

# 認識系統資料庫與系統物件

# A

## APPENDIX

SQL Server 屬於「關聯式資料庫管理系統（RDBMS）」，RDBMS 將資料間的關係以資料表的行列形式表達，也以表格的形式儲存數據，以方便查詢。因此，在 SQL Server 的環境裡每個資料庫都是由多個資料表組成。這些資料表裡面除了存放使用者的資料，也包含了其他物件定義，例如：檢視、索引、條件約束（Constraint）、預存程序、使用者自訂函數，以及與資料活動一起執行的觸發程序...等，其他還有關於安全的使用者、角色、金鑰、憑證、原則...等。

SQL Server 統一將上述相關物件的定義存放在系統資料庫或使用者資料庫的系統資料表中，這些資料統稱為中繼資料（metadata）。此外，系統資料庫還存放資料庫活動所產生的相關訊息，例如：記錄使用者資料庫存放位置、備份／還原資訊、作業執行歷程，以及資料庫維護計畫內容...等。

透過 Management Studio 的「物件總管」（如圖 A.1），可看到 SQL Server 將資料庫分為下列兩類：

- **系統資料庫：**存放使用者資料庫的中繼資料、使用者登入帳號資訊、SQL Server 自動化作業、系統執行運算時的暫存資料、乃至於複寫（Replication）、高可用性、連結伺服器...等。

- 使用者資料庫：使用者可自行建立的資料庫，通常與特定的主題或業務有關，像是財務、人事、製造、庫存...等資訊。使用者資料庫的系統資料表內會存放自行建立的物件定義，如資料表名稱、欄位名稱、索引定義等，並且也參考了大量的系統物件，如系統檢視表、系數函數、系統預存程序...等，讓使用者可以進行各種管理工作。

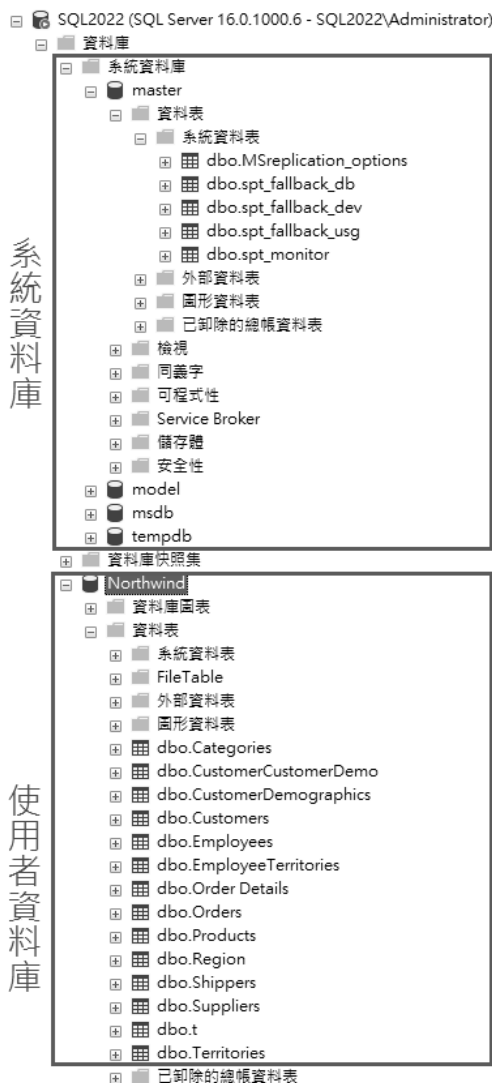


圖 A.1 系統資料庫與使用者資料庫所包含的系統資料表

在正式介紹如何管理資料庫與資料表之前，應先了解這些系統資料庫與系統物件的相互關係，才能有效操控或避免誤刪／改而導致系統異常。

## A.1 系統資料庫

SQL Server 在安裝時會自動建置數個系統資料庫，每個資料庫均有其特殊用途。這些系統資料庫包括：

- master：記錄 SQL Server 執行個體的所有系統層級資訊，如使用者資料庫的位置、使用者登入帳號、連結的伺服器設定...等。
- msdb：各種 SQL Server 輔助服務放置資訊的地方，例如 SQL Server Agent 用於作業/排程、警示和操作員的設定與執行歷程、以封裝為單位的 SSIS 部署、交易記錄傳送（log shipping）的資訊、DBMail、原則管理、使用者資料庫的備份歷程...等。
- model：用來當作 SQL Server 執行個體上建立使用者自訂資料庫或 tempdb 系統資料庫的範本。對 model 資料庫進行的修改（例如：資料庫大小、定序、復原模式和其他資料庫選項）會套用到使用者之後建立的資料庫。
- tempdb：保存暫存物件或中繼結果集的工作空間。每當 SQL Server 執行個體啟動時，就會重新建立此資料庫。也就是說，重新啟動 SQL Server 後，tempdb 裡先前存放的任何物件與資料都會被永久刪除。

- Resource<sup>1</sup>：是一個唯讀的資料庫，包含 SQL Server 的系統物件，如系統檢視表、系統預存程序和函數等。系統物件實際上會保存在 Resource 資料庫中，但映射到每個資料庫的 sys 結構描述中。此資料庫可以簡化升級作業。
- Distribution：當伺服器執行個體要在複寫架構下擔任散發者（Distributor）角色，會建立 Distribution 系統資料庫，以存放複寫的定義與各種複寫運作的狀況。而實際的資料庫名稱會在「設定散發精靈」的步驟中賦予。

除了 Distribution 系統資料庫，還有一些因相關功能啟動後才會建立的系統資料庫，如 Reporting Services 所需的系統資料庫：ReportServer 及 ReportServerTempdb 資料庫、SQL Server 2012 版本後專為 SSIS 服務建立的 SSIS 目錄資料庫（SSISDB）（若以 SSIS 專案為單位部署封裝，將會放在此 SSIS 目錄資料庫），以及 Data Quality Services、外部資料的 PolyBase 查詢服務所需的系統資料庫...等。

上述的資料庫和記錄檔的預設位置如下：

```
Program Files\Microsoft SQL Server\MSSQL16.<InstanceName>\MSSQL\DATA
```

但 Resource 資料庫檔案藏於程式所在的目錄下，實體檔案名稱為 mssqlsystemresource.mdf 及 mssqlsystemresource.ldf。

```
Program Files\Microsoft SQL Server\MSSQL16.<InstanceName>\MSSQL\Binn
```

<sup>1</sup>

Resource 資料庫不包含使用者資料或使用者中繼資料，它的實體檔案名稱為 mssqlsystemresource.mdf 和 mssqlsystemresource.ldf 位於 Binn 的目錄中，可將它視為二進位（.EXE）檔案而非資料庫檔案，SQL Server 無法備份它。Resource 資料庫讓更新 SQL Server 的程序變得更快且更容易，因舊版 SQL Server 更新 Service Pack 或 Hotfix 時需要卸除和建立系統物件，而 Resource 資料庫檔案包含所有系統物件，因此只要將單一 Resource 資料庫檔案複製到本機伺服器，即可完成升級。

如果在安裝 SQL Server 時有變更預設目錄，則這些系統資料庫實體檔案的位置也會跟著改變。

接下來，將介紹各系統資料庫的用途。

### A.1.1 master 資料庫

在 SQL Server 中，master 是最重要的系統資料庫，可以將它視為 SQL Server 的心臟，其用途是存放 SQL Server 整體的系統資訊：

- SQL Server 伺服器組態。
- 使用者登入帳號、角色。
- 端點組態（endpoint）。
- 連結伺服器（linked server）。
- 其他資料庫的組態設定（例如：檔案名稱、大小、路徑）。

master 的預設屬性說明如下：

檔案	邏輯名稱	實體名稱	檔案成長
主要資料庫檔	master	master.mdf	以 10%的比例自動成長，直到磁碟已滿。
記錄檔	mastlog	mastlog.ldf	以 10%的比例自動成長，最大至 2 TB。

管理 master 資料庫時，需注意下列幾點：

- 萬一 master 資料庫毀損，SQL Server 便無法正常運作。因此，建議定期為 master 資料庫進行備份，當執行下列工作之後應該要備份 master 資料庫：
  - 更改 SQL Server 其他資料庫組態。
  - 變更 SQL Server 伺服器組態。

- ☐ 新增或刪除使用者帳號。
- ☐ 建立／修改／卸除使用者資料庫。
- 盡量避免在 master 資料庫內建立資料表或其他資料庫物件。
- 請注意 master 資料庫無法執行以下工作：
  - ☐ 新增檔案或檔案群組
  - ☐ 更改資料庫定序（預設定序是伺服器定序）
  - ☐ 變更資料庫擁有者（擁有者只能是 sa）
  - ☐ 建立全文檢索索引與目錄
  - ☐ 在系統資料表建立觸發程序
  - ☐ 刪除 guest 使用者
  - ☐ 刪除或卸除 master 資料庫
  - ☐ 參與資料庫鏡射
  - ☐ 移除主要檔案群組、主要資料檔或記錄檔
  - ☐ 更改資料庫或檔案群組名稱
  - ☐ 將資料庫組態更改為 OFFLINE
  - ☐ 將資料庫或主要檔案群組的狀態更改為 READ\_ONLY
  - ☐ 啟用異動資料擷取

### A.1.2 model 資料庫

model 資料庫是所有新資料庫的範本。無論是用 Management Studio 或是 CREATE DATABASE 陳述式建立資料庫，SQL Server 都會先抄寫一份 model 資料庫的組態，作為建立新資料庫的基礎。但建立新資料庫時如有另外指定某些選項，這些選項設定則會直接套用在新資料庫。

model 的預設屬性說明如下：

檔案	邏輯名稱	實體名稱	檔案成長
主要資料庫檔	modeldev	model.mdf	以 64KB 自動成長，直到磁碟已滿。
記錄檔	modellog	modellog.ldf	以 64KB 自動成長，最大至 2 TB。

開啟 Management Studio 工具內的「檔案總管」，使用滑鼠右鍵點選 model 資料庫，選擇快捷選單內的屬性。檢視 model 料庫預設的資料庫檔和記錄檔的初始組態值，其資料檔案及記錄檔都是 8 MB，如圖 A.2 所示。



圖 A.2 model 資料庫的檔案組態設定

使用 Management Studio 建立新資料庫時，可發現在「檔案」頁籤內除了資料庫名稱外，大部分的設定值會參照 model 的選項設定，如圖 A.3 所示。

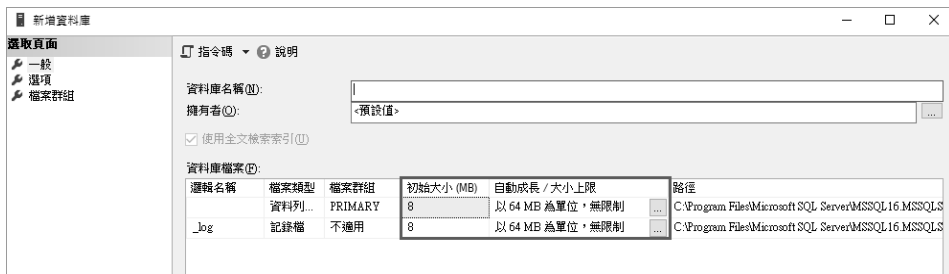


圖 A.3 建立新資料庫時預設值會參考 model 資料庫的組態設定

若使用下列 T-SQL 語法建立名稱為 myDB 的資料庫，在不指定資料檔及記錄檔大小以及其他屬性的情況下，新建立的資料檔會套用 model 系統資料庫的檔案設定。

```
CREATE DATABASE [myDB]
```

圖 A.4 是 model 資料庫內各選項的預設值。新建的資料庫會比照這些設定，但不是每一選項都可修改。若使用「資料庫屬性」對話窗修改 model 資料庫內的選項設定時，無法修改的項目會呈現灰色狀態，僅供檢視。若要使用 T-SQL 的 ALTER DATABASE 指令修改 model 資料庫屬性，請留意此部分。

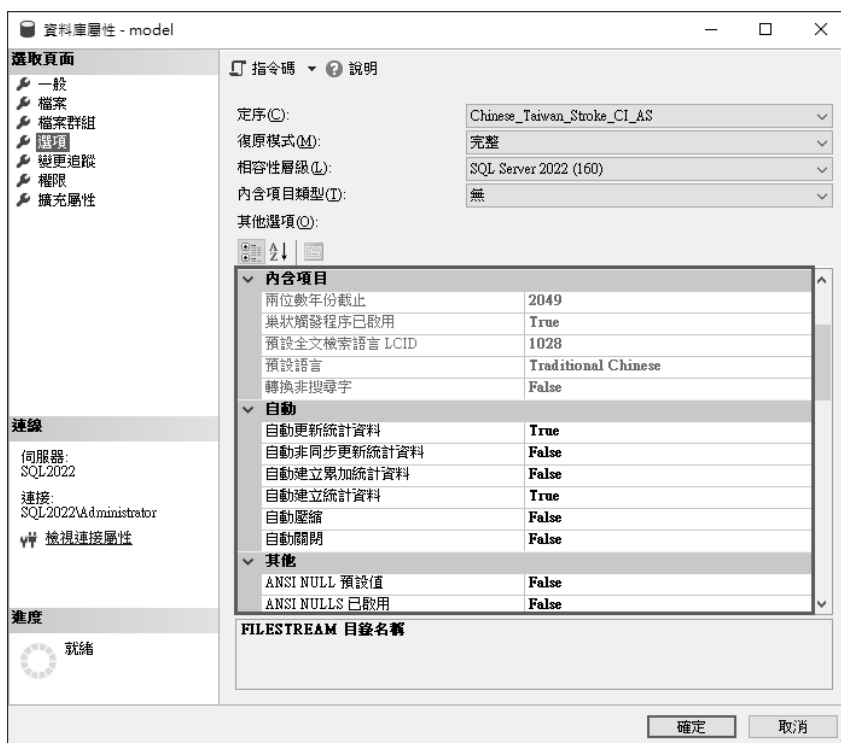


圖 A.4 model 資料庫的選項設定



例如透過以下的語法嘗試修改「可信任」屬性：

```
ALTER DATABASE model SET TRUSTWORTHY ON
```

會得到如下的錯誤訊息：

```
訊息 15309，層級 16，狀態 1，行 1
無法改變 model 或 tempdb 資料庫的可信任狀態。
訊息 5069，層級 16，狀態 1，行 1
ALTER DATABASE 陳述式失敗。
```

針對一般使用者自訂資料庫在特定的需求下，可以修改 TrustWorthy 屬性，但有安全風險，我們猜想這是為何在 model 系統資料不讓設定，避免誤設後，所有新建的使用者自訂資料庫都啟用了 TrustWorthy。

另外，若新增的資料庫預設要建立共通的物件，如參照用的資料表、預存程序、檢視等，可以先建立在 model 資料庫內，之後新增資料庫時便會自動建立這些物件。tempdb 預設需要建立共通物件時，也可使用此方法。

### A.1.3 msdb 資料庫

msdb 系統資料庫可以說是各種衍生服務的專屬資料庫，如 SQL Server Agent 服務、Service Broker、SQL Server Integration Services<sup>2</sup>、Log Shipping、Database Mail 等。只要在 SQL Server 設定任何自動化工作（例如：作業與警示），其組態資訊將存放在 msdb 資料庫。嚴格來說，msdb 系統資料庫如果毀損，SQL Server 雖不至於停擺，但原先設定的自動化工作及上述的各種服務將無法正常執行。

---

<sup>2</sup> SQL Server 2012 版後，SSIS 若採用專案部署模型，則會使用另一個系統資料庫 SSISDB。

msdb 系統資料庫的預設屬性說明如下：

檔案	邏輯名稱	實體名稱	檔案成長
主要資料檔	MSDBData	MSDBData.mdf	以 10%的比例自動成長，直到磁碟已滿。
記錄檔	MSDBLog	MSDBLog.ldf	以 10%的比例自動成長，最大至 2 TB。

管理 msdb 資料庫時，需注意下列幾點：

- 雖然可以直接修改 msdb 資料庫系統資料表的內容，但應盡量避免。
- 新增、修改或移除 SQL Server Agent 作業與警示、資料庫維護計畫、交易紀錄傳送、封裝部署 SSIS 等，應該備份 msdb 資料庫。
- msdb 資料庫無法執行下列工作：
  - ☐ 更改資料庫定序（預設定序是伺服器定序）
  - ☐ 刪除 guest 使用者
  - ☐ 刪除或卸除資料庫
  - ☐ 移除主要檔案群組、主要資料檔或記錄檔
  - ☐ 參與資料庫鏡像
  - ☐ 更改資料庫或檔案群組名稱
  - ☐ 將資料庫組態更改為 OFFLINE
  - ☐ 將主要檔案群組的狀態更改為 READ\_ONLY
  - ☐ 啟用異動資料擷取

## A.1.4 tempdb 資料庫

顧名思義，tempdb 是個「暫時性」的資料庫，當使用者建立的暫存物件（如暫存資料表）也會放在 tempdb，而暫存物件可以是由任何使用者建立，可視範圍有全域（global）與區域（local）<sup>3</sup> 兩種，差別在於區域暫存物件只能在自己當下的 session 可以看到，而全域物件則是其他使用者也可以參照。SQL Server 在有需要時也會自動建立暫存物件或記錄相關作業，以便完成語法、交易或復原交易。

tempdb 的預設屬性說明如下：

檔案	邏輯名稱	實體名稱	預設 初始大小	預設檔案成長
主要資料檔	tempdev	tempdb.mdf	8MB	以 64KB 自動成長，直到磁碟已滿
次要資料檔	temp#	tempdb_mssql_#.ndf	8MB	以 64KB 自動成長，直到磁碟已滿
記錄檔	templog	templog.ldf	8MB	以 64KB 自動成長，最大至 2 TB

與 tempdb 資料庫有關的工作包括：

- 建立「暫時性物件」，包括：全域/區域暫存資料表（global/local temporary tables）、暫時預存程序（temporary stored procedure）、資料表變數（table variable）、資料表值函式中傳回的資料表、指標（cursor）。
- 資料庫引擎執行連結、排序、彙整、資料指標、大型物件（LOB）或維護索引之類的工作時的暫時性資料。

<sup>3</sup> local temporary table 的 local 翻譯有些混亂，有時譯為「區域」或「本機」。但「本機」易有誤導之嫌，讓人誤以為相對的 global temporary table 可以跨機器。

- 存放特殊交易產生的不同資料行版本，包括：快照隔離交易（snapshot isolation transactions）、線上索引維護、MARS（Multiple Active Result Sets），以及 AFTER 觸發程序。

使用者自訂的暫存資料表一般常用於儲存計算後的結果，例如聯結多個不同的資料表並進行複雜計算，然後將結果儲存於暫存資料表，再進行下一階段的運算。

自 SQL Server 2016 (13.x) 版本後，對 tempdb 的效能：

- 暫存資料表及資料表變數會被快取，以減少頁面配置競爭。
- 改進配置頁面閘鎖通訊協定，減少 UP（更新）閘鎖數目。
- 降低 tempdb 的記錄負擔，減少記錄檔的磁碟 I/O。
- SQL Server 安裝程式在安裝執行個體時，新增 tempDB 資料庫檔案的設定，使用者可透過安裝程式介面或是命令列參數 /SQLTEMPDBFILECOUNT 指定 tempDB 資料庫檔案數量<sup>4</sup>；及資料檔案與記錄檔的大小及自動成長。

SQL Server 2019 版後，針對 tempdb 存放中繼資料的系統資料表；提供放入記憶體最佳化（memory-optimized）的選項，藉以減低多使用者同時增刪修中繼物件，因爭用而形成的效能瓶頸。但並非將暫存資料表或使用者自建資料表改以記憶體最佳化結構存放。

4

若作業系統邏輯處理器數目小於或等於 8，tempDB 資料檔數目的預設值會與邏輯處理器數目相同；若邏輯處理器數目大於 8，則 tempDB 的資料檔案數目預設會為 8 個。



圖 A.5 tempDB 資料庫檔案數量 及資料檔案與記錄檔的大小及自動成長

- 檔案自動成長的設定不需啟用追蹤旗標 1117，所有資料庫檔案預設會同時以相同大小自動成長。
- 不需啟用追蹤旗標 1118，SQL Server 預設會為 tempdb 中的物件配置 uniform extent<sup>5</sup>。
- 主要檔案群組啟用 AUTOGROW\_ALL\_FILES 屬性，且無法修改。

5

SQL Server 有兩種 extent 類型：

Uniform extents：由 1 個物件所擁有，extent 中的 8 個 page 只能被擁有的物件所使用。

Mixed extents：最多可由 8 個物件擁有，extent 中的 8 個 page 可以由不同物件所使用。

管理 tempdb 資料庫時，請留意：

- 每當 SQL Server 重新啟動，tempdb 資料庫會被重新建立。因此，若在此資料庫內建立永久性資料庫物件（例如：資料表、預存程序、檢視）需考慮其生命週期。如果 tempdb 資料庫一定需要某些相關物件，可建立在 model 資料庫中，當資料庫服務重新啟動時便會複製一份至 tempdb 資料庫（但也會複製到之後新建立的使用者資料庫中）。
- 登入 SQL Server 後，以 CREATE TABLE 陳述式建立的暫存資料表，在使用完畢後應明確刪除暫存資料表。除了讓資源得以回收，也可以避免重複建立同樣的暫存資料表所產生的錯誤。
- 為了提昇資料排序，或是索引建置效率，最好將 tempdb 資料庫移到存取速度較快的磁碟子系統（例如 SSD 硬碟），定義與 CPU 核心數相同但不大於 8 的資料庫檔案（若系統有 20 顆 CPU，則只需要建立 8 個資料庫檔），並設定相同大小和成長量。
- tempdb 資料庫無法執行以下工作：
  - 新增檔案群組。
  - 資料庫備份或還原。
  - 更改資料庫定序。預設定序是伺服器定序，若自訂資料庫的定序與伺服器不同時，需特別留意操作暫存資料表。因 tempdb 與使用者資料庫定序不相同，可能會遭遇定序衝突問題。
  - 變更資料庫擁有者（擁有者只能是 sa）。
  - 刪除 guest 使用者。
  - 刪除 tempdb 資料庫。
  - 參與資料庫鏡射。
  - 移除主要檔案群組、主要資料庫檔或記錄檔。

- ☐ 更改資料庫或檔案群組名稱。
  - ☐ 執行 DBCC CHECKALLOC、DBCC CHECKCATALOG。
  - ☐ 將資料庫組態更改為 OFFLINE。
  - ☐ 將資料庫或主要檔案群組的狀態更改為 READ\_ONLY。
  - ☐ 建立資料庫快照集。
  - ☐ 啟用異動資料擷取。
- 建議將 tempdb 資料庫的頁面確認設定使用 CHECKSUM，以保障資料的正確性，如圖 A.6 所示。

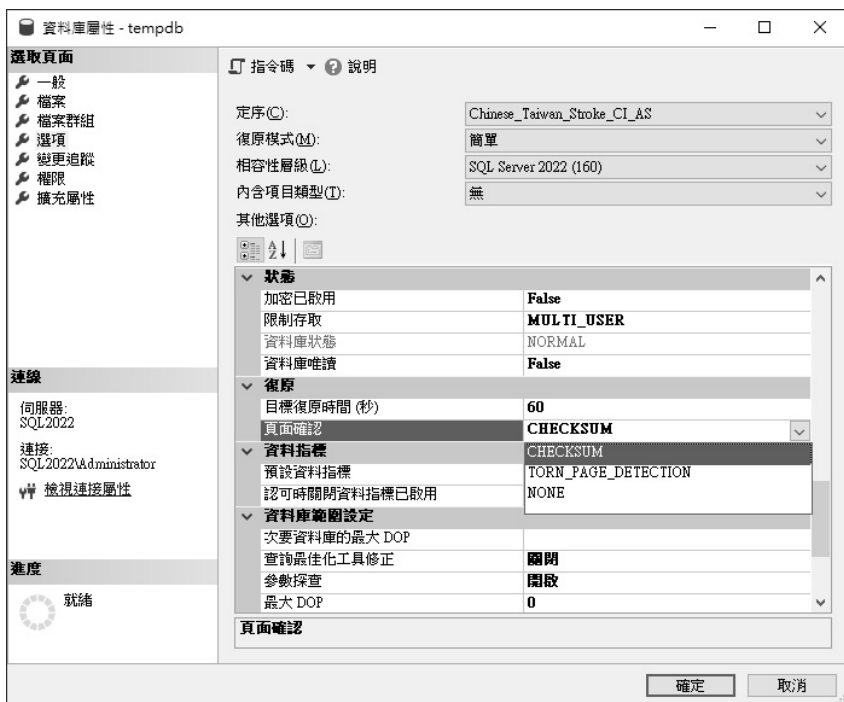


圖 A.6 tempdb 資料庫，「頁面確認」選項預設為 CHECKSUM

## A.1.5 Resource 資料庫

Resource 資料庫是 SQL Server 2005 版本後新增的系統資料庫，是個唯讀資料庫，用來存放系統物件，出現在每一資料庫的 sys 結構描述裡，例如 sys.objects 檢視表。每個 SQL Server 執行個體只有一個相關聯的 mssqlsystemresource.mdf、mssqlsystemresource.ldf 檔案，其預設路徑為 < drive >:\Program Files\Microsoft SQL Server\MSSQL<version>.\< instance\_name >\MSSQL\Binn\，且各個執行個體不共用此檔案。由於此系統資料庫並不會顯示於 Management Studio 的「物件總管」，許多 DBA 經常會忽略此資料庫的重要性。

在 SQL Server 2000 以前的版本，每個資料庫各自擁有系統物件，如系統資料表、系統預存程序、系統函數等。上述機制在 SQL Server 2005 後有了調整：

- 系統資料表仍存放在各自的資料庫內。
- 其他系統物件，如系統檢視表、預存程序...等，放在 Resource 資料庫。
- 每個資料庫內都有 Resource 資料庫提供的系統物件，例如：INFORMATION\_SCHEMA.XXX、sys.XXX...等，透過這些物件來管理資料庫。

SQL Server 本身無法備份及還原 Resource 資料庫，但可以將 mssqlsystemresource.mdf、mssqlsystemresource.ldf 檔視為二進位（.EXE）檔案而非資料庫檔案，進行以檔案或磁碟為基礎的備份機制，只能手動完成。另外，建議不應移動與更改 Resource 資料庫的位置，同時勿使用過期或不安全的 Resource 資料庫來複寫目前的資料庫，版本相關資訊可使用 T-SQL 陳述式查看：



## ■ Resource 資料庫的版本號碼

```
SELECT SERVERPROPERTY('ResourceVersion')
```

## Resource 資料庫上次更新的時間

```
SELECT SERVERPROPERTY('ResourceLastUpdateDateTime')
```

下表為各系統資料庫的整理：

系統資料庫	說明
master	<ul style="list-style-type: none"> <li>記載 SQL Server 執行個體所有系統層級資訊。例如 SQL Server 登入帳戶、端點、連結的伺服器，及系統相關組態設定等。</li> <li>若 master 系統資料庫毀損，SQL SERVER 服務將無法啟動。（可使用啟動參數 3607 試試能否將 SQL 服務帶起來。）</li> </ul>
msdb	<ul style="list-style-type: none"> <li>SQL Server Agent 記載 SQL Server 執行個體所有作業、排程和警示相關資訊。</li> <li>若 msdb 系統資料庫毀損，SQL SERVER 服務仍可啟動，但無法啟用 SQL Agent Services。</li> </ul>
model	<ul style="list-style-type: none"> <li>SQL Server 執行個體建立資料庫的範本。對 model 資料庫進行的修改（例如資料庫大小、定序、復原模式和其他資料庫選項）將會套用到之後建立的任何資料庫（含 tempdb）。</li> <li>若 model 系統資料庫毀損，將無法產生 tempdb 系統資料庫，因而導致 SQL SERVER 服務無法啟動（如使用啟動參數 3608，可略過 master 之外的資料庫，就可以將 SQL 服務帶起來）。</li> </ul>
Resource	<ul style="list-style-type: none"> <li>唯讀資料庫，其中包含 SQL Server 擁有的系統物件。各系統物件實際上保存在 Resource 資料庫中，邏輯上則會出現在每個資料庫 sys 結構描述。若以 Management Studio 的「物件總管」檢視，會在以「系統」為開頭的樹狀節點下。例如，「預存程序」節點下的「系統預存程序」節點內，有一堆屬於 sys 結構描述的系統預存程序，就是映射來自於 Resource 系統資料庫內的系統預存程序。</li> </ul>

系統資料庫	說明
tempdb	<ul style="list-style-type: none"> <li>保存暫存物件或中繼結果集。</li> <li>資料列版本：如資料庫使用資料列版本控制隔離讀取認可、快照集隔離交易。</li> <li>特定作業所產生的資料列版本：如線上索引作業、Multiple Active Result Set (MARS) 和 AFTER 觸發程序。</li> <li>若 tempdb 系統資料庫毀損，SQL SERVER 服務將無法啟動。</li> </ul>

DBA 應對 SQL Server 備份必要的系統資料庫，作為災難恢復的一部分，最重要的是記錄每個系統和使用者資料庫的位置。

## A.2 移動系統資料庫

所有系統資料庫的存放位置都可修改，所以可將系統資料庫移至更快更穩的儲存媒體上。但 Resource 資料庫是無法移動，因為套用 SQL Server Service Pack 和 Hotfix 會將資料庫還原至預設路徑。另外，若將容錯移轉叢集環境中的 Resource 資料庫移至非叢集位置，將會導致容錯移轉叢集失敗。本書所介紹的移動系統資料庫，是指在同一個 SQL Server 執行個體之下，異動資料庫位置。

### A.2.1 移動 master 資料庫

若要移動 master 資料庫，先確認 SQL Server 資料庫引擎的服務帳戶具有檔案新位置的完整存取權後，可遵循下列步驟：

**step01** 開啟「SQL Server 組態管理員」→點選「SQL Server 服務」節點→滑鼠右鍵點選要修改 master 資料庫位置的 SQL Server 執行個體→滑鼠右鍵選擇「內容」選項。在「SQL Server 內容」對話窗中切換到「啟動參數」頁籤→在「現有參數(E)」選擇要

異動的參數→修改「指定啟動參數(S)」裡的路徑→修改完畢後按「更新(U)」。

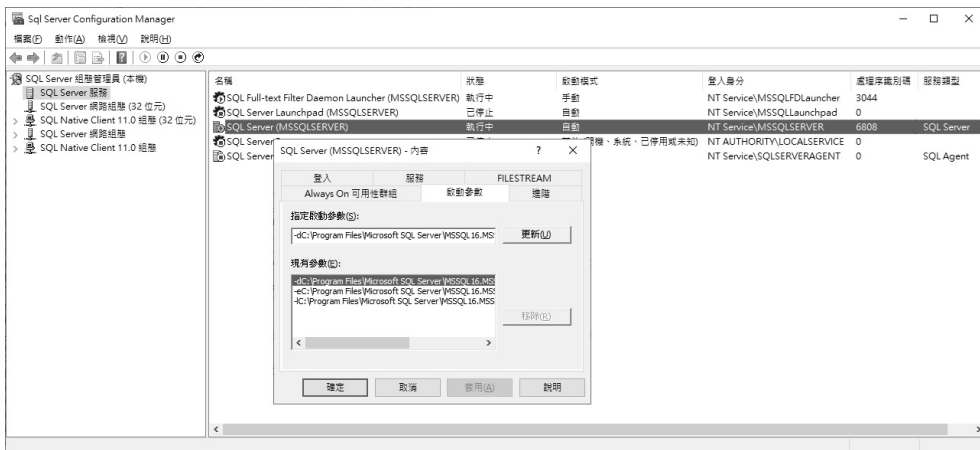


圖 A.7 修改 master 資料庫位置前，需更改 SQL Server 的啟動參數

**step02** 將「啟動參數」值改至 master 資料庫資料及記錄檔的新位置，然後點選「確定」按鈕。請注意：資料檔的參數是 -d，而記錄檔的參數是 -l。範例如下：

```
-dD:\SQLData\Master\master.mdf;  
-eC:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\LOG\ERRORLOG;  
-lE:\SQLData\Master\mastlog.ldf
```

-e 參數代表 SQL Server 錯誤記錄檔的存放位置，此項目是選擇性的，可依需求另外更改路徑。

**step03** 以滑鼠右鍵按一下執行個體名稱並選擇「停止」，即可停止 SQL Server 的執行個體。

**step04** 將 master.mdf 和 mastlog.ldf 檔移至新位置。

**step05** 重新啟動 SQL Server 的執行個體。

若找不到 master 資料庫，SQL Server 服務將無法啟動，同時也建議於登錄編輯程式（regedit）調整「電腦\HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL16.MSSQLSERVER\Setup」的 SQLDataRoot 值變更為新路徑，以避免未來因更新登錄失敗而導致無法修補和升級作業。

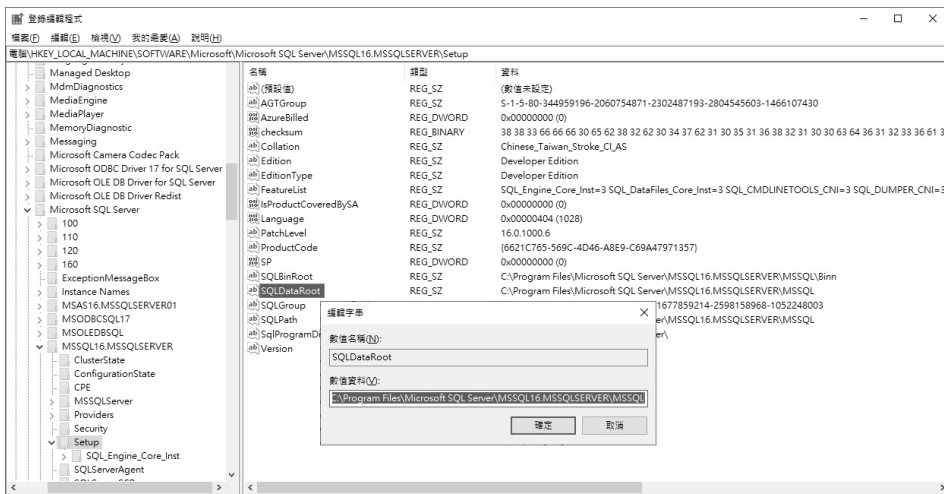


圖 A.8 登錄編輯程式調整的 SQLDataRoot 值變更為新路徑

master 系統資料庫的重建與還原，請參考本書第 5 章《第 5.17 節：master 資料庫的備份與重建》及《第 5.18 節：還原 master 資料庫》。

## A.2.2 移動 tempdb 資料庫

tempdb 是所有資料庫共用的暫存區，此資料庫所存放的實體硬碟位置最好能與使用者自訂資料庫的位置不同，除了可提升資料存取的 I/O 效

能，也能確保 tempdb 有較大的儲存空間，以避免 tempdb 硬碟空間不足的意外狀況<sup>6</sup>。

可透過下列步驟，將 tempdb 資料檔和記錄檔移動到新位置。

**step01** 在 Management Studio 環境開啟「物件總管」→滑鼠右鍵點選 tempdb→選擇其中的「屬性」選項→在「資料庫屬性」對話窗的「檔案」項目中判斷 tempdb 資料庫的邏輯檔案名稱；以及目前的磁碟位置，如圖 A.9 所示。



圖 A.9 透過「資料庫屬性」對話窗確定 tempDB 目前存放路徑

**step02** 在此畫面僅可以增加檔案，但無法修改舊有的設定。可透過 ALTER DATABASE 陳述式移動檔案路徑，範例如下：

```
ALTER DATABASE tempdb MODIFY FILE (NAME='tempdev',
FILENAME='E:\TempDB\tempdb.mdf')
ALTER DATABASE tempdb MODIFY FILE (NAME='templog',
FILENAME='E:\TempDB\templog.ldf')
ALTER DATABASE tempdb MODIFY FILE (NAME='temp2',
FILENAME='E:\TempDB\tempdb_mssql_2.ndf')
ALTER DATABASE tempdb MODIFY FILE (NAME='temp3',
FILENAME='E:\TempDB\tempdb_mssql_3.ndf')
```

<sup>6</sup> 當有大資料時，小心許多 T-SQL DML 語法在執行過程中會大量使用暫存物件，而導致耗損 tempdb 的檔案空間。

```
ALTER DATABASE tempdb MODIFY FILE(NAME='temp4',  
FILENAME='E:\tempdb\tempdb_mssql_4.ndf')
```

step03 完成修改後，重新啟動 SQL Server 執行個體。

step04 再次透過「資料庫屬性」對話窗或查詢 `sys.master_files` 系統資料表，確認檔案是否變更成功。可再進行原檔案路徑下未使用的 tempdb 檔案刪除作業。

## A.2.3 移動 msdb 及 model 資料庫

若要移動 msdb 或 model 的資料檔與記錄檔，可依照下列步驟執行。

step01 對於要移動的每個檔案執行下列陳述式：

```
--MSDB db file  
use master;  
go  
Alter Database MSDB Modify File (Name = MSDBData, FILENAME = '<path  
here>\MSDBData.mdf')  
GO  
  
--MSDB Log file  
ALTER DATABASE MSDB MODIFY FILE (NAME = MSDBLog, FILENAME = '<path  
here>\MSDBLog.ldf');  
GO  
  
--MODEL db file  
use master;  
go  
Alter Database Model Modify File (Name = Modeldev, FILENAME = '<path  
here>\Model.mdf')  
GO  
  
--MODEL log file  
ALTER DATABASE Model MODIFY FILE (NAME = Modellog, FILENAME = '<path  
here>\Modellog.ldf');
```

程式碼執行結果如圖 A.10 所示：



圖 A.10 移動 msdb 或 model 的資料檔與記錄檔後資料庫下次啟動時將會使用新路徑

### step 02 | 停止 SQL Server 執行個體以執行維護：

- 將資料庫檔案搬移到新位置。
- 重新啟動 SQL Server 的執行個體或伺服器。
- 執行下列查詢以驗證檔案變更。

```
SELECT name, physical_name AS CurrentLocation, state_desc
FROM sys.master_files
WHERE database_id = DB_ID(N'<database_name>');
```

若 SQL Server 執行個體有設定 Database Mail，則移動 msdb 資料庫後需再完成下列步驟；

### step 01 | 透過執行下列查詢，確認已經為 msdb 資料庫啟用 Service Broker。

```
SELECT is_broker_enabled
FROM sys.databases
WHERE name = N'msdb';
```

### step 02 | 透過傳送測試郵件，來確認 Database Mail 是否正常運作。

## A.3 系統物件

在 SQL Server 2005 之後的版本已將系統資料表（System Table）隱藏起來，並改以大量的系統檢視（System View）取代之。如「目錄檢視（Catalog View）」和「動態管理檢視/函數（Dynamic Management View, DMVs/Dynamic Management Functions, DMFs）」，這些系統檢視的內容可以分為兩種：靜態的系統中繼資料，以及用來呈現當下運行狀況的動態管理檢視。兩種系統檢視全都歸類在 sys 結構描述（schema）之下，動態管理檢視和函數除了名稱會以 dm 當前置詞外，並在系統資料庫和使用者自訂資料庫都可以看到，部分動態檢視用來呈現伺服器等級的資料，另一部分則是用來呈現資料庫的狀況，可透過下列語法查出動態管理檢視和函數。

```
SELECT name, type_desc FROM sys.system_objects WHERE name LIKE 'dm_%' ORDER BY name
```

查詢結果如圖 A.11 所示。



	name	type_desc
1	dm_audit_actions	VIEW
2	dm_audit_class_type_map	VIEW
3	dm_broker_activated_tasks	VIEW
4	dm_broker_connections	VIEW
5	dm_broker_forwarded_messages	VIEW
6	dm_broker_queue_monitors	VIEW
7	dm_cache_hit_stats	VIEW
8	dm_cache_size	VIEW
9	dm_cache_stats	VIEW
10	dm_cdc_errors	VIEW
11	dm_cdc_log_scan_sessions	VIEW
12	dm_change_feed_errors	VIEW
13	dm_change_feed_log_scan_sessions	VIEW
14	dm_clr_appdomains	VIEW
15	dm_clr_loaded_assemblies	VIEW
16	dm_clr_properties	VIEW
17	dm_clr_tasks	VIEW
18	dm_cluster_endpoints	VIEW
19	dm_column_encryption_enclave	VIEW
20	dm_column_encryption_enclave...	VIEW
21	dm_column_encryption_enclave...	VIEW
22	dm_column_store_object_pool	VIEW
23	dm_cryptographic_provider_algor...	SQL_INLINE_TABLE_VALUED_FUNCTION
24	dm_cryptographic_provider_keys	SQL_INLINE_TABLE_VALUED_FUNCTION
25	dm_cryptographic_provider_prop...	VIEW
26	dm_cryptographic_provider_sessi...	SQL_INLINE_TABLE_VALUED_FUNCTION
27	dm_database_backups	VIEW
28	dm_database_encryption_keys	VIEW
29	dm_database_external_policy_acti...	VIEW
30	dm_database_external_policy_pri...	VIEW
31	dm_database_external_policy_pri...	VIEW
32	dm_database_external_policy_role...	VIEW
33	dm_database_external_policy_role...	VIEW
34	dm_database_external_policy_roles	VIEW
35	dm_dh_column_store_row_encryp...	VIEW

圖 A.11 透過 sys.system\_objects 查出動態管理檢視和函數

除了系統檢視外，SQL Server 也提供系統預存程序、函數、變數、DBCC...等，但由於這些物件的應用偏向開發或是較為進階的管理，在此書並不討論。

以下將以幾個簡單的範例介紹如何使用 T-SQL 陳述式查詢系統資訊以協助日常管理。

### A.3.1 系統檢視

一般透過 Management Studio 的「物件總管」，可以在各資料庫內「檢視」→「系統檢視表」的節點下看到大量的系統檢視表，如圖 A.12

所示，透過 **sys.tables**、**sys.objects** 等目錄檢視，提供取得中繼資料的一般介面。

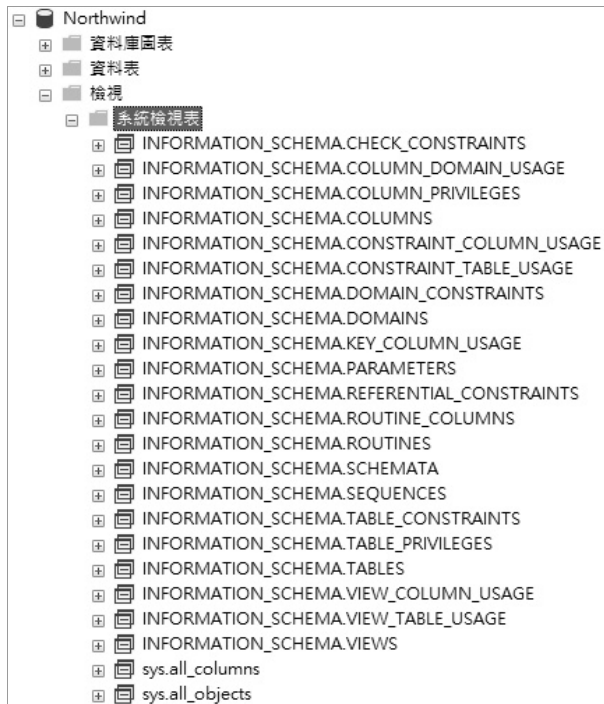


圖 A.12 透過物件總管檢視動態管理函數

更多的系統檢視請參考線上說明：<https://docs.microsoft.com/zh-tw/sql/relational-databases/system-catalog-views/catalog-views-transact-sql?view=sql-server-ver16>。

當中有「目錄檢視」、「相容性檢視」、「動態管理檢視和函數」、「資訊結構描述檢視」、「系統預存程序和資料表」與「資料層應用程式檢視表（Transact-SQL）」。

接下來介紹幾個常用的「目錄檢視」與「動態管理檢視和函數」的使用範例。

**範例程式 A.1：**使用 sys.objects 檢視 Northwind 資料庫當天修改過的物件。

```
USE Northwind
SELECT name AS object_name
      ,SCHEMA_NAME(schema_id) AS schema_name
      ,type_desc
      ,create_date
      ,modify_date
FROM sys.objects
WHERE modify_date > GETDATE() - 1
ORDER BY modify_date;
```

查詢結果如圖 A.13 所示。

	object_name	schema_name	type_desc	create_date	modify_date
1	Table_2	dbo	USER_TABLE	2022-07-29 09:48:24.233	2022-07-29 09:48:24.237

圖 A.13 使用 sys.objects 檢視 Northwind 資料庫當天修改過的物件

**範例程式 A.2：**檢視 Northwind 資料庫位置。

```
SELECT name, physical_name AS CurrentLocation
FROM sys.master_files
WHERE database_id = DB_ID(N'Northwind');
```

查詢結果如圖 A.14 所示。

	name	CurrentLocation
1	Northwind	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\Northwind.mdf
2	Northwind_log	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\Northwind_log.ldf

圖 A.14 利用目錄檢視查詢資料庫的存放位置

**範例程式 A.3：**利用 sys.objects 搭配 sys.schemas、sys.sysindexes 系統目錄檢視，回傳資料庫所有資料表所含記錄筆數及建立日期。

```
USE Northwind
SELECT db_name() AS 資料庫名稱, sys.schemas.name AS 結構描述名稱,o.name AS 物件名稱,
      i.rows AS 記錄筆數, o.crdate AS 物件的建立日期
FROM sys.sysobjects o
      LEFT JOIN sys.sysindexes i ON o.id=i.id AND i.indid<=1
      inner JOIN sys.schemas ON schema_id=uid
WHERE Type='U'
```

sys.sysobjects 目錄檢視的 type 欄位值可用來過濾資料庫的物件類別。例如：type='U' 代表使用者自訂資料表、type='P' 代表預存程序等，各欄位的定義可至線上說明<sup>7</sup>的“sys.sysobjects”目錄檢視。下圖為使用者自訂資料表的記錄筆數、物件的建立日期等資料：

	資料庫名稱	結構描述名稱	物件名稱	記錄筆數	物件的建立日期
1	Northwind	dbo	Employees	9	2014-10-07 14:48:11.973
2	Northwind	dbo	Categories	8	2014-10-07 14:48:12.087
3	Northwind	dbo	Customers	91	2014-10-07 14:48:12.090
4	Northwind	dbo	Shippers	3	2014-10-07 14:48:12.100
5	Northwind	dbo	Suppliers	29	2014-10-07 14:48:12.103
6	Northwind	dbo	Orders	833	2014-10-07 14:48:12.110
7	Northwind	dbo	Table_2	0	2022-07-29 09:48:24.233
8	Northwind	dbo	Products	77	2014-10-07 14:48:12.130
9	Northwind	dbo	Order Details	2155	2014-10-07 14:48:12.143
10	Northwind	dbo	CustomerCustomerDemo	0	2014-10-07 14:48:14.890
11	Northwind	dbo	CustomerDemographics	0	2014-10-07 14:48:14.893
12	Northwind	dbo	Region	4	2014-10-07 14:48:14.897
13	Northwind	dbo	Territories	53	2014-10-07 14:48:14.900
14	Northwind	dbo	EmployeeTerritories	49	2014-10-07 14:48:14.900
15	Northwind	dbo	Table_1	49	2015-09-11 09:20:36.193

圖 A.15 利用 sys.objects 搭配 sys.schemas、sys.sysindexes 系統目錄檢視，回傳資料庫所有資料表所含記錄筆數及建立日期

**範例程式 A.4：**使用 sys.identity\_columns 目錄檢視查詢各資料表是否包含 identity 欄位，並列出 identity 欄位名稱。

```
USE Northwind
SELECT s.Name AS 結構描述名稱, t.Name AS 資料表名稱,
CASE WHEN c.NAME IS NULL THEN 'N' ELSE 'Y' END [是否有 Identity],
c.NAME AS [Identity 欄位名稱]
FROM sys.sysobjects t
LEFT JOIN sys.identity_columns c ON object_id=id
JOIN sys.schemas s ON schema_id=uid
WHERE Type='U'
```

<sup>7</sup>

sys.sysobjects 目錄檢視各欄位的定義：<https://docs.microsoft.com/zh-tw/sql/relational-databases/system-compatibility-views/sys-sysobjects-transact-sql?view=sql-server-ver16>

查詢結果如圖 A.16 所示。

	結構描述名稱	資料表名稱	是否有Identity	Identity 欄位名稱
1	dbo	Employees	Y	EmployeeID
2	dbo	Categories	Y	CategoryID
3	dbo	Customers	N	NULL
4	dbo	Shippers	Y	ShipperID
5	dbo	Suppliers	Y	SupplierID
6	dbo	Orders	Y	OrderID
7	dbo	Table_2	N	NULL
8	dbo	Products	Y	ProductID
9	dbo	Order Details	N	NULL
10	dbo	CustomerCustomerDemo	N	NULL
11	dbo	CustomerDemographics	N	NULL
12	dbo	Region	N	NULL
13	dbo	Territories	N	NULL
14	dbo	EmployeeTerritories	Y	ID
15	dbo	Table_1	Y	c1

圖 A.16 使用 sys.identity\_columns 目錄檢視查詢各資料表是否包含 identity 欄位，並列出 identity 欄位名稱

除了上述幾個常用的系統目錄，也可以參考線上說明提供的範例，以輔助管理資料庫：

- [https://msdn.microsoft.com/zh-tw/library/ms345522\(v=SQL.130\).aspx](https://msdn.microsoft.com/zh-tw/library/ms345522(v=SQL.130).aspx)

### A.3.2 動態管理檢視和函數

使用「動態管理檢視」查詢系統資料時，若搭配動態管理函數（Dynamic Management Function），可查詢伺服器執行個體當下的執行狀況、監視伺服器活動並診斷是否有問題。例如連接的狀況、CPU、記憶體等硬體資源的使用、鎖定（Lock）的情形等。「動態管理函數」數量較檢視少，集中呈現在 master 系統資料庫的「可程式性」→「函數」→「系統函數」→「資料表值函數」之下，如圖 A.17 所示。

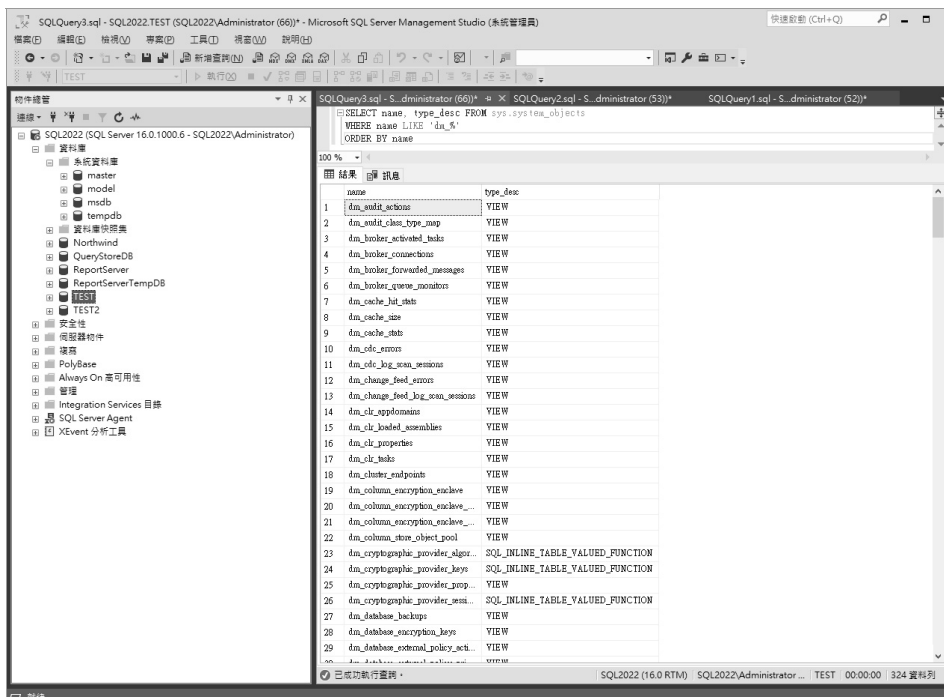


圖 A.17 透過物件總管檢視動態管理函數

動態管理檢視和函數的存取範圍可分為兩種，各需要的權限如下：

- 伺服器範圍：存在 master 資料庫且使用者需具有伺服器的 VIEW SERVER STATE 權限。
- 資料庫範圍：存在每個資料庫且使用者需資料庫的 VIEW DATABASE STATE 權限。

下列使用 T-SQL 語法，依不同的範圍來授與動態管理檢視和函數的存取範圍權限：

```
GRANT VIEW SERVER STATE TO <要授與權限的主體>;
```

或

```
GRANT VIEW DATABASE STATE to <要授與權限的主體>;
```

若要查詢動態管理檢視或函數，除了上述權限外還需要擁有對該物件的 SELECT 權限，以限制使用者存取動態管理檢視和函數。亦即對該使用者不准存取的動態管理檢視或函數，直接設定拒絕 SELECT 權限。

預設這些動態管理檢視都已賦予 Public 角色 SELECT 權限，而所有的登入帳號（Login）又可以對應到 master 資料庫的 guest 使用者，因 guest 使用者隸屬於 Public 角色，故一般的登入帳號應該都可以執行對系統動態管理檢視的查詢。

另外，在 DDL 的 DENY、GRANT 和 REVOKE 中，以 DENY 的優先順序最高。如果已授與使用者 VIEW SERVER STATE 權限，但拒絕 VIEW DATABASE STATE 權限，則使用者可以查看伺服器層級資訊，但不能查看資料庫層級資訊。

整體的動態管理檢視和函數可分為以下類型：

表 A.1 SQL Server 2022 所提供的各種動態管理檢視和函數類別

各種動態管理檢視	函數類別
資料庫相關的動態管理檢視	全文檢索搜尋與語意搜尋動態管理檢視和函數
物件相關的動態管理檢視和函數	空間資料相關動態管理檢視與函數
查詢通知相關的動態管理檢視	Filestream 及 FileTable 動態管理檢視
交易相關的動態管理檢視和函式	記憶體最佳化的資料表動態管理檢視
索引相關的動態管理檢視和函數	Always On 可用性群組動態管理檢視和函式
I/O 相關的動態管理檢視和函式	資料庫鏡像相關的動態管理檢視
安全性相關的動態管理檢視和函數	複寫相關的動態管理檢視
伺服器相關的動態管理檢視與函數	異地複寫動態管理檢視和函式（Azure SQL Database）
執行相關的動態管理檢視和函數	變更追蹤相關的動態管理檢視
擴充事件動態管理檢視	異動資料擷取相關的動態管理檢視

各種動態管理檢視	函數類別
資源管理員相關的動態管理檢視	Common Language Runtime 相關的動態管理檢視
SQL Server 作業系統相關的動態管理檢視	Azure Synapse 分析和平行 Data Warehouse 動態管理檢視
Service Broker 相關的動態管理檢視	Stretch Database 動態管理檢視

接下來將透過幾個簡單的範例，進一步說明動態管理檢視和函數。

**範例程式 A.5：**sys.dm\_exec\_sessions 動態管理檢視傳回的每筆記錄會描述著一個連接的工作階段狀態（此功能類似以往 master 資料庫 sysprocesses 系統檢視）。

```
SELECT login_name,host_name,login_time,program_name
FROM sys.dm_exec_sessions
WHERE Session_ID=@@spid
```

執行結果如圖 A.18 所示。

	login_name	host_name	login_time	program_name
1	SQL2022\Administrator	SQL2022	2022-08-16 16:52:14.000	Microsoft SQL Server Management Studio - 查詢

圖 A.18 透過系統檢視表查詢當下連線狀態

如果使用者具有伺服器的 VIEW SERVER STATE 權限，將可以看到 SQL Server 執行個體上所有執行中的工作階段，否則只會看到自己目前的工作階段。換句話說，若用一般的使用者身份查詢這個檢視，只會傳回自己的連線設定；若以管理者身分執行則會傳回 SQL Server 執行個體目前全部工作階段的資訊（可透過 @@spid 限制只回傳自己的工作階段）。或是透過 is\_user\_process=1 條件來排除 SQL Server 自己使用的工作階段，傳回所有使用者所建立的工作階段。



範例程式 A.6: sys.dm\_exec\_connections 動態檢視與上述查詢 Session 資訊近似，但該管理檢視偏向連接資料庫的方式，如登入認證的方式、網路組態設定等。下列 T-SQL 語法透過 sys.dm\_exec\_connections 動態管理檢視查詢當下的連線登入狀態。

```
SELECT connect_time, protocol_type, local_net_address, net_transport
FROM sys.dm_exec_connections
WHERE Session_ID=@@spid
```

查詢結果如圖 A.19 所示。

	connect_time	protocol_type	local_net_address	net_transport
1	2022-08-16 16:52:14.000	TSQL	NULL	Shared memory

圖 A.19 回傳連線的狀態

從圖 A.19 可以發現，sys.dm\_exec\_connections 動態檢視傳回使用者連線時間、認證方式、通訊協定類型與網路 IP 等管理人員常需要的資訊。

範例程式 A.7: 利用 sys.dm\_exec\_query\_stats 動態管理檢視及 sys.dm\_exec\_sql\_text 動態管理函數，查詢使用 CPU 平均時間最多之前五名。下列範例會傳回 SQL 陳述式的文字。

```
SELECT TOP 5 total_worker_time/execution_count
AS [平均 CPU 使用時間(毫秒)], --microsecond
SUBSTRING(st.text, (qs.statement_start_offset/2)+1,
((CASE qs.statement_end_offset
WHEN -1 THEN DATALENGTH(st.text)
ELSE qs.statement_end_offset
END - qs.statement_start_offset)/2) + 1) AS 查詢語法
FROM sys.dm_exec_query_stats AS qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS st
ORDER BY total_worker_time/execution_count DESC
```

執行結果如圖 A.20 所示。

平均 CPU 使用時間(%)	查詢語法
202078	SELECT db_id() AS database_id, o [type] as ModuleType, COUNT_BIG(*) as ModuleCount FROM sys.objects AS o WITH(nolock) WHERE o.type in (AF, F, FN, FS, FT, IF, P, PC, TA, TF, TR, X, C, D, PG, SM, SO, ...
148354	SELECT db_id() as database_id, sm [is_inlineship] AS InlineshipScalarCount, sm [inline_type] AS InlineType, COUNT_BIG(*) AS ScalarCount, COUNT_BIG(CASE WHEN sm.[definition] LIKE '%getstate%' OR s...
51790	SELECT sp_name AS [Name], SCHEMA_NAME(sp.schema_id) AS [Schema], Server[@Name] + quotename(CAST( serverproperty(@Servername) AS sysname),") + ']' + 'Database[@Name] + quotename(db_name(0,") + ']' + ...
50273	SELECT Server[@Name] + quotename(CAST( serverproperty(@Servername) AS sysname),") + ']' + 'Collation[@Name] + quotename(collname,") + ']' AS [Un], cl_name AS [Name], CAST(COLLATIONPROPERTY(name, C...
48960	SELECT Server[@Name] + quotename(CAST( serverproperty(@Servername) AS sysname),") + ']' + 'Collation[@Name] + quotename(collname,") + ']' AS [Un], cl_name AS [Name], CAST(COLLATIONPROPERTY(name, C...

圖 A.20 最耗 CPU 執行時間的前五名 SQL 查詢語法

## A.4 結語

本章介紹了 SQL Server 的系統資料庫及用途，並搭配系統物件檢視伺服器與資料庫的中繼資料，藉此讓你了解系統資料庫及系統相關檢視的內容，以便系統資料庫發生狀況時能迅速找到原因與資訊。