novática
Revista de la Asociación de Técnicos de Informática

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software).

<http://www.ati.es/novatica/>
<http://www.ati.es/reicis/>

ATI es miembro fundador de **CEPIS** (*Council of European Professional Informatics Societies*), representa a España en **IFIP** (*International Federation for Information Processing*) y es miembro de **CLEI** (*Centro Latinoamericano de Estudios de Informática*) y de **CECUA** (*Confederation of European Computer User Associations*). Asimismo tiene un acuerdo de colaboración con **ACM** (*Association for Computing Machinery*) y colabora con diversas asociaciones informáticas españolas.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid
Gutierre de Cetina 24, 28017 Madrid •Tlfn.914029391 <novatica@ati.es>
Administración y Redacción ATI Cataluña
Calle Ávila 50, 3a planta, local 9, 08005 Barcelona
Tlfn.934125235 <secregen@ati.es>
Redacción ATI Andalucía <secreand@ati.es>
Redacción ATI Galicia <secregal@ati.es>
Suscripción y Ventas <novatica.subscripciones@atinet.es>
Publicidad Gutierre de Cetina 24, 28017 Madrid
Tlnf.914029391 <novatica@ati.es>
Imprenta: Impresión Offset Derra S.L., Llull 41, 08005 Barcelona.
Depósito legal: B 15.154-1975 -- ISSN: 0211-2124; CODEN NOVAEC
Portada: "El guardián" - Concha Arias Pérez / © ATI
Diseño: Fernando Agresta / © ATI 2003

sumario

Nº 238, noviembre 2016 - febrero 2017, año XLII

Monografía del próximo número: "Salud y tecnología"

Benchmark de consultas de agrupamiento y ordenamiento difuso

Soraya Carrasquel[1], David Coronado[1], Ricardo Monascal[1], Rosseline Rodríguez[1,2], Leonid Tineo[1,2]

[1]Departamento de Computación y Tecnología de la Información, Universidad Simón Bolívar, Caracas (Venezuela); [2]Universidad Católica Nuestra Señora de la Asunción, Paraguay

<{scarrasquel, dcoronado, rmonascal, crodrig, leonid}@usb.ve>

Resumen: *Se han propuesto extensiones a SQL para soportar atributos cuyo dominio es un conjunto de etiquetas provisto de una relación de similitud, conocidos como atributos difusos Tipo 3. Dichas extensiones consideran la definición y manipulación de este tipo de dominios, así como consultas con ordenamiento y agrupamiento difuso. Mediante una arquitectura medianamente acoplada, estas nuevas funcionalidades fueron añadidas a PostgreSQL, surgiendo así fuzzydoDB. En este trabajo se presenta el análisis de desempeño de las consultas con ordenamiento y agrupamiento basado en atributos difusos Tipo 3, así como una base de datos experimental para este tipo de* benchmark*.*

Palabras clave: *Atributos difusos Tipo 3,* benchmark*,* GROUP BY*,* ORDER BY*, SGBD.*

1. Introducción

La mayoría de los sistemas gestores de bases de datos (SGBD) implementan el modelo relacional, que supone datos precisos, con valores perfectamente conocidos o ausentes. Sin embargo, en el mundo real, puede haber información parcial o imprecisa sobre valores de algunos datos. Una herramienta matemática que permite modelar este tipo de información es la teoría de conjuntos difusos [16].

La aplicación de esta teoría ha dado lugar al surgimiento del modelo relacional difuso generalizado GEFRED [10] y el modelo entidad relación extendido difuso Fuzzy EER [8]. Éste distingue cuatro tipos de atributos difusos: *Tipo 1*, datos precisos sobre los cuales se permiten consultas con etiquetas lingüísticas interpretadas como números difusos; *Tipo 2*, datos posibilísticos sobre dominios ordenados; *Tipo 3*, etiquetas lingüísticas no ordenadas provistas de una relación de similitud, y distribuciones de posibilidad sobre ellas; y *Tipo 4*, similar a los *Tipo 3* pero sin la relación de similitud.

Para los *Tipo 3*, sin distribuciones de posibilidad, se han estudiado sus implicaciones en consultas basadas en ordenamiento y agrupamiento [3][4]. Los resultados han sido implementados en una extensión [1] del SGBD MariaDB [9], la cual permite especificar y almacenar atributos *Tipo 3*, y procesar consultas que involucren estos atributos, particularmente ordenamiento y el agrupamiento difuso [3][4]. Esta extensión fue migrada a PostgreSQL y ampliada para dar soporte a otros tipos de datos difusos, esta nueva versión se llama *fuzzy*doDB [6]. Con esta extensión se planteó realizar el *benchmark* de consultas con atributos *Tipo 3*.

Fue necesario crear una base de datos experimental con atributos *Tipo 3* para este *benchmark*, pues, dado lo reciente de esta extensión, no existen bases de datos públicas que contemplen estos atributos. Aquí se presenta dicha base de datos experimental y se reporta el resultado del análisis de desempeño de *fuzzy*doDB, sobre ésta, en el procesamiento de consultas con agrupamiento y ordenamiento difuso.

Este artículo se organiza como sigue: En la **sección 2** se presenta el marco teórico, los dominios *Tipo 3*, y su uso en ordenamiento y agrupamiento difuso. En la **sección 3**, se describen algunos detalles relevantes de *fuzzy*doDB. En la **sección 4**, se explica la base de datos para el *benchmark*. En la **sección 5**, se muestran los detalles del diseño experimental. En la **sección 6**, se plantea el análisis de los resultados obtenidos en los experimentos. Finalmente, en la **sección 7**, se proponen las conclusiones y trabajos futuros.

2. Marco teórico

Un conjunto difuso F [16] es un subconjunto de un conjunto clásico X, caracterizado por una función de membresía $\mu_F: X \rightarrow [0,1]$, donde 0 es la completa exclusión, 1 la completa inclusión y hay elementos que pertenecen gradualmente (posiblemente). En bases de datos, este concepto permite expresar preferencias del usuario o particularidades del contexto al dar semántica a criterios vagos (o condiciones difusas), así como representar y manipular atributos de datos imprecisos (o difusos).

Para incluir consultas o datos difusos en SQL, se han propuesto varias extensiones. Una de ellas es FSQL [7], basada en GEFRED [10] y Fuzzy EER [8] que define cuatro tipos de atributos difusos. El *Tipo 3* consiste en un dominio formado por etiquetas, provisto de una relación de similitud.

Sea S un conjunto difuso en $X \times X$ caracterizado por $\mu_S: X \times X \rightarrow [0,1]$, S es una relación de similitud [4] si es 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 $\forall x,y,z \in X\ ((\mu_S(x,y)=1 \Rightarrow \mu_S(x,z)=\mu_S(y,z)) \wedge (\mu_S(y,z)=1 \Rightarrow \mu_S(x,z)=\mu_S(x,y)))$. S induce una partición difusa [3], $P_S=\{ S[x] \mid x \in X \}$, donde $S[x]$ se llama la clase difusa de x, con $\mu_{S[x]}(y)=\mu_S(x,y)$.

En trabajos previos [3][4] se proponen y formalizan extensiones a SQL para definir datos *Tipo 3* y su uso en consultas ordenadas y agrupadas, que se resumen a continuación.

2.1. Dominios de datos con relaciones de similitud

Un dominio *Tipo 3* se especifica así [4]: CREATE FUZZY DOMAIN *fd* AS VA-

μ_S	Europe	Oceania	Asia	Africa	Antarctica
Europe	1	0.4	0.9	0.8	0
Oceania	0.4	1	0.7	0.4	0.4
Asia	0.9	0.7	1	0.6	0.3
Africa	0.8	0.4	0.6	1	0.8
Antarctica	0	0.4	0.3	0.8	1

Tabla 1. Relación de similitud para CONTINENTS.

> “ Para incluir consultas o datos difusos en SQL, se han propuesto varias extensiones. Una de ellas es FSQL, basada en GEFRED y Fuzzy EER que define cuatro tipos de atributos difusos ”

LUES $l_1,\ldots,l_n$ [SIMILARITY { $(l_{il}, l_{jl})/v_1,\ldots, (l_{im}, l_{jm})/v_m$ }]. Donde *fd* es el nombre del dominio, $l_1,\ldots, l_n$ son las etiquetas en el dominio, $(l_{ik},l_{jk})/v_k$ es un par en la relación de similitud con grado de membresía v_k. Sólo es necesario especificar los pares de la relación base; los correspondientes a la reflexividad, simetría y transitividad están sobreentendidos; el resto tienen membresía cero.

Por ejemplo, así se especifica un dominio *Tipo 3* de continentes (ver **tabla 1**): CREATE FUZZY DOMAIN CONTINENTS AS VALUES (Europe, Oceania, Asia, Africa, Antarctica) SIMILARITY {(Europe, Oceania)/0.4, (Europe, Asia)/0.9, (Europe, Africa)/0.8, (Oceania, Asia)/0.7, (Oceania, Africa)/0.4, (Oceania, Antarctica)/0.4, (Asia, Africa)/0.6, (Asia, Antarctica)/0.3, (Africa, Antarctica)/0.8}.

2.2. Ordenamiento usando similitud

La cláusula ORDER BY se extendió así [4]: ORDER BY *criterio*$_1$,...,*criterio*$_o$, donde cada *criterio*$_i$ es de la forma: - k_i d_i siendo k_i un atributo y d_i un especificador de orden (ASC o DESC); - k_i START l_i siendo k_i un atributo difuso *Tipo 3*, l_i una etiqueta en el dominio de k_i. En este último caso, las tuplas se ordenan descendentemente por el grado de similitud del valor de k_i con l_i.

Por ejemplo, dada una tabla COUNTRY de nueve filas, con atributos name, region, continent y surfacearea, siendo continent un atributo *Tipo 3* del dominio CONTINENTS, la consulta SELECT name, region, continent, surfacearea FROM COUNTRY ORDER BY continent START Europe, produce como resultado la siguiente lista de registros ordenada según la relación de similitud de la **tabla 1**: [(France, Western Europe, Europe, 551500), (Spain, Southern Europe, Europe, 505992), (Armenia, Middle East, Asia, 29800), (Japan, Eastern Asia, Asia, 377829), (Algeria, Northern Africa, Africa, 2381741), (Ghana, Western Africa, Africa, 238533), (New Zealand, Australia and New Zealand, Oceania, 270534), (New Caledonia, Melanesia, Oceania, 18575), (Antarctica, Antarctica, Antarctica, 13120000)].

2.3. Agrupamiento usando similitud

Se extendió la cláusula GROUP BY así [3]: GROUP BY [SIMILAR] $k_1,\ldots,$ [SIMILAR] k_o, la palabra clave SIMILAR es opcional, sólo se usa si k_i es *Tipo 3* y se quiere considerar la partición difusa inducida por su relación de similitud.

Por ejemplo, dada la tabla COUNTRY (subsección anterior), el atributo continent es *Tipo 3*, su relación de similitud (ver **tabla 1**) induce la partición difusa {{Europe/1, Asia/0.9, Africa/0.8, Oceania/0.4}, {Oceania/1, Asia/0.7, Africa/0.4, Antarctica/0.4, Europe/0.4}, {Asia/1, Europe/0.9, Oceania/0.7, Africa/0.6, Antarctica/0.3}, {Africa/1, Antarctica/0.8, Europe/0.8, Asia/0.6, Oceania/0.4}, {Antarctica/1, Africa/0.8, Oceania/0.4, Asia/0.3}}.

La consulta: SELECT continent, COUNT(*) FROM COUNTRY GROUP BY SIMILAR continent; produce: [(Europe, 6.2), (Oceania, 5.4), (Asia, 4.7), (Africa, 6.2), (Antarctica, 4.0)]. COUNT se calculó según el operador $\sum count$ de Zadeh [17], se usó por ser el más sencillo [3].

3. SGBD extendido *fuzzy*doDB

En esta sección se describe brevemente *fuzzydo*DB [6], su arquitectura y catálogo. Para más detalles, se recomienda consultar el trabajo dedicado al prototipo inicial [1].

La versión actual incluye todas las variantes de atributos difusos según se proponen en Fuzzy EER [8]. En *fuzzy*doDB los *Tipo 3* no permiten el uso de distribuciones de posibilidad. Se ha extendido la clasificación con un *Tipo 5* que son distribuciones de posibilidad sobre *Tipo 3*. A continuación se describe la porción correspondiente a los atributos *Tipo 3*.

3.1. Catálogo difuso

Los metadatos correspondientes a dominios *Tipo 3* se incluyen en el catálogo relacional de objetos de la base de datos. Para ello, se define un esquema de base de datos compuesto de cuatro tablas: DOMAIN, LABEL, SIMILARITY y COLUMN.

DOMAIN (domainId, tableSchema, domainName) contiene los dominios *Tipo 3*, donde domainId es la clave primaria autogenerada, también identificado por su domainName dentro de una base de datos (tableSchema).

LABEL (labelId, domainId, labelName), almacena las etiquetas (labelName) de los dominios de datos difusos, referenciado por domainId, con clave primaria autogenerada labelId, y (domainId, labelName) una clave alterna.

SIMILARITY (label1Id, label2Id, value, derived) contiene los pares de etiquetas (label1Id, label2Id) que conforman una relación de similitud, indicando su grado de membresía (value) y el atributo derived que indica si el par pertenece a la relación base o fue derivado por reflexividad, simetría y transitividad; label1Id y label2Id son claves foráneas a LABEL.

COLUMN (tableSchema, tableName, columnName, domainId) contiene la definición de columnas cuyo tipo es un dominio difuso (domainId), el cual se identifica por su nombre (columnName), junto con el nombre de la tabla (tableName) y base de datos (tableSchema) a la que pertenece; (tableSchema, tableName, columnName) **es la clave primaria y domainId es una clave foránea a DOMAIN.**

3.2. Arquitectura

Para extender la funcionalidad de un SGBD se han propuesto tres arquitecturas: acoplamiento fuerte, acoplamiento débil y acoplamiento medio [15]. En *fuzzy*doDB [1][6] se usa una arquitectura medianamente acoplada (ver **figura 1**), que consiste en una implementación intermedia donde la lógica fue programada en el lenguaje nativo del SGBD original.

Esta extensión posee una capa externa programada en Java que implementa los esquemas de traducción del lenguaje extendido a SQL [2]. La capa externa es un mediador que consta de tres módulos [1]: el *front-end* es un analizador sintáctico o *parser* de SQL extendido que genera un árbol sintáctico abstracto (AST); el *core* es un traductor que transforma un AST del lenguaje extendido a un AST del lenguaje nativo; el *back-end* (*deparser*) es un generador de código que toma el AST y escribe la instrucción en SQL.

Durante el análisis sintáctico se genera la lista de columnas *Tipo 3* a ser traducidas. Para ello, se recorren las expresiones de las cláusulas SELECT, WHERE, ORDER BY y GROUP BY, y se busca cada columna en el catálogo difuso. Luego se invoca al traduc-

> “ En *fuzzy*doDB se usa una arquitectura medianamente acoplada que consiste en una implementación intermedia donde la lógica fue programada en el lenguaje nativo del SGBD original ”

tor, que utiliza el catálogo para generar un AST correspondiente a la sentencia traducida al lenguaje nativo de SQL, de acuerdo a los esquemas de traducción [2].

Posteriormente, se recorre de nuevo el AST para localizar expresiones que involucren columnas *Tipo 3*. Estas columnas son sustituidas en el AST por la columna de la tabla LABEL que contiene la etiqueta conocida por el usuario. Si esta sustitución se realizó en la cláusula SELECT, se asegura que la expresión mantenga su alias o el nombre por el cual se accede al valor en el resultado.

La generación de código realiza el proceso inverso (*deparser*) para transformar el AST en un string SQL. Finalmente, se envía el string SQL al manejador para que sea ejecutado, obteniendo el resultado de la instrucción original.

4. Base de datos experimental

El pgfoundry.org [11] es anfitrión de varios proyectos de software relacionados con PostgreSQL [12].

Para probar el prototipo *fuzzy*doDB se buscó uno de los ejemplos de bases de datos allí publicados. De entre las disponibles, se escogió la base de datos *World*.

Ésta es una base de datos de prueba encontrada en la documentación oficial de MySQL, la cual fue migrada a PostgreSQL por esta organización [11]. El interés de tomar una base de datos de este sitio obedece al hecho que son recursos disponibles públicamente y conocidos por la comunidad de desarrollo de PostgreSQL. Sin embargo, estas bases de datos no incluyen atributos difusos que es justamente la funcionalidad añadida por *fuzzy*doDB. De manera que para la base de datos experimental a usar en este trabajo se hizo imperativo el realizar modificaciones a la base de datos *World*, con el fin de incluir atributos difusos.

4.1. Modelo lógico

La base de datos *World* [11] posee tres tablas: COUNTRY, CITY y COUNTRYLANGUAGE. A éstas se realizaron modificaciones con el fin de incluir atributos difusos. A continuación, se describe el modelo lógico de la base de datos *World* modificada. Se colocan en cursiva los atributos difusos correspondientes a las modificaciones realizadas. Los dominios difusos se presentarán en la siguiente subsección.

COUNTRY (code, name, *continent*, region, surfacearea, indepyear, population, lifeexpentacy, gnp, gnpgold, localname, governmentform, headofstate, capital, code2) contiene la información acerca de los países del mundo con 239 registros, donde code es la clave primaria y capital es una clave foránea que referencia a CITY. Se modificó el atributo *continent* para que fuera del nuevo dominio difuso *Tipo 3* FUZZYCONTINENT.

CITY (id, name, countrycode, district, population, cars, *ePopulation*), contiene la información sobre ciudades de esos países con 4079 registros, donde id es la clave primaria y countrycode es una clave foránea que referencia a COUNTRY. Se agregó una nueva columna denominada *ePopulation* la cual contiene datos del dominio difuso *Tipo 2* FUZZYPOPULATION.

COUNTRYLANGUAGE (countrycode, language, isofficial, percentage) contiene los idiomas hablados en cada país con 985 registros, donde la clave primaria es (countrycode, language) y countrycode es una clave foránea que referencia a COUNTRY.

4.2. Dominios difusos

Se definieron dominios difusos, uno *Tipo 2* y el otro *Tipo 3*. Para este último, se mostrará la definición en SQL extendido pues este dominio es el objeto de este trabajo.

FUZZYPOPULATION: permite representar valores estimados para la cantidad actual de habitantes de una ciudad. Si bien la base de datos original tiene por cada ciudad un valor de población (*population*), dado que la población real es un dato cambiante, en un momento podría no tenerse el valor preciso, por lo que tiene sentido modelar esta información con un atributo difuso *Tipo 2*.

FUZZYCONTINENT: existen siete continentes en la base de datos, éstos pueden considerarse con cierta similitud entre ellos, por lo que puede modelarse como *Tipo 3*. En este caso, la similitud que se ha representado tiene que ver con cercanía geográfica, donde los grados de similitud se han puesto en forma arbitraria. Tal definición de similitud se justifica por ser un estudio experimental y el propósito aquí es enfocarse en el tema de desempeño.

Estos valores pueden cambiarse por otros dependiendo de las preferencias del usuario,

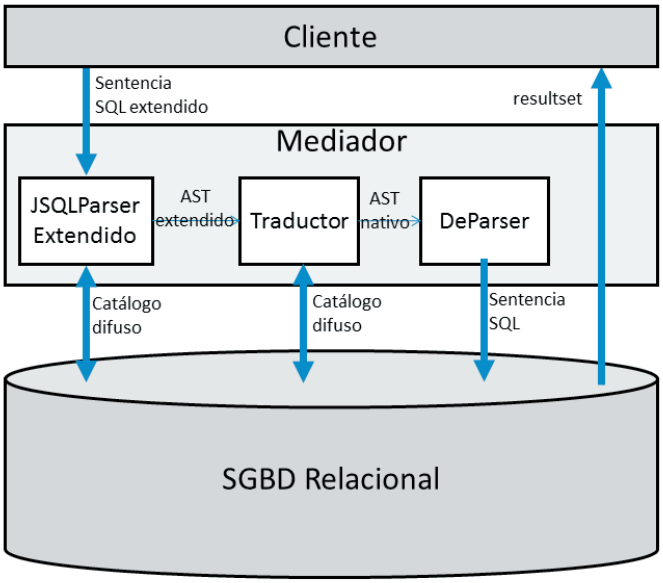

Figura 1. Arquitectura de *fuzzy*doDB. Fuente [1].

> “ Para la base de datos experimental a usar en este trabajo se hizo imperativo el realizar modificaciones a la base de datos *World*, con el fin de incluir atributos difusos ”

captando mejor la semántica que se quiera adoptar para la relación de similitud entre continentes.

La definición del dominio difuso sería CREATE FUZZY DOMAIN FUZZYCONTINENT AS VALUES (Europe, Oceania, Asia, North America, Africa, Antarctica, South America) SIMILARITY {(Europe, Oceania)/0.4, (Europe, Asia)/0.9, (Europe, North America)/0.5, (Europe, Africa)/0.9, (Europe, Antarctica)/0.2, (Europe, South America)/0.3, (Oceania, Asia)/0.7, (Oceania, North America)/0.2, (Oceania, Africa)/0.4, (Oceania, Antarctica)/0.8, (Oceania, South America)/0.4, (Asia, North America)/0.3, (Asia, Africa)/0.6, (Asia, Antarctica)/0.5, (Asia, South America)/0.3, (North America, Africa)/0.4, (North America, Antarctica)/0.2, (North America, South America)/0.9, (Africa, Antarctica)/0.3, (Africa, South America)/0.3, (Antarctica, South America)/0.6}.

5. Diseño experimental

Se realizó un estudio de desempeño experimental usando un modelo estadístico formal [5][14]. La idea de este método es explicar la influencia de diversos factores considerados en los valores observados de los experimentos. La importancia de un factor es medida por la proporción del total de variación en la respuesta que es explicada por el factor. En el análisis de desempeño se persigue mostrar resultados con el menor número de experimentos.

Se adoptó un diseño factorial $2^{(k+2)}$ [5][14] donde k es el número de factores utilizados, para cada factor se consideran únicamente dos niveles. Además, se realizaron cuatro réplicas por cada una de las pruebas para obtener mayor precisión, esta es la razón del exponente (k+2). Se consideraron dos factores relevantes (k=2) para explicar el comportamiento del sistema.

Hay dos grupos de experimentos, cada uno con un experimento para ordenamiento y otro para agrupamiento. El primer grupo considera el impacto del tipo de atributo en el SGBD extendido, mediante los factores: volumen (*V*) y tipo de atributo (*A*). El factor *A* se refiere al atributo involucrado en la cláusula que se está experimentando, el cual puede ser preciso (A_1) o *Tipo 3* (A_2). El segundo grupo considera el impacto de la extensión del SGBD respecto al original en consultas con atributos precisos, mediante los factores: volumen (*V*) y SGBD (*S*). El factor *S* puede ser PostgreSQL original (S_1) o PostgreSQL extendido (S_2), es decir *fuzzy*doDB.

En todos los experimentos se usa la tabla CITY. El volumen de datos original de esta tabla era 4079 tuplas que ocupaban 298 KB. Este volumen resultó ser poco representativo para los fines del análisis de desempeño. Para aumentar el volumen se hicieron réplicas de los datos. Se consideró el factor volumen (*V*) con dos niveles: el bajo (V_1) con 32632 registros (2413 KB) y el alto (V_2) con 30128 filas (9.713 KB).

Se elaboraron cuatro consultas experimentales. Para el ordenamiento y agrupamiento difuso se tomó el atributo *continent* de la tabla COUNTRY. Fue necesario hacer las consultas sobre las tablas CITY y COUNTRY, combinadas mediante la clave foránea countrycode y la clave primaria code. Para el ordenamiento y agrupamiento clásico se tomó el atributo district que está en la tabla CITY, sin embargo, a fin de tener igualdad de condiciones, en las consultas clásicas se usa también la combinación de las tablas CITY y COUNTRY.

Las consultas experimentales son:

- SELECT district FROM CITY, COUNTRY WHERE countrycode=code ORDER BY district;
- SELECT *continent* FROM CITY, COUNTRY WHERE countrycode=code ORDER BY *continent* START Europe;
- SELECT district, count(*) FROM CITY, COUNTRY WHERE countrycode=code GROUP BY district;
- SELECT *continent*, count(*) FROM CITY, COUNTRY WHERE countrycode=code GROUP BY SIMILAR *continent*;

Todas estas consultas se usan en el primer grupo de experimentos, donde se varía el tipo de atributo. Para el segundo grupo de experimentos, en el cual lo que se varía es el SGBD, se tomaron únicamente las consultas que usan atributos precisos.

Para la corrida de los experimentos, se fijaron las características del hardware, la carga del computador y las conexiones. Las pruebas se realizaron en una máquina VIT P2400 que posee un procesador Intel Core i3-3110M a 2.4 GHz, una memoria RAM de 1.8GB y un disco duro de 150 GB. El disco estaba particionado para alojar dos sistemas de operación y las pruebas se realizaron sobre el sistema operativo Ubuntu 14.04 LTS de 32 bits. La verdadera capacidad de la máquina es de 320 GB de disco duro y de 2 GB de RAM. En cuanto a la carga de la máquina, en el momento de realizar las pruebas se tenía únicamente como proceso operativo el SGBD. Las conexiones a las bases de datos trabajaron sobre el servidor local de la máquina.

Para garantizar igualdad de condiciones entre las corridas de los experimentos, antes de ejecutar cada consulta, se limpiaba la memoria caché del SGBD.

La variable observada en el estudio fue el tiempo de ejecución (*T*) de las consultas medido en milisegundos, obtenido con el comando “time”.

Este diseño experimental se corresponde con el modelo aditivo $T=T'+\beta_1 A+\beta_2 V+\beta_3 AV+\varepsilon$, para el primer grupo de experimentos, y $T=T'+\beta_1 S+\beta_2 V+\beta_3 SV+\varepsilon$, para el segundo grupo de experimentos.

6. Análisis de los resultados

En esta sección se presentan los resultados de los experimentos realizados (ver **tabla 2**) y su análisis descriptivo. Se utilizó el software estadístico R para las pruebas de ajuste del modelo aditivo, obteniendo las tablas ANOVA.

6.1. Análisis de varianza

Para el primer grupo de experimentos, según las correspondientes ANOVA (ver **tabla 3**), se obtuvo: En las consultas con ordenamiento, tanto el factor volumen como el tipo de atributo, influyen de manera significativa, así como su interacción, aunque en menor grado. En el caso de las consultas con agrupamiento, ninguno de los factores influye significativamente.

En cuanto al segundo grupo de experimentos, según la **tabla 3**, se obtuvo: Para las

> “En el caso de consultas con agrupamiento, la gráfica muestra que no hay crecimiento significativo del tiempo de ejecución con el aumento del volumen de datos. Adicionalmente, los tiempos promedio son similares para ambos tipos de atributos”

Atributo	Volumen	Réplica	ORDER BY	GROUP BY
Preciso	Bajo	R1	6660	4525
		R2	9063	4968
		R3	6764	4378
		R4	7904	4489
	Alto	R1	14705	4504
		R2	14861	5288
		R3	16044	4629
		R4	14455	4687
Difuso	Bajo	R1	6133	4242
		R2	5963	5634
		R3	6110	4264
		R4	6108	4390
	Alto	R1	11900	5063
		R2	11121	4895
		R3	11218	5093
		R4	10707	4769

(a)

SGBD	Volumen	Réplica	ORDER BY	GROUP BY
Original	Bajo	R1	5594	3466
		R2	5264	3393
		R3	4693	2860
		R4	5091	2996
	Alto	R1	9437	3807
		R2	8458	4193
		R3	8625	4090
		R4	10128	3404
Extendido	Bajo	R1	6660	4525
		R2	9063	4968
		R3	6764	4378
		R4	7904	4489
	Alto	R1	14705	4504
		R2	14861	5288
		R3	16044	4629
		R4	14455	4687

(b)

Tabla 2. Tiempo de ejecución (ms) observado. (a) Resultados de experimentos del primer grupo. (b) Resultados de experimentos del segundo grupo.

Grupo	Consulta	ANOVA
Atributo	ORDER BY	Df Sum Sq Mean Sq F value Pr(>F) A 1 28079401 28079401 55.60 7.67e-06 *** V 1 158168352 158168352 313.21 5.79e-10 *** A:V 1 5109860 5109860 10.12 0.00791 ** Residuals 12 6059970 504997
	GROUP BY	Df Sum Sq Mean Sq F value Pr(>F) A 1 48620 48620 0.294 0.598 V 1 259590 259590 1.568 0.234 A:V 1 18360 18360 0.111 0.745 Residuals 12 1986903 165575
SGBD	ORDER BY	Df Sum Sq Mean Sq F value Pr(>F) S 1 68748972 68748972 109.67 2.17e-07 *** V 1 130416400 130416400 208.05 6.08e-09 *** S:V 1 11675889 11675889 18.63 0.001 ** Residuals 12 7522298 626858
	GROUP BY	Df Sum Sq Mean Sq F value Pr(>F) S 1 5358068 5358068 53.353 9.42e-06 *** V 1 777483 777483 7.742 0.0166 * S:V 1 257810 257810 2.567 0.1351 Residuals 12 1205128 100427
		Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Tabla 3. ANOVA de cada uno de los cuatro experimentos realizados.

consultas con ordenamiento, ambos factores, SGBD y volumen de datos, son altamente significativos, mientras que la interacción lo es un poco menos. Para el caso de las consultas con agrupamiento, el factor realmente significativo es el SGBD.

6.2. Análisis descriptivo

6.2.1. Experimentos según tipo de atributo

Para las consultas con ordenamiento, la gráfica de la **figura 2(a)** muestra que el crecimiento del tiempo de ejecución aumenta según el volumen de datos donde la tendencia es más fuerte para atributos precisos.

Esto puede explicarse porque en el caso de consultas con atributos precisos, las tuplas se ordenaron en base a cadenas de caracteres correspondientes a los valores de la columna district, mientras que en el caso de atributos difusos se ordenan en base al grado de similitud de la etiqueta para el atributo *continent* respeto al valor Europe, los cuales son números reales. El ordenamiento de secuencias de caracteres es mucho más costoso que el ordenamiento de números reales.

Cabe mencionar que la escogencia del atributo district para este experimento fue intencional. En el caso de haber diseñado *continent* como un atributo clásico, éste sería una cadena de caracteres. Por otro lado, aun siendo *continent* un atributo difuso, en caso de no usar la opción START de la cláusula ORDER BY, tendría que ordenarse como cadena de caracteres.

Dada la significancia de influencia de los factores volumen y tipo de atributo, así como su interacción, y el comportamiento antes descrito, se podría concluir que para consultas ORDER BY, a la medida que crece el volumen de datos, el desempeño es mejor con atributos difusos que con atributos precisos.

En el caso de consultas con agrupamiento, la gráfica de la **figura 2(b)** muestra que no hay crecimiento significativo del tiempo de ejecución con el aumento del volumen de datos. Adicionalmente, los tiempos promedio son similares para ambos tipos de atributos. Se concluye que el desempeño de *fuzzy*doDB para consultas con GROUP BY es igual si el agrupamiento es difuso o preciso.

6.2.2. Experimentos según SGBD

Para las consultas con ordenamiento, la gráfica de la **figura 2(c)** muestra que el crecimiento del tiempo de ejecución aumenta según el volumen de datos donde la tendencia es más fuerte para el caso del manejador extendido. Hay una diferencia entre el tiempo de ejecución de las consultas sobre el manejador extendido que va aumentando a medida que el volumen crece. Esto se explica por el impacto del procesamiento adicional generado por la extensión.

En el caso de las consultas con agrupamiento, la gráfica de la **figura 2(d)** muestra que no hay crecimiento significativo del tiempo de ejecución con el aumento del volumen de datos. Los tiempos promedios para SGBD extendido son mayores que para el original, sin embargo, la diferencia en este caso pareciera ser relativamente constante, lo cual corrobora la significancia del factor SGBD.

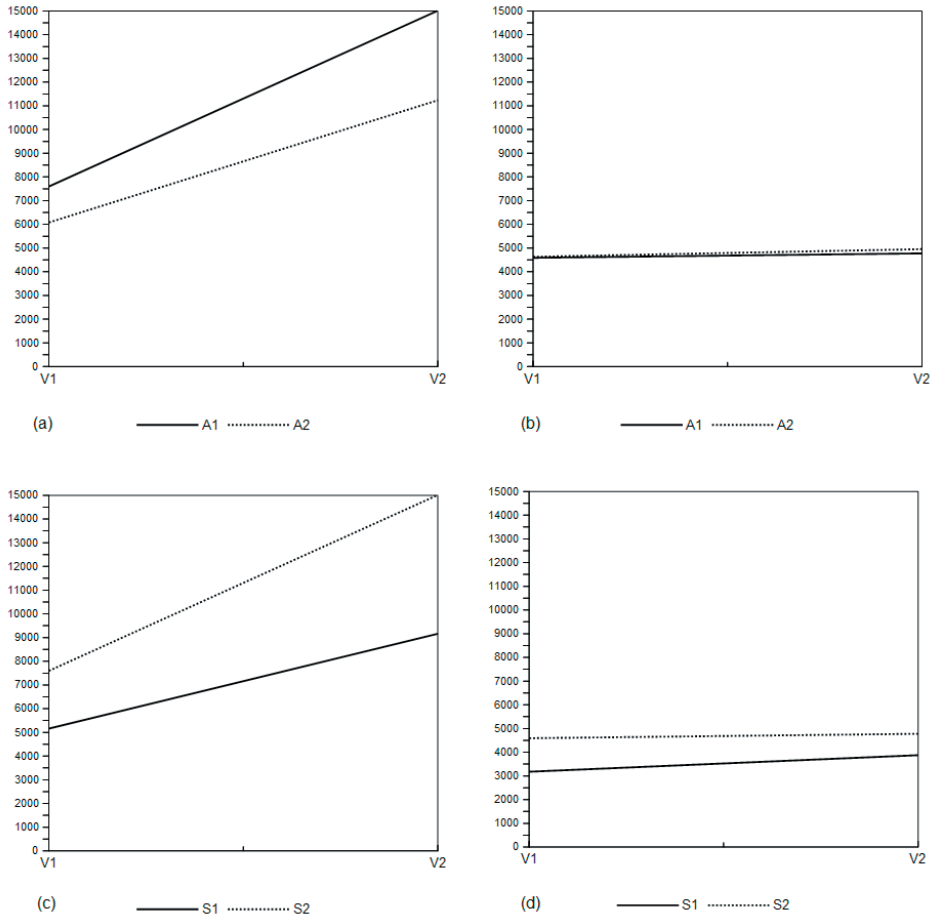

Figura 2. Promedio de tiempo de ejecución (ms), gráficas de interacción de factores. (a) Volumen-Atributo en consulta ORDER BY. (b) Volumen-Atributo en consulta GROUP BY. (c) Volumen-SGBD en consulta ORDER BY. (d) Volumen- SGBD en consulta GROUP BY.

7. Conclusiones y trabajos futuros

Este trabajo reporta un *benchmark* de consultas con ordenamiento y agrupamiento difuso sobre atributos *Tipo 3* en *fuzzy*doDB, una extensión con acoplamiento medio de PostgreSQL. A partir de una base de datos pública llamada *World* , se creó una base de datos experimental con atributos difusos que permite evidenciar funcionalidades de *fuzzy*doDB y puede ser usada en futuros *benchmark* con atributos difusos.

Se observó que para las consultas ORDER BY, a mayor volumen de datos, el ordenamiento difuso es mejor que el del ordenamiento preciso de etiquetas. Esto se debe a que el orden se realiza en base a grados de similitud y no en base a cadenas de caracteres. En el caso de consultas con la cláusula GROUP BY, el tiempo de ejecución de agrupamiento difuso es igual al preciso. Estos resultados sugieren que es razonable la inclusión de atributos difusos en los SGBD.

Se evidenció que la extensión degradó el desempeño del SGBD. Esto se debe a que todas las consultas pasan por el mediador, generando sobrecarga en el procesamiento. Este es el costo por más funcionalidad en una arquitectura de acoplamiento medio. Para evitar este costo, habría que implementar la extensión con una arquitectura de acoplamiento fuerte, lo cual sería mucho más complejo en términos de desarrollo pues involucra una intervención a código abierto.

Sólo se consideraron atributos *Tipo 3* y consultas con ordenamiento y agrupamiento sencillo. Se plantea a futuro hacer *benchmark* con otros tipos de atributos difusos y consultas más complejas.

Agradecimientos

Muchas gracias a nuestros alumnos en Miniproyecto de Desarrollo de Software de los últimos tres años, quienes contribuyeron con la implementación y pruebas de *fuzzy*doDB. Hablar de *World* y ordenamiento nos hace reflexionar que "*Tú ya eras Dios aun antes que las montañas se formaran y que crearas la tierra y el mundo. Tú eras y siempre serás Dios*" (Salmos 90:2).

Referencias

[1] S. Carrasquel, A. Gyomrey, S. Moreau, R. Rodríguez, B. Stornelli, C.Timaury, L. Tineo. "Extensión de MariaDB para ordenamiento y agrupamiento difuso". *Novática. Revista de la Asociación de Técnicos de Informática*, *Nº 229*, pp. 92-97, 2014.

[2] S. Carrasquel, R. Monascal, R. Rodríguez, L. Tineo. "Processing of Queries with Fuzzy Similarity Domains". *Handbook of Research on Innovative Database Query Processing Techniques*, L.Yan, Ed. Hershey, USA: IGI Global, 2016, pp. 88–128.

[3] S. Carrasquel, R. Rodríguez, L. Tineo. "Consultas con Agrupamiento basado en Similitud". *Ingeniare. Revista chilena de ingeniería*, *vol. 22 Nº 4*, pp. 517-527, 2014.

[4] S. Carrasquel, R. Rodríguez, L. Tineo. "Consultas con Ordenamiento basado en Similitud". *Telematique Vol. 12, N° 1*, pp. 24-45, 2013.

[5] O. Castejón. *Diseño y Análisis de Experimentos con Statistix*. Fondo Editorial Biblioteca Universidad Rafael Urdaneta, 2011.

[6] D. Coronado, S. Carrasquel, R. Monascal, R. Rodriguez, L. Tineo. "Portal de fuzzydoDB". *Memorias de la Tercera Conferencia Nacional de Computación, Informática y Sistema*. pp. 328–332. Caracas, Venezuela, octubre 2015.

[7] J. Galindo. "FSQL (Fuzzy SQL) A Fuzzy Query Language". Universidad de Málaga. <www.lcc.uma.es/~ppgg/FSQL/>.

[8] J. Galindo, A. Urrutia, M. Piattini. *Databases: Modeling, Design and Implementation*. Idea Group Publishing Hershey, USA, 2006.

[9] MariaDB Foundation. *MariaDB* <mariadb.org/>.

[10] J. Medina, O. Pons, A. Vila. "GEFRED: A Generalized Model of Fuzzy Relational Databases". *Information Sciences*, *Vol. 77, no. 6*, pp. 87-109, 1994.

[11] PostgreSQL Development Group. " *Forge*", 2014. <pgfoundry.org>.

[12] PostgreSQL Web Team. "*Pgfoundry* <wiki.postgresql.org/wiki/Pgfoundry>.

[13] R Core Team. "*R: A language and environment for statistical computing*". R Foundation for Statistical Computing, Vienna, Austria, 2015 <http://www.R-project.org/>.

[14] J. Raj. *The Art of Computer Systems Performance*, John Wiley & Sons, Inc, 1991.

[15] R. Timarán. "Arquitecturas de Integración del Proceso de Descubrimiento de Conocimiento con Sistemas de Gestión de Bases de Datos: un Estado del Arte". *Ingeniería y Competitividad Universidad del Valle, Colombia, vol. 3, no. 2*

[16] L. A. Zadeh. "Fuzzy Sets". *Information Control* , pp. 338-353. 1965.

[17] L. A. Zadeh. "A computational approach to fuzzy quantifiers in natural languages". *Computer Mathematics with Applications*. , pp. 149-183. 1983.