

SQL Server 檔案類型與管理實務指南

一、SQL Server 中的資料庫檔案類型

在 SQL Server 中，資料庫實際上是由不同類型的實體檔案所組成的，主要包含 MDF、NDF 和 LDF 三種檔案類型。這些檔案各自具有不同的作用和功能：

MDF (Master Data File)

- 主要資料檔
- 每個 SQL Server 資料庫必須有且只能有一個 MDF 檔案
- 包含資料庫的起始點和指向其他檔案的指標
- 儲存系統資料表、使用者資料表、檢視表、預存程序等物件
- 檔案名稱預設以 `.mdf` 為副檔名
- 通常包含資料庫的主要資料和系統物件

NDF (Secondary Data File)

- 次要資料檔
- 可選的資料檔案，資料庫可以有零個或多個 NDF 檔案
- 用於擴展資料庫儲存空間，可分散資料庫負載到多個磁碟上
- 儲存使用者資料，不儲存系統物件
- 檔案名稱預設以 `.ndf` 為副檔名
- 主要用於資料庫成長、效能調校和磁碟管理

LDF (Log Data File)

- 交易日誌檔
- 每個 SQL Server 資料庫必須至少有一個 LDF 檔案
- 記錄所有對資料庫的修改操作
- 用於交易復原、資料庫復原、鏡像、複寫和備份
- 檔案名稱預設以 `.ldf` 為副檔名
- 是資料庫一致性和完整性的關鍵部分

二、檔案的重要性

資料庫還原與復原

- LDF 檔案是實現「時間點還原」的基礎
- 交易日誌可用於回滾未完成的交易
- 系統崩潰後，可利用 LDF 檔案重做已完成但尚未寫入資料檔的交易

效能優化

- 可將 MDF、NDF 和 LDF 放在不同的實體磁碟上，提高 I/O 效能
- 通過增加 NDF 檔案，可以分散資料庫 I/O 負載

資料庫管理

- 這些檔案可以單獨設定大小和成長率
- 可針對不同類型的資料使用不同的檔案群組

三、檔案放置策略與最佳實踐

最佳實踐建議

1. 分離存放不同類型檔案

- 將 MDF 和 NDF 檔案放在不同於 LDF 檔案的磁碟上
- 理想情況：使用獨立的磁碟陣列 (RAID) 系統
- 資料檔案 (MDF/NDF) 適合 RAID 10 或 RAID 5
- 日誌檔案 (LDF) 最適合 RAID 1 或 RAID 10

2. tempdb 的特殊處理

- 將 tempdb 資料檔案放在單獨的高性能磁碟上
- tempdb 的 MDF 和 NDF 檔案數量應與 CPU 核心數量相同 (但不超過 8 個)

3. SSD vs HDD

- SSD 特別適合放置交易日誌檔案 (LDF)
- 在預算允許的情況下，將所有檔案都放在 SSD 上可顯著提升性能

4. 適當規劃檔案大小

- 預先配置足夠的空間，減少自動成長的頻率
- 自動成長會導致效能暫時下降

5. 監控和維護

- 定期監控檔案的成長情況
- 適時收縮交易日誌檔案
- 定期備份完整資料庫和交易日誌

6. 檔案群組策略

- 按照資料訪問頻率或資料類型將資料表分配到不同檔案群組
- 可實現較佳的負載平衡

四、建立及設定資料庫檔案的詳細參數

建立資料庫時設定檔案參數

```
CREATE DATABASE Electronics3CStore
ON PRIMARY
(
    NAME = 'Electronics3CStore_Data',          -- 邏輯名稱，用於在SQL Server內部識別檔案
    FILENAME = 'E:\SQLData\Electronics3CStore.mdf', -- 物理路徑和檔案名稱
    SIZE = 512MB,                               -- 初始大小
    MAXSIZE = 10GB,                             -- 最大大小，可設定為UNLIMITED
    FILEGROWTH = 256MB                          -- 成長增量，建議使用固定值而非百分比
),
FILEGROUP Electronics3CStore_ProductsFG      -- 建立專用檔案群組
(
    NAME = 'Electronics3CStore_Products',
    FILENAME = 'F:\SQLData\Electronics3CStore_Products.ndf',
    SIZE = 256MB,
    MAXSIZE = 5GB,
    FILEGROWTH = 128MB
),
FILEGROUP Electronics3CStore_IndexesFG      -- 索引專用檔案群組
(
    NAME = 'Electronics3CStore_Indexes',
    FILENAME = 'G:\SQLData\Electronics3CStore_Indexes.ndf',
    SIZE = 256MB,
    MAXSIZE = 5GB,
    FILEGROWTH = 128MB
)
LOG ON
(
    NAME = 'Electronics3CStore_Log',
    FILENAME = 'L:\SQLLogs\Electronics3CStore.ldf',
    SIZE = 256MB,
    MAXSIZE = 5GB,
    FILEGROWTH = 128MB
);
```

參數詳細說明

1. SIZE 參數

- 說明：指定檔案的初始大小
- 最佳實務：
 - 預先分配足夠大的空間，避免頻繁的自動擴展
 - 主資料檔案 (MDF) 通常需要較大的初始大小
 - 考慮預期的資料增長率

2. MAXSIZE 參數

- 說明：指定檔案允許的最大大小
- 設定選項：
 - `UNLIMITED`：無限制增長
 - 具體數值：限制檔案大小，避免單一檔案消耗過多磁碟空間
- 實務考量：
 - 必須考慮檔案系統的限制
 - 在 SAN 或雲端存儲環境中，設定適當的上限很重要

3. FILEGROWTH 參數

- 說明：指定檔案自動增長的大小
- 設定選項：
 - 固定大小：如 256MB、512MB
 - 百分比：如 10%、20%
- 最佳實務：
 - 使用固定大小而非百分比：百分比會導致增長量不可預測
 - 小型資料庫：64MB 到 256MB
 - 中型資料庫：256MB 到 512MB
 - 大型資料庫：512MB 到 1GB 或更大

4. 檔案群組 (FILEGROUP) 設定

- 說明：邏輯檔案群組用於組織相關資料和索引
- 實務應用：
 - 將表格和索引放在不同檔案群組
 - 高訪問頻率的資料可放在專用檔案群組
 - 唯讀資料可放在唯讀檔案群組

五、實際運用例子

1. 在特定檔案群組上建立資料表

```
-- 在次要檔案群組上建立產品資料表
CREATE TABLE Products
(
    product_id INT PRIMARY KEY,
    product_name NVARCHAR(200) NOT NULL,
    -- 其他欄位...
)
ON Electronics3CStore_ProductFG; -- 指定使用的檔案群組
```

2. 適用於 3C 零售業務的實際建議

主要資料分佈策略

1. 銷售交易資料

- 高頻寫入，適合放在快速存取的 MDF 或專用 NDF

```
CREATE TABLE Orders (  
    -- 欄位定義  
) ON PRIMARY; -- 使用主檔案群組
```

2. 產品資料

- 讀取頻繁，更新較少，適合單獨的檔案群組

```
CREATE TABLE Products (  
    -- 欄位定義  
) ON Electronics3CStore_ProductsFG;
```

3. 歷史銷售資料

- 較少訪問，可以放在單獨的檔案群組

```
CREATE TABLE OrderHistory (  
    -- 欄位定義  
) ON Electronics3CStore_HistoryFG;
```

六、檔案管理與維護

1. 增加新的資料檔案

```
ALTER DATABASE Electronics3CStore  
ADD FILE  
(  
    NAME = 'Electronics3CStore_Data2',  
    FILENAME = 'H:\SQLData\Electronics3CStore_Data2.ndf',  
    SIZE = 512MB,  
    MAXSIZE = 5GB,  
    FILEGROWTH = 256MB  
) TO FILEGROUP PRIMARY;
```

2. 增加新的日誌檔案

```
ALTER DATABASE Electronics3CStore
ADD LOG FILE
(
    NAME = 'Electronics3CStore_Log2',
    FILENAME = 'L:\SQLLogs\Electronics3CStore_Log2.ldf',
    SIZE = 256MB,
    MAXSIZE = 5GB,
    FILEGROWTH = 128MB
);
```

3. 修改現有檔案的大小

```
ALTER DATABASE Electronics3CStore
MODIFY FILE
(
    NAME = 'Electronics3CStore_Data',
    SIZE = 1GB,
    MAXSIZE = 20GB,
    FILEGROWTH = 512MB
);
```

4. 將表格移動到新的檔案群組

```
-- 首先創建新索引，並指定新的檔案群組
CREATE CLUSTERED INDEX CIX_Products_ID ON Products(product_id)
WITH (DROP_EXISTING = ON) ON Electronics3CStore_IndexesFG;
```

七、監控與優化

1. 定期監控檔案大小及增長

```
-- 檢查資料庫檔案使用情況
SELECT
    DB_NAME() AS DbName,
    name AS FileName,
    physical_name AS PhysicalName,
    type_desc AS FileType,
    size/128.0 AS CurrentSizeMB,
    size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS INT)/128.0 AS FreeSpaceMB,
    CAST(FILEPROPERTY(name, 'SpaceUsed') AS INT)/128.0 AS UsedSpaceMB,
    growth/128.0 AS GrowthMB,
    is_percent_growth AS IsPercentGrowth,
    max_size/128.0 AS MaxSizeMB
FROM sys.database_files;
```

2. 碎片整理

```
-- 執行定期碎片整理
ALTER INDEX ALL ON Products REORGANIZE;
-- 對於嚴重碎片的索引，使用重建
ALTER INDEX ALL ON Products REBUILD WITH (ONLINE = ON);
```

3. 壓縮交易日誌

```
-- 在完整恢復模式下，先備份日誌
BACKUP LOG Electronics3CStore TO DISK = 'L:\Backup\Electronics3CStore_Log.bak';
-- 在簡單恢復模式下，可以壓縮日誌
DBCC SHRINKFILE (Electronics3CStore_Log, 100); -- 壓縮到100MB
```

八、進階配置選項

1. 檔案群組備份策略

```
-- 備份特定檔案群組
BACKUP DATABASE Electronics3CStore
FILEGROUP = 'Electronics3CStore_ProductsFG'
TO DISK = 'F:\Backup\Electronics3CStore_Products.bak';
```

2. 分割資料表到不同檔案群組

```
-- 建立分割函數和分割方案
CREATE PARTITION FUNCTION OrderDateRangePF (datetime)
AS RANGE RIGHT FOR VALUES ('2025-01-01', '2025-04-01', '2025-07-01', '2025-10-01');

CREATE PARTITION SCHEME OrderDateRangePS
AS PARTITION OrderDateRangePF
TO (ProductsFG, IndexesFG, PRIMARY, ProductsFG, IndexesFG);

-- 使用分割方案建立資料表
CREATE TABLE Orders (
    order_id INT IDENTITY(1,1),
    order_date DATETIME,
    -- 其他欄位
    PRIMARY KEY (order_id, order_date)
) ON OrderDateRangePS(order_date);
```

SQL Server 中的檔案群組 (Filegroup) 說明

檔案群組是 SQL Server 中用於組織和管理資料庫檔案的邏輯結構。它們允許您將多個資料檔案 (MDF 和 NDF) 組織成邏輯單位，以便更有效地管理資料庫的儲存、效能和備份策略。以下是關於檔案群組的詳細說明：

檔案群組的基本概念

定義與用途

- **檔案群組**是一個或多個資料庫檔案的邏輯集合
- 用於組織和管理相關的資料庫物件和資料
- 每個資料庫必須至少有一個檔案群組，即**主要檔案群組 (PRIMARY)**
- **交易日誌檔案 (LDF)** 不屬於任何檔案群組

主要檔案群組 (PRIMARY)

- 每個資料庫只有一個**主要檔案群組**
- 包含**主要資料檔案 (MDF)**
- 儲存系統資料表和系統物件
- 若未指定，使用者物件默認存放於此

使用者定義檔案群組

- 可以建立多個使用者定義檔案群組
- 每個檔案群組可包含一個或多個資料檔案 (NDF)
- 可用於分散資料庫負載、改善效能及實現特定的儲存策略

檔案群組的優勢

效能優化

- 可將不同資料表或索引分配到不同的檔案群組
- 將檔案群組分布在不同的實體磁碟上，實現 I/O 負載平衡
- 可將經常一起使用的資料放在同一檔案群組，減少磁頭移動

管理便利性

- 簡化備份和還原操作 (可按檔案群組進行)
- 改善資料庫維護的靈活性
- 支援部分數據庫的線上維護操作

高可用性與資料保護

- 支援複製和鏡像
- 可將關鍵資料分配到冗餘儲存設備的檔案群組
- 特定檔案群組可設為唯讀，保護重要資料

建立與管理檔案群組

建立檔案群組

```
-- 在建立資料庫時定義檔案群組
CREATE DATABASE ElectronicsStore
ON PRIMARY
(
    NAME = 'ElectronicsStore_Primary',
    FILENAME = 'D:\SQLData\ElectronicsStore_Primary.mdf',
    SIZE = 100MB
),
FILEGROUP Products_FG -- 建立產品資料專用的檔案群組
(
    NAME = 'ElectronicsStore_Products',
    FILENAME = 'E:\SQLData\ElectronicsStore_Products.ndf',
    SIZE = 200MB
),
FILEGROUP Orders_FG -- 建立訂單資料專用的檔案群組
(
    NAME = 'ElectronicsStore_Orders',
    FILENAME = 'F:\SQLData\ElectronicsStore_Orders.ndf',
    SIZE = 300MB
)
LOG ON
(
    NAME = 'ElectronicsStore_Log',
    FILENAME = 'G:\SQLLogs\ElectronicsStore_Log.ldf',
    SIZE = 100MB
);
```

新增檔案群組到現有資料庫

```
-- 先新增檔案群組
ALTER DATABASE ElectronicsStore
ADD FILEGROUP Archive_FG;

-- 再將檔案加入檔案群組
ALTER DATABASE ElectronicsStore
ADD FILE
(
    NAME = 'ElectronicsStore_Archive',
    FILENAME = 'H:\SQLData\ElectronicsStore_Archive.ndf',
    SIZE = 500MB
)
TO FILEGROUP Archive_FG;
```

設定默認檔案群組

```
-- 變更默認檔案群組
ALTER DATABASE ElectronicsStore
MODIFY FILEGROUP Products_FG DEFAULT;
```

設定唯讀檔案群組

```
-- 設置檔案群組為唯讀
ALTER DATABASE ElectronicsStore
MODIFY FILEGROUP Archive_FG READONLY;

-- 恢復為可讀寫
ALTER DATABASE ElectronicsStore
MODIFY FILEGROUP Archive_FG READWRITE;
```

在檔案群組上建立物件

在指定檔案群組上建立資料表

```
-- 建立資料表在特定檔案群組上
CREATE TABLE Products
(
    ProductID int PRIMARY KEY,
    ProductName nvarchar(100),
    UnitPrice money
)
ON Products_FG; -- 指定檔案群組
```

在指定檔案群組上建立索引

```
-- 建立索引在特定檔案群組上
CREATE NONCLUSTERED INDEX IX_Products_ProductName
ON Products(ProductName)
ON Orders_FG; -- 索引位於與資料表不同的檔案群組
```

檔案群組的進階應用

資料分割 (Partitioning)

```
-- 建立分割函數
CREATE PARTITION FUNCTION OrderDateRangePF (datetime)
AS RANGE RIGHT FOR VALUES
('2025-01-01', '2025-04-01', '2025-07-01', '2025-10-01');

-- 建立分割方案，映射分割到不同檔案群組
CREATE PARTITION SCHEME OrderDateRangePS
AS PARTITION OrderDateRangePF
```

```
TO (Orders_FG, Products_FG, PRIMARY, Archive_FG, Orders_FG);

-- 使用分割方案建立資料表
CREATE TABLE Orders
(
    OrderID int,
    OrderDate datetime,
    CustomerID int,
    -- 其他欄位
    PRIMARY KEY (OrderID, OrderDate)
)
ON OrderDateRangePS(OrderDate); -- 根據日期分配到不同檔案群組
```

檔案群組備份策略

```
-- 備份特定檔案群組
BACKUP DATABASE ElectronicsStore
FILEGROUP = 'Products_FG'
TO DISK = 'D:\Backup\ElectronicsStore_Products.bak';
```

檔案群組規劃的最佳實踐

效能考量

1. 熱門資料與冷門資料分離
 - 將頻繁存取的資料 (如當前訂單) 與較少存取的資料 (如歷史訂單) 分開
 - 熱門資料放在較快的儲存裝置上
2. 索引與資料分離
 - 將資料表與其索引放在不同檔案群組可改善查詢性能
 - 尤其適用於大型資料表的索引
3. I/O 負載平衡
 - 根據讀寫模式分配檔案群組
 - 例如：交易密集的訂單資料與靜態的產品資料分開

資料管理考量

1. 按資料生命週期分組
 - 當前資料、近期歷史資料、封存資料可分別存放
 - 便於按需求實施不同的備份和維護策略
2. 按業務功能分組
 - 將相關的業務功能資料放在一起
 - 例如：產品資料、庫存資料、訂單資料分開管理
3. 資料移動與封存
 - 將舊資料從活躍檔案群組移至封存檔案群組

- 封存檔案群組可設為唯讀，提高安全性