1. Executive Summary

**Purpose of the Document**  
This document outlines the structure and operational logic of a layered SQL Server-based data ingestion and reconciliation pipeline. It covers the raw ingestion of JSON payloads, transformation into structured staging tables, and reconciliation processes to ensure data accuracy and completeness across layers.

**Scope:**

* Data ingestion from source into RAW\_LAYER
* JSON parsing and transformation into STG\_LAYER
* Auditing and reconciliation between layers using AUDITING schema

**Target Audience**  
This documentation is intended for:

* Data Engineers
* Data Analysts
* QA and Test Engineers
* Database Administrators

2. Table Documentation

**Table: RAW\_LAYER.BATCH\_EXTRACT**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| batch\_id | INT | Unique identifier for the data batch |
| extraction\_date | DATETIME | Timestamp when batch was ingested |
| json\_data | NVARCHAR(MAX) | JSON payload received from source |

Represents ingested raw JSON data per batch from external systems. Populated manually or through an ETL connector.

**Table: RAW\_LAYER.ENTITY\_EXTRACT**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| batch\_id | INT | ID of the batch the entity belongs to |
| entity\_event\_id | INT | Surrogate event ID |
| entity\_id | INT | Business key (customer\_id or product\_id) |
| extraction\_date | DATETIME | Timestamp of JSON extraction |
| entity\_type | NVARCHAR(50) | The type of entity |
| entity\_json\_data | NVARCHAR(MAX) | Raw JSON data snippet for the entity |

Populated by parsing JSON arrays in **RAW\_LAYER.BATCH\_EXTRACT** using OPENJSON.

**Table: STG\_LAYER.CUSTOMERS**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| batch\_id | INT | ID of the batch this record came from |
| entity\_event\_id | INT | Unique ID for the entity event |
| customer\_id | INT | Business key for customer |
| fname | NVARCHAR(50) | First name parsed from JSON |
| lname | NVARCHAR(50) | Last name parsed from JSON |
| gender | NVARCHAR(1) | Gender value (‘M’, ‘F’) |

Structured customer data extracted from **RAW\_LAYER.ENTITY\_EXTRACT**.

**Table: STG\_LAYER.PRODUCTS**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| batch\_id | INT | ID of the batch this record came from |
| entity\_event\_id | INT | Unique ID for the entity event |
| product\_id | INT | Business key for product |
| product\_name | NVARCHAR(50) | Name of the product |
| unit\_price | INT | Unit price |

Structured product data extracted from **RAW\_LAYER.ENTITY\_EXTRACT**.

**Table: AUDITING.RECONCILIATION\_REFERENCE**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| reference\_id | INT | Unique ID for the reconciliation rule |
| load\_phase | NVARCHAR(50) | Phase e.g. ‘RAW to STG’ |
| source\_schema | NVARCHAR(50) | Schema name of the source table |
| source\_object | NVARCHAR(50) | Source table name |
| source\_column | NVARCHAR(50) | Source column for reconciliation |
| join\_query | NVARCHAR(MAX) | Join logic (optional) |
| filter\_query | NVARCHAR(MAX) | Filter condition e.g. entity\_type=‘product’ |
| target\_schema | NVARCHAR(50) | Schema of target table |
| target\_object | NVARCHAR(50) | Target table name |
| target\_column | NVARCHAR(50) | Target column name |

Configuration table for reconciliation metadata between source and staging layers.

**Table: AUDITING.RECONCILIATION\_OVERVIEW**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| audit\_id | INT | Unique audit ID |
| audit\_job\_id | INT | Grouped audit job ID |
| load\_phase | NVARCHAR(50) | Load phase name |
| source\_object | NVARCHAR(50) | Source table |
| source\_column | NVARCHAR(50) | Source column |
| target\_object | NVARCHAR(50) | Target table |
| target\_column | NVARCHAR(50) | Target column |
| raw\_batch\_id\_count | INT | Unique batches in source |
| stg\_batch\_id\_count | INT | Unique batches in target |
| batch\_id\_audit\_result | NVARCHAR(20) | Match or Mismatch |
| batch\_id\_audit\_difference | INT | Difference in batch counts |
| raw\_records\_count | INT | Count of raw records |
| stg\_records\_count | INT | Count of staged records |
| records\_audit\_result | NVARCHAR(20) | Match or Mismatch |
| records\_audit\_difference | INT | Record count difference |
| created\_date | DATETIME | Audit timestamp |

Captures high-level batch and record count comparison between source and staging.

**Table: AUDITING.RECONCILIATION\_DETAIL**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| audit\_id | INT | Unique audit ID |
| audit\_job\_id | INT | Grouped audit job ID |
| load\_phase | NVARCHAR(50) | Load phase name |
| source\_object | NVARCHAR(50) | Source table |
| source\_column | NVARCHAR(50) | Source column |
| target\_object | NVARCHAR(50) | Target table |
| target\_column | NVARCHAR(50) | Target column |
| raw\_batch\_id\_count | INT | Unique batches in source |
| stg\_batch\_id\_count | INT | Unique batches in target |
| batch\_id\_audit\_result | NVARCHAR(20) | Match or Mismatch |
| batch\_id\_audit\_difference | INT | Difference in batch counts |
| raw\_records\_count | INT | Count of raw records |
| stg\_records\_count | INT | Count of staged records |
| records\_audit\_result | NVARCHAR(20) | Match or Mismatch |
| records\_audit\_difference | INT | Record count difference |
| created\_date | DATETIME | Audit timestamp |

Tracks detailed-level mismatches and missing data between RAW and STG.

3. Stored Procedure

**Procedure: RAW\_LAYER.LOAD\_ENTITY\_EXTRACT**

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Extracts individual entities from BATCH\_EXTRACT and populates ENTITY\_EXTRACT |
| Run Frequency | On-demand or scheduled post-ingestion |
| Input | RAW\_LAYER.BATCH\_EXTRACT |
| Output | RAW\_LAYER.ENTITY\_EXTRACT |
| Key Logic | Uses OPENJSON, applies incremental load using max batch\_id and assigns event\_id |
| Assumptions | JSON is valid and well-formed |

Procedure Code: RAW\_LAYER.LOAD\_ENTITY\_EXTRACT

See script/proc\_load\_raw\_entity\_extract.sql (full code with inline comments)

**Procedure: STG\_LAYER.LOAD\_STG\_CUSTOMERS\_AND\_PRODUCTS**

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Loads structured customer and product records from ENTITY\_EXTRACT to staging tables |
| Run Frequency | Daily or on-demand |
| Input | RAW\_LAYER.ENTITY\_EXTRACT |
| Output | STG\_LAYER.CUSTOMERS, STG\_LAYER.PRODUCTS |
| Key Logic | Filters on entity\_type, extracts JSON values with JSON\_VALUE(), incremental load |
| Assumptions | Stable JSON format |

Procedure Code: STG\_LAYER.LOAD\_STG\_CUSTOMERS\_AND\_PRODUCTS

See script/ proc\_load\_stg.sql (full code with inline comments)

**Procedure: AUDITING.LOAD\_RECONCILIATION\_OVERVIEW**

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Compares source vs target batch and record counts |
| Run Frequency | After staging loads |
| Input | AUDITING.RECONCILIATION\_REFERENCE metadata |
| Output | AUDITING.RECONCILIATION\_OVERVIEW |
| Key Logic | Uses dynamic SQL to compute counts and differences |
| Assumptions | batch\_id and entity\_event\_id are present in both source and target |

This procedure performs batch- and record-level comparisons between the RAW and STG layers.

* **Batch ID Count Comparison**:
  + Counts the number of distinct batch\_id values in source and target tables. Flags mismatch if counts differ.
* **Record Count Comparison**:
  + Compares total row counts (using entity\_event\_id) between RAW and STG.
* **Match/Mismatch Determination**:
  + A “Match” is declared if both batch ID count and record count match.
  + Otherwise, it logs a “Mismatch”.
* **Audit Job Metadata**:
  + audit\_id: Unique identifier for each audit line
  + audit\_job\_id: Groups multiple rows under the same job execution
  + created\_date: Timestamp when the reconciliation was performed

The procedure dynamically builds queries based on metadata defined in AUDITING.RECONCILIATION\_REFERENCE and stores results in AUDITING.RECONCILIATION\_OVERVIEW.

Procedure Code: AUDITING.LOAD\_RECONCILIATION\_OVERVIEW

See script/proc\_load\_reconciliation\_overview.sql (full code with inline comments)

**Procedure: AUDITING.LOAD\_RECONCILIATION\_DETAIL**

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Identifies missing records in target tables |
| Run Frequency | Post-ingestion and reconciliation overview |
| Input | AUDITING.RECONCILIATION\_REFERENCE metadata |
| Output | AUDITING.RECONCILIATION\_DETAIL |
| Key Logic | LEFT JOIN between source and target on entity\_event\_id, inserts unmatched records |
| Assumptions | entity\_event\_id is reliable unique key |

This procedure identifies records in RAW that are not found in STG.

* **Missing Record Detection**:
  + Performs a LEFT JOIN between RAW and STG on entity\_event\_id
  + Records in RAW with no match in STG are flagged as “Missing in target table”
* **Dynamic SQL**:
  + Joins and filters are generated using metadata in RECONCILIATION\_REFERENCE
* **QA Utility**:
  + Helps data engineers and QA teams identify specific rows that were not loaded correctly

Results are written into AUDITING.RECONCILIATION\_DETAIL for traceability.

Procedure Code: AUDITING.LOAD\_RECONCILIATION\_DETAIL

See script/proc\_load\_reconciliation\_detail.sql (full code with inline comments)

4. Operational Guidelines

**Manual Execution Steps:**

**EXEC** RAW\_LAYER.LOAD\_ENTITY\_EXTRACT;  
**EXEC** STG\_LAYER.LOAD\_STG\_CUSTOMERS\_AND\_PRODUCTS;  
**EXEC** AUDITING.LOAD\_RECONCILIATION\_OVERVIEW;  
**EXEC** AUDITING.LOAD\_RECONCILIATION\_DETAIL;

Logs can be audited via created\_date in OVERVIEW and DETAIL tables to track run history.