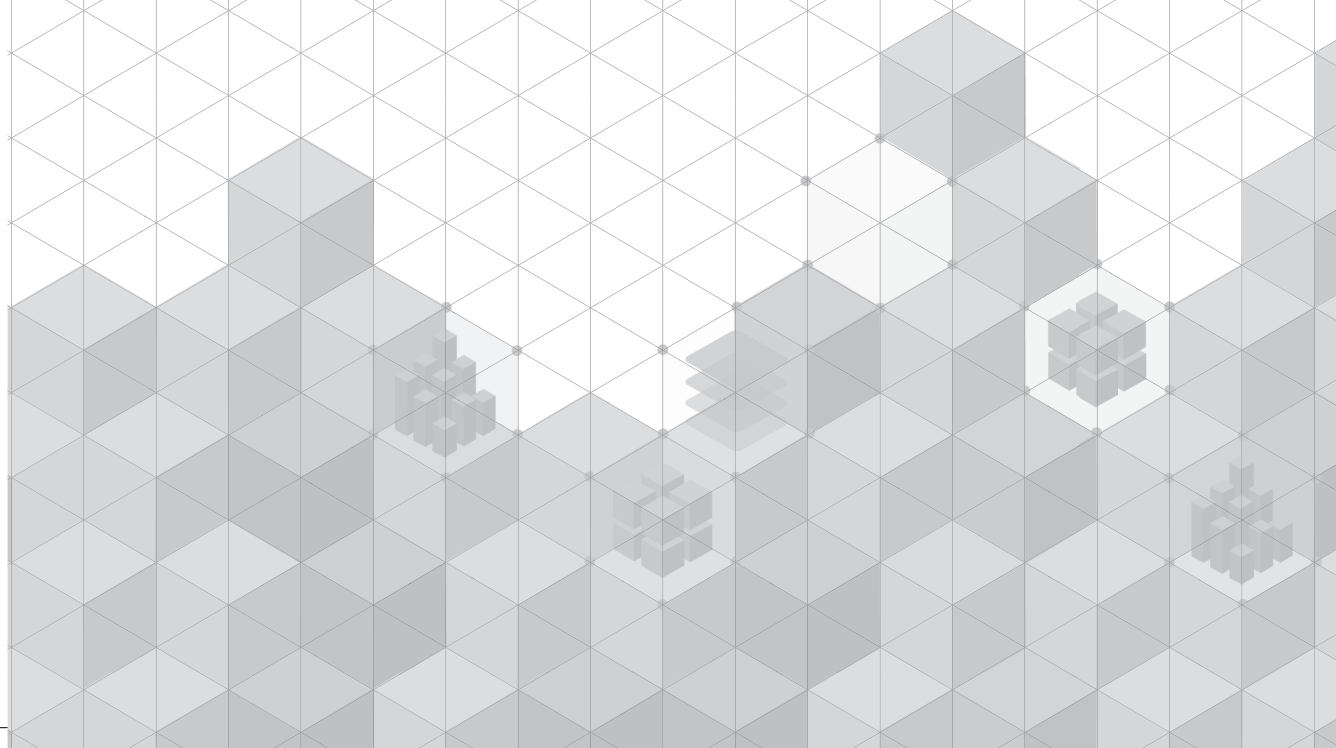


SQL Server® 2012
T-SQL資料庫設計

第07章基礎DAX 資料分析運算式

DAX基本語法原則
Excel基本函數
DAX擴充函數



「To be or not to be, that's the question.」這句莎翁舞台劇的經典台詞，點出了人生在世，總會面對的許多無奈與兩難。這句話套用在商業智慧分析上，也非常貼切。因為，對於高度的商業智慧使用者來說，「寫語法？還是靠介面？」，也是一個十足兩難的抉擇。

透過使用者介面來操作的好處，當然是可以大幅降低學習曲線，但可惜的是，老闆的需求通常不會配合工具來提出，「抱歉老闆，這個介面不支援…」這種話，最好還是只放在心裡。為了解決各種千變萬化的需求，語法仍會是進階分析者，不得不走的必經之路。

既然躲不掉，那麼該用甚麼語言來進行分析呢？第一代的商業智慧（純靠資料庫）中，使用者必須學習資料庫的SQL 查詢語言。SQL語法算是結構清楚且相對學習資源豐富的語言，但它最大的問題在於處理大量資料時的效能。當商業智慧進入多維度分析的時代，雖然效能問題解決了，但多維度分析所使用的MDX（Multi-dimensional Expression），一下子逼著大家立即將習慣三度空間的大腦切換為多維度模式。計算的單位也從表格與欄位，變成抽象的集合與Tuple。雖然筆者個人非常喜歡MDX這個語言，不過，也不得不承認，這的確不是容易學的東西（身為講師，當然最愛這種很難自學的技術）。因此，即使微軟推出MDX已超過10年，且MDX也成為IBM Cognos與Oracle Hyperion的標準語言，但除了少數的應用系統開發商之外，一般使用者鮮少使用此項技術，來進行深度的分析。

「輕量級、低學習門檻」是貫穿全新商業智慧架構的最重要原則。MDX這重口味的技術還是得封印起來，也因此，在這次的商業智慧語意模型（表格式）中，特別推出了全新的DAX（Data Analysis Expressions）語言。

又有新語言！先別擔心，微軟最會的就是推出新的縮寫字，所以，別被新的縮寫給嚇到了！DAX到底是甚麼？簡單說：「DAX是一種Excel公式的延伸語言」，更準確的說法是，DAX其實是一種中繼語言，它最後被編譯執行時，仍舊會轉換成MDX來執行。由於MDX難度過高，所以，微軟讓MDX披上了Excel公式的糖衣，讓使用者比較容易入口（不信的話，您可以試試看，利用MDX查詢BISM表格式模型，會發現仍可回傳正確的結果）。

微軟的思路非常簡單，既然要用語言，那麼全世界最多人會的分析語言是甚麼？不就是Excel的函數與公式嗎？對使用者來說，只要熟悉Excel公式，就可以以此

為基礎來學習DAX，以達到進階商業智慧分析的需求。DAX大致上可分成兩種類型，分別是DAX運算式與DAX陳述句。其中，DAX並無對應於MDX指令碼的原生指令碼，因為DAX的執行，會編譯為MDX指令碼。DAX運算式與Excel中的公式幾乎相同，分析者可利用函數組合成公式，以計算產生新的成員與數值，當然，其為商業智慧也增加了非常多的新函數。至於DAX陳述句，則企圖以類似函數的手法，來處理如同T-SQL般，回傳整個資料表的結果，其功能有點類似SQL Server的表格式值函數，也就是結構上是函數，但卻能回傳一整個資料表。我們將分別在本章與下一章中介紹。

不過，DAX 函數雖以 Excel 函數為基礎，但畢竟也只是「披著羊皮的狼」，與實際上的Excel函數仍有許多不同之處：

- Excel是以「資料格」或「資料範圍」做為計算基礎，而DAX是以「資料行」或「資料表」做為計算基礎，這是兩者差異最大的地方。
- 為了有效發揮時間智慧的商業智慧分析功能，DAX實際處理的是DateTime的資料型別，而Excel的日期資料型別，其實只是整數（自1900/1/1起算的天數）。
- Excel並無關聯式資料庫的概念，因此，Excel中，陣列和向量查閱函數（例如，大家最熟知的VLookup函數）的最大作用就是進行活頁簿間的交互查閱。但在DAX中，雖然提供了新的查閱函數，但基於關聯式資料庫的精神，使用者必須在資料模型設計階段，建立兩個資料表間的關聯性後，方能發揮作用。
- DAX 不支援 Excel 中的 variant 資料型別。且在相同資料行中的資料，只能擁有一種資料型別。

雖然DAX與Excel仍有差異，但就筆者的經驗看來，並不會造成學習上的困難。但有個最大的問題是，通常具備Excel函數能力的人，多為Power User，而非資訊部門。資訊部門的人，一般不太會去學習Excel函數，這也造成資訊人員在未來的商業智慧分析上，相較於Power User，可說是直接輸在起跑線。接下來的兩個章節中，我們將詳細介紹DAX的基本運算及進階的分析查詢。希望能讓資訊單位人員，對DAX能有進一步的認識。

在繼續介紹DAX之前，大家的腦海中，可能還存在一個疑問，難道MDX的時代結束了嗎？基本上，各位不需有此疑慮，因為我們說過了，DAX只是一種中繼語言，它執行編譯後，仍舊還是MDX，所以MDX仍會永遠存在。套句微軟Analysis Services研發團隊的說法：「商業智慧語意模型多維度式」與MDX，就像C++般，是永遠不會消失的。尤其當遇到難以解決的進階商業智慧分析時，還是必須請出來使用。但在大多數分析時，只需使用較輕量級的「商業智慧語意模型表格式」與DAX就可以了。他們就像C#，學習起來比較容易，也能讓商業智慧分析能力擴散在企業的每個角落。

07-01 DAX基本語法原則

7-1-1 物件命名原則

之前曾強調過，DAX是以「資料行」或「資料表」做為計算基礎，而在基本物件命名上，最重要的就是如何指定資料行與資料表。在DAX進行物件宣告很簡單，只要符合以下原則：

- 資料表名稱以單引號包覆，沒有區分大小寫。
- 資料行名稱以中括號包覆（若資料表名稱內沒有空格，就可以忽略單引號），沒有區分大小寫。如果所引用的資料行與加入計算的資料表，位於同一個資料表，那麼資料行名稱便不用特別標示資料表。

命名時，還要注意以下原則：

- 欄位或資料表的名稱開頭與結尾，不可以有空白字元。
- 欄位或者是資料表的名稱中，不能包含以下特殊字元：

.,':\^*|?&%\$!+=()[]{}<>

宣告資料行

[資料行名稱]

'資料表名稱' [資料行名稱]

資料表名稱 [資料行名稱]

宣告資料表

'資料表名稱'

各位可能發現少了「階層」？也沒有「量值」？「層級」？如果會問這些問題，那代表各位的思考方式仍處於「多維度模式」。在BISM表格式模型中，雖可透過介面自訂階層，但只是為了使用者拖拉操作時的方便而已，對於DAX來說，並不實際存在階層的概念，自然就不會提供宣告階層的語法。

各位可能會開始憑到納悶，如果沒有階層，到時候要如何計算跨階層的比例百分比？又該如何依據時間階層計算出去年同期的成長率？別擔心！在後續的章節中，我們會一一列舉實際的語法範例。但別忘了，DAX只是一種中繼語言，所以當遇到困難待解決時，各位還是可以隨時搬出MDX這個救兵來支援。

6-1-3 資料型別

目前DAX支援以下資料型別：

- 整數 (Integer)
- 實數 (Real)
- 貨幣金額 (Currency)
- 時間 (DateTime)
- 布林值 (Boolean)
- 字串 (String)
- 二進位檔 (Blob, binary large object)

若熟悉Excel的使用者，無需太在意資料型別，因為DAX延續了Excel強大的型別辨識與自動轉型的能力，因此，資料型別並不會對實務分析上，造成太大的困擾。以圖07-1來說，我在此做了一個很簡單的實驗，就是產生一個新的衍生欄位，其公式為 $[ListPrice]/3$ 。其中， $[ListPrice]$ 資料行的資料型別是整數。熟悉T-SQL語法的人都知道， $Select\ 7/3$ 得到的結果絕對不會是 $2.333\cdots$ ，而會是2。以資料型別的原則來看，整數除以整數得到的一定還是整數，因此，在執行除法前，一定要將分子或分母的一段先行轉型，才能得到正確的結果。但相同的實驗放在DAX，得到的就是截然不同的結果，使用者不需要刻意進行轉型，就能得到預期的結果。

ProductID	ProductName	ProductDesc	Brand	OperationSystem	ProductThumbnail	ListPrice	CalculatedColumn 1	加入資料行
79B5D08D...	Apple iPhone 4 1...	Apple 推出更...	Apple	OS 4	二進位資料	20800	6933.3333333333	
954FB6A4...	Apple iPhone 4 3...	Apple 推出更...	Apple	OS 4	二進位資料	23000	7666.6666666667	
P0CC80BE...	SAMSUNG GAL...	三星推出首款...	SAMSUNG	Android 4.0	二進位資料	20400	6800	
B63DAAB8...	NOKIA Lumia 900	諾基亞於世界...	NOKIA	Windows Phone 7.5	二進位資料	21000	7000	
35589AC9...	Apple iPhone 4 3...	Apple 推出更...	Apple	OS 4	二進位資料	23300	7766.6666666667	
F5223AC8...	Apple iPhone 4 1...	Apple 推出更...	Apple	OS 4	二進位資料	20400	6800	
70A0559A...	HTC TITAN II	HTC TITAN II...	HTC	Windows Phone 7.5	二進位資料	25100	8366.6666666667	
3C092E41-6...	ASUS PadFone	華碩 ASUS Pad...	ASUS	Android 4.0	二進位資料	17500	5833.3333333333	
25429C21...	Apple iPhone 4S	Apple 蘋果推...	Apple	OS 5	二進位資料	20800	6933.3333333333	
485EA302-F...	MOTO MOTOS...	MOTOROLA 紮...	MOTORO...	Android 2.3	二進位資料	5990	1996.6666666667	
2DC172C4...	HTC Radar 雷達...	宏達電推出旗...	HTC	Windows Phone 7.5	二進位資料	11000	3666.6666666667	
9D7A457E...	Apple iPhone 4 8...	Apple 推出更...	Apple	OS 4	二進位資料	17800	5933.3333333333	
9C607933-0...	HTC One X 極速...	HTC 推出首款...	HTC	Android 4.0	二進位資料	20900	6966.6666666667	
B4D5062F-1...	Acer Allegro	宏碁推出最新...	Acer	Windows Phone 7.5	二進位資料	7200	2400	
D7363398-9...	HTC Rhyme 音...	HTC Rhyme 音...	HTC	Android 2.3	二進位資料	9800	3266.6666666667	
98AB4A7B...	NOKIA Lumia 610	諾基亞鎖定年...	NOKIA	Windows Phone 7.5	二進位資料	7000	2333.3333333333	
4130CFC1...	Apple iPhone 4 8...	Apple 推出更...	Apple	OS 4	二進位資料	17900	5966.6666666667	
C0213DDA...	Apple iPhone 4S	Apple 蘋果推...	Apple	OS 5	二進位資料	26700	8900	
A8F8E506...	LG Optimus L7	LG Optimus L7...	LG	Android 4.0	二進位資料	8900	2966.6666666667	
0D85D6A1...	Apple iPhone 4S	Apple 蘋果推...	Apple	OS 5	二進位資料	24100	8033.3333333333	
9081973A-0...	HTC ChaCha	HTC ChaCha	HTC	Android 2.3	二進位資料	8200	2733.3333333333	
C89D799F...	SAMSUNG GAL...	為滿足年輕族...	SAMSUNG	Android 2.3	二進位資料	7800	2600	
410026CF...	HTC 7 Mozart	HTC 7 Mozart	HTC	Windows Phone 7	二進位資料	9500	3166.6666666667	
3E333226-2...	MOTO RAZR M...	摩托羅拉推出...	MOTORO...	Android 2.3	二進位資料	16400	5466.6666666667	

圖07-1：自動轉型

之前曾提過，DAX與Excel最大的不同處在於多了DateTime的資料型別，而Excel的日期資料型別，其實是整數（自1900/1/1起算的天數），但為了延續使用者在Excel的使用習慣，在DAX上，仍舊可使用數值的加減，來做為日期的加減運算。例如，「昨天」的語法就等於：

=Today()-1

7-1-3 DAX運算子

至於DAX的基本運算子與Excel公式並無太大差異，由於本書讀者有可能是資訊人員或是商務分析人員，在接下來的篇幅中，筆者將特別整理出「DAX小幫手」來提醒兩種不同使用者，學習DAX應該要注意的地方。

	運算子	功能說明	範例
算數	+	加	[銷售價格]+[税金]
	-	減	[銷售價格]-[成本]
	*	乘	[銷售價格]*[銷售數量]
	/	除	([銷售價格]-[成本])/ [銷售價格]
	^	次方	(1+[利率])^ [期數]

	運算子	功能說明	範例
比較	=	等於	[銷售價格]=100
	>	大於	[銷售價格]>100
	>=	大於等於	[銷售價格]<=100
	<	小於	[銷售價格]<100
	<=	小於等於	[銷售價格]<=100
	<>	不等於	[銷售價格]<>100
邏輯	&&	且	[銷售價格]>100&&[成本]<50
		或	[銷售價格]>100 [成本]<50
	!	非	!([銷售價格]>100)
	AND	且	AND ([銷售價格]>100,[成本]<50)
	OR	或	OR ([銷售價格]>100,[成本]<50)
	NOT	非	NOT([銷售價格]>100)
字串連接	&	字串連結	[姓氏]&[名字]



DAX小幫手

如果您是熟悉SQL或MDX的資訊人員

這裡要特別注意的是，邏輯運算子中的「AND」、「OR」的用法，與我們在T-SQL語法或MDX的使用方式不同。在此是沿襲在Excel中的使用模式，將「AND」、「OR」視為一種函數，所以要將要連結的條件置於函數後方的括弧之中。請一般資訊人員特別注意。

如果您是熟悉Excel的商務分析人員

DAX新增了原先Excel中不支援的「&&」、「||」與「!」，這些運算子可幫助我們更直覺地進行條件連結，建議大家多熟悉新運算子的用法。

7-1-4 DAX計算欄位

在傳統的多維度分析中，可在兩個地方進行計算，一是在資料來源檢視中，加入具名計算（使用SQL語法），或者是在建置完Cube後，再透過MDX加入導出成員。前者比較適合用在明細的層級，後者則比較適合用在彙總的層級。

在BISM表格式模型中，也承襲了這樣的觀念，但不需像過往必須熟悉兩種語法來撰寫計算成員，無論是明細或者是彙總，都只要使用DAX就行了。接著，我們就先來介紹撰寫計算欄位的地方。

在設計畫面中（不管是PowerPivot或BISM表格模型專案），可發現所有的資料表都會出現一條分隔線，資料只存在分隔線的上方。事實上，下方的空白區域是讓我們填入新的彙總計算欄位公式用（相當於在多維度分析中，撰寫MDX之功能），我們稱之為「量值」（與多維度分析的量值相同）；至於資料表的右方，則與一般的Excel公式類似，都是用來產生明細的計算欄位，我們稱之為計算欄位（相當於多維度分析在資料來源檢視中，具名計算的功能）。

Brand	OperatingSystem	ProductThumbnail	StartDate	EndDate	ListPrice	CalculatedColumn 1
Apple	OS 4	二進位資料	2010/9/1 上...		20800	41600
Apple	OS 4	二進位資料	2011/4/1 上...		23000	46000
SAMSUNG	Android 4.0	二進位資料			20400	40800
NOKIA	Windows Phone 7.5	二進位資料			21000	42000
Apple	OS 4	二進位資料	2010/9/1 上...		23300	
Apple	OS 4	二進位資料	2011/4/1 上...		20400	
HTC	Windows Phone 7.5	二進位資料			25100	
HTC	Android 2.3	二進位資料	2011/4/1 上...		9100	18200
ASUS	Android 4.0	二進位資料			17500	35000
Apple	OS 5	二進位資料	2011/12/1 ...		20800	41600
MOTORO...	Android 2.3	二進位資料			5990	11980
HTC	Windows Phone 7.5	二進位資料			11000	22000
Apple	OS 4	二進位資料	2010/9/1 上...		17800	35600
HTC	Android 4.0	二進位資料			20900	41800
HTC	Android 2.2	二進位資料	2012/5/1 上...		10600	21200
Acer	Windows Phone 7.5	二進位資料			7200	14400
HTC	Android 2.3	二進位資料			9800	19600
NOKIA	Windows Phone 7.5	二進位資料			7000	14000
Apple	OS 4	二進位資料	2011/4/1 上...		17900	35800
Apple	OS 5	二進位資料	2011/12/1 ...		26700	53400
LG	Android 4.0	二進位資料			8900	17800
Apple	OS 5	二進位資料	2011/12/1 ...		24100	48200
HTC	Android 2.3	二進位資料	2011/7/1 上...		8200	16400
					ListPrice 的總和: 505690	

圖07-2：可加入計算欄位的位置

兩種計算模式不僅位置不同，命名操作的模式也各有不同。首先，資料表右方的「計算欄位」是先撰寫完DAX公式之後，再用滑鼠點選右鍵選取「重新命名資料行」以調整計算欄位的名稱。

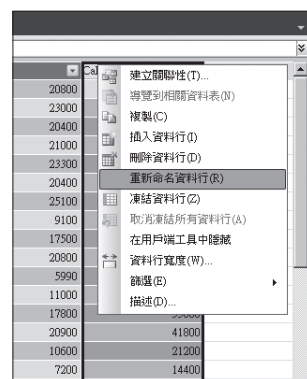


圖07-3 重新命名資料行

至於在分隔線下方的「量值」區域，則是直接在公式中定義量值名稱。其方式為：

量值名稱:=DAX計算公式

若是在這個區域按右鍵，是找不到任何重新命名相關的選項。



The screenshot shows the DAX formula bar for a calculated column named 'ListPrice'. The formula is `=SUM(ListPrice)`. Below the formula bar, a table is visible with columns: 'productThumbnail', 'StartDate', 'EndDate', and 'ListPrice'. The 'ListPrice' column contains values: 20800, 23000, 20400, 21000, 23300, 20400, and 25100. At the bottom of the table, a summary row shows 'ListPrice 的總和: 505690'.

productThumbnail	StartDate	EndDate	ListPrice
二進位資料	2010/9/1 上...		20800
二進位資料	2011/4/1 上...		23000
二進位資料			20400
二進位資料			21000
二進位資料	2010/9/1 上...		23300
二進位資料	2011/4/1 上...		20400
二進位資料			25100
			ListPrice 的總和: 505690

圖07-4：定義量值名稱

07-02 Excel基本函數

既然DAX被筆者形容為「披著Excel函數外皮的MDX」，自然學習DAX的最好起點，就是從原有的Excel共用函數著手了。這對於商務分析人員來說，自然是再熟悉不過了。但要注意的是，由於DAX的本質還是以資料表與資料行為計算基礎，而不是以資料格與資料範圍做為計算基礎，所以Excel函數並無法百分之百相容於DAX，因此，筆者在本節中，將會把所有可沿用的函數進行整理。

而對資訊人員來說，因大多數都不太熟悉Excel函數，等於要重新學起，且因其中有些功能與SQL語法有異同之處，筆者都會會在後續的內容中清楚標明，避免使用上的混淆與誤解。此外，DAX與Excel函數一樣都，是不分大小寫的，傳統Excel函數多半為英文字母縮寫，字節較短，因此，參考文件中大多全部是大寫。但新增加的DAX函數，多半屬多個英文單字的複合字，字節較長，因此筆者採用大駝峰式命名法（Upper Camel Case），將複合字的每個單字首字以大寫標示，其餘採小寫，以方便各位閱讀與記憶。至於到了實際應用時，大小寫就隨個人喜好了。

7-2-1 數學與三角函數

ABS 函數

ABS主要是用來計算數值的絕對值，此函數不論在T-SQL或Excel函數中，使用模式都是完全相同的。

函數介紹：回傳將數值取絕對值的十進位數字
ABS(<數值>)

CEILING 函數

CEILING函數雖然T-SQL中也有，但T-SQL中，CEILING函數僅能處理無條件進位至整數，但在Excel中，其引數多了一個進位基準值，舉例來說，CEILING(5.288,0.5)，回傳值會直接無條件進位至0.5的倍數，也就是回傳值為5.5。此DAX函數的計算模式與Excel完全相同。

函數介紹：回傳進位後的數字，需指定進位基準(significance)。
CEILING(<數值>, <進位基準>)

在此需要注意的是，DAX中共有兩種CEILING函數，一種是沿襲Excel使用模式的CEILING函數，另一種則是新增的ISO.CEILING模式。其實兩者差異並不大，後者的存在，只是為了與ISO定義的計算行為相容而已。當輸入數值為正數時，其實兩種模式的結果是相同的；當輸入數值為負數，且進位基準是正數時，兩種模式的計算也會完全相同。唯一會產生差異的地方在於，當輸入數值為負數，且進位基準是負數時，CEILING的進位方向為負向（向負無限大方向進位），而ISO.CEILING則是採用正向（向正無限大方向進位）。其差異可參考圖07-5的結果。

The screenshot shows a DAX formula bar with the formula `=CEILING(-4.888,-0.05)` and a table with columns Date, ListPrice, CalculatedColumn 1, and CalculatedColumn 2. The table contains data for ListPrice values 20800 and 23000. CalculatedColumn 1 shows the result of the CEILING function, which is -4.9 for both values. CalculatedColumn 2 shows the result of the ISO.CEILING function, which is -4.85 for both values.

Date	ListPrice	CalculatedColumn 1	CalculatedColumn 2
	20800	-4.9	-4.85
	23000	-4.9	-4.85

圖07-5：CEILING與ISO.CEILING的差異

DAX範例：計算進位後的客戶年齡

```
= CEILING ((Now()-[生日])/365,1)
```

EXP 函數

EXP主要用來計算自然對數為底數的乘幂值，此DAX函數的計算模式與Excel完全相同。例如，EXP(2)表示回傳e的2次方，此函數不論在T-SQL或Excel函數中，使用模式均完全相同。如果要計算任意指定底數的乘幂值時，則應改用POWER函數。

函數介紹：回傳以自然對數（e，2.71828182845904）為底數的乘幂值

EXP (<數值>)

FACT 函數

FACT函數主要用來計算階乘使用，此DAX函數的計算模式與Excel完全相同。若引數中的數值非整數，則只會計算至此數字的整數位數為止。在T-SQL中，並無相似的對應函數。

函數介紹：回傳數字的階乘。某數的階乘等於 1*2*3*... * 數字。

FACT (<數值>)

FLOOR 函數

FLOOR函數與CEILING函數相反，是計算無條件捨去後之數值，此DAX函數的計算模式與Excel完全相同。雖然T-SQL中也有，但在T-SQL中，FLOOR函數僅能處理無條件進位至整數，但在Excel中，其引數多了一個進位基準值。舉例來說，FLOOR (5.888,0.5)，回傳值會直接無條件捨去至0.5的倍數，也就是回傳值為5.5。在DAX中，使用方式同Excel。

函數介紹：回傳捨去位數後的數字，需指定進位基準（significance）。

FLOOR (<數值>, <進位基準>)

DAX範例：組成季別的字串（Q1, Q2...）

```
= " Q" &FLOOR (MONTH ([銷售日期])/3+1)
```

INT 函數

INT函數主要回傳最接近、且小於原數值的整數。此DAX函數的計算模式與Excel完全相同。

函數介紹：回傳最接近且小於原數值的整數

INT (<數值>)

在DAX中有多個函數都與取整數有關，在此將這幾個函數做完整比較如下：

- **CEILING**：無條件進位至指定進位基準的倍數。
- **ISO.CEILING**：無條件進位至指定進位基準的倍數。
- **FLOOR**：無條件捨去至指定進位基準的倍數。
- **INT**：最接近且小於原數值的整數。
- **ROUND**：四捨五入至指定位數。
- **TRUNC**：移除小數點後的部分。

為避免文字敘述較難理解，筆者也準備了幾個範例讓各位分辨函數間的差別：

- **CEILING(123.456,1)=124**（無進位捨去至1的倍數）
- **ISO.CEILING(123.456,1)=124**（無進位捨去至1的倍數）
- **FLOOR(123.456,1)=123**（無條件捨去至1的倍數）
- **INT(123.456)=123**（最接近且小於原數值的整數）
- **ROUND(123.456,0)=123**（四捨五入至第0位數）
- **TRUNC(123.456)=123**（移除小數點後的部分）

其中，CEILING、ISO.CEILING與FLOORING的意義在前面篇幅中已經介紹過，就不再贅述，而ROUND表示四捨五入，也是大家所熟悉的操作。比較令人困擾的是，INT與TRUNC的差異究竟為何呢？請看以下的範例：

- **INT(-123.456)=-124**（最接近且小於原數值的整數）
- **TRUNC(-123.456)=123**（移除小數點後的部分）

因為INT回傳值必須小於原數值，因此，會回傳-124，這與直接截去小數後部分的TRUNC，有顯著的不同。

LN 函數

LN函數主要用來計算自然對數為底數的對數值使用，此DAX函數的計算模式與Excel完全相同。在T-SQL語法中，並無相同的LN函數，但此函數的使用方式與功能與T-SQL中的LOG函數完全相同。

函數介紹：回傳以自然對數 (e, 2.71828182845904) 為底數的對數值
LN (<數值>)

LOG 函數

LN函數主要用來計算以指定底數的對數值使用，此DAX函數的計算模式與Excel完全相同。請注意，雖然T-SQL中也有名為LOG的函數，但此函數只有一個引數，與此處的LOG函數用法不同，T-SQL中的LOG函數意義應該對等於DAX中的LN函數。

函數介紹：回傳以指定底數的對數值
LOG (<數值>, <基數>)

LOG10 函數

LN函數主要用來計算以10為底數的對數值使用。此DAX函數的計算模式與Excel及T-SQL完全相同。

函數介紹：回傳以10為底數的對數值
LOG10 (<數值>)

MROUND 函數

MROUND函數主要用於計算將數值四捨五入至指定倍數基數的結果。此DAX函數的計算模式與Excel完全相同

當數字除以倍數的餘數大於或等於倍數基數的一半，則 MROUND 會以正向（往正無限大的方向）方式進位。其中，兩個引數（數值與倍數基數）還必須具有相同的符號，否則會回傳錯誤。

函數介紹：回傳將數值四捨五入至指定倍數基數的結果
MROUND (<數值>, <倍數基數>)

PI 函數

PI主要是用來回傳圓周率，此函數不論在T-SQL或Excel函數中，使用模式都是完全相同的。

函數介紹：回傳圓周率
PI ()

POWER 函數

POWER主要用來計算數值在指定乘幕後的結果，此函數不論在T-SQL或Excel函數，使用模式均完全相同。

函數介紹：回傳數值乘幕的結果
POWER (<數值>, <乘幕>)

DAX範例：計算存款複利
= [存款餘額] * POWER (1 + [年利率], [年數])

QUOTIENT 函數

回傳兩數相除，所得之商的整數部分。此函數可用來除去兩數相除結果的餘數部分。

函數介紹：回傳兩數相除所得之商的整數部分。
QUOTIENT (<分子>, <分母>)

RAND 函數

RAND()函數回傳大於等於 0 且小於 1 的均等分配隨機亂數。每次計算工作表時，都會回傳新的隨機亂數。此DAX函數的計算模式與Excel完全相同。T-SQL中也有RAND函數，不過，T-SQL中的RAND函數支援無引數（與DAX相同使用模式）以及單一引數（輸入隨機種子，用以確保每次執行，都可得到相同亂數）兩種模式略有不同，這是使用時要特別注意的。

函數介紹：回傳大於等於 0 且小於 1 的均等分配亂數
RAND ()

亂數的使用，通常在產生測試用範例資料或是用於資料採礦時，進行隨機抽樣所使用。

RANDBETWEEN 函數

RANDBETWEEN 函數用以回傳指定的數字之間的均等分配隨機亂數。在每次計算工作表時，都會回傳新的隨機亂數。

函數介紹：回傳指定的數字之間的均等分配隨機亂數

RANDBETWEEN (<亂數下界>, <亂數上界>)

RANDBETWEEN是由RAND衍生而來的，RANDBETWEEN(a,b)其實就等於 $b*(RAND()+a/b)$ 。

ROUND 函數

ROUND主要用來計算依指定小數位數四捨五入後的數值，此函數不論在T-SQL或Excel函數中，使用模式都是完全相同的。

函數介紹：回傳依指定小數位數四捨五入後的數值

ROUND (<數值>, <小數位數>)

ROUNDDOWN 函數

ROUNDDOWN主要用來計算依指定小數位數無條件捨去後的數值，此DAX函數的計算模式與Excel完全相同，ROUNDDOWN(數值,1)與FLOOR(數值,0.1)會得到相同的結果。

函數介紹：回傳依指定小數位數無條件捨去法後的數值

ROUNDDOWN (<數值>, <小數位數>)

ROUNDUP 函數

ROUNDUP主要用來計算依指定小數位數無條件進位後的數值，此DAX函數的計算模式與Excel完全相同，ROUNDUP (數值,1)與CEILING(數值,0.1)會得到相同的結果。

函數介紹：回傳依指定小數位數無條件進位法後的數值

ROUNDUP (<數值>, <小數位數>)

SIGN 函數

SIGN主要用來回傳指定數值之正負號。如果指定數值為正值，則回傳1；如果指定數值為負值，則回傳-1；如果指定數值為0，則回傳0。此DAX函數的計算模式與Excel完全相同。

函數介紹：回傳此數值的正負號
SIGN(<數值>)

SQRT 函數

SQRT主要是用來計算數值在取平方根後的結果，此函數不論在T-SQL或Excel函數中的使用模式均完全相同。如果指定數值為負值，將會回傳錯誤。

函數介紹：回傳數值的平方根
SQRT(<數值>)

SUM 函數

SUM函數是將指定欄位進行加總，此函數與T-SQL用法相同，其引數都是指定欄位。其與Excel用法是不一樣的，在Excel中，SUM函數可以是一組數值（例如，SUM(2,3)=5）或是一個範圍（SUM(A1:B5)），但在DAX中，僅能放入欄位。

SUM函數的引數如果指定了非數值欄位，並不會回傳錯誤，僅會回傳空白。若是在計算欄位之處使用了SUM函數，則會回傳整個欄位的加總。如果在量值中使用SUM函數，則會回傳符合維度交錯下的案例加總。當希望針對特定條件進行篩選（類似T-SQL中的SUM(CASE WHEN ... END)句型）時，就必須改用SUMX函數。

The screenshot shows a Power BI interface with a DAX formula bar at the top containing the formula `=SUM(ListPrice)`. Below the formula bar is a table with the following columns: ProductThumbnail, StartDate, EndDate, ListPrice, and CalculatedColumn1. The table contains several rows of data, including product thumbnails, start and end dates, and list prices. The CalculatedColumn1 column shows the result of the SUM function, which is 505690. The table also includes a row for the total sum of ListPrice, which is 505690.

ProductThumbnail	StartDate	EndDate	ListPrice	CalculatedColumn1
二進位資料	2010/9/1 上...		20800	505690
二進位資料	2011/4/1 上...		23000	505690
二進位資料			20400	505690
7.5 二進位資料			21000	505690
二進位資料	2010/9/1 上...		23300	505690
二進位資料	2011/4/1 上...		20400	505690
7.5 二進位資料			25100	505690
二進位資料	2011/4/1 上...		9100	505690
二進位資料			17500	505690
二進位資料	2011/12/1 ...		20800	505690
二進位資料			5990	505690
7.5 二進位資料			11000	505690
二進位資料	2010/9/1 上...		17800	505690
二進位資料			20900	505690
			ListPrice 的總和: 505690	

圖07-6：在計算欄位與量值使用SUM函數

函數介紹：將指定欄位進行加總

SUM (<數值欄位>)

TRUNC 函數

TRUNC函數主要可將指定數值截去小數部分，並回傳其整數部分。此DAX函數的計算模式與Excel完全相同。其中，第二個引數（小數位數）是可省略的，如果省略，則代入預設值為0。TRUNC函數與ROUNDDOWN函數的用法是完全相同的，但與INT函數略有差別。TRUNC函數會回傳整數部分，而INT函數則是回傳最接近的整數。因此，TRUNC(-10.6)會回傳-10，而INT(-10.6)則回傳-11。

函數介紹：取數值的絕對值

TRUNC (<數值>, [小數位數])

7-2-2 文字函數

BLANK 函數

BLANK函數會回傳空白。在DAX中，BLANK函數與Excel的BLANK函數略有不同。原則上，BLANK函數近似於T-SQL語法中的Null或是C#中的DBNull。至於在

DAX與Excel的使用上，差異在於BLANK函數與空字串（“”）的運算行為，主要有以下幾點差異：

- 在某些DAX函數中，會將BLANK函數與空字串（“”）視為相等。我們將在後續的篇幅出現個別函數時，加註說明。但在Excel中，是永遠不會相等的。
- 由於DAX會執行自動隱含轉型，因此，原始資料中的空字串或 Null會被轉型為BLANK()，但在Excel中存取外部資料時，並不會將來源資料中的空字串轉換為BLANK()。

函數介紹：回傳空白
BLANK ()

DAX範例：計算毛利率
=IF(SUM(‘銷售明細’[銷售金額])=0,BLANK(),(SUM(‘銷售明細’[銷售金額])-SUM(‘銷售明細’[銷售成本]))/SUM(‘銷售明細’[銷售金額]))

CONCATENATE 函數

CONCATENATE函數主要會將引數中的字串串接後，回傳。此DAX函數與Excel函數中的使用模式是完全相同的。CONCATENATE(“A”, “B”)的使用方式也與“A” & “B” 相同。

如果CONCATENATE函數指定了非文字之內容，CONCATENATE函數會將其隱含轉型為字串後，再進行串接。

函數介紹：回傳將指定文字串接後的字串
CONCATENATE (<文字1>,<文字2>,...)

EXACT 函數

EXACT 函數會比較兩個文字字串，如果兩者完全相同，則回傳TRUE；反之，則回傳 FALSE。EXACT 會區分大小寫，但忽略格式設定上的差異。使用 EXACT 可測試文件中的文字，此DAX函數與Excel函數中的使用模式是完全相同的。

函數介紹：回傳布林值已表示兩個字串是否完全相同
EXACT (<文字1>,<文字2>)

FIND 函數

FIND 函數的作用在於回傳「欲搜尋文字」在「被搜尋文字」中的開始位置，若有指定「無法搜尋填補值」時（通常是 0、-1 或 BLANK()），或當「被搜尋文字」中找不到「欲搜尋文字」的存在時，則會回傳「無法搜尋填補值」；若無指定，則回傳BLANK()。

其中，「欲搜尋文字」可使用萬用字元，包括：

- 問號(?)：符合任何單一字元。
- 星號(*)：符合任何順序的字元。
- 如果您想要尋找實際的問號或星號，請在字元前面輸入波狀符號(~)。

FIND 函數會區分大小寫，此DAX函數與Excel函數中的使用模式是完全相同的。而函數的作用與T-SQL中的PATINDEX函數有些類似。

函數介紹：回傳「欲搜尋文字」在「被搜尋文字」中的開始位置（整數）。

FIND(<欲搜尋文字>, <被搜尋文字>[, [<開始位數>][, <無法搜尋填補值>]])

FIXED 函數

FIXED 函數會將指定數字四捨五入到指定的小數位數後，再將結果當成文字回傳。您可以指定回傳的結果是否帶有逗號，當無逗點，註記為 1，則不會在回傳的文字中，顯示逗號。如果為 0 或省略，則會在回傳的文字中，顯示逗號。此DAX函數與Excel函數中的使用模式，是完全相同的。

函數介紹：回傳指定數字四捨五入到指定的小數位數後的文字字串

FIXED(<數值>, <小數位數>, <無逗點註記>)

FORMAT 函數

FORMAT 函數會根據指定的格式字串將值轉換成文字後回傳。如果「指定值」是BLANK()，則會回傳空字串。如果「格式字串」是BLANK()，則此值會使用「一般數字」或「一般日期」格式來格式化（根據指定值的原始資料型別）。

函數介紹：回傳根據指定的格式將值轉換後的文字。

FORMAT(<指定值>, <格式字串>)

會應用到FORMAT函數的，包括了「數值」與「日期」資料型別的值，兩者都同時提供了預定義的格式字串以及自訂的格式字串。基本上，定義都與Excel完全相同。

以下是數值的預定義格式字串：

格式字串	說明
"General Number"	顯示不含千分位符號的數字。
"Currency"	顯示含有千分位符號的數字；預設的小數點位數為兩位數，至於幣別符號，則是根據作業系統內的系統地區設定為基礎。
"Fixed"	在小數點的左邊至少顯示一位數，並顯示兩位小數位數。
"Standard"	顯示含有千分位符號的數字；在小數點的左邊至少顯示一位數，並顯示兩位小數位數。
"Percent"	顯示乘以 100，在正右方附加百分比符號（%）的數字，並顯示兩位小數位數。
"Scientific"	使用標準科學記號標記法，提供2個有效位數。
"Yes/No"	如果數字為零，則顯示No；否則，會顯示Yes。
"True/False"	如果數字為零，則顯示False；否則，會顯示True。
"On/Off"	如果數字為零，則顯示Off；否則，會顯示On。

若想要自訂格式字串，以下是數值適用的格式字串規格定義：

格式規格	說明
無	顯示沒有格式的數字。
0 (零字元)	在格式字串中，0表示「空值補零」。如果0出現在小數點的右方，例如，「.00」，則即使該數值僅有一位小數，也要補足為兩位小數。弱勢0出現在小數點左方，也必須利用前置0的方式來補足位數。
#	在格式字串中，#表示「有值時，顯示；無值時，不顯示」。
. (點字元)	表示小數點。
%	表示百分比。

格式規格	說明
, (逗號字元)	表示千分位符號，使用範例如下： <ul style="list-style-type: none"> "#,0." (井號-逗號-零-小數點) 是使用千分位符號，將數值10億格式為字串 "100,000,000"。 "#0,." (井號-零-逗號-小數點) 是使用1000做為進位基準，將數值10億格式為字串 "100000"。 "#,0,." (井號-逗號-零-逗號-小數點) 是使用千分位符號，並以1000做為進位基準，將數值10億格式為字串 "100,000"。
E、E+、e-、e+	表示科學記號。
-\$()	常值字元。這些字元會完全按照輸入，顯示在格式化字串中。若要顯示未列出的字元，請在該字元前面加上一個反斜線 (\) 或以雙引號 ("") 括住該字元。
\ (反斜線字元)	在格式字串中，顯示下一個字元。若要將有特殊義意的字元顯示為常值字元，請在該字元前面加上一個反斜線 (\)。反斜線本身不會顯示。使用反斜線就跟使用雙引號，把反斜線後面的字元括起來是一樣的。若要顯示一個反斜線，請使用兩個反斜線 (\\)。 <p>無法顯示為常值字元的字元範例，包括：日期格式和時間格式字元 (a、c、d、h、m、n、p、q、s、t、w、y、/ 和 :)、數值格式字元 (#、0、%、E、e、逗號和句號)，以及字串格式字元 (@、&、<、> 和 !)。</p>
"ABC"	顯示雙引號 (") 內的字串。若要在程式碼中包含格式引數中的字串，必須使用 Chr(34) 括住該文字 (34 代表雙引號 (") 的字元碼)。

以下是日期與時間的預定義格式字串：

格式規格	說明
"General Date"	顯示日期及/或時間。例如，3/12/2008 11:07:31 AM。日期顯示是由您應用程式目前設定的文化特性值決定。
"Long Date" 或 "Medium Date"	根據目前文化特性的完整日期格式顯示日期。例如，Wednesday, March 12, 2008。
"Short Date"	使用目前文化特性的簡短日期格式顯示日期。例如，3/12/2008。
"Long Time" 或 "Medium Time"	使用目前文化特性的完整時間格式顯示時間；通常包含：時、分和秒。例如，11:07:31 AM。
"Short Time"	使用目前文化特性的簡短時間格式顯示時間。例如，11:07 AM。

若想自訂格式字串，以下列出日期與時間適用的格式字串規格定義：

格式規格	說明
(:)	時間分隔符號，用以分隔小時、分鐘、秒鐘。
(/)	日期分隔符號，用以分隔年、月、日。
(%)	表示後一個字元應做為單一字母格式讀取，而無須理會後面的任何字母。
d	將日期顯示為沒有前置零的數字（例如，1）。如果這是使用者定義數值格式中唯一的字元，請使用%d。
dd	將日期顯示為有前置零的數字（例如，01）。
ddd	將日期顯示為縮寫（例如，Sun）。
dddd	將日期顯示為完整名稱（例如，Sunday）。
M	將月份顯示為沒有前置零的數字（例如，一月是以 1 來表示）。如果這是使用者定義數值格式中唯一的字元，請使用%M。
MM	將月份顯示為有前置零的數字（例如，01/12/01）。
MMM	將月份顯示為縮寫（例如，Jan）。
MMMM	將月份顯示為完整月份名稱（例如，January）。
gg	顯示期間/紀元字串（例如，A.D.）。
h	使用12小時制，將小時顯示為沒有前置零的數字（例如，1:15:15 PM）。如果這是使用者定義數值格式中唯一的字元，請使用%h。
hh	使用12小時制，將小時顯示為有前置零的數字（例如，01:15:15 PM）。
H	使用24小時制，將小時顯示為沒有前置零的數字（例如，1:15:15）。如果這是使用者定義數值格式中唯一的字元，請使用%H。
HH	使用24小時制，將小時顯示為有前置零的數字（例如，01:15:15）。
m	將分鐘顯示為沒有前置零的數字（例如，12:1:15）。如果這是使用者定義數值格式中唯一的字元，請使用%m。
mm	將分鐘顯示為有前置零的數字（例如，12:01:15）。
s	將秒鐘顯示為沒有前置零的數字（例如，12:15:5）。如果這是使用者定義數值格式中唯一的字元，請使用%s。
ss	將秒鐘顯示為有前置零的數字（例如，12:15:05）。
f	顯示秒的小數。例如，ff會顯示百分之一秒的秒數，而ffff則顯示萬分之一秒的秒數。您在使用者定義格式中，最多可使用7個f符號。如果這是使用者定義數值格式中唯一的字元，請使用%f。

07-02

格式規格	說明
t	使用12小時制，並以大寫A顯示正午之前的任何小時，而以大寫P顯示介於正午與11:59 P.M之間的任何小時。如果這是使用者定義數值格式中唯一的字元，請使用%t。
tt	針對使用12小時制的地區設定，以大寫AM顯示正午之前的任何小時，而以大寫PM顯示介於正午與11:59 P.M之間的任何小時。 針對使用24小時制的地區設定，則不顯示任何符號。
y	將年份顯示為沒有前置零的數字（0-9）。如果這是使用者定義數值格式中唯一的字元，請使用%y。
yy	將年份顯示為有前置零的兩位數數值格式。
yyy	將年份顯示為四位數的數值格式。
yyyy	將年份顯示為四位數的數值格式。
z	將時區時差顯示為沒有前置零的數字（例如，-8）。如果這是使用者定義數值格式中唯一的字元，請使用%z。
zz	將時區時差顯示為有前置零的數字（例如，-08）。
zzz	顯示完整的時區時差（例如，-08:00）

LEFT 函數

LEFT 函數會回傳指定字串左邊特定數量的字元，此DAX函數的使用方式與Excel以及T-SQL中的完全相同。

函數介紹：回傳指定字串左邊特定數量的字元
LEFT(<文字>, <字元數量>)

LEN 函數

LEN函數會回傳指定字串的長度，此函數與Excel、T-SQL中的使用方式相同。此函數套用在中文時（任何double-bit文字），僅視為1個字元。

函數介紹：回傳字串的字元數量
LEN(<文字>)

LOWER 函數

LOWER函數會將指定字串轉換為小寫文字後回傳，此函數使用方式與Excel、T-SQL語法相同。

函數介紹：回傳將指定字串轉換為小寫後的結果

LOWER (<文字>)

MID 函數

MID函數會從指定文字中，根據開始位置與擷取長度取出字串。此函數使用方式與Excel相同，並和T-SQL語法中的SUBSTRING函數使用方式相同。

函數介紹：回傳根據起始位數以及擷取長度取回的字串。

MID (<文字>, <開始位數>, <擷取長度>)

REPLACE 函數

REPLACE函數會將原始字串，從開始位數起至指定字元數量為止的字串取代為新字串，此函數使用方式與Excel相同，與T-SQL語法中的REPLACE函數使用方式略有不同。T-SQL語法中的REPLACE函數是直接指定被取代字串片段以及欲取代的字串片段，比較接近DAX和Excel中的SUBSTITUTE函數。以下是兩者使用方式的差別（要將「產品代號」轉換為「產品代碼」）：

■ DAX以及Excel：REPLACE("產品代號",3,1,"碼")

■ T-SQL：REPLACE("產品代號","號","碼")

函數介紹：回傳將原始字串，從開始位數起至指定字元數量為止的字串，取代為新字串後的結果

REPLACE (<原始文字>, <開始位數>, <字元數量>, <新文字>)

REPT 函數

REPT 函數的作用在於可依指定的次數重複顯示文字，例如，REPT("HI",2)就會回傳"HIHI"。此DAX函數的使用方法與Excel完全相同，在T-SQL中，並無對應功能之函數。

函數介紹：回傳依指定的次數重複顯示文字

REPT (<文字>, <重複顯示次數>)

RIGHT 函數

RIGHT 函數會回傳指定字串右邊特定數量的字元，此DAX函數的使用方式與Excel以及T-SQL完全相同。

函數介紹：回傳指定字串右邊特定數量的字元
RIGHT(<文字>, <字元數量>)

SEARCH 函數

SEARCH 函數回傳被搜尋文字，中第一次找到符合欲搜尋文字的字元位數。在「欲搜尋文字」中，可以使用萬用字元：問號(?)和星號(*)。問號表示符合任何單一字元，星號則符合任何不限定數量之字元。如果您想要尋找就是實際的問號或星號符號，請在字元前面輸入波狀符號(~)。

此函數有兩個選擇性引數，利用「開始位數」可指定開始搜尋的位置，若是忽略，則視為1；「未找到之填補值」表示當被搜尋文字中，找不到欲搜尋文字時，替代的填補值，預設值為0。

此函數使用方式與Excel相同，與T-SQL語法中的PATINDEX函數使用方式類似（PATINDEX中沒有「開始位數」與「未找到之填補值」兩個選擇性引數）。

函數介紹：回傳在被搜尋文字中，第一次找到符合欲搜尋文字的字元位數
SEARCH(<欲搜尋文字>, <被搜尋文字>[, [<開始位數>][, <未找到之填補值>]])

SUBSTITUTE 函數

回傳將指定文字中的「原始文字」片段取代為「新文字」後的結果，選擇性引數「取代次數」可以用來指定這種取代模式執行的上限。此函數使用方式與Excel相同，與T-SQL語法中的REPLACE函數使用方式類似（無「取代次數」的選擇性引數）。

函數介紹：回傳將指定文字中的「原始文字」片段取代為「新文字」後的結果
SUBSTITUTE(<文字>, <原始文字>, <新文字>, [<取代次數>])

TRIM 函數

TRIM函數回傳移除文字中的所有空格後的結果，但是字與字之間的單一空格除外。TRIM 函數原本的設計目的是要修剪文字中的ASCII 空格字元（值32）。在Unicode 字元集中，有一個額外的空白字元，它的十進位值為160。這個字元經常會在網頁中，以 的方式顯示，TRIM函數本身並不會移除這種空白字元。此函數使用方式與Excel相同，與T-SQL語法中，同時使用LTRIM函數與RTRIM函數的效果相同。

函數介紹：回傳移除文字中的所有空格後的結果

TRIM(<文字>)

UPPER 函數

UPPER 函數會將指定字串轉換為小寫文字後回傳，此函數只有方式與Excel、T-SQL語法相同。

函數介紹：回傳將指定字串轉換為大寫後的結果

UPPER (<文字>)

VALUE 函數

VALUE函數會將代表數字的文字字串轉換成數字。此函數只是單純為了與Excel函數相容而設計，在實務上，DAX並不需要使用此函數，因為DAX本身就已經具備隱含轉型的功能。

函數介紹：回傳將代表數字的文字字串轉換成數字後的結果。

VALUE (<文字>)

7-2-3 日期與時間函數

DATE 函數

DATE函數會根據指定的年、月、日組成對應的日期值，其各引數的計算邏輯如下：

引數	定義
year	<p>代表年的數字。</p> <ul style="list-style-type: none"> year引數值可包含一至四位數。year引數會根據您電腦所使用的日期設定，進行解釋。 支援從1990年3月1日開始的日期。 如果您輸入的數字有小數位數，則會四捨五入該數字。 如果是大於9999或小於零（負數值）的值，此函數會回傳#VALUE!錯誤。 如果year值介於0與1899之間，此值會加入到1900來產生最後的值，所以使用12並非回傳2012，而是回傳1912。
month	<p>表示月份的數字。</p> <ul style="list-style-type: none"> 如果月份是從1到12的一個數字，則它會表示該年度的某個月份。 如果輸入大於12的整數，超出12的部分會進位至年度。例如，DATE(2008, 18, 1)，此函數會回傳相當於2009年6月1日的日期時間值。 如果輸入負整數，則會扣除對應日期。例如，DATE(2008, -6, 15)，此函數會回傳相當於2007年6月15日的日期時間值。
day	<p>表示日期的數字。</p> <ul style="list-style-type: none"> 如果day是從1到給定月份之最後一天的數字，則它會表示該月份的某個日。 如果輸入的整數大於給定月份的最後一天，超出的部分會進位至月份。例如，DATE(2008, 3, 32) 公式中，DATE 函數會回傳相當於2008年4月1日的日期時間值。 如果輸入負整數，則會扣除對應日期。例如，DATE(2008, 5, -15) 公式中，DATE函數會回傳相當於2008年4月15日的日期時間值。 如果day包含小數部分，則會將它四捨五入到最近的整數值。

函數介紹：回傳根據指定的年、月、日組成對應的日期值

DATE(<year>, <month>, <day>)

DATEVALUE 函數

DATEVALUE函數會將代表日期的文字字串轉換成日期。此函數只是單純為了與Excel函數相容而設計，在實務上，DAX並不需要使用此函數，這是因為DAX本身就已經具備隱含轉型的功能。此外，由於各作業系統的日期設定不同，此函數並無法保證會得到相同的結果。舉例來說，美國的作業系統是以「月/日/年」的方式來定義，字串「6/4/2012」在美國的電腦，可以被正確解析為2012年6月4號，在台灣的電腦，則會解析失敗，所以筆者還是比較建議使用DATE函數。

函數介紹：回傳將代表日期的文字字串轉換成日期結果

DATEVALUE (日期字串)

DAY 函數

DAY函數會回傳指定日期是該月的第幾天，其範圍為1~31間的數字。此DAX函數的使用方式與Excel以及T-SQL中完全相同。

函數介紹：回傳指定日期是該月的第幾天

VALUE (<文字>)

DAX範例：計算月初日

= '銷售明細' [交易日]+1-DAY('銷售明細' [交易日])

EDATE 函數

EDATE函數回傳指定日期之前、或之後所指定月份數的日期。此DAX函數的使用方式與Excel完全相同。如果指定月份數並非整數，會截斷取整數部分（不是四捨五入）。如果要求的日期超過對應月份的最後一天，則會回傳該月份的最後一天。例如，EDATE(DATE(2012,1,31),1)，雖然2012年2月只有29天（閏月），但還是回傳2012年2月29日，這與T-SQL中，DATEADD(mm,1,'2012/1/31')的計算結果相同。

函數介紹：回傳指定日期之前或之後所指定月份數的日期

EDATE (<日期>, <月份數>)

EOMONTH 函數

EOMONTH函數回傳指定之月數之前或之後當月最後一天的日期。此DAX函數的使用方式與Excel完全相同。如果指定月份數不是整數，該數字會無條件進位或無條件捨去為最接近的整數。例如，EOMONTH(DATE(2012,6,4),3)的回傳值會是2012年9月30日。

函數介紹：回傳指定之月數之前或之後，當月最後一天的日期。

EOMONTH (<日期>, <月份數>)

HOUR 函數

HOUR函數會回傳指定時間是本日的第幾個小時，其範圍為0~23間的數字。此DAX函數的使用方式與Excel完全相同。T-SQL中並無對應函數，其效果與DATEPART(HH,<時間>)相同。

函數介紹：回傳指定時間是本日的第幾個小時
HOUR (<時間>)

MINUTE 函數

MINUTE函數會回傳指定時間是這個小時的第幾分鐘，其範圍為0~59間的數字。此DAX函數的使用方式與Excel中完全相同。T-SQL中，並無對應函數，其效果與DATEPART(MI,<時間>)相同。

函數介紹：回傳指定時間是這個小時的第幾分鐘
MINUTE (<時間>)

MONTH 函數

MONTH函數會回傳指定日期是該年度的第幾個月，其範圍為1~12間的數字。此DAX函數的使用方式與Excel以及T-SQL中，完全相同。

函數介紹：回傳指定日期是該年度的第幾個月
MONTH (<日期>)

NOW 函數

NOW 函數會回傳目前時間，此DAX函數與Excel中的使用方式完全相同，與T-SQL中的GetDate()相同。

函數介紹：回傳目前時間
NOW ()

SECOND 函數

SECOND函數會回傳指定日期是該分鐘的第幾秒，其範圍為0~59間的數字。此DAX函數的使用方式與Excel中完全相同。T-SQL中並無對應函數，其效果與DATEPART(SS,<時間>)相同。

函數介紹：回傳指定日期是該分鐘的第幾秒
SECOND (<時間>)

TIME 函數

DATE函數會根據指定的小時、分鐘、秒組成對應的日期值，其各引數的計算邏輯如下：

詞彙	定義
hour	代表小時的從0到23的數字。大於23的任何值都會被除以24後，取餘數。
minute	代表分鐘的從0到59的數字。大於59的任何值都會進位至小時。
second	代表秒鐘的從0到59的數字。大於59的任何值都會進位至分鐘。

函數介紹：回傳根據指定的小時、分鐘、秒組成對應的日期值

TIME (<小時>, <分鐘>, <秒>)

TIMEVALUE 函數

TIMEVALUE函數會將代表時間的文字字串轉換成時間。此函數只是單純為了與Excel函數相容而設計，在實務上，DAX並不需要使用此函數，因為DAX本身就已經具備隱含轉型的功能。

函數介紹：回傳將代表時間的文字字串轉換成時間的值

TIMEVALUE (<時間文字>)

TODAY 函數

TODAY 函數會回傳今日之日期，其時間會指向12:00 AM，此DAX函數與Excel中的使用方式完全相同。

函數介紹：回傳格式為日期的今天部分

TODAY ()

DAX範例：計算本月初日

=TODAY ()+1-DAY (TODAY ())

WEEKDAY 函數

WEEKDAY函數會回傳指定日期是星期幾。此DAX函數的使用方式與Excel中完全相同。T-SQL中，並無對應函數。其效果與DATEPART(W,<時間>)相同。此函數可以透過回傳類型，來得到不同結果：

- 1：一星期從星期日 (1)開始，到星期六 (7)結束，此為預設值。
- 2：一星期從星期一 (1)開始，到星期日 (7)結束。
- 3：一星期從星期一 (0)開始，到星期日 (6)結束。

函數介紹：回傳

WEEKDAY (<日期>, <回傳類型>)

WEEKNUM 函數

WEEKNUM函數會回傳指定日期是本年度的第幾週。此DAX函數的使用方式與Excel中，完全相同。T-SQL中並無對應函數，其效果與DATEPART(WW,<時間>)相同。此函數可以透過回傳類型來得到不同結果：

- 1：一星期從星期天開始，此為預設值。
- 2：一星期從星期一開始。

函數介紹：回傳指定日期是本年度的第幾週

WEEKNUM (<日期>, <回傳類型>)

YEAR 函數

YEAR函數會回傳指定日期的年度。此DAX函數的使用方式與Excel以及T-SQL中，完全相同。

函數介紹：回傳指定日期的年度

YEAR (<日期>)

YEARFRAC 函數

計算以兩個日期之間，整天天數表示的年份數。這個函數通常會用在計算與年利率相關的值，此DAX函數的使用方式與Excel中，完全相同。

- 0：US (NASD) 30/360，此為預設值。
- 1：實際天數/實際年度天數（一般年度365，閏年366）
- 2：實際天數/360
- 3：實際天數/365
- 4：European 30/360

函數介紹：回傳以兩個日期之間，整日天數表示的年份數
YEARFRAC (<開始日期>, <結束日期>, <計算基準>)

7-2-4 統計函數

統計函數主要是用來計算彙總資訊，大多會用來計算「量值」。雖然統計函數也可以用在「計算欄位」，但整個欄位回傳的都會是相同的彙總值，通常用來做為計算比例時的分母。



DAX小幫手

如果您是熟悉SQL或MDX的資訊人員

這裡要特別注意的是，雖然有些函數如：AVERAGE、COUNT、MAX、MIN看起來都是T-SQL中常見的函數，但是在T-SQL中，這些函數可以同時處理數值、字串、日期、布林值…等多種資料型別，但在DAX或Excel中，多半只能處理數值欄位（COUNT函數多了可處理日期資料行）。

如果您是熟悉Excel的商務分析人員

這一系列的函數雖然用法與Excel相似，但在Excel中，可以指定資料格或是資料範圍；若在DAX中，只能指定欄位（大多是數值欄位），所以還是略有差異的。

AVERAGE 函數

AVERAGE函數會用來回傳資料行中，所有數字的平均（算術平均）。計算時，如果遇到0，則會納入平均值計算；如果遇到空值，則排除於計算中。

函數介紹：計算資料行中所有數字的平均（算術平均）。
AVERAGE (<數值欄位>)

DAX範例：計算本月初日
平均銷售金額:=AVERAGE([銷售金額])

Custid	TrxDate	ProductId	銷售數...	銷售金額	產品成本
701003051	2010/9/1 上午 12...	79B5D08D-4...	1	24500	10500
701002551	2010/9/1 上午 12...	79B5D08D-4...	1	24500	10500
502003725	2010/9/1 上午 12...	35589AC9-B...	1	27000	12000
404000068	2010/9/1 上午 12...	35589AC9-B...	1	27000	12000
404006577	2010/9/1 上午 12...	35589AC9-B...	1	27000	12000
107000037	2010/9/1 上午 12...	35589AC9-B...	1	27000	12000
403005443	2010/9/1 上午 12...	79B5D08D-4...	1	24500	10500
				平均交易金額: 216277.236693092	

圖07-7：計算桌總量值

AVERAGEA 函數

AVERAGE函數會用來回傳資料行中，所有數字的平均（算術平均），包含非數值與文字資料。其處理邏輯如下：

- 布林值的TRUE會計數為1。
- 布林值的FALSE會計數為0。
- 包含非數值之文字的值會計數為0。
- 空字串（""）會計數為0。

函數介紹：計算資料行中包含數值、非數值與文字資料的平均（算術平均）。

AVERAGEA (<欄位>)

COUNT 函數

COUNT函數計算資料行中包含數字的資料列數目，僅能夠計算數值與日期資料行。需要注意的是，雖然T-SQL也有COUNT函數，但在T-SQL中，COUNT函數可用來計算各種非空值的欄位，與此處只能計算數值與日期的DAX COUNT函數不同，比較接近DAX中的COUNTA函數或是COUNTROWS函數。

函數介紹：計算資料行中包含數字的資料列數目。

COUNT (<欄位>)

COUNTA 函數

COUNTA 函數會計算資料行中非空白的資料列數目。它不只計算包含數值的資料列，也會計算包含非空白值的資料列，包括文字、日期和邏輯值。

函數介紹：回傳
COUNTA (<欄位>)

COUNTBLANK 函數

COUNTBLANK函數計算資料行中的空白資料格（BLANK()）數目。若是找不到空白，則會回傳空白。

函數介紹：計算資料行中的空白資料格數目。
COUNTBLANK (<欄位>)

MAX 函數

MAX函數會回傳資料行中最大的數值，僅能夠計算數值欄位。需要注意的是，雖然T-SQL裡面也有MAX函數，但在T-SQL中，MAX函數可計算各種非空值的欄位，與此處只能計算數值的DAX MAX函數不同。

函數介紹：回傳資料行中最大的數值。
MAX (<數值欄位>)

MAXA 函數

MAXA函數會回傳資料行中最大的值（包括數值、日期與布林值）。需要注意的是，MAXA函數並無計算文字之最大值功能，與T-SQL中的MAX不同。

函數介紹：回傳資料行中最大的值 (包括數值、日期與布林值)。
MAXA (<欄位>)

MIN 函數

MIN函數會回傳資料行中最小的數值，僅能夠計算數值欄位。需要注意的是，雖然T-SQL裡也有MIN函數，但在T-SQL中，MIN函數可用來計算各種非空值的欄位，與此處只能計算數值的DAX MIN函數不同，比較接近DAX中的MINA函數。

函數介紹：回傳資料行中最小的數值。
MIN (<欄位>)

MINA 函數

MINA函數會回傳資料行中最小的值（包括數值、日期與布林值）。需要注意的是，MINA函數並無計算文字之最大值功能，仍與T-SQL中的MIN不同。

函數介紹：回傳資料行中最小的值（包括數值、日期與布林值）。

MINA (<欄位>)

7-2-5 邏輯函數

AND 函數

檢查兩個引數中的邏輯條件是否皆為TRUE。如果是，則回傳TRUE；否則，便回傳FALSE。DAX與Excel中的AND 函數只接受兩個引數，若需針對多個運算式執行AND運算，還是比較建議改用「&&」運算子。

函數介紹：判斷兩個邏輯是否同時成立（如果是，則回傳TRUE；否則，則回傳FALSE）

AND (<邏輯條件1>,<邏輯條件2>)

DAX範例：判斷是否為新商品（上架日為最近三個月，且仍然有效）

```
=IF (AND (
    '產品別' [上架日]>EDATE (TODAY (), -3) ,
    '產品別' [有效註記]=TRUE ()
),
    TRUE (),
    FALSE ()
)
```

```
=IF (AND (
    '產品別' [上架日]>EDATE (TODAY (), -3) ,
    '產品別' [有效註記]=TRUE
),
    TRUE, FALSE
)
```

FALSE 函數

FALSE函數回傳邏輯值FALSE。需要注意的是，DAX中，由於內建了隱含轉型的功能，因此不論是寫FALSE()或是FALSE，都會回傳邏輯值FALSE。不過，前者不需多進行隱含轉型的步驟，效能較快。筆者強烈建議各位寫成FALSE()。

函數介紹：回傳邏輯值FALSE。

FALSE ()

IF 函數

檢查是否符合提供為第一個引數的條件。如果條件為 TRUE，回傳一個值；如果條件為 FALSE，則回傳另一個值。

函數介紹：回傳

IF(<條件判斷>,<成立時值>,<不成立時值>)



DAX小幫手

如果您是熟悉SQL或MDX的資訊人員

在T-SQL中，並沒有對應的IF函數，雖然CASE WHEN...END句型有點類似，但IF函數並無法指定多個不同條件的回傳值，只能判斷成立／不成立的回傳值。至於MDX中，也有類似的函數IIF，其用法是相近的。

如果您是熟悉Excel的商務分析人員

永遠別忘了DAX與Excel最大的差異在於，不能單獨指定資料格與隱含轉型。其中，隱含轉型的功能會造成與Excel上不同的計算行為，例如。IF(<條件式>,TRUE,0)在Excel中。因為每個資料格都可具有不同的資料型別，算出來的資料行中，每個資料格會根據條件判斷而產生不同的資料型別，但DAX在同一個資料行中，必須為相同的資料型別，因此，會隱含轉型為等同於IF(<條件式>,1,0)的結果。

IFERROR 函數

如果評估運算式時發生錯誤，會回傳指定的值；否則，會回傳運算式本身的值。例如，以零或空值的錯誤，負數開根號、負數開LOG…等，都是常使用的錯誤情境。

函數介紹：如果評估運算式時發生錯誤，則會回傳指定的值；否則，會回傳運算式本身的值

IFERROR(<量值運算式>,<運算錯誤時填補值>)

DAX範例：除以零錯誤（計算毛利率）

```
=IFERROR( '銷售明細' [總銷售毛利] / '銷售明細' [總銷售金額], BLANK() )
```

```
=IF(
    '銷售明細' [總銷售金額]=0,
    BLANK(),
    '銷售明細' [總銷售毛利] / '銷售明細' [總銷售金額]
)
```


NOT 函數

NOT函數會將邏輯值反轉，通常用於條件篩選時。

函數介紹：回傳將FALSE變為TRUE，或將TRUE變為FALSE
NOT (<邏輯條件>)

OR 函數

檢查兩個引數中的邏輯條件是否其中之一的TRUE。如果是，則回傳TRUE；否則，便回傳FALSE（兩者皆為FALSE）。

函數介紹：判斷兩個邏輯其中之一是否成立（如果是，則回傳TRUE；否則，則回傳FALSE）
OR (<邏輯條件1>, <邏輯條件2>)

TRUE 函數

TRUE函數回傳邏輯值TRUE。需要注意的是，DAX中已內建了隱含轉型的功能，因此不論是TRUE ()或TRUE，都會回傳邏輯值TRUE。但前者不需多進行隱含轉型的步驟，效能較快。筆者強烈建議各位寫成TRUE ()。

函數介紹：回傳邏輯值TRUE。
TRUE ()

7-2-7 資訊函數

資訊函數會查看引數，然後回傳該值是否符合預期的類型。

ISBLANK 函數

ISBLANK函數檢查某個值是否為空白，並根據檢查結果回傳TRUE或FALSE。

函數介紹：檢查某個值是否為空白，並回傳TRUE或FALSE。
ISBLANK (<值>)

DAX範例：除以零與空值錯誤（計算毛利率）
=IF (
OR (
'銷售明細' [總銷售金額]=0,
ISBLANK ('銷售明細' [總銷售金額])=TRUE ()
),
BLANK (),
'銷售明細' [總銷售毛利] / '銷售明細' [總銷售金額]
)

ISERROR 函數

ISERROR函數會檢查某個值是否為錯誤，並根據檢查結果回傳TRUE或FALSE。

函數介紹：檢查某個值是否為錯誤，並回傳TRUE或FALSE。

ISERROR(<值>)

ISLOGICAL 函數

ISLOGICAL函數檢查某個值是否為布林值（TRUE或FALSE），並根據檢查結果回傳TRUE或FALSE。

函數介紹：檢查某個值是否為布林值，並回傳 TRUE 或 FALSE。

ISLOGICAL(<值>)

ISNONTEXT 函數

ISNONTEXT函數會檢查某個值是否「不是」文字（包括空白資料格或非文字），並根據檢查結果，回傳TRUE或FALSE。

函數介紹：檢查某個值是否不是文字（空白資料格或非文字），並回傳TRUE或FALSE。

ISNONTEXT(<值>)

ISNUMBER 函數

ISNUMBER函數檢查某個值是否為數字，並根據檢查結果回傳TRUE或FALSE，此函數可協助判斷欄位中，是否因包含非數值資料，造成DAX無法有效地進行隱含轉型。

函數介紹：檢查某個值是否為數字，並回傳TRUE或FALSE。

ISNUMBER(<值>)

ISTEXT 函數

ISTEXT函數會檢查某個值是否「是」文字，並根據檢查結果，回傳TRUE或FALSE。

函數介紹：檢查某個值是否為文字，並回傳TRUE或FALSE。

ISTEXT(<值>)

07-03 DAX擴充函數

7-3-1 數學與三角函數

CURRENCY 函數

CURRENCY 函數會將引數值轉換為貨幣資料型別，貨幣資料型別最多僅包含4位小數，貨幣符號則會依照作業系統的地區與時間設定。

函數介紹：回傳
CURRENCY(<值>)

ISO.CEILING 函數

ISO.CEILING函數是與ISO定義之計算行為相容的新增函數，詳細說明請參考本章介紹CEILING之部分。

函數介紹：回傳進位後的數字，需指定進位基準（significance）。
ISO.CEILING(<數值>, <significance>)

SUMX 函數

SUMX函數會根據指定資料表（或其他函數回傳之資料表），進行指定運算式之加總。經常運用在條件式加總的情境下。

函數介紹：計算指定資料表中，運算式評估值的加總
SUMX(<資料表>, <量值運算式>)

DAX範例：銷售金額在10000元以上的銷售金額加總
總高額銷售金額:=SUMX(
 FILTER('銷售明細' , '銷售明細' [銷售金額]>10000) ,
 '銷售明細' [銷售金額]
)

Modelbim* x

[銷售金額] =SUMX(FILTER(銷售明細,銷售明細[銷售金額]>10000),銷售明細[銷售金額])

Custo...	TrxDate	ProductId	銷售數...	銷售金額	產品成本	StoreId	Payment
701003051	2010/9/1 上午 12:...	79B5D08D-4...	1	24500	10500	701	
701002551	2010/9/1 上午 12:...	79B5D08D-4...	1	24500	10500	701	
502003725	2010/9/1 上午 12:...	35589AC9-B...	1	27000	12000	502	
404000068	2010/9/1 上午 12:...	35589AC9-B...	1	27000	12000	404	
404006577	2010/9/1 上午 12:...	35589AC9-B...	1	27000	12000	404	
107000037	2010/9/1 上午 12:...	35589AC9-B...	1	27000	12000	107	
403005443	2010/9/1 上午 12:...	79B5D08D-4...	1	24500	10500	403	
				總高銷售金額: 1909728000			

圖07-8：條件式加總

7-3-2 文字函數

DAX函數中並無新增的文字函數。

7-3-3 日期與時間函數

以下的日期與時間函數皆為DAX新增函數，其使用方式需特別注意引數部分所指的「日期資料行」，並非資料表中任何日期資料行皆可適用，「日期資料行」必須是位於被標記為日期資料表的時間維度表上，才能夠套用以下的日期與時間函數，否則將會回傳語法錯誤。

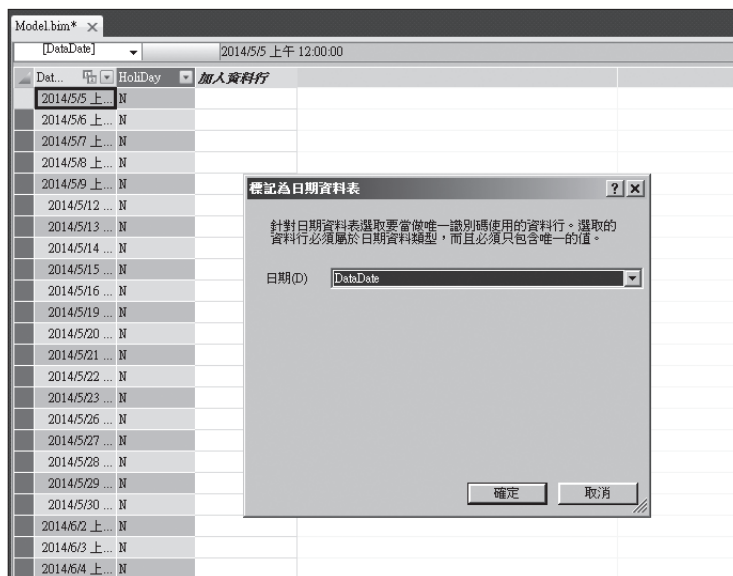


圖07-9：標記為日期資料表

ClosingBalanceMonth 函數

ClosingBalanceMonth 函數的作用就是計算「月底值」。它屬於半加成函數的一種，會根據所傳入日期資料行的日期值，回傳該月月底日的量值數值。

ClosingBalanceMonth 函數廣泛地使用於餘額計算、庫存計算等應用。

函數介紹：根據日期資料行，回傳該月最後一天的量值數值
ClosingBalanceMonth (<量值運算式>, <日期資料行> [, <篩選條件>])

ClosingBalanceMonth 函數其實是將 MDX 中的 ClosingPeriod 函數應用進行簡化，在 MDX 中，ClosingPeriod 函數必須指定時間維度成員與層級，回傳值是對應的時間維度成員。若要做到同樣的效果，必須於 ClosingPeriod 函數中，指定時間維度成員值（相當於 DAX 的日期資料行）以及日期階層的月層級（這部分 DAX 直接透過日期解析的邏輯，自動完成了）。最後，再透過此回傳時間成員與量值的 Tuple，回傳月底的量值。相較之下，DAX 只需利用 ClosingBalanceMonth 函數，就能夠輕鬆做到月底值的計算，是不是簡便許多呢？

DAX 範例：計算月底庫存成本
 = ClosingBalanceMonth (SUMX ('庫存明細' , '庫存明細' [庫存單位成本] * '庫存明細' [庫存單位數]), '日期別' [資料日])

ClosingBalanceQuarter 函數

ClosingBalanceQuarter 函數的作用就是計算「季底值」。它屬於半加成函數的一種，會根據所傳入日期資料行的日期值，回傳該日期所屬季底日的量值數值。

ClosingBalanceQuarter 函數廣泛使用於餘額計算、庫存計算等應用。

函數介紹：根據日期資料行，回傳該季最後一天的量值數值
ClosingBalanceQuarter (<量值運算式>, <日期資料行> [, <篩選條件>])

ClosingBalanceYear 函數

ClosingBalanceYear 函數的作用就是計算「年底值」。它屬於半加成函數的一種，會根據所傳入日期資料行的日期值，回傳該日期所屬年底日的量值數值。

ClosingBalanceYear 函數廣泛應用於餘額計算、庫存計算等。

函數介紹：根據日期資料行，回傳該年度最後一天的量值數值
ClosingBalanceYear (<量值運算式>, <日期資料行> [, <篩選條件>])

DateAdd 函數

DateAdd函數回傳的資料型別是包含日期資料行的「資料表」，而這些日期會根據指定的間隔數有所不同。如果間隔數是正數，則會從目前內容中的日期前移；若是負數，則向後移。

時間間隔列舉值可以是下列值：

- year
- quarter
- month
- day

函數介紹：回傳包含日期資料行的資料表，這些日期會根據指定的間隔數，從目前內容中的日期前移或後移時間

DATEADD (<日期資料行>, <間隔數>, <時間間隔列舉值>)



DAX小幫手

如果您是熟悉SQL或MDX的資訊人員

在T-SQL中，雖然也有DATEADD函數，但除了引數的排列順序不同外，時間間隔列舉值的內容也不同，使用時，需要特別注意。

如果您是熟悉Excel的商務分析人員

當時時間隔列舉值等於month時，DATEADD函數與Excel既有函數EDATE為類似的。

DatesBetween 函數

DatesBetween函數的回傳值也是「資料表」。其中，包含了日期資料行介於開始日期與結束日期間的資料行。

函數介紹：回傳包含了日期資料行介於開始日期與結束日期間的日期資料表

DATESBETWEEN (<日期資料行>, <開始日期>, <結束日期>)

DAX範例：計算今年春季銷售量

```
今年春季銷售量:=CALCULATE(SUM('銷售明細'[銷售金額]), DATESBETWEEN('銷售日別'[資料日],  
DATE(2012,1,1),  
DATE(2012,3,31)  
) )
```

Model.bim* x									
[銷售金額]									
=CALCULATE(SUM(銷售明細[銷售金額]), DATESBETWEEN(銷售日別[資料日],DATE(2012,1,1),DATE(2012,3,31)))									
Custo...	IdDate	ProductId	銷售數	銷售金額	產品成本	StoreId	PaymentMethodCode	折扣金額	
701003051	2010/9/1 上午 12:	79B5D08D-4...	1		24500	10500	701	1	
701002551	2010/9/1 上午 12:	79B5D08D-4...	1		24500	10500	701	1	
502003725	2010/9/1 上午 12:	35589AC9-B...	1		27000	12000	502	1	
404000668	2010/9/1 上午 12:	35589AC9-B...	1		27000	12000	404	1	
404006577	2010/9/1 上午 12:	35589AC9-B...	1		27000	12000	404	1	
107000037	2010/9/1 上午 12:	35589AC9-B...	1		27000	12000	107	4	
403005443	2010/9/1 上午 12:	79B5D08D-4...	1		24500	10500	403	2	
				今年春季銷售量: 224555000					

圖07-10：計算今年春季銷售量

DatesInPeriod 函數

DatesInPeriod函數類似DatesBetween函數與DateAdd函數的綜合體，會回傳包含從開始日期至指定間隔間的日期資料表。

函數介紹：回傳包含從開始日期至指定間隔間的日期資料表

DATESINPERIOD (<日期資料行>,<開始日期>,<間隔數>,<時間間隔列舉值>)

DAX範例：計算春節假期（2012/1/21起9天）的銷售量

今年春季銷售量:=CALCULATE(SUM('銷售明細'[銷售金額]), DATESINPERIOD('銷售日別'[資料日],DATE(2012,1,21),9,day))

DatesMTD 函數

DatesMTD函數會回傳包含月初至今之日期的日期資料表，與CALCULATE函數合併使用，就能夠計算月累計（MTD）的數值。但要注意的是，當此量值未與時間維度交錯時，是無法產生有意義的結果。如圖07-11的結果，各位可看到系統評估的結果為「空白」，但當我們利用PowerView來比較2012年1月每日的銷售量與MTD銷售量，就可以看到如圖07-12的結果。

函數介紹：回傳包含月初至今之日期的日期資料表

DATESMTD (<日期資料行>)

DAX範例：計算當月累積MTD銷售量

MTD銷售量:=CALCULATE(SUM('銷售明細'[銷售金額]), DATESMTD('銷售日別'[資料日]))

07-03

DAX擴充函數

08

Model.bim						
[銷售金額]						
=CALCULATE(SUM(銷售明細[銷售金額]), DATESMTD(銷售日期[資料日]))						
Orderid	OrderDate	Productid	銷售數...	銷售金額	產品成本	Storeid
701003051	2010/9/1 上午 12:...	79B5D08D-4...	1		24500	10500
701002551	2010/9/1 上午 12:...	79B5D08D-4...	1		24500	10500
502003725	2010/9/1 上午 12:...	35589AC9-B...	1		27000	12000
404000068	2010/9/1 上午 12:...	35589AC9-B...	1		27000	12000
404006577	2010/9/1 上午 12:...	35589AC9-B...	1		27000	12000
107000037	2010/9/1 上午 12:...	35589AC9-B...	1		27000	12000
403005443	2010/9/1 上午 12:...	79B5D08D-4...	1		24500	10500
				MTD銷售量: (空白)		

圖07-11：計算當月累積MTD銷售量

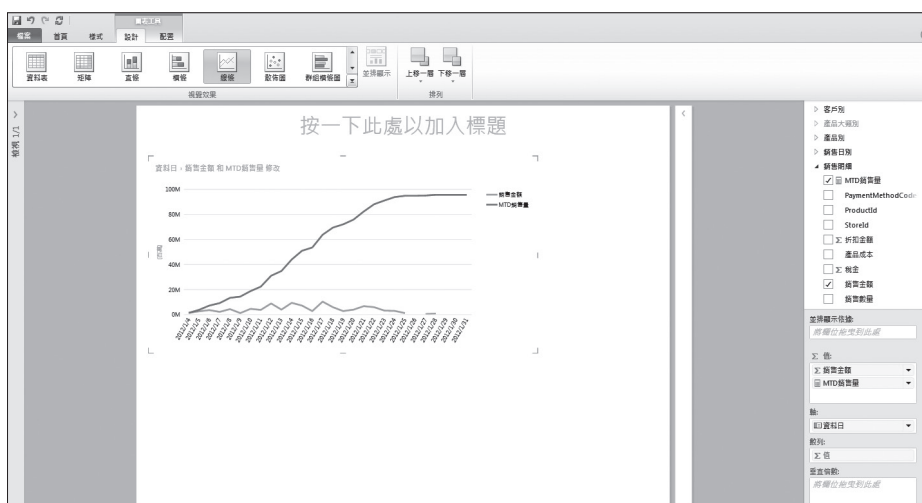


圖07-12：利用PowerView顯示當月銷售量與當月累積MTD銷售量比較

DatesQTD 函數

DatesQTD函數會回傳包含季初至今之日期的日期資料表，與CALCULATE函數合併使用，就能夠計算季累計（QTD）的數值。

函數介紹：回傳包含季初至今之日期的日期資料表

DATESQTD (<日期資料行>)

DatesYTD 函數

DatesYTD函數會回傳包含年初至今之日期的日期資料表，與CALCULATE函數合併使用，就能夠計算年累計（YTD）的數值。

函數介紹：：回傳包含年初至今之日期的日期資料表
DATESYTD (<日期資料行>)

EndOfMonth 函數

針對指定的日期資料行，回傳目前內容中，當月的最後一個日期。

函數介紹：回傳當月的最後一個日期
ENDOFMONTH (<日期資料行>)

EndOfQuarter 函數

針對指定的日期資料行，回傳目前內容中，當季的最後一個日期。

函數介紹：回傳當季的最後一個日期
ENDOFQUARTER (<日期資料行>)

EndOfYear 函數

針對指定的日期資料行，回傳目前內容中，當年度的最後一個日期。

函數介紹：回傳當年度的最後一個日期
ENDOFYEAR (<日期資料行>)

FirstDate 函數

針對指定的日期資料行回傳其中第一個日期。

函數介紹：回傳指定的日期資料行中的第一個日期
FIRSTDATE (<日期資料行>)

FirstNonBlank 函數

FirstNonBlank 函數會根據所指定的資料行，逐筆依照指定運算式進行判斷，取回資料行中，首次運算式評估值非空值的內容。

函數介紹：根據運算式判斷，回傳在資料行中，第一個運算式不為空值的值
FIRSTNONBLANK (<資料行>, <量值運算式>)

舉例來說，若要計算每位客戶的首次購物日，就可以利用 FirstNonBlank 函數指定「銷售明細」資料表的「銷售日」資料行，以「總銷售金額」為判斷運算式，就可以取出首次「總銷售金額」非空值的「銷售日」，結果請參考圖 07-13。

DAX範例：計算每個客戶首次購物日 (客戶維度表上的計算資料行)
=FIRSTNONBLANK('銷售明細'[銷售日], '銷售明細'[總銷售金額])

Custo...	Gender	BirthDate	Education	MaritalStatus	ActiveDate	RegistStoreId	DMFlag	首次購物日
604000895	男性	1966/2/27 上...	大學	M	2008/1/20 上...	604 Y		2010/9/17 上午 12:00:00
302000898	男性	1964/1/17 上...	大學	M	2008/4/22 上...	302 Y		
601000915	男性	1970/5/23 上...	大學	M	2008/3/10 上...	601 Y		
504000967	男性	1958/12/11 ...	大學	M	2008/3/22 上...	504 Y		2011/12/21 上午 12:00:...
404000971	男性	1964/5/22 上...	大學	M	2007/5/24 上...	404 Y		
802000979	男性	1961/3/22 上...	大學	M	2007/5/31 上...	802 Y		2011/1/18 上午 12:00:00
101000984	男性	1955/3/5 上...	大學	M	2008/9/19 上...	101 Y		
104000993	男性	1943/3/11 上...	大學	M	2008/7/1 上午...	104 Y		2011/2/13 上午 12:00:00
605001003	男性	1970/2/9 上...	大學	M	2006/12/14 上...	605 Y		2011/5/21 上午 12:00:00
303001006	男性	1943/11/14 ...	大學	M	2008/3/20 上...	303 Y		
801001012	男性	1944/10/3 上...	大學	M	2008/6/10 上...	801 Y		
110001013	男性	1945/4/24 上...	大學	M	2008/3/14 上...	110 Y		2012/3/14 上午 12:00:00
205001016	男性	1946/1/4 上...	大學	M	2006/12/24 上...	205 Y		2011/8/17 上午 12:00:00
703001021	男性	1947/7/20 上...	大學	M	2008/9/9 上午...	703 Y		
107001022	男性	1947/7/28 上...	大學	M	2008/2/25 上...	107 Y		
401001024	男性	1948/11/16 ...	大學	M	2006/12/14 上...	401 Y		2010/10/14 上午 12:00:...
603001026	男性	1948/10/19 ...	大學	M	2008/3/19 上...	603 Y		2012/4/12 上午 12:00:00
110001033	男性	1951/4/6 上...	大學	M	2008/8/5 上午...	110 Y		2012/1/11 上午 12:00:00
601001048	男性	1965/10/12 ...	大學	M	2008/12/15 上...	601 Y		2011/7/16 上午 12:00:00

圖07-13：計算每位客戶的首次購物日

LastDate 函數

針對指定的日期資料行，回傳其中最後一個日期。

函數介紹：回傳指定的日期資料行中的最後一個日期
LASTDATE (<日期資料行>)

LastNonBlank 函數

LastNonBlank函數會根據所指定的資料行，逐筆依照指定運算式進行判斷，取回資料行中，最後一次運算式評估值非空值的內容。

函數介紹：根據運算式判斷，回傳在資料行中最後一個運算式不為空值的值
LASTNONBLANK (<資料行>, <量值運算式>)

NextDay 函數

NextDay函數會根據指定日期資料行的第一個日期，回傳對應的次一資料日。這個函數在計算前期（日）成長率時，非常有用。

函數介紹：回傳指定日期資料行的第一個日期的次一資料日
NEXTDAY (<日期資料行>)

NextMonth 函數

NextMonth 函數會根據指定日期資料行的第一個日期，回傳對應的次一資料月份。

函數介紹：回傳指定日期資料行的第一個日期的次一資料月
NEXTMONTH (<日期資料行>)

NextQuarter 函數

NextQuarter 函數會根據指定日期資料行的第一個日期，回傳對應的次一資料季度。

函數介紹：回傳指定日期資料行的第一個日期的次一季度
NEXTQUARTER (<日期資料行>)

NextYear 函數

NextYear 函數會根據指定日期資料行的第一個日期，回傳對應的次一資料年度。

函數介紹：回傳指定日期資料行的第一個日期的次一年度
NEXTYEAR (<日期資料行>)

OpeningBalanceMonth 函數

OpeningBalanceMonth 函數的作用就是計算「月初值」。它屬於半加成函數的一種，會根據所傳入日期資料行的日期值，回傳該月月初日的量值數值。

OpeningBalanceMonth 函數廣泛使用於餘額計算、庫存計算等。

函數介紹：根據日期資料行，回傳該月第一天的量值數值
OPENINGBALANCEMONTH (<量值運算式>, <日期資料行>[, <篩選條件>])

OpeningBalanceQuarter 函數

OpeningBalanceQuarter 函數的作用為計算「季初值」。它屬於半加成函數的一種，會根據所傳入日期資料行的日期值，回傳該日期所屬季初日的量值數值。

OpeningBalanceQuarter 函數廣泛應用於餘額計算、庫存計算等。

函數介紹：根據日期資料行，回傳該季度第一天的量值數值
OPENINGBALANCEQUARTER (<量值運算式>, <日期資料行>[, <篩選條件>])

OpeningBalanceYear函數

OpeningBalanceYear函數的作用為計算「年初值」。它屬於半加成函數的一種，會根據所傳入日期資料行的日期值，回傳該日期所屬年初日的量值數值。

OpeningBalanceYear函數廣泛使用於餘額計算、庫存計算等。

函數介紹：根據日期資料行，回傳該年度第一天的量值數值
OPENINGBALANCEYEAR (<量值運算式>,<日期資料行>[,<篩選條件>])

ParallelPeriod函數

ParallelPeriod函數的功能在於根據指定的時間間隔，以回傳指定日期資料行平行的日期。

時間間隔列舉值可為下列值：

- year
- quarter
- month

函數介紹：回傳日期資料行平行的區間，在時間上前移或後移特定間隔數的日期。
PARALLELPERIOD (<日期資料行>,<時間間隔數>,<時間間隔列舉值>)



DAX小幫手

如果您是熟悉SQL或MDX的資訊人員

在MDX中，計算去年同期成長率時，也是利用ParallelPeriod函數，但在MDX與DAX中，兩者的行為模式是略有不同的。在DAX中，ParallelPeriod函數會回傳指定時間間隔下的完整成員。舉例來說，針對2012/7/1使用ParallelPeriod函數，當時時間間隔為year且間隔數為-1時，回傳的結果並非是2011/7/1，而是2011年下的所有完整日期，因此，並不適合用來計算去年同期成長率。如果想要計算去年同期成長率，比較建議使用DateAdd函數，或是後面即將介紹的SamePeriodLastYear函數。

DAX範例：計算去年同期銷售金額(量值)

去年同期銷售金額:=CALCULATE(SUM('銷售明細'[銷售金額]), PARALLELPERIOD('銷售日別'[資料日],-1,year))

去年同期銷售金額:=CALCULATE(SUM('銷售明細'[銷售金額]), DATEADD('銷售日別'[資料日],-1,year))

資料年月	總銷售金額	DATEADD	PARALLELPERIOD
2011/10/1	13720000	11069000	33443500
2011/11/1	126624500	8534000	33443500
2011/12/1	68934500	4900500	33443500
2012/1/1	95451000	7998000	788006000
2012/2/1	87104500	54890000	788006000
2012/3/1	41999500	28082500	788006000
2012/4/1	48279000	57205000	788006000
2012/5/1	54382000	36957500	788006000
2012/6/1	54820500	45795000	788006000
2012/7/1	74436000	77862500	788006000
2012/8/1	75801000	55339000	788006000
2012/9/1	63795000	91117500	788006000
2012/10/1	86762500	137200000	788006000
2012/11/1	62164500	126624500	788006000
2012/12/1	36445500	68934500	788006000
2013/1/1	58404500	95451000	775441000
2013/2/1	45090000	87104500	775441000
2013/3/1	22542500	41999500	775441000
2013/4/1	45471500	48279000	775441000
2013/5/1	30593500	54382000	775441000
2013/6/1	39669500	54820500	775441000
2013/7/1	68246000	74436000	775441000
2013/8/1	2820000	75801000	775441000

圖07-14：去年同期銷售金額兩種寫法的比較

PreviousDay 函數

PreviousDay 函數會根據指定日期資料行的第一個日期，回傳對應的前一資料日期。

函數介紹：回傳指定日期資料行的第一個日期的前一資料日

PREVIOUSDAY (<日期資料行>)

舉例來說，如果要計算前日的銷售金額，就可以在銷售明細表中，加入以下量值。但加入量值後，如同各位在圖07-15所見，此量值評估的結果為「空白」，是因為並無對應的時間維度與其交錯。但在圖07-16中，我們將此量值與銷售日期交錯時，就可以看到正確的回傳結果。各位也可以利用此結果與「當日銷售金額」，進行進一步的運算，以取得前日銷售金額成長率。

DAX 範例：計算前一日銷售金額（量值）

前一日銷售金額:=CALCULATE(SUM('銷售明細'[銷售金額]), PREVIOUSDAY('銷售日別'[資料日]))

Modelbim* x

[銷售金額]	=CALCULATE(SUM(銷售明細[銷售金額]), PREVIOUSDAY(銷售日期[資料日]))						
ProductId	銷售數...	銷售金額	產品成本	StoreId	PaymentMethodCode		
{35589AC9-BE9D-40BF-8D44-2F1C2...	1	189500	10500	605		1	
{35589AC9-BE9D-40BF-8D44-2F1C2...	1	189500	10500	605		1	
{35589AC9-BE9D-40BF-8D44-2F1C2...	1	189500	10500	605		1	
{35589AC9-BE9D-40BF-8D44-2F1C2...	1	189500	10500	101		1	
{35589AC9-BE9D-40BF-8D44-2F1C2...	1	189500	10500	101		1	
{35589AC9-BE9D-40BF-8D44-2F1C2...	1	189500	10500	204		1	
{35589AC9-BE9D-40BF-8D44-2F1C2...	1	189500	10500	204		1	
{35589AC9-BE9D-40BF-8D44-2F1C2...	1	189500	10500	204		1	
{35589AC9-BE9D-40BF-8D44-2F1C2...	1	189500	10500	204		1	
{35589AC9-BE9D-40BF-8D44-2F1C2...	1	189500	10500	204		1	
{35589AC9-BE9D-40BF-8D44-2F1C2...	1	189500	10500	103		1	
{35589AC9-BE9D-40BF-8D44-2F1C2...	1	189500	10500	103		1	
		MTD銷售量: (空白)	總產品成本: 9918...				
		總銷售金額: 1909728000	毛利率: 94.81%				
		前一日銷售金額: (空白)					

圖07-15：加入前一日銷售金額

資料日			前一日銷售金額	
2012/1/4	1182000			
2012/1/5	2424500	1182000		
2012/1/6	3398500	2424500		
2012/1/7	1970000	3398500		
2012/1/8	4181000	1970000		
2012/1/9	1012500	4181000		
2012/1/10	4389000	1012500		
2012/1/11	3579000	4389000		
2012/1/12	8690000	3579000		
2012/1/13	3765000	8690000		
2012/1/14	9194000	3765000		
2012/1/15	6988500	9194000		
2012/1/16	2599500	6988500		
2012/1/17	10162500	2599500		
2012/1/18	5729500	10162500		
2012/1/19	2605000	5729500		
2012/1/20	3748500	2605000		
2012/1/21	6523000	3748500		
2012/1/22	5746000	6523000		
2012/1/23	2993500	5746000		

銷售明細

☐ MTD銷售量
 ☐ PaymentMethodCode
 ☐ StoreId
 ☐ 毛利率
 ☐ 折扣金額
 ☒ 前一日銷售金額
 ☐ 產品成本
 ☐ 產品最高銷售額
 ☐ 税金
 ☐ 銷售金額
 ☐ 銷售數量
 ☐ 總產品成本
 ☒ 總銷售金額

並排顯示依據:

將欄位拖曳到此處

欄位:

資料日

總銷售金額

前一日銷售金額

圖07-16：前日銷售金額

PreviousMonth函數

PreviousMonth函數會根據指定日期資料行的第一個日期，回傳對應的前一資料月份，可用於計算前期（月）成長率。

函數介紹：回傳指定日期資料行，第一個日期的前一資料月
PREVIOUSMONTH(<日期資料行>)

PreviousQuarter 函數

PreviousQuarter 函數會根據指定日期資料行的第一個日期，回傳對應的前一資料季度，可用於計算前期（季）成長率。

函數介紹：回傳指定日期資料行的第一個日期的前一季度
PREVIOUSQUARTER (<日期資料行>)

PreviousYear 函數

PreviousYear 函數會根據指定日期資料行的第一個日期，回傳對應的前一資料年度，可用於計算前期（年）成長率。

函數介紹：回傳指定日期資料行的第一個日期的前一年度
PREVIOUSYEAR (<日期資料行>)

SamePeriodLastYear 函數

SamePeriodLastYear 函數會根據指定的日期資料行，回傳對應之前一年的日期。其功能與 DATEADD(<日期資料行>, -1, year) 的結果完全相同，普遍應用在計算去年同期成長率的情境。

函數介紹：回傳向前平移一年的對應日期
SAMEPERIODELASTYEAR (<日期資料行>)

StartOfMonth 函數

針對指定的日期資料行，回傳目前內容中，當月的第一個日期。

函數介紹：回傳當月的第一個日期
STARTOFMONTH (<日期資料行>)

StartOfQuarter 函數

針對指定的日期資料行，回傳目前內容中，當季的第一個日期。

函數介紹：回傳當季的第一個日期
STARTOFAUARTER (<日期資料行>)

StartOfYear 函數

針對指定的日期資料行，回傳目前內容中，當年的第一個日期。

函數介紹：回傳當年度的第一個日期
STARTOFYEAR (<日期資料行>)

TotalMTD函數

TotalMTD函數會根據指定之日期資料行，將指定量值運算式進行月累計（MTD），是非常常見的時間智慧運算。弱勢希望在實際商業分析時，使用MTD分析，可透過TotalMTD或是DatesMTD兩種計算方法。以下列出兩種寫法供各位參考。

函數介紹：回傳根據指定日期資料行，所計算之量值月累計值
TOTALMTD (<量值運算式>, <日期資料行> [, <篩選條件>])

DAX範例：計算月累計銷售金額

MTD銷售量:=TOTALMTD (SUM('銷售明細'[銷售金額]), '銷售日別'[資料日])

MTD銷售量:=CALCULATE (SUM('銷售明細'[銷售金額]), DATESMTD('銷售日別'[資料日]))

TotalQTD函數

TotalQTD函數會根據指定之日期資料行，將指定量值運算式進行季度累計（QTD）。

函數介紹：回傳根據指定日期資料行所計算之量值季度累計值
TOTALQTD (<量值運算式>, <日期資料行> [, <篩選條件>])

TotalYTD函數

TotalYTD函數會根據指定之日期資料行，將指定量值運算式進行年度累計（YTD）。

函數介紹：回傳根據指定日期資料行所計算之量值年度累計值
TOTALYTD (<量值運算式>, <日期資料行> [, <篩選條件>])

7-3-4 統計函數

AddColumns函數

AddColumns函數可協助在指定資料表中，加入計算資料行。

函數介紹：將「計算欄位」加入至給定的資料表或資料表運算式。

ADDCOLUMNS (<資料表>, <資料行名稱>, <量值運算式>[, <資料行名稱>, <量值運算式>]…)

AverageX 函數

AverageX函數是以指定資料表為基礎，計算運算式值的算術平均數。其計算的邏輯與Average完全相同，會排除所有非數字與空值。

函數介紹：回傳資料表中運算式值的算術平均數

AVERAGEX (<資料表>, <量值運算式>)

CountAX 函數

CountAX函數是以指定資料表為基礎，計算運算式值的計數。其計算的邏輯與CountA完全相同，會排除所有空值，但可納入非數值資料，包括：文字、日期和邏輯值。

函數介紹：回傳資料表中運算式值的計數 (非空值)

COUNTAX (<資料表>, <量值運算式>)

CountRows函數

CountRows函數會回傳資料表或資料表函數回傳結果之資料列筆數。

函數介紹：回傳資料表之資料筆數

COUNTROWS (<資料表>)

DAX範例：計算指定產品的銷售筆數 (將此計算資料行加入製產品維度表)

=COUNTROWS (RELATEDTABLE ('銷售明細'))

CountX 函數

CountX函數是以指定資料表為基礎，計算運算式值的計數。其計算的邏輯與Count完全相同，會排除所有非數字與空值。

函數介紹：回傳資料表中運算式值的計數
COUNTX (<資料表>, <量值運算式>)

CrossJoin函數

CrossJoin函數會計算資料表間的笛卡兒乘積（Cartesian product）。回傳結果的資料筆數必定等於輸入引數中的個別資料表的資料筆數乘積，此函數較常被應用於DAX陳述句查詢。

函數介紹：回傳指定資料表間的笛卡兒乘積

CROSSJOIN (<資料表1>, <資料表2>[, <資料表3>]…)

DAX範例：產生個年度與產品作業系統的排列組合（DAX陳述句查詢）

```
Evaluate(  
  CrossJoin(  
    Values('銷售日別'[銷售年度]),  
    Values('產品別'[作業系統])  
  )  
)
```



銷售日別[銷售年度]	產品別[作業系統]
2007	iOS 4
2011	iOS 4
2012	iOS 4
2010	iOS 4
2009	iOS 4
2008	iOS 4
2014	Android 4.0
2013	Android 4.0
2000	Android 4.0
2001	Android 4.0
2002	Android 4.0
2003	Android 4.0

圖07-17：CrossJoin

DistinctCount函數

DistinctCount函數可用來計算指定資料行內容去重複計數的結果。

函數介紹：回傳資料表中運算式值的去重複計數
DISTINCTCOUNT (<資料行>)

Generate函數

MDX中，Generate函數經常用來重組階層結構，在DAX也具有類似的作用。當分析者指定了兩組資料表的資料行時，利用Generate函數可以資料表1為基礎，逐一比對資料表2的資料內容來建構新的資料表，行為模式類似T-SQL中的LEFT JOIN。

函數介紹：回傳將資料表1的成員逐一代入資料表2之後，所計算之成員
GENERATE (<資料表1>, <資料表2>)

GenerateAll函數

GenerateAll函數與Generate函數類似，唯一的差別在於，Generate函數會排除資料表1中的空值，而GenerateAll函數則會予以保留。

函數介紹：回傳將資料表1的成員逐一代入資料表2之後，所計算之成員，但包含空值
GENERATEALL (<資料表1>, <資料表2>)

MaxX 函數

MaxX函數是以指定資料表為基礎，計算運算式值的最大值。其計算的邏輯與Max完全相同，會排除所有非數字與空值。

函數介紹：回傳資料表中運算式值的最大值
VALUE (<文字>)

MinX函數

MinX函數是以指定資料表為基礎，計算運算式值的最小值。其計算的邏輯與Min完全相同，會排除所有非數字與空值。

函數介紹：回傳資料表中運算式值的最小值
VALUE (<文字>)

Rank.EQ函數

Rank.EQ函數會回傳指定值在查閱資料表中的排序值。若是查閱資料表出現重複值時，則會出現相同排序，但下一筆資料排序，會是次序值加上重複值的數目（例如，有兩個值排序為3，則下一筆排序值會是5）。

排序模式可以是以下的輸入值：

- 0（也可以輸入「False」）：降冪（遞減）排序。
- 1（也可以輸入「True」）：升冪（遞增）排序。

此外，須注意的是，在Rank.EQ函數中，資料行不得來自於ADDCOLUMNS、ROW或SUMMARIZE所計算生成的資料行。

函數介紹：回傳指定值在查閱資料表中的排序值。

RANK.EQ(<數值>, <查閱資料行>[, <排序模式>])

DAX範例：產生產品售價的排名（在產品維度表的計算資料行）

=RANK.EQ('產品別'[ListPrice], '產品別'[ListPrice], 0)



產品名稱	產品描述	品牌	作業系統	產品類別	上架日	E...	ListPrice	售價...	加入資料行
Apple iPhone 4 1...	Apple 推出更...	Apple	iOS 4	二進位資料	2010/9/1 上...		20800	9	
Apple iPhone 4 3...	Apple 推出更...	Apple	iOS 4	二進位資料	2011/4/1 上...		23000	5	
SAMSUNG GAL...	三星推出首款...	SAMSUNG	Android 4.0	二進位資料			20400	11	
NOKIA Lumia 900	諾基亞於世界...	NOKIA	Windows Phone 7.5	二進位資料			21000	6	
Apple iPhone 4 3...	Apple 推出更...	Apple	iOS 4	二進位資料	2010/9/1 上...		23300	4	
Apple iPhone 4 1...	Apple 推出更...	Apple	iOS 4	二進位資料	2011/4/1 上...		20400	11	
HTC TITAN II	HTC TITAN II...	HTC	Windows Phone 7.5	二進位資料			25100	2	
HTC Desire S	HTC Desire S在...	HTC	Android 2.3	二進位資料	2011/4/1 上...		9100	26	
ASUS PadFone	華碩 ASUS Pad...	ASUS	Android 4.0	二進位資料			17500	16	
Apple iPhone 4S	Apple 蘋果推...	Apple	iOS 5	二進位資料	2011/12/1 ...		20800	9	
MOTO MOTOS...	MOTOROLA針...	MOTORO...	Android 2.3	二進位資料			5990	33	
HTC Radar 雷達...	宏達電推出旗...	HTC	Windows Phone 7.5	二進位資料			11000	21	
Apple iPhone 4 8...	Apple 推出更...	Apple	iOS 4	二進位資料	2010/9/1 上...		17800	15	
HTC One X 極速...	HTC推出首款...	HTC	Android 4.0	二進位資料			20900	8	
HTC Desire HD	HTC Desire HD...	HTC	Android 2.2	二進位資料	2012/5/1 上...		10600	22	
Acer Allegro	宏碁推出最新...	Acer	Windows Phone 7.5	二進位資料			7200	31	
HTC Rhyme 音...	HTC Rhyme 音...	HTC	Android 2.3	二進位資料			9800	23	
NOKIA Lumia 610	諾基亞鎖定年...	NOKIA	Windows Phone 7.5	二進位資料			7000	32	
Apple iPhone 4 8...	Apple 推出更...	Apple	iOS 4	二進位資料	2011/4/1 上...		17900	13	

圖07-18：計算產品售價排序

RankX 函數

RankX函數可用來處理較為複雜的排序。前面介紹的Rank.EQ主要是執行「排序查閱」，也就是找出某個數值在查閱資料表內的排序，而RankX則是直接針對指定資料表內的每一筆資料列，並針對量值運算式所算出來的結果，來進行排序值的計算。

次序判斷列舉值如下：

- Skip：出現重複值時，則會出現相同排序，但下一筆資料排序會是次序值加上重複值的數目，此為預設值。
- Dense：下一筆資料排序，會是次序值加上1。

函數介紹：回傳在指定資料表內，依照量值運算式結果的排序值

RANKX (<資料表>, <量值運算式>[, <數值>[, <排序模式>[, <次序判斷列舉值>]]], <量值運算式>[, <數值>[, <排序模式>[, <次序判斷列舉值>]]]…)

DAX範例：產生產品銷售金額的排名（在產品維度表的計算資料行）

=RANKX (ALL ('產品別'), SUMX (RELATEDTABLE ('銷售明細'), '銷售明細' [銷售金額]))

ROW 函數

ROW函數是藉由指定每個資料行名稱及其對應的值運算式，來建構單一的資料列。

函數介紹：回傳單一資料列

ROW (<資料行名稱>, <值運算式>[, <資料行名稱>, <值運算式>]…)

STDEV.P 函數

STDEV.P函數會用來計算整個母體（Population）的標準差，與T-SQL語法中，STDEVP的計算邏輯相同，會根據以下公式來計算標準差：

$$\sqrt{\frac{\sum (x - \bar{x})^2}{n}}$$

函數介紹：回傳整個母體的標準差。

STDEV.P (<資料行>)

但計算時，資料行的筆數必須大於等於2，否則，會回傳錯誤。

STDEV.S 函數

STDEV.S函數會用來計算樣本母體（Sample）的標準差，與T-SQL語法中，STDEV的計算邏輯相同，會根據以下公式來計算標準差：

$$\sqrt{\frac{\sum(x - \bar{x})^2}{n - 1}}$$

函數介紹：回傳樣本的標準差。

STDEV.S (<資料行>)

但計算時，資料行的筆數必須大於等於2，否則，會回傳錯誤。

STDEVX.P 函數

STDEVX.P函數與STDEV.P函數類似，只不過，它是透過資料表來做為計算母體，且能計算指定量值運算式的標準差。

函數介紹：回傳指定資料表中，做為整個母體的量值運算式標準差。

STDEVX.P (<資料表>, <量值運算式>)

STDEVX.S 函數

STDEVX.S函數與STDEV.S函數類似，只不過，它是透過並資料表來做為計算樣本母體，並可以計算指定量值運算式的標準差。

函數介紹：回傳指定資料表中做為樣本母體的量值運算式標準差。

STDEVX.S (<資料表>, <量值運算式>)

Summarize函數

Summarize函數最常見的功能就是用來產生彙總後的資料。在此函數中透過指定資料表、作為彙總GroupBy的資料行，以及要產出彙總值的運算式，即可完成彙總運算。這也是BISM表格式模型與BISM多維度模型最大的差異之處，後者是透過事前的彙總，來做為高速運算的基礎，而BISM表格式模型，則是利用Summarize函數即時透過xVelocity技術來進行彙總運算。Summarize函數通常用於DAX陳述句查詢中。比較複雜的功能，我們將在下一章中詳細介紹。

函數介紹：回傳彙總後的摘要資料表

SUMMARIZE (<資料表>, <groupBy資料行>[, <groupBy資料行>]...[, <資料行名稱>, <值運算式>]...)

DAX範例：產生產品銷售金額的排名

```
Evaluate(
Summarize (
    '銷售明細',
        '銷售日別'[銷售年度]
        , '客戶別'[性別]
        , "總銷售金額", SUM('銷售明細'[銷售金額])
        , "總銷售數量", SUM('銷售明細'[銷售數量])
    ))
```

07-03

MDXQuery3.mdx - AL...Administrator)*

Cube: Model

量值群組: <全部>

Model

- Measures
- KPI
- 客戶別
- 產品大類別
- 產品別
- 銷售日別
 - 是否為假日
 - 資料日
 - 資料年月
 - 銷售年度
 - 日曆日階層
- 銷售明細
 - PaymentMethodCode
 - StoreId
 - 折扣金額
 - 產品成本
 - 產品最高銷售額
 - 税金
 - 銷售金額
 - 銷售數量

100 %

訊息 結果

銷售日別[銷售年度]	客戶別[性別]	[總銷售金額]	[總銷售數量]
2013	女性	157635500	839
2013	男性	155202000	826
2011	女性	390392000	2020
2011	男性	397614000	2036
2012	女性	402961000	2097
2012	男性	372480000	1937
2010	女性	17193500	678
2010	男性	16250000	643

圖07-19：產生彙總資料表

TopN 函數

TopN函數可用來回傳指定資料表的前N筆資料。在計算業績最好的前N名業務員，或是找出毛利率最高的前N項產品，都非常有用。

排序模式可以是以下的輸入值：

- 0（也可以輸入「False」）：降冪（遞減）排序。
- 1（也可以輸入「True」）：升冪（遞增）排序。

函數介紹：回傳資料表中依照orderBy值運算式排序後，前n筆案例

TOPN(<前N名>, <資料表>, <orderBy值運算式>, [<排序模式>[, <orderBy值運算式>, [<排序模式>]]...])

DAX範例：找出最暢銷的前3種產品

```
Evaluate(
TopN(3,
SUMMARIZE('銷售明細'
, '產品別'[產品名稱]
, "總銷售金額", SUM('銷售明細'[銷售金額])
), '銷售明細'[總銷售金額]
))
```

The screenshot shows a software interface with a left-hand pane containing a tree view of data sources and a right-hand pane displaying a DAX query and its results. The DAX query is: `Evaluate(TopN(3, SUMMARIZE('銷售明細', '產品別'[產品名稱], "總銷售金額", SUM('銷售明細'[銷售金額])), '銷售明細'[總銷售金額]))`. The results pane shows a table with two columns: '產品別[產品名稱]' and '[總銷售金額]'. The data rows are: Apple iPhone 4 16GB (610464000), Apple iPhone 4 32GB (655189000), and Apple iPhone 4 8GB (644075000).

產品別[產品名稱]	[總銷售金額]
Apple iPhone 4 16GB	610464000
Apple iPhone 4 32GB	655189000
Apple iPhone 4 8GB	644075000

圖07-20：找出最暢銷的前三種產品

VAR.P函數

VAR.P函數會用來計算整個母體（Population）的變異數（標準差的平方）。與T-SQL語法中，VARP的計算邏輯相同，並根據以下公式來計算變異數：

$$\frac{\sum(x - \bar{x})^2}{n}$$

函數介紹：回傳整個母體的變異數。

VAR.P(<資料行>)

VAR.S 函數

VAR.S 函數會用來計算樣本母體（Sample）的變異數（標準差的平方）。與 T-SQL 語法中，VAR 的計算邏輯相同，並根據以下公式來計算變異數：

$$\frac{\sum (x - \bar{x})^2}{n - 1}$$

函數介紹：回傳樣本母體的變異數。

VAR.S (<資料行>)

VARX.P 函數

VARX.P 函數與 VAR.P 函數類似，只不過，它是透過資料表來做為計算母體，並可以計算指定量值運算式的變異數。

函數介紹：回傳指定資料表中做為整個母體的量值運算式變異數。

VARX.P (<資料表>, <量值運算式>)

VARX.S 函數

VARX.S 函數與 VAR.S 函數類似，只不過它是透過資料表來做為計算樣本母體，並可以計算指定量值運算式的變異數。

函數介紹：回傳指定資料表中，做為樣本母體的量值運算式變異數。

VARX.S (<資料表>, <量值運算式>)

7-3-5 邏輯函數

Switch 函數

Switch 函數，基本的使用模式與 C# 中的 Switch 完全相同。它的作用有點類似 T-SQL 語法中的 CASE WHEN ... ELSE END 句型，但不同之處在，於 T-SQL 語法可以在 WHEN 區段內，設定個別的條件判斷式，而 Switch 函數，則是先透過「值運算式」算出一個值，然後根據值的內容進行對應轉換。各位可以參考下列「產生年齡級距」範例。

函數介紹：回傳將列舉值依照條件式轉換對應後的結果

SWITCH(<值運算式>, <值1>, <結果1>[, <值2>, <結果2>]…[, <其餘值>])

DAX範例：產生年齡級距（客戶別維度表的計算資料行）

=SWITCH(FLOOR([年齡]/10,1), 0, "10歲以下", 1, "10~19歲", 2, "20~29歲", 3, "30~39歲", 4, "40~49歲", 5, "50~59歲", "60歲以上")

The screenshot shows a table with columns: Gender, MaritalStatus, ActiveDate, RegistStoreId, DMFlag, 首次購物日, 年齡, 年齡級距, and 加入資料行. The '年齡級距' column displays the result of the SWITCH function applied to the '年齡' column. For example, an age of 46 results in '40-49歲', and an age of 51 results in '50-59歲'.

Gender	MaritalStatus	ActiveDate	RegistStoreId	DMFlag	首次購物日	年齡	年齡級距	加入資料行
M		2008/1/20 上...	604	Y	2010/9/17 上午 12:00:00	46	40-49歲	
M		2008/4/22 上...	302	Y		49	40-49歲	
M		2008/3/10 上...	601	Y		42	40-49歲	
M		2008/3/22 上...	504	Y	2011/12/21 上午 12:00:...	54	50-59歲	
M		2007/5/24 上...	404	Y		48	40-49歲	
M		2007/5/31 上...	802	Y	2011/1/18 上午 12:00:00	51	50-59歲	
M		2008/9/19 上...	101	Y		57	50-59歲	
M		2008/7/1 上午...	104	Y	2011/2/13 上午 12:00:00	69	60歲以上	
M		2006/12/14 上...	605	Y	2011/6/21 上午 12:00:00	42	40-49歲	
M		2008/3/20 上...	303	Y		69	60歲以上	
M		2008/6/10 上...	801	Y		68	60歲以上	
M		2008/3/14 上...	110	Y	2012/3/14 上午 12:00:00	67	60歲以上	
M		2006/12/24 上...	205	Y	2011/8/17 上午 12:00:00	67	60歲以上	
M		2008/9/9 上午...	703	Y		65	60歲以上	
M		2008/2/25 上...	107	Y		65	60歲以上	

圖07-21：計算年齡級距

7-3-6 資訊函數

Contains函數

Contains函數可用來檢索指定的資料表中，是否存在資料行與指定值相符的案例。回傳值為布林值。

函數介紹：回傳是否存在資料行與指定值相符的案例

CONTAINS(<資料表>, <資料行1>, <值1>[, <資料行2>, <值2>]…)

CustomData函數

CustomData函數用來檢查連線字串中，是否存在CustomData屬性內容。如果有，則會回傳「OK」。

函數介紹：回傳連線字串中是否存在CustomData屬性內容

CUSTOMDATA ()

LookUpValue函數

LookUpValue函數可透過指定查閱資料行以及賦予查閱值，以回傳查閱結果中，指定資料行的值。這在進行代碼與代碼說明轉換時，非常實用。

函數介紹：回傳查閱值

LOOKUPVALUE (<回傳資料行>, <查閱資料行1>, <查閱值1>[, <查閱資料行2>, <查閱值2>]…)

DAX範例：查閱「NOKIA Lumia 900」的定價

=LOOKUPVALUE ('產品別'[ListPrice], '產品別'[產品名稱], " NOKIA Lumia 900 ")

UserName函數

UserName函數會回傳使用者端的完整名稱，其格式為「網域名稱(電腦名稱)\使用者名稱」，此函數可用來設計出更具彈性的安全性設定。

函數介紹：回傳使用者端的網域名稱及使用者名稱

USERNAME ()



