مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (30) العدد (1) 2008
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (30) No. (1) 2008

استخدام المنطق الضبابي في لغة الاستعلامات البنيوية (SQL) ومقارنتها مع المنطق الكلاسيكي

الدكتورة كندة أبو قاسم *
الدكتور رياض ضاهر**
مضر وينس ***

(تاريخ الإيداع 12 / 12 / 2007. قُبِل للنشر في 2008/2/18)

□ الملخّص □

أصبح لتطبيقات قواعد البيانات استخدامات جمة بسبب التطور الهائل الذي طرأ على تقانات قواعد البيانات، وتبنى معظم تطبيقات قواعد البيانات على نموذج قواعد البيانات العلائقية ومستفيدة من لغة الاستعلامات البنيوية Structured Query Language (SQL) التي تعتمد على المنطق ثنائي القيمة bi-valued logic لذلك فهي لا تمثل الواقع الحقيقي وتغفل جانب هام من الحقيقة. لقد سعينا من خلال هذا البحث إلى استخدام تقانات المنطق الضبابي Fuzzy Logic في لغة الاستعلامات البنيوية SQL بتصميم قاعدة بيانات علائقية وبناء استعلامات تعتمد المنطق الضبابي وتطبيقها في الزمن الحقيقي فحصلنا على نتائج أكثر دقة وتعبر عن الواقع الحقيقي.

الكلمات المفتاحية: قواعد البيانات – المنطق الضبابي – لغة الاستعلامات البنيوية SQL

* أستاذ مساعد – قسم هندسة الحاسبات والتحكم الآلي– كلية الهندسة الميكانيكية والكهربائية– جامعة تشرين– اللاذقية– سورية.
** أستاذ مساعد– قسم هندسة الحاسبات والتحكم الآلي– كلية الهندسة الميكانيكية والكهربائية– جامعة تشرين– اللاذقية– سورية.
*** طالب ماجستير–قسم هندسة الحاسبات والتحكم الآلي–كلية الهندسة الميكانيكية والكهربائية–جامعة تشرين–اللاذقية– سورية.

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (30) العدد (1) 2008
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (30) No. (1) 2008

Using Fuzzy Logic in Structured Query Language (SQL) and Comparing It with Classic Logic

Dr. Kinda Abu Kassem*
Dr. Reyad Daher **
Muddar wainis ***

(Received 12 / 12 / 2007. Accepted 18/2/2008)

□ ABSTRACT □

Due to database revolution, the database applications have been used widely, and the most common database application is based on relational database model, using the Structured Query Language (SQL), which depends on bi-valued logic. So it does not represent the real world and hide a very important side of the fact. In this research, we try to use fuzzy logic technique in SQL by designing relational database and building some queries depending on fuzzy logic in order to test it in real time so that we retrieve results closer to the real world.

Keywords: database, fuzzy logic, SQL**.**

*Associate Professor, Department of Computer Engineering and Automatic Control, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria.
**Asssociate Professor, Department of Computer Engineering and Automatic Control, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria.
***Postgraduate Student, Department of Computer Engineering and Automatic Control, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria.

مقدّمة:

لا يمكن التعبير عن الواقع الحقيقي بشكل دقيق عن طريق المنطق ثنائي القيم لذلك كان لا بد من إيجاد طريقة لتمثيل الواقع بصورة أكثر دقة، فكان مفهوم المنطق الضبابي Fuzzy Logic الذي أوجده الدكتور لطفي ذاده عام 1965 والذي كان الغرض الأساسي منه " التعامل مع الظواهر الطبيعية كما يفهمها العامل البشري". ومع الانتشار الواسع لقواعد البيانات ومتطلبات استرجاع البيانات منها (الاستعلام) برزت أهمية استخدام المنطق الضبابي في عملية الاستعلام.

هدف البحث وأهميته:

يهدف هذا البحث إلى توظيف المنطق الضبابي في عملية الاستعلام في قواعد البيانات بهدف الحصول على نتائج أقرب ما تكون إلى الطريقة التي يرغب العامل البشري في الحصول عليها. تم تطبيق هذا البحث على قاعدة البيانات للفواتير الهاتفية في مديرية الهاتف.

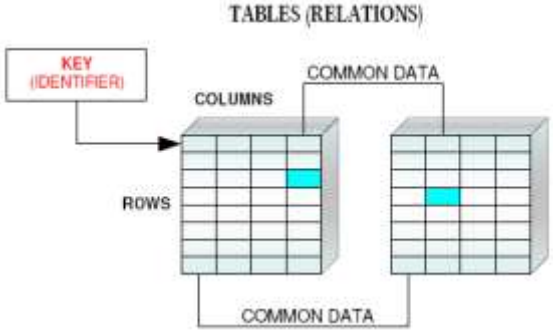
طريقة البحث وخطواته:

قمنا في هذا البحث بإتباع الخطوات الآتية:

1. دراسة قواعد البيانات العلائقية.
2. دراسة مفهوم المنطق الضبابي.
3. استخدام المجموعات الضبابية في لغة SQL.
4. بناء قاعدة بيانات وتطبيق الاستعلام الضبابي عليها.
5. مقارنة النتائج مع طرق الاستعلام التقليدية.

1-قواعد البيانات:

تعرف قاعدة المعطيات بأنها " مجموعة من المعطيات المهيكلة غير المتكررة المسجلة على وسط تخزين يسمح بالوصول إليها من قبل عدة برامج تطبيقية" [1] .ظهر مفهوم قواعد البيانات العلائقية عام 1970 بوصفه طريقة لمعالجة الكميات الكبيرة من البيانات، حجر الأساس فيها هو الجدول (عدد ثابت من الأعمدة وعدد متغير من الصفوف) كل تقاطع لصف وعامود يمثل عنصر بيانات واحدد[4].



الشكل (1) الجداول في قاعدة البيانات العلائقية

روابط قاعدة البيانات الاصطلاحية Join: تربط الجداول من خلال عملية تدعى join وتحدد الصفوف تبعاً لمحتويات أعمدتها، تسمى الطريقة القياسية لخلق–والولوج–و التعامل مع قاعدة بيانات (لغة الاستعلامات البنيوية SQL).

لغة الاستعلامات البنيوية SQL: إن لغة SQL هي لغة تصريحية يمكن من خلالها إعطاء النتائج المتوقعة أو العملية بدون تفصيلات محددة حول كيفية إنجاز المهمة. إن الخطوات المطلوبة لتنفيذ تعليمة SQL تعالج بشكل واضح في قاعة البيانات كما توصف لغة SQL بأنها ليست لغة إجرائية؛ لأن اللغات الإجرائية عموماً تتطلب تفاصيل العمليات ليتم تحديدها مثل (فتح وإغلاق الجداول، تحميل والبحث في الفهارس، تفريغ البيانات من الذاكرة المؤقتة، نسخ البيانات إلى ملفات النظام). لذلك فإن SQL تعتبر ذات مستوى افتراضي أعلى من اللغات الإجرائية كونه لا يتم استخدام المستويات المنطقية الأدنى والعمليات الفيزيائية حيث يتم استخدامها من خلال محرر SQL أو مخدم العمليات الذي ينفذها[14].

تعطى التعليمات على شكل أوامر تتألف من تعليمة SQL محددة وبارامتراتها، فضلاً عن معاملات تضاف إلى الأمر. تدعم بعض قواعد البيانات "الإجراءات المخزنة stored procedures" حيث تكون SQL مدمجة مع لغة إجرائية لتتعامل مع البيانات أو الجداول على المخدم من أجل دعم عمليات أكثر تعقيداً[14].

تصنف أوامر SQL بشكل عام في المجموعات الآتية[14][1]:

1–(Data Definition Language (DDL لغة تعريف البيانات:

تضم أوامر لإنشاء – تعديل –محي – إظهار– إخفاء جدول أو عنصر أخر في قواعد البيانات.

2–(Data Control Language (DCL لغة التحكم بالبيانات:

تضم أوامر لتحديد مستويات الصلاحية.

3–(Data Manipulation Language (DML لغة التعامل مع البيانات:

تضم أوامر لإنشاء – تعديل – حذف السجلات.

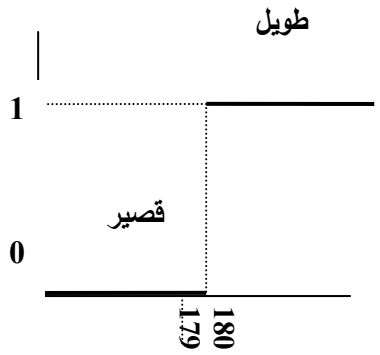
4–(Data Query Language (DQL لغة استعلام البيانات:

تضم أمر لتحديد السجلات من جدول أو أكثر

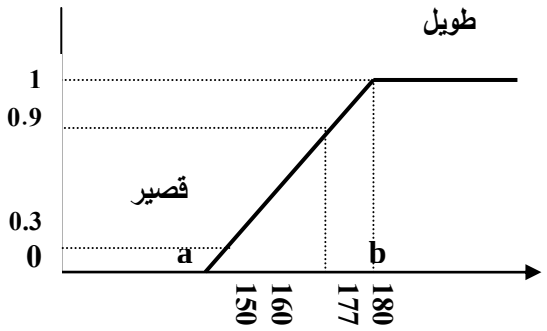
2– المنطق الضبابي:

إن مسألة انتماء العنصر إلى مجموعة ما أم عدمه هي مسألة هامة جداً في المنطق ثنائي القيمة ويمثل ذلك عادة على شكل جدول حقيقة بسيط يأخذ القيم "نعم" أو "لا"[9]. حصل تبدل لهذا المفهوم منذ ثلاثينيات القرن المنصرم تحديداً من قبل لوكاسيوكزس الذي وضع فرضية المنطق ثلاثي القيم. لاحقا في الستينيات عرف الدكتور لطفي ذاده ذلك بالمنطق الضبابي Fuzzy Logic [3].

يدرس المنطق الضبابي مقدار انتماء العناصر إلى مجموعة ما ويعبر عن النتائج بتابع درجة العضوية الذي يملك قيم حقيقية ضمن المجال [0,1]



الشكل (3)
تمثيل طول الإنسان في المنطق التقليدي



الشكل (2)
تمثيل طول الإنسان في المنطق الضبابي

لنأخذ كمثال لذلك طول الإنسان فإذا سلمنا أن الشخص الذي طوله 180 سم فما فوق يمثل شخصاً طويلاً فإن من طول 177سم وفق المنطق الثنائي التقليد يصنف بوصفه شخصاً قصيراً ونكتب :

$$\mu_{Tall} = \{height >= 180\}$$ **العلاقة (1)**

هذا الأمر يناقض طريقة التفكير البشرية، فأي شخص بالكاد يمكنه عملياً التمييز بين هاتين القيمتين للطول وبالتالي فإن أفضل طريقة للتعبير عن ذلك هو استخدام المفهوم الضبابي. كما هو موضح بالشكل(2) حيث يصنف شخص بطول 177 سم على أنه 0.9 طويل وشخص 160سم 0.3 طويل.

$$A(X)= \begin{cases} 0 & \text{if } x \le a \\ (x-a)/(b-a) & \text{if } a < x \le b \\ 1 & \text{if } x > b \end{cases}$$ **العلاقة (2)**

حيث a,b محددات تعريف الطول

3-مفهوم SQL الضبابي:

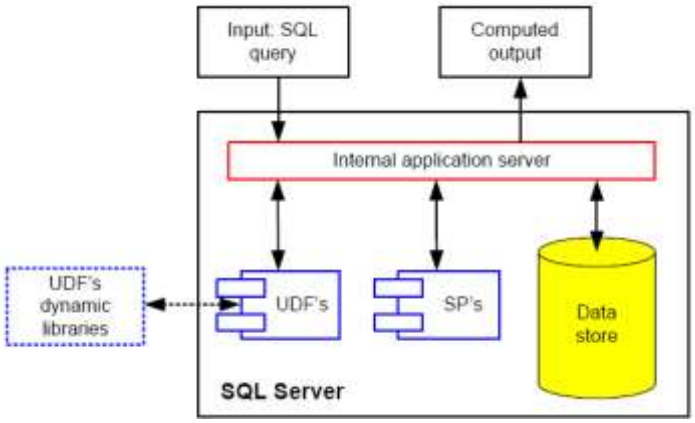
إن القاعد الرئيسية التي بنيت عليها لغة الاستعلام البنيوية SQL هي محاكاة طرق الإنسان الطبيعية في التعبير، بشكل عام فإن طريقة التعبير تتألف من شكل نحوى syntax form ودلالاته semantics. إن الشكل النحوي لعبارة SQL مكتملة بشكل كاف في معيار SQL الأساسي وله بنية مشابهة لعبارات الإنسان الطبيعية.و بالرغم من أن لغة SQL القياسية لا تحتوي على أي آلية أصلية لمعالجة العبارات الضبابية فإنه توجد طرق لتصميم وتطبيق البنى والطرق الجديدة تسمح بمعالجة العبارات الضبابية لكن اختلاف الطرق بين الشركات التي تقوم بتطوير لغة SQL يجعل Source Code غير قابل للنقل بشكل كلي بين نسخة وأخرى من لغة الاستعلام لكن الطرق المتبعة يمكن أن تطبق على قواعد البيانات بشكل عام حيث يمكن تطبيق العبارات الضبابية على الأغلب مع شروط

SQL.ويعتمد شكلها على طريقة المعالجة المختارة[16]. يمكن أن تعرف الشروط كعبارة تحصل على القيم TRUE أو NOT أو NULL .

يمكن استخدام العبارات الآتية للشرط في SQL:

- CASE conditional statement
- WHERE clause
- HAVING clause
- JOIN specification
- UPDATE command
- DLETE command
- IF, CASE, WHILE, REPEAT commands evaluation

إن تصاريح عبارات SQL الضبابية تعتمد على استخدام الوظائف المحددة من قبل المستخدم (UDF) User Defined Functions والإجراءات المخزنة Stored Procedures (SP)، إن كلا (UDF, SP) عبارة عن جزء متكامل من لغة تعريف البيانات Data Definition Language (DDL) والتي تشتمل مجموعة فرعية في SQL وعبارات SQL الضبابية لها نفس شكل تابع درجة العضوية، لهذا الأمر أثر أساسي في استخدام المعاملات القياسية (=، <، >، <=، >=، not) عندما تستخدم لمقارنة تابع درجة العضوية الخاص بها[15]



الشكل (4) بنية مخدم SQL

إن تصاريح المجموعات الضبابية في UDF لها مرحلتان. أولاً كتابة التابع المناسب بلغة Delphi أو أي لغة برمجة وترجمته. المرحلة الثانية تمثل تصريح التابع المناسب في SQL بتحديد اسم التابع والمكتبة الموجود بها متبوعة ببارمترات الدخل والخرج.

4-بناء قاعدة بيانات وتطبيق الاستعلام الضبابي عليها:

لنأخذ الجدول (**1**) كمثال عن قاعدة بيانات للدراسة:

الجدول (1) قاعدة بيانات افتراضية

	Age	Weight	Height	Name
	38	77	157	محمد
	52	98	184	بسام

سمير	181	71	25	
حسان	183	85	40	
قاسم	164	74	48	
زياد	178	91	44	

4-1التعابير الضبابية والبيانات الضبابية:

إن المجموعات الضبابية لا تمثل البيانات الضبابية، لكنها طريقة لترميز واستخدام معلومات(غير معرفة حدودها بشكل واضح). تربط المجوعة الضبابية كل قيمة من البيانات مع درجة عضويتها في مجموعة مسبقة التعريف.

لنأخذ المجموعة الضبابية التي تمثل الطول ونعبر عنها من خلال تعيين توابع العضوية للبيانات، إن كبر قيمتها تعكس مقدار مطابقة الدلالة أو المعنى للعنصر. باستخدام هذه الأداء يمكن إيجاد الزبائن الطويلة ضبابياً كما يأتي:

الاستعلام /2/ Crisp

Select *name* ***form*** *ieh.customer*
Where *customer.height >=180*

الاستعلام /1/ Fuzzy

Select *name* ***form*** *ieh.customer*
Where *customer.height is Tall*

كل قيمة في customer.height تحدد من المجموعة الضبابية Tall وترجع قيمة درجة عضويتها للمجموعة.

فإذا عرفنا μ(Tall) كما في الشكل (5) ونكتب رياضياً:

العلاقة (3)

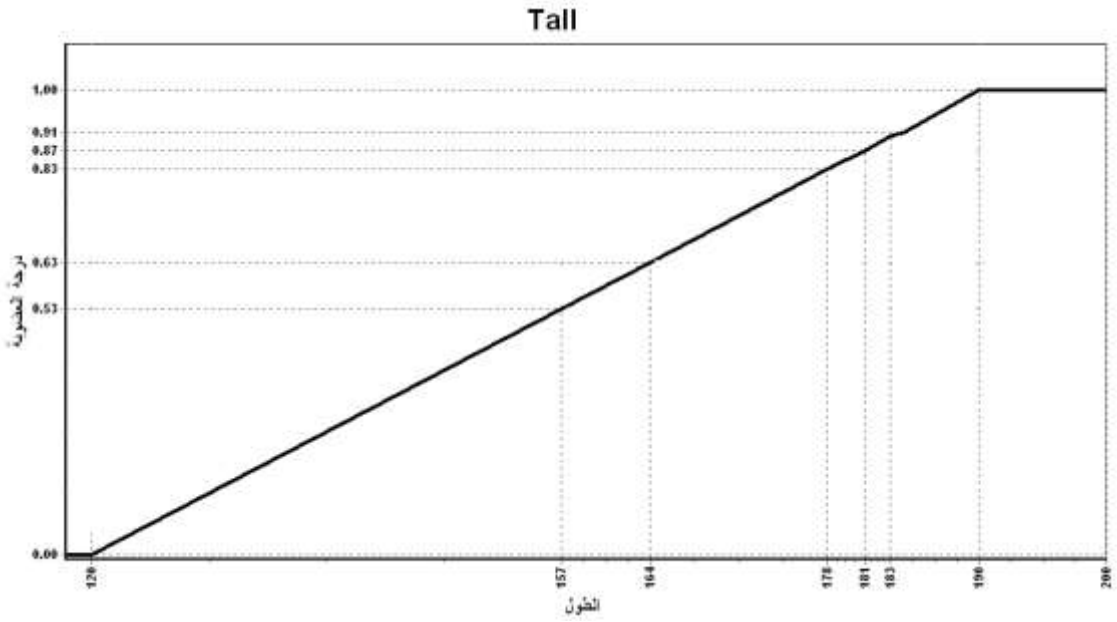
$$\mu(\text{Tall})=\begin{cases} 1 & \text{if } x \leq 190 \\ (\text{Tall}-120)/70 & \text{if } 120< x \leq 190 \\ 0 & \text{if } x > 120 \end{cases}$$

هذا النوع من الاستعلام يقيس درجة انتماء طول كل زبون إلى المجموعة الضبابية Tall ويستخدم لترتيب السجلات المحدد تبعاً لمطابقتها للاستعلام المطلوب. نتيجة الاستعلام تكون كما في الجدول (2):

الجدول (2) نتيجة الاستعلام /2/

القيم في العامود μ(Tall) تم الحصول عليها من تعويض قيم العامود Height في العلاقة (3)

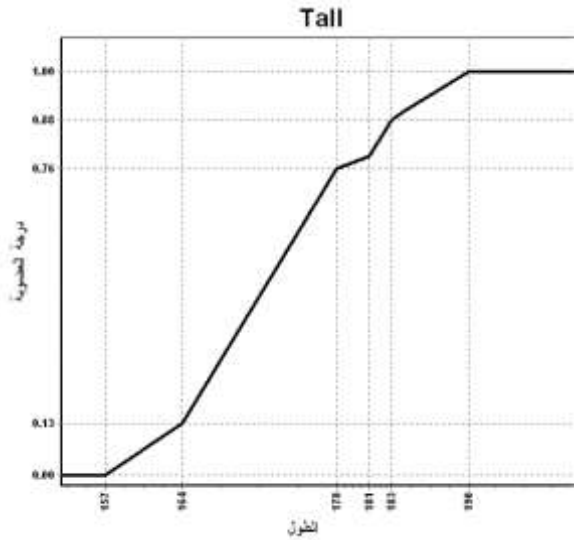
μ(Tall)	Age	Weight	Height	Name
0.91	52	98	184	بسام
0.90	40	85	183	حسان
0.87	25	71	181	سمير
0.83	44	91	178	زياد
0.63	48	74	164	قاسم
0.53	38	77	157	محمد



الشكل (5) تعريف الطول الضبابي

2-4 توليف المجموعات الضبابية مع الدلالات الاستعلامية:

يمكننا تعديل تعريف Tall بتعديل كل من مجال قيمه وشكل تابع العضوية، الشكل (6) يوضح إمكانية استخدام منحني مائل بدلا من المنحني الخطي.



الشكل (6)تعريف الطول الضبابي

في هذه الحالة فإن نتيجة الاستعلام السابق سوف تختلف لتصبح كما في الجدول (3):

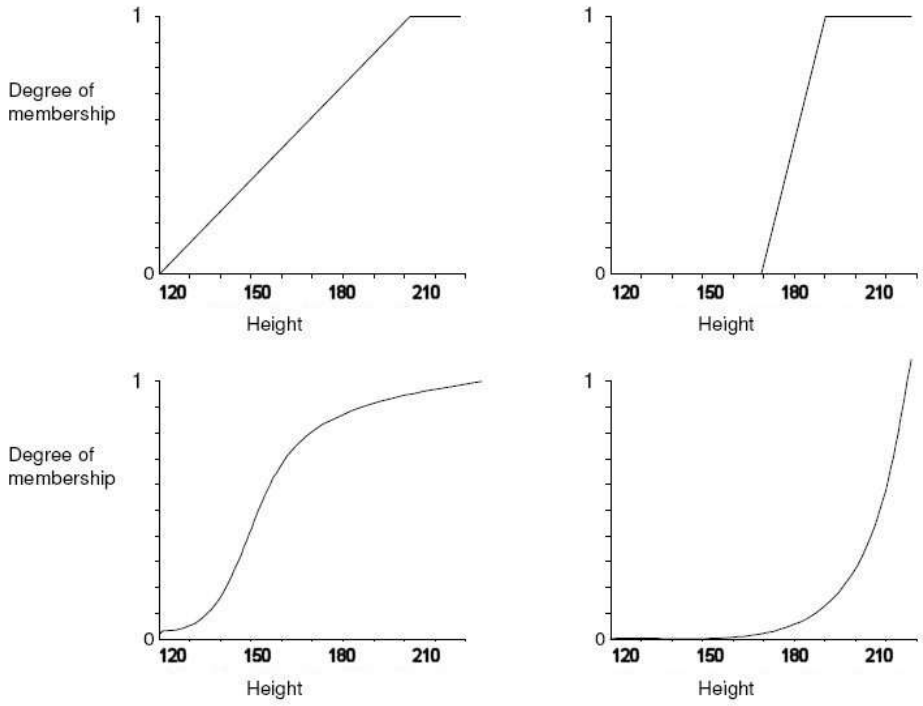
الجدول (3) نتيجة الاستعلام /2/ بعد تعديل التابع Tall

μ(Tall)	Age	Weight	Height	Name
0.90	52	98	184	بسام
0.88	40	85	183	حسان
0.79	25	71	181	سمير
0.76	44	91	178	زياد
0.13	48	74	164	قاسم
0.00	38	77	157	محمد

إن هذه المجموعة الجديدة لتوابع العضوية تقودنا إلى حقيقة مهمة جداً حول المجموعات الضبابية وهي، أن المجموعات الضبابية ليس وحيدة القيم عموماً فهي تعتمد على بياناتنا وعلى الطريقة التي نرى بها الأمر. إن شكل المجموعة الضبابية يمثل طريقة فهم الموضوع. وبالتالي فإن المجموعات الضبابية تزاوج بين دلالة طلبنا مع البنية الحقيقة للطلب.

كنتيجة لذلك يمكننا وضع استعلاماتنا في شكل مفهوم بالنسبة لنا كما هو مفهوم بالنسبة للحاسب. لكن يجب أولاً أن نخبر الحاسب ماذا نعني بتلك التعابير مثل Tall. يعبر عن هذا الأمر على شكل تابع عضوية للمجموعة الضبابية. إن تغير تابع العضوية يغير ما نعنيه عند تنفيذ الاستعلام.

الشكل (7) يشرح هذه الفكرة بعرض أربع نماذج مقبولة ومتباينة للمجموعة الضبابية Tall



الشكل (7) (أشكال مختلفة للمجموعة الضبابية Tall)

عند تصميم مجموعة ضبابية لنظام SQL ضبابي يجب مراعاة مقدار البيانات في الجدول التي تتعلق بدلالة الفكرة المراد الاستعلام عنها. ومن المهم معرفة أن هذا التخطيط لا يحتاج بالضرورة أن يكون دقيق مئة بالمئة.

4-3 طبيعة استعلامات SQL الضبابية:

إن ميزات الاستعلام الضبابي تبدو جلية عندما نستخدم أكثر من مجموعة ضبابية في الاستعلام. باستخدام مجموعات ضبابية متعددة يمكننا إيجاد البيانات المرشحة بدرجة ما والتي تملك خصائص الأفكار مجتمعة. لنأخذ استعلام للأطوال والأوزان. في الاستعلام التقليدي يجب وضع نقط حدية لتعريف الثقيل Heavy لنفترض أخذنا 88 كغ فيصبح الاستعلام :

الاستعلام /3/

```
Select name form ieh.customer
Where customer.height  >=180
and customert.weight  >=88
```

إن نتيجة هذا الاستعلام تكون عبارة عن سجل واحد يحقق الشرطين الجدول(4)

الجدول (4) نتيجة الاستعلام (3)

(Tall and Heavy)	Age	Weight	Height	Name
	38	77	157	محمد
⇨	52	98	184	بسام
	25	71	181	سمير
	40	85	183	حسان
	48	74	164	قاسم
	44	91	178	زياد

لكن إذا نظرنا للجدول نلاحظ بأنه وجود عدة زبائن يمكن عدها محقق للشرطين، لكن تم استبعادهم بسبب النقاط الحدية من قبل الاستعلام SQL التقليدي، لتحسن الاستعلام نستبدل الاستعلام بنسخة ضبابية مناسبة له:

الاستعلام /4/

```
Select name form ieh.customer
Where customer.height is Tall
and customert.weight  is Heavy
```

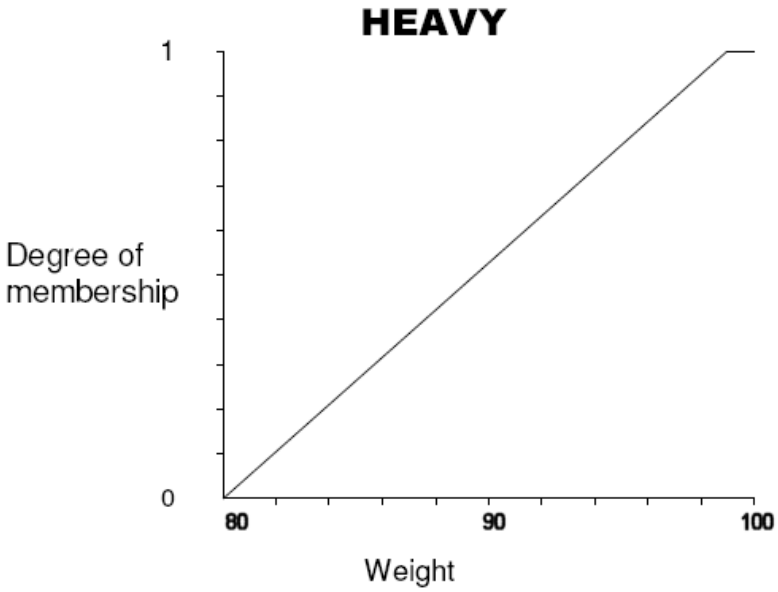
إن دخل هذا الاستعلام واضح للعيان (المطلوب الزبائن الطويلين وذوي الوزن الكبير). لتحديد نتيجة الاستعلام فإننا نحتاج أن نخبر الكمبيوتر ما يجعل الشخص ذو وزن كبير Heavy. يبين الشكل (8) أحدى الطرق لتمثيل هذه الفكرة.

وبالتالي عندما نأخذ قيمة من عامود *customert.weight* يمكننا إيجاد عضويتها في المجموعة الضبابية Heavy. يبين الجدول (5) درجة عضوية الأوزان في المجموعة الضبابية Heavy مرتبة حسب درجة عضويتها.

الجدول (5) درجة العضوية لكل وزن

μ(Tall)	μ(Heavy)	Age	Weight	Height	Name
0.91	0.95	52	98	184	بسام
0.83	0.55	44	91	178	زياد
0.90	0.25	40	85	183	حسان
0.87	0.00	25	71	181	سمير
0.63	0.00	48	74	164	قاسم

0.53	0.00	38	77	157	محمد



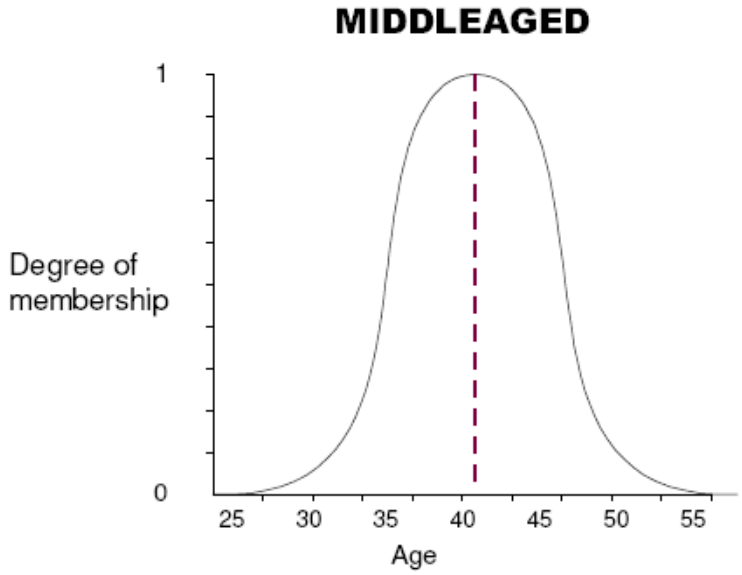
الشكل (8) المجموعة الضبابية لـ Heavy

باستخدام الاستعلام (4) نحصل على ثلاثة زبائن (بسام–زياد–حسان) لها عضوية في المجموعتين الضبابيتين Tall وHeavy وباستخدام العضوية الصغرى Membership Minimum أو معدل العضوية Membership Average يمكننا دمج توابع العضوية كما في الجدول (6):

الجدول (6) نتيجة الاستعلام /4/ باستخدام العضوية الصغرى ومعدل العضوية

Avr(Heavy,Tall)	Min(Heavy,Tall)	μ(Tall)	μ(Heavy)	Age	Weight	Height	Name
0.930	0.91	0.91	0.95	52	98	189	بسام
0.690	0.55	0.83	0.55	44	91	155	زياد
0.575	0.25	0.90	0.25	40	85	186	حسان
0.435	0.00	0.87	0.00	25	71	156	سمير
0.315	0.00	0.63	0.00	48	74	167	قاسم
0.265	0.00	0.53	0.00	38	77	160	محمد

يمكننا توسع الاستعلام ليضم مثالاً العمر، فإذا أردنا في الاستعلام الأشخاص متوسطي العمر نعرف MiddleAged كما في الشكل(9):



الشكل (9) تمثيل تابع العمر الضبابي

ونكتب الاستعلام:

الاستعلام /5/

Select *name* ***form*** *ieh.customer*
Where *customer.height* **is** *Tall*
and *customert.weight* **is** *Heavy*
and *customert.age* **is** *Middleage*

فنحصل على النتيجة التي تظهر في الجدول (7):

الجدول (7) نتيجة الاستعلام /5/

Avr	Min	μ(Tall)	μ(Heavy)	μ(M A)	Age	Weight	Height	Name
0.760	0.55	0.83	0.55	0.90	44	91	155	زياد
0.716	0.25	0.90	0.25	1.00	40	85	186	حسان
0.660	0.12	0.91	0.95	0.12	52	98	189	بسام
0.500	0.00	0.53	0.00	0.97	38	77	160	محمد
0.443	0.00	0.63	0.00	0.70	48	74	167	قاسم
0.323	0.00	0.87	0.00	0.10	25	71	156	سمير

5- مقارنة النتائج مع طرق الاستعلام التقليدية:

لقد قمنا بتطبيق نموذج الاستعلام الضبابي على قاعدة بيانات عملية صممت من قبلنا (سجل فواتير الهواتف) فقمنا ببناء نماذج استعلام وفق محددات معينة كانت الغاية منها الحصول على أرقام الهواتف والعناوين لمشتركي الهاتف وفق مواصفات محددة.

قمنا بتعرف ثلاثة معايير لقيم المكالمات:

LowValue المبالغ صغيرة القيمة وعرفناها كالآتي:

$$\text{LowValue}(X)=\begin{cases}1 & \text{if } x \le 700\\ (X-1500)/800 & \text{if } 700< x \le 1500\\ 0 & \text{if } x > 1500\end{cases} \qquad \textbf{العلاقة (4)}$$

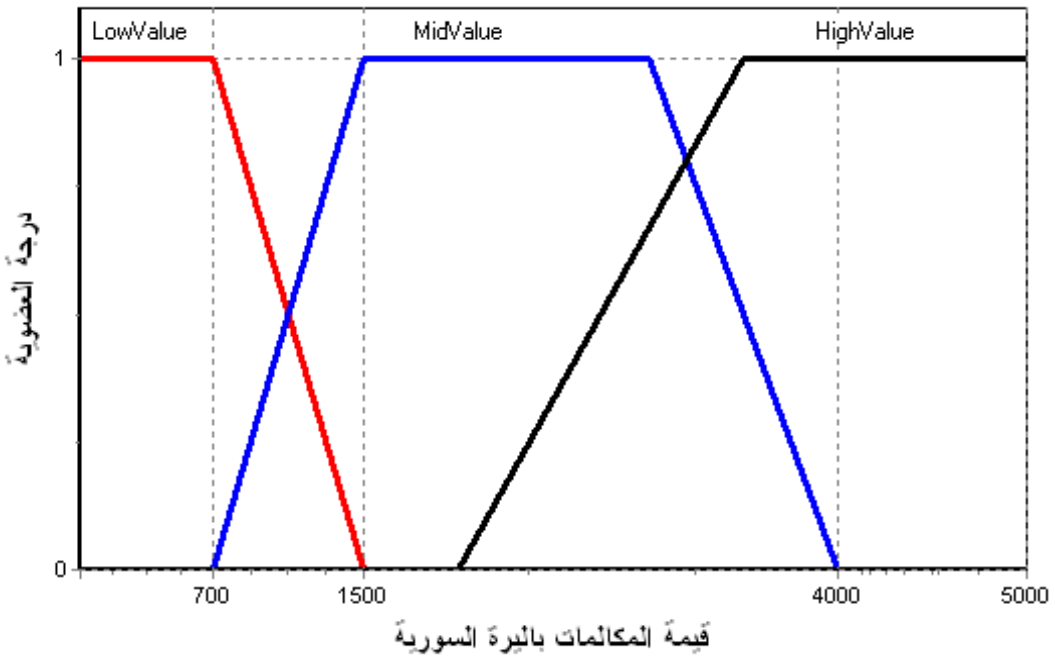
MidValue المبالغ متوسطة القيمة وعرفناها كالآتي:

$$\text{MidValue}(X)=\begin{cases}0 & \text{if } x \le 700\\ (X-700)/800 & \text{if } 700< x \le 1500\\ 1 & \text{if } 3000>x > 1500\\ 0 & \text{if } x > 3000\end{cases} \qquad \textbf{العلاقة (5)}$$

HighValue القيم العالية القيمة وعرفناها كالآتي:

$$\text{HighValue}(X)=\begin{cases}0 & \text{if } x \le 2000\\ (X-2000)/1500 & \text{if } 2000< x \le 3500\\ 1 & \text{if } x >3500\end{cases} \qquad \textbf{العلاقة (6)}$$

تلك العلاقات نعبر عنها بيانياً بالشكل (10)



الشكل (10) توابع العضوية HighValue- MidValue-LowValue

ويكتب الاستعلام كما يلي:

الاستعلام /6/

```
Select * form PhoneBills
  Where Bills is HighValue
```

خوارزمية تنفيذ الاستعلام:

- قياس درجة انتماء كل قيم للحقل Bills إلى المجموعة الضبابية HighValue
- ترتيب السجلات بحسب درجة انتمائها.
- عرض النتائج التي درجة انتمائها لمجموعة الضبابية HighValue أكبر من الصفر.

إذا تم استخدام شروط وعلاقات مناسبة لا يمكن الوصول إلى النتيجة نفسها في المنطق الكلاسيكي كون الشرط فيه حدي (صح أو خطأ).

إن التعريف السابق يتيح لنا إمكانية إجراء استعلامات مختلفة تبعاً للغرض المطلوب، لنأخذ مثلا "الإعلانات الموجهة" فلإرسال إعلان عن رحلات طيران فإن الفئة التي تهتم لمثل تلك الإعلانات هي الفئة التي لها ارتباطات دولية فننكب الاستعلام /7/ :

الاستعلام /7/

```
Select PhoneNumber,Name,Address, StateCallValue form PhoneBills
Where StateCallValue is ( HighValue or  MidValue)
```

تكون نتيجة هذا الاستعلام مجموعة مشتركي الهاتف التي قيمة المكالمات الدولية لفواتيرها الهاتفية متوسطة أو عالية.

أما لإرسال إعلان يخص مستخدمي الانترنت فنكتب:

الاستعلام /8/

```
Select PhoneNumber,Name,Address, InternetValue form PhoneBills
Where InternetValue  is not (LowValue)
```

كمقارنة بسيطة قمنا بمقارنة عدد السجلات الناتجة عن الاستعلام التقليدي مع الاستعلام الضبابي فكانت النتائج كما هو مبين في الجدول (7) :

الجدول (7) مقارنة بين الاستعلام الضبابي والاستعلام التقليدي

نوع الاستعلام	Crisp SQL	Fuzzy SQL
	Select PhoneNumber,Name,Address, InternetValue **form** PhoneBills **Where** InternetValue >=1500	**Select** PhoneNumber,Name,Address, InternetValue **form** PhoneBills **Where** InternetValue **is not** (LowValue)
عدد السجلات	287	941

من خلال ذلك نلاحظ بأن الاستعلام التقليدي Crisp قد تجاهل عدد كبير من السجلات كانت ستعتبر مقبولة لو أن العامل البشري قام بتدقيقها كونها توافق الشرط المطلوب.

يمكن التوسع باستخدام تلك الاستعلامات الضبابية لأغراض متنوعة كتحديد أوقات الذروة ودراسة تقديم تخفيضات في الأسعار خارجها لما لها من مرونة في الاستخدام وكون صياغتها اللغوية قريبة جداً من اللغة المحكية.

النتائج :

من خلال البحث قمنا بتصميم قاعدة بيانات علائقية وبنينا استعلامات تعتمد تقانات المنطق الضبابي في لغة الاستعلامات البنيوية SQL وتم تنفيذها بالزمن الحقيقي على بيانات الفواتير الهاتفية فحصلنا على نتائج أقرب بكثير إلى الواقع مما هو حال النتائج في حال استخدام لغة الاستعلام البنيوية التي تعتمد المنطق ثنائي القيمة، لذلك ينصح باستخدام المنطق الضبابي في لغة الاستعلام البنيوية لمحاكاة الواقع الفعلي بشكل أفضل.

المراجع :

1- رزوق، راكان؛ عبود، مادلين. *قواعد المعطيات*.الطبعة الرابعة، جامعة دمشق، سورية، 2004، 216.

2- GALINDO, J.; URRUTIA ,A.; PIATTINI,P. *Fuzzy Databases: Modeling, Design and implementation*. 1ST ed, Idea.Group.Publishing, United States of America, 2006, 320.

3- KLIR, G.; YUAN, B. *Fuzzy Sets AND Fuzzy logic theory and applications*. 1ST ed Prentice Hall, United States of America, 1995, 350.

4- NAYYERI, A.; OROUMCHIAN, F. *FuFaIR: a Fuzzy Farsi Information Retrieval System*. Computer Systems and Applications, 2006. IEEE International Conference on.Volume, Issue, March 8, 2006 Page(s): 1126 – 1130

5- DATTATRI, S.; JOY, K. *Implementing a Fuzzy Relational Database and Querying System With Community Defined Membership Values*,November 2004.

6- DATE, C. J. *"A Note on Relation-Valued Attributes" In An Introduction to Database Systems*. 8th ed, Pearson/Addison Wesley, Boston, 2004, 373-375.

7- WEB. Wikipedia, the free encyclopedia ,*Fuzzy logic*. 12/9/2007. <http://en.wikipedia.org/wiki/Fuzzy_logic>

8- CRABTREE, P. *Fuzzy Relational Database and Querying System With Compound Query Capabilities* A Directed Research Project, 2005,47**.**