

國 籍：(中文/英文)

1. 荷蘭/Netherlands
2. 美國/USA
3. 美國/USA
4. 美國/USA
5. 俄羅斯/Russia

四、聲明事項：

主張專利法第二十二條第二項 第一款或 第二款規定之事實，其事實發生日期為： 年 月 日。

申請前已向下列國家(地區)申請專利：

【格式請依：受理國家(地區)、申請日、申請案號 順序註記】

有主張專利法第二十七條第一項國際優先權：

美國；2007年2月5日；11/671,414

無主張專利法第二十七條第一項國際優先權：

主張專利法第二十九條第一項國內優先權：

【格式請依：申請日、申請案號 順序註記】

主張專利法第三十條生物材料：

須寄存生物材料者：

國內生物材料 【格式請依：寄存機構、日期、號碼 順序註記】

國外生物材料 【格式請依：寄存國家、機構、日期、號碼 順序註記】

不須寄存生物材料者：

所屬技術領域中具有通常知識者易於獲得時，不須寄存。

國 籍：(中文/英文)

1. 荷蘭/Netherlands
2. 美國/USA
3. 美國/USA
4. 美國/USA
5. 俄羅斯/Russia

四、聲明事項：

主張專利法第二十二條第二項 第一款或 第二款規定之事實，其事實發生日期為： 年 月 日。

申請前已向下列國家(地區)申請專利：

【格式請依：受理國家(地區)、申請日、申請案號 順序註記】

有主張專利法第二十七條第一項國際優先權：

美國；2007年2月5日；11/671,414

無主張專利法第二十七條第一項國際優先權：

主張專利法第二十九條第一項國內優先權：

【格式請依：申請日、申請案號 順序註記】

主張專利法第三十條生物材料：

須寄存生物材料者：

國內生物材料 【格式請依：寄存機構、日期、號碼 順序註記】

國外生物材料 【格式請依：寄存國家、機構、日期、號碼 順序註記】

不須寄存生物材料者：

所屬技術領域中具有通常知識者易於獲得時，不須寄存。

九、發明說明：

【發明所屬之技術領域】

本發明係有關可實現元素類型之類型流的查詢型樣。

【先前技術】

在電腦語言領域中，傳統上一直有許多不同的語言分類。例如，許多程式語言針對大多數終端使用者程式規劃目標提供一般解決辦法；同時另一方面，查詢語言常致力於依據資料庫查詢的資訊探勘。多年來一直不斷努力試圖將查詢能力加入至程式語言中，然而卻存在著諸多困難。

此種困難之一為，大多數現代程式語言係應用物件導向模型，其允許階層式、抽象性、模組性、封閉性(encapsulation)及數種其它語型變化，意圖減低程式任務複雜度。相反的，現代資料庫大多數為關聯式資料庫，因此查詢語言傾向應用關聯式模型而非物件導向模型。

另一困難為，程式語言發展環境(例如，整合式開發環境(Integrated Development Environments, IDE))已發展為可提供相當精密的輔助方法予開發者，如此若沒有他們將難以執行程式設計。一實例為諸如自動完成功能或機制之類的線內上下文資訊。這些自動完成機制在分類、變數名稱及其它結構已事先定義的物件導向環境中不會出現問題，但其常無法用於查詢語言，這是因為在該查詢被編譯或轉譯完成前無法實施自動完成所必需的元素類型之類型查核。此外，在查詢完成前無法轉譯該查詢，如此會造

成自動完成機制無實際效用。而另外，若查詢運算式變形，則編譯錯誤可為未知且難以修復。

【發明內容】

下文將簡單介紹本發明申請標的，以提供對本發明申請標的多種態樣的基礎認識。此發明內容並非對本發明之申請標的作一廣泛綜述。其無意要確定本發明申請標的之關鍵或重要元素，也無意界定本發明申請標的之範圍。其唯一目的為簡單介紹本發明申請標的之一些概念，作為準備在之後實施方式中更詳細敘述本發明。

此處所揭示及主張之申請標的，在其之一態樣中包含電腦可實施技術，用以增進一運算式中運算子之間的元素類型之類型流。運算子可為(但不限於)查詢運算子，且就其本身而言，類型可在整個查詢運算式中自一查詢子句流至下一查詢子句。

依據本發明申請標的之一態樣，運算子可被映射至一相關方法呼叫，且可依據一運算子型樣定義該方法呼叫。因此，一運算式中任何給定的運算子可預期依循形式化運算子型樣。因此，運算子及整個運算式可依據預期的方法呼叫來表示。應認知到該等方法可為實例方法、靜態方法、虛擬方法或延伸方法。

依據上述說明，藉由增進類型流及強制運算子符合運算子型樣，則此處揭露之結構可逐步推定出運算式中每一子句的元素類型。例如，可藉由將一來源碼類型與運算子

結合而決定元素類型，且可針對該運算式中每一連續子句依次即時完成此程序。因此，可區域性解決類型資訊而不需完整轉譯出整個運算式。因此，元素類型的類型核對可以較經濟(就資源利用性來說)、較快速的方式來執行，且在運算式建構期間即可達成，而不是僅能在運算式完成後達成，如此可增進額外優勢。

依據本發明申請標的之一態樣，此等額外優勢之一為可利用自動完成機制於運算式中。例如，藉由推定出作業中之元素類型，可提供依據該取得類型之上下文資訊，其可幫助運算式建構。

以下說明及附圖將詳細揭露本發明申請標的之某些範例性態樣。然而，這些態樣只表示各種不同方法中一些方法，在該等方法中利用了本發明申請標的之原理，且本發明申請標的意圖包含所有此等態樣及其等同態樣。透過以下對本發明申請標的之詳細說明並同時考慮所附圖示，本發明申請標的之其它優點及區別特徵將係顯而易見。

【實施方式】

現將參照所附圖示說明本發明申請標的，其中相同元件符號在所有圖示中係指相同元素。為了說明目的，以下說明中所提到的諸多特定細節係為了能完整說明本發明申請標的。然而明顯地，即使在缺乏這些特定細節下亦可實施本發明申請標的。在其它例子中，習知結構及裝置以方塊圖形式示出，以增進說明本發明申請標的。

此說明書中之用詞「組件」、「模組」、「系統」、「介面」等，其通常係指電腦相關實體，如硬體、軟硬體結合、軟體或執行中軟體。例如，一組件可為(但不限制條件)在一處理器上執行的一程序、一處理器、一物件、一可執行物件、一執行緒、一程式及/或一電腦。舉例來說，在一控制器上執行的一應用程式及該控制器二者可為一組件。一或多組件可配置在一程序及/或執行緒內，且一組件可被區域地配置在一電腦及/或分佈在二或更多電腦之間。在另一實例中，一介面可包括 I/O 組件及相關處理器、應用程式及/或 API 組件，且其可簡單如一命令行或可為一較複雜整合發展環境(Integrated Development Environment, IDE)。

此外，本發明申請標的可實現為一方法、設備或製造物，其使用標準程式及/或工程技術來產生軟體、韌體、硬體或其任何組合，以控制一電腦實施本發明揭示之申請標的。其中該用詞「製造物」係欲包含任何電腦可讀裝置、載體或媒體可存取的一電腦程式。例如，電腦可讀媒體可包括(但不限於此)磁性儲存裝置(如硬碟、軟碟、磁圈…)、光碟(如光碟(CD)、數位影音光碟(DVD)…)、智慧卡及快閃記憶體裝置(如記憶卡、記憶棒、隨身碟…)。此外，應認知到一載波可利用來搭載電腦可讀電子資料，如那些被用來傳送及接收電子信件的載波，或者利用來存取諸如網際網路或區域網路(local area network, LAN)的網路。當然，本領域習知技藝人士將了解到，在不背離本發明申請標的範疇或精神下可針對此種配置進行各種修改。

此外，此處之用詞「範例性」(“exemplary”)係指作為一範例、實例或說明例。在說明書中被稱為「範例性」之任何態樣或設計並不能視為較佳實施例或優於其它態樣或設計。此外，使用該字「範例性」係意圖以具體形式來表示本發明之概念。此說明書中之用詞「或」係指「包含或」而非「互斥或」。也就是說，除非有特定指示或由上下文可清楚分辨，「X 利用 A 或 B」係指任何的包含選擇。即若 X 利用 A；X 利用 B；或 X 利用 A 及 B 兩者，則「X 利用 A 或 B」可滿足上述三種情形之任一種。此外，說明書及所附的申請專利範圍中之冠詞「一」(“a”、“an”)一般應視為其係指「一或多數」，除非有特別指出或由上下文可明顯判定其係指一單數形式。

此說明書之用詞「推定」(“infer”或“inference”)一般係指依據多個事件及/或資料所得到的一觀察集合來推定(“reasoning about”或“inferring”)一系統、環境及/或使用之狀態之程序。可利用推定來識別一特定上下文或步驟，或者例如可產生狀態之機率分佈。推定可為機率的—即基於考慮資料及事件，對有興趣之狀態其機率分佈之計算。推定亦可指一種技術，其用於從一事件集合及/或資料中構成較高階級事件。此類推定可從一觀察到的事件集點及/或所儲存的事件資料中建構新事件或動作，不論該等事件是否在鄰近時間內相關且不論該等事件及資料係來自一或數個事件及資料來源。

現在依據附圖作說明，首先參照第 1 圖，此圖示出一

電腦可實施系統 100，其可利用查詢型樣來促進元素類型之類型流。一般情況下，系統 100 可包括一使用者介面 102，其可操作地耦接至習知電腦應用之硬體(例如控制器)、軟體(例如應用程式)及其它此處所提之組件。使用者介面 102 可接收一查詢運算式 104，其可包括一第一查詢子句及一或多數下一查詢子句。需知該第一查詢子句可包括一可查詢來源碼類型(例如一集合、一串流等)及一元素類型，其可由該第一查詢子句引進作為(例如)與該來源碼類型有關之控制變數。一般情況下，所有查詢子句包括一查詢運算子，然而將可發現，查詢運算式 104 之該等下一查詢子句不必係完整，而可以部份形式接收(例如逐步增加地)，如當使用者正輸入該子句時。以下將參照第 2A 及 2B 圖，更詳細地說明查詢運算式 104 及相關子組件。

系統 100 亦可包括任何有效查詢運算子用的一查詢運算子型樣 106(例如關聯第一查詢子句及接續的下一查詢子句之查詢運算子)。因此，任何數目之查詢運算子型樣 106 可存在於(例如)一型樣儲存 108 中，型樣儲存 108 可操作地耦接至系統 100。特別是，來自查詢運算式 104 之該等查詢子句(例如每一子句皆具有一特定查詢運算子)可符合相關的查詢運算子型樣 106，其可定義例如可存取實例方法。藉由符合查詢運算子型樣 106，可大大保證在一查詢子句與下一查詢子句間存在已知關係，且因此存在由一查詢運算子至該下一查詢運算子的關係。將參照第 3 圖更詳細說明查詢運算子型樣 106。

應了解到藉由對查詢運算式 104 設限制條件，而使得查詢運算式 104 之多個運算子符合查詢運算子型樣 106，系統 100 可確保由查詢運算式 104 之一查詢子句/運算子流至下一查詢子句/運算子之元素類型的類型流。此外，藉由確保元素類型如此處所述方式流動，某些優勢類型推定及/或上下文判定可在作業中實施，而毋需對完整查詢運算式 104 作習知轉譯/編譯，如參照第 4 圖所作的詳細說明。此外，前述推定及/或判定可在查詢運算式 104 完成前或甚至當查詢運算式 104 變形時提供。作為進一步說明但不限於此，以下將更詳細說明一符合查詢運算子型樣 106 的實例查詢運算式 104 及關聯此系統化的某些優勢。

參照第 1 圖的同時亦參照第 2A 圖，此圖說明一範例性查詢運算式 104。一般情況下，查詢運算式 104 可為一將一系列查詢運算子應用至一特定集合的運算式。一查詢運算子為一可立即應用到整個集合中一值集合的運算子(例如，FROM、WHERE、SELECT...)。查詢運算式 104 可包括一第一查詢子句 202 及任何數目的下一查詢子句 204₁、204₂ 等，每一查詢子句包含該等查詢運算子之一。

在許多其它查詢運算子可存在且被視為在本發明申請標的之精神及範籌內的情況下，以下將針對每一查詢運算子提供多個具體實施例及簡短說明。

- FROM 運算子可引進一或多數控制變數及指定一待查詢之集合或計算控制變數之值。

- RETURN 及 SELECT 運算子可指定輸出集合之形式。在某些情況下，SELECT 運算子可引進新的控制變數。
- WHERE 及 DISTINCT 運算子可限制集合之值。
- ORDERBY 運算子可強制集合之次序。
- SKIP、SKIPWHILE、TAKE 及 TAKEWHILE 運算子可依據次序或條件傳回一集合之一子集。
- UNION、UNIONALL、EXCEPT 及 INTERSECT 運算子可接收二個集合且結合產生一單一集合。
- GROUP 運算子可聚集一集合以及可傳回藉由一金鑰分類之集合。
- GROUPBY 運算子可依據一或多數金鑰分類集合。在某些情況下，此 GROUPBY 運算子可引進新的控制變數。
- AVG、SUM、COUNT、MIN 及 MAX 運算子可聚集一集合及產生一值。
- ANY 及 ALL 運算子可聚集一集合且依據條件傳回一 BOOLEAN 值。
- JOIN 運算子可接收二個集合且可依據取自元素的匹配金鑰產生單一集合。
- GROUPJOIN 運算子可依據擷取自元素的匹配金鑰執行二個集合之分類結合。

一查詢運算式 104 中查詢運算子係以該等運算子由左至右在查詢運算式 104 中的次序而被依序轉譯。典型地，

由於 FROM 運算子引進一待查詢之集合，因此查詢運算式 104 中第一查詢子句 202 包括 FROM 運算子。此外，查詢運算式 104 中最後查詢子句一般包括一 RETURN 及 SELECT 運算子，其指定由該查詢產生的集合最終形式。然而，應瞭解到最終 SELECT 或 RETURN 運算子可加以忽略。例如，若該查詢運算式 104 並未以一 RETURN 或 SELECT 運算子作為結束，則可假定有一隱式 SELECT 運算子，其可將範圍內所有控制變數作為傳回的匿名稱類型之特性。此外，應瞭解到在一 SELECT 運算子被應用在隨後之運算子可存取的控制變數可僅被限制在範圍內的那些控制變數的情況中之後，查詢運算式 104 可繼續。

依據本發明申請標的之一態樣，查詢運算式 104 可系統化如下：

QueryExpression ::= QueryOperatorList

QueryOperatorList ::=

FromOperator |

QueryOperatorList QueryOperator

QueryOperator ::=

FromOperator |

SelectOperator |

ReturnOperator |

DistinctOperator |

WhereOperator |

OrderByOperator |

PartitionOperator |

SetOperator |

GroupByOperator |

JoinOperator |

GroupJoinOperator

為了提供本發明申請標的的額外上下文及相關程式實現及/或語言的高水準概述，考慮範例性查詢運算式 104 之整體：

```
Dim names = _
    From C In Customers _
    Where C.State = "WA" _
    Select C.Name
```

此查詢運算式 104 可使用一客戶 (Customer) 物件集合，且針對在華盛頓 (Washington) 的每個客戶傳回具有單一名稱 (Name) 特性的一列集合。

明顯地，查詢運算式 104 語法可相當接近標準關聯結構查詢語言 (Structure Query Language, SQL) 語法，其意圖讓任何熟悉 SQL 的人能在有限的進一步指示下，將可使用查詢運算式 104。然而，語法不需要被 SQL 限制住，且另外，查詢運算式 104 不需要是由 SQL 轉譯成的其它程式語言。因為 SQL 係設計迴避純關連式模型，所以它的某些用語在一包含階層式概念的類型系統中無法正常運作。此外，某些語法及語義元素常被 SQL 衝突利用，或其無法與現行程式語言語法或語義良好整合。因此，當查詢運算式 104 對於熟悉 SQL 的使用者來說可為直覺式時，亦可存在某些區別。

例如，在 SQL 及在諸如程序語言/結構查詢語言 (Procedural Language/Structured Query Language, PLSQL) 的其它語言中，查詢及該等查詢之結果藉由評估該查詢全體而結合起來。如此，不需要能夠轉譯所應用的特定查詢運算子的元素類型。此外，需等到藉由相當昂貴的轉譯程序輸入並結合整個、無錯誤查詢後才可作出此一判定。相反的，本發明申請標的可依據一特定來源碼類型上個別查詢操作，而用於連結查詢運算式 104 輸入及查詢運算式 104 之結果。這些及其它優點可參照第 2B 圖作更詳細說明。

第 2B 圖更詳細示出範例性查詢運算式 104 的查詢子句。在此例中，第一查詢子句 202 包括一第一查詢運算子 (例如，FROM 運算子 206)，其可用來引進一可查詢來源碼類型 (例如，Customers 208)，且可宣告一用於來源碼類型之元素類型的控制變數 210 (例如，C)。如圖所示，第一查詢子句 202 之結果流 212 至下一查詢子句 204₁ (此處全體地是指下一查詢子句 204)，如此一習知關係可存在於鄰近查詢子句間。特別是，元素類型可流 212 至下一查詢運算子 (例如，WHERE 運算子 214)，以作為下一查詢子句 204₁ 之一來源碼類型。同樣的，下一查詢子句 204₁ 之輸出流 212 至關聯下一查詢運算子的下一查詢子句 204₂ (例如，SELECT 運算子 216)，其亦可產生一流 212 至隨後之下一查詢子句 204 的元素類型，或可產生一說明查詢運算式 104 之輸出的最終形式之元素類型。

特別是，每個下一查詢子句 204 的查詢操作可依據一

方法集(例如,查詢運算子型樣 106),每一方法關聯一特定查詢運算子,且其應用至一集合或序列(例如,來源碼類型)。可定義這些查詢運算子係有關於該集合的元素類型(或列)

然而,在給定具有元素類型 T 的一特定來源碼類型及一查詢運算子之特定應用情況下,流 212 至下一查詢運算子 204 之結果可為一具有一元素類型 S 的集合。因此,元素類型流 212 可依據關聯一給定查詢運算子的查詢運算子型樣 106 進行轉換。例如,關聯諸如 SELECT 及 RETURN 之類的投影查詢運算子的查詢運算子型樣 106,其一般會傳回與接收自流 212 的元素類型不同的元素類型。

在此例中,一元素類型 T(例如,其可為關聯 Customer 類型 208 的所有控制變數 210 的一聚集)流 212 至下一查詢子句 204₁ 的 WHERE 運算子 214。如參照第 3 圖將說明的,關聯 WHERE 運算子 214 的查詢運算子型樣 106 可篩選集合值,但一般並未改變元素類型。在這裡,不在華盛頓州的客戶(Customer)從集合中被篩選掉,但流 212 至下一查詢運算子的結果(即 SELECT 運算子 216)仍然是一具有 Customer208 元素類型的集合。

相反的,若假定名稱(Name)性質屬於 String 類型,則 SELECT 運算子 216 之結果為具有一元素類型 String 的一集合。結合支援用於查詢結果的 SELECT 及 RETURN 運算子二者,其可提供輸入及輸出的可預測語義至運算子。此外,當與關連式資料互動時,可增進更多彈性以更加類似

於期望的 SQL 語義。此外，當產生非表定結果時可增進簡單性，以及當重覆發展一查詢運算式時，可改善編譯器錯誤地區及所需要的改變。

此外，對於可查詢可擴展標記語言 (eXtensible Markup Language, XML) 資料 (例如，Xquery，其為包括有某些程式語言特徵的一查詢語言) 的特定查詢語言來說，本發明申請標的亦可與 XML 語言一起作用。例如，以下的範例性查詢運算式 104 可用來傳回包括所有華盛頓客戶名稱的一 XML 元素集合，其結果如同前文所說明的範例性查詢運算式 104 之結果。

```
Dim names = _
    From c In Customers _
    Where c.State = "WA" _
    Return <Name><%= c.Name %></Name>
```

更需認知到，某些查詢運算子如 FROM、SELECT 及 GROUPBY，可引進稱為控制變數的一特別類型區域變數 (如控制變數 C210)。預設上，一控制變數之範圍可從引進運算子到一運算子，該運算子當查詢評估時可隱藏控制變數及可表示一集合中個別列的性質或欄位。例如，以下查詢中：

```
Dim WACusts = _
    From C As Customer In Customers _
    Where C.State = "WA"
```

FROM 運算子引進分類為 Customers 的一控制變數。

可將一系列查詢運算子應用到一特定集合。為了應用該特定查詢運算子，每一查詢運算子之控制變數可為範圍內所有變數或所有變數之一子集合；而元素類型 T(其可由一查詢運算子流至下一查詢運算子)可為一範圍內所有控制變數聚集成的一匿名類型。因此，查詢運算子型樣 106 尤其可增進確保一類型 C 為一可查詢類型。在任一給定查詢運算式 104 情況下，若以下至少一者為真，則類型 C 為一可查詢類型：

- (1) 類型 C 包括一具有簽章 `AsQueryable()` 的可存取實例方法，其可傳回一可查詢類型。
- (2) 類型 C 包括一具有簽章 `AsEnumerable()` 的可存取實例方法，其可傳回(T 的) `IEnumerable`。
- (3) 類型 C 實現符合查詢運算子型樣 302-328(以下將詳細說明)的一或多查詢運算子，且針對所有定義在類型 C 上的查詢運算子，類型 T 必須匹配且必須不為一開放類型參數。

參照查詢運算子型樣 302-328，查詢運算子型樣 302 可包括 FROM 運算子。在查詢運算式 104 內，FROM 運算子典型地用於第一查詢子句中。如此，FROM 型樣 302 可引進查詢來源碼及待使用的控制變數，且可輸出具有元素類型 T 的一可查詢類型。應瞭解到，在評估查詢運算式之前，若類型 C 不滿足設計型樣(如不符合查詢運算子型樣 106)，則在查詢運算式 104 上可引起 `AsQueryable` 或 `AsEnumerable` 方法，且函式之傳回值可被儲存在一暫時位

置。

如前所述，FROM 運算子可引進一可被查詢之集合，及可代表該集合個別成員之一控制變數。查詢運算式：

```
From b As Book In Books ...
```

可視為等效於：

```
For Each b As Book In Books
```

```
...
```

```
Next b
```

集合運算式運算元一般必須歸類為一值且必須為一可查詢類型。由 FROM 運算子宣告的控制變數一般必須遵守正式規則，該規則用於宣告區域變數，包括命名及定範圍(如前述語法轉譯所指示)。因此，一控制變數一般無法隱藏所述方法中一區域變數之名稱或參數。

應了解到 FROM 運算子亦可引進一控制變數，其值可由一運算式而非一集合決定。對例如計算一值，該值在之後的查詢運算子中將被多次使用多次之程序來說，此一特徵可為有用的。(例如，計算該值一次而不需其後每次使用到時要再計算它)。例如：

```
Dim TaxedBookPrices = _
  From b in Books _
  From Tax = b.Price * 0.088 _
  Where Tax > 3.50 _
  Select b.Price, Tax, Total = b.Price + Tax
```

可視為等效於：

```
For Each b In Books
  Dim Tax = b.Price * 0.088
```

...
Next b

宣告一運算式控制變數的 FROM 運算子之語法可與 FOR LOOP 中一控制變數宣告相同，然而不同的是該控制變數一般係透過一顯式初始器進行初始化。一運算式控制變數不需要參照其它控制變數，因為如此做可能造成不確定值。一運算式控制變數通常無法作為查詢運算式 104 中所宣告的第一控制變數。

為了簡單及 / 或方便說明，FROM 運算子之運算元可省略 AS 子句，在此情況下控制變數之類型可由該變數所及範圍之集合或運算式推定。若控制變數之類型無法由查詢運算子方法取得，則將造成編譯-時間錯誤。FROM 運算子可概略定義如下：

FromOperator ::=

From FromDeclarationList

FromDeclarationList ::=

FromDeclaration |

FromDeclarationList, FromDeclaration

FromDeclaration ::=

VariableIdentifier [As TypeName] In Expression |

VariableDeclarator

此外，當 FROM 運算子一般以第一查詢子句 202 的查詢運算子出現時，一查詢運算式 104 可包括一個以上的

FROM 運算子。因此，當一 FROM 運算子為下一查詢子句 204 的查詢運算子時，則可產生一交叉乘積(例如，簡單隱式結合)。在此情況下，FROM 運算子可結合(見後文之 JOIN 運算子型樣 324)新的控制變數與現存的控制變數集。其結果可為在經結合的集合中的所有元素的交叉乘積。因此運算式(例如)

```
From b In Books _
From p In Publishers _
...
```

可視為等效於巢狀 For Each 迴圈

```
For Each b In Books
  For Each p In Publishers
    ...
  Next p
Next b
```

被引進至先前查詢運算子中的控制變數可在範圍內且可用於一包括在下一查詢運算子 204 中的 FROM 運算子內。依據 SQL，前述特徵可視為支援「相關子查詢(“correlated subqueries”)」。例如，以下查詢運算式中，第二 FROM 運算子係指第一控制變數之值：

```
From c As Customer In Customers _
From o As Order In c.Orders _
Select c.Name, o
```

此外，多個 FROM 運算子可出現在單一查詢子句中，在此情況下運算元間之逗號可完全等效於下一查詢子句 204 中之另一 FROM 運算子。因此，此實例：

```
From b In Books, p In Publishers _
Select b, p
```

可等效於

```
From b In Books _
From p In Publishers _
Select b, p
```

查詢運算子型樣 304 可包括一查詢運算子，其為被稱為 SELECT 或 SELECTMANY 的一可存取實例方法，其可接收一函式作為一參數。該函式可包括類型 T 之一參數，且產生一類型 S。此方法可傳回一具有元素類型 S 的可查詢類型。

SELECT 運算子可描述查詢運算式 104 之結果。SELECT 運算子可採用一宣告表單且建構該生成集合之一元素類型。例如，若該查詢運算式 104 為：

```
Dim CustAndOrderNames = _
    From c In Customers, o In c.Orders _
    Select c.Name, o.Product
```

則結果類型可為 IEnumerable(屬於 {Name As String, Product As String})，這是因為 Name 屬於類型及 Product 屬於 String 類型。

跟隨在後的此查詢(例如，查詢運算式 104)可等效於顯式地傳回一匿名類型：

```
Dim CustAndOrders = _
    From c In Customers, o In c.Orders _
    Return New With { c.Name, o.Product }
```

若 SELECT 運算子僅有一宣告，則結果可為具有一種性質的一匿名類型。如此，由於 Name 係屬於 String 類型，因此結果類型可為 IEnumerable (屬於 {Name As String})。例如，

```
Dim CustNames = _
    From c In Customers
    Select c.Name
```

用於一 SELECT 運算子中的宣告內的複數運算式，其在轉換至基本的匿名類型前係係連結在一起。更具體地說，一 SELECT 運算子中宣告內的運算式可開始進行一成員存取，其中乘積(dot)運算子不需要連結至查詢中之控制變數。此外，該乘積運算子反而可連結至一附加的 WITH 區塊(如果存在的話)。

應認知到除了 SELECT 運算子，查詢運算式 104 亦可包括其它投影運算子。例如，查詢運算式可包括一 RETURN 運算子，可存在一與其相關之查詢運算子型樣 106。類似 SELECT 運算子，RETURN 運算子可描述查詢運算式 104 之結果。RETURN 運算子可採用產生所生成集合之元素之一運算式。例如，該查詢：

```
Dim ReducedBookPrices = _
    From b in Books _
    Return b.Price * .8
```

可造成一價位集合，該等價位已被減低至其原始成本 80%。

以 RETURN 運算子作為結束的查詢運算式 104，其結果可為一集合，該集合之元素類型為經傳回的運算式之類型。例如，在以下查詢運算式中，由於 c.Name 係屬於 String 類型，因此結果類型為 IEnumerable (屬於 String)：

```
Dim CustNames = _
    From c In Customers _
    Return c.Name
```

若一查詢運算式 104 結束時沒有 RETURN 或 SELECT 運算子，則該集合之生成元素類型可為一具有範圍內所有控制變數之性質的一匿名類型：

```
' Result type is IEnumerable(Of {Name As String, Product As String})
Dim CustNames = _
    From c In Customers, O In C.Orders
```

即使一 SELECT 運算子可描述一查詢運算式 104 之結果，任一查詢運算式 104 可接續在一 SELECT 運算子之後。在此情況下，由 SELECT 敘述引進的控制變數為範圍內之控制變數，但所有先前控制變數一般為範圍外：

```

CustNames = _
  From c In Customers, O In C.Orders _
  Select Name = C.Name, Price = O.Price
  Where Price > 500
  Return Name

```

接續在一 SELECT 運算子之後的一查詢運算式 104，其可等效於一巢狀查詢：

```

Dim CustNames = _
  From X In (From c In Customers, O In C.Orders _
    Return New With { C.Name, O.Price }), _
  Name = X.Name, Price = X.Price
  Where Price > 500
  Return Name

```

RETURN 及 SELECT 運算子可分別由以下例子來定義：

```

ReturnOperator ::=
  Return Expression

```

```

SelectOperator ::= Select SelectDeclarationList

```

```

SelectDeclarationList ::=
  SelectDeclaration |
  SelectDeclarationList , SelectDeclaration

```

```

SelectDeclaration ::=
  VariableDeclarator

```

查詢運算子型樣 306 可包括一查詢運算子，其係一名為 WHERE 的可存取實例方法，其採用一函式為參數。T 類型可為該函式之一參數，且結果可在一可隱式轉換至布林類型之類型中。例如，結果類型可具有與可用於 IF、WHILE 或 DO 敘述的一運算式之類型相同的規則。因此，類型可具有一隱式轉換至布林類型，或者 IsTrue 及 IsFalse 運算子可能需要事先定義。WHERE 運算子可傳回一具有元素類型 T 的可查詢類型。

WHERE 運算子將集合中之值限制在滿足一給定條件之值。WHERE 運算子可接收一布林運算式，針對每一控制變數值集合評估該布林運算式。若運算式之值為真 (True)，則該等值可出現在輸出集合中，否則可略過該等值。查詢運算子：

```
From b In Books, p In Publishers _
Where b.PublisherID = p.PublisherID _
...
```

可視為等效於巢狀迴圈

```
For Each b In Books
  For Each p In Publishers
    If b.PublisherID = p.PublisherID Then
      ...
    End If
  Next p
Next b
```

WHERE 運算子可定義成：

WhereOperator ::= Where BooleanExpression

查詢運算子型樣 308 可包括一查詢運算子，其係一名為 ORDERBY 或 ORDERBYDESCENDING 的可存取實例方法且採用一函式為參數，其可輪流採用類型 T 之參數且產生類型 S。關聯型樣 308 之方法可傳回一具有元素類型 T 的次序集合。

ORDERBY 運算子可依據一次序將出現在控制變數中之值進行排序。ORDERBY 運算子可採用一運算式，該運算式指定應該用來排序控制變數的值。例如，以下查詢傳回依價格分類的書目：

```
From book In Books _
Order By book.Price
Select book.Title
```

次序可為升冪，在此情況下較小值排在較大值之前，或可為降冪，在此情況下較大值排在較小值之前。預設為升冪次序。例如，以下查詢傳回依價格分類的書目，且最貴的書排在最前面：

```
From book In Books _
Order By book.Price Descending
Select book.Title
```

ORDERBY 運算子亦可指定多個運算式用於進行排序，在此情況下集合可以一巢狀方式進行排序。例如，以下查詢運算式首先依據州對作者進行排序，接著依據各州之城市進行排序，及再依據各城市之 ZIP 碼進行排序：

```

From author In Authors _
Order By author.State, author.City, author.ZIP _
Select author.Name, author.State, author.City, author.ZIP

```

ORDERBY 運算子可由以下例子進行系統化：

OrderByOperator ::= Order By *OrderExpressionList*

OrderExpressionList ::=
OrderExpression |
OrderExpressionList , *OrderExpression*

OrderExpression ::=
Expression [*Ordering*]

Ordering ::= Ascending | Descending

此外，查詢運算子型樣 310 可包括一查詢運算子，其係一名為 DISTINCT 的可存取實例方法。典型地，此方法傳回一具有與來源類型 C 相同的元素類型的一可查詢類型。DISTINCT 運算子可將集合中之值限制為不同值（例如不傳回重覆值）。例如，查詢：

```

From c In Customers, o In c.Orders _
Select c.Name, o.Price _
Distinct

```

其將僅傳回一系列值，其包括每一對相異的客戶名稱及價格次序，即使在相同價格下之客戶具有多種次序。DISTINCT 運算子可系統化為：

DistinctOperator ::= Distinct

現在參照查詢運算子型樣 312，型樣 312 可包括一查詢運算子，其係一名為 CONCAT、UNION、INTERSECT 或 EXCEPT 的可存取實例方法。此方法可接收具有與來源碼 C 相同的元素類型 T 的一集合作為一參數，且可傳回一具有元素類型 T 的可查詢類型。

下一查詢運算子型樣，型樣 314 可包括一查詢運算子，其可為一名為 TAKE 或 SKIP 的可存取實體方法，該方法可接收一值作為參數，且其可傳回一具有與來源碼類型 C 相同的元素類型的可查詢類型。

TAKE 運算子可自一集合產生一給定數量的元素。當與 WHILE 修飾語一起使用時（見例如，查詢運算子型樣 316），在條件保持為真的同時，TAKE 運算子可產生一系列元素。

SKIP 運算子可忽略來自一集合的一給定數量的元素且接著傳回該集合中之其它剩餘者。當與 WHILE 修飾語一起使用時，在條件保持為真的同時，SKIP 運算子略過元素，且接著傳回該集合中其它者。一範例性 TAKE 及 SKIP 方法提供如下。

PartitionOperator ::=
 TakeExpression |
 SkipExpression

TakeExpression ::=

Take[While] *Expression* |

SkipExpression ::=

Skip [While] *Expression*

查詢運算子型樣 316 可包括一查詢運算子，其係一名為 TAKEWHILE 或 SKIPWHILE 的可存取實體方法。此方法可採用一函式為參數，其中該函式可接收類型 T 之一參數，且產生一布林邏輯值。此方法可傳回一具有元素類型 T 的可查詢類型。

查詢運算子型樣 318 可包括一查詢運算子，其係名為 SUM、MIN、MAX、COUNT 或 AVERAGE 的一可存取實體方法。此方法可接收一函式作為參數，其交替接收一類型 T 之參數及產生一數字類型。此方法可傳回一數字類型。

查詢運算子型樣 320 可包括一查詢運算子，其係一名為 MIN 或 MAX 的可存取實體方法，此方法可採用具有類型 T 之參數的一函式為參數且產生一類型 S。此方法可傳回一具有元素類型 S 的可查詢類型。

查詢運算子型樣 322 可包括一查詢運算子，其係一名為 ANY 或 ALL 的可存取實體方法，其可接收一函式作為參數。該函式可包括類型 T 之一參數，且產生一類型其可隱式轉換至布林。此方法可傳回一類型其可隱式轉換至布林邏輯值。

應了解到關聯型樣 318-322 的查詢運算子可與一 AGGREGATE(聚集)運算子一起使用。在一 Aggregate 查詢

子句內，標準 AGGREGATE 運算子集可應用到範圍內的經分組控制變數。AGGREGATE 運算子可包括但不限於：ANY、ALL、COUNT、LONGCOUNT、SUM、MIN、MAX、AVERAGE 及 / 或 GROUP。

ANY 聚集運算子可確定群組中是否有一元素滿足一給定條件。ALL 聚集運算子可決定是否群組中所有元素滿足一給定條件。例如，以下的範例性查詢運算式 104 可確認是否有任何客戶年齡低於 18 歲：

```
From cust In Customers _
Group By cust.State _
Aggregate YoungerThan18 = Any(cust.Age < 18)
```

而以下實例查詢運算式 104 可傳回在一給定狀態中具有至少 5 種訂單的所有客戶：

```
From cust In Customers _
Group By cust.State _
Aggregate AtLeast5 = All(cust.Orders.Count() > 5)
```

COUNT 及 LONGCOUNT 聚集運算子可採用一選擇性布林運算式，且計算群組中滿足該給定條件的元素數量。

```
From cust In Customers _
Group By cust.State _
Aggregate Seniors = Count(cust.Age > 50)
```

SUM 聚集運算子可依據一特定選擇器運算式來計算一組元素之和。例如，以下範例性查詢運算式 104 可計算所有依種類分組的所有訂單值之總和：

```
From order In Orders _
```

```

Group By order.Category
Aggregate Total = Sum(order.Price)

```

MIN 及 MAX 聚集運算子可依據某些選擇器運算式來計算出一組元素的最小(或最大)值。例如，以下實例查詢運算式 104 可針對每一州計算出最年輕及最年長的顧客：

```

From cust In Customers _
Group By cust.State
Aggregate Oldest = Max(cust.Age), Youngest = Min(cust.Age)

```

AVERAGE 聚集運算子一般可依據一特定選擇器運算式來計算一組元素之平均值。例如，以下實例查詢運算式 104 可計算出依每一種類分組的所有產品的平均價格：

```

From prod In Products _
Group By prod.Category
Aggregate AveragePrice = Average(prod.Price)

```

特殊 GROUP 聚集運算子可依據一選擇性選擇器運算式，將各組別之所有元素累積至一顯式集合。例如，以下範例性查詢運算式 104 可將一特定狀態下之所有顧客名稱聚集成一單一集合：

```

From cust In Customers _
Group By cust.State
Aggregate Names = Group(cust.Name)

```

查詢運算子型樣 324 可包括一稱為 JOIN 的可存取實例方法的查詢運算子。該方法可採用一具有類型 T 之元素的可查詢類型 S 作為參數；一函式，其作為一內部選擇器，

其可採用類型 T 之參數及產生代表一金鑰 K 之類型；一函式，其作為一外部選擇器，其可採用類型 T 之參數及產生代表該金鑰 K 之類型；及/或一函式，其作為結合條件，其分別採用二種類型 T 及 T' 以及產生類型 V 之一雜湊值。此方法可傳回一具有一元素類型 V 的可查詢類型。

因此，JOIN 運算子可採用二個集合並依據取自元素的匹配金鑰產生單一集合。某些限制可應用於當指定結合條件時。例如，至條件運算式的運算元可能需要可隱式轉換至 Boolean 邏輯值。以下為 JOIN 方法的一實例：

JoinOperator ::=

*Join LoopControlVariable [As TypeName] In Expression Where
JoinConditionExpression*

JoinConditionExpression ::=

*RelationalOperatorExpression |
LikeOperatorExpression*

查詢運算子型樣 326 可包括一查詢運算子，其係一名為 GROUPJOIN 的可存取實例方法，該方法可接收具有類型 T' 元素的一可查詢類型 S 作為參數；一函式，其作為採取類型 T 之參數及產生代表金鑰 K 之類型的外部鍵選擇器；一函式，其作為採取類型 U 之參數及產生代表金鑰 K 之類型的內部鍵選擇器；及/或一可產生該結果之函式，其中該函式可採取類型 T 之參數，及一參數其為一具有元素類型 U 的一集合，其產生一經選擇類型 V。此方法可傳回

一具有元素類型 V 的可查詢類型。

GROUPJOIN 運算子可依據取自元素的匹配金鑰，針對二個集合產生一分組結合。運算子可產生階層式結果(例如，與匹配內部元素之集合配成對的外部元素)且在關聯式資料庫項目中不需要直接對等物。以下實例查詢運算式 104 可依據客戶訂單對客戶執行分組結合，產生一具有客戶名稱及客戶訂單總值的匿名稱類型之集合：

```

From cust In db.Customers _
Group Join order In db.Orders On cust.ID = order.CustID _
Aggregate TotalOrders = Sum(order.Total) _
Select cust.Name, TotalOrders

```

可有任意數量的額外運算子，其在固定集上發生作用以計算單一值。在查詢上下文中，可以說這些運算子聚集經計算值。要聚集一集合，一般必須先引進該集合。在一查詢中，可有各種不同方式以指定一集合，用以聚集計算。在引進一群組後，該群組可被聚集以計算單一值。因此，除了分組運算子 GROUPJOIN(及下文中與查詢運算子類型 328 相關討論的 GROUPBY)，其它分組運算子亦可存在。其中一例為 OVER 運算子。

OVER 運算子可具體指出應該累計一分組用於聚集理解。OVER 運算子可用作一獨立運算式，或其可用來具體指出一預先形成集合應該用於聚集計算。例如，以下實例查詢運算式 104 可聚集 2006 年 1 月 1 日前的訂單總和：

```

Over order In Orders _
Where order.Date <= #01/01/2006# _
Aggregate Sum(order.Total)

```

OVER 運算子的每一應用一般必須由一對應 AGGREGATE 運算子結束，以依據集合內容計算單一值。查詢結果可為符合聚集計算的單一值，在此一例中其為一整數 (Integer)，其代表 2006 年之前的所有訂單的總和。

在 OVER 及 AGGREGATE 運算子之間可能可以使用任何查詢運算子。在 AGGREGATE 運算子中，只有 AGGREGATE 運算子 (例如，如以上說明的關聯型樣 318-322 的運算子) 可用在由先前 OVER 運算子所指定的控制變數上。

當一集合已形成，如同常見的具有階層式物件圖，OVER 運算子可用於具體指出一集合應該用於聚集計算。例如，以下範例性查詢運算式 104 可聚集 2006 年之前華盛頓客戶所有訂單之總和：

```

From cust In db.Customers _
Where cust.State = "WA" _
Over order In Orders _
Where order.Date <= #01/01/2006# _
Aggregate Sum(order.Total)

```

查詢結果可為一可查詢類型，其元素類型係一具有二個性質的匿名類型：1) 稱為 cust 的一 Customer 性質及 2) 稱為 Sum 的一整數 (Integer) 性質。

查詢運算子型樣 328 可包括一查詢運算子，其係一稱為 GROUPBY 的可存取實體方法，該方法採用一函式（其作為接收類型 T 之參數的金鑰選擇器，且產生代表該金鑰 K 的一類型）作為參數；及/或一產生該結果的函式，其採用類型 K 之參數，及一參數，其係具有產生一經選擇類型 V 的元素類型 T 的一集合。該方法可傳回具有元素類型 V 的一可查詢類型中之結果。

GROUPBY 運算子可依據一或多常用金鑰運算式分組（例如，邏輯地）一集合之元素。對於初始集合的這些分割中的每一分割來說，一隨後的 AGGREGATE 敘述可具體指出如何將這些群組中每一群組聚集成一單一值或列。

例如，以下範例性查詢運算式 104 可依據所在之州分組所有客戶，且接著計算每一群組的計數及平均年齡：

```
From cust In Customers _
Group By cust.State
Aggregate Total = Count(), AverageAge = Avg(cust.Age)
```

如同 SELECT 運算子，GROUPBY 運算子可引進範圍內金鑰選擇器中所宣告的變數，將其作為控制變數且可隱藏先前在範圍內的所有控制變數。相反的，不像 SELECT 運算子的某些特定態樣，這些被隱藏的控制變數可由 AGGREGATE 運算子再次引進至範圍中，且使用於隨後的 AGGREGATE 運算子內。此外，不像 OVER 運算子的某些實施方式，沒有其它查詢運算子可用於 GROUPBY 及 AGGREGATE 運算子之間，

```

From cust In Customers _
Group CustomersByState By cust.State

```

此建構可等效於藉由顯式聚集方法來建構分組，但可有效用於當在一 GROUPBY 運算子子句之後立即終止查詢時。因此，前述運算式可等效於下列運算式：

```

From cust In Customers _
Group By cust.State _
Aggregate CustomersByState = Group(cust)

```

GROUPBY 運算子的實際實施方式不需要實際上建立每一個別群組的代表。相反的，實施方式可使用一基本實施方式，其將具有隨後聚集之分組結合至一單一運算子內。

應認知到控制變數的類型及名稱可由查詢運算式 104 之標的及所施用的查詢運算子(其將於下文參照第 4 圖更詳細說明)來推定。因此，元素類型(其係範圍內控制變數的一聚集)可以同樣方式推定。對於前述之每一規則，所推定的元素類型之類型可為 T，且當可查詢類型 C 為用於例如 FOR EACH...NEXT 敘述中的集合時，元素類型一般必須匹配元素類型 T。

應更認知到一經排序集合可定義為一實現下列運算子之一子集合的一集合：一名為 THENBY 或 THENBYDESCENDING 的可存取實例方法，其採用一函式為參數，該函式採用類型 T 之參數且產生類型 S。此方法可傳回一具有元素類型 T 的經排序集合。

由於查詢運算子型樣 106 可幫助將查詢語法連結至可

實現一特定查詢運算的方法，因此次序保存不需要由所使用的基本語言來指定。更確切的說，次序保留可由運算子本身之實施方式決定。這可類似於使用者定義 (user-defined) 運算子，其中實施方式過載例如一使用者定義之數值類型的相加運算子，該實施方式可能不執行任何類似相加的運算。然而，為了保存本發明申請標的固有的可預測性，實現不符合使用者期望的運算子可能不是建議的方式。

為了幫助更加完整了解本發明申請標的，額外特徵、態樣及/或實施方式在此將作進一步說明。例如，以下某些節次將詳細說明每一查詢運算子映射至標準查詢運算子方法呼叫之範例性方法。其它節次將說明元素類型資訊如同前述介紹之方式流動的方法，且將於稍後之節次利用此方法 (例如，參照第 4 圖及下文之說明)。

查詢運算子轉譯

應瞭解到各種不同的查詢運算子之每一運算子可直接映射至方法呼叫，該等呼叫可依據一標準查詢運算子型樣 (例如，查詢運算子型樣 106) 進行定義。因此，一查詢運算子 (或查詢運算式總體) 之意義可依據待呼叫之查詢運算子方法來表示。此等方法可為被查詢物件之實例方法，或為物件之外的延伸方法。例如，查詢：

```
Dim names = _
    From c In Customers _
    Where c.State = "WA" _
```

```
Return c.Name
```

可等效於：

```
Dim names = Customers.Where(c => c.State).Select(c => c.Name)
```

複合控制變數

由於本發明申請標的可利用控制變數於此處所述之各種不同目的時，應認知到一終端使用者一般無法直接存取控制變數。某些查詢運算子如 WHERE 或 SELECT，其採用以在查詢中範圍內之控制變數為參數的運算式。這些運算式被表示為區域函式，其採用控制變數且傳回該運算式之結果。例如，查詢：

```
Dim WACusts = _
    From cust In Customers _
    Where cust.State = "WA"
```

可等效於：

```
Dim WACusts = Customers.Where(c => c.State = "WA") _
    .Select(c => New With {c.Name})
```

其中該控制變數 `cust` 用作 λ 運算式的參數。同樣的，在多控制變數係在範圍內的情況下，控制變數一般必須聚集成為一單一匿名類型，如此可將他們傳送至該區域函式。例如，查詢：

```
Dim WACustsAndOrders = _
    From cust In Customers _
```

```

From order In cust.Orders _
Where cust.State = "WA" AndAlso order.Price > 10.50

```

可等效於：

```

Function Filter1(it As {c As Customer, o As Order}) As Boolean
    Return it.c.State = "WA" AndAlso it.o.Price > 10.50
End Function

```

```

Function Join1(c As Customer) As _
    IEnumerable(Of {c As Customer, o As Order})
    Function Select1(o As Order) As {c As Customer, o As Order}
        Return New {c, o}
    End Function

```

```

    Return c.Orders.Select(AddressOf Select1)
End Function

```

```

Dim WACustsAndOrders = _
    Customers.SelectMany(AddressOf Join1).Where(AddressOf
    Filter1)

```

其中控制變數 c 及 o 被結合至一稱為 IT 之複合控制變數，其類型係一匿名類型 {c As Customer, o As Order}。

更認知到亦可提供支援 λ 運算式且可視該支援在本發明申請標的之精神及範疇內。進一步注意到，複合控制變數通常不為巢狀。舉例來說，

```

From b In Books _
From ba In BookAuthors _
From a In Authors _
...

```

可等效於：

```
Function Join1(b As Book) As IEnumerable(Of {b As Book, ba As BookAuthor})
    Function Select1(ba As BookAuthor) As {b As Book, ba As BookAuthor}
        Return New { b, ba }
    End Function
```

```
    Return BookAuthors.Select(AddressOf Select1)
End Function
```

```
Function Join2(it As {b As Book, ba As BookAuthor}) As _
    IEnumerable(Of {b As Book, ba As BookAuthor, a As Author})
    Function Select2(a As Author) As _
        {b As Book, ba As BookAuthor, a As Author}
        Return New {b, ba, a}
    End Function
```

```
    Return Authors.Select(AddressOf Join2)
End Function
```

```
Books.SelectMany(AddressOf Join1).SelectMany(AddressOf Join2)
```

FROM 運算子轉譯

一標準 FROM 運算子一般只引進一查詢之來源碼及待使用之控制變數。在本質上，其不轉譯成任何特定查詢運算子呼叫。在 JOIN 之情況中，當與一正規控制變數結合時，FROM 運算子可被轉譯成一 SELECTMANY 查詢運算子方法呼叫；而當與一運算式控制變數結合時，FROM 運算子被轉譯成一 SELECT 查詢運算子方法呼叫。因此查詢運算式：

```

From b In Books, p In Publishers _
Select b.Title, p.Name

```

可等效於：

```

Function Join1(b As Book) As IEnumerable(Of {b As Book, p As Publisher})
    Function Select2(p As Publisher) As {b As Book, p As Publisher}
        Return New With {b, p}
    End Function

```

```

    Return Publishers.Select(AddressOf Select2)
End Function

```

```

Function Select1(it As {b As Book, p As Publisher}) As _
    {Title As String, Name As String}
    Return New With {it.b.Title, it.p.Name}
End Function

```

```

Books.SelectMany(AddressOf Join1).Select(AddressOf Select1)

```

而查詢運算式：

```

From b In Books, Tax = b.Price * 0.088 _
Select b.Title, Tax

```

可等效於：

```

Function Select1(b As Book) As {b As Book, Tax As Double}
    Return New With {b, Tax = b.Price * 0.088}
End Function

```

```

Function Select2(it As {b As Book, Tax As Double}) As {Title As String, Tax As

```

```

Double}
    Return New With {it.b.Title, it.Tax}
End Function

```

```
Books.Select(AddressOf Select1).Select(AddressOf Select2)
```

每一查詢運算子可在附屬一特定型樣之來源碼類型上被連結至一基礎方法。此型樣指定運算子之結果之元素類型，且在某些情況下對可被傳送至該基礎方法的運算式設定限制。

顯式分類控制變數

若 FROM 運算子使用 As 子句為控制變數指定一目標類型物件，且 T 之類型無法由來源碼類型推定得，則元素類型可使用 Cast(OF Object) 運算子進行轉換。該查詢運算式：

```

Dim nums = _
    From addrNum As Object In publisher.Address _
    Where addrNum < 5

```

is equivalent to:

```

Function Filter1(addrNum As Integer) As Boolean
    Return addrNum < 5
End Function

```

```
Dim names = publisher.Address.Cast(Of Object)().Where(AddressOf Filter1)
```

若 FROM 運算子使用一 As 子句為控制變數指定一目

標類型 T(非 Object)，且 T 之類型無法由來源碼類型推定得，或若該目標類型不匹配來源碼之元素類型，則可使用 Cast 查詢運算子將元素類型轉換成 Object，且可利用一隨後之 SELECT 運算子以獲得目標類型。例如，查詢運算式：

```
Dim nums = _
    From addrNum As Integer In publisher.Address _
    Where addrNum < 5
```

is equivalent to:

```
Function Filter1(addrNum As Integer) As Boolean
    Return addrNum < 5
End Function
```

```
Function Select1(addrNum As Integer) As String
    Return addrNum
End Function
```

```
Dim names = publisher.Address.Cast(Of Object)().Select(AddressOf
Select1).Where(AddressOf Filter1)
```

若 FROM 運算子使用一 As 子句為控制變數指定一目標類型 T，且 T 之類型匹配來源碼之元素類型，則不需要隨後應用 Cast 或 SELECT 運算子。

WHERE 運算子轉譯

WHERE 運算子可轉譯成 Where 查詢運算子方法呼叫。例如查詢運算式：

```
From book In Books _
Where book.PublisherID = 10 _
```

```
Return book.Title
```

可等效為：

```
Books.Where(b => b.PublisherID = 10).Select(b => b.Title)
```

ORDERBY 運算子轉譯

ORDERBY 運算子在第一分類下可轉譯成 ORDERBY 或 ORDERBYDESCENDING 查詢運算子方法呼叫(依據分類類型)，而在隨後分類下可轉譯成 THENBY 及 THENBYDESCENDING 查詢運算子方法呼叫。例如查詢運算式：

```
From a In Authors _
Order By a.State, a.City Descending _
Return a.Name
```

可等效為：

```
Authors.OrderBy(a => a.State).ThenByDescending(a => a.City). _
Select(a => a.Name)
```

RETURN 及 SELECT 運算子轉譯

RETURN 運算子可轉譯成一 SELECT 查詢運算子方法呼叫。例如：

```
From e In Employees _
Return e.Name
```

可等效為：

```
Employees.Select(e => e.Name)
```

SELECT 運算子可轉譯成 SELECT 查詢運算子方法呼叫及一匿名類型之建構。例如，

```
From e In Employees _
Select e.Name, e.Salary
```

可等效為：

```
Employees.Select(e => New With {e.Name, e.Salary})
```

DISTINCT 運算子轉譯

DISTINCT 運算子可轉譯為 DISTINCT 查詢運算子方法呼叫。例如：

```
From e In Employees _
Distinct
```

可等效為：

```
Employees.Distinct()
```

TAKE 或 SKIP[WHILE]運算子轉譯

TAKE 或 SKIP 運算子可呼叫個別 TAKE 或 SKIP 查詢運算子方法。例如：

```
From e In Employees _
Order By e.Salary Descending _
Return e.Name
Take 5
```

可等效為：

```
Employees.OrderByDescending(e => e.Salary).Select(e => e.Name).Take(5)
```

而查詢：

```
From e In Employees _
Order By e.Salary Descending _
Return e.Name
Skip 5
```

可等效為：

```
Employees.OrderByDescending(e => e.Salary).Select(e => e.Name).Skip(5)
```

當與 WHILE 修飾語一起使用時，可產生一條件運算式以應用於基礎 TAKEWHILE 或 SKIPWHILE 查詢運算子。例如，查詢：

```
From e In Employees _
Order By e.Salary Descending _
Take While e.Salary > 50000 _
Return e.Name
```

可等效於：

```
Employees.OrderByDescending(e => e.Salary). _
    .TakeWhile(e => e.Salary > 50000).Select(e => e.Name)
```

GROUPBY 及 AGGREGATE 運算子轉譯

於 GROUPBY 運算子之後的一聚集 SELECT 或 RETURN 運算子可轉譯成 GROUPBY 查詢運算子方法呼叫，該方法由一經建構群組之聚集跟隨。例如查詢：

```
From C In Customers _
```

```

Group By C.State _
Aggregate Youngest = Min(c.Age), Oldest = Max(c.Age)
Select State, Youngest, Oldest

```

可等效於：

```

Customers
.GroupBy(c => New With { .State = c.State }, _
(g, k) => New With { .State = k, _
.Youngest = g.Min(c => c.Age),
.Oldest = g.Max(c => c.Age) })

```

實例物件

此處所包含的所有實例使用二個別物件集之一。

Company 物件可表示一公司之資訊，且可階層式連結：

```

Class Company
    Dim Name As String
    Dim Employees As List(Of Employees)
    Dim Customers As List(Of Customer)
End Class

```

```

Class Customer
    Dim Name As String
    Dim Orders As List(Of Order)
End Class

```

```

Class Order
    Dim Product As String
    Dim Price As Integer
End Class

```

Class Employee

Dim Name As String

Dim Birthday As Date

Dim Position As String

Dim Salary As Decimal

End Class

Dim Employees As List(Of Employee)

Dim Customers As List(Of Customer)

Book 物件可表示關於例如書店內書本之資訊，且可相關性連結：

Class Book

Dim BookID As Long

Dim Title As String

Dim PublisherID As Long

Dim Price As Double

End Class

Class Author

Dim AuthorID As Long

Dim Name As String

Dim Address As String

Dim City As String

Dim State As String

Dim ZIP As String

End Class

Class BookAuthor

Dim BookID As Long

Dim AuthorID As Long

End Class

Class Publisher

Dim PublisherID As Long

Dim Name As String

Dim Address As String

Dim State As String

End Class

Dim Books As List(Of Book)

Dim Authors As List(Of Author)

Dim BookAuthors As List(Of BookAuthor)

Dim Publishers As List(Of Publisher)

基於前文所述且參照第 4 圖，示出一電腦可實施系統 400，其可即時推定一元素類型及/或依據所推定的元素類型逐步地提供上下文資訊。一般來說，系統 400 可包括使用者介面 102，其可接收一查詢運算式 104，運算式 104 已在上文參照第 1 圖詳細說明過。此外，系統 400 可包括型樣儲存 108，其可操作地耦接(未示出)至使用者介面 102 及/或耦接至一上下文組件 402。上下文組件 402 可檢察查詢運算式 104，並即時作出相關元素類型之推定。例如，如前所述，由於每一查詢子句之查詢運算子符合一特定型樣(例如第 1 圖中之查詢運算子型樣 106)，且此外，由於元素類型可自一查詢子句流至下一查詢子句，因此可利用此二則資訊以決定任何既定下一查詢子句之一元素類型。

當查詢運算子被描述為方法呼叫，且當等查詢運算子

用作一純機械式轉換時，一般需要在過載決議(overload resolution)期間發生類型決議。如此一來，在查詢運算子類型可自一運算子流至另一運算子之條件下，一運算子通常必須完全結合以能夠提供類型至下一待應用運算子。因此，若過載決議由於語法錯誤或某些其它問題而失敗，則無法透過習知機制以提供類型資訊至下一被應用之運算子。不具有此類型流機制的複合查詢運算子(將於後參照第6圖進行詳細說明)導致不特定的錯誤訊息、不同的運算式語義、缺少用於自動完成的上下文資訊以及更慢的編譯器總處理量。

如所說明的，本發明之申請標的可採用查詢運算子之型樣形式化，其能夠使元素類型可不依賴完整方法結合，改為依據正被使用的特定運算子類型樣。進一步舉例說明，考慮由使用者介面 102 接收的現行範例性運算式 104(例如正被分類的查詢運算式 104)：

```
Dim q = From cust In Customers _
    Where cust.Name = "XYZ" & errorUnresolvedVarName _
    Select cust.
```

第一查詢子句包括 FROM 運算子，其可引進一可查詢來源碼類型“Customers”，作為一匿名元素類型“cust”，其可為來源碼類型之一元素。如所說明的，元素類型可流至下一查詢子句，其中 WHERE 運算子可篩選集合，但不改變元素類型。最後，元素類型流至下一查詢子句，其中可發現 SELECT 運算子及 SELECT 運算子方法期望接收到的

函式之一部分，但下一查詢子句其餘部分尚未完整。

可流至 SELECT 子句的元素類型現在可用作該特定查詢子句的一來源碼類型。上下文組件 402 可因此依據該來源碼類型及該查詢運算子，即時推定此查詢子句之元素類型（例如，cust）。因此，該推定可完全為區域性，且不需查詢運算式 104 的完整轉譯。此外，上下文組件 402 可以類似於背景編譯器之方式執行推定。

此外，上下文組件 402 之推定可以多種方式來增進效率。例如，當執行轉譯時只需執行推定一次。因此，若有二個 WHERE 運算子於查詢運算式 104 中，則不必一定要轉譯該第二 WHERE 運算子。此外，可結合使用推定及自訂的較昂貴完整編譯，以短路一般必須執行的類型發現。如此，有多種方式可事先推定元素類型，以加快編譯時間。

此外，一旦上下文組件 402 推定出元素類型（此處為在“Select cust.”查詢子句中的“cust”），上下文組件 402 可逐步地提供上下文資訊 404。例如，當“cust.”中的小點被輸入時，可以一方便且類似於彈出視窗之形式動態地提供上下文資訊 404，以允許如所說明的自動完成，亦可以其它形式實施。需瞭解到由於查詢運算式（按慣例）必須先轉譯然後才可決定元素類型，因此此一特徵習知上不用於查詢運算式。而且，由於供給一完整且無錯誤查詢運算式為習知轉譯/編譯事前所必需，因此任何類似的上下文及/或自動完成機制如今已被排除。

當上下文資訊 404 可用於查詢運算式，且主要以在一

IDE 中提供上下文回授之概念說明上下文資訊 404 時，任何情況皆不必需。例如，此處所述可不只是應用於一查詢運算式之查詢運算子。特別是，類型從一運算子與下一運算子之間流動之概念及一運算子與下一運算子之間之關係，可應用於任一運算子或任一 API，其中類型流可為有用的。存在一實例，其用於將 API 呼叫連結在一起(例如在一類型意圖由一命令流至下一命令的命令列中，包含於其中之管線)。另一沒有類型之實例可在晚期連結，但可將某些形式之流用於如晚期連結(late-bound)最佳化。

應更瞭解到，由上下文組件 402 執行的推定可依據接收到的已知資訊完全預先決定(例如，一關聯一已知及經定義查詢運算子之來源碼類型)，或者依據其它態樣，該推定可為機率的。例如，當以不同運算子類型使用上下文組件 402 時，上下文組件 402 可用於檢察所有可用資料或其一子集合，且可透過經由事件及/或資料取得之一組觀察，來推定系統、環境及/或使用之狀態。可利用推定以識別一特定上下文或動作，或推定可例如產生狀態之機率分佈。推定可為機率的，也就是說有興趣之狀態之機率分佈計算係考慮到資料及事件。推定亦可指用來從一組事件及/或資料建構較高階事件之技術。

此推定可造成由一組觀察到的事件及/或儲存的事件資料來建構新事件或動作，不論該等事件是否在鄰近時間內相關，亦不論該等事件及資料是否來自一或數個事件及資料來源。可利用各種不同的分類(經顯式及/或隱式訓練)

方案及/或系統(例如支援向量機、神經網路、專家系統、貝氏信念網路、模糊邏輯、資料融合引擎…)來執行與本發明所主張申請標的相關的自動及/或推定動作。一分類器可為一函式，其將一輸入屬性向量 $x = (x_1, x_2, x_3, x_4, x_n)$ 映射至一信賴(confidence)，其確定該輸入屬於某一分類，即 $f(x) = confidence(class)$ 。此分類可利用一機率分析及/或應用統計的分析(例如將分析效能及費用納入考量)以預測或推定使用者希望自動執行的動作。一支援向量機(support vector machine, SVM)係可使用之分類器之一實例。SVM 藉由在可能輸入空間中找到超面(hypersurface)進行運作，其中該超面試圖將觸發標準與非觸發事件區隔。直覺地，此使該分類可正確用於測試附近之資料，但無法用於訓練資料。其它引導及非引導模型分類方法包括，例如 naïve Bayes、貝氏網路、決策樹、神經網路、模糊邏輯模型，及機率分類模型提供可利用的不同獨立型樣。如此處所使用的分類亦包括統計回歸，其用於發展優先順序模型。

第 5 圖依據本發明之申請標的說明一方法。同時為了簡單說明之目的。此處所提此方法及其它方法以一系列動作示出及說明，而瞭解並認知本發明所主張申請標的並不限制在所述之步驟順序，而某些動作可以與本文示出及說明之不同順序發生及/或同時與其它動作一起發生。例如，本領域之熟習技藝人士將了解並認知到一方法可替代地表示為一系列有相互關係之狀態或事件，如在一狀態圖中。

此外，並非所有提到的動作皆為實施依據本發明所主張申請標的之方法所必需。另外，應更認知到本發明所揭露之方法可儲存在一製造物上，以能夠將此等方法傳送且轉換至電腦上。此處之用詞製造物意圖包含任何電腦可讀裝置、載體或媒體可存取之電腦程式。

參照第 5 圖，其說明一用於增進元素類型之類型流之電腦可實施方法。一般情況下，在元件符號 502 可接收到包含有一查詢算子的一查詢子句之一部分。查詢運算子可映射到一相關方法呼叫，其可接收參數(例如函式、值、集合…)及可傳回結果類型(例如可查詢類型、數字類型、命令集合…)。可依據一查詢運算子類型定義方法呼叫。

在元件符號 504，可利用一來源碼類型及查詢運算子來決定一元素類型。例如，由於可形式化查詢運算子型樣，因此可利用關聯來源碼類型之查詢運算子之類型，以推定元素類型。在元件符號 506，可利用一先前查詢之元素類型作為元件符號 504 所說明的該利用動作的來源碼類型。依據上述說明，元素類型可按遞歸方式由一查詢子句流至下一查詢子句。

由前述說明，將認知並瞭解到某些特徵及/或態樣可用於以其它方式擴展程式語言查詢能力。如一實例中，可有效利用此處發展之態樣以提供應用電腦及/或應用語言的構想，如具有按慣例與 API 呼叫連結的構成能力的理解。將參照第 6 圖說明這些及其它相關概念。

現在參照第 6 圖，其提供為一電腦可實施系統 600 之

方塊圖，此系統可增進可組合性查詢理解及/或可擴展性查詢運算式。一般來說，系統 600 可包括一轉換組件 602，其可接收初始資料 604。如圖所示，初始資料 604 可為範例性查詢運算式 606 中之第一查詢子句，其中該第一查詢子句一般被限制為一 FROM 子句(例如，第一查詢子句之查詢運算子係一 FROM 運算子)。初始資料 604 可包括一集合(例如 Customers，如圖所示)或一運算式(例如 $\text{Tax} = \text{b.Price} * 0.088$ ，如前文中 FROM 運算子(已參照第 3 圖詳細說明)的相關說明)。在任一情況下，視各種不同應用，初始資料亦可包括一控制變數，其與集合(例如 C 為一控制變數)或運算式(例如 Tax 為一控制變數)有關。

轉換組件 602 亦可接收一具有查詢運算式 606 之特徵的序列中之一查詢子句 608 集。應認知到查詢子句 608 集可為一空集合，在此情況下查詢運算式 606 由一單一查詢子句(例如用作初始資料的 FROM 子句)構成，此將在本文後作更詳細說明。如初始資料 604 可完成的，轉換組件 602 亦可解決(例如 populate 或轉換)集合 608 中每一查詢子句範圍內之控制變數。例如，轉換組件 602 可依查詢運算式 608 要求之次序藉由將每一查詢子句之查詢運算子應用至可用類型，來依序處理集合 608 中每一查詢子句。應認知到依據本發明所主張申請標的，查詢子句 608 集不需要被一普遍有的句法樣板限制，例如由 SQL 實施中所期望的 Xquery 品牌或 Select-From-Where 之版本所定義的 FLWOR/FLWR。此外，集合 608 其本質為可組合性，如此

即使當不確定運算子次序時，個別查詢子句同樣可被有效供給、組合式地拼在一起及產生直覺結果。

依據前述說明，系統 600 亦可包括一理解組件 610，其可管理由查詢運算式 606 所引進的所有控制變數之範圍。一控制變數之範圍可依據一查詢子句之運算子來決定，且可傳送範圍內控制變數至下一查詢子句。例如，轉換組件 602 可建立被視為一管線的東東，如此某些資訊可藉由管線從一查詢子句流至下一查詢子句。因此，轉換組件可接收一關聯一查詢子句之集合、依據關聯查詢子句之運算子轉換(例如篩選、映射…)集合及將經轉換集合輸出至管線，如此其可用於下一查詢子句。

同樣的，範圍中控制變數亦可傳送至下一查詢子句，且理解組件 610 可依據接收到的查詢子句類型明確定義範圍內之控制變數為何。基本上，下一查詢子句只可存取範圍內之變數。除了第 6 圖外，此特徵及其它特徵可進一步參照第 7 圖來說明。

第 7 圖係關於實例查詢運算式 606 之控制變數之範圍的範例性圖示說明，其同樣可由理解組件 610 決定。依據本發明所主張申請標的之一態樣，理解組件 610 可依據查詢子句/運算子之一類型來宣告一新的控制變數(且可將該新的控制變數引入範圍內)。例如，除了已存在範圍內之控制變數外，某些查詢子句類型可引進新的控制變數，即 FROM、LET、SELECT 或 GROUPBY 查詢子句可產生此結果。由 FROM 子句所引進的控制變數一般會累積，直到一

下一 SELECT 或 GROUPBY 子句，在其之後他們可被理解組件 610 引至範圍外。此一情況係由 FROM 子句 702、704 說明(其可映射至初始資料 604 及包含於查詢子句 608 集內的初始查詢子句)，其控制變數保留在範圍內，直到 SELECT 子句 701 之應用為止。

子句 702 引進 Customer 集合及控制變數 C。因此，可藉由管線將資訊提供至下一子句。因此，(C As Customer) 可用於子句 704。子句 704 亦可為一 FROM 子句，這是由於本發明所主張申請標的之組合性質。子句 704 利用控制變數 C 來引進一新的集合 C.Orders(例如所有客戶的訂單)及新的控制變數 O，理解組件 610 可將其引入範圍內。因此，下一查詢子句 706 可存取範圍內之二控制變數 C 及 O。

如前文之具體說明，當某些查詢子句類型可影響範圍內變數時，其它查詢子句類型(例如 WHERE、ORDERBY、DISTINCT 等)則不需要。因此，當查詢子句 706 及 708 可顯響集合時，範圍內變數(例如 C 及 O)可相同，且因此當每一子句由轉換組件 602 接收時，變數可分別由查詢子句 708 及 710 存取。

依據本發明所主張申請標的之一態樣，理解組件 610 可依據一所接收到的查詢子句類型將一現存控制變數引至範圍外。SELECT 查詢子句 710 針對此特徵提供一特性說明。查詢子句 710 暴露於控制變數 C 及 O，但 SELECT 子句之特徵之一可為隱藏範圍內現行控制變數且將新的控制變數變引進範圍內(例如，Name As String 及 Price As Int)。

可提供這些新的控制變數 (Name 及 Price) 給查詢子句 712，然而一般來說無法存取先前控制變數 (例如 C 及 D)，其被理解組件 610 引至關聯 SELECT 子句 710 之範圍外。為了完整性，應注意到一些特徵。第一，查詢子句 710 已說明逗點區隔之組合運算元之另一實例。

需注意的第二特徵為 WHERE 子句 712 跟隨在 SELECT 子句 710 之後。習知查詢語言依據一具有諸如 SELECT、RETURN 之類的映射運算子之查詢子句之應用來終止該查詢。若此類習知查詢語言之使用者在一映射後仍想要續繼查詢操作，則必需實現一可參考前一查詢結果的新的查詢。然而，例如由於本發明申請標的之組合本質，查詢運算式 606 可包括多個 SELECT 子句，及在發生一 SELECT 子句之後查詢運算式 606 仍可繼續 (例如，如子句 712 所說明的)。在相同概念下，當在前節提到查詢運算式 606 時，應指出查詢運算式 606 亦包括二個 FROM 子句 (例如，子句 702 及 704)，其為習知查詢語言無法實施的另一態樣。

當運算式實例未示出查詢運算式 606 時，查詢運算式 606 亦可包括其它查詢子句類型，且特別是，許多的這些其它查詢子句類型可影響控制變數之範圍。例如，於接收到一 RETURN 子句時，理解組件 610 可將所有控制變數引至範圍外。在另一實例中，依據查詢子句之類型，理解組件 610 可宣告新的控制變數並將其納入範圍 (例如引進範圍內)，同時針對隨後之特定查詢子句集之現存控制變數 (例如一先前查詢子句之範圍內控制變數) 納入範圍。

GROUPBY、AGGREGATE 等可增進此類行為，藉此可將現存控制變數納入擴及每一 GROUP 運算子之 AGGREGATE 查詢子句之範圍內。

依據本發明所主張申請標的之另一態樣，理解組件 610 可依據所接收到的查詢子句之類型，產生範圍內所有控制變數之一元組之別名。例如，一 INTO 查詢子句可例如藉由理解組件 610 來促進建立該別名，即使該別名本身不需要為一控制變數。

仍然參照第 6 及 7 圖，可強調各種額外的本發明所主張申請標的。當存在關於有效查詢運算式 606 之可延伸性之諸多優點時(例如，可以幾乎任何數量之幾乎任何類型之查詢子句延伸查詢運算式 606)，另一優點可為查詢運算式 606 亦可在任一點終止。因此，當查詢運算式 606 可能包括不確定數量之查詢子句時，整個查詢運算式 606 亦可僅由一單一查詢子句構成，該單一查詢子句在引進一單一控制變數後終止。因此，一完整且有效查詢運算式 606 的最簡單實例之一，為一僅包括分始資料 604 之查詢運算式 606：

From C In Customers

而查詢運算式 606 不需要包括一顯式 SELECT、RETURN 或其它終端查詢運算子。

舉例說明，轉換組件 602 可經組態以將一隱式 SELECT 或 RETURN 添加至每一接收到的查詢子句。因此，只要一包含於管線內之生成集合為一期望狀態，則不需要針對一

顯式敘述選擇或傳回這些中繼結果。進一步說明，若查詢運算式 606 在給定子句(例如 702-708)之任一子句之後結束，則可輸出管線中之集合作為結果，且可依據範圍內控制變數之數量推定該輸出之格式。

例如，若查詢運算式 606 在子句 704-708(其中每一情況下範圍內有二個控制變數 C 及 O)中任一子句之後終止，則可推定期望的輸出應以名稱值對集合的方式存在。因此，輸出可為一具有每一控制變數的一欄位的別名。相反的，在僅有範圍內控制變數之情況下，如在子句 702 之後，查詢運算式 606 之一實現不期望取回一具有一欄位 c 的別名。進一步說明，實現者可能想要客戶集合。所以在範圍內僅有一控制變數之情況下，輸出可就是集合之基礎值。

依據本發明之申請標的，由一 FROM 子句引進之控制變數可連結一運算式及一集合。在一代表性實例中，考慮以下上述引進之查詢片段：

```
From b in Books _
From Tax = b.Price * 0.088 _
```

第一 FROM 子句為集合 Books 引進一控制變數 b，其被引入範圍內且可由下一查詢子句存取。下一 FROM 子句引進關聯一運算式(而非一集合) b.Price * 0.088 之控制變數 Tax。對於二個 FROM 子句之後者，其影響可為提供一可存取名稱至運算式之結果。藉由控制變數此名稱可稱作 tax，其可保持在範圍中直到例如一隨後 SELECT 子句為

止。此特徵可視為類似查詢內之一程序敘述，其允許計算一次一運算式之值，即使該等值隨後可能被多次使用到，亦不需要重新計算該等值。

依據本發明之申請標的之另一態樣，操作集可利用在一單一查詢運算式 606 內。當習知查詢語言一般供提供一單一集合作為一輸入時，轉換組件 602 可接收二個集合作為輸入。例如，轉換組件 602 可經組態以接收多個集合，且可輸出一具有關聯該等多個集合之控制變數之單一集合，其中該等集合係依據查詢子句(例如一 UNION 子句、一 INTERSECT 子句…)之一類型合併。經合併控制變數可被引入範圍內，透過關聯操作集被傳送，及成為可由下一查詢子句存取，因此可繼續查詢運算式 606。雖然控制變數一般必須為相同類型且一般必須具有相同名稱，但是當然他們可連結二個不同的集合。

現參照第 8 圖，說明一電腦可實施方法 800，用於以一組合式方式增進建立查詢理解。由元件符號 802 開始，可取得一集合(或運算式)及一連結該集合之控制變數(或運算式)。一般情況下，此獲取的資料為一查詢運算式之第一查詢子句之產物，其通常為一 FROM 子句。在元件符號 804，可接收到來自包含於一查詢運算式之一查詢子句集的一現行查詢子句。需知不像習知語言一樣需要整體理解，因此不需設定語法次序限制於現行查詢子句，或大多數情況下，不需設定此限制於查詢運算式整體。

在元件符號 806，可依據現行查詢子句修改集合。簡

單地說，接收作為輸入的集合可依據連結現行查詢子句的查詢運算子之規範進行轉換。需知可依據連結現行查詢子句的一期望類型而非依據整體語法樣板，來評估現行查詢子句。接著，在元件符號 808，可將經修改的集合傳送至下一查詢子句。因此，可利用現行查詢子句之輸出作為下一查詢子句輸入。

在元件符號 810，可依據現行查詢子句決定控制變數之範圍。例如，某些類型之查詢子句不促進現存控制變數之任何改變，而其它查詢子句可幫助引進新的控制變數(在僅針對一特定系列之下一查詢子句的某些情況下)，另外其它查詢子句當(選擇性)引進新的控制變數時，可幫助將現存控制變數引至範圍外。在元件符號 812，可允許下一查詢子句存取現行查詢子句範圍內之控制變數。

現參照第 9 圖，其圖示說明一範例性電腦系統方塊圖，其可操作以執行本發明所揭示結構。為了提供本發明各種態樣之額外背景，第 9 圖及下文之說明希望能提供一簡短、一般性之合適運算環境 900 之說明，在該環境中可實現本發明之各種態樣。此外，當前文已說明本發明之電腦可實施指令(其可執行於一或多電腦上)之一般架構時，本領域之熟習技藝人士將認知到此發明亦可利用其它程式模組及/或軟硬體結合來實現。

一般來說，程式模組包括例行程序、應用程式、組件、資料結構等可執行特定任務或實施特定抽象資料類型。此外，本領域之熟習技藝人士將認知到，本發明所揭露方法

可以其它電腦系統組態實施，其包括單一處理器或多處理器電腦系統、微型電腦、主機電腦以及個人電腦、手持電腦裝置、應用微處理器或可程式化消費性電子產品等，上述每一裝置或系統可操作地耦接至一或多相關裝置。

圖示說明的本發明態樣亦可實施於分散式運算環境中，其中某些特定任務係藉由遠端處理裝置執行，該等遠端處理裝置透過一通訊網路連結。在一分散式運算環境中，程式模組可位於區域及遠端記憶體儲存裝置中。

電腦基本上包括多種電腦可讀取媒體。電腦可讀取媒體可為任何可由電腦存取的媒體，其同時包括揮發性與非揮發性媒體、可移除與不可移除媒體。舉例來說(而非限制)，電腦可讀取媒體可包含電腦儲存媒體與通信媒體。電腦儲存媒體同時包括揮發性與非揮發性、可移除與不可移除媒體，其可以任何方法或技術來實施以儲存資訊，例如電腦可讀取指令、資料結構、程式模組或其它資料。電腦儲存媒體包括(但不限於)RAM、ROM、EEPROM、快閃記憶體或其它記憶體技術，CD-ROM、數位多功能碟片(DVD, "Digital versatile disk")或其它光碟儲存元、磁匣、磁帶、磁碟儲存元或其它磁性儲存裝置，或任何其它可用於儲存所想要資訊並可由電腦存取之媒體。

通信媒體基本上包含了電腦可讀取指令、資料結構、程式模組或其它在一調變資料信號中的資料，例如載波或其它輸送機制，並包括任何資訊傳遞媒體。該術語「調變資料信號」代表一信號中其一或多項特性為利用方法設定

或改變以在該信號中編碼資訊。舉例來說(而非限制),通信媒體包括有線媒體,像是有線網路或直接線路連線,以及無線媒體,像是聲波、RF、紅外線及其它無線媒體。任何上述的組合亦必須包含在電腦可讀取媒體的範疇內。

參照第9圖,用於實施本發明各種態樣之範例性環境900,其包括一電腦902,電腦902包括一處理單元904、一系統記憶體906及一系統匯流排908。系統匯流排908將多種系統組件(包括系統記憶體,但不限於此)耦合至處理單元904。處理單元904可為各種市售處理器中任一種。雙微處理器及其它多處理器結構亦可用作處理單元904。

系統匯流排908可為數種匯流排結構之任何一種,其可進一步內部連接至一記憶體匯流排(具有或沒有一記憶體控制器)、一周邊匯流排、及一使用多種市售匯流排架構中任一種的本地匯流排。系統記憶體906包括唯讀記憶體(read-only memory, ROM)910及隨機存取記憶體(random access memory, RAM)912。一基本輸入/輸出系統(Basic input / output system, BIOS)儲存於諸如ROM、EPROM、EEPROM之類的一非揮發記憶體中,其包含有基本的例式來協助電腦內之元素間的資訊轉換,例如在開機期間。RAM 912亦可包含一高速RAM,如用於儲存資料的靜態RAM。

電腦902更包括一內部硬碟機(hard disk drive, HDD)914(例如EIDE、SATA),內部硬碟機914亦可經組態外部用於一適當的機殼(未示出)、一磁式軟碟驅動機

(floppy disk drive, FDD)916(例如讀取一可移除磁片 918 或寫入至其中)及一光碟驅動機 920(例如讀取一 CD-ROM 光碟 922, 或讀取諸如 DVD 之類的高容量光學媒體或寫入至其中)。硬碟機 914、磁碟驅動機 916 及光碟驅動機 920 可分別經由硬碟機介面 924、磁碟驅動機介面 926 及光碟驅動機介面 928 連結至系統匯流排 908。用於實現外部驅動機的介面 924 至少包括以下一者或同時包括兩者：通用序列匯流排(Universal Serial Bus, USB)及 IEEE1394 介面技術。其它外部驅動機連結技術亦在本發明之考慮範圍內。

驅動機及其相關電腦可讀取媒體提供非揮發資料儲存、資料結構、電腦可執行指令等。對於電腦 902, 驅動機及媒體包含以適當數位格式儲存任何資料之儲存。雖然上述描述的電腦可讀取媒體係指 HDD、可移除磁片及諸如 CD 或 DVD 之類的可移除光學媒體, 然而本領域之熟習技藝人士應認知到可由電腦讀取之其它類型媒體(如 zip 磁碟機、磁帶、快閃記憶體卡、卡帶等)亦可用於範例性操作環境; 此外, 任何此種媒體可包含電腦可執行指令, 其用於執行本發明之方法。

多個程式模組可儲存於驅動機及 RAM912 中, 其包括一作業系統 930、一或多應用程式 932、其它程式模組 934 及程式資料 936。作業系統、應用程式、模組及/或資料, 上述全部或部分可要儲存於 RAM912 中。需知本發明可以各種市面上可取得作業系統或其結合來實施。

一使用者可經由一或多輸入裝置輸入指令及資訊到電

腦 902，輸入裝置如一鍵盤 938 及一指向裝置，像是滑鼠 940。其它輸入裝置(未示出)可包括麥克風、IR 遠端控制、搖桿、遊戲板、電子筆、觸控螢幕等等。這些及其它輸入裝置通常透過耦合至系統匯流排 908 之一輸入裝置介面 942 連接到處理單元 904，但其亦可由其它介面做連接，像是平行埠、IEEE1394 序列埠、遊戲埠、通用序列匯流排(USB) 埠及一 IR 介面等。

一監視器 944 或其它種類的顯示裝置亦可經由一介面連接到系統匯流排 908，例如視訊配接器 946。除了監視器 944 外，電腦一般可包括其它周邊輸出裝置(未示出)，像是喇叭、印表機等。

電腦 902 可使用邏輯連線經由有線及/或無線通訊連結到一或多部遠端電腦(例如遠端電腦 948)，而在一網路化環境中操作。遠端電腦 948 可為工作站、伺服器電腦、路由器、個人電腦、可攜式電腦、應用微處理器娛樂裝置、一對等裝置或其它常見網路節點，且基本上包括許多或所有關於上述電腦 902 之元素，但是為了簡要說明目的，僅示出一記憶體/儲存裝置 950。示出的邏輯連線包括有線/無線連結，其連結至一區域網路(Local area network, LAN)952 及/或一大型網路，例如一廣域網路(Wide area network, WAN)954。這些 LAN 及 WAN 網路化環境常見於辦公室、公司，且可增進企業化電腦網路，如企業內網路，上述所有網路可連結至一全球通訊網路，例如網際網路。

當用於 LAN 網路環境中時，電腦 902 經由一有線及/

或無線通訊網路介面或配接器 956 連接到網路 952。配接器 956 可增進有線或無線通訊至 LAN952，其亦可包括一無線存取點，其配置用於與無線配接器 956 通訊。

當用於 WAN 網路環境中時，電腦 902 可包括一數據機 958，或其連結至 WAN954 上一通訊伺服器，或者其具有其它構件用於在 WAN954 上建立通信，如以網際網路的方式連結。數據機 958 可為內接或外接式及有線或無線裝置，其可透過序列埠介面 942 連接到系統匯流排 908。在一網路化環境中，相關電腦 902 或其一部份所述之程式模組可儲存在遠端記憶體/儲存裝置 950 中。應可瞭解到所示之網路連結僅為示例性，其可使用其它構件來建立電腦之間的通信鏈結。

電腦 902 可操作以與任何無線裝置或實體通信，該等裝置或實體操作地配置在無線通訊中，例如印表機、掃瞄器、桌面及/或可攜式電腦、可攜式資料助理、通信衛星、任何關聯一無線可偵測標籤之設備或位置(例如涼亭、報攤、廁所)，及電話。此包括至少 Wi-Fi 及 Bluetooth™ 無線技術。因此，通訊可為一預先定義結構，其定義為具有一習知網路或特別地只具有至少兩個裝置間通訊之結構。

Wi-Fi(Wireless Fidelity, 無線相容認證)允許自家中沙發、旅館床上或工作會議室無線連結至網際網路。Wi-Fi 係一類似使用於手機之無線技術，其使諸如電腦之類的裝置在一基地台範圍內能夠於室內外傳送及接收資料。Wi-Fi 網路使用稱為 IEEE802.11(a,b,g 等)之無線電技術以提供

安全、可靠、快速的無線連結。Wi-Fi 網路可用以將電腦互相連結、連結至網際網路及連結至有線網路(使用 IEEE802.3 或乙太網路)。Wi-Fi 網路操作在未經許可的 2.4 及 5 GHz 無線電頻帶，例如以 11Mbps(802.11a) 或 54Mbps(802.11b) 資料速率傳輸，或使用包含二頻帶(雙頻帶)之產品，因此網路可提供類似基礎用於許多辦公室之 9BaseT 有線乙太網路的工作效能。

現參照第 10 圖，其示出一範例性電腦編譯系統方塊圖，可操作該系統以執行本發明揭示結構。系統 1000 包括一或多客戶 1002。客戶 1002 可為硬體及/或軟體(例如執行緒、應用程序、運算裝置)。例如，客戶 1002 可藉由利用本發明來包含 cookie 及/或相關上下文資訊。

系統 1000 亦可包括一或多伺服器 1004。伺服器 1004 亦可為硬體及/或軟體(例如執行緒、應用程序、運算裝置)。例如，伺服器 1004 可藉由利用本發明來包含執行緒以執行轉換。在一客戶 1002 與一伺服器 1004 間之一可能通信可為一資料包之形式配置在二或更多電腦程序之間傳送。例如，資料包可包括一 cookie 及/或相關上下文資訊。系統 1000 包括一通信結構 1006(例如一全球通信網路，如網際網路)，其可用於增進客戶 1002 與伺服器 1004 間之通信。

可經由一有線(包括光纖)及/或無線技術增進通信。客戶 1002 可操作地連結至一或多客戶資料儲存 1008，其可用於儲存客戶 1002 本地之資訊(例如 cookie 及/或相關上下

文資訊)。同樣的，伺服器 1004 可操作地連結至一或多伺服器資料儲存 1010，其可用於儲存伺服器 1004 本地之資訊。

前文已說明了各種具體實施例。當然，以實施例方式說明，不可能說明所有可能的組件或方法組合，但是本領域之熟習技藝人士之一可認知到許多其它的組合及替代方案亦為可行。因此，本段實施方式之說明意圖包含本發明所主張申請標的之精神及範圍內之所有此類變更、修改及變化。

特別是且關於上述組件、裝置、電路、系統等所執行的各種功能，用於描述此類組件的用語(包括提及「構件」(means))，其意圖(除非另有指示外)符合每一組件所執行的特定功能(例如功能等效用語)，即使結構上不等效所揭露結構，其執行本文揭示的實施例範例性態樣之功能。在這點上，將認知到具體實施例包括一系統及一具有可執行指令之電腦可讀取媒體，用於執行各種方法之步驟及/或事件。

此外，當一特定特徵已僅以數種實施方式之一來揭露時，此特徵可結合其它實施方式之一或多其它特徵，作為任何給定或特定應用之期望的及有利的特徵。此外，該用詞「包含」及其變化用於實施方式及申請專利範圍中，這些用詞意圖包括「至少包含」之意。

【圖式簡單說明】

第 1 圖為一電腦實現系統方塊圖，該系統可利用一查詢型樣以增進元素類型之類型流。

第 2A 圖說明一範例性查詢運算式的一非限制代表性第一查詢子句及各種非限制代表性下一查詢子句。

第 2B 圖更詳細說明該範例性查詢運算式的查詢子句及一元素類型之類型流。

第 3 圖示出具有範例性輸入及輸出的查詢運算子型樣之各種非限制實施例。

第 4 圖為一電腦實現系統方塊圖，該系統可即時推定一元素類型及/或逐步提供基於該推定元素類型之上下文資訊。

第 5 圖示出程序之一範例性流程圖，其定義一電腦實現方法，用以增進一查詢運算式內元素類型之類型流。

第 6 圖為一電腦實現系統方塊圖，該系統可增進可組合性查詢理解及/或可延伸查詢運算式。

第 7 圖說明與控制變數範籌相關之範例性方塊圖，該等控制變數起因於一範例性查詢運算式之範例性查詢子句。

第 8 圖示出程序之一範例性流程圖，其定義一電腦實現方法，用以增進一組合性方式中查詢理解之建構。

第 9 圖示出一電腦方塊圖，該電腦可操作執行所揭示架構。

第 10 圖說明一範例性計算環境概要方塊圖。

【主要元件符號說明】

- | | | | |
|--------------------|----------------|-------------|-------------|
| 100 | 電腦可實施系統 | 102 | 使用者介面 |
| 104 | 查詢運算式 | 106 | 查詢運算子型樣 |
| 108 | 型樣儲存 | 202 | 第一查詢子句 |
| 204 _{1,2} | 下一查詢型樣 | 206 | FROM 運算子 |
| 208 | 來源碼類型 Customer | 210 | 控制變數 C |
| 214 | WHERE 運算子 | 216 | SELECT 運算子 |
| 302-328 | 查詢運算子型樣 | 400 | 電腦可實施系統 |
| 402 | 上下文組件 | 404 | 上下文資訊 |
| 600 | 電腦可實施系統 | 602 | 轉換組件 |
| 604 | 初始資料 | 606 | 查詢運算式 |
| 608 | 查詢子句集 | 610 | 理解組件 |
| 702,704 | FROM 子句 | 706,708,712 | 查詢子句 |
| 710 | SELECT 子句 | 900 | 運算環境 |
| 902 | 電腦 | 904 | 處理單元 |
| 906 | 系統記憶體 | 908 | 匯流排 |
| 910 | 隨機存取記憶體 (RAM) | 912 | 唯讀記憶體 (ROM) |
| 914 | 內(外)部硬碟機 (HDD) | 916 | 軟碟機 (FDD) |
| 918 | 磁碟片 | 920 | 光碟機 |
| 922 | 光碟片 | 924-928 | 介面 |
| 930 | 作業系統 | 932 | 應用程式 |
| 934 | 模組 | 936 | 資料 |
| 938 | 鍵盤 | 940 | 滑鼠 |
| 942 | 輸入裝置介面 | 944 | 監視器 |

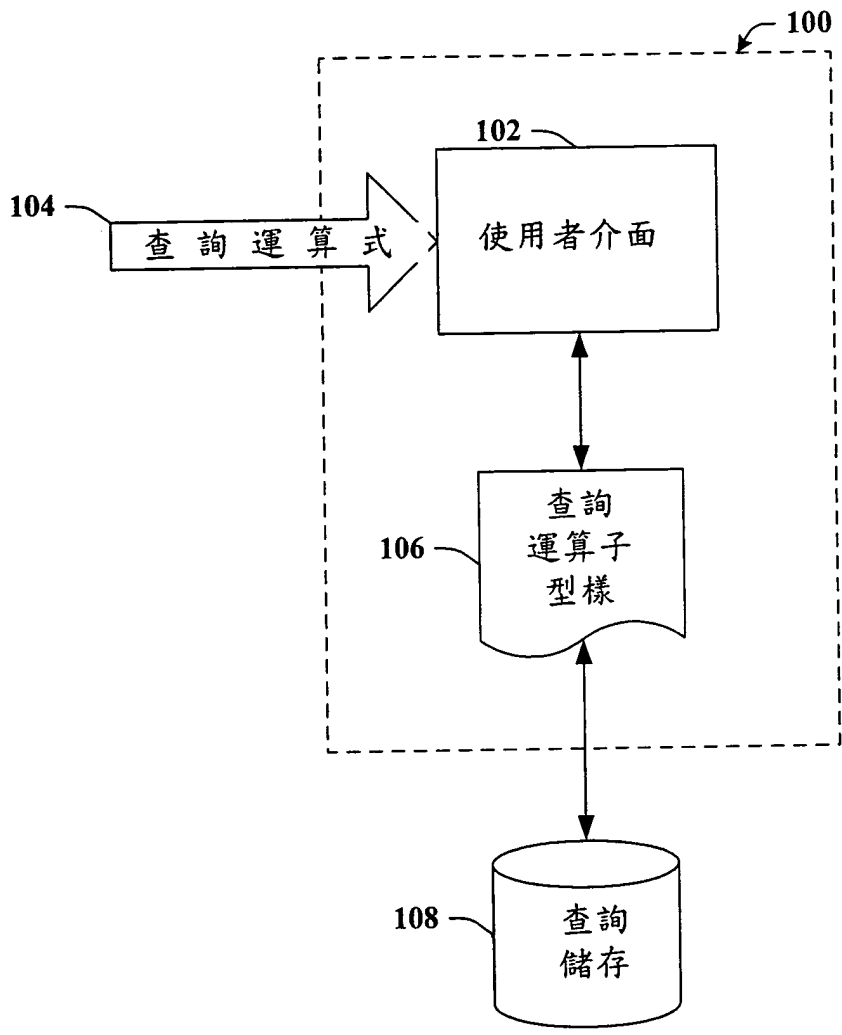
946	視訊配接器	948	遠端電腦
950	記憶體/儲存裝置	952	區域網路(LAN)
954	廣域網路(WAN)	956	網路配接器
958	數據機	1000	系統
1002	客戶	1004	伺服器
1006	通信結構	1008	客戶資料儲存
1010	伺服器資料儲存		

五、中文發明摘要：

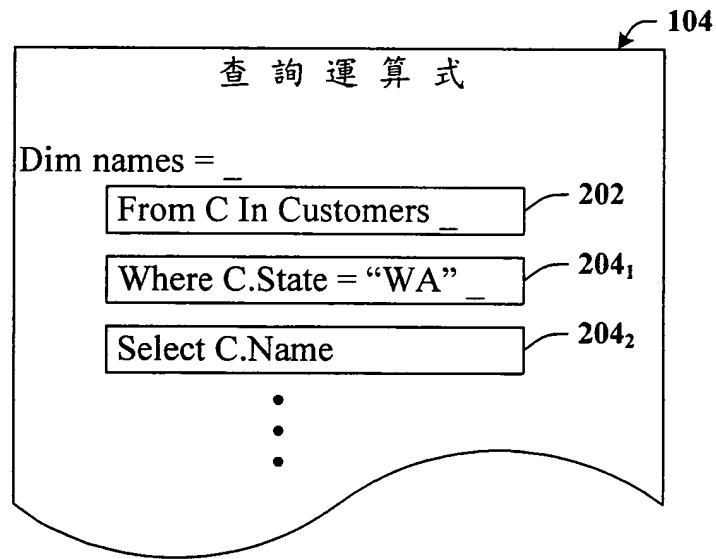
本發明申請標的係有關將運算子型樣形式化以增進一運算式第一子句中之一運算子與該運算式下一子句中之一運算子兩者間的元素類型之類型流。可結合流至下一子句的類型及原本存在該下一子句的運算子，以推定該下一子句的元素類型。如此一來，可逐步達成類型查核、自動完成及其它優勢特點，而不需事先完整轉譯該運算式。

六、英文發明摘要：

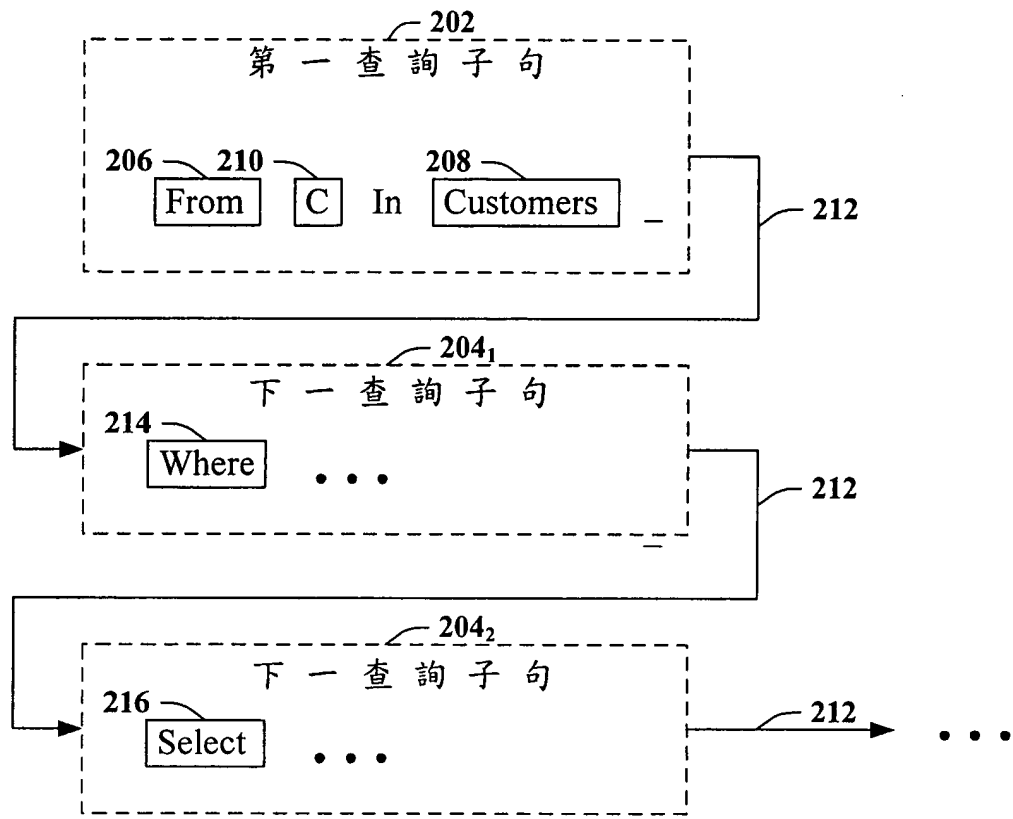
The claimed subject matter relates to a formalization of operator patterns that can facilitate type flow of element types between an operator in a first clause of an expression to an operator in a next clause of the expression. The type that flows to, and the operator extant in, the next clause can be combined to infer the element type of the next clause. As such, type-checking, autocompletion and other advantageous features can be accomplished incrementally without the need for a prior full translation of the expression.



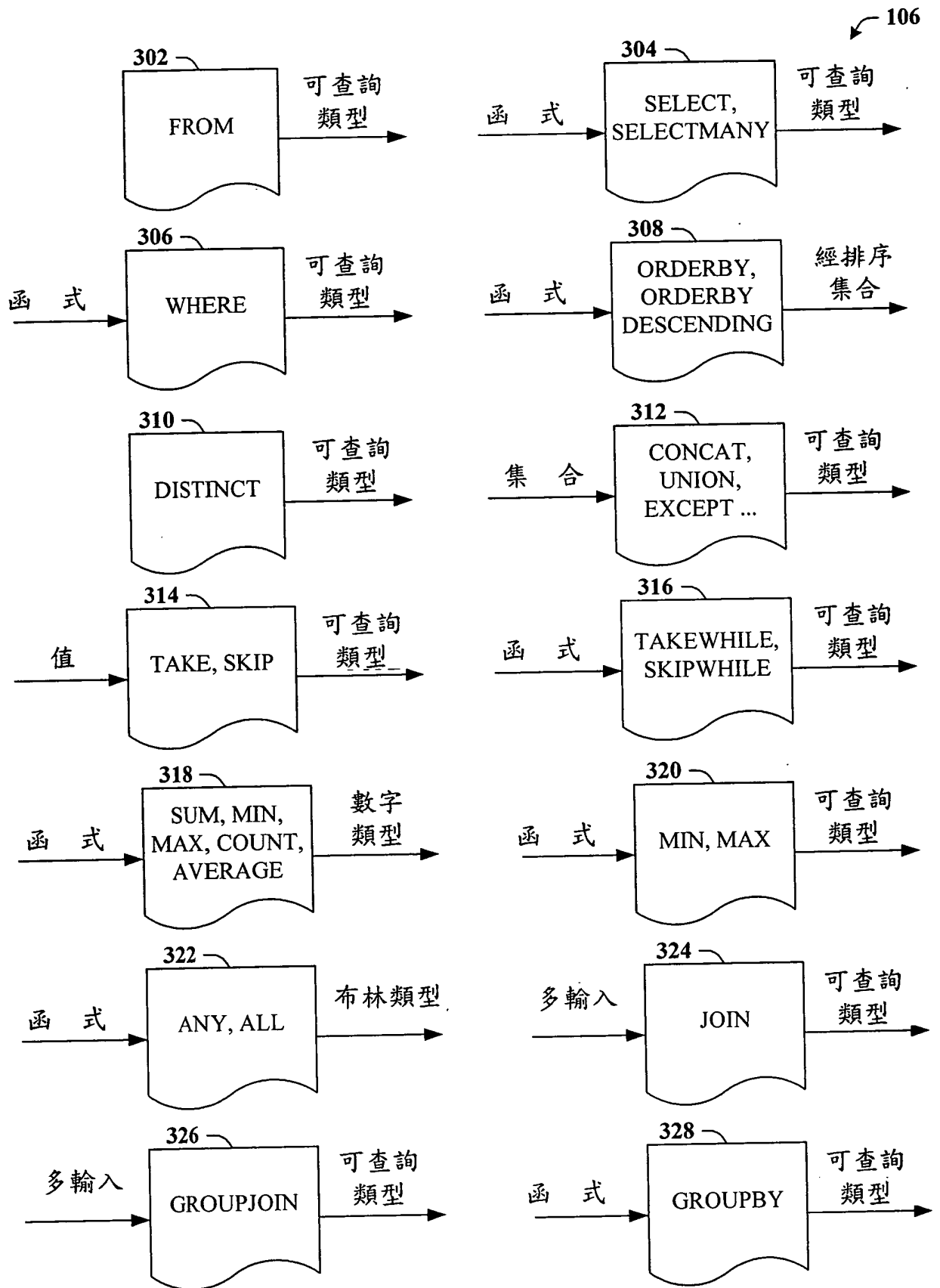
第1圖



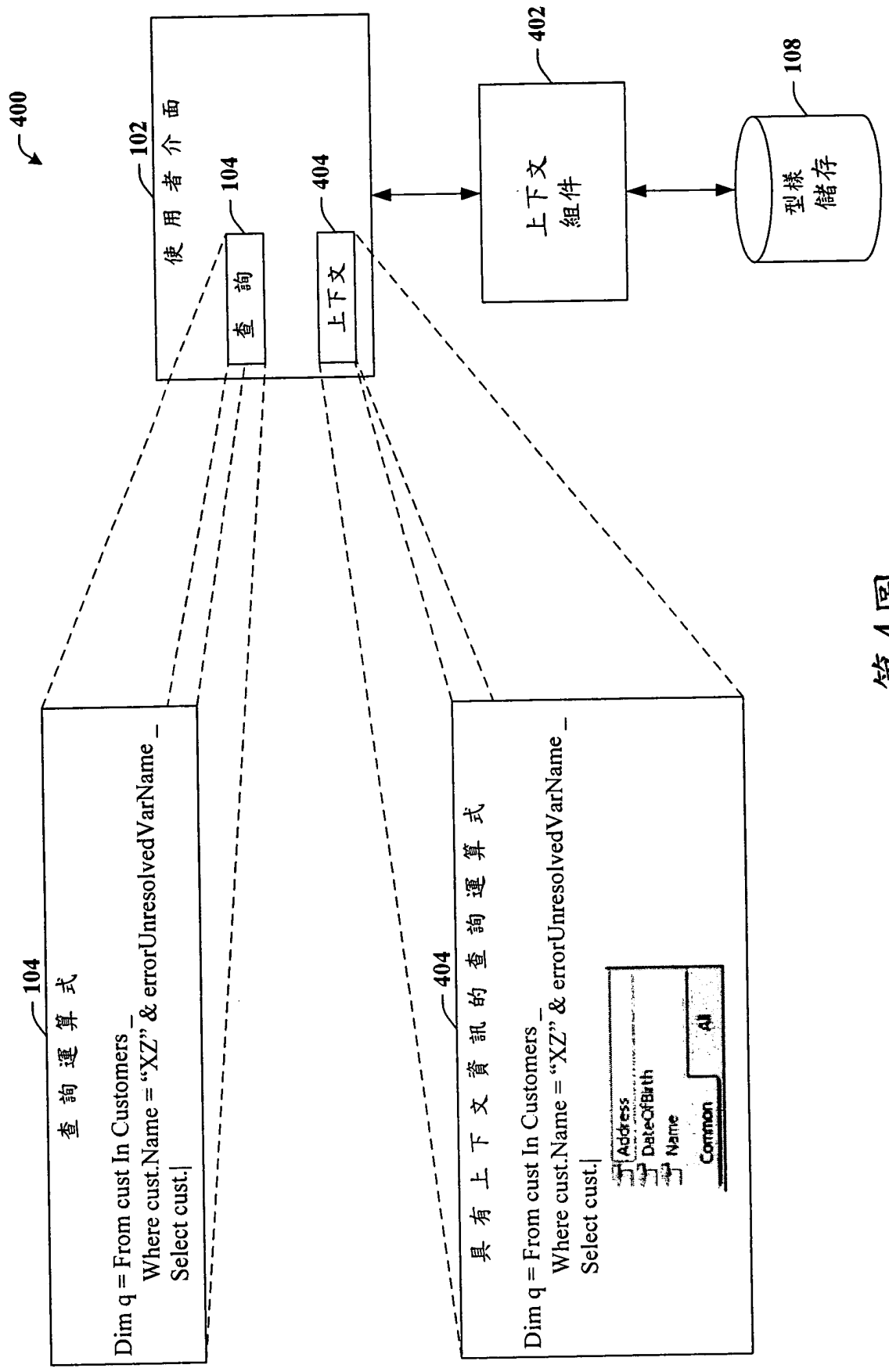
第2A圖



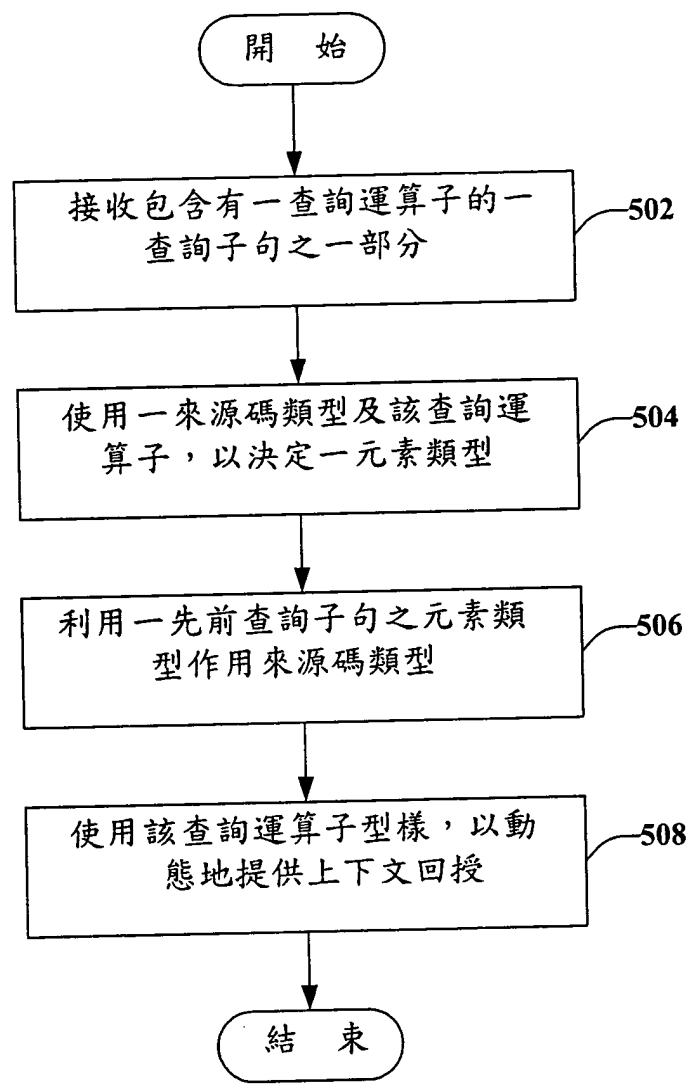
第2B圖



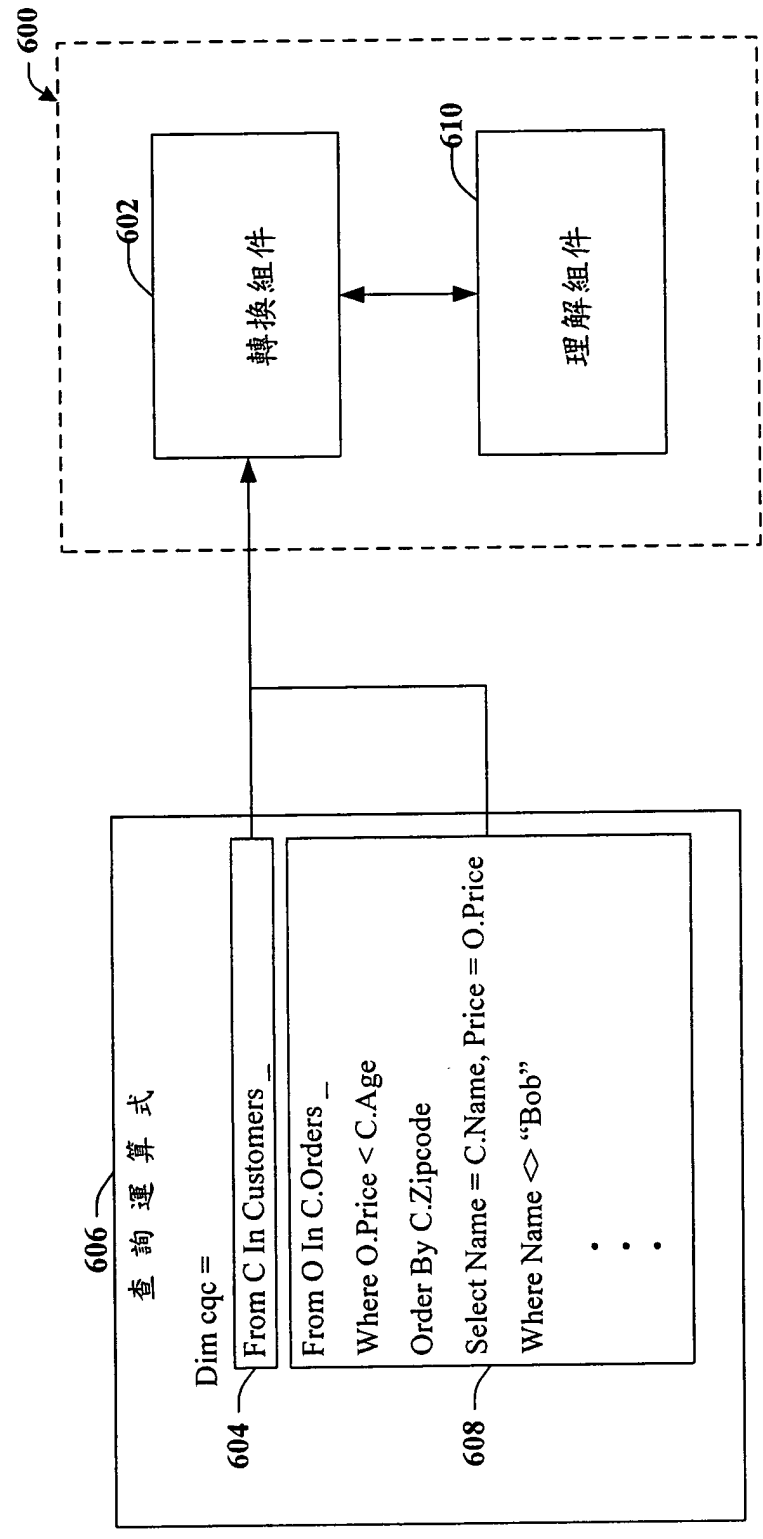
第3圖



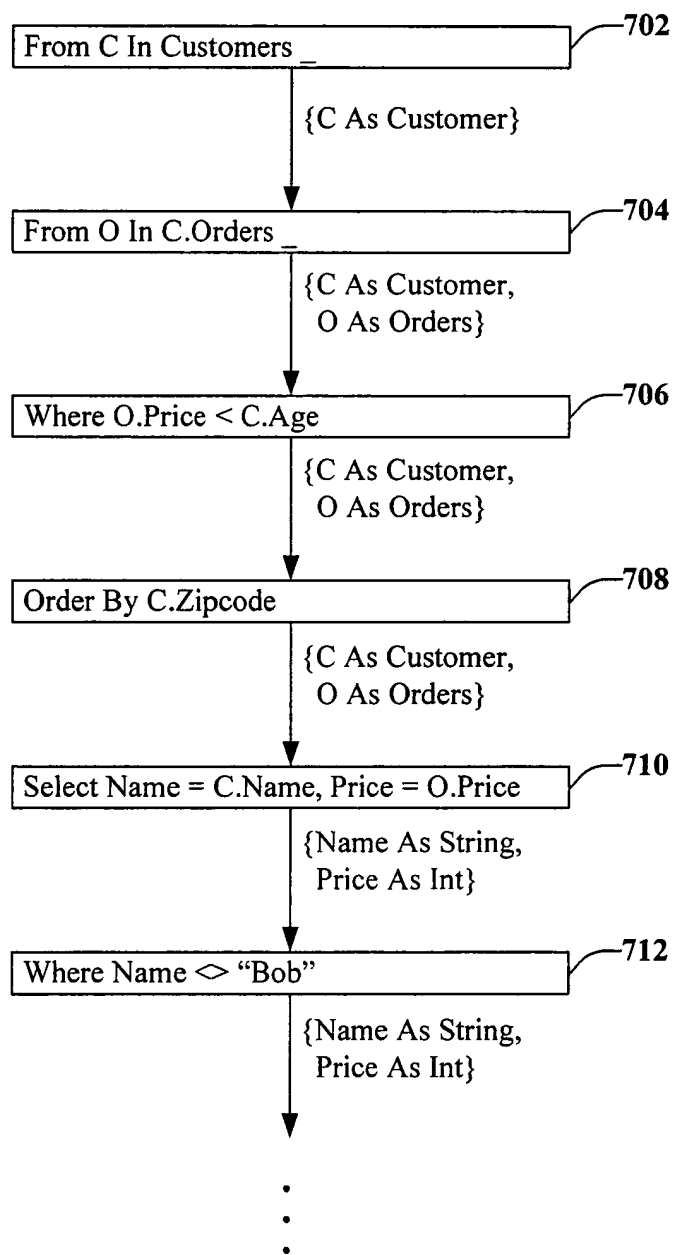
第4圖



第5圖

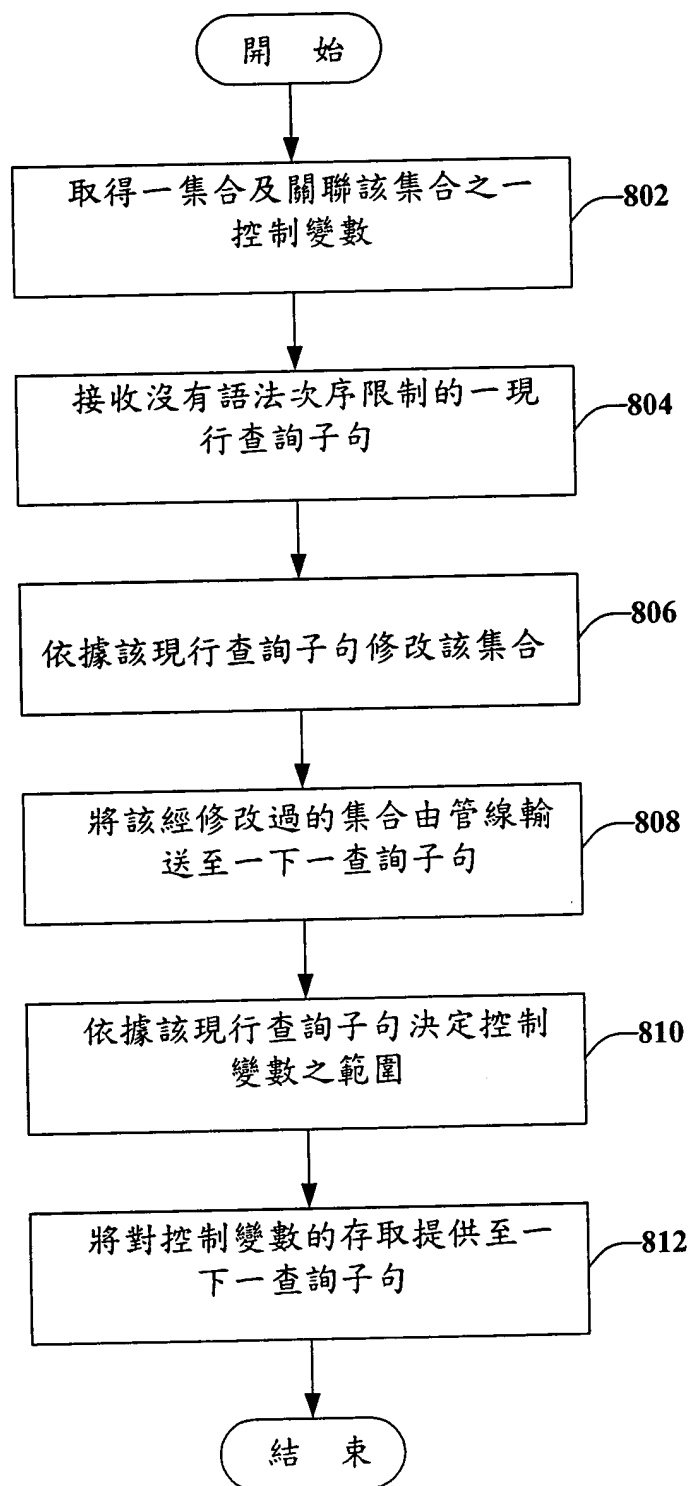


第6圖

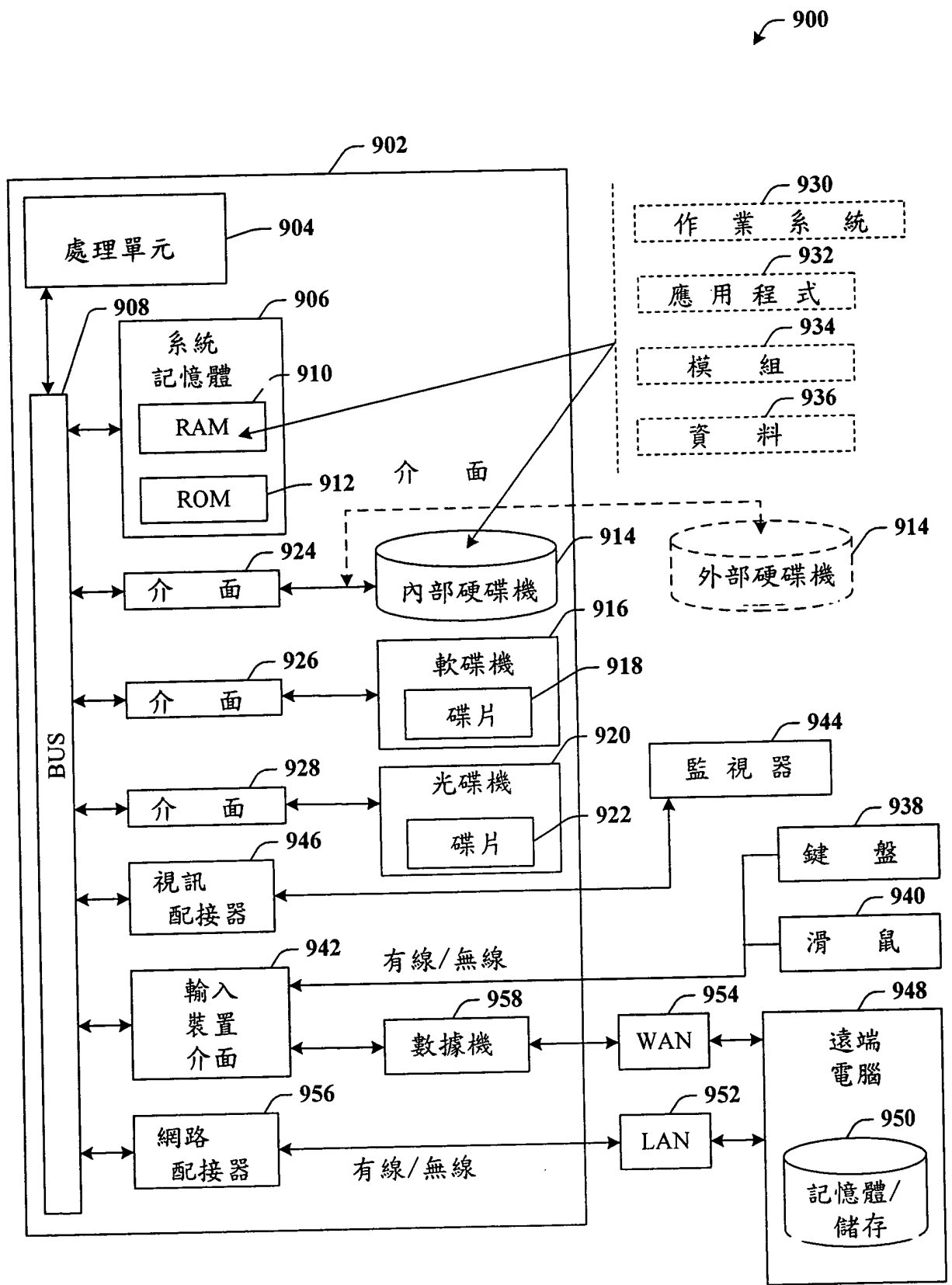


第7圖

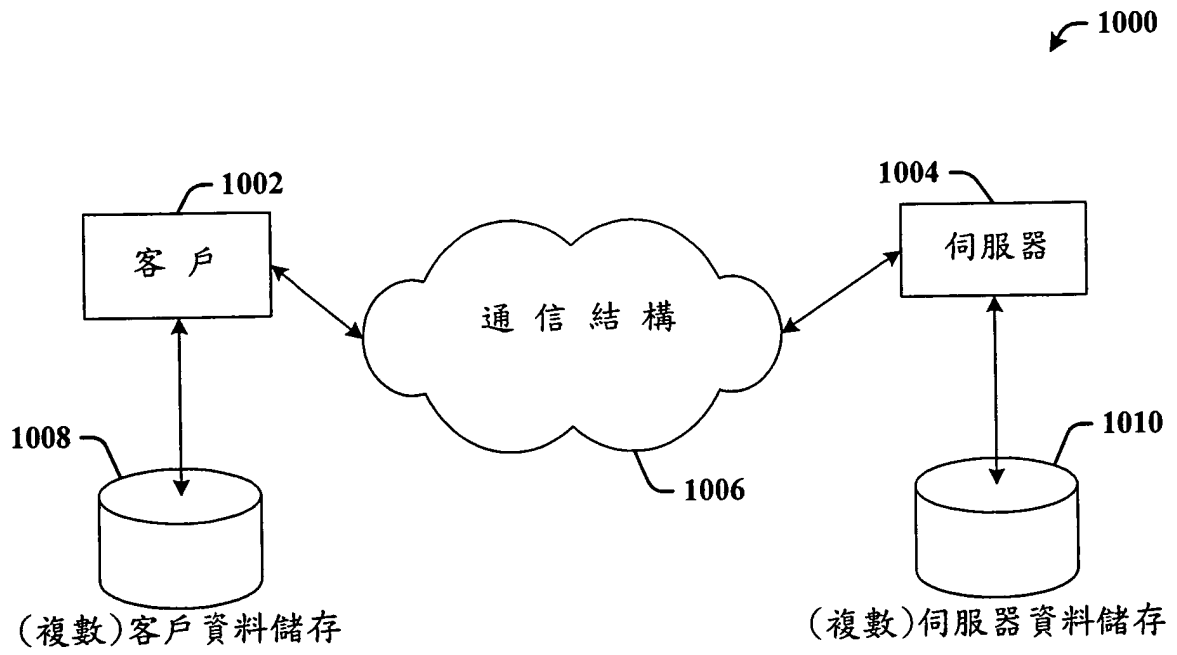
800



第8圖



第9圖



第10圖

七、指定代表圖：

(一)、本案指定代表圖為：第(1)圖。

(二)、本代表圖之元素代表符號簡單說明：

100	電腦可實施系統	102	使用者介面
104	查詢運算式	106	查詢運算子型樣
108	型樣儲存		

八、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

無

發明專利說明書**公告本**

(本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫)

※ 申請案號：97103306

※ 申請日期：2008 年 1 月 29 日

※ IPC 分類：G06F 17/30 2006.01

G06F 7/64 2006.01

一、發明名稱：(中文/英文)

用於處理查詢運算式的方法、電腦程式產品及電腦系統

METHOD, COMPUTER PROGRAM PRODUCT, AND COMPUTER
SYSTEM FOR PROCESSING A QUERY EXPRESSION**二、申請人：(共 1 人)**

姓名或名稱：(中文/英文)

美商·微軟公司

Microsoft Corporation

代表人：(中文/英文)

艾華那諾爾 D 巴特萊

EPPENAUER, D. BARTLEY

住居所或營業所地址：(中文/英文)

美國華盛頓州列德蒙微軟路 1 號

One Microsoft Way, Building 8, Redmond, WA 98052-6399, U.S.A.

國 籍：(中文/英文)

美國/USA

三、發明人：(共 5 人)

姓 名：(中文/英文)

1. 梅傑漢瑞奇斯約翰尼斯瑪莉亞/MEIJER, HENRICUS JOHANNES MARIA
2. 席爾維亞曼達 K/SILVER, AMANDA K.
3. 維克保羅 A 二世/VICK JR., PAUL A.
4. 札保克里茲奇/ZABOKRITSKI, EUGUENI
5. 辛古茲艾雷克塞 V/TSINGAUZ, ALEKSEY V.

100. 7. 27 年 月 日修正本

第97103306號專利案100年7月修正

十、申請專利範圍：

1. 一種在包括一處理器及系統記憶體的一電腦系統處用於處理一查詢運算式的電腦可實施方法，該方法包含以下步驟：

(A) 存取一查詢運算式 (query expression) 的一動作，該查詢運算式包括多個鏈結的查詢子句，該等多個鏈結的查詢子句包括至少一第一查詢子句鏈結至一下一查詢子句，該等多個鏈結的查詢子句的次序係界定於該查詢理解中，使得該次序係獨立於該查詢理解的基礎計算語言，其中：

(B) 該第一查詢子句包括一可查詢來源碼類型、一第一查詢運算子、及該來源碼類型的一元素類型，該元素類型具有與其相關聯的一控制變數，該控制變數具有藉由該查詢運算子所決定的一相關聯的範圍，該查詢來源碼類型指示待查詢的來源碼資料的類型，該元素類型指示該第一查詢子句的該等結果的該元素類型，該查詢運算子映射至根據一查詢運算子型樣所界定的一方法；及

(C) 該下一查詢子句包括待應用至該第一查詢子句的該等結果的一下一元素類型及一下一查詢運算子，該下一元素類型指示該下一查詢子句的該等結果的該元素類型，該下一查詢運算子映射至根據一下一查詢運算子型樣所界定的一下一方法；

(D) 評估該第一查詢子句，以從基於該查詢來源碼

類型的該來源碼資料中獲得該元素類型的一組元素，

(E) 評估該第一查詢子句的步驟包括以下步驟：呼叫該方法以在該來源碼資料上實施該查詢運算子；

(F) 推定出該下一查詢子句的一下一來源碼類型係為該元素類型，其中該推定係基於該查詢運算子所考慮的該來源碼類型，其中該推定無須轉譯該整體查詢理解而實行；其中將該下一來源碼類型推定為該元素類型的該步驟係為即時的；

(G) 傳送該組元素至該下一查詢子句，其中當該下一查詢子句支援該控制變數的該範圍時，該相關聯的控制變數被傳送；

(H) 基於該推定出的元素類型逐步地提供上下文資訊至該下一查詢子句；及

(I) 評估該下一查詢子句，以從基於該下一查詢來源碼類型及該上下文資訊的該組元素中獲得該下一元素類型的一下一組元素，評估該下一查詢子句的步驟包括以下步驟：呼叫該下一方法以在該組元素上實施該下一查詢運算子。

2. 如申請專利範圍第 1 項所述之方法，其中該第一查詢運算子係為一笛卡兒乘積 (Cartesian product) 形式，且該元素類型係為一匿名類型，其為關於該來源碼類型之範圍內所有控制變數之一聚集。

3. 如申請專利範圍第 1 項所述之方法，其中該查詢運算子型樣包括一投影或映射查詢運算子，其經組態成接收包括該來源碼資料之一參數的一函式 (function) 作為一參數 (argument)，且傳回該元素類型之一參數。
4. 如申請專利範圍第 1 項所述之方法，其中該第二查詢運算子型樣包括一篩選或限制查詢運算子，其經組態成接收包括元素類型之一參數的一函式作為一參數，且傳回該元素類型之一可查詢類型。
5. 如申請專利範圍第 1 項所述之方法，其中該第二查詢運算子型樣包括一分類或排序查詢運算子，其經組態成接收包括元素類型之一參數的一函式作為一參數，且傳回該元素類型之一經排序的集合。
6. 如申請專利範圍第 1 項所述之方法，其中該查詢運算子型樣包括一複製移除查詢運算子，其經組態成傳回該元素類型之一可查詢類型。
7. 如申請專利範圍第 1 項所述之方法，其中該下一查詢運算子型樣包括一組查詢運算子，其經組態成接收元素類型 (作為該來源碼類型) 之一集合作為一參數，且傳回該元素

類型之一可查詢類型。

8. 如申請專利範圍第 1 項所述之方法，其中該下一查詢運算子型樣包括一條件選擇或篩選查詢運算子，其經組態成接收包括元素類型之一參數的一函式作為一參數，且傳回該元素類型之一可查詢類型。

9. 如申請專利範圍第 1 項所述之方法，其中該下一查詢運算子型樣包括一聚集查詢運算子，其經組態成接收包括元素類型之一參數的一函式作為一參數，且傳回一數值類型。

10. 如申請專利範圍第 1 項所述之方法，其中該下一查詢運算子型樣包括一聚集查詢運算子，其經組態成接收包括元素類型之一參數的一函式作為一參數，且傳回該下一元素類型之一可查詢類型。

11. 如申請專利範圍第 1 項所述之方法，其中該下一查詢運算子型樣包括一匹配查詢運算子，其經組態成接收包括元素類型之一參數的一函式作為一參數，且傳回可隱式轉換成一布林邏輯值之一類型。

12. 如申請專利範圍第 1 項所述之方法，其中該下一查詢運算子型樣包括一結合查詢運算子，其經組態成接收包括

元素類型之一參數及另一元素類型之一參數的一函式作為一參數，且傳回具有該下一元素類型之一可查詢類型。

13. 如申請專利範圍第 1 項所述之方法，其中該下一查詢運算子型樣包括一巢狀結合查詢運算子，其經組態成接收包括元素類型之一參數及具有另一元素類型之一集合的一參數的一函式作為一參數，且傳回該下一元素類型之一可查詢類型。

14. 如申請專利範圍第 1 項所述之方法，其中該下一查詢運算子型樣包括一分組查詢運算子，其接收一函式作為包括類型 K 之一參數(其中類型 K 係為一金鑰)及該元素類型之一集合之一參數的一參數，且傳回該下一元素類型之一可查詢類型。

15. 如申請專利範圍第 1 項所述之方法，其中推定出該下一查詢子句的一下一來源碼類型係為該元素類型，而該推定係基於該查詢運算子所考慮的該來源碼類型之該動作包含一上下文組件的動作，其本質上即時推定出該下一來源碼類型係為該元素類型。

16. 如申請專利範圍第 15 項所述之方法，其中該上下文組件基於該所推定的元素類型逐步地提供上下文資訊。

17. 一種用於一電腦系統的電腦程式產品，該電腦程式產品用於實施處理一查詢運算式的一方法，該電腦程式產品包含一或多個電腦儲存媒體，具有電腦可執行指令儲存於其上，當於一處理器執行時，該等電腦可執行指令造成該電腦系統實行該方法，該方法包括以下步驟：

(A) 存取一查詢運算式 (query expression)，該查詢運算式包括多個鏈結的查詢子句，該等多個鏈結的查詢子句包括至少一第一查詢子句鏈結至一下一查詢子句，該等多個鏈結的查詢子句的次序係界定於該查詢理解中，使得該次序係獨立於該查詢理解的基礎計算語言，其中：

(B) 該第一查詢子句包括一可查詢來源碼類型、一第一查詢運算子、及該來源碼類型的一元素類型，該元素類型具有與其相關聯的一控制變數，該控制變數具有藉由該查詢運算子所決定的一相關聯的範圍，該查詢來源碼類型指示待查詢的來源碼資料的類型，該元素類型指示該第一查詢子句的該等結果的該元素類型，該查詢運算子映射至根據一查詢運算子型樣所界定的一方法；及

(C) 該下一查詢子句包括待應用至該第一查詢子句的該等結果的一下一元素類型及一下一查詢運算子，該下一元素類型指示該下一查詢子句的該等結果的該元素類型，該下一查詢運算子映射至根據一下一查詢運算子型樣所界定的一下一方法；

(D) 評估該第一查詢子句，以從基於該查詢來源碼類型的該來源碼資料中獲得該元素類型的一組元素，

(E) 評估該第一查詢子句的步驟包括以下步驟：呼叫該方法以在該來源碼資料上實施該查詢運算子；

(F) 推定出該下一查詢子句的一下一來源碼類型係為該元素類型，其中該推定係基於該查詢運算子所考慮的該來源碼類型，其中該推定無須轉譯該整體查詢理解而實行；其中將該下一來源碼類型推定為該元素類型的該步驟係為即時的；

(G) 傳送該組元素至該下一查詢子句，其中當該下一查詢子句支援該控制變數的該範圍時，該相關聯的控制變數被傳送；

(H) 基於該推定出的元素類型逐步地提供上下文資訊至該下一查詢子句；及

(I) 評估該下一查詢子句，以從基於該下一查詢來源碼類型及該上下文資訊的該組元素中獲得該下一元素類型的一下一組元素，評估該下一查詢子句的步驟包括以下步驟：呼叫該下一方法以在該組元素上實施該下一查詢運算子。

18. 一種電腦系統，該電腦系統包含：

系統記憶體；

一或多個處理器；及

一或多個電腦儲存媒體，具有電腦可執行指令儲存於其上代表一使用者介面及一轉換組件，其中該使用者介面經組態成：

(A) 存取一查詢運算式 (query expression) 用於查詢一關聯式資料庫，該查詢運算式包括多個經排序的查詢子句，該等多個經排序的查詢子句包括至少一第一查詢子句，跟隨著為一下一查詢子句，該等多個經排序的查詢子句的次序係界定於該查詢運算式中，使得該次序係獨立於該查詢理解的基礎計算語言，其中：

(B) 該第一查詢子句包括一可查詢來源碼類型、一第一查詢運算子、及該來源碼類型的一元素類型，該元素類型具有與其相關聯的一控制變數，該控制變數具有藉由該查詢運算子所決定的一相關聯的範圍，該查詢來源碼類型指示待查詢的來源碼資料的類型，該元素類型指示該第一查詢子句的該等結果的該元素類型，該查詢運算子映射至根據一查詢運算子型樣所界定的一方法；及

(C) 該下一查詢子句包括待應用至該第一查詢子句的該等結果的一下一元素類型及一下一查詢運算子，該下一元素類型指示該下一查詢子句的該等結果的該元素類型，該下一查詢運算子映射至根據一下一查詢運算子型樣所界定的一下一方法；

(D) 評估該第一查詢子句，以從基於該查詢來源碼類型的該來源碼資料中獲得該元素類型的一組元素，

(E) 評估該第一查詢子句的步驟包括以下步驟：呼叫該方法以在該來源碼資料上實施該查詢運算子；

(F) 推定出該下一查詢子句的一下一來源碼類型係為該元素類型，其中該推定係基於該查詢運算子所考慮的該來源碼類型，其中該推定無須轉譯該整體查詢理解而實行；其中將該下一來源碼類型推定為該元素類型的該步驟係為即時的；

(G) 傳送該組元素至該下一查詢子句，其中當該下一查詢子句支援該控制變數的該範圍時，該相關聯的控制變數被傳送；

(H) 基於該推定出的元素類型逐步地提供上下文資訊至該下一查詢子句；及

(I) 評估該下一查詢子句，以從基於該下一查詢來源碼類型及該上下文資訊的該組元素中獲得該下一元素類型的一下一組元素，評估該下一查詢子句的步驟包括以下步驟：呼叫該下一方法以在該組元素上實施該下一查詢運算子。