systems. *Fuzzy Sets and Systems*, 90, 183-191, 1997.
- Kumar, S., et al., On extended fuzzy relational database model with proximity relations. *Fuzzy sets and Systems*, 117, 195-201, 2000.
- Li, H. y Yen, V. *Fuzzy Sets and Fuzzy Decision - Making*. CRC Press. Nueva York, 93 – 121, 1995.
- Lipschutz, S. Data Structures. Schaum's Outline Series, McGraw-Hill Book Company, Nueva York, 1986.
- Medina, J., et al., Fuzzy Knowledge Representation in relational databases.
<http://decsai.ugr.es/borroso/tre.html>, 1994.
- Medina, J. M., et al., *GEFRED: A Generalized Model of Fuzzy Relational Databases*. *Information Sciences*, 76, 87 – 109, 1994.
- ORACLE CORPORATION. *Understanding query expressions*. Oracle8(TM) ConText(R) Cartridge Application Developer's Guide, Release 2.0.
http://technet.oracle.com/doc/context206/A54630_01/ch03.htm, 1997.
- Pedrycz, W. *Fuzzy Control and Fuzzy Systems*. Research Studies Press Limited, Somerset, 1993.
- Pons, O., et al., *Inference from a fuzzy database using fuzzy rules*. Publicado en la página del grupo de investigación ARAI.
<http://decsai.ugr.es/borroso/tre.html>, 1995.
- Roux, L. y Desachy, J. *Multisources Information Fusion Application for Satellite Image Classification*. Paper incluido en el libro "Fuzzy Information Engineering - A guided tour of Applications", de Dubois, Prade y Yager. Primera edición, Wiley Computer Publishing, New York, 1997.
- SONALYSTS INC. *Fuzzy Query*.
<http://www.sonalysts.com/fq>, 2000.
- Wang, P. *SmartRanker*. 1.0.
<http://www.cogsci.indiana.edu/farg/peiwang/SmartRanker/>, 1998.
- Zadeh, L. Fuzzy Sets. *Information and Control* 8, 338-353, 1965.
- Zapata, C. M. Incorporación de operadores borrosos en Manejadores de Bases de Datos Relacionales. Tesis de Grado de Maestría en Ingeniería de Sistemas, Universidad Nacional de Colombia, 2002.