

一個智慧型資料庫查詢架構

楊亨利 詹子平

國立政治大學資訊管理學系

摘要

傳統的資料庫對於查詢的處理能力有限，使用者必須明確知道他自己想查詢的項目是什麼，也必須瞭解資料庫的結構。若找不到資料，也無法知道為什麼是真實世界中本來就不存在這種資料，還是他不會問問題。本研究探討這相關領域的研究：合作問答、鄰近查詢問答、關連查詢問答、放鬆問答等並整合其他智慧型查詢之研究的觀念；在傳統的關連式資料庫上提出一個智慧型資料庫查詢系統架構，包含一些預建的知識（（1）資料字典、（2）延伸資料字典 含概念樹與專有名詞、（3）資料抽象化觀念、（4）企業法則、（5）主題樹、與（6）對使用者的了解），來對使用者提供更有幫助的查詢問答協助，以驗證架構之可行性。最後，並以一個大學資料庫為應用領域之例，用 VB 6.0 建立一個雛形系統。

關鍵詞：智慧型資料庫查詢、合作問答、鄰近查詢問答、關連查詢問答、放鬆問答

1. 研究背景與目的

傳統的資料庫基於封閉世界的假設（Closed World Assumption）對於查詢（Query）的處理是只能接受明確指定（Precisely Specified）所需項目的查詢，並且也只能提供它所找到的完全符合查詢的答案（Exact Answers）。如此一來，就要求使用者完全瞭解問題領域（Problem Domain）及資料庫的概廓（Schema）。而且，也只能提供有限的資訊，甚至完全沒有任何資訊（如果找不到完全符合的答案的話，即「找不到就是否定」 “Negation As Failure”）。然而，隨著資料庫越來越龐大與複雜，不太可能期望每位使用者均對資料的領域與結構有足夠的瞭解以期能建立準確而有用的查詢。即使對組織內的資料庫，使用者也常在對所需的資料種類並無準確瞭解的情形下，對其查詢，以期能做成決策。若談到對組織外的資料庫查詢，如對於圖書資料庫或其他的旅遊等公共資料庫的查詢，使用者更是對資料庫所含內容只能憑猜測。

我們考慮一個例子：假設一個旅遊者想由台北搭早上 7 點至 12 點的飛機去紐約市。他下了一個查詢：「由台北市至紐約市的早上飛航班次及其起飛時刻為何？」傳統的資料庫只會列出一個兩欄「飛航班次」及「起飛時刻」的表格。然而，人類的旅行社專家們可能會有不同的解答方式。有的專家可能會加列出第三

欄「起飛城市」及第四欄「到達城市」，並在表格中加列出起飛城市為高雄市或到達城市為華盛頓市的飛機班次。這是基於他專家的職業判斷，認為起飛與到達城市在某個小範圍差距情形(Neighbourhood)可能是可接受的，旅遊者所關心的可能是時間的便利性，或是價錢的差異。而別的旅游專家可能不認為需要提供起飛城市與到達城市的資料，既然旅遊者指明了台北市與紐約市，也就只需考慮這兩端，但是，需要額外提供第三欄「到達時刻」、第四欄「飛航班次有效期限」(如：有的班次不固定，只在暑假才有或加班航次)、第五欄「飛航班次每週飛行日」(如：有的班次不是每天早上有，而只是週一、三、五才有)，及第六欄「票價」(有的班次，是用特別的飛機，如協和號，飛得很快，但也很貴)。以提供旅遊者更多的資訊判斷。為什麼要提供額外的資訊呢？其基本觀念在於使用者每做一個查詢，其目的不僅在得到答案以增進其知識，而是在其背後的意圖(Intention)利用資訊來採行某些行動，這些行動有可能會運用到某些額外資訊。

以上的情形，說明了一個研究領域——合作問答(Cooperative Answering)興起的背景。本研究之目的希望在傳統查詢系統加上智慧機制，試圖幫助使用者查詢他們想要的資料，分別如下：

1. 提出一整合的智慧型查詢之架構：在以前的研究，雖有多人提出智慧型查詢、合作問答的方法，但似乎僅強調其特定作法之智慧性，缺乏整體對智慧型查詢介面之整合。本研究希望對於文獻中智慧查詢之機制加以整理，釐出智慧性查詢系統應有之架構。
2. 實作一智慧型資料庫查詢雛形：基於前述所提出之整合架構，本研究希望實作一個智慧資料庫查詢系統。傳統的查詢系統，當使用者下查詢後，無論有無結果，系統處理即告一段落。本研究希望在查詢時，當查無結果時，能給予使用者協助；此外，對於顯示結果後，也能提供額外相關的資訊。

2. 文獻探討

在以往，有很多文獻在探討如何更有智慧地來回答使用者的查詢。其中很重要的研究就是合作查詢問答(CQA，即 Cooperative Query Answering)。這包含鄰近查詢問答(NQA，即 Neighborhood Query Answering)及關連查詢問答(AQA，即 Associate Query Answering)。前者為對原查詢所得之結果進一步提供其某一語意距離相鄰的或概化的資訊。後者為進一步提供原查詢並未明確問到，但概念相關的額外資訊。

NQA 的作法可以放鬆原始的查詢(Query Relaxation，如將「早上 8 點」的

起飛時間放鬆為「早上」)或放鬆所得原答案的結果(Match Relaxation)。這裡牽涉到如何決定「相鄰」(Neighborhood)的語意。有的人(如[4, 5])利用型態抽象階層樹(Type Abstraction Hierarchy)來達成。此種階層樹的概念其實即為一般化階層樹(Generalization Hierarchy), 如「旅行方式」可分「搭飛機」、「搭火車」及「搭公車」等方式, 各方式其下又可在細分。這其實也就是 Han 等人[15]的概念樹(Conceptual Hierarchy)。有的試圖對資料庫中每個屬性加上語意距離(Semantic Distance)的度量(Measure)函數觀念並附上比例尺(Scaling Factor) 相對權重(Relative Weight) 標準相鄰半徑(Neighborhood Radius)等, 來擴大回答答案或找出最佳優先解[19]。

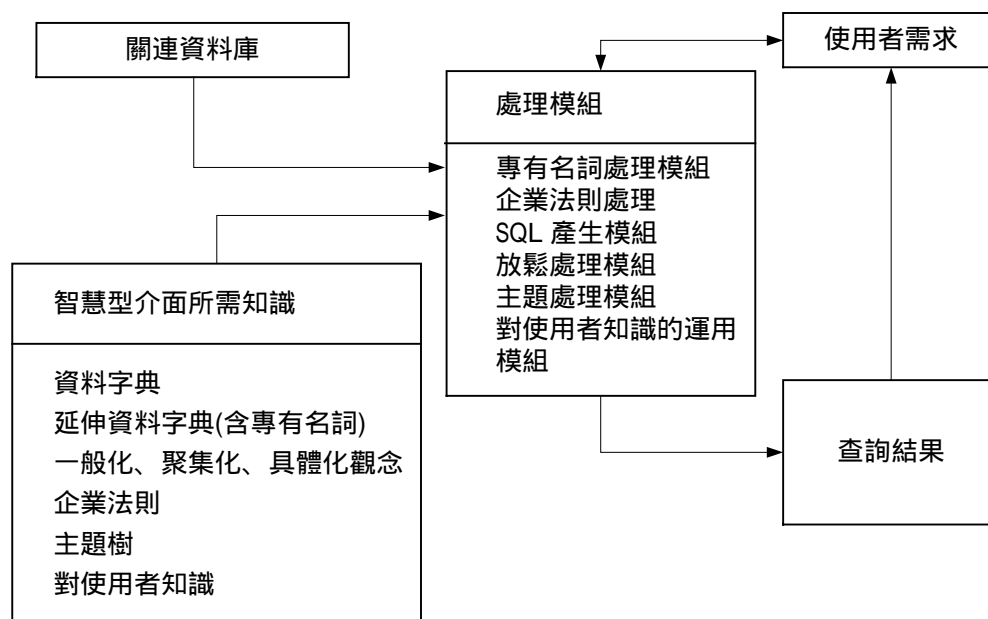
至於 AQA 的作法, 如在關連式資料庫(Relational Database)中, 對其解答加欄位(Width-Extension)、加列數(Height-Extension)、延伸表格(Table-Extension)[13, 14]。在邏輯資料庫(Logic Database)中, 則變成改寫敘述語(Rewrite Predicates)、放寬變數定義域及打破連結的相依(Join Dependency)[11]。但這牽涉到一個問題, 即如何知道那些是概念相關(Associative)? 有研究者提出藉著對資料庫中相關領域知識的瞭解, 以話題(Topic)的觀念將相關屬性值間關係, 以階層方式建立起來[7, 8, 9]。例如, 「時間表」的話題可有「起飛時間」、「到達時間」為其相關屬性。「地區」的話題可有「起飛城市」、「到達時間」為其相關屬性。Chu 與 Chen 更進而將話題明確化將之稱為主題(Subject)[5]。

除此而外, 也有其他對智慧回答的方式。例如, 可利用對資料庫語意的瞭解。基本上, 一個傳統關連式資料庫的語意, 除在其概廓(Schema)的描述外, 尚有語意完整性限制(Semantic Integrity Constraints)可用以過濾或簡化查詢[10, 11]。若進而更可用其背後所隱含的語意物件模式(如實體關聯模式)[23]。或可假設系統中蘊涵了其他知識(如某些屬性與時間有關、使用者的接取權限)[18]。甚至可在系統內留存語意辭典(Thesaurus)以供查考[3]。其目的在於提供內含的知識給資料庫查詢者[21, 22]。

另外, 有的(如 [2, 3])想藉助對於使用者的瞭解, 來提供更合適使用者的資料。這也牽涉到對使用者建模(User Modeling)的研究(如[17])。也有人[6]研究對於查詢不到的情況(Zero Result)給予改正的間接回答(Corrective Indirect Response 告知所詢問之前提即錯誤)或建議的間接回答(Suggestive Indirect Response 雖原查詢找不到答案, 但對可能的預期下一個詢問, 提供建議的答案)。

然而，以往研究對各種作法的整合較少，本研究即在企圖整合 Han 的概念樹及 Chu 與 Chen 的主題，並兼採用語意完整性限制、專有名詞及實體關係模式的知識，來智慧地回答使用者對資料庫的查詢。

3. 智慧型查詢系統架構



圖一、智慧型查詢系統架構

基於合作問答的精神，本研究對關連式資料庫提供一個智慧型查詢系統架構（如圖一），以協助使用者查詢他們所想要查的資料。相較先前文獻中所提之架構，本研究所提之架構，較具整合性。尤其是「延伸資料字典」、「資料庫抽象化觀念」與「企業法則」在以往的 CQA 研究中並未有效的整合進去。以下分別對本架構所需用到的知識及其模組，略加描述。

3.1. 智慧型介面所需用到的知識

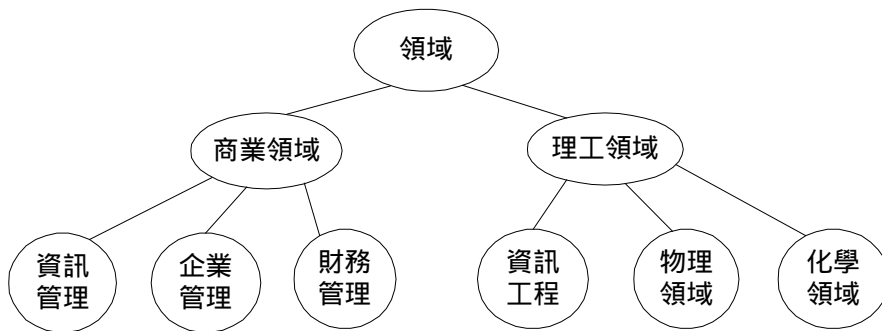
（一）資料字典

本架構下所需最基本的知識即為對關連表格（Relational Table）及其屬性定義之資料字典（含表格名稱、屬性、型態、長度、範圍等）。

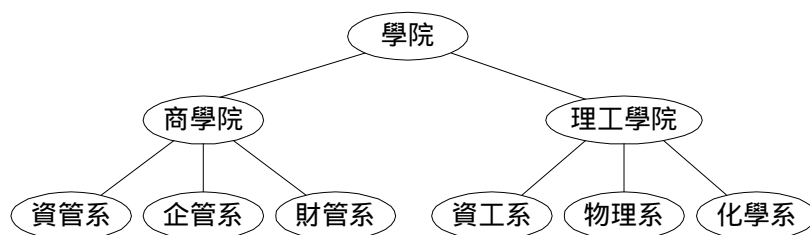
(二) 延伸資料字典

延伸資料字典包含延伸的實體關係模式 (Extended Entity-Relationship Model, 簡稱 EER) 中所有的實體、牽涉到的關係、對應比率(Cardinality)及其與關連表格之對應的定義, 亦包含概念樹以及專有名詞部分。

本架構在處理聚集化與與一般化之專有名詞以及查詢不到時放鬆處理上, 引用 Han 的概念樹, 也就是屬性值之向上一般化 (Generalization, 即 is_a) 或聚集化 (Aggregation, 即成分組成聚集) 的階層關係。以學校領域為例, 可假設概念樹資料內有一個「領域」的一般化概念樹, 「系所」的聚集化概念樹, 分別如圖二、三所示。這些概念樹在處理專有名詞時, 會用以將上層節點向下取代(如依圖三「商學院」可取代為三個系); 而在放鬆時, 會用以將下層節點向上放鬆(如「資管系」可放鬆為商學院)。



圖二、「領域」之一般化概念樹



圖三、「學院」之聚集化概念樹

專有名詞包含一般性專有名詞與資料庫設計者依領域知識, 而定義之專業術語; 例如: 上午、下午等都是一般的名詞; 而死當、學業被 1/2 (指的是學生因被當學分數超過修課學分數之二分之一, 因而被退學) 等則是在學校領域內特有的專有名詞。以專有名詞的複雜程度可分成四大類:

1. **衍生字**: 為不儲存在資料庫內之屬性, 但可用其他屬性轉換者 (即 Derived Attribute)。例如: 年紀可以生日之屬性運用函式方式來定義。
2. **簡單同義字**: 簡單同義字的處理方式, 例如: 使用者所下之限制式「成績 =

活當」，本架架構應會將之轉成「成績 < 60 且 成績 >= 50」；對「上課時間 = 上午」則會轉成「上課時間 BETWEEN 8:00 AND 12:00」。

3. **運用前述概念樹的同義字**：此乃運用前述一般化或聚集化概念樹的同義字，例如：假設商學院是個聚集化觀念，包含企管系、資管系以及財管系，若使用者下「老師任教=商學院」，則本架構應可自相關的概念樹中，將商學院展開成企管系、資管系、財管系等。
4. **較複雜的專有名詞**：例如：當查詢被 1/2 的學生時，需先查詢該學生這學期修課的總學分數，再評估被當的學分數是否大於總學分數的 1/2。對於此種專有名詞，則會有個別對應的專有名詞處理程序來處理，待實際專有名詞轉換時，系統便執行該程序。

這四類中，第一類為「屬性」之同義字，而第二、三類為「屬性值」的同義字，第四類則可能為「屬性」的同義字或「屬性值」的同義字，但需採用額外的程序來處理。

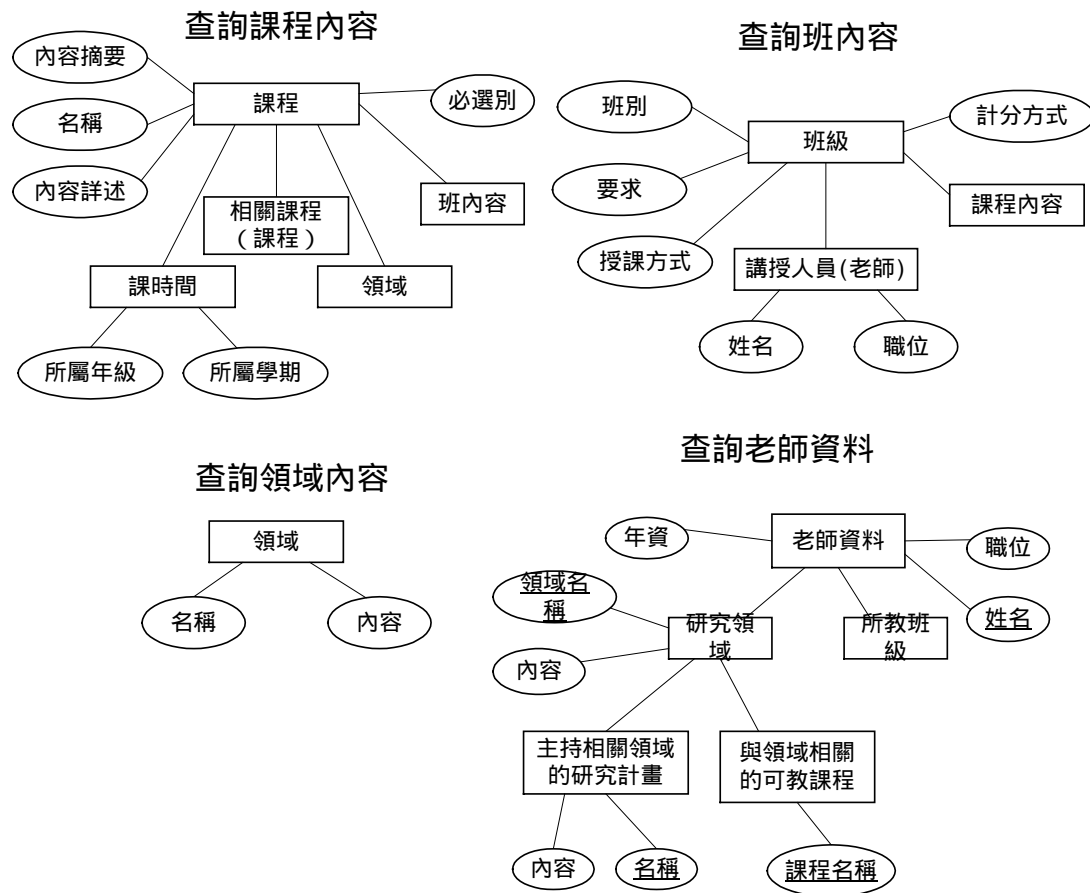
(三) 資料庫抽象化觀念

本研究將原 Chen 所提之實體關係模式 (ER model) 加入一般化 (Generalization, 相對的即為特殊化 Specialization) 具體化 (Materialization) 聚集化 (Aggregation) 三種資料抽象化觀念成為延伸的 ER 即 EER, 藉此來設計領域之資料查詢的處理方式，並協助使用者較易瞭解資料內涵。一般化及特殊化的觀念，如「大學生」是 (is a)「學生」，有助於系統在處理使用者查詢時，自動判知上下層的關係，並加入繼承屬性。具體化乃一實體抽象程度之降低，由高階的抽象概念 Concept 降至真實的具體 Concrete [12]。如「課程」具體化後，會有多個對應的「班級」實體。課程有其自己的屬性(如課程規劃者、共開幾班)，也有一些可由班級繼承的屬性。此處的聚集化則指的是實體間的複合實體聚集化關係，例如：「汽車」是由「車輪」、「車殼」、「引擎」等實體組合而成。系統支援此種聚集化關係之查詢時，當使用者下「查詢汽車之材料為何」時，系統會自動往其組成物延伸，給予「車輪」之材料、「車殼」之材料以及「引擎」的材料。

(四) 企業法則

企業法則包含一般性規則，如：「若 $a > b, b > c$, 則 $a > c$ 」，以及是領域內的特殊規則 (即語意完整性限制, Semantic Integrity Constraints)。如以學校領域為例，可能有「大學生每學期至少修課九學分、最多修 25 學分」的規則。利用來這些規則有助於簡化對查詢的處理。

(五) 主題樹



圖四、主題樹之例

主題樹是針對某領域的查詢路徑，先對該領域內知識思考與分析，由領域專家對經常的查詢回答與方式加以歸納，可以形成一些「特定主題」之主題，並列出一些相關的領域知識，來運用主題樹。相關之主題實體組合成一主題樹，而一主題實體包含一些屬性。主題樹中的主題或次主題可能對應到 EER 圖的實體，而主題與次主題的連結可能對應到 EER 中的關係。

舉例來說，一般在查詢課程時，除了查詢課程內容外，也可能同時查詢「課程所屬領域」、「班級」、「授課老師」等資料。因此，就規劃「查詢課程內容」的主題時，會包含「課程」、「領域」、「班級」等實體。此乃因為這些實體都與查詢課程內容相關，也會包括一些相關的屬性，如：「課程」之「內容摘要」、「課程」之「領域」等。當然，並不是在課程底下的所有屬性都會被納入查詢課程內容的主題，判斷的基準在於這個屬性是否真的會與主題相關。找尋適合的主題樹時，

是以使用者所選的屬性為關心焦點的基準，而使用者所選擇的關係以及目前所引用該領域的經驗法則（即領域知識）則會影響到主題樹的使用及延伸。

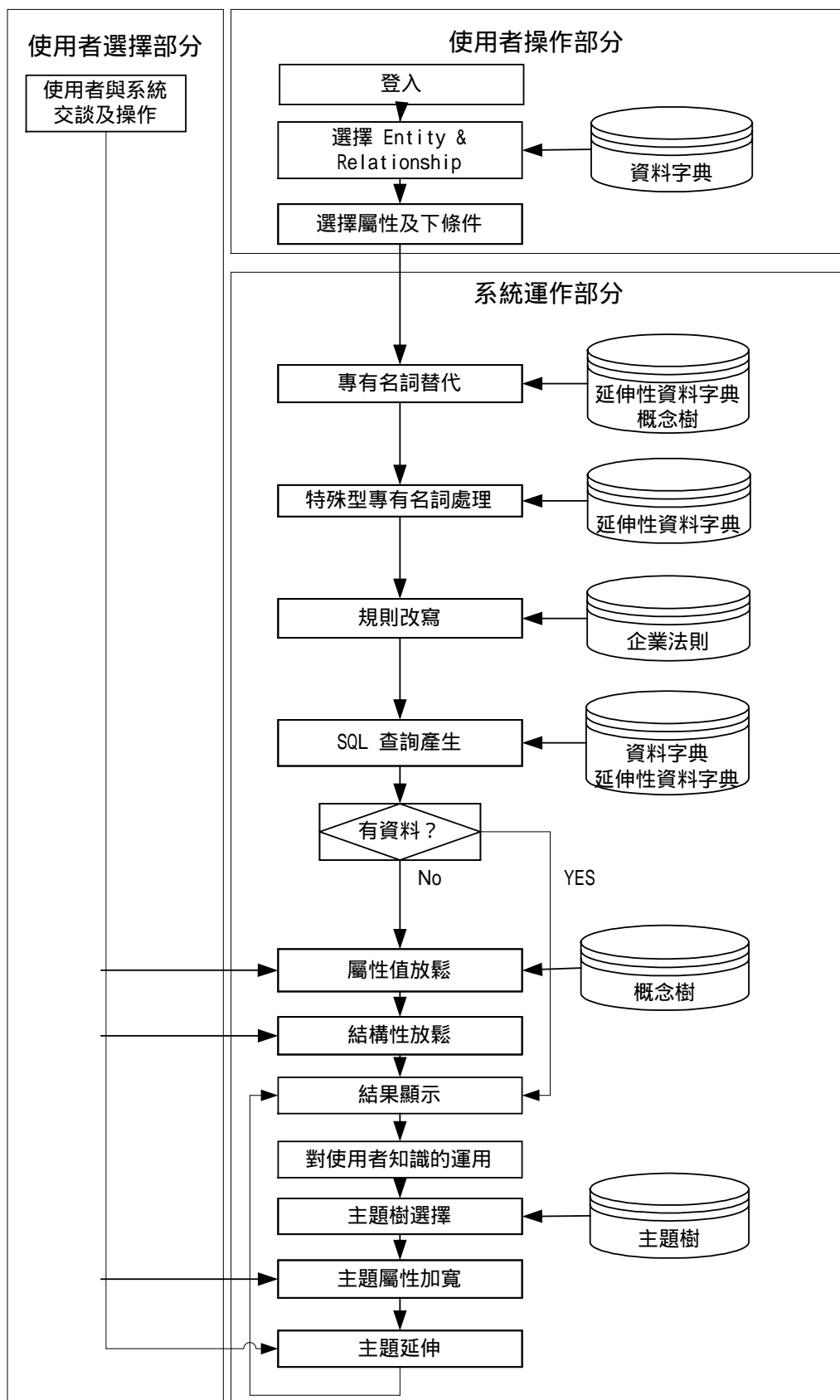
在主題樹中，長方形的根為待查詢之主題（稱為「來源主題」）其他長方形為「次主題」（這些次主題也有可能出現在另外樹的根，而成為一些可獨立之來源主題），橢圓形為主題實體之「屬性」。如圖四中共有 4 個主題樹，其中「課程」為主題實體、「課程」之「名稱」、「內容摘要」等為其屬性。使用者查詢課程內容之意涵從樹的根節點進入，但此「查詢課程內容」主題會延伸至「查詢領域內容」的次主題，這在主題樹中靠著「課程」與「領域」之連結表達。其實這也反映出其 ER 圖中「課程屬於領域」這個關係。「查詢領域內容」也可獨立進入成為一來源主題，但「查詢課時間」就只是「查詢領域內容」的次主題，而不可獨立。另外，「查詢老師資料」主題具有相關的領域知識：*「若使用者為研究生，則該生詢問有關老師資料之主題時，詢問是否要找指導老師；若是，則詢問有興趣的該生主修系所所屬之研究領域，並給予相同領域之老師資料。」*

（六）對使用者知識

此乃系統對使用者事先的了解，即 User Profile 的知識，以及多次上機的所學習到的習性，用以提供客戶化(Customize)的資訊。

3.2. 整體模組流程

本架構下之整體流程如圖五，分成三大部分：1.使用者操作部分、2.系統運作部分、3.使用者選擇部分。其中 1、3 乃使用者與系統交談的部分，包含最初使用者登入時，系統辨識其身份，是否有使用系統之權利。使用者然後輸入欲查詢之實體與關係，以及主要查詢之屬性、限制等。而在其後系統運作中，若系統無法找到查詢資料而需放鬆已找到查詢結果而可進一步提供相關屬性或主題，使用者也會介入選擇。



圖五、整體流程圖

系統運作部分為本架構主要處理的核心部分。依使用者查詢先作專有名詞處理，自延伸性資料字典取出同義字之資料，處理查詢中的同義字、衍生字以及複雜的專有名詞，以一般化或聚集化之概念樹來處理一般化或聚集化之專有名詞。

自企業法則之資料庫內取出規則，並利用規則來縮小問題範圍或過濾問題。前者例如，若有一規則為：「若角色為大學生，且有輔修或雙修者，一學期可修學分數大於 25，否則可修學分數小於 25」；當查詢「修課學分數大於 25 之學生」時，即可將查詢範圍加入「角色為大學生，且有輔修或雙修者」二條件，以縮小查詢範圍。後者例如，若有一條規則為：「無論任何條件下，修課學分數不得超過 25」；當查詢「修課學分數大於 25 之學生」時，可即判知使用者的問題於本資料庫中沒有符合的資料存在。

接著系統按照使用者之查詢透過 SQL 產生模組將使用者輸入之「實體」、「關係」、「屬性」、「限制式」等轉成標準之 SQL。並事先透過資料庫介面試將查詢結果傳回，若查無資料時，系統會作查詢放鬆之動作。

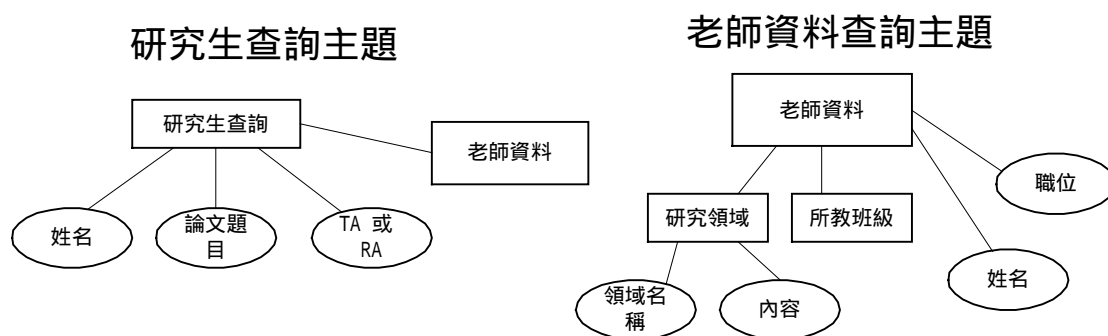
系統會查考對使用者知識來作下列之放鬆與主題樹的選擇。對查詢將作屬性值放鬆與查詢結構放鬆，以期得到查詢結果。屬性值放鬆有數字型與非數字型。數字型有不同作法。假設查詢限制式為「年紀大於等於 90」，若能先從資料庫目前資料中分佈狀況，查出查詢出資料群中的極端值——年紀最大的為 50 歲，可將查詢限制式改成「年紀大於等於 50」。較簡單者為採預設公式的方式來處理，依限制式之運算子大於、或小於的情況將原來屬性值乘以某倍數。若此倍數為 0.88，上述查詢將先改為「查詢年紀大於 44 的人」。對非數字型之放鬆，可利用屬性值或聚集化概念樹來作屬性值提升，如將「資管系」放鬆成「商學院」。而結構性放鬆則可將查詢限制式予以改寫，如將一些限制式去除，或是將限制式間之運算子改寫（如 AND 改成 OR 等）。

將查詢的結果傳回前，系統先利用使用者所下之查詢，配合對使用者的瞭解，推測使用者查詢所意涵之主題，若有找到對應的主題樹時，則可以對該查詢作主題屬性加寬、或是主題的延伸，以提供其他的屬性、相關表格等額外的資訊。

所謂主題屬性加寬指給予同一查詢主題內之實體額外之屬性。假設系統內有「研究生查詢」之主題，如圖六，主題內有「研究生姓名」、「研究生論文題目」、「是否為 TA 或 RA」等屬性。如原本查詢「研究生之姓名」，匹配到「研究生查詢」之主題後，再給予「研究生論文題目」、「是否為 TA 或 RA」等額外之屬性。而主題之延伸，以圖六為例，如現在查詢之主題為「研究生查詢」，可進一步查

詢老師的資料，而將主題移轉至「老師資料查詢」，然後得知老師之姓名、職位等。同樣地，「老師資料」也可進一步瞭解老師之「研究領域」。

當使用者看到初步結果時，可選擇其有興趣之主題屬性加寬或主題延伸，要求系統再執行，顯示結果。



圖六、「研究生查詢」及「老師資料查詢」之主題圖

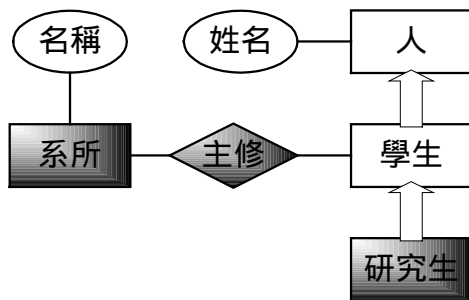
3.3. 重要模組之個別描述

由於篇幅限制，在此對個別模組之描述，僅止於較需要討論的是在「SQL 產生模組」中之「一般化的資料抽象化處理子模組」及「主題樹處理模組」。這兩個部分均係本研究之重要研究成果。以下的描述重點在於討論期間問題，故不以虛程式碼為之。

(一) 一般化資料抽象化處理子模組

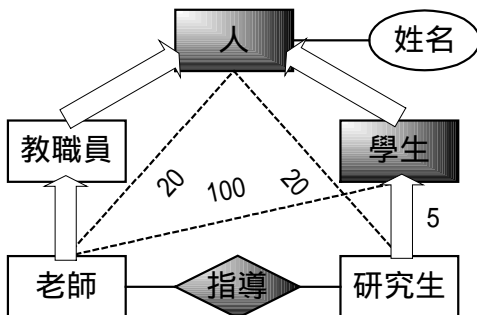
以學校資料庫為例，假設學校內「人」身份之一般化階層會有如下：「人」可分成「教職員」與「學生」，「教職員」可分成「職員」及「教師」，「學生」可分成「大學生」及「研究生」，「研究生」可再分成「碩士生」與「博士生」。

本研究之模組應能將使用者所選擇的一般化實體往上或向下自動連結至正確之實體。例如：實體選擇「研究生」、「系所」，關係選擇「主修」，屬性選擇「人」的「姓名」與「系所」之「名稱」，其意義為：「查詢研究生姓名及其主修系所名稱」。則模組會主動將研究生與學生作連結。此乃因為從圖七 ER 模式來看，「學生」才與系所做「主修」之連結，而「研究生」為「學生」之特殊化實體。



圖七、使用者選取「研究生」、「主修」、「系所」、「研究生姓名」、「系所.名稱」

假設使用者選擇實體「人」、「學生」，關係選擇「指導」，屬性選擇「人」之「姓名」，其意義為：「查詢有指導學生之人的姓名」，此時模組依圖八自動會將「人」轉成「教師」，並將「學生」轉成「研究生」(因為大學生與老師間無「指導」關係，學生不可能匹配到大學生)。



圖八、使用者選取「人」、「學生」、「指導」、「學生.姓名」
線上之數字為對應之關係距離

模組能判斷何者較為「明確」的原因在於本研究所創新提出之距離矩陣 (Distance Matrix) 的觀念，如表一。此觀念有點類似 Motro 之 Semantic Distance[19]。

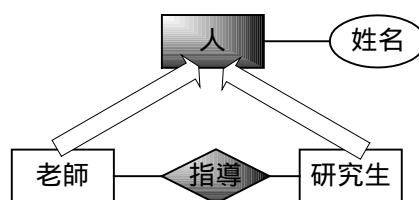
表一、一般化之關係距離

距離	人	學生	教職員	教師	職員	大學生	研究生	博士生	碩士生
人	0	20	20	20	20	20	20	20	20
學生	20	0	100	100	100	5	5	10	10
教職員	20	100	0	5	5	100	100	100	100
教師	20	100	5	0	100	100	100	100	100
職員	20	100	5	100	0	100	100	100	100
大學生	20	5	100	100	100	0	100	100	100
研究生	20	5	100	100	100	100	0	5	5
博士生	20	10	100	100	100	100	5	0	100
碩士生	20	10	100	100	100	100	5	100	0

表格中數字越小者，表示兩實體意義越接近，數字為 0 表示為同義、數字為 100 表示兩實體絕對不可轉換，亦即不為一般化之階層架構上下從屬關係，而可能為兄弟關係。兩實體若為相隔一層之父子關係，其距離為 5（如「碩士生」與「研究生」），若為祖父、子關係，其距離為 10（如「碩士生」與「學生」）。但因目前之 ER 中，在人之下的特殊化階層中最多只有 3 層，故將「人」與其他實體距離均設定為 20，表示人雖可轉換成階層樹中其他任一實體，但其優先序較低。以圖八例子來說明，使用者所選擇之「指導」關係，模組會由 ER 中判知此二元關係之連結實體為「老師」及「研究生」，再者因為使用者選擇了「人」與「學生」二個的實體，模組皆下來列出「人 老師」、「人 研究生」、「學生 老師」、「學生 研究生」之四對距離分別為 20、20、100、5（不會列出「人」與「學生」之比較，此乃因為使用者所選擇為二元關係，而且其選擇了兩個實體，此二實體不可能為同一實體）。因此，模組先將距離最小之二實體配對取出並作連結轉換，此例為「學生 研究生」，接著，再自剩下實體再重新列出其關係距離並匹配之動作。此例中，「學生」與「研究生」匹配後，由待匹配實體中去除，只剩下之「人」、「老師」作匹配。

上述為使用者直接選擇之一般化階層樹中實體數目（以下簡稱「一般化實體數目」）與選擇之關係相關的實體屬一般化階層樹者（以下簡稱「關係中一般化實體」）相等的情況，若這兩部分實體數目不一樣時，下述分別討論之：

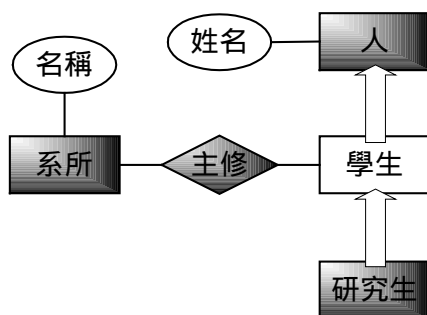
1. 若未選擇一般化實體數目或選擇之關係中一般化實體數目為 0 時，此為錯誤之語法，表示使用者並未選擇相關之一般化實體，可是由其所選擇之關係卻包含一般化階層樹之實體；或是使用者多選了一般化實體，但其選擇之關係卻沒有相關的實體屬於一般化階層樹。例如：使用者選實體「人」、「班級課程」與「課程」，關係選「開班」，此例中，按照「開班」之關係，其對應實體應為「班級課程」與「課程」，並未包含一般化之實體，使用者多選了「人」，故為錯誤之選法。
2. 若所選之一般化實體數目少於關係中一般化實體數目，此種情況實際上並不會發生，故本模組會視為錯誤之選法。例如：使用者選「人」，關係選「指導」並選擇「人」之「姓名」，如圖九所示。



圖九、錯誤之關係實體選法圖例，使用者僅選擇實體「人」，關係「指導」

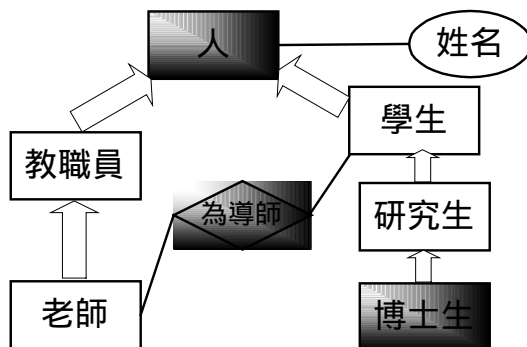
因為「指導」對應之一般化實體為「老師」及「研究生」。在前述第 12 頁中假設的一般化階層樹中，不可能同一個人可能同時兼具「老師」與「研究生」之身分。故此例為錯誤之選法。

3. 若所選之一般化實體數目多於關係中一般化實體數目，則分二階段進行：第一階段依照相等之方法，先將所有匹配之關係距離列出，找出最小關係距離之匹配，直到剩下所匹配之實體中已無關係中之一般化實體。第二階段將剩下所選之一般化實體重新對所有關係中一般化實體匹配，將所剩之一般化實體匹配完畢。例如：於圖十中，使用者選「人」、「研究生」、「系所」，關係選「主修」，其意圖為查詢「有主修某系所之研究生，並顯示這些人的姓名」。因此，在第一階段中，模組列出「研究生 學生」與「人 學生」的可能配對，並利用距離矩陣得知「研究生 學生」之關係距離最小，故先將「研究生」與「學生」作匹配，然後進入第二階段。在第二階段中，所剩之實體「人」再與關係中所有之一般化實體「學生」作匹配，將「人」轉成「學生」。



圖十、使用者實體選「人」、「研究生」、「系所」，關係選「主修」

至於將「人」與其他實體之關係距離設為 20，而非按照父子、祖孫關係給予 5、10，在此進一步解釋：如此做的目的是在作匹配動作時，給予其他非「人」實體間匹配較高優先權。舉例來說，當使用者選「人」、「博士生」，關係選「為導師」，其意圖為查詢「為博士生導師之人」如圖十一。



圖十一、使用者選擇「人」、「博士生」，關係選「為導師」

今假設「人」對其他實體之關係距離不為 20 而是設定成路徑數乘以 5 的方式。因此模組列出「人 學生」、「人 老師」、「博士生 學生」、「博士生 老師」分別為 5、10、10、100，因此「人」會與「學生」匹配，但接下來剩下實體「博士生」與「老師」便發生無法匹配之情況。

若採用原來表格之設計，「人 學生」、「人 老師」、「博士生 學生」、「博士生 老師」之關係距離分別為 20、20、10、100，因此「博士生」會與「學生」作匹配，而剩下實體「人」就可以與「老師」正常匹配。而這種「人 老師」、「博士生 學生」匹配對應才與使用者查詢意圖一致。

(二) 主題樹處理

假設以三種表格來儲存有關主題樹的資訊。「主題匹配表」用以匹配使用者所選之屬性，其所儲存內容均為主題樹之根節點。而「主題延伸表」作為概念延伸用，不作匹配之判斷，其所儲存的均為主題樹中沒有獨立之次主題；他們為主題樹中非根之長方形節點，而且不出現在匹配表中。「主題樹節點表」則用以連接主題樹與次主題樹（不管次主題是否可獨立）。如表二、三、四為儲存圖四部分資料之例。

然而，考量對資料擷取的安全權限，我們還需要在這三個表中新增之「條件」欄位。在作主題屬性放鬆或作主題延伸時，唯有「條件」內之條件為真，模組才給予放鬆之目前欄位或延伸主題。如查詢「教職員之姓名」時，僅在使用者為管理人員時，才給予額外「教職員」的「薪水」欄位。另在表四中，還需一個「連結」欄位，用以記錄主題與次主題間以何種「關係」作連結。如以「課程」與「班級內容」間是以「開課」作為連結。

主題樹之使用方式為主題屬性加寬與主題樹之延伸。主題屬性加寬之運作為對於原查詢加入額外相同主題之屬性，而主題延伸之作法則是對於原查詢加入額外次主題之實體、屬性以及連結。主題延伸運作時，會使用「主題匹配表」、「主題延伸表」、「主題節點表」，以及若主題與次主題間為一般化關係則會利用「一般化表格」，若為其他關係則會利用延伸資料字典中的「關係定義表格」以及「實體定義表格」。在主題延伸時，原來的限制會持續至次主題。例如，原查詢「課程為「資料庫」之內容摘要」，其中「課程為「資料庫」」為查詢之限制式，假設目前要將主題延伸至「班內容」時，並非單純給使用者額外的「班內容」之「班級名稱」，而是給「課程為「資料庫」之班級名稱」。

表二、雛形之主題匹配表之例

主題名稱	實體	屬性
查詢課程內容	課程	名稱
查詢課程內容	課程	內容摘要
查詢班內容	班級	班別
查詢班內容	班級	要求
查詢老師資料	老師	姓名
查詢老師資料	老師	職位
查詢領域內容	領域	名稱
查詢領域內容	領域	內容

表三、雛形之主題延伸表之例

延伸主題名稱	實體	屬性
查詢班內容之講授人員	老師	姓名
查詢班內容之講授人員	老師	職位
查詢課程內容之課時間	課時間	所屬年級
查詢課程內容之課時間	課時間	所屬學期

表四、雛形之主題節點表之例

來源主題名稱	延伸主題名稱
查詢課程內容	查詢班內容
查詢課程內容	查詢領域內容
查詢課程內容	查詢課程內容之課時間
查詢班內容	查詢班內容之講授人員

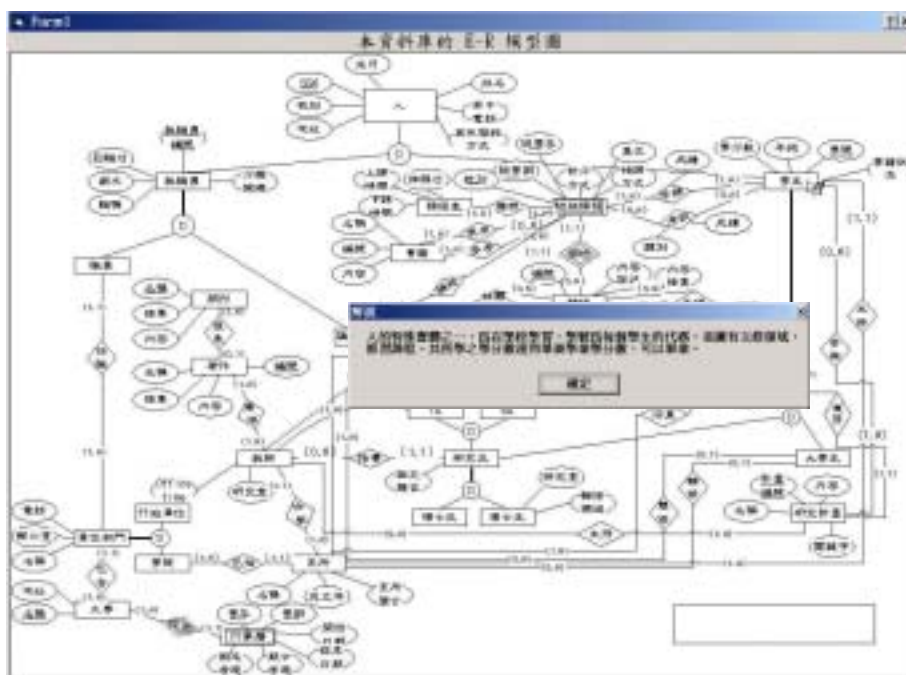
4. 智慧型查詢系統實作與展示

根據前述架構，本研究建立一智慧查詢雛形系統，選定學校作為應用領域，在 Pentium III 800、Windows 2000 的作業平台，以 Visual Basic 第 6 版發展程式，利用 ODBC 來對 Access 資料庫作存取。此學校資料庫共 26 個實體、28 個關係，轉成關連式資料庫後有 33 個表格。其資料字典與延伸資料字典（含「課程」、「領域」之一般化概念樹，「時間」、「機構」之聚集化概念樹，及「大於」、「等於」、「小於」、「人.年紀」、「教職員.年資」、「上午」、「中午」、「下午」、「晚上」、「活當」、「死當」、「被當」、「及格」、「1/2」、「總平均」等專有名詞）均儲存於 Access 表格。本系統模擬學校領域內使用者查詢之情境，設計出 14 個主題樹，包含「查詢課程內容」、「查詢班內容」、「查詢課程時間」、「查詢班時間」、「聯絡老師」、「查詢老師資料」、「查詢教職員資料」、「查詢書籍」、「輔修雙修」、「研究計畫查詢」、「學生資料查詢」、「系所查詢」、「基本資料查詢」以及「研究生查詢」等主題。企業規則雖經初步設計（包含一般化規則 89 條、領域規則 8 條），但由於時間因素，未予與整體系統整合。故目前雛形缺少企業規則之資料描述及處理模組。

有關使用者知識的運用模組，目前也僅用到其身份（學生、教職員）來協助放鬆與主題樹之資料，而且其身份係利用大學資料庫內之資料，並未另有資料結構。



圖十二、系統主畫面



圖十三、目前應用領域之 ER 圖

本雛形系統主畫面如圖十二所示，而其後使用者操作也採用圖形化介面的方式。系統會自動將使用者之查詢轉成 SQL，接著再將資料結果傳回。系統具有線上求助的功能，當使用者不瞭解資料庫的內容時，可於主畫面點選「ER 顯示」，系統將會展示出資料庫所根據之 ER 模式圖（如圖十三），使用者可進一

步於圖上點選不瞭解的實體、關係或屬性，系統即以對話盒方式顯示該物件之解釋。此外，使用者可於主畫面（圖十二）上點選「求助」或按下 F1，系統即會出現求助畫面。該求助包含本研究之描述、本雛形系統之功能解說、ER 模式、資料庫內資料之解釋。使用者若不瞭解 ER 中何謂實體時，可於該求助畫面進一步點選「何謂實體」。系統即對這些專門術語的意義、種類、圖形、例子作解釋。此外，該求助畫面還對目前的應用資料庫內容加註其語意，使用者可瞭解資料庫內的特定組成及意義。若使用者點選某實體（如「教師」）的求助，系統除對該「實體 教師」之涵義作解釋、告知其所有屬性外，還可連結至與其相關的「主鍵」、「關係」（如「指導」）的求助。

在系統運作，首先，系統會要求使用者輸入名稱及密碼。輸入正確時，系統才會准予登入。系統並依登入之使用者名稱，來判斷是否為學生或是教職員，並將資訊顯示在標題上；如圖十二中使用者身份為研究生。接著系統會出現一主畫面，並有「新建查詢」、「執行結果」、「ER 顯示」的功能，若經過多次查詢後，該主畫面下方會以簡單的自然語言記錄使用者查詢之歷史紀錄。如圖十二中已有四項查詢紀錄。



圖十四、使用者點選關係實體之例

舉例來說，如果使用者要查詢：「查詢老師指導之研究生姓名」。就先在主畫面上點選「新建查詢」系統即會列出所有之實體與屬性（如圖十四）。如點選相關之實體：「教職員」及「學生」，關係選「指導」。接著就會出現系統查詢之主要輸入畫面（如圖十五），在選擇屬性的畫面只會出現剛點選的實體：「教職員」及「學生」。以選擇的實體而言，除了本身的屬性外，也會將該實體因一般化抽

象關係所繼承的屬性一同列出；如本例中的「學生」實體，就將「人」之屬性(姓名、年紀、聯絡電話等)一起列出來供使用者選擇。接著點選實體中「學生」之「姓名」屬性，以得知學生姓名的資料。使用者可於圖十五中加入限制式，本系統考慮多個限制式中有括弧的情況，將加入限制式型態分成三種運算，假設 S 為已經加入的限制式字串， A 為目前欲加入之限制式， OA 為 A 與其他限制式間之運算式，目前有三種作法：(1)「加入目前子限制」後， S 變成 $SOAA$ ；(2)「另起新的子限制」後， S 變成 $SOA(A)$ ；(3)加入並結束目前子限制 後， S 變成 $SOA A$ 。至於左右括弧的數目，系統會自動判斷對右括弧加以增減，不會發生限制式錯誤之情況。



圖十五、系統出現使用者點選之實體及其相關屬性

當系統接到使用者所選擇之「實體」與「關係」後，便將實體中分成一般化實體與非一般化實體，並查詢資料字典，將關係所連結之實體亦作一般化與非一般化實體之分類，如此作的原因在於一般化實體可以上下階層轉換匹配，系統自動會將合適之實體予以配對。分類完成後，先檢查使用者所選的一般化實體是否與關係中一般化實體數目相同，若否表示使用者選擇錯誤之關係或實體。接著利用距離矩陣來匹配一般化之實體，條件及情況如前述。當一般化實體間配對成功後，利用一般化表格將連結之資訊加入限制式中，產生配對連結條件。並利用匹配後之結果產生一般化上下階層（如「學生」與「人」）連結條件。最後利用延伸性資料字典，將關係之連結加入限制式中，並加入使用者所選擇之屬性名

稱、限制式等。當輸入完查詢時，系統便配合專有名詞表來檢查是否有衍生字、簡單同義字、複雜同義字，替代其原始字串並新增必要之查詢條件。也會利用屬性化概念樹及聚集化概念樹來判斷是否有運用概念樹之專有名詞，將之取代成樹葉的節點，經過專有名詞處理後，產生對應的 SQL 語法。按下圖十五中「預覽 SQL」鈕，可得知目前的 SQL 語句，及相關實體的轉換及連結；若按下圖十五中之「確定」扭即可得知查詢結果。

在顯示查詢的結果之前，系統會將目前查詢作主題匹配的動作，看目前使用者查詢相關的主題為何。系統會依匹配的主題，予以詢問使用者是否需要額外的資訊。如果使用者選「是」時，系統自動放寬該主題同一「主題實體」下所有的屬性；選「否」時，則系統按照使用者原來所選擇的屬性來顯示。

圖十六、系統放鬆畫面圖例

然而，當假設使用者下之查詢，資料庫查無所需之資料時，系統會採限制式之屬性值與結構之放鬆。例如使用者所選擇的實體為：「教師」、「學生」，關係選擇「指導」，屬性選「學生.姓名」，而限制式：「教師.年紀 大於 50」且「老師.

任教 等於 '商學院' 」且「學生學籍狀況 等於 1/2 」。查詢其意為「尋找有任教於商學院、年紀大於 50 之老師、其指導的學生被 1/2 的情況，這種的學生姓名」。在這個查詢中，也包含「衍生字」(「年紀」) 專有名詞，系統會將其轉成「Year(Now)-Year(人.生日)」。此時，系統會發現資料庫內並無所需之資料，接著系統進入放鬆步驟，顯示查詢放鬆的畫面。如圖十六所示。

系統在查不到資料的情況下可就限制式之屬性值與結構兩方面來作放鬆。就屬性值而言，「年紀大於 50」會擴大範圍至「年紀大於 44」，「商學院」可放鬆至「教學單位」，而「學籍狀況等於 1/2」則因語意明確，則不作屬性值之放鬆。在結構放鬆方面，使用者可將勾選限制式取消，系統會將該限制式去除。按下按鈕則可作 AND/OR 運算之切換。例如：如果使用者將限制結構放鬆中之「教師年紀大於 50」這項條件勾選取消，並將「教師任教等於 '商學院」限制式後的 AND 按鈕按成 OR，查詢限制式就會變成「教師任教等於 '商學院」或「學生學籍狀況等於 1/2」。在圖十六中按下「選擇好了」回到原來輸入查詢的畫面。按下確定系統會將資料傳回，若無資料時，系統會再次作放鬆的動作，直到有查詢結果為止。圖十七即為查詢後的結果，使用者此時除了可看見查詢的結果外，也可分別點選[Entity]、[Relationship]、[Constraint]來協助其回憶他們自己所下的實體、關係及限制等資訊。



圖十七、放鬆後顯示查詢學生之結果

此外，系統以原來的查詢匹配其所推測的主題，並檢查是否有主題相關的領域知識；若有，則先執行該領域知識；若沒有，才繼續執行主題查詢。就目前使用者所匹配的「查詢研究生資料」的主題而言，並無該主題之領域知識，因此僅就該主題之結構，在圖十七中，容許給予額外之資訊，如「論文題目」、「TA 或 RA」等屬性。



圖十八、查詢老師之結果顯示

另一例子：如果使用者查詢為「尋找有任教於商學院、年紀大於 50 之老師、其所指導的學生被 1/2 的情況，這種老師姓名」，其所選擇的實體、關係、限制式與前查詢相同，而屬性選「老師.姓名」。假設與前者一樣，因為這些限制式下並無資料，所以系統會先經過放鬆的階段，然後才會進入資料執行結果。在顯示結果前，先會執行主題匹配之動作，此時，因使用者的焦點放在老師的資料，系統會自動匹配目前所屬主題為「查詢老師資料」。接著再檢查是否與該主題相關之領域知識，當系統到目前階段發現「查詢老師資料」主題具有如前述之相關的領域知識：「若使用者為研究生，則該生詢問有關老師資料之主題時，詢問是否要找指導老師；若是，則詢問有興趣的該生主修系所所屬之研究領域，並給予相同領域之老師資料。」因此，系統會先應用此領域知識，判斷登入之使用者是否為研究生，若是則顯示「是否要找指導老師」。若使用者此時選「是」，則顯示相關之研究領域。假設目前登入之使用者為資管系之研究生，系統會出現資管系相關

之研究領域，給使用者勾選有興趣之研究領域。執行結果如圖十八所示。該查詢結果即為符合查詢限制與使用者要求之研究領域的老師姓名資料。除了額外的欄位外，也可依圖四進一步查詢「同一老師的研究領域以及其所教班級」，這即所謂達到表格延伸的機制。如在圖十八，若此時使用者對老師的研究領域有興趣，則可點選「老師研究領域」然後按下「請查詢」按鈕。系統就會出現該老師的研究領域，如圖十九。此時主題就從「查詢老師資料」移轉至老師之「研究領域」，主題樹中主要預設的屬性會先被帶出，如「查詢老師資料之研究領域」的屬性「領域名稱」就會先顯示出來，其餘的屬性使用者可從圖十九「其他欄位...」來點選。在圖十九中，可查詢此領域的其他欄位如「領域」之「內容」。或是繼續往下延伸到與老師所主持相關領域的研究計畫。也可回至圖十八中，點選「老師所教班級」之主題。



圖十九、由「老師研究領域」到「老師研究領域的相關研究計畫」之結果顯示

5. 結論

隨著資訊科技之發達，使用者所接觸到的資訊從不足到過多。因此智慧型查詢介面變得在今後的環境中，變得相當地重要，一個良好的智慧型查詢介面能協助使用者找到他們所想要的資料。智慧型查詢所應用的範圍不僅在於資料庫之資料檢索，Internet 或 Intranet 的搜尋引擎、文件搜尋等都是應用的範圍。然而目前的產品中，多數還是僅利用文字匹配並配合布林運算的方式來檢索。

利用本研究的成果，除了提供上述領域更具智慧型機制外，將可在組織內整合決策系統(Decision Support System)或主管資訊系統(Executive Information System)，有效提供決策者或主管決策所需資訊。在組織外，智慧性地答覆與引導一般外界使用者對公共資料庫查詢。而若與知識發掘領域加以結合，更可利用知識發掘的結果，建立多層次的資料庫(Multiple Layered Database)，來回答使用者查詢。也可運用在純知識庫的查詢環境上。

5.1. 貢獻

本研究在架構與系統的貢獻，簡言之如下：

(一) 架構上的貢獻

1. 智慧查詢架構較完整：本研究架構整合各家對查詢之智慧機制，架構上較其他文獻所提者完整。整合後的架構更能利用各種先前知識來協助使用者。
2. 架構模組之應用擴充性高：本架構採用的模組資料內容可抽換，雛形系統與應用領域獨立，因此可應用到不同的領域。

(二) 系統上之貢獻

1. 提供智慧型查詢之圖形化介面：讓使用者方便輸入查詢，並處理放鬆與主題巡覽等。
2. 領域內資料擴充性高：對於資料的儲存，本雛形多採用資料庫來儲存，而非內建在程式碼中。以後研究者可方便地替換本雛形內容，即可應用至其他領域。而在資料庫之介面上採 ODBC 的方式，故對於領域內資料擴充性高。

5.2. 後續建議

本研究整合各智慧型查詢機制，但迫於時間因素，不論在概念的深度與實作的功能上，均不及文獻中專門探討模組者，因此建議的後續研究包含下列各項

(一) 對研究架構的建議

1. 目前對查詢不到結果情況的處理採放鬆作法，但其放鬆機制仍較粗略。相對文獻上，如 Corella 等人的作法[6]，本研究仍有改進的空間。
2. 本研究參考以往文獻提出主題樹的運用，但目前對大部分主題只是使用 ER 中實體、關係與屬性，領域知識仍有限。這方面仍有待改善。另外，對於如何尋找出適當的主題樹，也有研究的必要。
3. 本研究架構中雖有 User Modeling 的觀念，但目前只是粗略的想法，應如何

搜尋、運用、學習使用者的知識，均有待研究。

(二) 對系統實作的建議

1. 增補原架構中所提到、但雛形未完全完成的功能如企業法則處理模組、使用者知識運用模組及對實體聚集化觀念的運用。
2. 輸入介面的改進：目前系統在輸入雖採圖形化介面點選但仍有幾點限制，包含 (a)使用者無法下達巢狀查詢(Nested Query)、與(b)對條件的下達，受到限制（例如無法任意加小括弧）。對 (a)的改進可能要 SQL 的語法編譯器；(b)的情形則需深入思考如何設計。但以長遠來看，未來應加強對自然語言輸入方式的研究。另外，也可提供多重輸入方式，如容許其直接輸入完整的 SQL；但這樣就要有 SQL 的語法編譯器。若有此編譯器也可容許對以圖形介面輸入後的 SQL 預覽畫面加以修改。
3. 將此雛形以網際網路伺服器之語言(如 ASP、PHP、Perl 等)改寫，使期能在任一台機器 Web 瀏覽器上執行。

參考文獻

- [1] Bosc, P. and Pivert, O., "Some Approaches for Relational Databases Flexible Querying," *Journal of Intelligent Information Systems*, Vol. 1, 1992, pp. 323-354.
- [2] Brajnik, G., Guida, G., and Tasso, C., "User Modeling in Intelligent Information Retrieval," *Information Processing & Management*, Vol. 23, No. 4, 1987, pp. 305-320.
- [3] Chen, H., "Knowledge-based Document Retrieval: Framework and Design," *Journal of Information Science*, 1992, pp. 293-314.
- [4] Chu, W.W., Lee, R. C., and Chen, Q., "Using Type Inference and Induced Rules to Provide Intensional Answers," *7th Int. Conference Data Engineering*, Washington, D.C. 1991, pp. 396-403.
- [5] Chu, W.W. and Chen, Q., "Neighborhood and Associative Query Answering," *Journal of Intelligent Information Systems*, Vol. 1, 1992, pp. 355-382.
- [6] Corella, F., Kaplan, J. S., Wiederhold, G., and Yesil, L., "Cooperative Responses to Boolean Queries," *1th Int. Conference Data Engineering*, 1984, pp. 77-85.
- [7] Cuppens, F. and Demolombe, R., "Cooperative Answering: A Methodology to Provide Intelligent Access to Databases," in *Proceedings from the Second International Conference on Expert Database Systems*, edited by L. Kerschberg, Benjamin/Cummings, 1989a, pp. 621-645.
- [8] Cuppens, F. and Demolombe, R., "Extending Answers to Neighbour Entities in

- a Cooperative Answering Context, ” *Decision Support Systems*, Vol. 7, 1991, pp. 1-11.
- [9] Cuppens, F. and Demolombe, R., “ How to Recognize Interesting Topics to Provide Cooperative Answering, ” *Information Systems*, Vol. 14, No. 2, 1989b, pp. 163-173.
- [10] Gaasterland, T., “ Cooperative Answering through Controlled Query Relaxation, ” *IEEE Expert*, September/October 1997, pp. 48-59.
- [11] Gaasterland, T., Godfrey, P., and Minker, J., “ Relaxation as a Platform for Cooperative Answering, ” *Journal of Intelligent Information Systems*, Vol. 1, 1992, pp. 293-321.
- [12] Goldstein, R.C. and Storey, V.C., “ Materialization, ” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No. 5, 1994, pp. 835-842.
- [13] Han, J., Fu, Y., and Ng., R.T., “ Cooperative Query Answering Using Multiple Layered Databases, ” in *Proceedings of the Second International Conference Cooperative Information Systems*, Toronto, Canada, May 1994, pp. 47-58.
- [14] Han, J., Huang, Y., Cercone, N., and Fu, Y., “ Intelligent Query Answering by Knowledge Discovery Techniques, ” *IEEE Transaction on Knowledge and Data Engineering*, Vol. 8, No. 3, 1996, pp. 373-390.
- [15] Han, J. and Fu, Y., “ Exploration of the Power of Attribute-Oriented Induction in Data Mining, ” in *Advances in Knowledge Discovery and Data Mining*, edited by U.M. Fayyad et al., AAAI Press, Menlo Park, 1996, pp. 399-421.
- [16] Imiflinski, T., “ Intelligent Query Answering in Rule Based Systems, ” *Journal of Logic Programming*, Vol. 4, No. 3, September 1987, pp. 229-257.
- [17] Kass, R. and Fimin, T., “ General User Modeling: A Facility to Support Intelligent Interaction, ” in *Intelligent User Interfaces*, edited by J.W. Sullivan and S.W. Tyler, Addison-Wesley, Reading, 1991, Chapter 6, pp. 111-128.
- [18] Kok, A.J., “ The Design and Implementation of an Intelligent Query Tool for Relational Databases, ” *Expert Systems*, Vol. 12, No. 4, November 1995, pp. 347-351.
- [19] Motro, A., “ Extending the Relational Database Model to Support Goal Queries, ” in *Proceedings from the First International Conference on Expert Database Systems*, edited by L. Kerschberg, Benjamin/Cummings, 1987, pp. 129-149.
- [20] Motro, A., “ Using Integrity Constraints to Provide Intensional Answers to Relational Queries, ” *Proceedings of the Fifteenth International Conference on Very Large Data Bases*, Amsterdam, 1989, pp. 237-246.
- [21] Motro, A. and Yuan, Q., “ Querying Database Knowledge, ” *Proceedings of ACM SIGMOD International Conference on Management of Data*, Atlantic City,

May 1990, pp. 51-60.

- [22] Motro, A., "Intensional Answers to Database Queries," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No.3, June 1994, pp. 444-454.
- [23] Papazoglou, M.F., "Unraveling the Semantics of Conceptual Schemas," *Communications of the ACM*, Vol. 38, No. 9, September 1995, pp. 80-94.
- [24] Pirotte, A. and Roelants, D., "Constraints for Improving the Generation of Intensional Answers in a Deductive Database," *5th Int. Conference Data Engineering*, 1989, pp. 652-659.

