

資料庫創建

在 SQL Server 2022 中，`ON`，`LOG ON`，`COLLATE`，和 `FOR ATTACH` 是與資料庫創建、管理及配置相關的重要關鍵字。以下我將逐一說明這些關鍵字的含義，並提供實際應用的範例，幫助你更好地理解它們在 SQL Server 中的用途。

1. ON

說明

`ON` 是用於指定資料庫主要檔案組 (Primary Filegroup) 中資料檔案的存放位置和屬性。它定義了資料庫的 `.mdf` (主要資料檔案) 或 `.ndf` (次要資料檔案) 的物理位置、大小及成長方式。

語法

```
ON ( NAME = logical_file_name, FILENAME = 'os_file_name', SIZE = size, MAXSIZE = max_size, FILEGROWTH = growth_increment )
```

- **NAME:** 檔案的邏輯名稱，用於資料庫內部參考。
- **FILENAME:** 檔案在作業系統中的物理路徑和名稱。
- **SIZE:** 檔案的初始大小 (通常以 MB 或 GB 為單位)。
- **MAXSIZE:** 檔案的最大大小 (可設為 UNLIMITED)。
- **FILEGROWTH:** 檔案自動增長的增量 (以 MB 或百分比表示)。

應用實例

假設你想創建一個名為 `MyDatabase` 的資料庫，並指定其主要資料檔案存放於 `D:\SQLData` 目錄，初始大小為 100MB，最大不超過 1GB，每次增長 50MB：

```
CREATE DATABASE MyDatabase
ON
(
    NAME = MyDatabase_Data,
    FILENAME = 'D:\SQLData\MyDatabase.mdf',
    SIZE = 100MB,
    MAXSIZE = 1GB,
    FILEGROWTH = 50MB
);
```

說明：這個範例創建了一個資料庫，資料檔案位於 `D:\SQLData\MyDatabase.mdf`，初始分配 100MB 空間，當空間不足時，每次自動增長 50MB，但總大小不會超過 1GB。

2. LOG ON

說明

LOG ON 用於指定資料庫交易日誌檔案 (**.ldf**) 的存放位置和屬性。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。

語法

```
LOG ON ( NAME = logical_file_name, FILENAME = 'os_file_name', SIZE = size, MAXSIZE = max_size, FILEGROWTH = growth_increment )
```

- 參數含義與 **ON** 類似，但適用於日誌檔案。

應用實例

延續上面的範例，假設你希望將 **MyDatabase** 的日誌檔案存放在 **E:\SQLLogs**，初始大小為 50MB，無大小限制，每次增長 10MB：

```
CREATE DATABASE MyDatabase
ON
(
    NAME = MyDatabase_Data,
    FILENAME = 'D:\SQLData\MyDatabase.mdf',
    SIZE = 100MB,
    MAXSIZE = 1GB,
    FILEGROWTH = 50MB
)
LOG ON
(
    NAME = MyDatabase_Log,
    FILENAME = 'E:\SQLLogs\MyDatabase_Log.ldf',
    SIZE = 50MB,
    MAXSIZE = UNLIMITED,
    FILEGROWTH = 10MB
);
```

說明：這個範例為 **MyDatabase** 配置了一個日誌檔案，存放在 **E:\SQLLogs\MyDatabase_Log.ldf**，初始大小 50MB，無最大大小限制，當空間不足時每次增長 10MB。

3. COLLATE

說明

COLLATE 用於指定資料庫、欄位或表達式的排序規則 (Collation)，定義如何比較和排序字元資料。排序規則決定了字元的大小寫敏感性、語音敏感性以及區域語言的排序方式。

- 層級：**COLLATE** 可應用於伺服器、資料庫、欄位或查詢層級。
- 常見選項：

- **CI**: Case Insensitive (大小寫不敏感) 。
- **CS**: Case Sensitive (大小寫敏感) 。
- **AI**: Accent Insensitive (語音不敏感) 。
- **AS**: Accent Sensitive (語音敏感) 。
- **BIN**: 二進位排序 。

語法

```
COLLATE collation_name
```

- **collation_name**: 可以是 Windows 排序規則 (如 `Latin1_General_CI_AS`) 或 SQL 排序規則 (如 `SQL_Latin1_General_CP1_CI_AS`) 。

應用實例

範例 1：創建資料庫時指定排序規則

假設你需要一個對中文排序敏感的資料庫，使用 `Chinese_Taiwan_Stroke_CS_AS` 排序規則：

```
CREATE DATABASE ChineseDB
COLLATE Chinese_Taiwan_Stroke_CS_AS
ON
(
    NAME = ChineseDB_Data,
    FILENAME = 'D:\SQLData\ChineseDB.mdf',
    SIZE = 100MB
)
LOG ON
(
    NAME = ChineseDB_Log,
    FILENAME = 'E:\SQLLogs\ChineseDB_Log.ldf',
    SIZE = 20MB
);
```

說明：這個資料庫的預設排序規則為 `Chinese_Taiwan_Stroke_CS_AS`，適用於所有字元欄位，確保中文資料按筆畫排序，且大小寫敏感。

範例 2：查詢中解決排序規則衝突

假設你有兩個資料庫，排序規則不同，當進行聯結 (JOIN) 時可能會發生衝突。你可以在查詢中使用 `COLLATE` 來統一排序規則：

```
SELECT *
FROM Database1.dbo.Table1 t1
JOIN Database2.dbo.Table2 t2
    ON t1.Name COLLATE Latin1_General_CI_AS = t2.Name COLLATE Latin1_General_CI_AS;
```

說明：這個查詢確保 `Name` 欄位的比較使用 `Latin1_General_CI_AS` 排序規則，避免因排序規則不同而導致的錯誤。

4. FOR ATTACH

說明

FOR ATTACH 用於將現有的資料庫檔案 (**.mdf** 和 **.ldf**) 附加到 SQL Server 實例中，而不需要重新創建資料庫。通常用於資料庫遷移或從備份檔案恢復資料庫。

語法

```
CREATE DATABASE database_name
ON ( FILENAME = 'os_file_name' )
[ , ...n ]
LOG ON ( FILENAME = 'os_file_name' )
[ , ...n ]
FOR ATTACH;
```

- **FILENAME:** 現有資料庫檔案的物理路徑。
- **FOR ATTACH:** 指示 SQL Server 附加現有檔案，而不是創建新檔案。

應用實例

假設你有一個已存在的資料庫檔案，位於 **D:\SQLData\OldDatabase.mdf** 和 **E:\SQLLogs\OldDatabase_Log.ldf**，你希望將其附加到 SQL Server 2022：

```
CREATE DATABASE OldDatabase
ON
(
    FILENAME = 'D:\SQLData\OldDatabase.mdf'
)
LOG ON
(
    FILENAME = 'E:\SQLLogs\OldDatabase_Log.ldf'
)
FOR ATTACH;
```

說明：這個命令將現有的 **OldDatabase** 資料庫檔案附加到 SQL Server，使其成為可用的資料庫。如果日誌檔案遺失，可以使用 **FOR ATTACH_REBUILD_LOG** 重建日誌。

綜合應用實例

假設你是一家公司的資料庫管理員，需要為一個新專案創建一個資料庫，同時需要遷移一個舊資料庫。以下是一個綜合應用 **ON**，**LOG ON**，**COLLATE**，和 **FOR ATTACH** 的場景：

情境

- **新資料庫**：創建一個名為 `ProjectDB` 的資料庫，使用 `Latin1_General_CI_AS` 排序規則，資料檔案存放在 `D:\SQLData`，日誌檔案存放在 `E:\SQLLogs`。
- **舊資料庫**：將一個名為 `LegacyDB` 的舊資料庫檔案附加到伺服器，檔案位於 `F:\Backup`。

T-SQL 腳本

```
-- 創建新資料庫 ProjectDB
CREATE DATABASE ProjectDB
COLLATE Latin1_General_CI_AS
ON
(
    NAME = ProjectDB_Data,
    FILENAME = 'D:\SQLData\ProjectDB.mdf',
    SIZE = 200MB,
    MAXSIZE = 2GB,
    FILEGROWTH = 100MB
)
LOG ON
(
    NAME = ProjectDB_Log,
    FILENAME = 'E:\SQLLogs\ProjectDB_Log.ldf',
    SIZE = 50MB,
    MAXSIZE = UNLIMITED,
    FILEGROWTH = 20MB
);

-- 附加舊資料庫 Legacy-
CREATE DATABASE LegacyDB
ON
(
    FILENAME = 'F:\Backup\LegacyDB.mdf'
)
LOG ON
(
    FILENAME = 'F:\Backup\LegacyDB_Log.ldf'
)
FOR ATTACH;
```

說明：

- `ProjectDB` 是一個新創建的資料庫，配置了特定的排序規則和檔案屬性，適合新專案的需求。
- `LegacyDB` 是通過附加現有檔案恢復的舊資料庫，無需重新創建結構，適用於資料遷移場景。

注意事項

1. ON 和 LOG ON：

- 確保指定的檔案路徑在伺服器上存在且 SQL Server 服務帳戶有寫入權限。

- 合理規劃 `SIZE`、`MAXSIZE` 和 `FILEGROWTH`，以避免頻繁的檔案增長影響效能。

2. COLLATE：

- 選擇適合應用程式語言和區域的排序規則。例如，中文環境可能需要 `Chinese_Taiwan_Stroke_CS_AS`。
- 在跨資料庫查詢時，注意排序規則衝突，必要時使用 `COLLATE` 統一。

3. FOR ATTACH：

- 附加資料庫前，確保檔案未被其他 SQL Server 實例使用。
- 如果日誌檔案損壞或遺失，使用 `FOR ATTACH_REBUILD_LOG` 可能導致資料不一致，需謹慎操作。

4. SQL Server 2022 特性：

- SQL Server 2022 支援更高效的檔案管理和排序規則處理，特別是在雲端混合環境 (如 Azure SQL Managed Instance) 中。
- 使用 `sys.fn_helpcollations()` 查詢所有可用排序規則，確保選擇正確的配置。