

資料庫檔案與檔案群組

在 SQL Server (包括 SQL Server 2022) 中，**資料庫檔案**、**檔案群組**和**分頁**是資料庫儲存結構的核心概念，它們共同決定了資料如何在磁碟上組織、存取和管理。以下我將詳細說明這三者的定義、用途，並提供實際範例來說明它們的應用。

1. 資料庫檔案 (Database Files)

說明

資料庫檔案是 SQL Server 用來儲存資料庫內容的物理檔案，這些檔案儲存在伺服器的檔案系統中。每個 SQL Server 資料庫至少包含兩種類型的檔案：

- **主要資料檔案 (.mdf) :**
 - 儲存資料庫的資料、結構 (如資料表、索引) 和元資料。
 - 每個資料庫只有一個主要資料檔案。
- **日誌檔案 (.ldf) :**
 - 儲存交易日誌，記錄資料庫的所有變更，用於確保資料一致性和支援還原操作。
 - 一個資料庫可以有一個或多個日誌檔案。
- **次要資料檔案 (.ndf) (可選) :**
 - 用於分散資料儲存，通常在多個檔案群組中使用。
 - 可有多個次要資料檔案。

特性

- 每個檔案都有**邏輯名稱** (用於資料庫內部參考) 和**物理路徑** (檔案系統中的實際位置) 。
- 檔案的屬性包括初始大小 (`SIZE`) 、最大大小 (`MAXSIZE`) 和自動增長增量 (`FILEGROWTH`) 。
- 檔案的儲存位置和配置影響資料庫的效能和可擴展性。

範例

創建一個名為 `SalesDB` 的資料庫，包含一個主要資料檔案和一個日誌檔案：

```
CREATE DATABASE SalesDB
ON
(
    NAME = SalesDB_Data,
    FILENAME = 'D:\SQLData\SalesDB.mdf',
    SIZE = 100MB,
    MAXSIZE = 1GB,
    FILEGROWTH = 50MB
)
LOG ON
(
    NAME = SalesDB_Log,
```

```
FILENAME = 'E:\SQLLogs\SalesDB_Log.ldf',  
SIZE = 50MB,  
MAXSIZE = UNLIMITED,  
FILEGROWTH = 10MB  
);
```

說明：

- 主要資料檔案 `SalesDB.mdf` 位於 `D:\SQLData`，初始大小 100MB，最大 1GB，每次增長 50MB。
- 日誌檔案 `SalesDB_Log.ldf` 位於 `E:\SQLLogs`，初始大小 50MB，無最大大小限制，每次增長 10MB。

2. 檔案群組 (Filegroups)

說明

檔案群組是資料庫中資料檔案的邏輯容器，用於組織和管理多個資料檔案。檔案群組允許將資料表、索引等物件分配到特定的檔案群組，從而提高效率、可管理性和可擴展性。

類型

- **主要檔案群組 (Primary Filegroup)：**
 - 包含主要資料檔案 (`.mdf`) 和系統表 (如資料庫目錄)。
 - 預設情況下，所有資料表和索引都儲存在主要檔案群組，除非指定其他檔案群組。
- **使用者定義檔案群組 (User-Defined Filegroups)：**
 - 由使用者創建，用於存放次要資料檔案 (`.ndf`)。
 - 可用於分散 I/O 負載或將特定資料表分區到不同磁碟。
- **預設檔案群組：**
 - 如果未明確指定，新的資料表和索引會儲存在預設檔案群組 (通常是主要檔案群組，但可以更改)。

用途

- **效能優化：**將經常存取的資料表分配到快速磁碟上的檔案群組。
- **資料分區：**將大型資料表分區到多個檔案群組，支援並行處理。
- **備份管理：**支援檔案群組級別的備份和還原，減少備份時間。

範例

創建一個資料庫 `InventoryDB`，包含主要檔案群組和一個使用者定義檔案群組：

```
CREATE DATABASE InventoryDB  
ON PRIMARY  
(  
    NAME = InventoryDB_Primary,  
    FILENAME = 'D:\SQLData\InventoryDB.mdf',  
    SIZE = 100MB,  
    MAXSIZE = 1GB,  
    FILEGROWTH = 50MB
```

```

),
(
    NAME = InventoryDB_Secondary,
    FILENAME = 'E:\SQLData\InventoryDB_Secondary.ndf',
    SIZE = 200MB,
    MAXSIZE = 2GB,
    FILEGROWTH = 100MB
)
LOG ON
(
    NAME = InventoryDB_Log,
    FILENAME = 'F:\SQLLogs\InventoryDB_Log.ldf',
    SIZE = 50MB,
    MAXSIZE = UNLIMITED,
    FILEGROWTH = 10MB
);

-- 將使用者定義檔案群組設為預設
ALTER DATABASE InventoryDB
MODIFY FILEGROUP [PRIMARY] DEFAULT;

-- 創建資料表並指定檔案群組
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName NVARCHAR(100),
    Price DECIMAL(10, 2)
) ON [PRIMARY];

```

說明：

- 資料庫包含兩個檔案群組：`PRIMARY`（包含 `InventoryDB.mdf`）和一個次要檔案群組（包含 `InventoryDB_Secondary.ndf`）。
- `Products` 資料表明確指定儲存在 `PRIMARY` 檔案群組。
- 可通過 `ALTER DATABASE` 將預設檔案群組更改為其他檔案群組。

3. 分頁 (Pages)

說明

分頁是 SQL Server 儲存資料的基本單位。所有資料（包括資料表、索引和元資料）都儲存在固定大小為 **8 KB** 的分頁中。分頁是資料庫檔案內部的邏輯結構，SQL Server 通過分頁來管理磁碟空間和資料存取。

分頁類型

- **資料分頁 (Data Pages)**：儲存資料表中的資料列。
- **索引分頁 (Index Pages)**：儲存索引的鍵值和指標。
- **文字/圖片分頁 (Text/Image Pages)**：儲存大型物件資料（如 `VARCHAR(MAX)`、`NVARCHAR(MAX)`、`VARBINARY(MAX)`）。
- **全域配置圖 (GAM, Global Allocation Map)**：追蹤哪些分頁已被分配。

- 共享全域配置圖 (SGAM, Shared Global Allocation Map)：追蹤混合分頁的使用情況。
- 索引配置圖 (IAM, Index Allocation Map)：追蹤特定物件的分頁分配。
- 差異變更圖 (DCM, Differential Changed Map)：記錄自上次完整備份後變更的分頁。
- 大量變更圖 (BCM, Bulk Changed Map)：記錄大量操作（如 `BULK INSERT`）影響的分頁。

分頁組織

- 範圍 (Extents)：分頁以 8 個連續分頁為一組，稱為範圍 (64 KB)。
 - 統一範圍 (Uniform Extents)：屬於單一物件。
 - 混合範圍 (Mixed Extents)：可由多個物件共享，適合小型物件。
- SQL Server 使用分頁和範圍來有效分配和回收磁碟空間。

用途

- 資料存取：SQL Server 通過分頁讀取和寫入資料，減少磁碟 I/O。
- 空間管理：分頁結構允許高效的空間分配和碎片管理。
- 效能優化：索引分頁的設計影響查詢效能，適當的索引維護（如重建索引）可減少分頁碎片。

範例

檢查資料庫中某個資料表的分頁使用情況：

```
-- 創建資料表
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    OrderDate DATE,
    CustomerID INT,
    Amount DECIMAL(10, 2)
);

-- 插入範例資料
INSERT INTO Orders (OrderID, OrderDate, CustomerID, Amount)
VALUES (1, '2025-04-01', 101, 500.00),
       (2, '2025-04-02', 102, 750.00);

-- 使用 DBCC 檢查分頁
DBCC SHOWCONTIG ('Orders');

-- 使用動態管理檢視查詢分頁分配
SELECT
    OBJECT_NAME(object_id) AS TableName,
    index_id,
    page_count,
    avg_fragmentation_in_percent
FROM sys.dm_db_index_physical_stats(DB_ID(), OBJECT_ID('Orders'), NULL, NULL, 'LIMITED');
```

說明：

- `DBCC SHOWCONTIG` 顯示資料表的分頁和碎片資訊，幫助診斷效能問題。

- `sys.dm_db_index_physical_stats` 動態管理檢視提供分頁數量和碎片百分比，適用於索引維護。
- 當資料表增長或頻繁更新時，分頁可能會出現碎片，需使用 `ALTER INDEX REBUILD` 或 `REORGANIZE` 進行優化。

綜合應用範例

假設你是一家電子商務公司的資料庫管理員，需要設計一個高效的資料庫來儲存產品和訂單資料，並利用檔案群組分散儲存負載，同時監控分頁使用情況。

T-SQL 腳本

```
-- 創建資料庫，包含主要檔案群組和兩個使用者定義檔案群組
CREATE DATABASE ECommerceDB
ON PRIMARY
(
    NAME = ECommerceDB_Primary,
    FILENAME = 'D:\SQLData\ECommerceDB.mdf',
    SIZE = 100MB,
    MAXSIZE = 1GB,
    FILEGROWTH = 50MB
),
FILEGROUP ProductFG
(
    NAME = Product_Data,
    FILENAME = 'E:\SQLData\Product_Data.ndf',
    SIZE = 200MB,
    MAXSIZE = 2GB,
    FILEGROWTH = 100MB
),
FILEGROUP OrderFG
(
    NAME = Order_Data,
    FILENAME = 'F:\SQLData\Order_Data.ndf',
    SIZE = 300MB,
    MAXSIZE = 3GB,
    FILEGROWTH = 150MB
)
LOG ON
(
    NAME = ECommerceDB_Log,
    FILENAME = 'G:\SQLLogs\ECommerceDB_Log.ldf',
    SIZE = 50MB,
    MAXSIZE = UNLIMITED,
    FILEGROWTH = 20MB
);

-- 創建產品資料表，儲存在 ProductFG
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName NVARCHAR(100),
    Price DECIMAL(10, 2),
```

```

        Stock INT
    ) ON ProductFG;

-- 創建訂單資料表，儲存在 OrderFG
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    OrderDate DATE,
    CustomerID INT,
    Amount DECIMAL(10, 2)
) ON OrderFG;

-- 插入範例資料
INSERT INTO Products (ProductID, ProductName, Price, Stock)
VALUES (1, 'Laptop', 1200.50, 50),
       (2, 'Smartphone', 800.75, 100);

INSERT INTO Orders (OrderID, OrderDate, CustomerID, Amount)
VALUES (1, '2025-04-01', 101, 1200.50),
       (2, '2025-04-02', 102, 800.75);

-- 檢查分頁和碎片
SELECT
    OBJECT_NAME(object_id) AS TableName,
    index_id,
    page_count,
    avg_fragmentation_in_percent
FROM sys.dm_db_index_physical_stats(DB_ID('ECommerceDB'), NULL, NULL, NULL, 'LIMITED')
WHERE OBJECT_NAME(object_id) IN ('Products', 'Orders');

-- 優化索引（如果碎片過高）
ALTER INDEX ALL ON Products REBUILD;
ALTER INDEX ALL ON Orders REBUILD;

```

說明

1. 資料庫檔案：

- 資料庫包含三個資料檔案：**ECommerceDB.mdf**（主要檔案）、**Product_Data.ndf** 和 **Order_Data.ndf**（次要檔案），以及一個日誌檔案 **ECommerceDB_Log.ldf**。
- 每個檔案配置了不同的初始大小、最大大小和增長增量，適應不同的儲存需求。

2. 檔案群組：

- PRIMARY** 檔案群組儲存系統表和元資料。
- ProductFG** 用於儲存產品相關資料，位於 **E:\SQLData** 的快速磁碟。
- OrderFG** 用於儲存訂單資料，位於 **F:\SQLData** 的高容量磁碟。
- 這種分離提高了 I/O 效能，並方便針對特定檔案群組進行備份。

3. 分頁：

- 使用 **sys.dm_db_index_physical_stats** 檢查 **Products** 和 **Orders** 資料表的分頁分配和碎片情況。
- 如果碎片比例 (**avg_fragmentation_in_percent**) 過高，執行 **ALTER INDEX REBUILD** 重新組織分頁，減少磁碟 I/O 開銷。

注意事項

1. 資料庫檔案：

- 確保檔案路徑存在，且 SQL Server 服務帳戶有寫入權限。
- 分散資料和日誌檔案到不同磁碟（例如 SSD 和 HDD），以提高效能。
- 避免過小的 `FILEGROWTH` 設定，以減少頻繁的檔案擴展。

2. 檔案群組：

- 使用多個檔案群組可提高並行處理能力，特別適用於大型資料庫。
- 定期備份特定檔案群組，減少完整備份的時間和資源消耗。
- 使用 `ALTER DATABASE` 動態新增檔案或檔案群組，支援資料庫擴展。

3. 分頁：

- 定期檢查分頁碎片，特別是對經常更新的資料表。
- 使用 `ALTER INDEX REORGANIZE` 進行輕量級碎片整理，或 `ALTER INDEX REBUILD` 進行徹底重建。
- 避免過多的小型範圍（混合範圍），以減少空間管理的開銷。

4. SQL Server 2022 特性：

- SQL Server 2022 支援更高效的檔案管理和分頁分配，特別是在雲端混合環境中。
- 新的 `sys.dm_db_page_info` 函數可詳細檢查單一分頁的內容，幫助診斷問題：

```
SELECT * FROM sys.dm_db_page_info(DB_ID(), 1, 1, 'DETAILED');
```