CONSULTAS CON ORDENAMIENTO BASADO EN SIMILITUD

(Queries with ordering based on similarity)

Recibido: 25/01/13 **Aceptado**: 03/05/2013

Carrasquel Oropeza, Soraya Odalis
Universidad Simón Bolívar - Venezuela
carrasquel@ldc.usb.ve

Rodríguez de Tineo, Rosseline Carmen
Universidad Simón Bolívar - Venezuela
crosseliner@gmail.com

Tineo, Leonid
Universidad Simón Bolívar - Venezuela
leonid@usb.ve

RESUMEN

Los conjuntos difusos permiten representar conceptos vagos donde la pertenencia de un objeto puede ser gradual. Se ha propuesto el uso de esta teoría en bases de de datos. Como resultado ha surgido el modelo relacional difuso y algunas extensiones al lenguaje SQL. En particular los atributos de una relación podrían tener dominios provistos de características difusas. Las implicaciones de tal tipo de atributos en operadores de consulta que se basan en el ordenamiento de datos, no han sido suficientemente exploradas. El presente trabajo se aboca a la definición de atributos en dominios provistos de una relación difusa de similitud, que es una extensión de las relaciones de equivalencia en la teoría de conjuntos difusos. Se propone una nueva definición para las relaciones difusas de similitud que resulta más adecuado al problema de bases de datos. Se extiende SQL para definir y manipular estos dominios. Se extiende la cláusula ORDER BY para consultas con ordenamiento basado en relaciones de similitud. Estas extensiones aumentan la expresividad del lenguaje de bases de datos, con respecto a propuestas anteriores.

Palabras clave: SQL, Consultas Difusas, Relaciones de Similitud, Atributos Difusos, ORDER BY.

ABSTRACT

The fuzzy sets allow represent vague concepts where membership of an object can be gradual. It has been proposed the use of this theory in databases. As result, fuzzy relational model and some extensions to the SQL language have emerged. In particular, domains with features fuzzy can supply the attributes of a relation. The implications of these attributes kind over query operators based on the ordering of data have not been sufficiently explored. This paper tackles to the definition of attributes in domains with a similarity fuzzy relation, which is an extension of equivalence relations in fuzzy set theory. We propose a new definition for similarity fuzzy relations that are more appropriate to the

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

problem of databases. We extend SQL to define and manipulate these domains. We extend the ORDER BY clause for queries with orderings based on similarity relations. These extensions increase the expressivity of the database language, with respect to previous proposals.

Keywords: SQL, Fuzzy Queries, Similarity Relation, Fuzzy Attributes, ORDER BY.

INTRODUCCIÓN

Las bases de datos tradicionales sólo manejan datos y condiciones precisos que en muchas ocasiones no representan las necesidades reales de información de los usuarios. La teoría de conjuntos difusos (Zadeh, 1965) provee un marco matemático y computacional formal para representar las nociones de naturaleza vaga o imprecisa. La incorporación de algunos de estos conceptos para el modelado y manipulación de bases de datos, dio origen a propuestas de modelo relacional difuso (Fukami et al, 1979) (Buckles y Petry, 1982). Éstas fueron luego generalizadas surgiendo un modelo extendido para bases de datos relacionales difusas, conocido como GEFRED (Medina, Pons y Vila, 1994).

Algunos esfuerzos se han realizado para dar mayor flexibilidad al lenguaje estándar de bases de datos SQL, incorporando elementos de datos y condiciones de consultas basados en los conjuntos difusos. Entre estos se destacan SQLf (Bosc y Pivert, 1995) y FSQL (Galindo et al, 2006). FSQL y SQLf son las extensiones más completas existentes para la incorporación de conjuntos difusos en SQL. Estas dos propuestas tienen enfoques complementarios: FSQL se centra en la extensión de los datos mientras que SQLf en la extensión de las expresiones de consulta (Urrutia et al, 2008). Muchos trabajos de investigación y desarrollo se han realizado a partir de estas dos propuestas.

Los conjuntos difusos se caracterizan por una función de membresía cuyo rango está en el intervalo real [0,1]. Cuánto más se acerca a 1 el grado de membresía de un elemento, éste está más posiblemente (o certeramente) incluido en el conjunto. Así 0 es la medida de completa exclusión y 1 la de completa inclusión. En bases de datos, este concepto permite dar semántica a criterios vagos (o condiciones difusas) que expresan preferencias del usuario y/o particularidades del contexto de los datos o dominio de aplicación. Otra aplicación en bases de datos es la representación y manipulación de atributos de datos imprecisos, llamados datos difusos.

Para representar datos difusos FSQL (Galindo et al, 2006) definen cuatro tipos de atributos difusos: Tipo 1, atributos con valores de datos precisos provistos con etiquetas lingüísticas, interpretadas como números difusos, con el propósito de ser usadas en condiciones difusas; Tipo 2, atributos con valores de datos difusos representados como números difusos, son distribuciones de posibilidad en un dominio ordenado; Tipo 3, atributos con valores en un dominio formado por etiquetas provisto de una relación de similitud entre las etiquetas, adicionalmente permite distribuciones de posibilidad; y Tipo 4, similar a los atributos del tipo 3, pero sin las relaciones de similitud.

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

El lenguaje estándar para bases de datos SQL provee constructores que permiten hacer consultas basadas en el ordenamiento y/o en el particionamiento de las relaciones según los valores de atributos específicos. Si se permiten atributos difusos como los propuestos en el modelo GEFRED, tales constructores de consulta deben extenderse de forma que provean una semántica adecuada en presencia de datos difusos. Sin embargo, la definición de FSQL lo que hace es prohibir que se usen estos atributos en el criterio de ordenamiento o particionamiento en una consulta, lo cual resulta poco satisfactorio. Otras propuestas conocidas de extensión a SQL con conjuntos difusos ni siquiera consideran estos tipos de atributos.

El presente trabajo se restringe al problema de atributos en cuyo dominio es un conjunto de etiquetas, dotado de una relación difusa de similitud. Las relaciones de similitud son una herramienta usada en problemas de toma de decisiones en diversas áreas como la medicina, la industria petrolera, sociales, económicas, de gestión, etc. Además, permiten modelar conceptos relacionados a la psicología, sociología, lingüística entre otras. Han sido utilizadas en estudios de espacios financieros (Lazzari et al 2008) en áreas como geobotánica son una herramienta importante para la clasificación de los datos de acuerdo con similitudes y diferencias. El objetivo de clasificar es agrupar unidades similares en tipos, formar grupos de similar entidad basados en atributos.

Como motivación, se presenta el caso donde se requiere hacer la búsqueda de Ventas de Repuestos de Automóviles. Dada la escasez de repuestos actual, muchas veces es necesario buscar no solo en la misma ciudad donde se habita, sino que hay que buscar quizás en ciudades cercanas. Al utilizar motores de búsqueda conocidos como InfoGuía.com o las Páginas Amarillas de CANTV, se obtienen 333 resultados en el primer caso y 165 resultados en el segundo caso. La búsqueda puede restringirse un poco, indicando la ciudad deseada, pero no se proveen mecanismos que permitan establecer cierta similitud entre los resultados obtenidos. Si se define un dominio para las ciudades, la relación de cercanía no es parte de la semántica asociada a este dominio. Es importante resaltar que existen otros dominios semejantes que pueden ser de interés, como son, las urbanizaciones, los estados, los países. Así también, existen otro tipo de dominios donde podría ser de interés contar con relaciones de similitud, tal es el caso de los colores, las marcas, los modelos de vehículos, las taxonomías y las ontologías.

El objetivo de la investigación aquí reportada es dar una semántica adecuada a consultas basadas en ordenamiento (cláusula ORDER BY de SQL) cuando el criterio expresado involucra atributos con valores en un dominio formado por etiquetas provisto de una relación de similitud (Tipo 3 de FSQL). Se restringe el trabajo al caso en que el atributo toma como valor exactamente una etiqueta. El caso de permitir una distribución de posibilidades requiere primero haber resuelto este caso más simple, por lo que sería tema de trabajo futuro.

El presente trabajo se organiza como se describe a continuación. La segunda sección presenta un panorama de las definiciones que se han propuesto a las relaciones de similitud, así como las características de las mismas, concluyendo con una definición que es adecuada para el contexto de las Bases de Datos. La tercera sección describe la cláusula básica de ordenamiento (ORDER BY) según la definición de los estándares SQL.

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

La cuarta sección presenta la definición de dominios de datos con relaciones de similitud y las operaciones sobre estos dominios. La quinta sección describe la extensión de la cláusula ORDER BY con relaciones de similitud. La sexta sección expone los esfuerzos que se han hecho en el área y cómo la propuesta mejora los resultados obtenidos hasta el momento. Finalmente la séptima sección presenta las Conclusiones y Trabajos Futuros de la investigación.

RELACIONES DE SIMILITUD

La teoría de conjuntos difusos fue propuesta por Zadeh (1965) como una forma de representar la imprecisión y la incertidumbre, y su motivación inicial eran las aplicaciones de sistemas de control, pero con el tiempo se comenzaron a usar en predicción y optimización, reconocimiento de patrones y sistemas expertos.

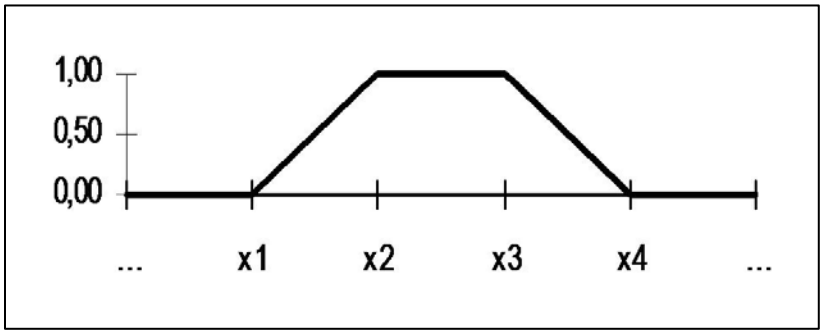
En los conjuntos clásicos la pertenencia de un elemento a un conjunto es rígida, definida por una función indicatriz cuyo rango es {0,1}, donde el 0 representa la exclusión, mientras que el 1 la inclusión. Un conjunto difuso F en un universo X admite pertenencia gradual definida por una función de membresía μ_F: $X \rightarrow [0,1]$, permitiendo no solo elementos incluidos y excluidos, sino también elementos parcialmente incluidos, aquellos cuyo grado de membresía está en el intervalo (0,1). Al conjunto formado por los elementos parcialmente incluidos se le conoce como el borde del conjunto difuso, formalmente $borde(F)=\{x \in X \mid 0<\mu_F(x)<1\}$. El conjunto de los elementos completamente incluidos se les llama el núcleo, $nucleo(F)=\{x \in X \mid \mu_F(x)=1\}$. Los elementos que no están completamente excluidos conforman el soporte, es decir, $soporte(F)=\{x \in X \mid \mu_F(x)>0\}$. Se dice que un elemento tiene pertenencia total cuando $\mu_F(x)=1$.

La función de membresía de un conjunto difuso puede definirse de distintas formas. En caso que el conjunto difuso sea definido sobre un universo numérico ordenado, la representación más sencilla y usual de la función de membresía es la forma trapezoidal (figura 1), la cual se especifica simplemente con una cuádrupla (x_1, x_2, x_3, x_4) de elementos ordenados del dominio $(x_1 \leq x_2 \leq x_3 \leq x_4)$ que definen los vértices del trapecio $\{(x_1,0), (x_2,1), (x_3,1), (x_4,0)\}$.

El μ-corte de un conjunto difuso *A* sobre el universo X, $\alpha \in [0,1]$, es el conjunto clásico A_α definido como $A_\alpha=\{x \in X \mid \mu_A(x) \leq \alpha\}$. El α-corte estricto escoge elementos con grado de membresía estrictamente mayor a α. Así A_1 llamado el 1-corte de un conjunto difuso es el núcleo y A_0 llamado el 0-corte estricto es el soporte.

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

Figura 1: Modelo de función de membresía trapezoidal



Fuente: elaboración propia

Zadeh (1971) introdujo el concepto de relación de similitud como una generalización de la equivalencia, asimismo, definió el concepto de ordenamiento difuso como la extensión difusa del orden. En este trabajo Zadeh provee un marco conceptual para las relaciones difusas, como una extensión a la teoría clásica de relaciones. Según la definición original de Zadeh (1971), una **relación de similitud** S en un universo X, es un subconjunto difuso de $X \times X$, cuya función de membresía es reflexiva $\mu_S(x,x)=1, \ \forall x \in X$, simétrica $\mu_S(x,y)=\mu_S(y,x), \ \forall x,y \in X$ y transitiva $\mu_S(x,z)=max_y(min(\mu_S(x,y), \mu_S(y,z))), \ \forall x,y,z \in X$.

Posteriormente, Jacas (1990) estudia relaciones de similitud desde el punto de vista del teorema de representación de operadores T-indistinguibles. Para T una t-norma continua, se dice que R es un operador T-indistinguible si cumple que: $\forall x,y,z \in X$ es reflexiva $R(x,x)=1$, simétrica $R(x,y)=R(y,x)$, y T-transitiva $T(R(x,y),R(y,z)) \leq R(x,z)$. Esta definición da las bases para construir algoritmos que tratan con relaciones de similitud, como por ejemplo, el problema de encontrar el generador minimal de una familia para una relación de similitud sobre un conjunto finito X.

Calvo (1992) estudia las relaciones de similitud en universos finitos, basándose en la siguiente definición: Dada una relación difusa $R: X \times X \rightarrow [0,1]$ se identifica R con su matriz asociada $A_R=(a_{ij})$ donde $a_{ij}=\mu_R(x_i,x_j) \forall i,j \in \{1,\ldots,n\}$. La reflexividad y la simetría de R tienen interpretaciones obvias en la matriz A_R. La transitividad se define considerando dos operadores binarios $(\circ,*)$ en el intervalo $[0,1]$. Estos operadores se usan para definir el $(\circ,*)$-producto de dos matrices $n \times n$, $B \circ C=(d_{ij})$, donde $d_{ij} = (b_{i1} * c_{1j}) \circ \ldots \circ (b_{in} * c_{nj})$, siendo $B=(b_{ij})$ y $C=(c_{ij})$. Una relación R es $(\circ,*)$-transitiva si $A_R \circ A_R = A_R$. Si R es reflexiva, simétrica y $(\circ,*)$-transitiva, se dice que A_R es una matriz de similitud. Usualmente el par de operadores $(\circ,*)$ es una t-norma $*$ y su co-norma $\circ$.

Ovchinnikov (1991) define las relaciones de similitud como una clase especial de las relaciones de proximidad, también conocidas como relaciones de tolerancia. Éstas son relaciones binarias difusas definidas sobre familias de conjuntos difusos, R es reflexiva, simétrica y para cada par (x,y) en la relación de proximidad R, su función de membresía es menor que la de ambos pares reflexivos $R(x,y)$ y $R(y,y)$. Según Ovchinnikov (1991), las relaciones de similitud son relaciones de proximidad transitivas, estudia la equivalencia

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

entre relaciones de proximidad y cubrimientos y el análogo difuso de relaciones de equivalencia y particiones.

Dado un conjunto finito X y A un conjunto finito de sus atributos tal que cada $\alpha \in X$ tiene al menos un atributo $p \in A$, sea $X(p)$ el conjutno de todos los $\alpha \in X$ con atributo p. Los conjuntos $X(p)$ tales que $X=\cup X(p)$ se llaman un cubrimiento de X, entonces una relación R es de proximidad cuando: aRb si existe $p \in A$ tal que $a,b \in X(p)$. una relación de similitud es una relación de proximidad transitiva, es decir: $\forall x,y,z \in X$ es reflexiva $R(x,x)=1$, simétrica $R(x,y)=R(y,x)$, transitiva $R(x,y) \wedge R(y,z) \leq R(x,z)$, donde $x \wedge y = min(x,y)$. Introduce dos clases de relación difusa de preferencia: preferencia débil y preferencia estricta. De acuerdo a su definición una relación binaria difusa R es una relación *cuasitransitiva* si R es fuertemente completa y negativamente transitiva. Esto es, si R satisface $(\forall x,y,z \in X)((R(x,y) \vee R(y,x)=1) \wedge (R(x,z) \leq R(x,y) \vee R(y,z)))$, entonces R es dual a una relación R' tal que: $(\forall x,y,z \in X)\ ((R'(x,y) \wedge R'(y,x)=0) \wedge (R'(x,y) \wedge R'(y,z) \leq R'(x,z)))$.

Faurous y Fillard (1993), reconsidera la definición de reflexividad y transitividad de las relaciones similitud para dar un nuevo soporte intuitivo a la definición de equivalencia difusa, de manera que pueda ser más provechoso para el concepto de partición difusa. En dicho trabajo se propone una nueva definición de transitividad que es consistente con la definición de partición difusa disjunta. Para ello, propone la necesidad que exista al menos un elemento con pertenencia total, es decir, con grado de membresía igual a uno. El par transitivo (x,z) de la relación aparece cuando existe entre ellos un elemento y que tenga pertenencia total. Es decir, la relación μ es reflexiva si $(\forall x \in X)(\mu(x,x)>0)$ y es transitiva si satisface las siguientes condiciones: $\{y \in X \mid \mu(y,y)=1\} \neq \varnothing$ y $(\forall x,y,z \in X)(\mu(y,y)=1 \Rightarrow \mu(x,z) \geq \mu(x,y)\ \mu(y,z))$. Esta definición es útil en el campo de procesamiento de imágenes por computador.

Belohlavek (1999), estudia similitudes en estructuras conceptuales: Similitud de Objetos y Atributos. Similitud de Conceptos; Similitud de Redes de Conceptos. El primero prueba que dicha similitud puede ser determinada por el L-contexto, lo cual es importante desde el punto de vista computacional. El segundo considera colecciones de elementos "similares" más que los elementos particulares, utilizando el proceso de abstracción por factorización donde el sistema original se considera un "sistema módulo similitudes". El tercero define el grado de similitud de dos redes de conceptos B_1, B_2 de tal modo que para cada concepto en B_1 existe un concepto en B_2 similar a éste y viceversa. Esta definición reduce el costo computacional del cálculo de las similitudes de los correspondientes contextos. Define una relación de similitud como una $\otimes$-relación binaria difusa de similitud R sobre un universo X tal que $\forall x,y,z \in X$, $R(x,x)=1$, $R(x,y)=R(y,x)$ y $R(x,y) \otimes R(y,z) \leq R(x,z)$.

En el siguiente ejemplo se aplican las diferentes definiciones de transitividad dadas anteriormente. Sea X el conjunto $X=\{Caracas, Los\ Teques, La\ Guaira\}$, las distancias entre estas ciudades están dadas en la tabla 1.

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

Tabla 1: Distancia entre ciudades cercanas

Caracas	Los Teques	32,5
Caracas	La Guaira	32,2
Los Teques	La Guaira	56,9

Fuente: LasDistancias.com

y la relación de similitud dada por la función de membresía que relaciona cada par (x,y) en el intervalo $[0,1]$: $\mu(Caracas, Los\ Teques)=0{,}82$, $\mu(Caracas, La\ Guaira)=0{,}85$ y $\mu(Los\ Teques, La\ Guaira)=0{,}56$. Como se puede observar es prácticamente la misma distancia entre Caracas-Los Teques y Caracas-La Guaira, sin embargo, la distancia entre Los Teques-La Guaira es prácticamente el doble.

Según Zadeh (1971), al aplicar la transitividad al par $(Los\ Teques, La\ Guaira)$ obtenemos $\mu(Los\ Teques, La\ Guaira)=0{,}82$. Lo cual nos sugiere que las ciudades de Los Teques y La Guaira están cerca pero eso no es cierto ya que están a una distancia de 56,9 Km.

Según Jacas (1990), usando la t-norma del mínimo obtenemos $min(\mu(Los\ Teques, Caracas), \mu(Caracas, La\ Guaira))=min(0.85,0.82)$ como $\mu(Los\ Teques, La\ Guaira)=0.56$, no se cumple que $T(\mu(x,y), \mu(y,z))\leq \mu(x,z)$. Si usamos la norma del producto entonces $\mu(Los\ Teques, Caracas), \mu(Caracas, La\ Guaira)=0.85\times0.82=0.6804$ nuevamente falla la condición.

Aplicando la transitividad dada por Calvo (1992), la matriz $A_R=(a_{ij})=R(x_i,x_j)$ es

$$A_R = \begin{vmatrix} 1 & 0{,}82 & 0{,}85 \\ 0{,}82 & 1 & 0{,}56 \\ 0{,}85 & 0{,}56 & 1 \end{vmatrix}$$

Si consideramos el par $(°, *)$ como (max, min), el producto $A_R \circ A_R$ es la matriz

$$A_R = \begin{vmatrix} 1 & 0{,}82 & 0{,}85 \\ 0{,}82 & 1 & 0{,}82 \\ 0{,}85 & 0{,}82 & 1 \end{vmatrix}$$

Se observa que es distinta de la matriz A.

Según Ochinnikov (1991), la relación R es transitiva si $\mu(Los\ Teques, Caracas) \wedge \mu(Caracas, La\ Guaira)\leq \mu(Los\ Teques, La\ Guaira)$, lo cual no es cierto. Por otra parte, se tiene que R no es fuertemente completa ya que $\mu(Los\ Teques, Caracas) \wedge \mu(Caracas, La\ Guaira)=0{,}82 \wedge 0{,}85=min(0{,}82,0{,}85) \neq 1$. En consecuencia no puede ser cuasitransitiva (tampoco es negativamente transitiva).

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

Aplicando la transitividad definida por Faurous y Fillard (1993), se tiene que la condición $\{y \in X | \mu_S(y,y)=1\}$ se cumple para todos los elementos del conjunto X y se puede observar que no se satisface que $\mu(Caracas, Caracas)=1 \Rightarrow \mu(Los\ Teques, La\ Guaira) \geq \mu(Los\ Teques, Caracas).\mu(Caracas, La\ Guaira)$.

Se ha demostrado que las definiciones de transitividad hechas por Zadeh (1971), Jacas (1990), Calvo (1992), Ovchinnikov (1991) y Faurous y Fillard (1993) no son aplicables, esto permite definir para una relación de similitud reflexiva y simétrica una nueva condición de transitividad, la cual admite que hayan tanto pares transitivos como pares no transitivos. Se da entonces la siguiente definición de Relación de Similitud.

Sea X un conjunto de atributos, R un subconjunto difuso de $X \times X$ y $\mu_R: X \times X \rightarrow [0,1]$ la función de membresía que denota el grado de pertenencia del par (x,y) al conjunto difuso R, definimos una relación de similitud difusa R_F como una relación que es Reflexiva ($\mu_R(x,x)=1$, $\forall x \in X$), Simétrica ($\mu_R(x,y)=\mu_R(y,x)$, $\forall x,y \in X$) y Transitiva ($(\forall x,z \in X,\ x \neq z\ (\forall y(\mu_R(x,y)=1 \wedge \mu_R(y,z)=\beta) \Rightarrow \mu_R(x,z)=\beta) \wedge (\forall x,z \in X,\ x \neq z\ (\forall y(\mu_R(x,y)=\beta \wedge \mu_R(y,z)=1) \Rightarrow \mu_R(x,z)=\beta))$. Observe que cuando la relación es tanto reflexiva como simétrica, basta con tomar una de las condiciones de transitividad, pues la otra se obtiene como consecuencia de estas dos propiedades. La relación de similitud induce una partición difusa sobre el conjunto de valores del atributo X, cada elemento de esta partición difusa es llamado *clase difusa* y se define como sigue. Para un elemento fijo $x \in X$, la clase difusa de x, es el conjunto de todos los valores en X que son similares al elemento x. Es decir, $c_{Rf}(x)=\{y \in X | \mu_{Rf}(x,y)>0\}$.

Ejemplo: Suponga que se tiene la distancia de las ciudades próximas a Caracas, como se muestra en la tabla 2. Es posible establecer una relación de similitud que describa que tan cercana es una ciudad a Caracas.

Tabla 2: Distancia entre Caracas y algunas ciudades próximas

Caracas	Maracay	120
Caracas	Los Teques	32.5
Caracas	Valencia	174
Caracas	Charallave	50.6
Caracas	La Guaira	32,2
Caracas	Guarenas	46

Fuente: LasDistancias.com

La relación de similitud estaría definida por el grado de membresía del par de ciudades (x,y) en el intervalo $[0,1]$, como se describe a continuación:

$Cercana=\{(Caracas,Maracay)/0.22,(Caracas, LosTeques)/0.82,$

$(Caracas, Valencia)/0.15,(Caracas,Charallave)/0.52,$

$(Caracas,LaGuaira)/0.85,(Caracas,Guarenas)/0.57\}$

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

CLAUSULA CLÁSICA DE ORDENAMIENTO EN SQL

SQL proporciona un bloque de consulta básica que tiene la estructura
SELECT [DISTINCT] <column list>
FROM <relations>
WHERE <condition>
GROUP BY <column list>
HAVING <search condition>
ORDER BY <column [<direction>] list>

La respuesta de esta consulta es un multiconjunto de filas en el producto cartesiano de las relaciones de la cláusula FROM que satisfacen la condición de la cláusula WHERE. Estas filas están formadas por las columnas de la cláusula SELECT. La palabra clave DISTINCT en la cláusula SELECT especifica que el resultado sea un conjunto en lugar de ser un multiconjunto. Si se especifica la cláusula GROUP BY, se particiona el resultado en grupos de acuerdo a las columnas especificadas. La condición de búsqueda de la cláusula HAVING se aplica a cada grupo del resultado. El resultado de la cláusula HAVING es una tabla con los grupos para los cuales la condición de búsqueda es cierta.

La cláusula ORDER BY, ordena las tuplas del resultado según el valor de una o varias columnas especificadas. El especificador <column> debe ser una columna válida dentro de las tablas especificadas en la cláusula FROM. El especificador <direction> es opcional, puede ser ASC para producir un orden ascendente o DESC para que orden sea descendente. La cláusula ORDER BY es opcional; Si ésta es especificada, entonces el resultado es ordenado. Se dice que un par <column><direction> es una especificación de orden. Cada especificación de orden debe identificar una columna válida en el producto cartesiano resultante de la cláusula FROM.

Como la cláusula ORDER BY es el objeto de esta investigación, se explicará en más detalle su funcionamiento. Cada especificación de orden puede indicar la dirección de ordenamiento para la clave de orden (columna k_i) correspondiente. Si no se especifica DESC, entonces el sentido del ordenamiento de k_i es ascendente y el operador de cómputo aplicable es el “menor que”. De lo contrario, la dirección de ordenamiento para k_i es descendente y el operador de cómputo aplicable es “mayor que”.

Sea P una fila de la tabla de resultados y sea Q cualquier otra fila de esa tabla, y sea v_{Pi} y v_{Qi} los valores de la columna k_i en estas filas, respectivamente. La posición relativa de las filas P y Q en el resultado se determina mediante la comparación de v_{Pi} y v_{Qi} de acuerdo con las reglas del predicado de comparación (“mayor que” o “menor que"), según sea el operador de cómputo aplicable para k_i. Los valores nulos tienen un tratamiento especial de acuerdo a la implementación, considerándolos menores que cualquier valor no nulo o considerándolos mayores que cualquier valor no nulo. En la tabla de resultados, la posición relativa de la fila P aparece antes que la fila Q si y sólo si el valor v_{Pi} precede al valor v_{Qi} para algún r mayor que 0 y menor que el número de especificaciones de orden y $v_{Pi} = v_{Qi}$ para todo $i<r$. El orden relativo de dos filas que no son distintas es dependiente de la implementación.

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

La formalización del resultado para la cláusula ORDER BY que es el objeto de este estudio se describe a continuación. Sea C la consulta

SELECT $c_1,c_2, \ldots,c_n$ FROM T ORDER BY $k_1\ d_1,\ldots,k_o\ d_o$

donde $k_i \in \{ c_1,c_2, \ldots,c_n \}$ y $d_i \in \{$ ASC, DESC $\}$.

Entonces, el resultado de C es la secuencia:

$$resultset\ (C) = \langle\ (t_i.c_1,\ t_i.c_2,\ \ldots,\ t_i.c_n) \mid t_i \in T \rangle\ i \in \{1,\ldots,m\}$$

El orden de las tuplas en la secuencia cumple con la restricción:

$$\forall p,q \in \{1,\ldots,m\}\ (\ \exists r \in \{1,\ldots,o\}\ (\ \rho(t_p.k_r\ ,\ t_q.k_r) \wedge \forall j \in \{1,\ldots,r\text{-}1\}\ t_p.k_j = t_q.k_j\ )\ \Rightarrow (p \leq q)\ )$$

donde

$$(d_r\text{=ASC} \Rightarrow \rho(t_p.k_r\ ,\ t_q.k_r) \equiv (t_p.k_r < t_q.k_r\ )\ ) \wedge$$

$$(d_r\text{=DESC} \Rightarrow \rho(t_p.k_r\ ,\ t_q.k_r) \equiv (t_p.k_r > t_q.k_r\ )\ )$$

Por ejemplo, si se tienen las ventas de repuestos de la tabla 3, al especificar la consulta

SELECT Nombre, Teléfono, Ciudad

FROM VentasRepuestos

ORDER BY Ciudad;

Se produce como resultado (*resultset*) la tabla 4. En ella se puede observar que el orden dado es ascendente alfabético; aunque en este resultado específico, ORDER BY no es de mucha utilidad. Las ciudades Barcelona y Barquisimeto no son cercanas, aunque están vecinas en el resultado. Algo parecido ocurre entre Barquisimeto y Caracas. La ciudad más cercana a Caracas es Los Teques, sin embargo en el resultado no son vecinas. En este caso el ORDER BY no posee una semántica que agregue valor al resultado.

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

Tabla 3: Ventas de Repuestos de Automóviles (VentasRepuestos)

Nombre	Dirección	Teléfono	Ciudad	Concesionario
Kansei Motor	Av. Andrés Bello	(0212)793.7606	Caracas	Mazda
Reggio Cars	Las Acacias	(0212)632.8325	Caracas	Todos
Autoaccesorios Goma Cars	Km 27	(0212)321.1832	Los Teques	Todos
Inversora y Promotora Don José	Simón Rodríguez	(0251)445.9421	Barquisimeto	Todos
Repuestos Douglas 2007	Centro	(0251)446.1321	Barquisimeto	Volkswagen
Repuestos Guatimotors	Calle Zamora Guatire	(0212)3445868	Guarenas	Todos
Diesel Tuy 2011	Centro	(0239)414.2100	Charallave	Todos
Direco CA	La Candelaria	(0241)853.4334	Valencia	Todos
Annarys	Calle Sucre	(0281)276.8892	Barcelona	Todos

Fuente: InfoGuía.com

Tabla 4: Resultado de la consulta con ORDER BY

Nombre	Teléfono	Ciudad
Annarys	(0281)276.8892	Barcelona
Inversora y Promotora Don José	(0251)445.9421	Barquisimeto
Repuestos Douglas 2007	(0251)446.1321	Barquisimeto
Kansei Motor	(0212)793.7606	Caracas
Reggio Cars	(0212)632.8325	Caracas
Diesel Tuy 2011	(0239)414.2100	Charallave
Repuestos Guatimotors	(0212)344.5868	Guarenas
Autoaccesorios Goma Cars	(0212)321.1832	Los Teques
Direco CA	(0241)853.4334	Valencia

Fuente: InfoGuía.com

DOMINIOS DE DATOS CON RELACIONES DE SIMILITUD

SQL permite la definición de dominios para describir nuevos tipos de datos dentro de un esquema. En este trabajo se extiende el SQL-DDL (SQL *Data Definition Language*) con la posibilidad de crear dominios de datos difusos que tengan asociada una relación de similitud. La sintaxis de definición de un dominio difuso de datos sería

CREATE FUZZY DOMAIN <name> AS
VALUES (<label >[,<label>,…,<label>])
[SIMILARITY { (<label>,<label>) / <value>
[,(<label>,<label>) / <value>,…, (<label>,<label>) / <value>] }]

Donde, <name> es el nombre del nuevo dominio; (<label >[,<label>,…,<label>]) es la lista de etiquetas que definen el dominio; los especificadores (<label>,<label>)/<value> corresponden a los pares de la relación difusa de similitud para ese dominio; siendo <value> el valor del grado de membresía de dicho par en la relación. Solo es necesario

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

especificar los pares básicos de la relación, pues los correspondientes a la reflexividad, simetría y transitividad están sobreentendidos. La relación de similitud es opcional.

La formalización de esta sentencia usando la teoría de conjuntos difusos sería

CREATE FUZZY DOMAIN *fd* AS
VALUES ($l_1,l_2,\ldots,l_k$)
[SIMILARITY { (l_{i1}, l_{j1}) / v_1, (l_{i2}, l_{j2}) / $v_2,\ldots,$ (l_{in}, l_{jn}) / v_n }]

La cual define un universo $fd=\{l_1,l_2,\ldots,l_k\}$ provisto de una relación de similitud s. Adicionalmente especifica una relación difusa σ definida por $\forall r \in 1..n\ \mu_\sigma(l_{ir}, l_{jr}) = v_r$ a la cual llamaremos relación base de la relación de similitud s.

La relación de similitud s definida por esta sentencia satisface

$\forall x,y \in fd\ \mu_\sigma(x,y) \neq 0 \Rightarrow \mu_s(x,y) = \mu_\sigma(x,y)$

$\forall l \in fd\ \mu_s(l,l) = 1$ (reflexividad)

$\forall x,y \in fd\ \mu_s(x,y) = \mu_s(y,x)$ (simetría)

$\forall x,z \in fd\ x \neq z\ \forall y \in fd\ (\ \mu_s(x,y) = 1 \wedge \mu_s(y,z) = v \Rightarrow \mu_s(x,z) = v\ )$ (transitividad)

El grado de membresía a la relación de similitud s es cero para cualquier par de valores no especificado en la cláusula, el cual no se pueda obtener por aplicación de las reglas de reflexividad, simetría y transitividad.

Por ejemplo, para la relación *Cercana* mencionada anteriormente, se crea el dominio de las ciudades, el cual tiene asociado una relación de similitud, como se describe con la sentencia

CREATE FUZZY DOMAIN Ciudades AS
VALUES (Caracas,Maracay,LosTeques,Valencia,Charallave,LaGuaira,Guarenas)
SIMILARITY { (Caracas, Maracay)/0.22, (Caracas, Los Teques)/0.81,
(Caracas, Valencia)/0.15, (Caracas, Charallave)/0.52,
(Caracas, La Guaira)/0.85, (Caracas, Guarenas)/0.57,
(Maracay,LosTeques)/0.35, (Maracay,Valencia)/0.20,
(Maracay, Charallave)/0.20, (LosTeques,Valencia)/0.18,
(LosTeques,Charallave)/0.65, (LosTeques, LaGuaira)/0.13,
(LosTeques,Guarenas)/0.14, (Valencia,Charallave)/0.14,
(Charallave,LaGuaira)/0.12, (Charallave,Guarenas)/0.10,
(LaGuaira,Guarenas)/0.12}

De acuerdo a la semántica expresada anteriormente la relación de similitud obtenida sería la que se muestra en la tabla 5.

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

Tabla 5: Relación de Similitud para el dominio Ciudades

μ_s	Caracas	Maracay	LosTeques	Valencia	Charallave	LaGuaira	Guarenas
Caracas	1	0.22	0.81	0.15	0.52	0.85	0.57
Maracay	0.22	1	0.35	0.48	0.20	0	0
LosTeques	0.81	0.35	1	0.18	0.65	0.13	0.14
Valencia	0.15	0.48	0.18	1	0.14	0	0
Charallave	0.52	0.20	0.65	0.14	1	0.12	0.10
LaGuaira	0.85	0	0.13	0	0.12	1	0.12
Guarenas	0.57	0	0.14	0	0.10	0.12	1

Fuente: elaboración propia

También se agrega la posibilidad de añadir nuevos valores al dominio difuso, usando la siguiente sintaxis.

ALTER FUZZY DOMAIN <name> ADD VALUES (<label>[,<label>,…,<label>])

Usando el dominio creado *fd* previamente, la formalización de esta sentencia se expresaría como ALTER FUZZY DOMAIN *fd* ADD VALUES (e_1,e_2,…,e_m) con $e_i \neq l_j \; \forall i \neq j$; cuya semántica viene dada por $fd=\{l_1,l_2,\ldots,l_k\} \cup \{e_1,e_2,\ldots,e_m\}$. Por ejemplo, si se quiere agregar nuevas ciudades al dominio, tales como La Victoria y Cúa, se usaría la siguiente sentencia

ALTER FUZZY DOMAIN Ciudades ADD VALUES (LaVictoria, Cua)

Asimismo, la posibilidad de agregar nuevos pares a la relación difusa de similitud, usando la sintaxis:

ALTER FUZZY DOMAIN <name> ADD SIMILARITY {(<label>,<label>) / <value>) [,(<label>,<label>) / <value>),…, (<label>,<label>) / <value>)]}

La formalización para esta sentencia estaría dada por ALTER FUZZY DOMAIN *fd* ADD SIMILARITY { (e_{i1}, e_{j1}) / w_1, (e_{i2}, e_{j2}) / w_2,…, (e_{in}, e_{jp}) / w_p } los cuales cumplen $\forall i,j \; e_{ij} \in fd$ y agrega a la relación base σ definida para el dominio *fd* los grados de membresía especificados por $\forall r \in 1..p \; \mu_\sigma(e_{ir}, e_{jr}) = w_r$.

Por ejemplo, para agregar algunos nuevos pares a la relación de similitud que indica la cercanía entre ciudades, se utilizaría la sentencia

ALTER FUZZY DOMAIN Ciudades ADD SIMILARITY {(Caracas, LaVictoria)/0.42, (Caracas, Cua)/0.45}

Además, se permite eliminar etiquetas al dominio de datos difusos, o pares de la relación de similitud, con la siguiente sintaxis:

ALTER FUZZY DOMAIN <name> DROP VALUES (<label>[,<label>,…,<label>])

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

ALTER FUZZY DOMAIN <name> DROP SIMILARITY {(<label>,<label>) [,(<label>,<label>),…, (<label>,<label>)] }

En este caso, la formalización de la primera sentencia estaría dada por ALTER FUZZY DOMAIN *fd* DROP VALUES (e_1,e_2,…,e_m) con $\forall i \exists j (e_i = l_j)$ y $m<k$; cuya semántica se expresa como $fd=\{l_1,l_2,\ldots,l_k\} - \{e_1,e_2,\ldots,e_m\}$. La segunda sentencia se formaliza como ALTER FUZZY DOMAIN *fd* DROP SIMILARITY {(e_{i1}, e_{j1}), (e_{i2}, e_{j2}),…, (e_{ip}, e_{jp}) }, altera la definición de la relación base σ haciendo $\forall r \in 1..p\ (\mu_\sigma(e_{ir}, e_{jr}) = 0)$.

Por ejemplo, si se quiere eliminar la ciudad Guarenas del dominio ciudades, junto con todos sus pares correspondientes a la relación de similitud, se utilizaría la sentencia

ALTER FUZZY DOMAIN Ciudades DROP VALUES (Guarenas)

Si sólo se quiere eliminar el par Caracas-Guarenas de la relación de similitud para el dominio Ciudades, se utilizaría la sentencia

ALTER FUZZY DOMAIN Ciudades DROP SIMILARITY {(Caracas, Guarenas)}

Finalmente, se incluye la posibilidad de borrar un dominio de datos difuso, lo cual incluye su relación de similitud si ésta existe. Para ello se usa la siguiente sintaxis:

DROP FUZZY DOMAIN <name>

ORDENAMIENTO BASADO EN RELACIONES DE SIMILITUD

Dada la definición clásica del funcionamiento del ORDER BY y la extensión de difusa de los dominios de datos, las relaciones de similitud difusas asociadas a estos dominios, permiten dar una nueva semántica a la cláusula ORDER BY. Como la relación de similitud se caracteriza por ser reflexiva, simétrica y transitiva, esta relación genera una partición difusa sobre el conjunto de valores de un atributo. Cada posible valor tiene asociado una clase difusa de todos los valores similares a éste.

Por ejemplo, en la relación *Cercana* que se ha venido tratando las ciudades similares a Caracas, son aquellas cuyo par tiene un grado de membresía mayor a cero dentro de la relación de similitud. Este grado se observa en la tabla 3. Es decir, Maracay, Los Teques, Valencia, Charallave, La Guaira y Guarenas. Estas ciudades forman una clase difusa junto con Caracas. Las ciudades que no tienen asociado un par con Caracas en la relación de similitud, dicho par tiene por defecto grado de membresía cero, por lo que no pertenecen a la clase difusa de Caracas.

Las clases difusas, permiten extender la cláusula ORDER BY con una nueva semántica cuyo fin sea ordenar los elementos de acuerdo al grado de membresía, en la relación de similitud, con respecto a un elemento fijado. La sintaxis de la cláusula ORDER BY extendida sería

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

ORDER BY $criterio_1,...,criterio_o$

donde cada $criterio_1$ es de alguna de las formas posibles:

k_i d_i siendo k_i un atributo y d_i un especificador de orden ASC o DESC

k_i START v_i siendo k_i un atributo difuso tipo 3, v_i una etiqueta en el dominio de ese atributo

puede también usarse simplemente k_i como forma abreviada sinómino de k_i ASC

para la forma k_i START v_i se proveen adicionalmente las formas verbosas

k_i STARTING FROM v_i
SIMILARITY ON k_i START v_i
SIMILARITY ON k_i STARTING FROM v_i

El uso de un criterio de ordenamiento de la forma k_i START v_i cláusula produce que el conjunto de tuplas resultado esté restringido a las tuplas cuyo valor para k_i está en la clase difusa asociada a v_i. Además, las tuplas serán ordenadas descendentemente por el grado de membresía μ_s asociado a la relación de similitud s del dominio correspondiente a v_i.

La formalización de este resultado para la nueva semántica de la cláusula ORDER BY se describe a continuación. Sea C la consulta

SELECT $c_1,c_2, \ldots,c_n$ FROM T ORDER BY $k_1\ d_1,\ldots,k_o\ d_o$

donde $k_i \in \{ c_1,c_2, \ldots,c_n \}$ y $d_i \in \{$ ASC, DESC , START v $\}$.

Entonces, el resultado de C es la secuencia:

$$resultset\ (C) = \langle\ (t_i.c_1,\ t_i.c_2,\ \ldots,\ t_i.c_n) \mid t_i \in T \rangle\ i \in \{1,\ldots,m\}$$

El orden de las tuplas en la secuencia cumple con la restricción:

$$\forall p,q \in \{1,\ldots,m\}\ (\ \exists r \in \{1,\ldots,o\}\ (\ \rho(t_p.k_r\ ,\ t_q.k_r) \wedge \forall j \in \{1,\ldots,r\text{-}1\}\ t_p.k_j = t_q.k_j\ )$$

$$\Rightarrow (p \leq q)$$

)

donde

$$(d_r\text{=ASC} \Rightarrow \rho(t_p.k_r\ ,\ t_q.k_r) \equiv (t_p.k_r < t_q.k_r\ )\ ) \wedge$$

$$(d_r\text{=DESC} \Rightarrow \rho(t_p.k_r\ ,\ t_q.k_r) \equiv (t_p.k_r > t_q.k_r\ )\ ) \wedge$$

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

(d_r= START $v \Rightarrow \rho(t_p.k_r , t_q.k_r) \equiv (\mu_s(v, t_p.k_r) < \mu_s(v, t_q.k_r)$)

Como ilustración del uso de esta cláusula, para las Ventas de Repuestos de Automóviles mostradas en la tabla 3, la consulta

SELECT Nombre, Teléfono, Ciudad

FROM VentasRepuestos

ORDER BY SIMILARITY ON Ciudad STARTING FROM 'Caracas';

Produce como resultado las tuplas de la tabla 6, a las cuales se le ha agregado el valor μ. Éste corresponde al grado de membresía a la relación de similitud del par formado por el valor del campo Ciudad junto con Caracas. En este resultado se observa que las ventas en las ciudades de Barcelona y Barquisimeto no aparecen pues el grado de membresía (μ) de los pares (Caracas, Barcelona) y (Caracas, Barquisimeto) es cero. Además, el resultado aparece ordenado por el grado de membresía (μ) de la relación de similitud, por lo que las ciudades más cercanas a Caracas aparecen primero. Esta respuesta es mucho más significativa para el usuario, que aquella mostrada en la tabla 4.

Tabla 6: Resultado de la consulta con ORDER BY SIMILARITY, especificando el valor de μ

Nombre	Teléfono	Ciudad	μ
Kansei Motor	(0212)793.7606	Caracas	1
Reggio Cars	(0212)632.8325	Caracas	1
Autoaccesorios Goma Cars	(0212)321.1832	Los Teques	0.81
Repuestos Guatimotors	(0212)344.5868	Guarenas	0.57
Diesel Tuy 2011	(0239)414.2100	Charallave	0.52
Direco CA	(0241)853.4334	Valencia	0.15

Fuente: InfoGuía.com

TRABAJOS RELACIONADOS

La teoría de conjuntos difusos (Zadeh 1965) permite dar un tratamiento matemático-computacional a conceptos vagos del mundo real. Por ello, algunos autores han propuesto utilizar esta teoría para el manejo de bases de datos que incorporen información o requerimientos imprecisos. Entre estas propuestas se destacan SQLf (Bosc y Pivert 1995) y FSQL (Galindo et al, 2006). Ambas son extensiones del estándar SQL que incorporan conceptos provenientes de la teoría de conjuntos difusos.

El lenguaje difuso para bases de datos SQLf (Bosc y Pivert 1995) fue concebido para resolver el problema de la rigidez de las consultas clásicas en base de datos. Para ello, SQLf incorpora términos lingüísticos vagos cuya semántica es definida mediante conjuntos difusos. Estos términos difusos se clasifican en: predicados, comparadores, modificadores, conectores y cuantificadores. La creación de tales términos como objetos de la base de datos fue propuesta por Tineo (1998) (REF). Con ellos se pueden expresar condiciones en lógica difusa. SQLf permite el uso de este tipo de condiciones en cualquier

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

lugar en que SQL clásico admite una condición en lógica booleana. Más recientemente, González et al (2009) han actualizado SQLf para las operaciones provistas por el estándar SQL:2003.

Los atributos de datos con que trabaja SQLf son siempre precisos. De hecho, las primeras versiones de SQLf trabajaban sólo con bases de datos relacionales, produciendo como resultado relaciones difusas. Esto es, tablas en las cuales cada fila es dotada de un grado de membresía. Éste grado corresponde al grado en que la tupla satisface la consulta difusa. La versión de González et al (2009) permite que sean almacenadas relaciones difusas. Sin embargo, los atributos siguen siendo definidos en dominios precisos, sin características provenientes de los conjuntos difusos.

El concepto de relación difusa de similitud no ha sido incorporado de manera explícita e intencional en SQLf. No obstante, los comparadores difusos de SQLf son definidos como relaciones difusas binarias. De manera que, una relación de similitud puede ser definida como un comparador difuso. El asunto es que, como el concepto no es soportado, no se hacen las verificaciones ni las derivaciones correspondientes a las propiedades de reflexividad, simetría y transitividad. De manera que sería más trabajo para el programador o usuario.

En SQLf tendría que definir el dominio del conjunto de etiquetas, que no sería difuso. Luego definir aparte el comparador que fungirá como relación de similitud difusa. La especificación será mucho más extensa. Si el dominio está compuesto de n valores, habría que incluir los n pares reflexivos con grado 1.0. Si la relación base es de 1.0 pares con sus grados, se debe añadir también los m pares simétricos. En el mejor de los casos no hay pares no reflexivos que tengan grado 1.0. De no ser así, hay que considerar también los pares que se añaden por transitividad. En el peor de los casos los m pares de la relación base tienen grado 1.0 y forman una secuencia de longitud n-1 de forma $\langle(l_{i0}, l_{i1})$, $(l_{i1}, l_{i1}),\ldots, (l_{in-2}, l_{in-1})\rangle$ $\forall j,k\ 0\leq j<k\leq n\text{-}1$ $(l_{ij}\neq l_{ik})$. En ese caso, el número de pares que se infieren por transitividad son n^2-2n -1. En general, esto nos lleva a que en SQL habría que especificar al menos n y a lo más n^2-(n-1) pares más de los que se tendrían que especificar con la definición de dominio difuso aquí propuesta.

A pesar que SQLf provee una gran variedad de expresiones consultas, no tiene una que sea equivalente a la consulta ordenada por el grado de similitud que se propone en este artículo. Si se define un comparador para simular la relación de similitud difusa, éste comparador puede usarse en una condición en la cláusula WHERE, pero no puede usarse en la cláusula ORDER BY. Por definición de SQLf, en la respuesta a una consulta, las tuplas son ordenadas en forma decreciente, de acuerdo a su grado de membresía a relación difusa especificada por la expresión de consulta.

De manera que una consulta de la forma.

SELECT $c_1,c_2, \ldots,c_n$ FROM T ORDER BY q STARTING FROM v

Puede simularse en SQLf con la consulta

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

SELECT $c_1,c_2, \ldots,c_n$ FROM *T* WHERE *q comp v*

siendo *comp* el comparador difuso en SQLf para la relación difusa de similitud

Pero el tipo de consulta propuesto en este artículo va más allá; el uso de la relación de similitud puede no ser el único criterio de ordenamiento. Si se tiene una consulta con una cláusula ORDER BY que involucre al menos dos expresiones de ordenamiento, donde una de ellas tiene una especificación basada en una relación de similitud difusa, esta consulta no tiene un equivalente en SQLf.

Por otro lado, el lenguaje de bases de datos relacionales difusas FSQL (Galindo et al, 2006) fue concebido para dar un tratamiento a datos y consultas difusas. Este lenguaje contempla cuatro tipos de datos difusos, uno de los cuales (llamado tipo 3) está concebido para el manejo de atributos cuyos valores son etiquetas sobre las cuales se define una relación de similitud.

A pesar que la definición de FSQL considera este tipo de datos y relaciones, hay una deficiencia en el concepto que se maneja para la relación de similitud y en su forma de manipularla. En principio, es una relación reflexiva donde cada etiqueta es completamente similar a sí misma (grado 1). Pero la simetría es opcional, lo cual no parece estar en correspondencia con la semántica de relación difusa de similitud, de acuerdo a lo que han propuesto los distintos autores.

Finalmente, la transitividad es una propiedad que ni siquiera es nombrada en la definición de FSQL, por lo que podríamos tener casos en que la relación solo tenga pares con grado 1 (clásico), sea reflexiva y simétrica, mas no transitiva. Esto, estaría violando la idea de ser la extensión en conjuntos para lo que es una relación de equivalencia en conjuntos clásicos.

En FSQL tendría que definir primero el nombre del tipo de dato, diciendo que es difuso y que es de tipo 3. Esto se hace mediante una sentencia de la forma: CREATE FDATATYPE <nombre> AS FTYPE3. A pesar que se está diciendo que es de tipo 3, no se están especificando las etiquetas ni los grados de membresía de la relación que define la similitud. Esto se hace en otra instrucción diferente: CREATE NEARNESS ON <nombre> LABEL $l_1, l_2,\ldots, l_n$ VALUES $d_1, d_2,\ldots, d_m$.

En caso de no ser simétrica la relación, la longitud m de esta lista es n^2 y la secuencia $\langle d_1, d_2,\ldots, d_m\rangle$ corresponde a $\langle \mu_s(l_1,l_1), \mu_s(l_1,l_2),\ldots, \mu_s(l_1,l_n), \mu_s(l_2,l_1), \mu_s(l_2,l_2),\ldots, \mu_s(l_2,l_n),\ldots, \mu_s(l_n,l_1), \mu_s(l_n,l_2),\ldots, \mu_s(l_n,l_n)\rangle$ siendo s la relación que se está definiendo. En caso de ser simétrica la relación, m es $(n^2-n)/2$ y la secuencia $\langle d_1,d_2,\ldots, d_m\rangle$ corresponde a $\langle \mu_s(l_1,l_2),\ldots, \mu_s(l_1,l_n)$, $\mu_s(l_2,l_3),\ldots, \mu_s(l_{n-1},l_n)\rangle$ siendo s la relación que se está definiendo. Nótese que en el mejor de los casos, el programador o usuario tiene que especificar $(n^2-n)/2$ y en el peor caso n^2, sin la posibilidad de casos intermedios, con redundancia y sin transitividad. FSQL provee una sentencia ALTER para estas relaciones que no permite en una sola instrucción cambiar más de un par, además este par podría colocarse con un valor no correspondiente al par simétrico, aunque la definición original de la relación haya sido simétrica.

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

A pesar que FSQL permite la creación de tipos datos difusos y usarlos para definir atributos de las tablas, en la definición de este lenguaje se considera un error el usar un atributo definido como de tipo 3 en una cláusula ORDER BY. Sólo se permite que se coloque bajo una función de conversión TO_CHAR, considerándose el orden alfabético. Este tipo de excepciones atenta contra la ortogonalidad del lenguaje. Por otro lado, el ordenamiento alfabético parece no ser el más conveniente. Sin embargo, podría usarse la relación de similitud en forma implícita en una cláusula WHERE, usando el comparador FEQ (igualdad difusa). En ese caso, es posible adicionalmente usar la función CDEG (grado de compatibilidad) y ordenar por el resultado de esta función. A juicio de los autores, esta forma de expresión resulta un poco engorrosa.

De manera que una consulta de la forma.

SELECT $c_1,c_2, \ldots,c_n$ FROM *T* ORDER BY *q* STARTING FROM *v*

Puede simularse en FSQL con la consulta

SELECT CDEG(*q*),$c_1,c_2, \ldots,c_n$ FROM *T* WHERE *q* FEQ *v* ORDER BY CDEG(*q*) DESC

A diferencia de lo que ocurre en SQLf, con FSQL sí se podría emular el caso en que se tiene una consulta con una cláusula ORDER BY que involucre varias expresiones de ordenamiento, donde una o más de ellas tiene una especificación basada en una relación de similitud difusa. Esto se haría con un CDEG para cada uno de los atributos que se quiere usar para el ordenamiento y así como su respectiva condición FEQ en la cláusula WHERE. Sin embargo, estos no funcionarán si la consulta involucrara otras condiciones difusas sobre alguno de estos atributos.

Se ha visto las posibilidades y dificultades que tendríamos con SQLf y FSQL para definir dominios de datos dotados de relaciones difusas de similitud y su uso para las consultas con ordenamiento basado en estas relaciones. Existen otras propuestas para la incorporación de conceptos provenientes de la teoría de conjuntos difusos en SQL, tales como: OMRON (Nakajima et al 1983), FQUERY (Kacpryzyk y Zadrożny, 1995), ISKREOT (Loo y Lee, 2000), PFSQL (Takači y Škrbić, 2008), SOFTSQL (Bordogna y Psaila, 2008), entre otras (Zadrożny et al 2008). Ellas no fueron consideradas en esta sección, debido a que no proveen una manera que permita especificar relaciones difusas de similitud. Adicionalmente SQLf y FSQL son propuestas que han tenido mayor impacto.

CONCLUSIONES Y TRABAJOS FUTUROS

El modelo de bases de datos relacional difuso extendido incorpora, entre otras bondades, la posibilidad de tener atributos cuyos valores pertenezcan a dominios con características tomadas de la teoría de conjuntos difusos. En particular, algunos autores han propuesto permitir atributos cuyo dominio sea un conjunto de etiquetas que esté provisto de una relación de similitud. Sin embargo, estos trabajos previos no han profundizado en cuanto a aspectos sintácticos y semánticos para la incorporación de tales atributos al lenguaje SQL.

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

Un primer aporte del trabajo reportado es un estudio de algunas propuestas existentes para el concepto de relación difusa de similitud, analizando su posible aplicación para la definición de dominios de atributos de bases de datos. En general, dado que una relación difusa de similitud es la extensión difusa de las relaciones de equivalencia clásica, las propuestas existentes consideran una extensión de las propiedades de reflexividad, simetría y transitividad.

Las definiciones que se han propuesto previamente para la transitividad no resultan satisfactorias en el estudio presentado, por lo que se ofrece una nueva propuesta de concepto de relación difusa de similitud. Aquí la reflexividad es definida así: cualquier valor x en el dominio es completamente similar a sí mismo (grado 1). La simetría es trivial: el grado de similitud de un valor x con otro valor y es exactamente el mismo de este otro y con x. La transitividad es definida así: sean x,z diferentes, sea y cualquiera, si grado de similitud de x con y es 1 y el grado de similitud de x con z es el mismo que aquél de y con z, asimismo si grado de similitud de y con z es 1 y el grado de similitud de x con z es el mismo que aquél de x con y.

Dado que hay un interés en permitir atributos cuyo dominio sea un conjunto de etiquetas que esté provisto de una relación difusa de similitud, en este trabajo se ha extendido el sistema de objetos y tipos de SQL, de manera que se puedan especificar y usar tales dominios. Para ello se ha definido un nuevo tipo de objetos del catálogo de una base de datos que es el FUZZY DOMAIN, con sus respectivas operaciones de creación (CREATE), modificación (ALTER) y destrucción (DROP).

Al dominio se le asocia un conjunto de valores (cláusula VALUES) y una relación difusa de similitud (SIMILARITY). En la especificación de tal relación, pueden omitirse los grados que se obtienen por reflexividad, simetría o transitividad. En caso de incluirse alguno de ellos, el gestor de bases de datos debe hacer la verificación, si no, debe hacer la inferencia. Los pares que no se especifican ni se obtienen por reflexividad, simetría o transitividad, están completamente excluidos de la relación (grado 0). Como cualquier objeto de una base de datos, a un FUZZY DOMAIN se le asocia un identificador. Con este nombre pueden definirse atributos del tipo especificado.

El tener atributos cuyos valores sean etiquetas en un dominio con una relación difusa de similitud tiene consecuencias en algunas de los operadores de consultas de SQL, las cuales deben ser definidas. Particularmente, una relación difusa de similitud induce relaciones de orden parcial en el universo en que está definida. Estas relaciones de orden parcial pueden ser usadas para extender la semántica de la cláusula ORDER BY de SQL. En este trabajo se propone que si en la cláusula ORDER BY de una consulta, se usa un atributo definido sobre un FUZZY DOMAIN dotado de una relación difusa de similitud, mediante una nueva la frase clave STARTING FROM, se puede especificar un valor del dominio como el inicial del ordenamiento. Los resultados se presentan entonces en orden decreciente del grado de similitud del valor de atributo en cuestión respecto al valor de inicio especificado. En este caso se estaría usando un orden parcial inducido por la relación difusa de similitud.

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

La noción de ordenamiento de filas de una tabla en SQL es también usada para dar semántica a lo que se conoce como ventanas o, en inglés, *windowed tables*. De manera que en un trabajo futuro, habría que estudiar las implicaciones de los atributos provistos de relaciones de similitud en las operaciones de SQL sobre ventanas. También SQL provee el concepto consulta sobre tablas particionadas, a través de la cláusula GRUOP BY.

Dado que una relación difusa de similitud induce una partición difusa, es necesario extender la semántica de este tipo de consultas. Esto será tema de trabajos futuros. De acuerdo al modelo relacional difuso para bases de datos, también es posible tener atributos de valores imprecisos definidos mediante distribuciones de posibilidad. Al momento no se conocen trabajos en que se haya considerado el tema del ordenamiento de este tipo de valores difusos en el marco de las bases de datos. Tampoco sus implicaciones en consultas basadas en ventanas o consultas particionadas. Esto es un tema abierto a la investigación.

Los aportes presentados en este artículo enriquecen la expresividad y semántica del lenguaje de definición y consulta a bases de datos relacionales SQL. Sería conveniente hacer análisis de desempeño de la extensión presentada, y en caso de ser necesario, proponer algoritmos y estructuras adecuados para garantizar una óptima prestación de servicios. Aquí hay trabajo por hacer en desarrollo y experimentación, el cual se reportaría cuando se tengan resultados.

AGRADECIMIENTOS

Damos gracias a Aquél que nos ayudó a lograrlo: “Y si alguno prevaleciere contra uno, dos le resistirán; y cordón de tres dobleces no se rompe pronto” Eclesiastés 4:12 (Reina-Valera 1960)

REFERENCIAS BIBLIOGRÁFICAS

Belohlavek, R. (1999). Similarity Relations in Concept Lattices. Research Report No. 20. University of Ostrava, Czech Republic.

Bordogna, G. y Psaila, G. (2008). Customizable Flexible Querying for Classical Relational Databases. In: Galindo, J. (Ed.), Handbook of Research on Fuzzy Information Processing in Databases, Vol. 1, Pp. 191-217.

Bosc, P. y Pivert, O. (1995). SQLF: a relational database language for fuzzy quering. IEEE transactions on fuzzy systems, Vol. 3, No. 1. Pp. 1-17

Buckles, B. y Petry, F. (1982). A fuzzy representation of data for relational databases. Fuzzy Sets and Systems, Vol. 7, No. 3, Pp. 213-226.

Calvo, T. (1992). On fuzzy similarity relations. Fuzzy Sets and Systems. Vol. 47, No. 1 Pp. 121-123.

Faurous, P. y Fillard, J. (1993). A New Approach to the Similarity Relations in the Fuzzy Set Theory. Information Sciences, Vol. 75, No. 3, Pp. 213-221.

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013

Fukami, S.; Umano, M.; Muzimoto, M. y Tanaka, H. (1979). Fuzzy database retrieval and manipulation language (Tech. rep. No. AL-78-85). IEICE Technical Reports, 78(233), Pp. 65-72.

Galindo, J.; Urrutia A. y Piattini, M. (2006). Fuzzy Databases: Modeling, Design and Implementation. USA. Idea Group Publishing Hershey.

González, C.; Goncalves, M. y Tineo, L. (2009). A New Upgrade to SQLf: Towards an Standard in Fuzzy Databases. 20th International Workshop on Database and Expert Systems Application Proceeding. Austria. Pp. 442-446.

Jacas, J. (1990). Similarity Relations - The calculation of minimal generating families. Fuzzy sets and systems. Vol. 35, No. 2, Pp. 151-162.

Kacpryzyk, J. y Zadrożny, S. (1995). Fuzzy Queries in Microsoft Access v.2. Proceedings of Workshop on Fuzzy Database Systems and Inf. Retrieval, Pp. 61-66.

Lazzari, L; Mouliá, P y Eriz M. (2008). Crisp and Fuzzy Relations. Application to Financial Space. Cuadernos del Cimbage, N° 10, Pp. 17-46.

Loo, G. y Lee, K. (2000). An Interface to Databases for Flexible Query Answering: A Fuzzy-Set Approach. LNCS. Vol. 1873, Pp. 654-663.

Medina, J.; Pons, O. y Vila, A. (1994). GEFRED: A Generalized Model of Fuzzy Relational Databases, Information Sciences, Vol. 77, No. 6, Pp. 87-109.

Nakajima, H.; Sogoh, T. y Arao, M. (1983) Fuzzy Database Language and Library-Fuzzy Extension to SQL. Proceedings of 2nd Fuzz-IEEE, Vol. 1. Pp. 477-482.

Ovchinnikov, S. (1991). Similarity relations, fuzzy partitions, and fuzzy orderings. Fuzzy Sets and Systems, Vol. 40, No. 1, Pp. 107-126

Takači, A. y Škrbić, S. (2008) Data Model of FRDB with Different Data Types and PFSQL. In: Galindo, J. (Ed.), Handbook of Research on Fuzzy Information Processing in Databases. USA. Information Science Reference.

Urrutia, A.; Tineo, L. y González, C. (2008). FSQL and SQLf: Towards a Standard in Fuzzy Databases. In: Galindo, J. (Ed.), Handbook of Research on Fuzzy Information Processing in Databases, Vol. I, Pp. 270-298.

Zadeh, L. (1965). Fuzzy Sets. Information Control, Vol. 8, No. 3, Pp. 338-353.

Zadeh, L. (1971). Similarity Relations and Fuzzy Orderings. Information Sciences, Vol. 3, No. 2, Pp. 177-200.

Zadrożny, S.; De Tré, G.; De Caluwe, R. y Kacprzyk, J. (2008). An overview of fuzzy approaches to flexible database querying. In: Galindo, J. (Ed.), Handbook of Research on Fuzzy Information Processing in Databases. USA: Information Science Reference.

Depósito Legal: PPX200002ZU2142 / ISSN 1856-4194. Volumen 12 Edición No 1 – Enero / Junio 2013