

SQL Server® 2012  
T-SQL資料庫設計

## 第02章 全新商業智慧架構

---

Analysis Services 2012  
BISM表格式模型儲存技術  
PowerPivot



在第1章中我們介紹了商業智慧2.0的主要意涵，但畢竟本書還是以微軟商業智慧解決方案為主的，因此，我們在本章中，將會更深入地來介紹，微軟因應商業智慧2.0所推出的各項技術。

## 02-01 Analysis Services 2012

### 2.1.1 商業智慧語意模型

在Analysis Services 2012中，最大的突破就是徹底的改變了既有的商業智慧資料模型架構，將原有的統一維度模型（UDM, Unified Dimension Model）轉換為全新的商業智慧語意模型（BISM, Business Intelligence Semantic Model）。

一直以來，商業智慧的使用者必須在兩種型態的資料中擺盪，一個是以資料庫為核心的關聯性資料，最主要的功能在於產出商業智慧所需的明細型態資料；另一種則是行之有年的多維度分析資料，它的優點在於能以極高效能產出彙總型態資料。以往的商業智慧開發者都必須嫻熟這兩種不同的資料型態，關聯性資料必須運用T-SQL進行查詢，並使用Reporting Services之類的報表工具做為呈現端；多維度資料則是以MDX語言進行查詢，使用者可以利用Excel樞紐分析表、PerformancePoint Services或是第三方廠商提供的應用程式進行存取。

將兩種型態資料大一統的想法，以前並不是沒有，像是微軟的多維度分析支援鑽研功能，可以讓使用者調閱指定資料格對應的明細資料，但在實際應用上的限制頗多，不但無法變更欄位名稱、顯示計算欄位，同時也伴隨嚴重的效能問題。即使在Analysis Service 2005增加了報表鑽研功能，讓多維度分析能與報表服務進一步整合，但開發者仍必須實作多維度條件與報表參數間的對應，開發難度較高，且難以窮舉所有排列組合的可能性。

終於，在SQL Server 2012中，微軟向這不可能的任務遞出了第一份成績單。也就是正式推出了統合兩種資料模式的商業智慧語意模型。簡單地描述商業智慧語意模型的特色，就是「一種模型，滿足多種使用者需求」。在此，各位讀者必須要弄清楚商業智慧語意模型與現有模型間的關係，並非是在現有模型之外，推出了商業智慧語意模型，將舊有的統一維度模型（UDM）納入了商業智慧語意模型

之中，變成商業智慧語意多維度模型（BISM Multi-dimensional）。同時，商業智慧語意模型也推出了新的商業智慧語意表格式模型（BISM Tabular）。也就是說，在Analysis Services 2012中，其實只有「一種資料模型，兩種模式」（微軟最近很愛這種架構，Windows 8不也算是一種作業系統兩種模式）。

新的商業智慧語意表格式模型的資料儲存是透過以記憶體為基礎的xVelocity儲存體，它不但能夠存放明細層級的原始資料，同時也能以高效率運算來提供彙總資訊。此外，微軟也發展出全新的函數化語言DAX（Data Analysis eXpression）來提供查詢功能。

商業智慧語意模型兩種模式從儲存體、原生查詢語言都各有不同。但微軟的終極目標是要讓兩者在查詢語言上能夠彼此互通。當企業選擇開發商業智慧語意表格式模型時，除了原生的DAX之外，也能與MDX相容。也因此，企業原本在舊有商業智慧架構下的前端工具投資，也不至於因為商業智慧資料模型架構的變動，在一夕間化為烏有。

雖然目前的版本（SQL Server 2012 SP1）尚未做到，微軟已在2012年的SQL Server 專家大會中宣布，在不久的將來，即使過去開發的多維度分析Cube，只要升級為Analysis Service 2012商業智慧語意多維度模型之後（應該會是在SP2），就能夠以MDX與DAX進行查詢，而以DAX查詢為核心的Power View等工具，屆時也能和多維度分析模型相容。

除了前端工具可互通之外，企業原先為了商業智慧所設計的維度資料表與事實資料表也不會白費，開發者仍然可選擇使用新的商業智慧語意表格式模型架構來重新設計資料模型，以享受明細彙總合一的好處，整個商業智慧語意模型的架構，如圖02-1所示。

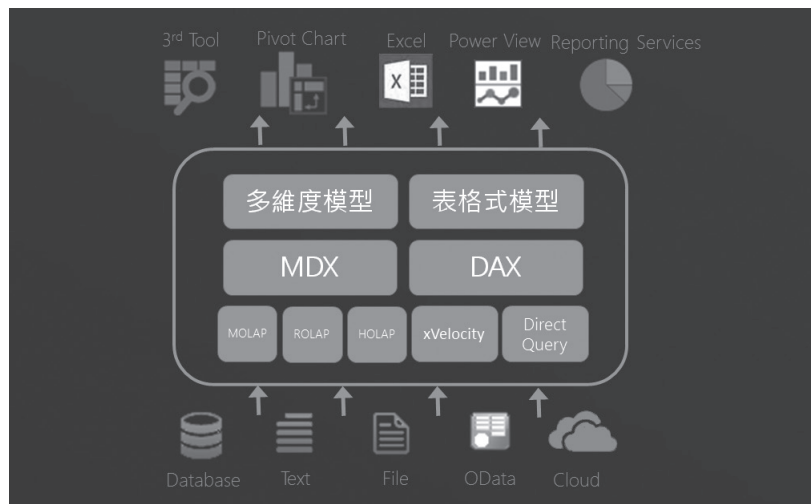


圖02-1：商業智慧語意模型架構

至於該選用哪種模式，就變成企業在導入商業智慧前，需要多加思考的重要議題。筆者從以下幾個角度來分析兩種模式的差異。

## 功能面

BISM表格式的問世最重要的精神就是提供企業一個「輕量級」的資料模型選擇，以便開發者可以從複雜的設定解脫，用最有效率的方法來實作大多數的商業分析。因此，當然會有許多功能在設計的便利性下，做了取捨，筆者將兩種資料模型間的功能做了以下的對照表。

	多維度模式	表格式模式
一般維度	●	●
殘缺維度	●	
父子式維度	●	▲
時間維度	●	●
角色扮演維度	●	
參考的維度	●	●
多對多維度	●	▲

表02-1：商業智慧語意模型兩種模式的功能比較（續）

	多維度模式	表格式模式
資料採礦	●	
支援圖檔		●
多國語言翻譯	●	
KPI	●	●
鑽研	●	▲
動作	●	
查詢明細		●
回寫	●	
資料分割	●	●
檢視方塊	●	●
維度層級安全性	●	●
維度成員安全性	●	●
資料格安全性	●	

表02-1：商業智慧語意模型兩種模式的功能比較

以下說明幾個主要的差異之處：

- **殘缺維度（Ragged dimension）與父子式維度（Parent-child dimension）：**  
這兩種功能主要是用來實作不對稱結構的維度，殘缺維度主要是運用在層級數固定，但局部不對稱的狀況。在BISM多維度模型可以設定「HideMemberIf」屬性來選擇性隱藏成員；而父子式維度則可以實作層級數不固定或是層級數超大的複雜結構，例如：組織架構、物料表（BOM, Bill of Material）或會計科目。基本上，在BISM表格式模型中，只支援對稱結構的維度，因此，是無法設定殘缺維度的。至於父子式維度，雖然在DAX函數中提供一系列的「Path」函數可用來實作接近父子式維度的效果，但仍必須事先知道維度的層級數，否則就無法做到不對稱以及超大層級數的維度。
- **角色扮演維度（Role-playing dimension）：**角色扮演維度主要適用於一個實體，並透過不同的關聯來轉換為不同的分析意義。舉例來說，在BISM多維度

模型中，只需一個時間維度表，就可以根據與事實資料表串接的鍵值不同，而變成下單日、入帳日、出貨日…等不同意義。但在BISM表格式模型裡，若是要同時呈現下單日、入帳日、出貨日三種意義，就必須載入3個相同的時間維度表。

- **多對多維度（Many-to-Many dimension）**：多對多維度，是BISM多維度模型中非常進階的技巧，它主要是透過中繼事實資料表的串接來實作複雜的計算需求，主要用在解決多維度分析不擅長的去重複計數（Distinct count）、複選題分析或者是分項加總不等於總數的特殊問題。就筆者過往的實務經驗來看，多對多維度最主要是運用在去重複計數，但由於BISM表格式模型本身就能快速計算去重複計數值，因此，就少掉一大半要運用多對多維度的可能性。至於「分項加總不等於總數」（例如，銀行的共用帳戶、業績分析時兩人拆分一個專案的獎金…），也可透過DAX語法的撰寫來達成。但在BISM多維度模型中，可以做到多重多對多維度彼此交纏的麻花餅式多對多維度（Pretzel-like Many-to-Many Dimensions），在BISM表格式模型就無法達成。

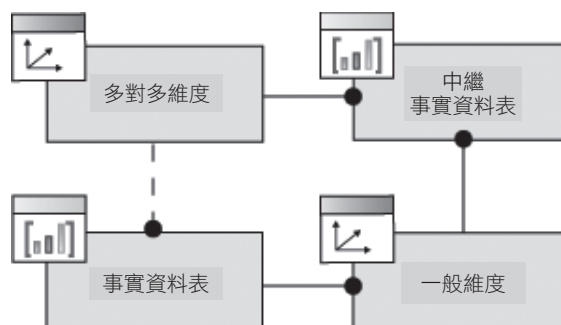


圖02-2：多對多維度

- **資料採礦**：微軟的Analysis Services其實也提供了資料採礦演算法，目前是放在多維度模式中。如果分析者需要進行預測性模型的建置，就必須使用BISM多維度模型。
- **圖檔支援**：目前僅有BISM表格式模型可以透過直接存取資料庫中的Varbinary(max)資料型別欄位，或是透過指定圖檔位置的Url字串，將圖檔納入資料模型之中。如此一來，就能夠大幅提高分析報表的視覺豐富性。這是BISM多維度模型目前無法做到的。但目前微軟已在2012年SQL Server專家大會

中，宣布在下一版中（應該會是SQL Server 2012 SP2），多維度分析將會加入指定圖檔位置的Url字串之功能。

- **多國語言翻譯**：在BISM多維度模型中，可透過設定多個語言參考，偵測前端使用者電腦語言與地區設定，給予對應語系的翻譯內容。這項功能是在BISM表格式模型完全無法實作的。
- **回寫（Write-back）**：回寫是指提供API讓使用者修改Cube內的數值，主要用在預算編列或模擬試算使用。目前BISM表格式模型並未提供這種API。但筆者認為影響不大。現在即使是BISM多維度模型也只提供底層回寫（最細粒度資料），在BISM表格式模型中，我們可以將須調整的資料以Excel方式儲存，透過直接修改Excel，即可達到近似的效果。
- **鑽研與動作**：鑽研的目的主要是為了訂閱明細，然而BISM表格式模型的最大優點就是能夠兼容彙總與明細查詢。因此，它的鑽研機制會與BISM多維度模型不同。至於動作（Action）是當使用者滑鼠點觸到維度或者資料格時，引發驅動其他的外部程式，目前BISM表格式模型也未提供這類的API。
- **資料格安全性**：BISM多維度模型是以資料格作為分析的基礎，BISM表格式模型則是以欄位作為分析的基礎。在BISM表格式模型中，並無資料格的概念，因此，無法實作資料格安全性。不過，資料格安全性在過往的經驗中，使用的頻率極低，大多數採用維度層級與維度成員安全性，而這些在BISM表格式模型中都有支援。

照這樣看來，BISM表格式模型有一天會完全取代掉BISM多維度模型嗎？筆者認為**不能、也不應該**。不能的原因在於，兩者模型的設計概念完全不同。表格式模型中沒有資料格的概念，在功能面上很難找到一對一的取代。至於「不應該」的原因在於，BISM表格式模型的最大賣點就是，讓使用者能夠以較簡便的方法來進行開發，如果為了功能完整性卻讓開發變得愈來愈困難，那就失去推出新模型的意義了。套用微軟高層的一句話，他認為未來的BISM多維度模型就像C++，永遠不會消失。但僅在遇到非常困難的商業問題時，才需要實作這個完整而複雜的機制；至於BISM表格式模型則是使用C#，學習十分容易上手，也能被多數人接受。

說實在話，筆者從開始學習、到教授多維度模型長達十年。對於多維度分析自有非常深的情感，所以在BISM表格式模型問世時，情感上仍難以接受，但當我清



楚了兩者的實際定位後，便能彈性地選擇所需的資料模型模式。大致上，八成以上的商業智慧都可透過**BISM表格式模型**來快速導入，只有當涉及複雜的組織或財務議題時，才非得請出多維度模型不可。若要說微軟哪裡還不盡完善，那就是希望Power View支援**BISM多維度模型**的那一天趕快到來，如此一來，才算是真正的無縫接合兩種模式。

### 前端工具面

如果企業內部已經有支援微軟多維度模型的前端工具，那麼這些工具都可以延用至新的**BISM表格式模型**（但是無法呈現圖檔以及查詢明細）。如果各位希望使用微軟全新的視覺化工具Power View，那目前您就必須改採用**BISM表格式模型**。雖然日前微軟曾承諾會讓Power View支援**BISM多維度模型**，但是目前尚未有具體的時程，但是在2012 SQL Server專家大會中已經有現場展示，應該會在SQL Server 2012 SP2中實現。

### 軟體授權面

SQL Server 2012分為三個主要版本：標準版、商業智慧版以及企業版。在標準版中，只提供**BISM多維度模型**，且不支援半加成（semi-additive）量值的設計。如果要使用**BISM表格式模型**或完整**BISM多維度模型**的功能，就必須選擇商業智慧版或企業版。關於詳細的各版本間比較，請參考《第3-1節：安裝SSRS 2012的前置需求》。

### 硬體配備面

**BISM多維度模型**主要利用將彙總儲存於多維度結構體（硬碟），來達到快速查詢的效果；而**BISM表格式模型**則是將彙總儲存於記憶體。因此，兩者在硬體需求上，截然不同，如果要使用**BISM表格式模型**，筆者建議硬體的記憶體擴充性會是非常重要的考量。

## 2-1-2 BISM多維度模型

**BISM多維度模型**就是以往大家所熟悉的多維度分析。多維度分析的基礎在於事先產生資料彙總，分析人員也可以根據彙總及所需的資料檢視角度，快速地組合產生所需的決策資訊。



其中，「分析角度」就稱之為**維度（Dimension）**；所有排列組合資料格集合，我們稱之為**立方體（Cube）**；在資料格彙總的數值，我們稱之為**量值（Measure）**。多維度分析透過事前的結構定義，將使用者所需要的分析維度列舉後（日常報表中的橫軸、縱軸與篩選條件），透過多維度分析引擎產生所有分析角度的排列組合，最後，將量值依照對應的排列組合依序填入至彙總，供日後分析者查詢存取，這就是多維度分析最基礎的概念。

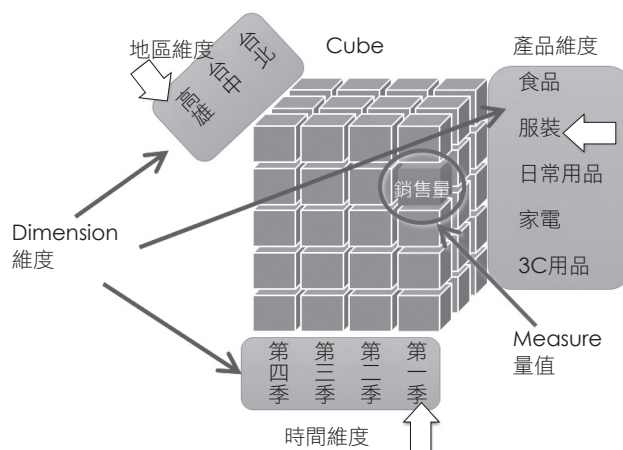


圖02-3：多維度分析

在BISM多維度模型中，「維度」指的是分析的**商業實體（Business Entity）**，所謂的商業實體，指的是分析的總和抽象觀念。**維度資料表（Dimension Table）**就是用來儲存商業實體的資料表，它最大的特色在於，所有的分析實體都會具備一個最小的分析資料**粒度（Granularity）**，而這個最小粒度必須是唯一且不重複的，因此，所有的維度資料表都具備**主索引鍵（Primary Key）**。除了最小粒度之外，同時還包含描述此商業實體相關的敘述性欄位，我們稱之為「屬性（attribute）」。

例如，以銷售分析來說，「時間」就是一個商業實體，因為它包含了與「時間」相關的所有分析角度。而時間的最小粒度就是「日期」，因此，時間維度表常會使用「日期」來做為主索引鍵。除了這個主索引鍵之外，其他如：「日曆年」、「日曆季」、「日曆月」或「星期幾」等等，都是用來描述時間的敘述性欄位，也都維護在維度資料表中。除了時間外，產品別、部門別、會計科目別…等，也都是典型的分析維度。

至於**事實資料表（Fact Table）**主要是用來記錄企業營運資訊「可彙總計算（additive）」的「數值（numeric）」細節，並透過大量的**外部索引鍵（Foreign Key）**與其他維度資料表連結。一般而言，事實資料表資料較為龐大，同時所記錄的資訊會與時間密切相關，資料量也會隨時間遞移，而累積膨脹。

BISM多維度模型的資料儲存體分為兩種主要模式。分別為**多維度OLAP（MOLAP, Multi-dimensional OLAP）**以及**關聯式OLAP（ROLAP, Relational OLAP）**。以上這兩種儲存模式最大的分野其實就在於彙總資料與明細資料的儲存位置。

MOLAP是將彙總以及明細資料同時儲存於**多維度結構體**之中（多維度結構體，其實就是檔名經過編碼過的一系列Xml檔），而ROLAP則是將彙總與明細資料同時儲存於**關聯式資料庫**中，如圖02-4。這兩種有何差別呢？筆者從幾個角度來分析其中差異。

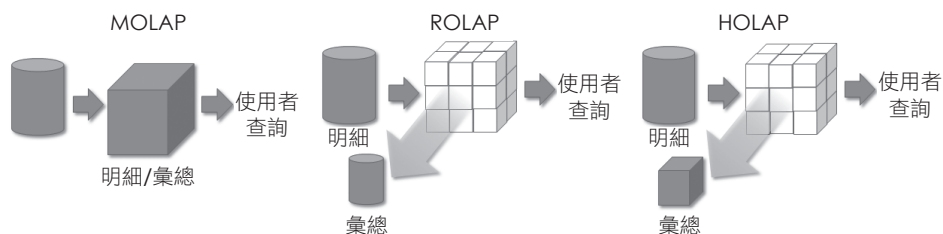


圖02-4：多維度分析儲存模式示意圖

首先，從「查詢效率」面來看，當查詢MOLAP時，系統就直接調閱對應的匯總即可；但當查詢ROLAP時，則必須先將查詢的多維度MDX語言轉換為二維的T-SQL查詢語法，以從關聯式資料庫中傳回查詢彙總結果，並將二維的回傳資料表轉換為多維的資料格。相較之下，MOLAP不須經過這麼多的反覆轉換，因此，在查詢彙總資料時，**MOLAP的效率會較ROLAP的效率來得好**。查詢MOLAP資料源資料庫時，即使關閉未啟動也不會影響查詢，至於ROLAP，必須確保SSAS以及資料庫隨時都在運轉狀態中。

不過，從「資料儲存」的層面看來，MOLAP會將明細資料抄錄至多維度結構體中，同時再產生額外的彙總資料，相較於ROLAP，並不需要額外複製明細資料，僅需要利用資料表儲存彙總資料，因此，**ROLAP需要的儲存硬碟空間較小**。

當多維度資料「處理彙總」時，受限於MOLAP必須先將明細資料載入多維度結構體後，方能計算彙總，ROLAP省去了載入明細資料的步驟，因此，在少量資料的狀況下，ROLAP處理的效率會較快速。不過，在大量資料的狀況下，由於ROLAP所產生的彙總資料表必須是完全索引化（full-indexed）的資料表，因此，建立索引的時間會拖慢效率，反而會比MOLAP來得慢。

總結來說，MOLAP查詢速率較快，但缺點是耗用較多的硬碟空間。ROLAP查詢效率較慢，但比較節省硬碟空間。就處理效能來看，資料量小的時候，是ROLAP速率較快，但當要處理巨量資料時，MOLAP反而處理時間較短。以大多數企業的考量來看，硬碟的成本相對來說，是屬於較便宜的項目，因此，在多維度分析中，大多建議採用查詢效能較佳的MOLAP架構。目前市場上以MOLAP為市場主流，例如，微軟、IBM Cognos、Oracle Hyperion都屬於MOLAP的陣營，目前僅有SAP BO以及MicroStrategy屬於ROLAP的技術。

MOLAP與ROLAP看起來像是天秤的兩端，另外，在多維度技術中還有第三種選擇稱之為混合式OLAP（HOLAP, Hybrid OLAP）。HOLAP是擷取ROLAP的優點，將明細資料儲存於關聯式資料庫中，而彙總資料仍舊儲存於多維度結構體內。因此，查詢效率近似MOLAP，而硬碟耗用量也介於兩者之間，所有的項目都是第二名，筆者並不建議使用。

### 2-1-3 BISM表格式模型

在BISM表格式模型中，不像BISM多維度模型這般，將維度資料表與事實資料表分得如此涇渭分明。如圖02-5所示，在BISM表格式模型中只有一系列的「資料表」，每個資料表都可以是維度資料表，也都可以作為事實資料表。

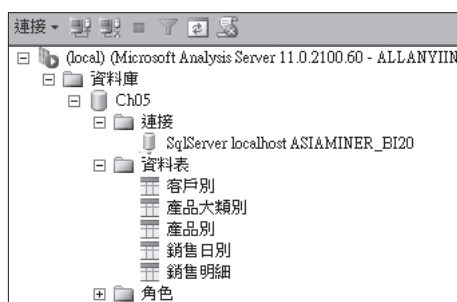


圖02-5：表格式模型

但這並不代表企業原先設計的維度資料表與事實資料表，在BISM表格式模型中就不管用了。事實上，這些概念都還可以持續沿用，只不過，BISM表格式模型賦予了更大的設計彈性。對於初學者來說，筆者還是建議遵循經典的維度資料表與事實資料表概念。

BISM表格式模型的重點在於定義資料表之間的關聯性，以及各個資料表內的資料行所代表的商務意義。在後續的章節中，我們會介紹如何使用SSDT（SQL Server Data Tool）進行表格式模型的設定，至於資料行所代表的商務意義，除了資料行名稱之外，還包括該資料行未來在視覺化工具呈現時所代表的意義，例如，在卡片檢視中的標題與縮圖。



圖02-6：卡片檢視

至於資料儲存體方面，在多維度模型還在針對彙總與明細該放在多維度結構體或關聯式資料庫而爭議不休之時，BISM表格式模型則提出了第三條路：放在記憶體之中。

各位其實可以把BISM表格式模型看成一種全新的資料庫結構，只不過，這個資料庫沒有實體的儲存體，而是把所有的資料都存放在記憶體之中。一般資料庫絕大多數的計算門檻都是在於系統的I/O，因此，當資料全部都放在記憶體時，效能便會大幅提升。

在記憶體中的資料是以甚麼型態儲存呢？目前BISM表格式模型提供了兩種不同的儲存模式，分別是xVelocity模式（在SQL Server 2012上市前，微軟是稱作Vertipaq模式）以及DirectQuery模式。簡單來說，xVelocity模式是一種以資料行為基礎（Column-based）的索引機制，透過高度資料壓縮，可大幅減少存放在記憶體內的資料體積，以達到相較於傳統資料庫百倍以上的運算效能。至於DirectQuery模式，則僅是將BISM表格式模型當做語意邏輯層，並不會產生實際

的記憶體快取，它只是通過此邏輯層，將查詢的DAX語言轉換為T-SQL查詢語法，向原始資料庫進行查詢。

在SQL Server 2012中，比較建議採用xVelocity模式，因為它能真正地達到高速運算的效能。至於DirectQuery模式，相較之下，效能較為緩慢，而且限制不少，來源資料庫也必須是SQL Server。筆者認為它比較接近在SQL Server 2008 R2中的報表模型的概念，其唯一的好處就是，可以即時存取來源系統上的資料。

## 02-02 BISM表格式模型儲存技術

接著，我們將要再詳細說明BISM表格式模型中，兩種儲存模式的技術細節。

### 2-2-1 xVelocity記憶體運算引擎

xVelocity是一種以資料行為基礎（column-based）的索引儲存機制。首先，來比較一下這與以資料列為基礎（row-based）的一般資料庫儲存機制有何差異。

包括SQL Server在內的關聯式資料庫，都是以資料列為基礎的儲存機制。目前SQL Server儲存檔案是以每8K做為一個分頁，隨著一筆資料列大小不一定，每個分頁可儲存一個或多個資料分頁。當SQL Server要讀取某筆資料列，就必須先找到存放這筆資料列的資料分頁。

當資料量變大時，為了要能夠快速地存取資料，就會建立索引（Index）。以叢集式索引為例，會在資料庫中根據指定鍵值進行排序，日後即可透過排序過的資料分頁以及索引結構，快速地找出所需要的資料列。即使如此，若希望針對特定欄位進行分析，那麼在以資料列為基礎的儲存結構中，仍必須將所篩選出來的資料列中的所有欄位，都先進行資料表遍歷（full table scan），而這部分工作的效能與所使用的欄位數量並沒有顯著的關係，即使加上了索引，在商業智慧分析這種經常使用到彙總、連結等情境下，仍舊效果不彰。

## 02-02

ID	Product Name	Product Category	Brand
1	iPhone	智慧型手機	Apple
2	Lumia	智慧型手機	Nokia
3	Galaxy	智慧型手機	Samsung
4	iPad	平板電腦	Apple
5	Galaxy	平板電腦	Samsung
6	Lumia	智慧型手機	Nokia

Row ID	Row Data
1	1, iPhone,智慧型手機, Apple
2	2, Lumia,智慧型手機, Nokia
3	3, Galaxy,智慧型手機, Samsung
4	4, iPad,平板電腦, Apple
5	5, Galaxy,平板電腦, Samsung
6	6, Lumia,智慧型手機, Nokia

圖02-7：資料列基礎儲存

至於以資料行基礎的儲存結構的思考邏輯，就恰好相反。如圖02-8，是以資料行（Column）作為儲存的基礎單位。在SQL Server中，是將一個資料行當成一個資料分頁。在絕大多數的商業智慧分析中，即使資料表欄位再多，但其實一張圖表或交叉分析中所需用到的欄位數並不多，透過以資料行為基礎的儲存結構，可以省卻找尋資料分頁的問題，還能免除資料表遍歷的動作，只針對所需要的資料行進行分析，即可達到大幅提升查詢速度的效果。

雖然在一般商業智慧的彙總分析上變快了，但相對來說，這種儲存結構對於單筆資料行的處理就會顯得較不擅長。舉例來說，若要新增或刪除一筆資料，就必須反覆地針對儲存結構中的每個欄位，針對正確的資料列位置進行刪除與新增。目前在微軟的xVelocity儲存體中，為了效能考量，是不支援針對單筆資料進行新增、刪除、修改的操作。但微軟已在2012年的SQL Server專家大會中，宣布下一代SQL Server將推出代號為Hekaton的叢集式資料行儲存技術，屆時，就能夠運用記憶體儲存結構，快速地針對單筆資料進行新增、刪除與修改，這意味著，記憶體運算將從商業智慧分析拓展至線上交易的應用。

ID	Product Name	Product Category	Brand
1	iPhone	智慧型手機	Apple
2	Lumia	智慧型手機	Nokia
3	Galaxy	智慧型手機	Samsung
4	iPad	平板電腦	Apple
5	Galaxy	平板電腦	Samsung
6	Lumia	智慧型手機	Nokia

Column ID	Column Data
ID	1,2,3,4,5,6
Product Name	iPhone, Lumia, Galaxy, iPad, Galaxy, Lumia
Product Category	智慧型手機, 智慧型手機, 智慧型手機, 平板電腦, 平板電腦, 智慧型手機
Brand	Apple, Nokia, Samsung, Apple, Samsung, Nokia

圖02-8：資料行基礎儲存



至於Analysis Services 2012所自豪的xVelocity技術，則是將傳統的資料行基礎儲存再進一步改良。除了仍舊以資料行做為資料儲存單位之外，同時還為每個資料行建立了一個獨立的**目錄結構（Dictionary）**，以存放此資料行內的去重複值，同時將原先的資料行內的值轉換成為此目錄對應的Bitmap索引值，如圖02-9。因為多了這層資料目錄結構，一方面可讓xVelocity快速地計算**去重複計數**，也能使資料量大幅**壓縮**，以提升記憶體運算的資源耗用效率。平常存放在記憶體中的資料就是以這種壓縮的型態儲存，但當xVelocity引擎在查詢結果完成後，則能夠進行動態地解壓縮，以還原成正確的查詢結果。

ID	Product Name	Product Category	Brand
1	iPhone	智慧型手機	Apple
2	Lumia	智慧型手機	Nokia
3	Galaxy	智慧型手機	Samsung
4	iPad	平板電腦	Apple
5	Galaxy	平板電腦	Samsung
6	Lumia	智慧型手機	Nokia

Column ID	Dictionary	Value
ID	1,2,3,4,5,6	0,1,2,3,4,5
Product Name	Galaxy, Lumia, iPad, iPhone	3,1,0,2,0,1
Product Category	平板電腦, 智慧型手機	1,1,1,0,0,1
Brand	Apple, Nokia, Samsung	0,1,2,0,2,1

圖02-9：資料行基礎儲存（xVelocity）

圖02-10是比較使用xVelocity前後的效能差異數據（1T資料量，600萬以上資料筆數，共計24個資料表）。根據設定的9種查詢情境，xVelocity可獲得平均8~20倍的效能提升。原則上，分析時，所使用到的欄位數較少，則xVelocity就能顯示出較為卓越的查詢效能。如果要調閱一個欄位數很多的資料表中的全部欄位（就像是在SQL Server上最常做的SELECT \*操作），此時，xVelocity就會變得毫無用武之地。在此證明了，xVelocity只適合用於商業智慧分析，而不適合做為一般OLTP資料庫，這與Oracle所推出的記憶體運算OLTP資料庫TenTimes恰好相反。



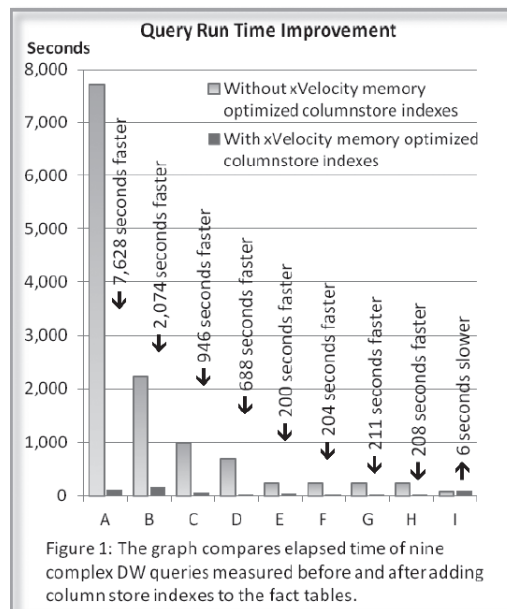


圖02-10：使用xVelocity前後的效能比較

xVelocity是一種以資料行為基礎的記憶體儲存引擎，而它原生的查詢語言正是之前所提到的資料分析語言（DAX, Data Analysis eXpression）。各位可能會擔心，現行的OLAP前端工具都是以MDX為查詢語言，是否這些工具的投資都會付諸流水？這大可不必煩惱，因為對於微軟來說，新的商業智慧架構是一種模型（商業智慧語意模型）中有兩種模式（多維度以及表格式），因此，不論是使用哪種查詢語言，都是互通的。如圖02-11所示，當使用者發動DAX或MDX查詢時，首先，會經過Analysis Services的語法解析器判別查詢語言種類，如果是DAX，則直接將語法送至DAX公式引擎中進行查詢處理；如果是MDX，則會再多經一層的語法轉換，將MDX轉換成對應的DAX操作。

相較之下，如果是查詢表格式模型，雖然兩種語言都通，但畢竟DAX少了一層轉換的步驟，因此，效能會優於MDX查詢，再加上DAX能同時涵蓋彙總與明細的查詢，所以DAX仍舊是我們建議的查詢語言首選。

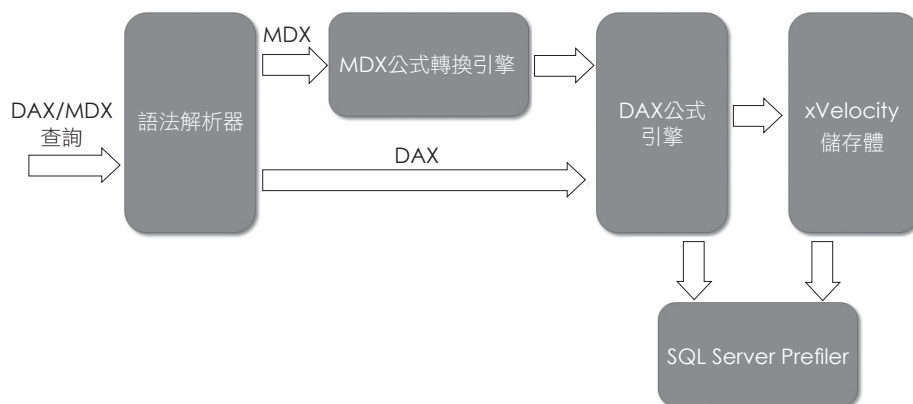


圖02-11：xVelocity查詢執行流程

### 2-2-2 DirectQuery模式

不管是多維度模型或表格式模型，都有其特殊的資料儲存體。在多維度模型，是存放在硬碟上的多維度結構體中；而表格式模型，則是存放在xVelocity儲存體中。在多維度分析中，提供了ROLAP，它可以將明細與彙總都擺放在關聯式資料庫中；而表格式模型中，也提供了一種類似的機制，我們稱之為DirectQuery模式。

DirectQuery模式是完全沒有任何明細或彙總的快取（這與ROLAP有點不同，ROLAP是可以設定產生額外的彙總的）。當使用者發動查詢時，它會將DAX語法轉換為來源資料庫（一定是SQL Server）能夠接受的SQL語法，並將此語法轉交給來源資料庫進行查詢。

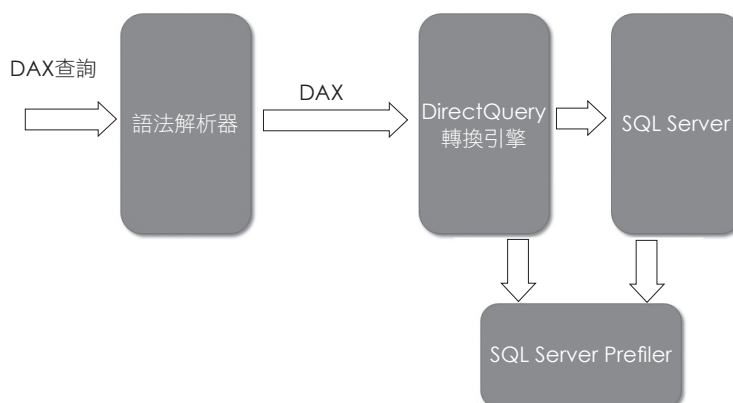


圖02-12：DirectQuery查詢執行流程

目前使用DirectQuery模式，會有以下的限制：

- DirectQuery目前只支援DAX查詢，而無法使用MDX查詢。
- 即便是DAX語法，也因DirectQuery的精神在於轉換為原生的資料庫SQL語法，再回拋至資料庫查詢，因此，未必所有的DAX語法都可以支援。
- DirectQuery不支援「計算資料行」，只支援「量值」的計算。
- DirectQuery的來源資料「一定」只能是SQL Server，目前不支援其他種類的資料庫。

乍看之下，DirectQuery真是有點食之無味，甚麼都不支援，到底存在的價值為何？其實就只有一個最重要的優點，那就是落實了真正的「即時」商業智慧。

所謂的即時商業智慧，原因在於DirectQuery模式未完全以各種形式儲存資料快取，而是直接向來源資料庫進行查詢，因此，它是完全與來源資料無時差的。不過，各位不要聽到即時商業智慧就覺得很高興，這裡的「即時」指的是當使用者發動查詢時，回傳的結果會是**查詢當下的資料結果**，但千萬別忘了，這不包括查詢結果會「即時」回傳至使用者端。因為DirectQuery模式沒有任何快取，因此，它的查詢效能是較差的。

為了解決查詢效能的問題，通常筆者會建議在來源資料庫端配合設定**資料行存放區索引（ColumnStore Index）**，也就是運用資料庫端所提供的以資料行為基礎的索引技術（目前僅有SQL Server 2012才提供）。如此一來，就可以確保即便使用了DirectQuery模式，也能有合理的回應速度。

原則上，若要導入商業智慧表格式模型，筆者仍會比較推薦使用xVelocity模式，除非您只想把表格式模型純粹視為來源資料庫的語意邏輯層，來做為即時資訊查詢之用，而不在乎它的查詢效能，此時，才比較適合使用DirectQuery模式。在後面的章節中，我們也會特別標記DirectQuery模式無法相容的DAX函數。

## 02-03 PowerPivot

在上一節中，我們花了不少篇幅介紹xVelocity技術，事實上，除了商業智慧語意表格式模型之外，xVelocity還有另一種形式，那就是「PowerPivot」。

初聽到PowerPivot這個字眼時，很容易會聯想成這是一個「更有威力的樞紐分析表」，而錯把它當成商業智慧的前端使用者介面。但實際上，它其實是一種資料來源，也就是個人版的xVelocity資料儲存體。在PowerPivot中，包含了兩個主要的產品模組，分別是PowerPivot for Excel與PowerPivot for SharePoint。

### 2-3-1 PowerPivot for Excel

PowerPivot for Excel首次問世於SQL Server 2008 R2上市時，它是在Excel 2010上的增益集（需額外安裝）。在此要注意的是，PowerPivot並非Office家族的產品，從它完整的產品名稱「Microsoft SQL Server 2012 PowerPivot for Microsoft Excel 2010」就可以輕易判斷，它其實是SQL Server家族的產品，更精確地說，它隸屬於Analysis Services產品線。

PowerPivot for Excel的產品定位就是個人版的商業智慧語意表格式模型，它透過商務使用者所熟悉的Excel介面做包裝，讓分析者可以便利地處理資料載入、商務邏輯設定、資料表關聯並將資料轉換為xVelocity儲存體。使用者還可以利用熟悉的Excel樞紐分析表技術存取PowerPivot儲存體，以彈性的設計分析儀表板。整個流程都可以在Excel上獨力完成，而這也正是微軟一直強調的「自助式」商業智慧。

在安裝了PowerPivot for Excel之後，Excel會加入名為「PowerPivot」的索引標籤，圖02-13。點選「PowerPivot視窗」按鈕後，即可啟動PowerPivot設計介面。



圖02-13 PowerPivot for Excel 2010

## 02-03

### PowerPivot

03

04

05

06

07

08

當啟動PowerPivot設計介面之後（見圖2-14），同時也會建立一個本機的Analysis Services執行個體，分析者可利用PowerPivot for Excel所提供的資料提供者，將資料載入本機的Analysis Services之中，以轉換為xVelocity儲存體。

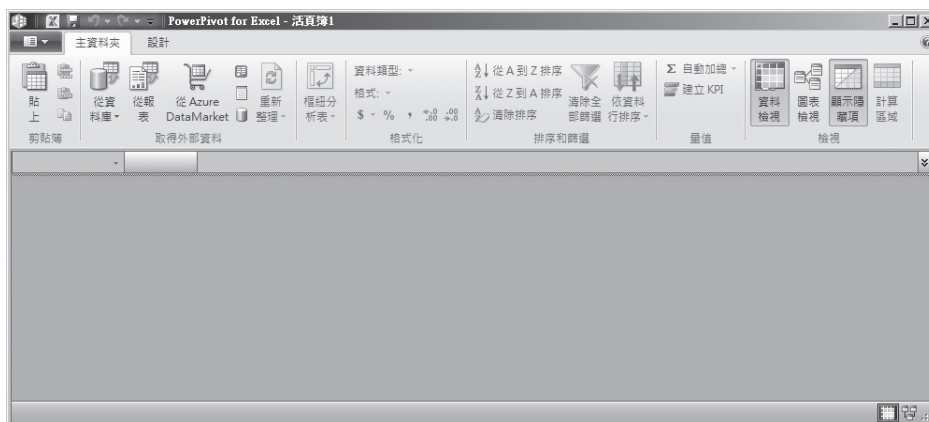


圖02-14：PowerPivot for Excel設計介面

PowerPivot for Excel除了提供了xVelocity引擎來儲存資料之外，還提供了對應的資料模型設計介面，用來設計商業智慧語意表格式模型。當儲存活頁簿時，PowerPivot for Excel會將資料模型定義備份檔（item1.abf）與快取資料（item1.data）序列化，並以Xml的格式儲存於活頁簿中。

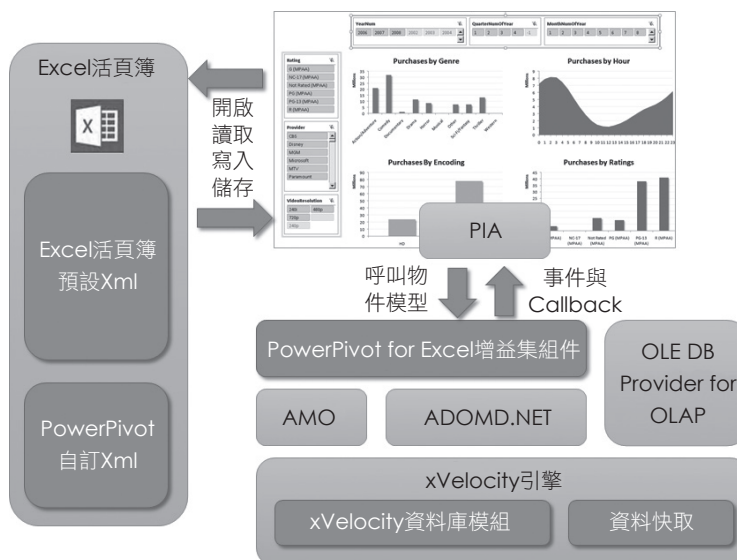


圖02-15：PowerPivot for Excel架構

既然說PowerPivot是個人版的商業智慧語意表格式模型，那它與真正的商業智慧語意表格式模型間有甚麼差異呢？以下整理出兩者的差異之處：

- PowerPivot for Excel定位為個人版使用，並無法設定安全性。如果將PowerPivot活頁簿分享給其他使用者，也等於將資料模型內的邏輯奉送給他人，至於其中的資料，由於開啟PowerPivot活頁簿時，本機Analysis Services會還原資料快取備份，並試圖根據原有的連線資訊重新載入來源資料。如果當初連線字串是儲存帳號與密碼，而這台電腦又與資料庫位於同樣的網段，就可以重新載入來源資料。筆者在此建議，PowerPivot for Excel還是盡量以本機使用為主，最好避免將PowerPivot活頁簿轉寄或分享給他人。如果希望將PowerPivot的成果分享給他人，則應該採用後續介紹的PowerPivot for SharePoint機制。
- 如果您使用的是32-bit的Excel，由於32-bit系統使用的記憶體上限為2G，因此Excel以及xVelocity所共用的記憶體總量就不能超過2G。如果您用的是64-bit的Excel，雖然沒有記憶體使用量的上限限制，但由於大多數PowerPivot for Excel的使用情境是將它發行至SharePoint之上。但SharePoint本身具有上傳檔案大小2GB的限制，所以雖然記憶體使用沒了上限，但在上傳檔案時，還是必須要注意上傳檔案的限制。再次提醒各位，PowerPivot活頁簿儲存的是資料模型定義與xVelocity資料庫的備份，因此，活頁簿遠小於實際資料的大小。若您要處理的資料量較大，正式的商業智慧表格式模型將會是較佳的選擇。或者您也可以先使用少量資料設計雛形，待發佈至PowerPivot for SharePoint之後，再套用正式的資料量。
- PowerPivot for Excel無論在開發或查詢時，皆會耗用本機的記憶體資源，至於商業智慧表格式模型，則是從開發階段起，就以耗用伺服器的記憶體資源為主。
- PowerPivot for Excel中並無包含自動化的API，無論是開發、查詢、更新資料，都只能手動作業。目前未開放的像是AMO這類的程式介面，讓使用者可進行延伸開發。如果您希望做到資料自動更新等機制，最好還是將它發行至PowerPivot for SharePoint。

到了Excel 2013中，大體上的架構沒有太大變化，唯一不同之處在於，PowerPivot for Excel不再是獨立的增益集，而是內建於Excel 2013之中（預設仍為停用，需手動啟用）。相較於之前一個活頁簿只夠儲存一個PowerPivot資料模型，到了Excel 2013，每個工作表都能夠維護自己的PowerPivot資料模型結構。



圖02-16：PowerPivot for Excel 2013

## 2-3-2 PowerPivot for SharePoint

PowerPivot for SharePoint是 Analysis Services 與 Microsoft SharePoint Server 的整合，當我們將PowerPivot活頁簿上傳至SharePoint之後，系統會將活頁簿內的 Analysis Services備份還原至實體的Analysis Services資料庫中，並將活頁簿內的工作表，逐一透過Excel Services轉換為網頁。

PowerPivot for SharePoint中所使用的Analysis Services資料庫，其實就是商業智慧語意表格式模型（如圖02-17），但其操作涉及了SharePoint Server與Analysis Services間的複雜協同作業。因此，在安裝SQL Server 2012時，特別將PowerPivot for SharePoint的整合模式獨立為特別的安裝選項（下一章中，將會介紹詳細的安裝過程）。

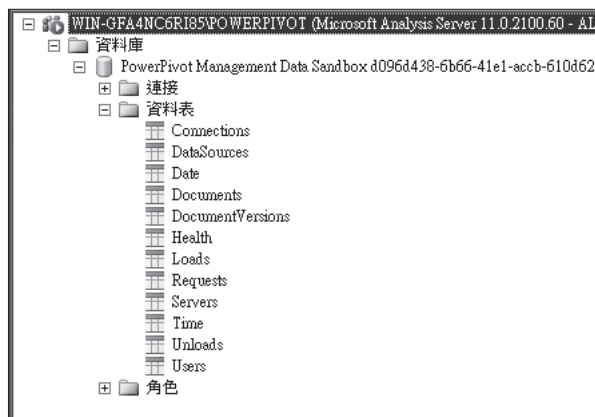


圖02-17：PowerPivot整合之Analysis Services

當安裝了PowerPivot for SharePoint整合模式之後，SharePoint Server上也會安裝一系列的管理頁面，包括了：



- **PowerPivot圖庫**：這是專屬於PowerPivot與Power View的文件庫型態，它以Silverlight技術為核心，可以透過不同瀏覽模式列出所有的PowerPivot文件，見圖02-18。
- **PowerPivot管理儀表板**：其中，包含了PowerPivot系統診斷數據收集器，可收集整個SharePoint上PowerPivot的運行數據，並透過預設的PowerPivot儀表板範本，協助管理者了解目前全體PowerPivot的維運狀態。
- **資料更新計畫**：使用者可利用此介面設定PowerPivot資料的更新排程，讓資料定期更新，無須透過資訊單位便可完成。

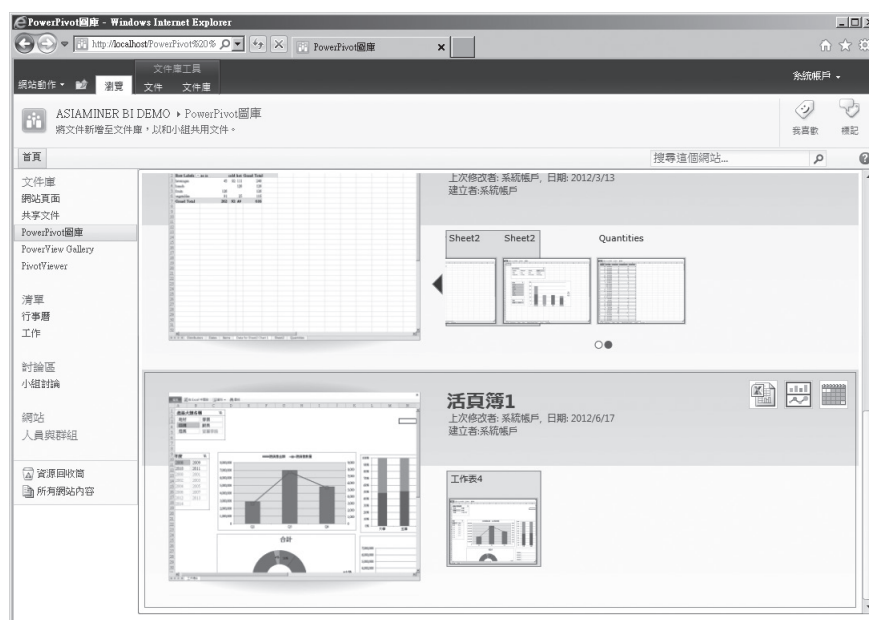


圖02-18：PowerPivot圖庫

使用PowerPivot for SharePoint最重要的特點在於，使用者僅需透過存放在文件庫內的Excel檔案路徑，即可做為存取背後對應的商業智慧語意表格式模型的連線字串。當使用者從瀏覽器發動查詢時，會透過SharePoint Excel Services並根據此Excel檔案路徑，來找到對應的Xlsx檔案。接著，PowerPivot Services會負責將PowerPivot活頁簿內的資料需求，轉交給背後的Analysis Services，以取回所需之資訊，並更新Excel Services活頁簿的內容，如圖02-19。

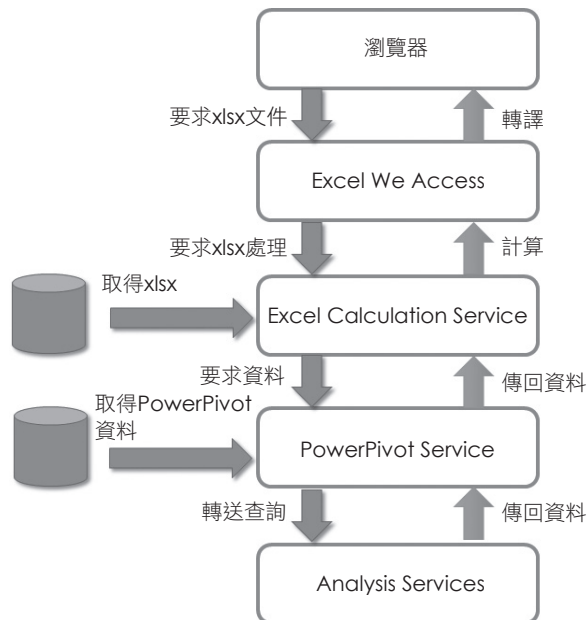


圖02-19：存取PowerPivot

PowerPivot for Excel可協助使用者運用Excel設計資料模型，並透過Excel樞紐分析表，直接存取xVelocity儲存體來設計出儀表板；接著，將Excel上傳至SharePoint，此時，PowerPivot for SharePoint會接手將活頁簿內的xVelocity儲存體還原至伺服器端，並透過Excel Services將Excel活頁簿轉換為網頁，最後，再提供一系列的管理與排程工具，讓使用者可以「自食其力」地完成整個商業智慧的開發過程。同時，將商業智慧所耗用的資源從本機轉移至伺服器，一方面得以提升執行效能，也便於分享給企業內部其他人員使用。◆◆