

稽核的主要用途

1. 追蹤資料處理過程

- 用途：
 - 記錄 ETL 流程中每個步驟的執行詳情，例如處理的記錄數、執行時間、來源和目標資訊。
 - 提供資料流的透明性，方便監控和驗證資料是否正確處理。
- 實現方式：
 - **Audit Transformation**：在資料流中新增稽核欄位，例如套件名稱 (`PackageName`)、執行 ID (`ExecutionInstanceGUID`)、機器名稱 (`MachineName`) 或執行時間 (`ExecutionStartTime`)。
 - **Row Count Transformation**：統計每個資料流的記錄數，儲存到變數並記錄到稽核表。
 - **Derived Column**：新增自訂稽核欄位，例如資料來源標記 (`source = "Excel"`) 或載入時間 (`GETDATE()`)。
- 範例應用：
 - 在您之前的複雜 ETL 案例中，使用 Audit Transformation 為 `[dbo].[ConsolidatedJobs]` 的每筆記錄新增 `ExecutionInstanceGUID`，追蹤每次執行的唯一標識。
 - 使用 Row Count 記錄 Excel、SQL Server 和 CSV 來源的行數，儲存到 `[dbo].[AuditLog]`：

```
INSERT INTO [dbo].[AuditLog] (Source, RowCount, LoadDate)
VALUES ('Excel', @ExcelRows, GETDATE())
```

- 與您案例的關聯：
 - 在匯入 `[dbo].[104]` 時，記錄處理的行數和錯誤數，確保無資料遺失 (解決截斷問題)。
 - 在 `DELETE FROM [dbo].[JOBS] WHERE job_id > 1000` 後，稽核刪除的記錄數。

2. 確保資料完整性

- 用途：
 - 驗證資料在 ETL 過程中未被意外修改、遺失或損壞。
 - 比較來源與目標的記錄數，確保所有資料正確載入。
- 實現方式：
 - **Row Count Transformation**：計算來源和目標的記錄數，比較是否一致。
 - **Checksum 或 Hash**：使用腳本元件或 Derived Column 計算記錄的校驗和，驗證資料一致性。
 - **Conditional Split**：將無效記錄 (例如空值或格式錯誤) 分流到錯誤表，記錄問題詳情。
- 範例應用：
 - 在複雜 ETL 案例中，使用 Conditional Split 過濾 `[dbo].[ConsolidatedJobs]` 的無效記錄 (例如 `ISNULL(job_id)`)，並記錄到 `[dbo].[ErrorLog]`：

```
INSERT INTO [dbo].[ErrorLog] (job_id, description, error_message)
VALUES (?, ?, 'Missing job_id')
```

- 比較 Excel 和 SQL Server 來源的總行數與 `[dbo].[ConsolidatedJobs]` 的行數，確保無遺漏。
- 與您案例的關聯：
 - 在修正 `info_description` 截斷問題時，稽核截斷的記錄數，確保所有資料正確處理。
 - 在 Unicode/非 Unicode 不匹配 (`J_person` 和 `JOB_PERSON`) 時，記錄轉換成功的記錄數。

3. 符合合規性要求

- 用途：
 - 滿足法規或內部政策要求 (例如 GDPR、SOX)，記錄誰、何時、如何處理資料。
 - 提供可追溯的稽核軌跡，供審計或調查使用。
- 實現方式：
 - **Logging**：啟用 SSIS 記錄，儲存事件 (例如 `OnError`、`OnInformation`、`OnPostExecute`) 到資料庫 (`sysssislog`) 或檔案。
 - **Audit Transformation**：新增合規性相關欄位，例如執行使用者 (`UserName`) 或套件版本 (`VersionGUID`)。
 - **Custom Audit Table**：建立自訂稽核表，記錄詳細資訊 (例如來源、行數、錯誤訊息、執行時間)。
- 範例應用：
 - 在複雜 ETL 案例中，啟用 SSIS 記錄，將所有事件儲存到 `[dbo].[AuditLog]`：

```
SELECT source, message, starttime FROM dbo.sysssislog WHERE executionid = ?
```

- 使用 Audit Transformation 新增 `UserName`，記錄誰執行了套件。
- 與您案例的關聯：
 - 在新增 `job_id` 到 `[dbo].[104]` 時，記錄執行者的使用者名稱，符合合規性要求。
 - 在 Merge Transformation (需排序) 中，稽核合併的記錄數，確保符合資料整合規範。

4. 錯誤診斷與問題追蹤

- 用途：
 - 識別 ETL 流程中的錯誤 (例如資料類型不匹配、截斷、連線失敗)，並記錄詳細資訊以便快速診斷。
 - 提供錯誤記錄的上下文 (例如失敗的記錄、來源、時間)。
- 實現方式：
 - **Error Output**：在資料流元件 (例如 OLE DB Destination) 中設定錯誤輸出，將失敗記錄寫入錯誤表。
 - **Logging**：記錄 `OnError` 和 `OnWarning` 事件，包含錯誤碼和訊息。
 - **Script Component**：自訂錯誤處理邏輯，記錄特定欄位的問題 (例如超長的 `info_description`)。
- 範例應用：
 - 在複雜 ETL 案例中，若 CSV 來源包含重複記錄，使用 Conditional Split 將重複記錄寫入 `[dbo].[ErrorLog]`：

```
INSERT INTO [dbo].[ErrorLog] (job_id, description, error_message)
VALUES (?, ?, 'Duplicate job_id')
```

- 記錄 `[dbo].[104]` 匯入失敗的記錄（例如截斷錯誤）。
- 與您案例的關聯：
 - 在修正 Unicode/非 Unicode 不匹配時，記錄轉換失敗的記錄，診斷問題原因。
 - 在 Excel 輸出 TXT 時，稽核無效記錄（例如空值）。

5. 效能監控與優化

- 用途：
 - 監控 ETL 流程的執行效能（例如執行時間、處理速度）。
 - 識別瓶頸（例如排序或大型資料集處理），提供優化依據。
- 實現方式：
 - **Logging**：記錄 `OnPreExecute` 和 `OnPostExecute` 事件的時間戳，計算執行持續時間。
 - **Row Count**：統計每個資料流的記錄數，分析處理量。
 - **Custom Metrics**：使用變數記錄自訂指標（例如每秒處理的行數）。
- 範例應用：
 - 在複雜 ETL 案例中，記錄 Merge Transformation 的執行時間，檢查排序（Sort Transformation）是否造成瓶頸：

```
INSERT INTO [dbo].[AuditLog] (Component, Duration, LoadDate)
VALUES ('Merge', DATEDIFF(SECOND, @StartTime, @EndTime), GETDATE())
```

- 與您案例的關聯：
 - 在 Merge Transformation（需排序）中，稽核排序和合併的效能。
 - 在 Excel 輸出 TXT 時，記錄處理的行數和時間，優化資料流。

6. 支援資料治理與追溯

- 用途：
 - 提供資料來源、轉換邏輯和目標的完整追溯，支援資料治理。
 - 記錄資料的生命週期（例如從 Excel 到 `[dbo].[ConsolidatedJobs]` 的流向）。
- 實現方式：
 - **Audit Transformation**：新增來源標記（例如 `source = "CSV"`）和批次 ID。
 - **Derived Column**：新增自訂追溯欄位（例如 `batch_id = @BatchID`）。
 - **Version Control**：將 SSIS 套件提交到 Git，記錄變更（參考您之前的 Audit 和版本控制問題）。
- 範例應用：
 - 在複雜 ETL 案例中，為每筆記錄新增 `source` 和 `load_date`，追溯資料來源：

```
source = "Excel", load_date = GETDATE()
```

- 使用 Schema Compare 稽核 `[dbo].[ConsolidatedJobs]` 的結構變更。
- 與您案例的關聯：

- 在新增 `job_id` 時，記錄新增操作的批次 ID，支援追溯。
- 在 `DELETE FROM [dbo].[JOBS]` 時，稽核刪除的記錄來源。

SSIS 稽核的實現方式

1. Audit Transformation

- 用途：快速新增標準稽核欄位（例如 `PackageName`, `ExecutionInstanceGUID`）。
- 配置：
 - 在資料流中拖曳 **Audit Transformation**。
 - 選取所需欄位（例如 `UserName`, `ExecutionStartTime`）。
 - 將稽核欄位映射到目標表（例如 `[dbo].[ConsolidatedJobs]` 的 `source`）。
- 範例：
 - 在複雜 ETL 案例中，新增 `ExecutionInstanceGUID` 到 `[dbo].[ConsolidatedJobs]`。

2. SSIS Logging

- 用途：記錄套件執行的事件（例如錯誤、警告、執行進度）。
- 配置：
 - 在 SSIS 套件中，選擇 **SSIS > Logging**。
 - 選取提供者（例如 SQL Server，儲存到 `sysssislog`）。
 - 選取事件（例如 `OnError`, `OnInformation`, `OnPostExecute`）。
- 範例：
 - 記錄 `[dbo].[104]` 匯入的錯誤訊息：

```
SELECT source, message, starttime FROM dbo.sysssislog WHERE executionid = ?
```

3. Row Count Transformation

- 用途：統計資料流的記錄數，儲存到變數並記錄到稽核表。
- 配置：
 - 在資料流中新增 **Row Count Transformation**。
 - 定義變數（例如 `@ExcelRows`）。
 - 將變數值寫入稽核表：

```
INSERT INTO [dbo].[AuditLog] (Source, RowCount) VALUES ('Excel', @ExcelRows)
```

4. Derived Column

- **用途**：新增自訂稽核欄位（例如來源標記、時間戳記）。
- **配置**：
 - 在資料流中新增 **Derived Column Transformation**。
 - 定義欄位（例如 `source = "CSV", load_date = GETDATE()`）。
- **範例**：
 - 在 Excel 輸出 TXT 時，新增 `ExportTime`：

```
(DT_WSTR, 50)GETDATE()
```

5. Error Output

- **用途**：將錯誤記錄寫入錯誤表，記錄失敗原因。
- **配置**：
 - 在 OLE DB Destination 中設定錯誤輸出，連接到 OLE DB Destination（錯誤表）。
 - 記錄錯誤欄位和訊息（例如 `ErrorCode, ErrorColumn`）。

6. Custom Audit Table

- **用途**：儲存詳細稽核資訊，支援自訂需求。
- **範例結構**：

```
CREATE TABLE [dbo].[AuditLog] (  
    AuditID INT IDENTITY(1,1),  
    Source NVARCHAR(50),  
    RowCount INT,  
    ErrorCount INT,  
    LoadDate DATETIME,  
    ExecutionID UNIQUEIDENTIFIER,  
    ErrorMessage NVARCHAR(MAX)  
)
```