

推薦序

我們生活在一個瞬息萬變的時代，資訊科技的創新與變革無時無刻不在推進著世界的前行。然而，即使在這樣的變化中，有一樣東西始終是我們無法忽視的基石，那就是資料庫。

從我進入微軟，擔任 SQL Server 的 presales 開始，我就有幸認識了百敬老師。他是我認識的最專業，最有熱情的 SQL Server 專家之一。在資訊科技的海洋中，百敬老師就像是一座燈塔，為我們照亮前方的道路。

在台灣，SQL Server 一直是支撐著科技產業發展的重要力量。在這裡，我們每一位工作者的努力，都以 SQL Server 為基礎，建造出壯麗的科技樓宇。而現在，透過百敬老師的新書《SQL Server 管理實戰》，我們有了一把更強大的工具，可以更好地管理和運用這座巨大的基石。

這本書不僅是一本技術手冊，更是一本啟示錄。百敬老師用他深厚的學識和豐富的經驗，帶領我們走入 SQL Server 的世界，讓我們看到其深處的奧秘和可能。透過這本書，我們可以精進我們的技術，提升我們的資料庫管理能力，更好地面對未來的挑戰。

最後，我希望這本書能對所有的讀者都有所幫助，讓我們一同在 SQL Server 的世界裡探索、學習，並在未來的日子裡創造出更多的可能。

感謝百敬老師的付出，感謝所有在這個行業中默默耕耘的人。讓我們一起，用 SQL Server 寫下科技產業的新篇章。

台灣微軟雲端解決方案副總經理 **宋明遠**

作者序

因應系統相連、巨量資料、服務不停機、安全沒漏洞等需求，資料庫技術蓬勃發展。就企業而言，資料倍增、應用變多、存取量大，人員、流程、技術都在加速整合。

在應用程式群中，SQL Server 有如心臟，若管理者不能熟練地架設、維運與監控，勢必讓整體系統不穩。換言之，企業日益需要更優秀的 DBA，讓系統更大、更穩、更快、更安全。我們 IT 人只有一再地學習、規劃、應用、檢討、改善，但求與時俱進，方能安身立命。

自 1970 年 E. F. Codd 提出關聯理論，並由 IBM 實驗室發展出 SQL 語言至今，成千上萬的菁英在資料庫領域耕耘了 50 多年，其領域知識博大精深。從任何一個面向切入大型的資料庫產品，例如微軟 SQL Server 自 1989 年上市，累積超過了 30 年的進化，已有大量可說明的內容。因此，我們每次在 SQL Server 更版時，都計畫以一系列的書籍來介紹，期待書籍間也有層次架構，建立起研讀的縱深。而本書定位在起步，幫助 SQL Server 的入門者可以上手操作。但並非深入的 SQL Server 專項議題，希望能讓讀者按圖索驥。

本書的章節選定有些困難，既然是管理，理當討論建立帳戶，以認證、授權。但這又是安全的基本。應該放在《SQL Server 安全》一書，不能讓你花錢買重複的內容。幸好建帳號不難，因此本書不著墨，將其留待討論安全的專書，讓其理論與操作能一氣呵成。其他的議題，如 T-SQL 語言、開發資料庫內物件、效能調校...等，皆是如此，我們期待為不同主題提供深入的解析。

操作型的書很難「十年磨一劍」，畢竟昨是今非/今是昨非更迭快速，少有恆常與經典。但我們仍希望書籍內容能隨著 SQL Server 產品成長，藉由更版先前的著作，讓範例越來越扼要明白，畫面越來越整合，說明越來越清楚，隨著時間累積，能在廣泛的經驗中凝結出通則。並以團隊合作的方式追趕速度，增加廣度。

當 SQL Server 2022 版推出時，本以為它玩不出多少新意。沒想到它讓人驚豔，在許多方面都提出改良與增強，以善用當前新的軟硬體。且此版強化的多是針對引擎本體，換句話說，大多數人只要升級，或多或少都有好處，而不需要改寫程式或變更維護流程。

玉茹、妙謙、巧玫、英豪與輝瑞浸淫 SQL Server 多年，都任職開發與管理資料庫的工作，有幸結伴，透過文字切磋，一起為所學所知留下註腳，實是幸運。但因自己協同作業不足，所學不夠而有疏漏，須先向你說聲抱歉，感謝你的支持與包容。也同時感謝編輯，有妳們的敦促與校閱，本書才得以即時而正確地面市。

匆匆復匆匆，日子輪轉著，悲戚凋零，欣悅後進。有幸攜手同遊，願將努力成果與慧共享。

備份與還原資料庫

5

CHAPTER

資料庫備份的重要性在資料庫日常作業裡定是前幾名，無論資料庫處於再怎麼安全的環境，仍無法完全預防天災或人禍造成的資料毀損風險，例如地震、火災、斷電，甚至使用者惡意或無意刪除了資料庫、資料庫內的物件或資料表裡的記錄。資料庫永遠有毀損的可能性，除了在乎平時做好備份工作妥善保存備份資料，最好可以將備份資料存於異地並留意檔案的安全性管理¹。另外，定期做資料庫還原或災難演練，除了可確認備份資料的完好也能熟練復原程序，當資料庫不幸毀損時才可在第一時間內修復，並在最短時間裡回復資料庫的可用性。

本章將說明 SQL Server 資料庫的備份方式及各種備份技巧，包括：

- 建立與使用備份裝置。
- 各種備份類型的意義與使用時機。
- 執行各種資料庫備份的方式。
- 設計備份策略。

¹ 管理者雖會注意線上系統的安全性，卻容易對備份資料卻疏於防範，一旦有心人士拿到備份也就等於拿到資料庫，不可不慎。SQL Server 2014 版後增加了備份加密功能，可以補強備份資料的安全性。

5.1 SQL Server 基本備份觀念

探討 SQL Server 備份方式以前，先認識以下兩個基本觀念以瞭解 SQL Server 的備份機制：

- SQL Server 的線上備份機制運作原理。
- SQL Server 備份裝置。

5.1.1 SQL Server 的線上備份機制

為了資料庫的可用性（Availability），SQL Server 支援線上備份，即使在運作時也以可備份整個資料庫或是只備份交易記錄。資料庫記錄檔（ldf）裡不存放資料庫物件，只存放交易記錄，部分使用者常誤以為 SQL Server 的備份只備份資料檔而不包含記錄檔！

假設備份 MyDB 資料庫需耗時 30 分鐘，某天 8:00 pm 開始執行資料庫備份，實際備份完成的時間為 8:30 pm。若備份作業執行至 8:29 pm 時正好有其他使用者異動了某幾筆資料，但這幾筆資料在 8:29 pm 前就已經被備份過資料檔（.mdf、.ndf）了。

執行備份期間仍有使用者存取或異動資料，這是線上備份很可能碰到的狀況。SQL Server 為了解決此問題，在進行完整資料庫備份時會遵循下列原則：

- 在備份過程中先備份資料檔。
- 當備份動作即將完成時，會將“備份期間所產生的交易記錄”也備份下來。

藉由這些交易記錄，SQL Server 便可以追蹤哪些資料在備份期間被更改，之後資料庫還原時只需先回存資料檔，再利用交易記錄來更新資料，便能夠將資料庫回復至備份時的狀態。

以上述的例子而言，8:30pm 完成的備份會抄寫 8:29pm 的異動記錄，因此回復時，可以回到原始 8:30pm 完整的記錄內容。

5.1.2 與備份 / 還原有關的資料庫選項－復原模式

SQL Server 資料庫的「復原模式」選項，可以針對交易記錄的保存設定三種模式：完整、簡單、大量記錄。這三種模式影響了資料庫備份方式及還原程序，資料庫建立時復原模式預設值會繼承自 Model 系統資料庫，之後可以手動調整。設定畫面如圖 5.1：

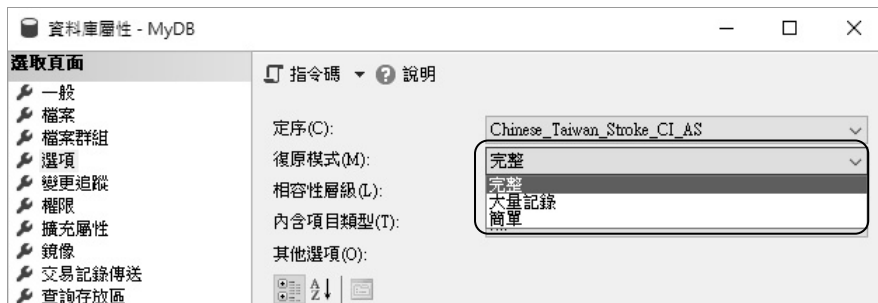


圖 5.1 SQL Server 資料庫的三種「復原模式」

復原模式三種選項的說明如下表：

表 5.1 三種「復原模式」的意義

復原模式	說明
完整	SQL Server 會在資料庫的記錄檔內保存所有交易記錄，直到執行交易記錄備份後交易記錄檔空間才可重複使用。完整復原模式的資料庫，可以執行完整、差異與交易記錄備份。
簡單	SQL Server 在每次發生檢查點 (checkpoint) 事件後，與資料檔 (mdf、ndf) 同步已完成 (commit) 或回復 (rollback) 的交易記錄，SQL Server 便可重複使用存放這些交易記錄的硬碟空間。簡單復原模式的資料庫，只能執行完整備份與差異備份，無法備份已清掉的交易記錄。

復原模式	說明
大量記錄	<p>大多數的交易記錄會被保留，但不保存 BULK 行為（如建立索引或大量載入資料）產生的交易記錄。大量記錄復原模式的資料庫支援完整、差異與交易記錄備份，但無法保證可將資料庫還原至大量資料載入期間中的時間點。</p> <p>且因為大量記錄在 ldf 檔內只記載了受影響的資料 page，在交易記錄備份時，會從資料檔（mdf、ndf）抄寫這些 page，所以備份交易記錄出來的檔案大小可能超過來源 ldf 的大小。</p>

若想要更改資料庫的復原模式選項設定，可以使用 ALTER DATABASE 陳述式，範例如下：

```

/*完整模式*/
ALTER DATABASE [資料庫名稱] SET RECOVERY FULL WITH NO_WAIT
/*簡單模式*/
ALTER DATABASE [資料庫名稱] SET RECOVERY SIMPLE WITH NO_WAIT
/*大量記錄模式*/
ALTER DATABASE [資料庫名稱] SET RECOVERY BULK_LOGGED WITH NO_WAIT

```

5.2 使用備份裝置

接下來介紹 SQL Server 裡對應磁碟檔案的邏輯備份裝置。

5.2.1 什麼是備份裝置

執行備份作業時，最簡單的方式就是直接將資料庫備份到檔案或磁帶，但這種方式難以查詢每個備份檔案或磁帶存放了哪些資料。如圖 5.2，將 MyDB 資料庫備份至 MyDB.bak，MyDB.bak 對於 DBA 來說就是

一個單純的檔案。若多次備份到不同檔案後，要從多個檔案查詢備份時間、備份種類或是備份次數將會較為麻煩²。



圖 5.2 指定備份目的地為 MyDB.bak 檔案

SQL Server 提供了「備份裝置」的伺服器物件，是對應實體檔案/磁帶設備的邏輯管理名稱，方便管理各資料庫的備份。備份裝置需先建立好並指定所對應的檔案或磁帶設備，之後便可透過此備份裝置進行備份。

² 若從 MSDB 系統資料庫讀取相關的資料表，仍可以知道資料庫備份的維護過程。

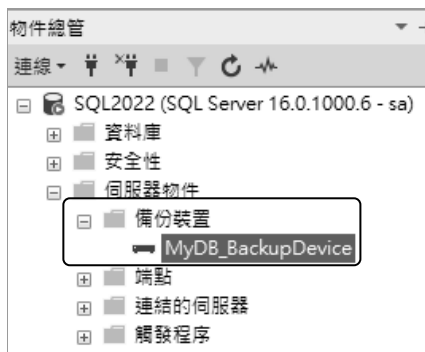


圖 5.3 透過「物件總管」/「伺服器物件」/「備份裝置」節點管理「備份裝置」

備份裝置有以下優點：

❖ 簡化備份程序

執行備份動作時，不需指定磁帶機名稱或備份檔存放位置（這些資訊在建立備份裝置時便已指定）。如圖 5.4，備份目的地可直接設定為 MyDB_BackupDevice 備份裝置。

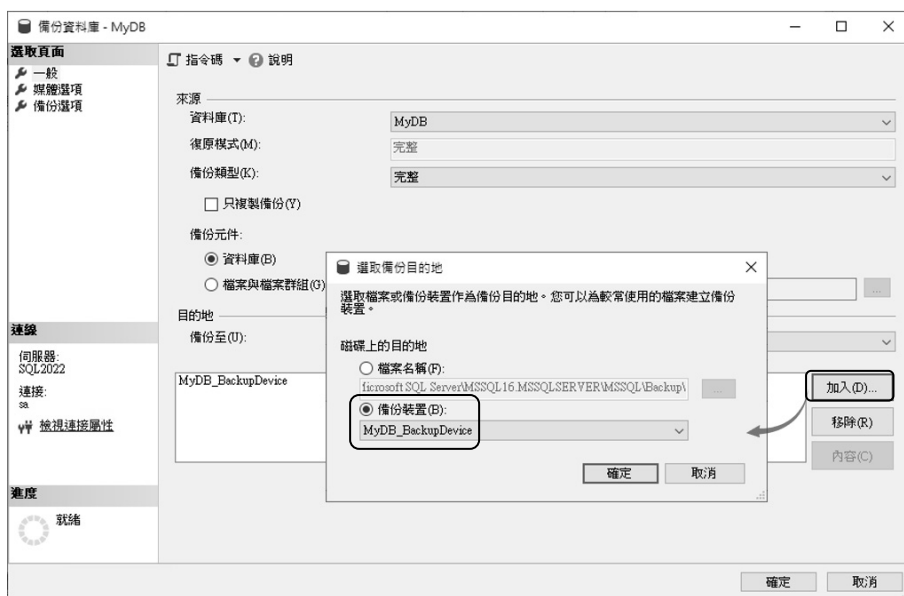


圖 5.4 將備份目的地更改為備份裝置

❖ 容易查詢備份內容

在「物件總管」內雙擊備份裝置名稱→開啟「備份裝置」視窗（參見圖 5.5）→切換至「媒體內容」頁面，便可以查看備份裝置所對應的檔案或磁帶存放了哪些資料。這些資訊包括：備份類型、伺服器名稱、資料庫名稱、備份時間、資料大小、執行備份的使用者名稱等。

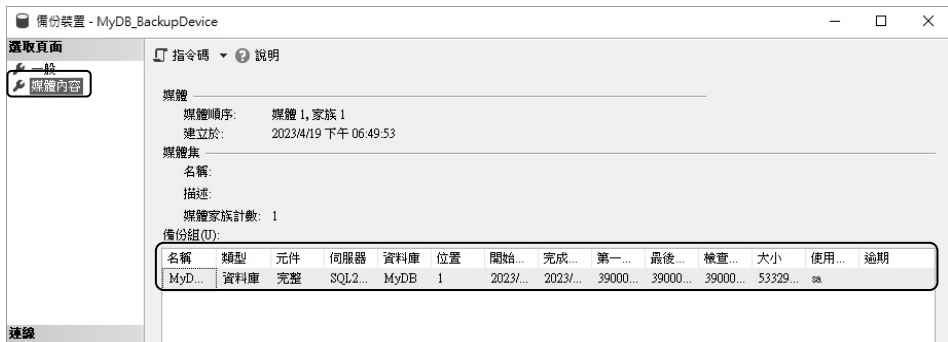


圖 5.5 檢視備份裝置的媒體內容

相較於手動管理備份裝置，備份成個別獨立的檔案另有管理上的優點，例如批次管理多個備份檔案的增刪作業。當利用資料庫維護計畫定期備份資料庫時，可搭配 SQL Agent Service 的「作業」定期產生附有時間當作檔名的多個備份檔案，而且只要透過設定就可定期刪掉過時或不需要的備份檔案。

本章的重點在於備份與還原的操作，因備份裝置與檔案相差無幾，將選用表示法較為簡短的備份裝置，但日常一般的備份流程可採用維護計畫搭配檔案。

5.2.2 建立備份裝置

使用 SQL Server Management Studio（以下簡稱 Management Studio）建立備份裝置的方法如下列步驟：

- step01 在「物件總管」內→展開「伺服器物件」節點。
- step02 選取「備份裝置」後→點選滑鼠右鍵→執行「新增備份裝置」。
- step03 在「備份裝置」視窗→指定裝置名稱與檔案存放位置（存放備份檔案的目錄需已存在，若不存在需先建立）。

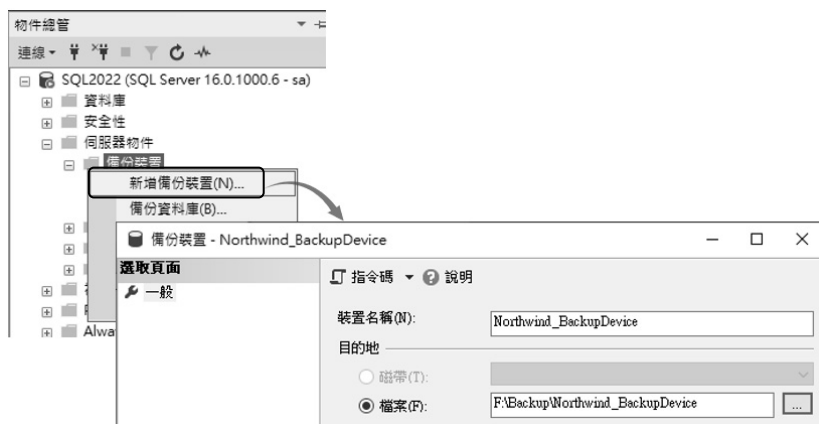


圖 5.6 建立備份裝置

TIP

若建立備份裝置並曾將資料庫備份到某個.bak 檔案，可以新增一個備份裝置再將檔案名稱設定為該.bak 檔案備份檔，便可將該檔案掛載為 SQL Server 備份裝置。

也可以使用 `sp_addumpdevice` 系統預存程序建立備份裝置，語法範例如下：

```
USE [master]
GO
EXEC master.dbo.sp_addumpdevice @devtype = N'disk',
@logicalname = N'Northwind_BackupDevice',
@physicalname = N'F:\Backup\Northwind_BackupDevice'
GO
```

5.3 SQL Server 資料庫的備份類型

SQL Server 2022 支援四種備份類型：完整、差異、交易記錄、檔案/檔案群組。

對於 SQL Server DBA 來說，前三種備份類型在實務上經常需相互搭配執行，因此本節著重介紹前三種備份類型的意義。《第 5.6 節：如何規劃備份策略》將探討備份策略的設計與規劃，檔案/檔案群組備份通常用在超大型資料庫，留待《第 5.14 節：大型資料庫的備份還原策略》再進一步介紹

5.3.1 完整備份

完整備份是最簡單的備份方式，包含下列備份資料：

- 資料檔內目前存放資料的部分（不包含未使用的空間）。包含 FileStream 檔案群組內的檔案、OLTP 記憶體最佳化資料表的資料和差異檔案組。
- 備份作業期間產生的交易記錄。

執行邏輯如圖 5.7 所示：

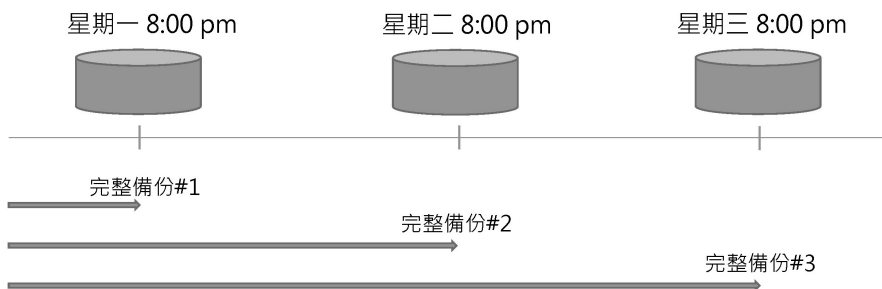


圖 5.7 在不同時間點執行完整備份

完整備份的優缺點如下表：

表 5.2 完整備份的優缺點比較

完整備份的特性或優點	完整備份缺點
<ul style="list-style-type: none"> • 作為其他備份類型的「備份/還原基準點」 • 容易執行 • 簡化還原程序 	<ul style="list-style-type: none"> • 備份時間長，需要較大量的備份媒體（檔案或磁帶）。 • 容易造成備份媒體的浪費（例如，兩次完整備份的間隔只產生幾筆交易，卻得備份整個資料庫）。 • 無法將資料庫還原到某個時間點（不支援 point-in-time recovery）。

為了說明上的方便，以下範例暫不考慮備份動作花費的時間。如圖 5.7，在星期一～星期三 8:00 pm 分別執行完整備份，若資料庫在星期四之後毀損，只能還原至星期一、二或三的 8:00 pm。

若是大型資料庫，星期一晚上執行完整備份後，縱使星期二只有幾筆資料被異動，當天晚上若再次執行完整備份，不僅是浪費了時間，對於存放備份的媒體投資成本也相對增加。

上述問題可透過 SQL Server 提供第二種備份方式：差異備份來解決。

5.3.2 差異備份

差異備份是從最後一次執行完整備份的時間點，開始備份所有資料異動情況。如圖 5.8，當星期二 8:00 pm 執行差異備份時，只會備份星期一 8:00 pm 以後異動的資料。星期三 8:00 pm 執行差異備份時，同樣也是以最後一次執行完整備份的時間點為基準，備份星期一 8:00 pm ~ 星期三當下異動的資料。也就是說，第二次的差異備份會涵蓋星期一晚上到星期三晚上的異動資料。

執行邏輯如圖 5.8：

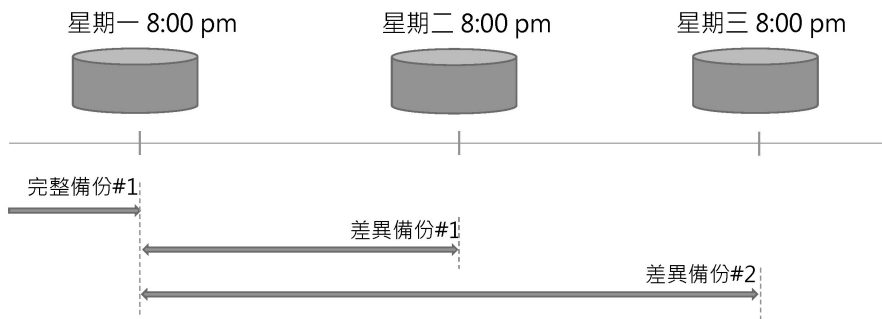


圖 5.8 差異備份以最後一次完整備份的時間點為基準

差異備份與交易記錄備份容易產生混淆，此處以一實例來分析差異備份的基本運作方式：

資料庫在星期一早上新增一筆訂單，金額為 10000 元。此筆訂單在星期二早上被更改，金額變成 12000 元；星期二下午因客戶追加訂購產品，最後訂單總金額是 15000 元。

從交易記錄的角度來看，這筆訂單在星期二共產生兩筆交易記錄：

```
更新 10000 至 12000
更新 12000 至 15000
```

如果在星期二晚上執行交易記錄備份，這兩筆交易記錄都會被備份起來。但若是在星期二晚上執行差異備份，只會備份資料庫內最後的異動結果：

```
更新 10000 至 15000
```

差異備份只備份資料庫的最後結果而不備份歷程，除節省備份時間外，也縮短了還原時間。

但 SQL Server 記錄差異備份的單位是「範圍（extent 八個實體連續頁面的集合=8K*8=64K）」，即使只變更了 1 個 byte，差異備份還是會包含該位元組的 64K「範圍」。若變動的資料量不大但很分散，那麼差異備份仍可能需要備份大量資料。

5.3.3 交易記錄備份

第三種備份方式是交易記錄備份。如圖 5.9，星期一 8:00 pm 執行完整備份，接著執行六次交易記錄備份。這樣做的優點是保留完整的資料異動歷程，如果資料庫在星期三 8:00 pm 以後毀損，可以使用交易記錄備份將資料庫復原至星期一以後的任何時間點。執行邏輯如圖 5.9：

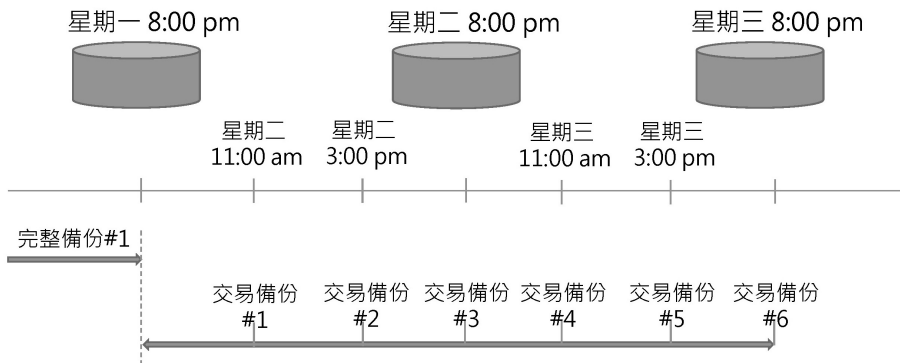


圖 5.9 在完整備份後執行交易記錄備份

交易記錄備份的特性：

- 資料庫的「復原模式」必須是完整或大量記錄。
- 保存資料庫某段期間的完整交易記錄。
- 若交易記錄皆已備份，則可以將資料庫復原至過去的任一時間點

5.5 使用 Management Studio 備份資料庫

除了 BACKUP 陳述式，也可以利用 Management Studio 圖形介面工具來備份資料庫，讓備份工作更簡潔直覺。

5.5.1 備份資料庫

使用 Management Studio 備份資料庫的方式為：

step01 在「物件總管」內選取欲備份的資料庫→按下滑鼠右鍵後執行「工作」→「備份」。

step02 在「備份資料庫」視窗的「一般」頁面，指定下列備份資訊：

- 備份類型：選項有
 - 完整
 - 差異
 - 交易記錄（當資料庫的「復原模式」設為完整時，此選項才可使用）
- 備份元件：指定要備份的是「資料庫」或「檔案與檔案群組」。
- 目的地：可指定備份媒體是檔案、備份裝置或 URL（Windows Azure 儲存體）。




圖 5.10 「備份資料庫」視窗的「一般」頁面

5.5.2 備份某個檔案或檔案群組

若資料庫的復原模式設定為「完整」或「大量記錄」，在「備份資料庫」視窗內便可以針對檔案或檔案群組進行備份：

step01 開啟「備份資料庫」視窗。

step02 在「一般」頁面內，將「備份元件」設定為「檔案與檔案群組」或是按一下「檔案與檔案群組」的 ，便會跳出如圖 5.11 的「選取檔案與檔案群組」視窗：

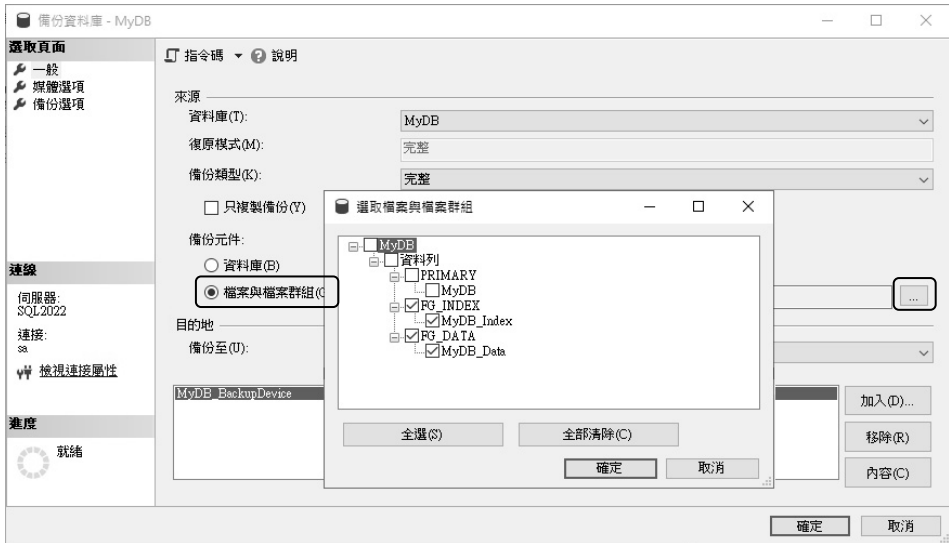


圖 5.11 備份元件選取檔案或檔案群組

step 03 在「選取檔案與檔案群組」視窗內，選取要備份的檔案群組或檔案後按下「確定」，被選取的檔案或檔案群組會顯示於「檔案與檔案群組」的欄位裡。如選擇多個檔案或檔案群組，則檔案或檔案群組間會以逗號隔開，如圖 5.12：



圖 5.12 備份多個檔案或檔案群組

5.5.3 備份交易記錄

使用 Management Studio 備份交易記錄的方式與備份資料庫完全相同，差別僅在於「備份類型」需設定為交易記錄：



圖 5.13 備份交易記錄時，需將「備份類型」更改為「交易記錄」

5.5.4 指定備份選項

「備份資料庫」視窗的「媒體選項」及「備份選項」頁面內提供多種進階設定。這些設定會對應到 BACKUP DATABASE (或 BACKUP LOG) 陳述式的選項：

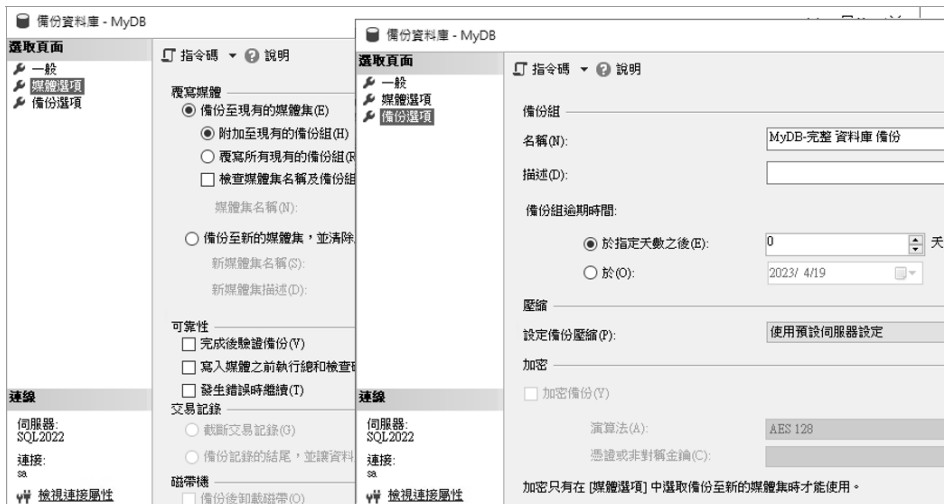


圖 5.14 備份資料庫媒體選項及備份選項

頁面上各選項與 BACKUP DATABASE 陳述式的對應關係如下表：

表 5.5 各種備份選項所對應的 BACKUP DATABASE/LOG 控制選項

媒體選項	對應的 BACKUP DATABASE 陳述式選項
複寫媒體	
備份至現有的媒體集	
附加至現有的備份組	NOINIT
複寫所有現有的備份組	INIT
檢查媒體集名稱及備份組是否逾期	NOSKIP
媒體集名稱	MEDIANAME
備份至新的媒體集，並清除所有現有的備份組：FORMAT	
新媒體集名稱	MEDIANAME
新媒體集描述	MEDIADESCRIPTION
可靠性	
完成後驗證備份	備份完整後執行 RESTORE VERIFYONLY，確認備份資料是否有效。
寫入媒體之前執行總和檢查碼	CHECKSUM
發生錯誤時繼續	CONTINUE_AFTER_ERROR
交易記錄	
對應的 BACKUP LOG 陳述式選項	
截斷交易記錄	備份後不清除交易記錄。 NO_TRUNCATE
備份記錄的結尾，並讓資料庫保持在還原狀態	備份後將資料庫保留在 RESTORING 狀態，讓其他人無法對資料庫增刪修查。 NORECOVERY
磁帶機	
備份後卸載磁帶	UNLOAD
卸載之前倒轉磁帶	REWIND

備份選項	對應的 BACKUP DATABASE 陳述式控制選項
備份組	
名稱	NAME
描述	DESCRIPTION
備份組逾期時間	EXPIREDATE 與 RETAIN_DAYS
壓縮	
設定備份壓縮	COMPRESSION NO_COMPRESSION
加密：ENCRYPTION	
演算法	ALGORITHM
憑證或非對稱金鑰	SERVER CERTIFICATE

5.6 如何規劃備份策略

備份策略是管理 SQL Server 時，需謹慎評估規劃的重點之一，完善的備份策略除了可延長平均運作時間 MTTF (**MeanTime To Failure**)，更可以縮短平均修復時間 MTTR (**MeanTime To Recovery**)。

以圖 5.15 說明 MTTF 及 MTTR：

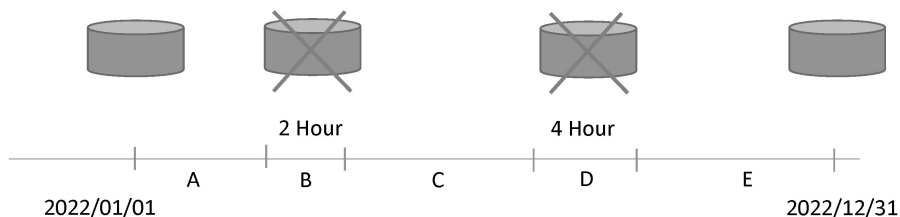


圖 5.15 某資料庫的運轉時間 (A、C、E) 與停機時間 (B、D)

❖ MTTR (平均修復時間)

假設一年 365 天當中因為資料庫毀損需停機兩次：第一次修復耗費 2 小時，第二次耗費 4 小時

=> 一年內資料庫的 MTTR 即為 3 小時：

$$\frac{\text{第一次耗費2小時} + \text{第二次耗費4小時}}{\text{一年內曾修復資料庫2次}} = \text{平均修復時間為3小時}$$

❖ MTF (平均運作時間)

圖 5.15 裡，A、C、E 三個時段資料庫均正常運作，但 B、D 時段資料庫因毀損而停機。

=> 以一年 365 天，一天 24 小時計算，扣掉 6 小時的資料庫毀損修復時間，可計算出一年的 MTF 為 2918 小時：

$$\frac{(24\text{小時} * 365\text{天}) - 6\text{小時毀損修復時間}}{\text{資料庫正常運作的時段}} = \frac{8754}{3} = 2918\text{小時}$$

這裡的 2918 小時（相當於 121.58 天）代表 A、C、E 三個時段的平均值，也就是「平均運作時間」，代表 SQL Server 每次啟動後平均可以運作 121 天。圖 5.15 中，meantime to failure 是資料庫從正常運作到毀損停機的時間間隔，分別是 A、C、E 三個區間。根據 meantime to failure 的定義，這三個時段的平均值即是資料庫的平均運作時間。

因此，要提升 SQL Server 資料庫的高可用性，須盡可能縮短資料庫的平均修復時間（MTTR）並延長資料庫的運作時間（MTTF）。

5.6.1 完整備份 + 交易記錄備份

完整備份因程序簡單，適用於小型資料庫。缺點是無法還原至特定時間點，若要克服此問題，可在完整備份後搭配一連串的交易記錄備份，如圖 5.16：

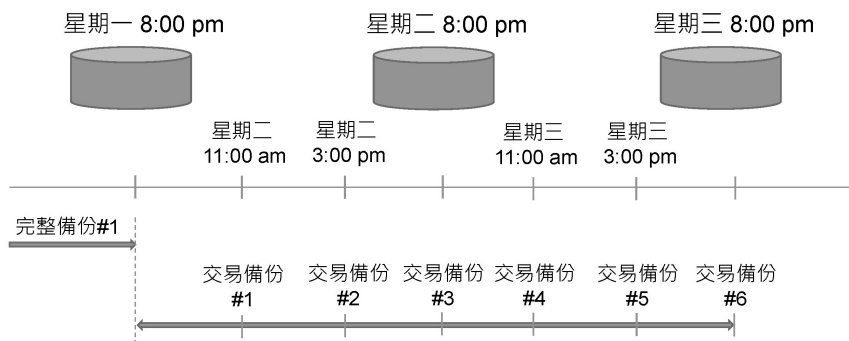


圖 5.16 完整備份 + 交易記錄備份

雖然這種備份方式可以還原到某一特定時間點，但一連串的交易記錄備份很可能會拉長資料庫的還原時間。假設資料庫在星期三上午 11:00 左右毀損，如果要將資料庫還原到星期三上午 11:00 的時間點，共需要五個備份資料檔：完整備份#1 以及交易記錄備份#1~交易記錄備份#4，當然也需執行五次還原動作。

如果將備份策略稍作調整，分別在星期二與星期三晚上增加一次完整備份（如圖 5.17），那麼要還原資料庫到星期三早上 11:00 只需要兩項備份資料：完整備份#2 與交易記錄備份#4。理論上，利用這兩項備份資料來還原資料庫的耗時較短，並且還原程序也較簡易。同理，如需將資料庫還原到星期三晚上某時間，可利用完整備份#3 的資料進行還原，再搭配之後的交易記錄備份即可。

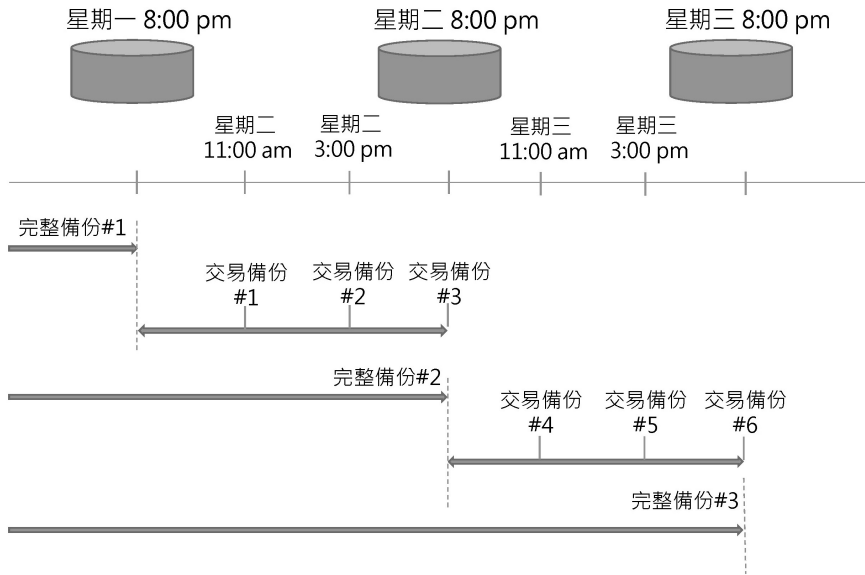


圖 5.17 完整備份與交易記錄備份的另一種搭配方式

僅有完整備份與交易記錄備份，也許還是無法應付各種資料庫可能產生的備份問題。比方完整備份可能造成備份媒體的浪費，或增加備份時間。所以，圖 5.17 裡的備份策略仍可再作改善。

5.6.2 完整備份 + 差異備份 + 交易記錄備份

圖 5.18 為改良後的備份策略：將完整備份#2 改成差異備份#1。在此備份策略下，要將資料庫還原至星期三早上 11:00 只需要三項備份資料：完整備份#1，差異備份#1 與交易記錄備份#4。雖然還原程序包含三個階段，但是差異備份#1 的備份時間並不會太長⁴（與圖 5.17 的完整備份#2 相比較），而且也可節省備份媒體空間。

⁴ 更新的資料雖然不多，但若大量散布在不同的「範圍（extent）」，仍有可能導致差異備份的量變大。

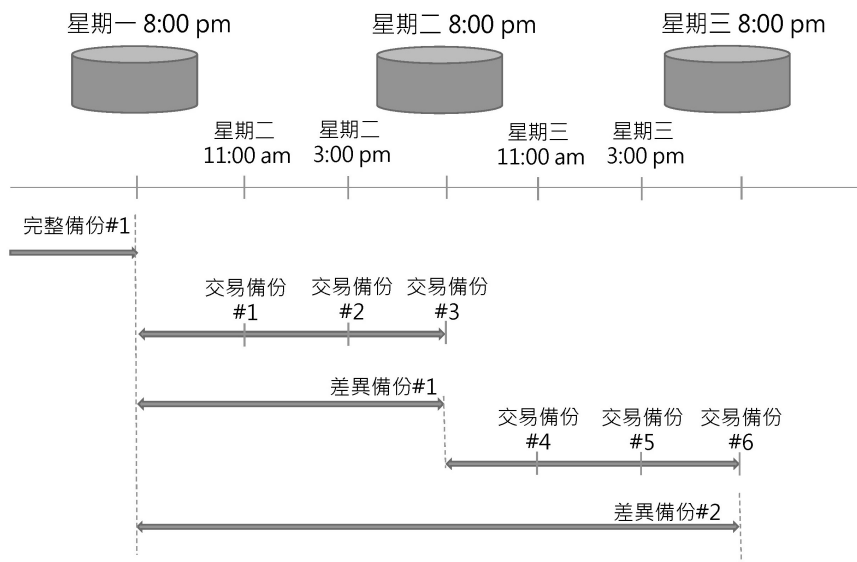


圖 5.18 完整備份 + 差異備份 + 交易記錄備份

以上介紹了各種備份策略如何相互搭配，以及備份執行時間的選擇，但實際需根據資料庫的運作情況才能評估出最佳備份策略。

5.7 使用兩個以上備份裝置

備份中大型資料庫（比方備份壓縮後資料量 > 1TB）時，常可會碰到一些狀況，例如：

- 巨大的資料量導致備份時間過長。
- 現有的儲存設備不足以應付備份需求：比方硬碟容量僅有 2TB，但資料庫備份壓縮後 1TB，需要保留三次週期的資料。

此時，可以同時使用兩個以上備份裝置，運用平行處理來備份資料庫以提升備份效率。理論上，兩台磁碟機同時運作，備份時間應可縮短一半，且也可以使用多個小容量的磁碟來備份大型資料庫。



圖 5.19 同時指定兩個備份裝置

圖 5.19 將 MyDB 資料庫同時備份到網路共享上的兩個檔案，模擬多裝置的備份。完成後，這兩個備份檔案個別資料量近似於單一檔案備份的一半，也就是說 SQL Server 會平均寫入備份資料

名稱	修改日期	類型	大小
MyDB_Backup.bak	2022/10/3 下午 05:14	BAK 檔案	6,268 KB
MyDB_backup1.bak	2022/10/3 下午 05:14	BAK 檔案	4,288 KB
MyDB_backup2.bak	2022/10/3 下午 05:14	BAK 檔案	4,152 KB

圖 5.20 多個備份裝置平均寫入對比單一備份裝置

在圖 5.20 中，MyDB_Backup1.bak 和 MyDB_Backup2.bak 是將大約將各 50%的資料庫備份同時寫入到兩個備份檔，而 MyDB_Backup.bak 則是只使用一個檔案完成整個資料庫的備份。

如果某兩個備份裝置（稱為媒體集）曾經同時執行過某種備份動作，之後即無法單獨使用其中一個備份裝置來備份資料庫（除非設定

FORMAT 選項與 INIT 選項)。如圖 5.21，單獨使用 MyDB_Backup1.bak 檔案來備份 MyDB 資料庫會回應錯誤訊息：



圖 5.21 備份時單獨使用媒體集其中一個備份裝置，會回應錯誤訊息

5.8 備份至 URL

SQL Server 為了降低儲存體成本及提高可用性，在 SQL Server 2012 版本已提供將資料庫備份至 Azure 儲存體 (Storage) 的功能，讓使用者能輕鬆地建立資料庫異地備份。2016 版強化了備份至 URL 的設計，除原有的使用金鑰建立存取儲存體帳戶的認證之外，在安全性方面還新增了共用存取簽章功能。另外，以往備份至 URL 時備份媒體是以分頁 (Page) Blob 格式儲存，2016 版新增區塊 (Block) Blob 格式，且支援備份至多個 Blob，SQL Server 2022 更新增將備份寫入 AWS S3 相容物件儲存體的

能力，使用 REST API 對 S3 連接器的支援，並擴充了 BACKUP/RESTORE TO/FROM URL 語法。

要資料庫備份至 Azure 「儲存體」/「容器」內，須先在 Azure 訂閱內建立儲存體帳戶（這個帳戶如同系統管理帳戶，擁有建立容器及物件的系統管理權限），之後 SQL Server 便可以使用儲存體帳戶名稱及金鑰或是容器的共用存取簽章，透過 T-SQL/SMO（SQL Server Management Object）/PowerShell Cmdlet/Management Studio 發送 Backup 或 Restore 陳述式，讀寫區塊 Blob 或分頁 Blob 至 Azure 儲存體⁵。（若用金鑰認證方式則會使用分頁 Blob；而共用存取簽章的認證方式則會使用區塊 Blob）

SQL 2016 版的備份至 URL 功能，除了原有的使用 WITH CREDENTIAL 選項，另外新增了「共用存取簽章」並建立對應「認證」的方式。接下來介紹如何使用 T-SQL 及 SQL Server 管理工具將資料庫備份到 Azure 儲存體。

5.8.1 使用 T-SQL 搭配 WITH CREDENTIAL 選項

首先請到 Azure 入口網站（或使用 Powershell）產生儲存體帳戶的「存取金鑰」，複製 key1 或 key2 的機碼以便在 T-SQL 陳述式中使用：

⁵ 儲存體帳戶說明以及建立方法請參考線上文件：<https://learn.microsoft.com/zh-tw/azure/storage/common/storage-account-create?tabs=azure-portal>



圖 5.22 從 Azure 入口網站產生儲存體帳戶的存取金鑰

接著執行以下的 T-SQL 陳述式，使用 Azure 儲存體帳戶「存取金鑰」的 key1 或 key2 機碼建立認證，搭配 WITH CREDENTIAL 陳述式即可以執行 BACKUP DATABASE TO URL⁶，將資料庫備份至 Azure 儲存體容器內。

```
-- 建立認證
CREATE CREDENTIAL [AzureStorage]
WITH IDENTITY = 'ellie', --儲存體帳戶名稱
SECRET = 'uc54Pcz3BohUVDGuTmG4SEQFwZ2MT2jN...' --可使用 key1 機碼或 key2 機碼
GO

--使用儲存體帳戶的身分識別和存取金鑰備份資料庫至 URL
BACKUP DATABASE AdventureWorks
TO URL = N' https://ellie.blob.core.windows.net/db-backup/AdventureWorks.bak' --
儲存體容器的 URL 及備份媒體名稱。
```

⁶ SQL Server 備份裝置名稱上限為 259 個字元，BACKUP TO URL 會使用 36 個字元指定 URL 必要字元，保留 223 個字元供儲存體帳戶、容器和 Blob 名稱共用。（'https://儲存體帳戶.blob.core.windows.net/容器/備份媒體名稱.bak'）

```
WITH CREDENTIAL='AzureStorage', --認證名稱
COMPRESSION
```

5.8.2 使用 TSQL 搭配「共用存取簽章」

在 Azure 入口網站點選儲存體帳戶的「共用存取簽章」，勾選「允許的資源類型」並輸入簽章有效的開始及到期日期/時間後，按下右下角的「產生 SAS 與連接字串」：



圖 5.23 從 Azure 入口網站設產生儲存體帳戶的共用存取簽章

複製 SAS 權杖內容後回到 SSMS，建立 SQL Server 內對應到共用存取簽章的認證，然後執行 BACKUP DATABASE TO URL 的陳述式，即可將資料庫備份至 Azure 儲存體：

```
--使用共用存取簽章建立認證
CREATE CREDENTIAL [https://ellie.blob.core.windows.net/db-backup] --儲存體容器的
URL
```

```
WITH IDENTITY = 'SHARED ACCESS SIGNATURE',
SECRET = 'sv=2021-12-02&ss=bfqt&srt=sco&sp=rwldacupiytfx&se=2023-04-20T20:20:36Z&st=2023-04-20T12:20:36Z&spr=https&sig=RmRMm6QBp7dQiFTqcn7up6x%2FYdfu6DzjUTKULStmSNo%3D' --SAS 權杖 (移除最前面的?)
```

```
--使用共用存取簽章備份資料庫至 URL
BACKUP DATABASE [AdventureWorks]
TO URL = 'https://ellie.blob.core.windows.net/db-backup/AdventureWorks_2.bak' --
儲存體容器的 URL 及備份媒體名稱
WITH COMPRESSION, STATS = 5
GO
```

5.8.3 使用 SQL Server 管理工具

若要使用 Management Studio 將資料庫備份至 URL，可依照以下步驟：

step01 在「物件總管」內選取要備份的資料庫，按下滑鼠右鍵後執行「工作」→「備份」。

step02 在「備份資料庫」視窗的「一般」頁面，目的地選擇「備份至 URL」，然後按下右下角的「新增」按鈕，此時會開啟「選取備份目的地」視窗：

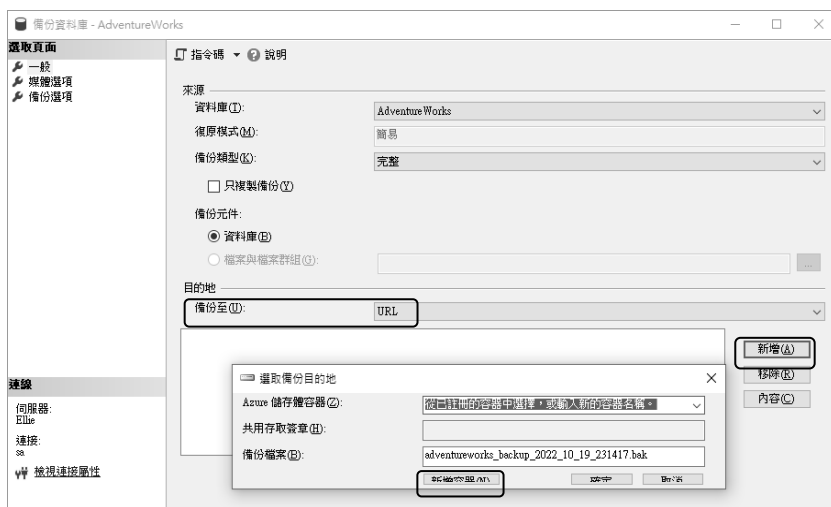


圖 5.24 使用 Management Studio 將資料庫備份至 URL

若已經有使用共同存取簽章建立的認證，可以在「Azure 儲存容器」下拉選單中直接選取，然後按下「確定」將資料庫備份至 URL。或是點選「新增容器」開啟「連接至 Microsoft 訂用帳戶」視窗，登入 Microsoft Azure，選取訂用帳戶/儲存體帳戶/Blob 容器並設定簽章的到期日，接著按下「建立認證」鈕，即會產生共用存取簽章，並建立對應的認證以供當次資料庫備份使用：



圖 5.25 使用 Management Studio 連接至 Microsoft Azure 新增容器並建立共用存取簽章及相對應的認證

接著在「選取備份目的地」視窗選取 Azure 儲存體容器後，按下「確定」回到「備份資料庫」視窗，再按下「確定」即可將資料庫備份至 Azure 儲存體：

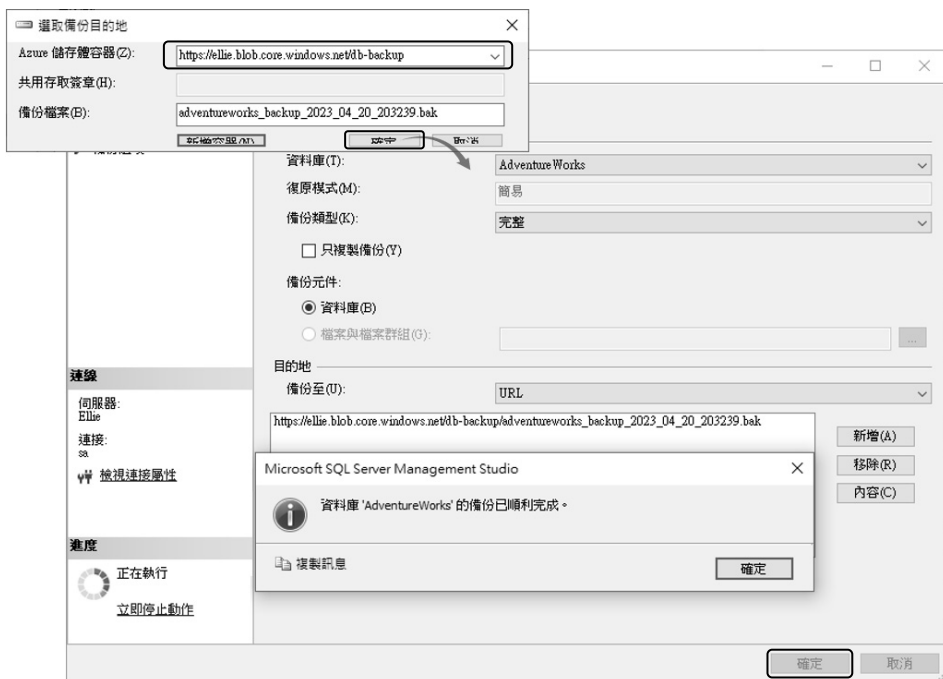


圖 5.26 使用 Management Studio 將資料庫備份至 Azure 儲存體的完成畫面

若要檢視放在 Azure 儲存體的備份資料，可以使用 Management Studio → 「物件總管」 → 「連接」 → 「Azure 儲存體」 → 在開啟的「連線至 Microsoft 訂用帳戶」視窗登入 Microsoft Azure，並選取儲存體帳戶及 Blob 容器 → SQL Server 的物件總管便會列出儲存體帳戶裡的容器及 Blob 資訊；另一個方法是從 Microsoft Azure 管理入口網站 (<https://portal.azure.com/>) 查看。

程作業，只需再次執行該預存程序並將 `new_state` 設定改為 1 即可。只是此預存程序會暫停執行個體裡所有資料庫的受控備份，無法針對個別資料庫設定。

```
--暫停執行受管理的備份
EXEC msdb.managed_backup.sp_backup_master_switch
@new_state=0
GO

--繼續執行受管理的備份
EXEC msdb.managed_backup.sp_backup_master_switch
@new_state=1
```

若要直接停用受管理的備份，可以執行預存程序 `msdb.managed_backup.sp_backup_config_basic`，指定資料庫名稱並將 `enable_backup` 設定為 0：

```
--停用受管理的備份
EXEC msdb.managed_backup.sp_backup_config_basic
    @database_name = 'AdventureWorks',
    @enable_backup = 0
```

其他更詳細受控備份的說明及設定進階選項請參考線上說明文件：

<https://learn.microsoft.com/zh-tw/sql/relational-databases/backup-restore/configure-advanced-options-for-sql-server-managed-backup-to-microsoft-azure?redirectedfrom=MSDN&view=sql-server-ver16>

本章資料庫備份介紹到此，接下來將說明資料庫的還原工作。

5.10 還原資料庫的前置工作

在大部分的狀況下若是資料庫毀損（例如，資料表被使用者刪除或是資料被誤改），通常只要執行還原程序將備份還原成另外一個暫存的資料庫，再選擇正確的資料覆蓋回原誤改內容就可以修復。但若碰到 SQL Server 或 Windows 作業系統毀損，甚至伺服器的硬體設備故障，可能無法單純憑藉資料庫還原程序來修復，而必須重裝軟/硬體後才可繼續還原資料庫。

資料庫若毀損，救援資料庫的第一件事並非立刻執行還原程序，而是應先思考幾個問題：

- 導致資料庫毀損的主因？
- 資料庫的「復原模式」若非簡單模式，線上系統是否還允許連線以備份結尾交易記錄（Tail-Log Backups）？
- 備份策略是否涵蓋所需要的備份資料？
- 備份檔是否有效？
- 還原若發生錯誤是否要繼續還原？
- 需要將資料庫還原到什麼時間點？

以上這些問題，可由下列程序尋求答案：

- 檢視備份策略，決定還原時間點
- 檢查備份裝置的媒體內容，確認備份資料所在位置
- 執行 RESTORE VERIFYONLY 陳述式，檢查備份資料是否有效

5.10.1 檢視備份策略 & 決定還原時間點

為什麼需要在還原資料庫前先檢視備份策略呢？若不清楚在什麼時間點執行了何種備份方式，將很難有效率地執行還原作業。因此，在執行還原作業前應先決定：

- 還原時間點
- 還原作業所需要的備份資料

以圖 5.32 的備份策略為例，假設資料庫在星期三下午 3:00 以後毀損，可以將還原時間點定為星期三下午 3:00。如果要將資料庫還原至這個時間點，將會需要以下的備份資料：

- 星期一晚上 8:00 的備份資料（完整備份#1）。
- 星期一晚上 8:00 以後到星期三下午 3:00 之間備份的交易記錄（交易記錄備份#1～#5 五個交易記錄備份檔）。
- 若損毀當時還可以取得結尾交易記錄備份，則最後需還原結尾記錄備份。

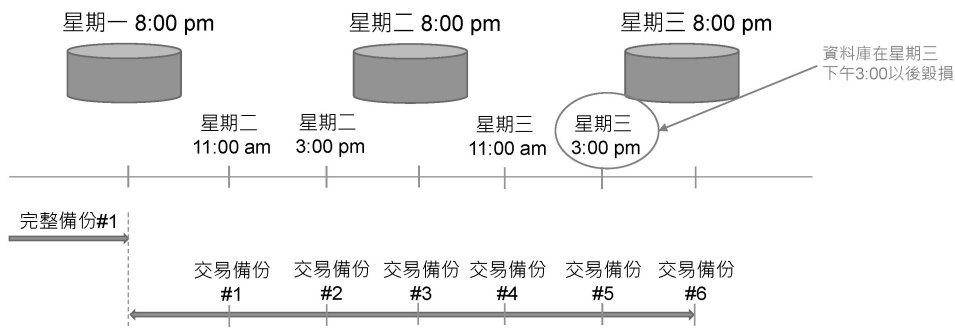


圖 5.32 第一種備份策略

針對同一個還原時間點，如果備份策略不同，執行還原作業所需的備份資料也會不同。

以第二種備份策略為例（圖 5.33），如果要將資料庫還原到星期三下午 3:00，將需要以下的備份資料：

- 星期二晚上 8:00 的備份資料（完整備份#2）。
- 星期二晚上 8:00 以後到星期三下午 3:00 之間備份的交易記錄（交易記錄備份#4 與#5 兩個交易記錄備份檔）。
- 若損毀當時還可以取得結尾交易記錄備份，則最後需還原結尾記錄備份。

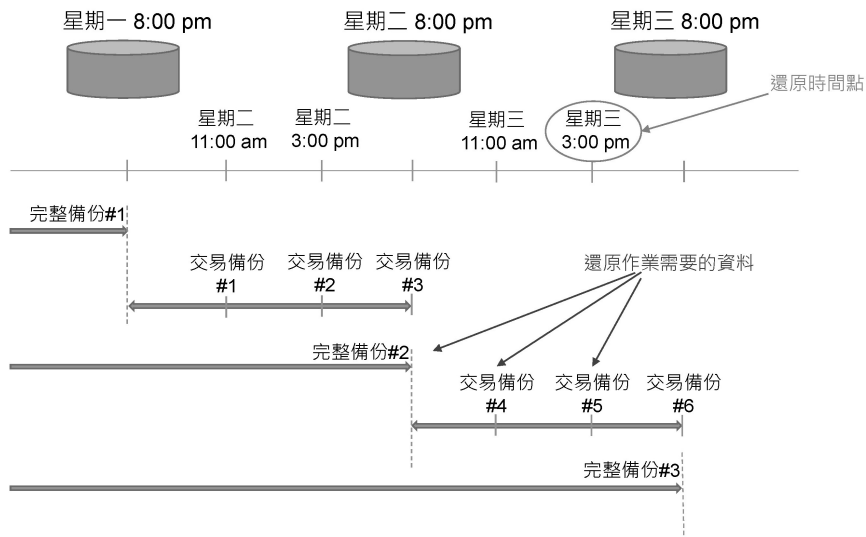


圖 5.33 第二種備份策略

再看圖 5.34 的第三種備份策略。若要將資料庫還原至星期三下午 3:00，最快的方式是利用下列備份資料：

- 星期一晚上 8:00 的備份資料（完整備份#1）。
- 星期二晚上 8:00 的備份資料（差異備份#1）。
- 星期二晚上 8:00 以後到星期三下午 3:00 之間備份的交易記錄（交易記錄備份#4 與#5 兩個交易記錄備份檔）。

- 若損毀當時還可以取得結尾交易記錄備份，則最後需還原結尾記錄備份。

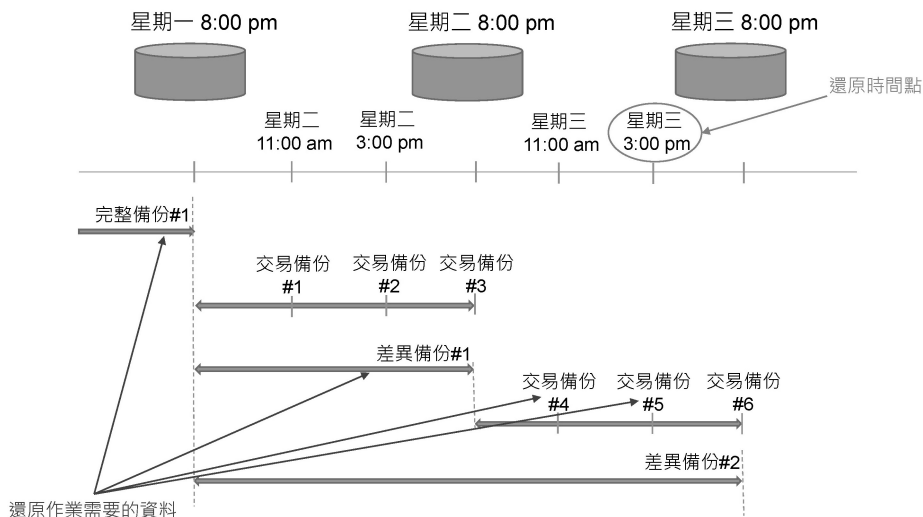


圖 5.34 第三種備份策略

透過以上三種備份策略應可瞭解，不同的備份策略除了影響還原作業的執行流程，也會影響還原作業的執行效率。

5.10.2 確認備份資料的位置

決定還原時間點後，接著必須確認備份資料存放在哪些檔案或備份裝置：

- 在「物件總管」→以滑鼠左鍵雙擊備份裝置名稱→開啟「備份裝置」視窗。在「媒體內容」頁面檢視備份資料（參見圖 5.35）。

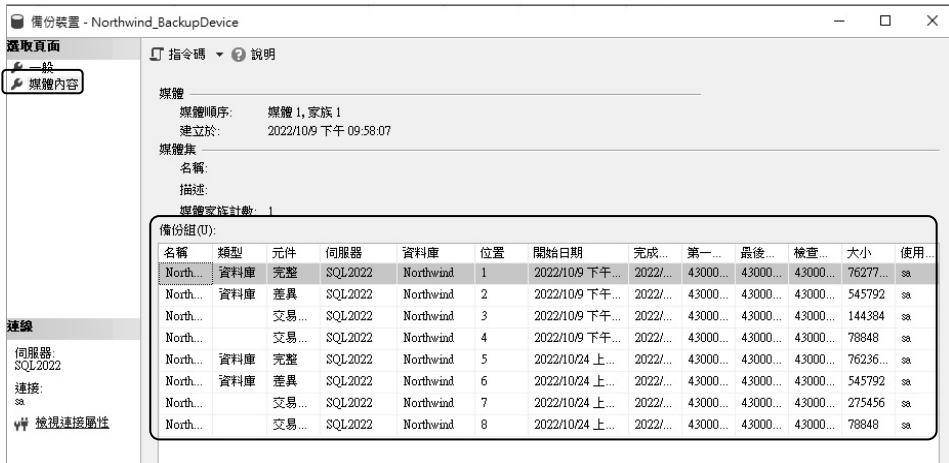


圖 5.35 「備份裝置」視窗的「媒體內容」頁面

也可以執行 RESTORE HEADERONLY、RESTORE FILELISTONLY 陳述式，查詢備份裝置的內容（參見圖 5.36）：

```
RESTORE HEADERONLY FROM [Northwind_BackupDevice];
RESTORE FILELISTONLY FROM [Northwind_BackupDevice];
```

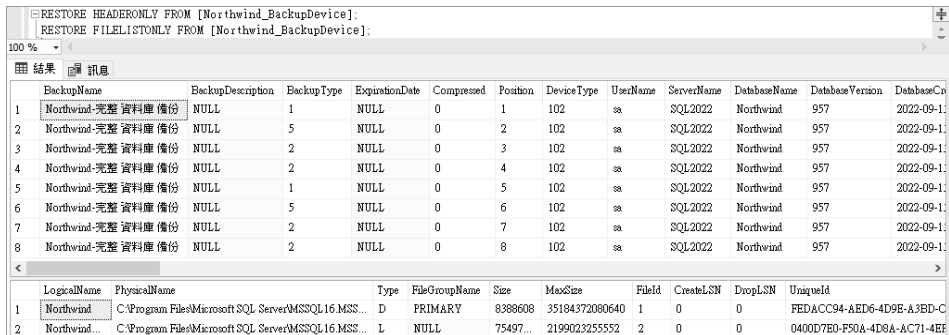


圖 5.36 執行 RESTORE HEADERONLY、RESTORE FILELISTONLY 陳述式

5.10.3 確認備份資料是否有效

執行還原作業以前可先執行 RESTORE VERIFYONLY 陳述式驗證備份資料是否有效，以避免備份資料毀損（例如磁帶發霉或硬碟磁區損毀）或不完整，導致還原作業失敗。

本章《第 5.7 節：使用兩個以上備份裝置》，曾經將 MyDB 資料庫同時備份到 MyDB_backup1 及 MyDB_backup2 備份裝置，針對此種備份策略，執行還原作業時也必須同時使用兩個備份裝置。在圖 5.37 左側 SQL 查詢視窗中，以 RESTORE VERIFYONLY 驗證 Northwind_1 備份裝置的內容時，SQL Server 會回應下列訊息：

```
訊息 3132，層級 16，狀態 1，行 1
媒體集有 2 個媒體家族，但是僅提供 1。必須提供所有成員。
訊息 3013，層級 16，狀態 1，行 1
VERIFY DATABASE 正在異常結束。
```

由此可知，MyDB 資料庫的還原作業無法只透過 MyDB_backup1 或 MyDB_backup2 單一備份裝置達成。

圖 5.37 右側的 SQL 查詢視窗則是同時驗證 MyDB_backup1 與 MyDB_backup2 備份裝置的內容，STATS 選項用以指定完成多少百分比（例如 25%）需回報一則訊息。

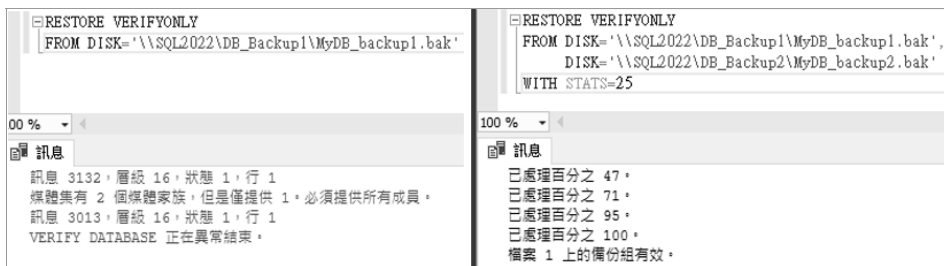


圖 5.37 執行 RESTORE VERIFYONLY 陳述式驗證備份資料

5.19 結語

備份資料庫是 SQL Server 重要的管理工作之一，DBA 必須熟悉各種備份類型的意義與執行時機，也要視資料庫實際運作情況定義完善的備份策略，並經常演練還原資料庫流程，確認備份裝置與資料正確性及還原流程所需的時間，當資料庫不幸毀損時才能在最短時間內復原。本章除了介紹還原資料庫的前置工作，也描述了各種還原資料庫的技巧，包括：

- 決定還原時間點。
- 確認備份資料是否有效。
- 還原資料庫的三種復原選項（RECOVERY、NORECOVERY 與 STANDBY）。
- 還原資料庫實例探討。

另外，master 是重要的系統資料庫，它不僅存放登入帳號/權限、所有資料庫的實體結構資訊，也記錄 SQL Server 執行個體的系統組態。萬一 master 資料庫毀損，但之前卻沒有備份此資料庫，那肯定是 DBA 的災難！為了防範未然，建議妥善備份 master 資料庫，並熟悉 master 資料庫的重建或還原方式，才能在問題時發生後及時修復。

而另一個系統資料庫 msdb 也對輔助的服務很重要，例如：SQL Server Agent Job、資料收集、維護計畫...等。可將 msdb 視同一般資料庫進行備份、還原維護作業，以避免 msdb 損毀時輔助的服務無法正確執行。

建議將備份作業建立自動化管理作業週期性地自動執行，並可搭配「通知」在作業成功/失敗或完成時發送電子郵件通知資料庫相關管理人員。