

一個具有模糊查詢能力之物件導向資料庫查詢語言(EQLf)的設計與製作

The Design and Implementation of an Object-Oriented Database Query Language (EQLf) for Fuzzy Querying

計畫編號 : NSC87-2213-E009-001

國立交通大學資訊科學系及研究所

計畫主持人：楊維邦

摘要

有鑑於資訊科技應用的普及，使用者與電腦間的溝通介面也越具親和性。對於資料庫的使用而言，查詢語言扮演著使用者與資料庫間最重要的溝通工具。為了能夠讓資料庫查詢語言更具有親和性，讓使用者能夠更輕易方便的使用資料庫，一個重要的關鍵就是擴充資料庫查詢語言的功能，讓它具有接受與解釋不精確查詢條件的能力。

本計畫的目的在設計與製作一個在物件導向資料庫上具有描述模糊及不精確查詢條件之查詢語言，並希望透過以下方法來完成這個目標：一、分析與討論關於處理不精確查詢的相關研究，並比較他們之間的優缺點；二、擴充一個現有的物件導向資料庫查詢語言 EQL，讓它具有描述不精確查詢的功能，並且對於每一種不精確的查詢提出其解釋方式；三、採用物件導向軟體工程之技術並配合視窗介面之程式設計來製作這個在使用和學習上親和性都極佳之模糊查詢語言。我們將這個查詢語言命名為 **EQLf (An Easy-to-Use Query Language with Fuzzy Capability for Object-Oriented Databases)**。

Abstract

With the popularity of applied information technology, the user interface that acts the role for communications between the users and the computer has been designed to be more and more friendly. For users who want to access a database, the query language fits as the most important user interface. To make the query language more user-friendly, an important issue arose is to extend the query language with the functionality to accept and explain imprecise queries.

The objective of this project is to design and implement an object-oriented database query language for fuzzy querying. We achieve this objective through the following steps: 1) Analyze and discuss researches on processing of imprecise queries and compare the advantages and disadvantages among them; 2) Extend an existing query language, namely EQL, to incorporate the capabilities of expressing imprecise predicates and making imprecise queries; 3) Implement the complete query language to validate its excellence in being both

user-friendly and easy to learn by adopting object-oriented engineering technology and window programming paradigm. We call this query language EQLf (An Easy-to-Use Query Language with Fuzzy Capability for Object-Oriented Databases).

第一章 簡介

在資料庫的領域中，查詢語言扮演著使用者與資料庫間溝通的橋樑，但是傳統的關連式資料庫或正在發展的物件導向資料庫中，由於其解釋所需要資料的方式是使用布林邏輯，因此不是造成查詢結果太過嚴苛，就是無法運用人們的思考模式在現有的資料庫系統上，使得資料庫無法接受人們日常生活中最常使用的不精確查詢。對於查詢中不精確性的處理，已有許多相關的研究，並且被實作在一些實驗的雛形上，這些研究根據解釋不精確條件的方式，可以分為相對式和絕對式兩種方法。

一、相對式不精確(Relative Imprecision)的解釋法

所謂的相對式不精確就是不給予每個成員一個對不精確條件的評估值，而是給每個成員一個在所有成員中的排名，如果一個成員比其它成員更滿足此不精確條件，則此成員的排名就會比其它成員高，否則就比其它成員低。對於不精確條件所採取的相對式解釋方法在實作上是很容易的，因為它的理論基礎是建立在成員的排序上，但查詢的結果卻與我們的直覺並不相符。另一個缺點就是每個模糊術語都必須是某一個成員屬性的單調函數，因此模糊條件的廣泛性就會受限制。

二、絕對式不精確(Absolute Imprecision)的解釋法

所謂的絕對式不精確(Absolute Imprecision)，就是給予每個成員一個直接的評估值，來表示此成員對不精確條件的符合度。根據使用方法的的不同，可分為兩類，一為以明確集合(Crisp Sets)為基礎的研究，另一類為以模糊集合(Fuzzy Sets)為基礎的研究。

從以上的討論可以對這些方法歸納出它們的優缺點，我們以四種層面來分析比較這些方法的優劣點，星號個數越多表示該項的程度越大，最後的結果如表一所示：

	Relative	Absolute Interpretation
--	----------	-------------------------

	Interpretation	Crisp Set Based		Fuzzy Set Based
	Deduce2	ARES	Vague	EQL, SQLf
Extra Knowledge	*	**	**	***
Integration	**	*	*	***
Intuition	*	**	***	****
Implementation Effort	*	**	**	***

表一：不精確查詢的解釋方法比較表

① Extra Knowledge :

表此種方法在傳統資料庫上須多加多少額外的資訊才能解釋它能表示的不精確的查詢。

② Integration :

表此種方法對不精確條件表達的完整性，是否所有不精確條件都可以表達和解釋。

③ Intuition :

表此種方法查詢後的結果是否與我們的直覺相符。

④ Implementation Effort :

表此種方法在實作上的難易度。

本計畫即根據以上初步的比較結果，選擇在完整性和直覺性都較佳的方法-以模糊集合為基礎的解釋方法，來設計及製作一個具有模糊查詢能力之物件導向資料庫查詢語言 EQLf。

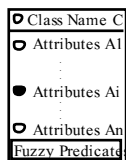
第二章 EQLf 之設計

EQLf 是由 EQL 語言擴充而來，字尾的 f 代表模糊(Fuzzy)之意，這並不是表示 EQLf 是一個定義不清楚、語法模糊的語言，它想要表達的是，除了 EQL 原有的功能外，EQLf 還能清楚的描述不精確條件，並且加以處理。

以下我們針對 EQLf 對 EQL 所做的擴充部份加以說明，除了對語法的解釋外，更重要的是對於各種不同的模糊查詢，我們也初步地解釋了它們的語意，也就是定義出所對應的匹配函數。

(1) 模糊選擇(Fuzzy Selection)

在 EQL 中擷取的條件只能是傳統的布林運算(Boolean Expression)，因此我們可以藉由引入模糊條件，而構成所謂的模糊選擇。這是最簡單的一種模糊查詢，其結構為：



在 EQLf 中每一個類別都由一個頂點圖示(Vertex Icon)加以表示，其中顯示了此類別和屬性的名稱以及對此類別所要下達的擷取條件。屬性前黑色的元球符號“●”表示所要投影(Projection)的屬性。此結構所代表的意義就是對類別 C 中的物件，根據條件計算它們的歸屬度，並將它們投影到屬性 Ai 上。

EQLf 中每個查詢在執行前都可以由使用者指定最後的結果要如何產生，指定參數的形式為 [n | t | n, t]，n

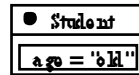
表示個數，也就是只要看前 n 個最滿足的物件，t 表示介於 0 到 1 之間的臨界值，也就是只要看歸屬度大於 t 的物件。

頂點中所指定的 Fuzzy Predicates 可以同時是模糊條件和布林條件的組合，模糊條件的部份可以簡單模糊條件、修正模糊條件、組合模糊條件以及二元模糊關係。在查詢的結果中，對於屬於 C_f 的 a 值歸屬度可以根據方程式(1)所列的匹配函數計算而得。

$$\mu_{C_f}(a) = \sup_{o \in C \text{ and } o.A_i = a} \mu_{\text{Fuzzy Predicates}}(o)$$

上式為方程式 (1)，其中 O 表示類別 C 中的物件，o.A_i 表示物件 o 在屬性 A_i 上的投影，a ∈ C.A_i，C_f 表示模糊類別(Fuzzy Class)，由於所有物件對屬性 A_i 投影後的結果可能會產生重複值，Sup (Supreme)的意義就是對這些重複值取最小上界值。接下來我們舉一些相關的例子，並根據方程式(1)來計算每個物件的歸屬度。

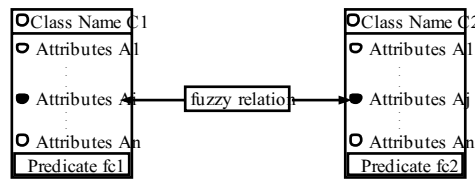
FQ1) 找出年紀大的學生資料。



在這個例子中，模糊的條件為「年紀大」，因此我們將每個物件的年齡帶入老的歸屬函數，並且加以計算，就可以得出物件的歸屬度為 $\mu_{\text{Student}}(o) = \mu_{\text{old}}(\text{age})$ 。當然「老」的歸屬函數必須由使用者或系統管理者自行定義。

(2) 模糊合併(Fuzzy Join)

在 EQL 中，合併是用來描述兩頂點間的限制，使用的符號為“ $\leftarrow \bowtie \rightarrow$ ”，這個符號用來連結相等的兩屬性，表示相等合併(equiv-join)。EQLf 針對 EQL 中之相等合併作一擴充，將原來的符號改為“ $\leftarrow \text{fuzzy relation} \rightarrow$ ”，表示模糊合併，因此查詢的結構為：



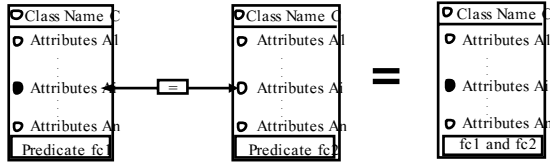
模糊合併的兩頂點分別為一個模糊選擇，fuzzy relation 可以為任何的二元明確關係或模糊關係，例如：“=”表示相等合併，“>>”表示「大很多」模糊關係，“≈”表示「類似於」模糊關係等。在這個查詢的結果中，對於 C_f 中的(a, b)值，其歸屬度可以根據方程式(2)所列的匹配函數加以計算而得。

$$\mu_{C_f}(a, b) = \sup_{\substack{o_1 \in C_1 \text{ and } o_1.A_i = a \text{ and} \\ o_2 \in C_2 \text{ and } o_2.A_j = b}} \min(\mu_{fc1}(o_1), \mu_{fc2}(o_2), \mu_{\text{fuzzy relation}}(o_1.A_i, o_2.A_j))$$

方程式 (2)

其中 o_{c1} 表示類別 C1 中的物件，o_{c2} 表示類別 C2 中的物件，C_f 表示模糊類別(Fuzzy Class)。有一點值得我們注意的是，當類別 C1 等於 C2，A_i 等於 A_j，而且為

關鍵屬性，以及模糊關係為“=”時，模糊合併可以用模糊選擇加以代替，亦即



此時我們就必須考慮查詢一致性(Query Equivalence)，也就是說物件經由左邊模糊合併算出的歸屬度要等於右邊模糊查詢所算出的歸屬度，這樣所定義出的匹配函數才是合理的。

首先我們來檢視左邊的匹配函數為：

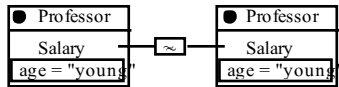
$$\mu_{C_j}(a) = \text{Sup}_{\substack{o_1 \in C_1 \text{ and } A_i = a \\ o_2 \in C_2 \text{ and } A_j = a}} \min(\mu_{fc_1}(o_1), \mu_{fc_2}(o_2), \mu_{=}(o_1.A_i, o_2.A_j))$$

而右邊的匹配函數為：

$$\mu_{C_j}(a) = \text{Sup}_{o \in C \text{ and } o.A_i = a} \min(\mu_{fc_1}(o), \mu_{fc_2}(o))$$

我們可以發現這兩個匹配函數是一樣的，因此滿足查詢一致性。接下來看個例子：

FQ2) 找出年輕且薪水差不多的教授資料

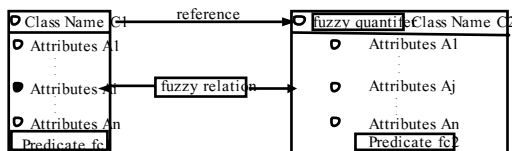


在這個例子中，合併的是兩個模糊選擇，合併的方式是對教授的薪水做模糊合併，合併的關係為「差不多」，因此我們可以根據公式(2)，可計算合併後的物件歸屬度：

$$\mu_{C_j}(o, o_j) = \text{Sup}_{o_1, o_2 \in \text{Professor}} \min(\mu_{\text{young}}(o, \text{age}), \mu_{\text{young}}(o, \text{age}), \mu_{\approx}(o, \text{salary}_o, \text{salary}_j))$$

(3) 具有模糊量詞(Fuzzy Quantifier)的查詢

在 EQL 中，多值屬性(Multivalue)和一對多(One-to-many)關係的內定處理方式是用存在性(Existential)量詞加以處理，如果使用者考慮的是全域性(Universal)量詞，就必須在查詢的類別名稱前指定全域量詞符號“ \forall ”。因此我們對 EQL 中之存在性和全域性量詞作一擴充，允許在查詢中使用模糊量詞，查詢的結構如下：



根據使用者對此查詢的使用方式，歸屬度的計算可以分為五種情形。

1. 沒有模糊關係且模糊量詞是絕對性的。

匹配函數定義為：

$$\mu_{C_j}(a) = \text{Sup}_{o \in C_1 \text{ and } o.A_i = a} \min(\mu_{fc_1}(o), \mu_{\text{fuzzy quantifier}}(\sum_{i=1}^n \mu_{fc_2}(o.o_{r_i})))$$

方程式 (3)

其中 o 屬於 C_1 之物件， $o.A_i$ 表示物件 o 在屬性 A_i 上的投影， $a \in C_1.A_i$ ， C_j 表示模糊類別(Fuzzy Class)， o_{r_i} 表示被 o 參考之 C_2 物件， n 表示被 o 參考之物件個數。

2. 沒有模糊關係且模糊量詞為相對性的。

$$\mu_{C_j}(a) = \text{Sup}_{o \in C_1 \text{ and } o.A_i = a} \min(\mu_{fc_1}(o), \mu_{\text{fuzzy quantifier}}(1/n \sum_{i=1}^n \mu_{fc_2}(o.o_{r_i})))$$

方程式 (4) 為其匹配函數，其中 o 屬於 C_1 之物件， $o.A_i$ 表示物件 o 在屬性 A_i 上的投影， $a \in C_1.A_i$ ， C_j 表示模糊類別(Fuzzy Class)， o_{r_i} 表示被 o 參考之 C_2 物件， n 表示被 o 參考之物件個數。

3. 沒有模糊條件 fc_2 且模糊量詞為絕對性的。匹配函數定義為：

$$\mu_{C_j}(a) = \text{Sup}_{o \in C_1 \text{ and } o.A_i = a} \min(\mu_{fc_1}(o), \mu_{\text{fuzzy quantifier}}(\sum_{i=1}^n \mu_{\text{fuzzy relation}}(o.A_i, o.o_{r_i}.A_j)))$$

上式為方程式 (5)，其中 o 屬於 C_1 之物件， $o.A_i$ 表示物件 o 在屬性 A_i 上的投影， $a \in C_1.A_i$ ， C_j 表示模糊類別(Fuzzy Class)， o_{r_i} 表示被 o 參考之 C_2 物件， $o.o_{r_i}.A_j$ 表示物件 $o.o_{r_i}$ 在屬性 A_j 上的投影， n 表示被 o 參考之物件個數。

4. 沒有模糊條件 fc_2 且模糊量詞為相對性的。匹配函數定義為：

$$\mu_{C_j}(a) = \text{Sup}_{o \in C_1 \text{ and } o.A_i = a} \min(\mu_{fc_1}(o), \mu_{\text{fuzzy quantifier}}(1/n \sum_{i=1}^n \mu_{\text{fuzzy relation}}(o.A_i, o.o_{r_i}.A_j)))$$

上式為方程式 (6)，其中 o 屬於 C_1 之物件， $o.A_i$ 表示物件 o 在屬性 A_i 上的投影， $a \in C_1.A_i$ ， C_j 表示模糊類別(Fuzzy Class)， o_{r_i} 表示被 o 參考之 C_2 物件， $o.o_{r_i}.A_j$ 表示物件 $o.o_{r_i}$ 在屬性 A_j 上的投影， n 表示被 o 參考之物件個數。

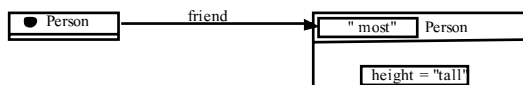
5. 具有模糊條件 fc_2 及模糊關係且模糊量詞為相對性的。匹配函數定義為：

$$\mu_{C_j}(a) = \text{Sup}_{o \in C_1 \text{ and } o.A_i = a} \min(\mu_{fc_1}(o), \mu_{\text{fuzzy quantifier}}(\frac{\sum_{i=1}^n \min(\mu_{\text{fuzzy relation}}(o.A_i, o.o_{r_i}.A_j), \mu_{fc_2}(o.o_{r_i}))}{\sum_{i=1}^n \mu_{fc_2}(o.o_{r_i})}))$$

方程式 (7)

其中 o 屬於 C_1 之物件， $o.A_i$ 表示物件 o 在屬性 A_i 上的投影， $a \in C_1.A_i$ ， C_j 表示模糊類別(Fuzzy Class)， o_{r_i} 表示被 o 參考之 C_2 物件， $o.o_{r_i}.A_j$ 表示物件 $o.o_{r_i}$ 在屬性 A_j 上的投影， n 表示被 o 參考之物件個數。接下來是一些相關的例子。

FQ3) 找出這個人的資料，條件為他大部份的朋友是身高很高的。



這個查詢中的模糊量詞為「大部份」，此查詢的結構屬於第二種查詢，所以根據公式(4)，物件的歸屬度為：

第三章 EQLf 系統之實作

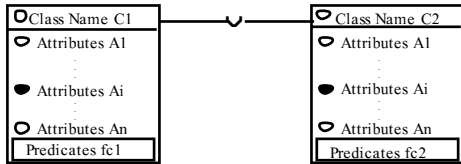
以下我們簡單說明 EQLf 系統各部份的主要功能：

1. EQLf 使用者介面(EQLf User Interface):這部份負責與使用者溝通,主要有四大功能:資料庫的連結與解除、查詢的建立、模糊語意的管理、查詢結果的展示。
2. 模糊管理器(Fuzzy Manager):此管理器負責儲存、修改使用者建立的模糊語意,並提供執行處理器查詢模糊語意的介面。
3. 查詢處理器(Query Processor):此處理器處理解析使用者所下達的查詢,並根據 EQLf 的設計決定每個查詢的匹配函數,並通知執行管理器匹配函數的種類。
4. 執行管理器(Execution Manager):根據查詢處理器解析後的結果,將資料庫的物件取出,並計算其歸屬度,然後將結果傳回給使用者介面。

$\mu_{Person_i}(o) = \mu_{max}(\prod_{j=1}^n \mu_{f_{j1}}(o, friend_j, height))$ 。「高」和「大部份」的歸屬函數需由使用者加以定義。

(4) 模糊聯集(Fuzzy Union)

將前面介紹的模糊查詢利用聯集圖示結合後就成為這類的查詢,其查詢結構為

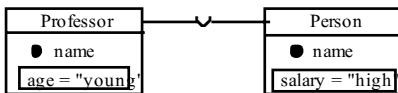


我們可以利用模糊集合中的聯集概念來解釋此種查詢,在 EQLf 中是以 max 來解釋聯集運算,因此匹配函數定義為:

$$\mu_{C_j}(a) = \max(\mu_{fc1}(a), \mu_{fc2}(a))$$

方程式 (8)

FQ4) 找出年輕的教授和有錢人的姓名。

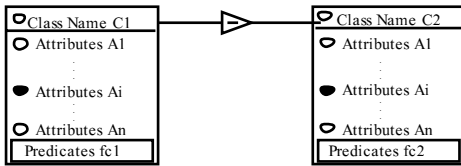


根據公式(8),物件的歸屬度為:

$$\mu_{C_j}(a) = \max(\mu_{young}(a), \mu_{high}(a))$$

(5) 模糊差集(Fuzzy Difference)

將前面介紹的模糊查詢利用差集圖示結合後就成為這類的查詢,其查詢結構為

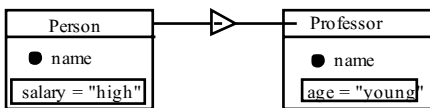


我們可以利用模糊集合中,交集和補集的概念來解釋此種查詢,在 EQLf 中是以 min 來解釋交集運算,以標準模糊互補運算來解釋互補,因此匹配函數定義為:

$$\mu_{C_j}(a) = \min(\mu_{fc1}(a), 1 - \mu_{fc2}(a))$$

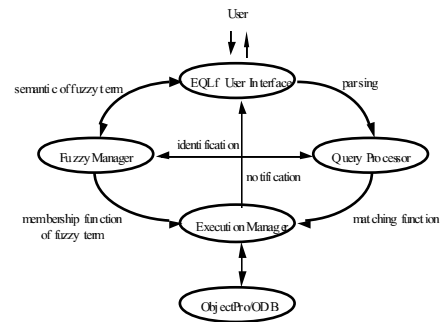
方程式 (9)

FQ5) 找出有錢但不是年輕教授的人的姓名。

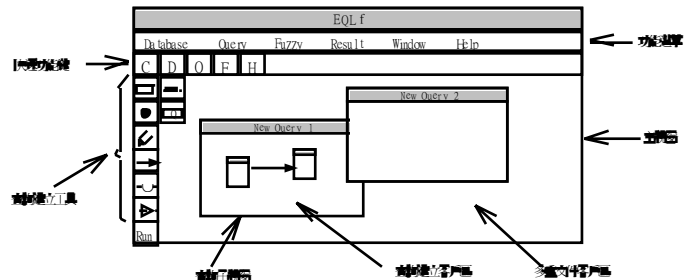


根據公式(9),歸屬度為:

$$\mu_{C_j}(a) = \min(\mu_{high}(a), 1 - \mu_{young}(a))$$



圖一: EQLf 系統架構圖



圖二: EQLf 介面外觀

第四章 計畫總結與未來工作

本計畫旨在設計與製作一個在物件導向資料庫具有描述不精確查詢條件之查詢語言,我們探討了一些處理不精確條件的方法,並分析歸納它們的優缺點,由於模糊查詢最基本的要求即必須滿足使用者的直覺性,並且對於各種模糊查詢均能提供合理的解釋,以模糊集合為基礎的解釋方式在這兩點的要求上,均具有最佳的表現,因此我們將模糊集合應用在模糊查詢的解釋上。

在語言設計方面,我們研究了 EQL 可以模糊化的部份,以不影響 EQL 原來特色為前提,利用模糊集合的相關概念來定義各種模糊查詢的匹配函數,我們選擇了最常用的模糊集合操作運算來設計我們的匹配函數,希望查詢的結果更能符合使用者的需求。

目前我們已對 EQLf 中可模糊化的部份加以擴充,模糊查詢處理的最佳化將是我們下一步研究的主题。