



Ingeniare. Revista Chilena de Ingeniería

ISSN: 0718-3291

facing@uta.cl

Universidad de Tarapacá

Chile

Carrasquel, Soraya; Rodríguez, Rosseline; Tineo, Leonid

Consultas con agrupamiento basado en similitud

Ingeniare. Revista Chilena de Ingeniería, vol. 22, núm. 4, octubre, 2014, pp. 517-527

Universidad de Tarapacá

Arica, Chile

Disponible en: <http://www.redalyc.org/articulo.oa?id=77231865007>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Consultas con agrupamiento basado en similitud

Queries with grouping based on similarity

Soraya Carrasquel¹ Rosseline Rodríguez¹ Leonid Tineo¹

Recibido 19 de agosto de 2013, aceptado 8 de mayo de 2014

Received: August 19, 2013 Accepted: May 8, 2014

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 datos. Como resultado ha surgido el modelo relacional difuso GEFRED, el modelo entidad relación extendido difuso FuzzyEER y lenguaje de consultas difusas FSQ, entre otros. Estas extensiones incluyen algunas formas de atributos difusos, como los llamados Tipo 3. Estos atributos toman valores de un conjunto de etiquetas provisto de una relación difusa de similitud, que es una extensión de las relaciones de equivalencia en la teoría de conjuntos difusos. Las implicaciones de tal tipo de atributos en operadores de consulta particionadas no han sido previamente exploradas. La contribución del presente artículo consiste en la definición sintáctica y semántica de una extensión a SQL para permitir consultas particionadas difusas basadas en atributos Tipo 3. Esta extensión aumenta la expresividad del lenguaje de bases de datos, respecto de propuestas anteriores.

Palabras clave: SQL, consultas difusas, relaciones de similitud, atributos difusos, GROUP BY.

ABSTRACT

The fuzzy sets allow for the representation of vague concepts, where membership of an object could be gradual. The use of this theory in the data bases has been proposed. As result of that, the fuzzy relational model GEFRED has emerged, as well as the fuzzy extended Entity-Relationship model FuzzyEER and the fuzzy querying language FSQ, among others. These extensions include some forms of fuzzy attributes, such as the so-called Type 3. These attributes have values in a set of labels provided with a fuzzy similarity relation, which is an extension of equivalence relations in the fuzzy set theory. The implications of such attributes in partitioned query operators have not been previously explored. The contribution of this paper is on the syntactic and semantic definition of an extension to SQL, in order to allow for partitioned queries with Type 3 fuzzy attributes in grouping criteria. This extension increases the expressive power of the database language with respect to previous proposals.

Keywords: SQL, fuzzy queries, similarity relation, fuzzy attributes, GROUP BY.

INTRODUCCIÓN

Las bases de datos tradicionales solo manejan datos y condiciones precisos que en muchas ocasiones no representan las necesidades reales de información de los usuarios. La teoría de conjuntos difusos [30] 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 [5, 9]. Estas fueron luego generalizadas en GEFRED [19], un modelo extendido para bases de datos relacionales difusas.

¹ Departamento de Computación y Tecnología de la Información. Universidad Simón Bolívar. Valle de Sartenejas, Baruta, Caracas, Venezuela. E-mail: scarrasquel@usb.ve; crodrig@usb.ve; leonid@usb.ve

A partir de este se extendió el modelo Entidad Relación y el lenguaje SQL, surgiendo FuzzyEER [10] y FSQL [10].

Para representar datos difusos FuzzyEER [10] define 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 que admiten como valores 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 ellas, adicionalmente puede permitir distribuciones de posibilidad; y Tipo 4, similar a los Tipo 3, sin relaciones de similitud. Estos tipos de atributos se encuentran en FSQL [10], solo que con algunas limitaciones impuestas por la representación relacional definida por los autores para el almacenamiento de estos atributos, llamada FIRST-2 [10].

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 FuzzyEER, 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 que 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 Tipo 3, cuyo dominio es un conjunto de etiquetas dotado de una relación difusa de similitud. Algunos ejemplo de estos tipos de datos que se pueden conseguir en algunos dominios de aplicación son: los colores, identificados por sus nombres, ellos tienen una relación de similitud, en el sentido de ser parecidos entre sí; localidades geográficas que pueden tener una relación de similitud, en el sentido de ser cercanos; especies en la taxonomía animal o vegetal pueden modelarse con relación a su similitud, que sea un agregado de distintas características que les hacen parecidas o que las diferencian, entre otros.

Las relaciones de similitud son una herramienta usada en problemas de toma de decisiones en diversas áreas como la medicina, la industria petrolera, social, económica, de gestión, etc. Además permiten modelar conceptos relacionados a la psicología, sociología, lingüística, entre otras. Han sido utilizadas en estudios de espacios financieros [15]. En áreas como geobotánica son una herramienta importante para la clasificación de los datos de acuerdo con similitudes y diferencias.

Es importante aclarar el uso que se da en este artículo al término en castellano “agrupamiento”, de forma que no se genere confusión, pues traduce dos términos completamente diferentes del idioma inglés: *grouping* y *clustering*. El presente trabajo se refiere a expresiones de consulta en bases de datos que trabajan sobre tablas particionadas según criterios establecidos en la especificación del término *grouping*. Por otro lado, se hace la salvedad que este trabajo no trata sobre *clustering*, esta se refiere a métodos de minería de datos que permiten descubrir automáticamente diferentes grupos existentes dentro de un conjunto de datos.

Como motivación se presenta el caso en que se requiere hacer la búsqueda de ventas de repuestos de automóviles. A veces resulta necesario buscar no solo en la ciudad o sector donde habita la persona interesada, sino que hay que incluir localidades cercanas. Por ejemplo, en Venezuela se dispone de servicios de búsqueda conocidos como Infogüa.com o las Páginas Amarillas de CANTV; al buscar ventas de repuestos de automóviles, uno de los servidores devuelve 333 resultados, mientras que el otro 165. La búsqueda puede mejorarse un poco, indicando la ciudad deseada, sin embargo, esto resulta muy limitado, pues deja fuera de consideración ciudades cercanas, en ellas podría estar una solución más satisfactoria para el usuario. Para dar un mejor soporte a estos requerimientos sería deseable, por ejemplo, poder definir un dominio para las ciudades que esté provisto de una relación de similitud que represente la cercanía entre ellas. En teoría esto es factible con los datos Tipo 3 del modelo FuzzyEER. Sin embargo, el lenguaje FSQL no da soporte a cierto tipo de consultas cuando se involucra un atributo de esta naturaleza.

Con el propósito de dar semántica a consultas con ordenamiento sobre atributos Tipo 3, en un trabajo

previo [6] se estudiaron definiciones existentes para las relaciones difusas de similitud, y se hizo una nueva propuesta. En ese mismo artículo [6] se extendió el sublenguaje de definición de datos de SQL para la creación y uso de dominios de datos difusos Tipo 3. También en el referido trabajo [6] se definió la nueva forma de la cláusula ORDER BY de SQL, extendida para atributos Tipo 3.

El objetivo de la investigación aquí reportada es dar una sintaxis y semántica adecuada a consultas basadas en particionamiento (cláusulas GROUP BY y HAVING de SQL) cuando el criterio expresado involucra atributos Tipo 3. Se restringe el trabajo al caso en que el atributo toma como valor exactamente una etiqueta. Para el caso de permitir distribuciones de posibilidad, se requiere primero haber resuelto este caso más simple, por lo que sería tema de trabajo futuro.

Este trabajo se organiza como se describe a continuación. La segunda sección presenta los conceptos de la teoría de conjuntos difusos que son relevantes a esta investigación, en particular un nuevo concepto de relación difusa de similitud, propuesto por los autores, y la partición difusa inducida por esta. La tercera sección describe las cláusulas básicas de particionamiento (GROUP BY y HAVING) según la definición de los estándares SQL e introduce un ejemplo que servirá de base para ilustrar los nuevos conceptos. La cuarta sección presenta la extensión de SQL para definición de dominios de datos con relaciones de similitud. La quinta sección presenta la definición formal de las consultas con agrupamiento difuso en SQL propuestas en este trabajo, asimismo desarrolla paso a paso un ejemplo ilustrativo de los conceptos involucrados en este nuevo tipo de consultas. La sexta sección pone en evidencia la originalidad de la contribución aquí presentada al compararla con los esfuerzos que se han hecho en el área. Finalmente la séptima sección presenta las Conclusiones y Trabajos Futuros de la investigación.

PARTICIONES DIFUSAS

La teoría de conjuntos difusos fue propuesta por Zadeh [30] 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. Este concepto es flexibilizado en los conjuntos difusos, permitiendo no solo elementos incluidos y excluidos, sino también elementos parcialmente incluidos, con grados en el intervalo $(0,1)$.

Definición 1. Un **conjunto difuso** F en un universo X está definido por una **función de membresía** $\mu_F: X \rightarrow [0,1]$, donde los elementos tienen pertenencia gradual. Se dice que $x \in X$ tiene **pertenencia total** o está **completamente incluido** cuando $\mu_F(x)=1$; en formal dual se dice que está **completamente excluido** cuando $\mu_F(x)=0$; finalmente, está **parcialmente incluido** o **gradualmente incluido** cuando $\mu_F(x) \in (0,1)$.

Al conjunto formado por los elementos parcialmente incluidos se le conoce como el borde del conjunto difuso. El conjunto de los elementos completamente incluidos se les llama el núcleo. Los elementos que no están completamente excluidos conforman el soporte.

Definición 2. Sea F un conjunto difuso F en un universo X , se definen los conjuntos clásicos: **núcleo**(F) = $\{x \in X \mid \mu_F(x)=1\}$; **borde**(F) = $\{x \in X \mid 0 < \mu_F(x) < 1\}$; y **soporte**(F) = $\{x \in X \mid \mu_F(x) > 0\}$.

Para representar por extensión un conjunto difuso F en el universo X , se usa la notación $F = \{x_1/\mu_F(x_1), x_2/\mu_F(x_2), \dots, x_n/\mu_F(x_n)\}$ la que asocia cada elemento x_n con su grado de membresía $\mu_F(x_i)$. Si un elemento de X no aparece en la representación por extensión de F es porque está completamente excluido. Análogamente, para representar por comprensión un conjunto difuso F en el universo X se usa la notación $F = \{x/\mu_F(x) \mid x \in X\}$.

Zadeh [31] introdujo el concepto de relación de similitud como una generalización de la equivalencia. Este trabajo provee un marco conceptual para las relaciones difusas, como una extensión a la teoría clásica de relaciones. Una relación difusa en un universo X es un subconjunto difuso de $X \times X$. Según la definición original de Zadeh [31], una *relación de*

similitud S es una relación difusa reflexiva $\forall x \in X \mu_S(x, x) = 1$, simétrica $\forall x, y \in X \mu_S(x, y) = \mu_S(y, x)$ y transitiva $\forall x, y, z \in X \mu_S(x, z) = \max_y(\min(\mu_S(x, y), \mu_S(y, z)))$.

Con el propósito de dar semántica al ordenamiento en bases de datos sobre atributos con valores en un dominio formado por etiquetas provisto de una relación de similitud entre las etiquetas, en un trabajo previo [6] se estudiaron diferentes definiciones existentes para las relaciones difusas de similitud. En dicho trabajo se determinó que tales propuestas no eran satisfactorias en el contexto de estudio. Particularmente la insatisfacción obedece a la característica de transitividad, esta es justamente la que presenta más diferencia entre las definiciones consideradas. En consecuencia se propuso una nueva definición, que se presenta a continuación.

Definición 3. Sea X un conjunto de valores, S un subconjunto difuso de $X \times X$ caracterizado por $\mu_S: X \times X \rightarrow [0, 1]$, diremos que S es una **relación difusa de similitud** si y solo si cumple con las propiedades de **reflexividad** $\forall x \in X \mu_S(x, x) = 1$, **simetría** $\forall x, y \in X \mu_S(x, y) = \mu_S(y, x)$ y **transitividad** $\forall x, y, z \in X ((\mu_S(x, y) = 1 \wedge \mu_S(y, z) = \beta \Rightarrow \mu_S(x, z) = \beta) \wedge (\mu_S(x, y) = \beta \wedge \mu_S(y, z) = 1 \Rightarrow \mu_S(x, z) = \beta))$.

La transitividad propuesta es más estricta que otras que se encuentran en la literatura, en el sentido que exige que el grado de similitud de uno de los pares sea 1, para poder aplicar la regla transitiva.

Ejemplo 1. Suponga que se tienen nombres de algunos sectores de Caracas como “23 de Enero”, “Agua Salud”, “Campo Claro”, “San Agustín”. Es posible establecer una relación de similitud que describa la cercanía entre sectores. Dicha relación estaría definida por el grado de membresía del par (x, y) en el intervalo $[0, 1]$, como se observa en la Tabla 1.

Tabla 1. Relación de Similitud Difusa S (Cercanía entre sectores).

μ_s	23 de Enero	Agua Salud	Campo Claro	San Agustín
23 de Enero	1	1	0	0,5
Agua Salud	1	1	0	0,5
Campo Claro	0	0	1	0,3
San Agustín	0,5	0,5	0,3	1

La reflexividad se observa en la diagonal principal de la Tabla 1. La simetría es también evidente. La transitividad se refleja en el hecho de que las filas de “23 de Enero” y “Agua Salud” tienen los mismos valores.

Según Zadeh [31], una relación de similitud induce una partición difusa del conjunto X . Con cada $x \in X$ se asocia una clase de similitud $S[x]$ que es un conjunto difuso en X caracterizado por $\mu_{S[x]}(y) = \mu_S(x, y)$. Otros autores como Jacas [13], Ovchinnikov [21] adoptan el concepto de partición difusa inducida por una relación de similitud propuesto por Zadeh [31], salvo que cada uno concibe la relación de similitud con una definición distinta de transitividad. Por su parte, Faurous y Fillard [8] se basan en la misma idea de Zadeh [31] para inducir una partición difusa a partir de una relación de similitud, mas cambiando la acepción de transitividad y reflexividad e introduciendo el concepto de base, un conjunto de elementos que garantizan la separación de las partes mientras que permite solapamiento para elementos fuera de la base.

Existen otras definiciones de particiones difusas que no son inducidas por una relación de similitud, tales son los casos de las propuestas de Ma, Turksen y Kandel [17], Marsala [18], Guillaume [11], Ruspini [23], Bezdek [1], Tamura [26], Intan y Mukaidono [12], Yu, Mu y Shi [29].

Debido a que en el presente trabajo se está interesado en dar semántica al agrupamiento en bases de datos sobre atributos con valores en un dominio formado por etiquetas provisto de una relación de similitud, se tomará la noción de partición difusa inducida por una relación de similitud, como se indica en el trabajo de Zadeh [31], pero con la nueva concepción de relación difusa tomada de [6] (definición 3).

Definición 4. Sea X un conjunto de valores, S una relación de similitud difusa en X : S induce una **partición difusa** P_S sobre el conjunto de valores X , $P_S = \{S[x] \mid x \in X\}$; a cada elemento $S[x]$ de esta partición difusa lo llamamos la **clase difusa** de x , este es el conjunto difuso descrito por la función de membresía $\mu_{S[x]}(y) = \mu_S(x, y)$.

Ejemplo 2. La partición difusa inducida por la relación de similitud de la Tabla 1 es la siguiente: $P = \{S[\text{“23 de Enero”}], S[\text{“Campo Claro”}], S[\text{“San$

Agustín”]]. Siendo $S[“23 de Enero”] = S[“Agua Salud”] = \{“23 de Enero”/1, “Agua Salud”/1, “San Agustín”/0,5\}$; $S[“Campo Claro”] = \{“Campo Claro”/1, “San Agustín”/0,3\}$; $S[“San Agustín”] = \{“23 de Enero”/0,5, “Agua Salud”/0,5, “Campo Claro”/0,3, “San Agustín”/1\}$.

AGRUPAMIENTO CLÁSICO EN SQL

El lenguaje estándar de consultas a bases de datos relacionales SQL permite expresar consultas particionadas mediante la estructura sintáctica siguiente:

```
SELECT <resulting columns>
FROM <source relations>
WHERE <tuple condition>
GROUP BY <grouping columns>
HAVING <group condition>
```

La respuesta de esta consulta es un conjunto de filas de acuerdo con las siguientes especificaciones: la cláusula FROM establece la tabla de origen para los datos de la consulta. Esta tabla es formada por el producto cartesiano de las <source relations>, filtrado por el cumplimiento de la <tuple condition> en la cláusula WHERE. La cláusula GROUP BY indica que las tuplas de la tabla de origen son agrupadas de acuerdo con las distintas combinaciones de valores actuales para las <grouping columns>. Estos grupos son filtrados por el cumplimiento de la <group condition> en la cláusula HAVING. Por cada uno de los grupos seleccionados hay una fila en el resultado de la consulta. Los atributos en <resulting columns> deben estar contenidos en <grouping columns> o ser usados como argumento en una función de agregación en <resulting columns>. Así cada fila en el resultado es un representante de un grupo, posiblemente con valores agregados.

Como las cláusulas GROUP BY y HAVING son el objeto de esta investigación, se explicará con más detalle su funcionamiento. Ambas cláusulas son opcionales y producen una tabla agrupada, que es un conjunto de grupos, derivados durante la evaluación de dichas cláusulas. Un grupo es un conjunto de filas en las que todos los valores de la(s) columna(s) pertenecientes a la <grouping columns> son iguales. Si no se especifica una cláusula GROUP BY el grupo es toda la tabla.

Una tabla agrupada puede considerarse como una colección de tablas. Las funciones de agregación operan sobre las tablas individuales (o grupos) que están dentro de la tabla agrupada.

Se dice que una columna referenciada en la cláusula GROUP BY es una columna de agrupamiento. El resultado de la cláusula GROUP BY es una partición de la tabla en un conjunto de grupos. El conjunto está formado por el número mínimo de grupos tales que, para cada columna de agrupamiento de cada grupo de más de una fila, no hay dos valores distintos de esa columna de agrupamiento. Cada fila de un grupo dado contiene los mismos valores en todas las columnas de agrupamiento listadas. Cuando una condición de búsqueda es aplicada a un grupo, una referencia a una columna de agrupamiento es una referencia a su valor.

Cuando se especifica una cláusula HAVING, la tabla agrupada se deriva de la eliminación de los grupos, de la tabla resultante de la cláusula GROUP BY previamente especificada, que no cumplen la <group condition>. La cláusula HAVING solo tiene sentido cuando se ha especificado previamente una cláusula GROUP BY, por lo que cada columna referenciada en la <group condition> debe ser una columna de agrupamiento o estar especificada en una función de agregación o ser una referencia externa.

Cuando la <group condition> se aplica a un grupo dado, ese grupo es el argumento de la función de agregación contenida en la condición, a menos que la columna referenciada en la función de agregación sea una referencia externa. Si en la <group condition> aparece una subconsulta, esta es ejecutada para cada grupo de la tabla agrupada. El resultado de la subconsulta es utilizado en la evaluación de la <group condition> para cada grupo dado de la tabla agrupada. Si la subconsulta ejecutada contiene una referencia externa a una columna de la tabla agrupada, entonces la referencia es a los valores de esa columna en el grupo dado de dicha tabla.

La formalización del resultado para las cláusulas GROUP BY y HAVING que es el objeto de este estudio se describe a continuación.

Definición 5. Sea C la consulta SQL (clásico) $SELECT\ c_1, c_2, \dots, c_n\ FROM\ T\ GROUP\ BY\ k_1, \dots, k_o\ HAVING\ cond(G_i)$ donde $c_j \in \{k_1, \dots, k_o\}$

o c_j es de la forma $agg_j(a_j)$ con agg_j una función de agregación. Esta consulta establece la relación de equivalencia ρ sobre las tuplas de T : $\rho(t_p, t_q) \equiv \forall r \in \{1, \dots, o\} (t_p.k_r = t_q.k_r)$, la relación ρ induce sobre T una partición clásica P_ρ . Sea Γ la proyección de T en los atributos k_1, \dots, k_o , $\Gamma = \{(t.k_1, t.k_2, \dots, t.k_o) \mid t \in T\} = \{\gamma_1, \gamma_2, \dots, \gamma_l\}$ para $i \in \{1 \dots l\}$ sea $G_i \in P_\rho$ la clase tal que $\exists t \in T (\gamma_i = (t.k_1, t.k_2, \dots, t.k_o) \wedge t \in G_i)$. Entonces, **el resultado de C es: resultset (C)** $= \{(v_{i1}, v_{i2}, \dots, v_{in}) \mid i \in \{1 \dots l\} \wedge cond(G_i)\}$, donde $v_{ij} = \gamma_i.c_j$ si $c_j \in \{k_1, \dots, k_o\}$ y $v_{ij} = agg_j\{t.a_j \mid t \in G_i\}$ si $c_j = agg_j(a_j)$. Si la cláusula HAVING no se especifica la $cond(G_i) \equiv \text{true}$.

Ejemplo 3. Dada la relación de la Tabla 2 y la consulta `SELECT sector, COUNT(*) FROM VentasRepuestos GROUP BY sector`; el resultado es la Tabla 3.

Tabla 2. Ventas de Repuestos de Automóviles (VentasRepuestos).

	Nombre	Sector	tipo
t1	"AR y EA el Viejo"	"23 de Enero"	"Genéricos"
t2	"AR los Robles"	"Agua Salud"	"Genéricos"
t3	"Repuestos Pepecars"	"Campo Claro"	"Originales"
t4	"Inversiones Suniaka 777"	"San Agustín"	"Genéricos"
t5	"Versamy Distribuidora"	"San Agustín"	"Originales"

Tabla 3. Resultado de consulta con GROUP BY.

Sector	COUNT(*)
"23 de Enero"	1
"Agua Salud"	1
"Campo Claro"	1
"San Agustín"	2

DOMINIOS DE DATOS DIFUSOS

SQL permite la definición de dominios para describir nuevos tipos de datos dentro de un esquema. En [6] se extendió el SQL-DDL (SQL Data Definition Language) para permitir la creación y uso de dominios de datos difusos que tengan asociada una relación de similitud.

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

```
] , (<label>, <label>) / <value>,
..., (<label>, <label>) / <value>
}] }
```

Donde, $\langle \text{name} \rangle$ es el nombre del nuevo dominio; $(\langle \text{label} \rangle [, \langle \text{label} \rangle, \dots, \langle \text{label} \rangle])$ es la lista de etiquetas que definen el dominio; los especificadores $(\langle \text{label} \rangle, \langle \text{label} \rangle) / \langle \text{value} \rangle$ corresponden a los pares de la relación difusa de similitud para ese dominio; siendo $\langle \text{value} \rangle$ el valor del grado de membresía de dicho par en la relación. Solo es necesario especificar los pares básicos de la relación, como en un diagrama de Hasse, pues el resto de los pares se derivan por las reglas de reflexividad, simetría y transitividad.

Ejemplo 4. Para la relación de la Tabla 1 se crea el dominio de las localidades, así: `CREATE FUZZY DOMAIN sector AS VALUES ("23 de Enero", "Agua Salud", "Campo Claro", "San Agustín") SIMILARITY { ("23 de Enero", "Agua Salud") / 1, ("San Agustín", "23 de Enero") / 0.5, ("Campo Claro", "San Agustín") / 0.3 }`

También se permite modificar o eliminar la definición de un dominio difuso mediante las instrucciones `ALTER` y `DROP`.

AGRUPAMIENTO DIFUSO EN SQL

Es posible extender las SQL para que permita agrupamiento basado en particiones difusas inducidas por relaciones de similitud definidas para dominios de atributos difusos.

Se propone modificar la sintaxis de la cláusula `GROUP BY` permitiendo que cada columna en la lista aparezca sola, como es usual, o aparezca precedida de la palabra nueva clave `SIMILAR`:

```
GROUP BY [SIMILAR] <column> ...
[ SIMILAR] <column>
```

A continuación se presenta formalmente la nueva semántica de la cláusula `GROUP BY`.

Definición 6. Una **consulta de agrupamiento difuso** es una expresión de la forma `SELECT c_1, c_2, \dots, c_n FROM T GROUP BY $m_1 k_1, \dots, m_o k_o$ HC`. Donde: T es una expresión SQL que denota una tabla; k_1, \dots, k_o son atributos de T ; cada c_j es tal que $c_j \in \{k_1, \dots, k_o\}$

o $c_j = \text{agg}_j(a_j)$ con agg_j una función de agregación y a_j atributo de T ; HC es la cláusula Having, $HC = \varepsilon$ (el *string* vacío) o HC es de la forma **HAVING cond**, siendo *cond* una expresión lógica llamada la condición de grupo; cada $m_i = \text{SIMILAR}$ o $m_i = \varepsilon$ (el *string* vacío), en el caso que $m_i = \text{SIMILAR}$, $\text{dom}(k_i)$ debe ser un dominio difuso fd_i dotado de una relación de similitud S_i .

Ejemplo 5. Sea la relación VentasRepuestos de la Tabla 2 con el atributo ciudad de tipo localidad con la definición del Ejemplo 4, la siguiente sería una consulta de agrupamiento difuso: **SELECT tipo, sector, COUNT(*) FROM VentasRepuestos GROUP BY tipo, SIMILAR sector.**

Definición 7. Sea C una consulta de agrupamiento difuso **SELECT c_1, c_2, \dots, c_n FROM T GROUP BY $m_1 k_1, \dots, m_o k_o$ HC.** Diremos que la **relación difusa σ establecida por C** sobre las tuplas de T es una relación definida mediante la función de membresía: $\mu_\sigma(t_p, t_q) = \min_{r \in \{1, \dots, o\}} \mu_r(t_p, k_r, t_q, k_r)$, donde para cada m_r la relación μ_r sobre $\text{dom}(k_r)$ se define por los casos siguientes. En el caso que $m_r = \varepsilon$: si $t_p.k_r = t_q.k_r$ entonces $\mu_r(t_p, k_r, t_q, k_r) = 1$, si $t_p.k_r \neq t_q.k_r$ entonces $\mu_r(t_p, k_r, t_q, k_r) = 0$. En el caso que $m_r = \text{SIMILAR}$ sea $\text{dom}(k_i)$ el dominio difuso fd_i dotado de una relación de similitud S_i , entonces $\mu_r(t_p, k_r, t_q, k_r) = \mu_{S_i}(t_p, k_r, t_q, k_r)$.

Ejemplo 6. Sea la relación VentasRepuestos de la Tabla 2, la consulta C del Ejemplo 5 establece la relación difusa σ de la Tabla 4.

Tabla 4. Relación difusa σ .

μ_σ	t1	t2	t3	t4	t5
t1	1	1	0	0,5	0
t2	1	1	0	0,5	0
t3	0	0	1	0	0,3
t4	0,5	0,2	0	1	0
t5	0	0	0,3	0	1

Proposición 1. La relación difusa σ establecida por una consulta de agrupamiento difuso C es una relación de similitud.

Demostración:

Sea $C = \text{SELECT } c_1, c_2, \dots, c_n \text{ FROM } T \text{ GROUP BY } m_1 k_1, \dots, m_o k_o \text{ HC}$.

σ es reflexiva: Sea $j \in \{1, \dots, o\}$ tal que $\mu_\sigma(t_p, t_p) = \mu_j(t_p, k_j, t_p, k_j)$. Si $m_j = \varepsilon$ entonces $\mu_j(t_p, k_j, t_p, k_j) = 1$ por lo que $\mu_\sigma(t_p, t_p) = 1$. Si $m_j = \text{SIMILAR}$ entonces $\mu_j(t_p, k_j, t_p, k_j) = \mu_{S_j}(t_p, k_j, t_p, k_j)$. Como S_j es una relación de similitud, S_j es reflexiva entonces $\mu_{S_j}(t_p, k_j, t_p, k_j) = 1$ por lo que $\mu_\sigma(t_p, t_p) = 1$.

σ es simétrica: Sea $r \in \{1, \dots, o\}$; en el caso que $m_r = \varepsilon$, si $t_p.k_r = t_q.k_r$, $\mu_r(t_p, k_r, t_q, k_r) = \mu_r(t_q, k_r, t_p, k_r) = 1$, si $t_p.k_r \neq t_q.k_r$, $\mu_r(t_p, k_r, t_q, k_r) = \mu_r(t_q, k_r, t_p, k_r) = 0$; en el caso que $m_r = \text{SIMILAR}$, como S_r es simétrica $\mu_r(t_p, k_r, t_q, k_r) = \mu_{S_r}(t_p, k_r, t_q, k_r) = \mu_{S_r}(t_q, k_r, t_p, k_r) = \mu_r(t_q, k_r, t_p, k_r)$. Luego $\mu_\sigma(t_p, t_q) = \min_{r \in \{1, \dots, o\}} \mu_r(t_p, k_r, t_q, k_r) = \min_{r \in \{1, \dots, o\}} \mu_r(t_q, k_r, t_p, k_r) = \mu_\sigma(t_q, t_p)$.

σ es transitiva: Sea $\mu_\sigma(t_p, t_q) = 1$ y $\mu_\sigma(t_q, t_u) = \beta$. Sea $r \in \{1, \dots, o\}$; en el caso que $m_r = \varepsilon$, se cumple que $t_p.k_r = t_q.k_r$ pues $\mu_r(t_p, k_r, t_q, k_r) = 1$ para todo r , por lo que $\mu_r(t_p, k_r, t_u, k_r) = \mu_r(t_q, k_r, t_u, k_r)$; en el caso que $m_r = \text{SIMILAR}$, como S_r es transitiva y $\mu_r(t_p, k_r, t_q, k_r) = \mu_{S_r}(t_p, k_r, t_q, k_r) = 1$ entonces $\mu_r(t_p, k_r, t_u, k_r) = \mu_{S_r}(t_p, k_r, t_u, k_r) = \mu_{S_r}(t_q, k_r, t_u, k_r) = \mu_r(t_q, k_r, t_u, k_r)$. De donde $\beta = \mu_\sigma(t_q, t_u) = \min_{r \in \{1, \dots, o\}} \mu_r(t_q, k_r, t_u, k_r) = \min_{r \in \{1, \dots, o\}} \mu_r(t_p, k_r, t_u, k_r) = \mu_\sigma(t_p, t_u)$.

QED

De acuerdo con la definición 4, y la proposición 1, la relación σ induce sobre las tuplas de la tabla T una partición difusa $P_\sigma = \{\sigma[t] \mid t \in T\}$.

Ejemplo 7. Sea la relación difusa de similitud σ establecida por C según el Ejemplo 6, se tiene la partición $P_\sigma = \{\sigma[t] \mid t \in \text{VentasRepuestos}\}$, donde $\sigma[t1] = \sigma[t2] = \{t1/1, t2/1, t4/0.5\}$; $\sigma[t3] = \{t3/1, t5/0.3\}$; $\sigma[t4] = \{t1/0.5, t2/0.5, t4/1\}$; y $\sigma[t5] = \{t3/0.3, t5/1\}$.

Definición 8. Sea C una consulta de agrupamiento difuso **SELECT c_1, c_2, \dots, c_n FROM T GROUP BY $m_1 k_1, \dots, m_o k_o$ HC.** Sea σ la relación de similitud difusa establecida por C . El **resultado de C es el conjunto difuso de tuplas $\text{resultset}(C) = \{(v_{i1}, v_{i2}, \dots, v_{in}) / \mu_i \mid i \in \{1 \dots l\}\}$** . Donde l es la cardinalidad de la proyección de T en los atributos k_1, \dots, k_o , $\Gamma = \{(t.k_1, t.k_2, \dots, t.k_o) \mid t \in T\}$, cuya extensión es $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_l\}$; para cada $i \in \{1 \dots l\}$ sea $\tau_i \in T$ cualquier tupla tal que $\gamma_i = (\tau_i.k_1, \tau_i.k_2, \dots, \tau_i.k_o)$, si $c_j \in \{k_1, \dots, k_o\}$ entonces $v_{ij} = \tau_i.c_j$, si c_j es de la forma $\text{agg}_j(a_j)$ entonces $v_{ij} = \text{agg}_j\{\tau_i.a_j / \mu_{\sigma[\tau_i]}(\tau_i) \mid \tau_i \in \sigma[\tau_i]\}$, si $HC = \varepsilon$ entonces $\mu_i = 1$, si HC es de la forma **HAVING cond** entonces $\mu_i = \mu(\text{cond}(\sigma[\tau_i]))$, esto es el grado en que el grupo $\sigma[\tau_i]$ satisface la expresión *cond*.

Ejemplo 8. Sea C la consulta del Ejemplo 5, su resultado es el conjunto difuso de tuplas $resultset(C) = \{(\tau_i, tipo, \tau_i.sector, COUNT\{\tau/\mu_{\sigma[\tau_i]}(\tau) \mid \tau \in \sigma[\tau_i]\}) / 1 \mid i \in \{1 \dots 5\}\}$ cuya extensión se muestra en la Tabla 5. Se observa que en todas las tuplas el grado es 1 debido a que C no tiene cláusula HAVING.

Tabla 5. Resultado de la consulta con GROUP BY.

Tipo	sector	COUNT(*)	μ
“Genéricos”	“23 de Enero”	2,5	1
“Genéricos”	“Agua Salud”	2,5	1
“Genéricos”	“San Agustín”	2	1
“Originales”	“Campo Claro”	1,3	1
“Originales”	“San Agustín”	1,3	1

Las condiciones de grupo que aparecen en una cláusula HAVING *cond*, por lo general contienen funciones de agregación. En los actuales momentos la evaluación de este tipo de funciones sobre conjuntos difusos es todavía un problema abierto, el que debe ser explorado. Sin embargo, para el caso particular de la función COUNT que calcula el número de elementos del grupo, su interpretación difusa es equivalente al cálculo de la cardinalidad de un conjunto difuso. Varias propuestas se han realizado para obtener dicha cardinalidad [7, 22, 27, 32]. De ellas la que más ha sido usada en el tema de particiones difusas es la propuesta original de Zadeh [32], conocida como $\Sigma count$. Adoptamos esta semántica para la función COUNT.

Definición 9. Sea $COUNT(a_j)$ en una consulta de agrupamiento difuso C , con σ la relación de similitud difusa establecida por C , τ_i una tupla en T la tabla base de C , definimos $COUNT(a_j)$ como $COUNT\{\tau.a_j/\mu_{\sigma[\tau_i]}(\tau) \mid \tau \in \sigma[\tau_i]\} = \Sigma\{\mu_{\sigma[\tau_i]}(\tau) \mid \tau \in \sigma[\tau_i]\}$ (cardinalidad $\Sigma count$).

Ejemplo 9. Sea P_σ la partición difusa del Ejemplo 7, $COUNT(t1.*) = COUNT(t2.*) = \Sigma\{1, 1, 0,5\} = 2,5$; $COUNT(t3.*) = \Sigma\{1, 0,3\} = 1,3$; $COUNT(t4.*) = \Sigma\{0,5, 0,5, 1\} = 2$; y $COUNT(t5.*) = \Sigma\{0,3, 1\} = 1,3$.

Debido a que al momento no se dispone de una semántica establecida para las funciones de agregación en general sobre conjuntos difusos, al momento adoptaremos una aproximación precisa de tales funciones, la extensión para una semántica más adecuada será tema de trabajos futuros.

Definición 10. Sea $agg_j(a_j)$ una función de agregación distinta de $COUNT(a_j)$ en consulta de agrupamiento difuso C , con σ la relación de similitud difusa establecida por C , t_i una tupla en T la tabla base de C , definimos $agg_j(a_j)$ como sigue: $agg_j\{t.a_j/\mu_{\sigma[t_i]}(t) \mid t \in \sigma[t_i]\} = agg_j\{t.a_j \mid t \in soporte\ \sigma[t_i]\}$.

TRABAJOS RELACIONADOS

Existen varias propuestas para la incorporación de conceptos provenientes de la teoría de conjuntos difusos en SQL, como: OMRON [20], FQUERY [14], ISKREOT [16], PFSQL [25], SOFTSQL [2], SQLf [3] y FSQ [10], entre otras [33]. De ellas las más destacadas por su completitud son SQLf y FSQ [28].

En lo que concierne a este trabajo, FSQ es la única de estas propuestas que contempla toda la variedad de datos difusos del modelo FuzzyEER [10], en particular los datos Tipo 3: etiquetas sobre las cuales se define una relación de similitud. Sin embargo, en un trabajo previo [6] se observó que FSQ tiene una deficiencia en el concepto que se maneja para la relación de similitud y en su forma de manipularla.

A pesar que FSQ permite la creación de tipos de datos difusos y usarlos para definir atributos de las tablas, en definición de este lenguaje no se permite usar un atributo definido como de Tipo 3 en una cláusula GROUP BY. Los autores de FSQ [10] no indican la razón por la que se impone esta restricción. Puede inferirse del hecho que las operaciones de agrupamiento y ordenamiento en consultas a bases de datos se encuentran estrechamente vinculadas por razones operativas. El hecho de no tener una semántica definida para el ordenamiento de un tipo de datos naturalmente dificultaría no solo las consultas ordenadas, sino también las particionadas.

Por otro lado, SQLf [3] ha sido más recientemente extendido [4] con una cláusula GROUP BY difusa. Esta extensión trabaja con dominios de atributos clásicos sobre los cuales se define un conjunto de términos lingüísticos que denotan conjuntos difusos de una partición del dominio de los atributos. No consideran el caso de atributos Tipo 3.

También un operador de agrupamiento basado en similitud fue propuesto por Silva, Aref y Ali [24].

Ellos extienden la cláusula GROUP BY de SQL para que las particiones no sean por igualdad, sino por similitud. Sin embargo los grupos que consideran no son particiones difusas, sino particiones clásicas obtenidas por técnicas de minería, no consideran el uso de relaciones de similitud difusas.

CONCLUSIONES

El modelo de bases de datos relacional difuso extendido incorpora, entre otras bondades, el permitir 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.

Debido a que hay un interés en permitir atributos cuyo dominio sea un conjunto de etiquetas que esté provisto de una relación difusa de similitud, en un trabajo previo se ha extendido el sistema de objetos y tipos de SQL, de manera que se puedan especificar y usar tales dominios.

El tener atributos cuyos valores sean etiquetas en un dominio con una relación difusa de similitud tiene consecuencias en algunos de los operadores de consultas de SQL, las que deben ser definidas. Particularmente, una relación difusa de similitud induce una partición difusa en el universo en que está definida. Estos conceptos pueden ser usados para extender la semántica de las consultas de agrupamiento en SQL. El presente artículo explica mejor la especificación y uso de atributos Tipo 3 respecto de lo que se había hecho en FSQ, ya que ese lenguaje simplemente prohibía el uso de tales atributos en particionamiento.

En este trabajo se propone una extensión para este tipo de consultas en presencia de atributos definidos sobre dominios constituidos por un conjunto de etiquetas provisto de una relación difusa de similitud. En esta extensión, cada atributo de la cláusula GROUP BY puede ser precedido por la nueva palabra clave SIMILAR. El uso de esta indica que en el agrupamiento debe considerarse la relación de

similitud definida para el dominio de tal atributo. Esta cláusula define una relación difusa sobre las tuplas de la tabla base que combina los distintos atributos de agrupamiento. Aquí se demuestra que tal relación es una relación difusa de similitud. Basado en esta relación se define formalmente la semántica de una consulta de agrupamiento difuso. Se considera el uso de funciones de agregación sobre las clases de la partición difusa inducida por esta relación. Por su uso común y sencillez se adopta la cardinalidad $\Sigma count$ de Zadeh para dar semántica a la función COUNT. Para el resto de las funciones de agregación se usa la agregación clásica sobre el soporte de la clase difusa. Esto obedece al hecho de que al momento no hay una semántica establecida para tales funciones sobre conjuntos difusos, esto será tema a desarrollar en trabajos futuros.

La noción de agrupamiento 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.

De acuerdo con el 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 y agrupamiento 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. La contribución acá es más bien teórica en cuanto a la definición del lenguaje SQL. Se abren las puertas a trabajos futuros que traten sobre algoritmos para el tipo de consultas aquí presentadas, extensiones de productos de tecnología de bases de datos que incorporen esta funcionalidad y métodos que permitan su integración en aplicaciones de usuario final. Esto permitirá mejorar los sistemas de información y de toma de decisiones que puedan hacer uso de la nueva funcionalidad definida.

En cuanto a aspectos de rendimiento, no ha sido objetivo de este trabajo. 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 que se reportaría cuando se tengan resultados.

AGRADECIMIENTOS

Este trabajo es resultado del Proyecto de Grupo “Desafíos del Modelo Relacional Difuso” que cuenta con el apoyo de la Universidad Nacional Experimental Politécnica “Antonio José de Sucre”, Vicerrectorado de Barquisimeto. Agradecemos el apoyo brindado por el profesor Carlos Lamedá, coordinador adjunto de este proyecto, así como a los otros miembros del grupo. Finalmente reconocemos que “Toda buena dádiva y todo don perfecto descende de lo alto, del Padre de las luces, en donde no hay mudanza, ni sombra de variación” (Santiago 1:17).

REFERENCIAS

- [1] J.C. Bezdek. “Pattern Recognition with Fuzzy Objective function Algorithms”. Plenum Press, New York, USA. 1981.
- [2] G. Bordogna and G. Psaila. “Customizable Flexible Querying for Classical Relational Databases”. In: J. Galindo (Ed.). Handbook of Research on Fuzzy Information Processing in Databases. Vol. 1, pp. 191-217. Information Science Reference, Hershey, PA, USA. 2008.
- [3] P. Bosc and O. Pivert “SQLf: A Relational Database Language for Fuzzy Querying”. IEEE Trans. on Fuzzy Systems. Vol. 3, Issue 1. 1995.
- [4] P. Bosc and O. Pivert. “On a fuzzy group-by clause in SQLf”. Proc. of the IEEE International Conference on Fuzzy Systems, pp. 1-6. 2010.
- [5] B.P. Buckles and F.E. Petry. “A fuzzy representation of data for relational databases”. Fuzzy Sets and Systems. Vol. 7, pp. 213-226. 1982.
- [6] S. Carrasquel, R. Rodríguez y L. Tineo. “Consultas con Ordenamiento basado en Similitud”. Telematique. Vol. 12 N° 1, pp. 24-45. 2013.
- [7] D. Dubois and H. Prade. “Fuzzy cardinality and the modeling of imprecise quantification”. Fuzzy Sets and Systems. Vol. 16, Issue 2, pp. 199-230. 1985.
- [8] P. Faurous and J.P. Fillard. “A New Approach to the Similarity Relations in the Fuzzy Set Theory”. Information Sciences. Vol. 75, pp. 213-221. 1993.
- [9] S. Fukami, M. Umamo, M. Muzimoto and H. Tanaka. “Fuzzy database retrieval and manipulation language”. (Tech. Rep. N° AL-78-85). IEICE Technical Reports. Vol. 78, Issue 233, pp. 65-72. 1979.
- [10] J. Galindo, A. Urrutia and M. Piattini. “Fuzzy Databases: Modeling, Design and Implementation”. Idea Group Publishing Hershey, USA. 2006.
- [11] S. Guillaume and B. Charnomordic. “Generating an Interpretable Family of Fuzzy Partition From Data”. IEEE Transactions on Fuzzy Systems. Vol. 12, Issue 3, pp. 324-335. 2004.
- [12] R. Intan and M. Mukaidono. “Degree of Similarity in Fuzzy Partition”. AFSS, pp. 20-26. 2002.
- [13] J. Jacas. “Similarity Relations-The Calculation of Minimal Generating Families”. Fuzzy Sets and Systems. Vol. 35, pp. 151-162. 1990.
- [14] J. Kacprzyk and S. Zadrozny. “Fuzzy Queries in Microsoft Access v.2”. Proceedings of Workshop on Fuzzy Database Systems and Inf. Retrieval, pp. 61-66. 1995.
- [15] L. Lazzari, P. Moulié and M. Eriz. “Crisp and Fuzzy Relations”. Application to Financial Space. Cuadernos del Cimbage. N° 10, pp. 17-46. 2008.
- [16] G. Loo and K. Lee. “An Interface to Databases for Flexible Query Answering: A Fuzzy-Set Approach”. LNCS, N° 1873, pp. 654-663. Springer-Verlag. 2000.
- [17] M. Ma, I.B. Turksen and A. Kandel. “Fuzzy partition and fuzzy rule base”. Journal of Information Sciences. Vol. 108, pp. 109-121. 1998.
- [18] C. Marsala. “Fuzzy Partition Inference Over a Set of Numerical Values”. Technical Report, LAFORIA-IBP. University Pierre et Marie Curie. France. Octubre, 1995.
- [19] J. Medina, O. Pons and A. Vila. “GEFRED: A Generalized Model of Fuzzy Relational

- Databases". *Information Sciences*. Vol. 77, Issue 6, pp. 87-109. 1994.
- [20] H. Nakajima, T. Sogoh and M. Arao. "Fuzzy Database Language and Library-Fuzzy Extension to SQL". *Proceedings of 2nd Fuzz-IEEE*, pp. 477-482. 1983.
- [21] S. Ovchinnikov. "Similarity relations, fuzzy partitions, and fuzzy orderings". *Fuzzy Sets and Systems*. Vol. 40, pp. 107-126. 1991.
- [22] D. Ralescu. "Cardinality, quantifiers, and the aggregation of fuzzy criteria". *Fuzzy Sets and Systems*. Vol. 69, pp. 355-365. 1995.
- [23] E.H. Ruspini. "A new approach to clustering". *Information and Control* 15, pp. 22-32. 1969.
- [24] Y. Silva, W. Aref and M. Ali. "Similarity Groupby". *Proceedings of the 25th International Conference on Data Engineering, ICDE 2009*, pp. 904-915. 2009.
- [25] A. Takači and S. Škrbić. "Data Model of FRDB with Different Data Types and PFSQL". In: J. Galindo (Ed.), *Handbook of Research on Fuzzy Information Processing in Databases*. Hershey, PA, USA. Information Science Reference. 2008.
- [26] S. Tamura, S. Higuchi and K. Tanaka. "Pattern classification based on fuzzy relations". *IEEE Trans. Syst. Man & Cybern. SMC-1*, N° 1, pp. 61-66. 1971.
- [27] L. Tineo. "A Fuzzy Quantifiers' Interpretation for Database Querying". *Fuzzy Information Processing Theories and Applications, FIP 2003*, pp. 423-428. Beijing. 2003.
- [28] A. Urrutia, L. Tineo and C. González. "FSQL and SQLf: Towards a Standard in Fuzzy Databases". In: J. Galindo (Ed.), *Handbook of Research on Fuzzy Information Processing in Databases*. Vol. 1, pp. 270-298. Hershey, PA, USA. Information Science Reference. 2008.
- [29] J. Yu, Y. Mu and H. Shi. "A short note on Fuzzy Partition". *Proceedings of the First International Conference on Machine Learning and Cibernetics*, pp. 1451-1454. Beijing, Pekín. 2002.
- [30] L.A. Zadeh. "Fuzzy Sets". *Information Control*. Vol. 8, pp. 338-353. 1965.
- [31] L.A. Zadeh. "Similarity Relations and Fuzzy Orderings". *Information Sciences*. pp. 177-200. 1971.
- [32] L.A. Zadeh. "A computational approach to fuzzy quantifiers in natural languages". *Computer Mathematics with Applications*. N° 9, pp. 149-183. 1983.
- [33] S. Zadrozny, G. De Tré, R. De Caluwe and J. Kacprzyk. "An overview of fuzzy approaches to flexible database querying". In: J. Galindo (Eds.), *Handbook of Research on Fuzzy Information Processing in Databases*. Hershey, PA, USA. Information Science Reference. 2008.