

DECLARE 是 **SQL Server** 中用於宣告變數、資料指標或表變數的關鍵字，主要用於批次處理、預存程序或函數中，協助進行資料暫存與邏輯控制。以下是詳細說明與實際應用範例：

一、基本語法與用途

1. 宣告變數

- 語法：

```
DECLARE @變數名稱 資料型別 [= 初始值];
```

- 特性：

- 變數名稱以 `@` 開頭（如 `@Count`）。
- 支援常見資料型別（如 `INT`、`VARCHAR`、`DATETIME`）。
- 可同時宣告多個變數（用逗號分隔）。
- 未賦值時預設為 `NULL`。

範例：

```
DECLARE @EmployeeID INT = 100; -- 宣告整數變數並賦值
DECLARE @StartDate DATE, @EndDate DATE; -- 宣告多個日期變數
```

二、變數賦值與使用

1. 使用 `SET` 賦值

```
DECLARE @SalesTotal MONEY;
SET @SalesTotal = 2000000; -- 直接賦值
```

2. 使用 `SELECT` 賦值

```
DECLARE @MaxPrice DECIMAL(18,2);
SELECT @MaxPrice = MAX(Price) FROM Products; -- 從查詢結果賦值
```

3. 查詢中引用變數

```
DECLARE @Department NVARCHAR(50) = 'IT';
SELECT * FROM Employees WHERE Department = @Department;
```

三、特殊變數類型

1. 表變數 (TABLE)

- 用途：暫存小型資料集，生命週期限於批次或函數內。
- 語法：

```
DECLARE @TempTable TABLE (  
    ID INT PRIMARY KEY,  
    Name NVARCHAR(50)  
);
```

- 範例：

```
INSERT INTO @TempTable VALUES (1, 'Alice'), (2, 'Bob');  
SELECT * FROM @TempTable WHERE ID > 1;
```

2. 資料指標 (CURSOR)

- 用途：逐筆處理查詢結果。
- 語法：

```
DECLARE EmployeeCursor CURSOR FOR  
SELECT EmployeeID, Name FROM Employees;
```

- 操作流程：

```
OPEN EmployeeCursor;  
FETCH NEXT FROM EmployeeCursor INTO @EmployeeID, @Name;  
WHILE @@FETCH_STATUS = 0 BEGIN ... END  
CLOSE EmployeeCursor;  
DEALLOCATE EmployeeCursor;
```

四、進階應用

1. 預存程序中的變數

```
CREATE PROCEDURE GetEmployeeSalary  
    @EmployeeID INT  
AS  
BEGIN  
    DECLARE @Salary MONEY;  
    SELECT @Salary = Salary FROM Employees WHERE EmployeeID = @EmployeeID;  
    PRINT 'Salary: ' + CAST(@Salary AS VARCHAR);  
END;
```

2. 動態 SQL 與變數

```
DECLARE @SQL NVARCHAR(MAX);
SET @SQL = 'SELECT * FROM Orders WHERE OrderDate > ''' + CONVERT(VARCHAR, GETDATE(), 120) +
'''';
EXEC sp_executesql @SQL;
```

五、效能注意事項

- 變數與執行計畫：
使用變數作為查詢條件時（如 `WHERE Column = @Value`），可能導致執行計畫誤判，建議改用 參數化查詢（`sp_executesql`）以提高效能。
- 表變數限制：
資料量超過 1000 行時，改用 暫存表（`#TempTable`）以利用統計資訊優化查詢。

六、綜合範例

批次處理與變數

```
DECLARE @CurrentDate DATE = GETDATE();
DECLARE @LastMonthStart DATE = DATEADD(MONTH, -1, @CurrentDate);

SELECT
    OrderID,
    OrderDate,
    CASE
        WHEN OrderDate >= @LastMonthStart THEN 'Recent'
        ELSE 'Old'
    END AS OrderStatus
FROM Orders;
```

函數回傳表變數

```
CREATE FUNCTION GetHighValueProducts(@Threshold DECIMAL(18,2))
RETURNS @Result TABLE (ProductID INT, ProductName NVARCHAR(100))
AS
BEGIN
    INSERT INTO @Result
    SELECT ProductID, ProductName FROM Products WHERE Price > @Threshold;
    RETURN;
END;
```

七、常見錯誤與解決

錯誤情境	解決方案
變數未賦值導致 <code>NULL</code> 問題	宣告時設定預設值或檢查 <code>ISNULL()</code>
資料指標未正確關閉	確保執行 <code>CLOSE</code> 與 <code>DEALLOCATE</code>
表變數效能低下	改用暫存表 (<code>#TempTable</code>)

透過合理使用 `DECLARE`，可有效管理 SQL Server 中的資料流與邏輯控制，但需注意變數生命週期與效能影響。