

T-SQL 簡介

T-SQL (Transact-SQL) 是 Microsoft SQL Server 使用的結構化查詢語言 (SQL) 的擴展版本。它基於標準 SQL (ANSI/ISO)，但增加了許多專有功能，使其更適合處理複雜的資料庫操作、管理和程式設計。T-SQL 是與 SQL Server 互動的主要語言，用於執行資料查詢、資料庫管理、程式邏輯控制等任務。

以下將詳細介紹 T-SQL 的核心特性、功能、語法結構，並提供應用範例，幫助你理解其在 SQL Server 2022 中的應用。

T-SQL 的核心特性

1. 標準 SQL 基礎：

- T-SQL 支援標準 SQL 的核心功能，如 **SELECT**、**INSERT**、**UPDATE**、**DELETE** 等，用於資料操作。
- 支援資料定義語言 (DDL)，如 **CREATE**、**ALTER**、**DROP**，用於管理資料庫結構。
- 支援資料控制語言 (DCL)，如 **GRANT**、**REVOKE**，用於權限管理。

2. 程式設計功能：

- 提供流程控制結構，如 **IF...ELSE**、**WHILE** 迴圈、錯誤處理 (**TRY...CATCH**)。
- 支援變數宣告 (**DECLARE**)、儲存程序 (Stored Procedures)、觸發器 (Triggers) 和使用者定義函數 (User-Defined Functions)。
- 支援批次處理 (Batch) 和交易 (Transaction) 管理，確保資料一致性。

3. 效能優化：

- 提供進階查詢功能，如視窗函數 (Window Functions)、公用表達式 (CTEs)、和動態 SQL。
- 支援索引管理、查詢計畫分析 (Execution Plan) 和效能調校工具。

4. SQL Server 專有功能：

- 支援 SQL Server 2022 的新功能，如增強的智慧查詢處理 (Intelligent Query Processing)、Ledger 區塊鏈技術、和 Azure 整合。
- 提供系統檢視表 (System Views) 和動態管理檢視 (DMVs)，用於監控和診斷伺服器狀態。

5. 跨平台支援：

- T-SQL 可在 SQL Server 的多個部署環境中使用，包括本機伺服器、Azure SQL Database、Azure SQL Managed Instance 等。

T-SQL 的主要組成部分

1. 資料操作語言 (DML)

用於操作資料庫中的資料：

- **SELECT**: 查詢資料。
- **INSERT**: 新增資料。
- **UPDATE**: 修改資料。
- **DELETE**: 刪除資料。

範例：

```
-- 查詢員工資料
SELECT EmployeeID, FirstName, LastName
FROM Employees
WHERE Department = 'IT';

-- 新增員工記錄
INSERT INTO Employees (FirstName, LastName, Department)
VALUES ('John', 'Doe', 'IT');

-- 更新員工部門
UPDATE Employees
SET Department = 'HR'
WHERE EmployeeID = 1;

-- 刪除員工記錄
DELETE FROM Employees
WHERE EmployeeID = 1;
```

2. 資料定義語言 (DDL)

用於定義和管理資料庫結構：

- **CREATE:** 創建資料庫、資料表、索引等。
- **ALTER:** 修改現有物件。
- **DROP:** 刪除物件。

範例：

```
-- 創建資料表
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName NVARCHAR(100),
    Price DECIMAL(10, 2),
    Stock INT
);

-- 修改資料表，新增欄位
ALTER TABLE Products
ADD Category NVARCHAR(50);

-- 刪除資料表
DROP TABLE Products;
```

3. 流程控制

T-SQL 提供程式設計結構，用於實現邏輯控制：

- **IF...ELSE:** 條件判斷。
- **WHILE:** 迴圈。

- **BEGIN...END**: 程式區塊。

範例：

```
DECLARE @Stock INT = 10;

IF @Stock > 0
BEGIN
    PRINT 'Product is in stock.';
END
ELSE
BEGIN
    PRINT 'Product is out of stock.';
END
```

4. 錯誤處理

使用 `TRY...CATCH` 捕捉和處理錯誤。

範例：

```
BEGIN TRY
    -- 嘗試除以零
    SELECT 1 / 0;
END TRY
BEGIN CATCH
    SELECT
        ERROR_NUMBER() AS ErrorNumber,
        ERROR_MESSAGE() AS ErrorMessage;
END CATCH;
```

5. 儲存程序

儲存程序是一組預編譯的 T-SQL 語句，可重複使用，提高效能和安全性。

範例：

```
-- 創建儲存程序
CREATE PROCEDURE GetEmployeeByID
    @EmployeeID INT
AS
BEGIN
    SELECT FirstName, LastName, Department
    FROM Employees
    WHERE EmployeeID = @EmployeeID;
END;

-- 執行儲存程序
EXEC GetEmployeeByID @EmployeeID = 1;
```

6. 視窗函數

視窗函數用於對資料集進行分組計算，如排名、累計總和等。

範例：

```
-- 計算每個部門的員工薪資排名
SELECT
    FirstName,
    LastName,
    Department,
    Salary,
    RANK() OVER (PARTITION BY Department ORDER BY Salary DESC) AS SalaryRank
FROM Employees;
```

7. 交易管理

T-SQL 使用 `BEGIN TRANSACTION`、`COMMIT` 和 `ROLLBACK` 管理資料一致性。

範例：

```
BEGIN TRANSACTION;
BEGIN TRY
    UPDATE Accounts
    SET Balance = Balance - 100
    WHERE AccountID = 1;

    UPDATE Accounts
    SET Balance = Balance + 100
    WHERE AccountID = 2;

    COMMIT;
END TRY
BEGIN CATCH
    ROLLBACK;
    PRINT 'Transaction failed: ' + ERROR_MESSAGE();
END CATCH;
```

T-SQL 在 SQL Server 2022 中的新功能

SQL Server 2022 引入了多項增強功能，T-SQL 也隨之擴展，支援以下新特性：

1. 時間點還原 (Point-in-Time Restore)：

- T-SQL 支援更精細的日誌還原，允許將資料庫還原到特定時間點。
- 範例：

```
RESTORE DATABASE MyDatabase
FROM DISK = 'D:\Backup\MyDatabase.bak'
WITH RECOVERY, STOPAT = '2025-04-17 10:00:00';
```

2. Ledger 功能：

- SQL Server 2022 引入了資料庫 Ledger，支援不可變的資料記錄，適用於區塊鏈式應用。
- 範例：

```
CREATE TABLE LedgerTable (  
    ID INT PRIMARY KEY,  
    TransactionData NVARCHAR(100)  
) WITH (LEDGER = ON);
```

3. 增強的智慧查詢處理：

- T-SQL 查詢可利用記憶體優化 (Memory-Optimized Tables) 和參數化查詢記憶體授權 (Parameter-Sensitive Plan Optimization)。
- 範例：

```
CREATE TABLE MemoryOptimizedTable (  
    ID INT PRIMARY KEY NONCLUSTERED,  
    Name NVARCHAR(50)  
) WITH (MEMORY_OPTIMIZED = ON);
```

4. JSON 功能增強：

- T-SQL 提供更強大的 JSON 處理功能，如 `JSON_PATH_EXISTS` 和 `JSON_MODIFY`。
- 範例：

```
DECLARE @json NVARCHAR(MAX) = '{"name": "John", "age": 30}';  
SELECT JSON_VALUE(@json, '$.name') AS Name;
```

5. APPROX_COUNT_DISTINCT：

- 提供高效的近似計數功能，適用於大規模資料分析。
- 範例：

```
SELECT APPROX_COUNT_DISTINCT(CustomerID) AS ApproxUniqueCustomers  
FROM Sales;
```

T-SQL 的應用場景

1. 資料庫管理：

- 創建和維護資料庫結構，如資料表、索引、檢視表。
- 管理使用者權限和安全性設定。

2. 資料分析：

- 撰寫複雜查詢，進行資料聚合、篩選和視窗分析。
- 配合 Power BI 或 SSRS 產生報表。

3. 自動化任務：

- 使用儲存程序和 SQL Server Agent 自動執行資料清理、備份或 ETL (Extract, Transform, Load) 流程。
4. 應用程式後端：
- 為 Web 或桌面應用程式提供資料存取邏輯，通過儲存程序或函數與前端互動。
5. 效能調校：
- 分析查詢計畫，優化索引和 T-SQL 程式碼，提升查詢效能。

學習 T-SQL 的建議

1. 熟悉基本語法：
 - 從 `SELECT`、`INSERT`、`UPDATE`、`DELETE` 開始，逐步學習 `JOIN`、子查詢和聚合函數。
2. 練習程式設計結構：
 - 掌握 `IF...ELSE`、`WHILE` 和 `TRY...CATCH`，並學習如何撰寫儲存程序和觸發器。
3. 使用官方文件：
 - Microsoft Learn 的 [Transact-SQL Reference](#) 是權威資源，涵蓋所有語法和範例。
4. 實作練習：
 - 使用 SQL Server Management Studio (SSMS) 或 Azure Data Studio，在本機或雲端環境中練習 T-SQL。
 - 試著還原範例資料庫 (如 AdventureWorks) 進行查詢練習。
5. 參與社群：
 - 加入 SQL Server 相關論壇 (如 Stack Overflow 或 Microsoft Q&A)，與其他開發者交流。

範例：綜合應用

假設你需要為一家零售公司創建一個簡單的銷售資料庫，包含資料表、儲存程序和查詢：

```
-- 創建資料庫
CREATE DATABASE RetailDB
COLLATE Latin1_General_CI_AS;

-- 創建銷售資料表
CREATE TABLE Sales (
    SaleID INT PRIMARY KEY IDENTITY(1,1),
    ProductName NVARCHAR(100),
    SaleDate DATE,
    Amount DECIMAL(10, 2)
);

-- 插入範例資料
INSERT INTO Sales (ProductName, SaleDate, Amount)
VALUES ('Laptop', '2025-04-01', 1200.50),
       ('Phone', '2025-04-02', 800.75);

-- 創建儲存程序，查詢某日期範圍的銷售總額
CREATE PROCEDURE GetSalesByDateRange
    @StartDate DATE,
```

```
@EndDate DATE

AS
BEGIN
    SELECT SUM(Amount) AS TotalSales
    FROM Sales
    WHERE SaleDate BETWEEN @StartDate AND @EndDate;
END;

-- 執行儲存程序
EXEC GetSalesByDateRange @StartDate = '2025-04-01', @EndDate = '2025-04-30';

-- 使用視窗函數計算每日銷售排名
SELECT
    ProductName,
    SaleDate,
    Amount,
    RANK() OVER (ORDER BY Amount DESC) AS SaleRank
FROM Sales;
```

說明：

- 創建一個資料庫和資料表，插入範例資料。
- 定義一個儲存程序，計算指定日期範圍的銷售總額。
- 使用視窗函數對銷售金額進行排名。

結論

T-SQL 是 SQL Server 的核心語言，結合了標準 SQL 的查詢能力與強大的程式設計功能，適用於資料庫管理、資料分析和應用程式開發。SQL Server 2022 進一步增強了 T-SQL 的功能，使其在雲端整合、大規模資料處理和安全性方面更具優勢。透過學習 T-SQL，你可以高效地管理資料庫並實現複雜的業務邏輯。

如果需要更深入的 T-SQL 主題（如觸發器、動態 SQL 或效能優化）或特定範例，歡迎告訴我！