ETL Process Revamp   
Design Document

Version 0.1

# Overview

## Project Background and Description

|  |  |
| --- | --- |
|  | Currently the Daily ETL Process is taking more than the acceptable time to complete successfully. Some time it is not clear why it takes more time to than acceptable time limit. This impacts the daily reporting data delivery commitment to the business. It also strains the IT resources on prolonged monitoring and hand-holding. |

## Project Scope & Solution Overview

|  |  |
| --- | --- |
|  | The scope of the engagement for this phase is limited to revamping the ETL process with minimum disruption to the current setup. The scope includes revamping the current ETL Process by a Python based customized ETL Package specific to the current business requirement (Business Area: LIABILITIES & ASSETS). The new ETL Solution must be stable, high dependability, High Data Integrity and faster than the current ETL Process time.  Solution Overview:   * Customized Python based ETL Package * The solution consists of Current System Study in Detail, Design Modification (if required), Development, Modification of Loading Strategy (if required), Unit Testing, Data Integrity & Accuracy Testing, Performance Testing, Training & Document Preparation, Deployment and Limited Post Deployment Support |

## ETL Process Design Overview

|  |  |
| --- | --- |
|  | The Design for the Revamped ETL Process has many features added to it. To over come the current pain points, the revamped ETL Process would come up with many features that would ensure seamless execution of the ETL Process with Job Statistics, Failover, Dependency Check, Data Integrity & Error Handling. |

The revamped ETL Process is classified into three sub-process. They are as follows:

1. Pre-ETL Process
2. Main ETL Process
3. Post ETL Process

The schematic representation of the three process and the sequence are mentioned here.

More details about the three processes are elucidated here.

## Pre-ETL Process

|  |  |
| --- | --- |
|  | Pre-ETL Processes is to be meant for checking the health of the entire data pipeline of the ETL Process. This process checks any potential bottlenecks or any potential data inconsistencies in the data pipeline that could be found after the daily/nightly ETL Batch Process. Some of the Pre-ETL Tasks could be to check whether the Target Tables Space and related extents; this Pre-ETL tasks can also be utilized to verify any changes in the master data tables that are to be reflected in the DWH Master Data Management. |

**Implementing the Pre-ETL Process:**

The Pre-ETL Process would be implemented with the following points (more can be added whenever it is required):

1. Why Pre-ETL Process and its Objective
2. Scope of the Pre-ETL Process
3. How to execute Pre-ETL Process
4. Time of the Pre-ETL Process Run
5. What Should be verified after the Run?
6. What-if the Objectives are not handled before the daily/nightly ETL Batch Run?

**Flow Chart for Pre-ETL Process**:

**Start**

**Check the Audit Tables Availability**

Notify

**No**

**Available**

**Yes**

**Check the Table Space & Extents in Target Zone**

**Yes**

**To Extend**

Notify

**No**

**Check the Source Master Tables**

**No**

Notify

**Change**

**Yes**

**Update the Audit Table about the Change**

**Stop**

Each of the above points are elucidated for more clarity.

1. **Why Pre-ETL Process and its Objectives**

The Pre-ETL Tasks are to prevent Daily/Nightly Jobs failures due to Target or any Intermediate/Temporary Tables Space and related extents; this Pre-ETL tasks can also be utilized to verify any change in the master data tables in the source system that are to be reflected in the DWH Master Data Management. If this were not handled properly there could be inconsistencies in the DWH that subsequently leads to in-accuracies and in-consistencies when the it is reported for end-user consumption.

1. **Scope of the Pre-ETL Process**

The scope for the Pre-ETL Process currently limited to checking any Temporary or Target Tables Space and Extents, Changes to Source Master Data. This is an evolving over the period.

1. **How to execute Pre-ETL Process**

The execution of the Pre-ETL Process is sub-divided in to many smaller tasks. The smaller tasks can be arranged in certain sequence for achieving it objectives.

1. Listing all the tables/instances for Pre-ETL Process
2. Create Audit Tables to store the meta-data
3. Check the available Table Spaces and Extents in the target table
4. Store the meta-data from Task 3 in the audit tables
5. Check any change to the Master Tables in the Source System
6. Capture the changes in the Master Data and Store ONLY the Meta Data information in the Pre-ETL Audit Tables
7. Notify the relevant stakeholders about the availability of the Table Spaces/Extents threshold breach and Changes to the Master Data in the Source System
8. Enforce the Stakeholders to do the necessary house-keeping tasks
9. Run the Pre-ETL Process once again (well before the Daily/Nightly ETL Batch) to ensure that all the housekeeping tasks are completed

Now let us get into more details about each step.

1. Listing of all tables/instances for Pre-ETL Process: First identifying all tables and instances in the target zone to check whether the table spaces and table extents are enough for next Daily/Nightly ETL Batch Run. The sample of table list can be as follows:

Table 1 (Pre-ETL Process Audit Table)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name | Source Master or Target | Observation | Observation Timestamp | Correction | Correction Timestamp | Ready for Main ETL |
| Table A | Source Master | No Change | ddmmyyyy  hhmmss |  | ddmmyyyy  hhmmss | Yes |
| Table B | Source Master | Data Insert/update | ddmmyyyy  hhmmss | DWH Master Updated | ddmmyyyy  hhmmss | Yes |
| Table C | Target Container | Nil | ddmmyyyy  hhmmss |  | ddmmyyyy  hhmmss | Yes |
| Table D | Target Container | Table Extent non-availability | ddmmyyyy  hhmmss | Table Extent Added | ddmmyyyy  hhmmss | Yes |

The above table list and other information should be available in the Target Zone area for easy access. The following are the sequential steps to achieve the Pre-ETL Process.

***Step 1.0:***

A Database Admin Level script would be fired to check the available Table Spaces and its extents. This script would be executed after the completion of the Main ETL Process. This script would be executed during the Post-ETL Process. The extracted information would be saved in a temporary table as follows:

…. Select Table Space, Table Extent, SQL Plan Cost from Table C, Table D, Table E & Table F

And Insert into Temp Table ‘Table 2’

The Temp Audit table ‘Table 2’ may have a structure like this:

Table 2 (Target Table Space & Extent Information)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table Name | Table Space % | Breached Threshold | Obs TS | Table Extent % | Breached Threshold | Obs TS | SQL Plan Cost | Breached Threshold | Obs TS | Notification Sent |
| Table C | 56 | N | DMY  HMS | 42 | N | DMY  HMS | xxx | N | DMY  HMS | N |
| Table D | 92 | Y | DMY  HMS | 64 | N | DMY  HMS | xxx | N | DMY  HMS | Y |
| Table E | 56 | N | DMY  HMS | 95 | Y | DMY  HMS | xxx | Y | DMY  HMS | Y |
| Table F | 88 | N | DMY  HMS | 35 | N | DMY  HMS | xxx | N | DMY  HMS | N |

The above Temp Audit table can be made as persistent if we need to keep the information for any future reference. The information from this Table 2 will be moved to Table 1 appropriately.

***Step 1.1***

Select Data from Table 2 that are having ‘Breached Threshold’ = Y

And Insert into Table 1

***Step 2:***

A SQL script would be fired to check any change in the source master tables. This script would be executed during Pre-ETL Process. The extracted information would be saved in a temporary table as follows:

…. Select Date Modified from Table A, Table B,

And Insert into Temp Table ‘Table 3’

The Temp Audit table ‘Table 3’ may have a structure like this:

Table 3 (Source Master Change Information Audit Table)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name | Col Name | Change Type | TS | Notification Sent |  |  |
| Table A | Nil | Nil | Nil | N |  |  |
| Table B | Column 22 | Insert | DMYHMS | Y |  |  |
| Table B | Column 17 | Update | DMYHMS | Y |  |  |
| Table B | Column 20 | Delete | DMYHMS | Y |  |  |

The above Temp Audit table can be made as persistent if we need to keep the information for any future reference. The information from this table will be moved to Table 1 appropriately.

***Step 2.1***

Select Data from Table 3 that are having ‘Col Name’ Not = ‘Nil’, ‘Change Type’ Not = ‘Nil’, ‘TS’ Not = ‘Nil’

And Insert into Table 1

Note: Need to consider whether consolidated Table Level Notification of Row Level Notification has to be sent.

***Step 2.2*** (Image of the Pre-ETL Audit Table after Step 2.1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name | Source Master or Target | Observation | Observation Timestamp | Correction | Correction Timestamp | Ready for Main ETL |
| Table A | Source Master | No Change | ddmmyyyy  hhmmss |  |  | Yes |
| Table B | Source Master | Data Insert/update | ddmmyyyy  hhmmss |  |  | No |
| Table C | Target Container | Normal | ddmmyyyy  hhmmss |  |  | Yes |
| Table D | Target Container | Table Space non-availability | ddmmyyyy  hhmmss |  |  | No |
| Table E | Target Container | SQL Plan Cost | ddmmyyyy  hhmmss |  |  | No |
| Table E | Target Container | Table Extent non-availability | ddmmyyyy  hhmmss |  |  | No |

***Step 2.3 (***After the appropriate Action has been taken)

1. Extending the Table Space
2. Adding the Table Extents
3. Fine tuning the query
4. Re-calibrating the Master Data in DWH

Execute a Query to make the ‘Pre-ETL Process Audit Table’ Ready for Main ETL Process Run.

The query could be the re-run of the Step 1.0, 1.1, 2.0 & 2.1 Queries or slightly modified to reflect the changes happened to the Table 2, Table 3 & Table 1.

After this step the ‘Pre-ETL Process Audit Table’ should look like this.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name | Source Master or Target | Observation | Observation Timestamp | Correction | Correction Timestamp | Ready for Main ETL |
| Table A | Source Master | No Change | ddmmyyyy  hhmmss | No Action | ddmmyyyy  hhmmss | Yes |
| Table B | Source Master | Data Insert/update | ddmmyyyy  hhmmss | DWH Master Updated | ddmmyyyy  hhmmss | Yes |
| Table C | Target Container | Normal | ddmmyyyy  hhmmss | No Action | ddmmyyyy  hhmmss | Yes |
| Table D | Target Container | Table Space non-availability | ddmmyyyy  hhmmss | Table Space Added | ddmmyyyy  hhmmss | Yes |
| Table E | Target Container | SQL Plan Cost | ddmmyyyy  hhmmss | SQL Fine Tuned | ddmmyyyy  hhmmss | Yes |
| Table E | Target Container | Table Extent non-availability | ddmmyyyy  hhmmss | Table Extend Added | ddmmyyyy  hhmmss | Yes |

Only If the ‘Ready for Main ETL’ column has the value = ‘’Yes’’ in all the rows of the ‘Pre-ETL Process Audit Table’, the Main ETL Process would run normally. **Otherwise, it may need a ‘’Forced’’** start. More on this in the ‘’Main ETL Process’’ Section.

## Main ETL Process

|  |  |
| --- | --- |
|  | The Main ETL Process is the important component to load the data from heterogenous sources to the Data Warehouse System after necessary Data Transformation. The Main ETL Process would perform all necessary health check of the system, Data Preparation, Data Management (if any), Data Transformation (either before the Load or after the Load). The Main ETL Process can be sub-divided into many sub-tasks. |

**Implementing the Main ETL Process:**

The Main ETL Process would be implemented with the following points to be considered (more can be added whenever it is required):

1. The Objectives of the Main ETL Process
2. Scope of the Main ETL Process
3. How to execute Main ETL Process
4. Time of the Main ETL Process
5. What Should be verified after the Main ETL Process?
6. What-if the Objectives are not handled during the daily/nightly ETL Batch Run?

Each of the above points are elucidated for more clarity.

1. **The Objectives of the Main ETL Process**

The Main ETL Process is to Load the Heterogenous Source Data into the Target Zone after the necessary Transformation. The Main ETL Process should be reliable, predictable, easy to maintain, high in performance.

1. **Scope of the Main ETL Process**

The scope for the Main ETL Process is to Extract Data from ASSETS & LIABILITIES Source System and Transform them appropriately and load into the Target Zone. The Data Mart in the Target Zone will be highly de-normalized format. The other important tasks of the Main ETL Process is to provide reliable Job Statistics, Job Status, Job Logging, Failover Mechanism, Data Integrity and Data Accuracy.

1. **How to execute Main ETL Process**

The execution of the Main ETL Process is sub-divided in to many smaller tasks. The smaller tasks can be arranged in certain sequence for achieving it objectives.

1. Listing all the Source Tables/Files meta data for Main ETL Process
2. Mapping of Source and Destination
3. Create Audit Tables to store the meta-data for the Data Extraction
4. Data Extraction Log & Audit Information
5. Staging Data Log & Audit Information
6. Staging Data – Data Accuracy & Integrity Audit Information
7. Data Load Log & Audit Information
8. Data Load – Data Accuracy & Integrity Information
9. ETL Process Completion

Details about the each of the above points is elucidated for more clarity:

**Listing all the Source Tables/Files meta data for Main ETL Process**

Listing of all the Source Tables and the associated columns need to be listed in the ETL Process Zone. The Python Wrapper would refer the List for Extracting the Data. No hard coding of the source data in the wrapper. If the Data Source is in the File Format (currently supports only CSV format), then the file Meta Data would be considered as the Source List. The Wrapper initially checks the Source List of the Tables and its Columns then Checks the Target Details through the Source-Target Mapping. If any discrepancy is found, then the Main ETL Process would not start unless and until the issue is resolved.

The Source – Target Mapping Table might Look like this: (A SQL Script Must be Prepared to insert the data into this Table).

Table 4.0: Source – Target Mapping List

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Source Table Name | Column Name | Data Type | Target Table Name | Column Name | Data Type | Active? | Active Inactive Date | Mod  Date | Abnormality | Notify |
| Table A | Col 01 | Int | Cont A | Col 02 | Int | Y | ddmmyyyy | ddmmyyyy | No | No |
| Table A | Col 02 | Date | Cont A | Col 03 | Int | Y | ddmmyyyy | ddmmyyyy | No | No |
| Table B | Col 23 | Alpha N | Cont B | Col 20 | Alpha N | Y | ddmmyyyy | ddmmyyyy | No | No |
| File A | Col 01 | Int | Cont C | Col 02 | Int | Y | ddmmyyyy | ddmmyyyy | No | No |
| File A | Col 33 | Alpha N | Cont C | Col 30 | Alpha N | Y | ddmmyyyy | ddmmyyyy | No | No |

Table 4.1 Source – Target Summary Table (off-shoot of Table 4)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Total No of Source Tables | Total No: Source Files | Total No: Source Table Columns | Total No of Files Column | Total No: Target Tables | No: of Target Columns | Remarks |
| 15 | 3 | 255 | 123 | 10 | 366 |  |

Table 4.2 Source Data Extraction Audit Information

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table or File Name | No: of Columns | No: of Records @ Source | No: of Bytes | Extraction  Type | Modified TS in Source | No: of Records @ Staging | No: of Bytes | Time to Extract | Abnormality | Notify |
| Table A | 2 | 500 | 120MB | Incremental | 23-Nov-2018 23:55:43 | 500 | 120MB | 00:00:23 | No | No |
| File B | 23 | 10000 | 987MB | Full | 23-Nov-2018 23:55:43 | 10000 | 987MB | 00:01:55 | No | No |
| Table B | 23 | 3000 | 22MB | Incremental | 23-Nov-2018 23:55:43 | 2990 | 21.8MB | 00:00:33 | Yes | Yes |

Table 4.3 Source Data Extraction Data Integrity & Data Accuracy Audit Table

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Source Table Name | Source Column Name | Source Min Value | Source Max Value | Source Avg Value | Col Value Agg | Stage Table Name | Stage Col Name | Stage Min Value | Stage Max Value | Stage Avg Value | Col Value Agg | Cal  Date | Abnormality | Notify |
| Table A | Col 10 | 7230 | 22233 | 54566 | 56.23L | Con A | Col 22 | 7230 | 22233 | 54566 | 56.23L | Ddmmy | N | N |
| Table B | Col 23 | 1234 | 98765 | 23456 | 22.43L | Con B | Col 32 | 2341 | 78765 | 23456 | 20.43L | Ddmmy | Y | Y |

A Notification alert or an Email will be sent at this moment to concern stake holders about the abnormality on data accuracy and Integrity would be sent for any corrective action. No email alert would be sent if there were no abnormality is found.

Table 4.4 Data Load Log & Audit Information

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Target Table Name | No: of Columns | No: of Records @ Target | No: of Bytes | Load  Type | Modified TS in Source | No: of Records @ Staging | No: of Bytes | Time to Load | Abnormality | Notify |
| Con A | 26 | 500 | 120MB | Incremental | 23-Nov-2018 23:55:43 | 500 | 120MB | 00:00:23 | No | No |
| Con B | 23 | 10000 | 987MB | Full | 23-Nov-2018 23:55:43 | 10000 | 987MB | 00:01:55 | No | No |
| Table B | 23 | 2990 | 21.8MB | Incremental | 23-Nov-2018 23:55:43 | 3000 | 22MB | 00:00:33 | Yes | Yes |

*\*Note: More Discussion Needed*

A Notification alert or an Email will be sent at this moment to concern stake holders about the abnormality on data accuracy and Integrity would be sent for any corrective action. No email alert would be sent if there were no abnormality is found.

Table 4.5 Target Data Load - Data Integrity & Data Accuracy Audit Table

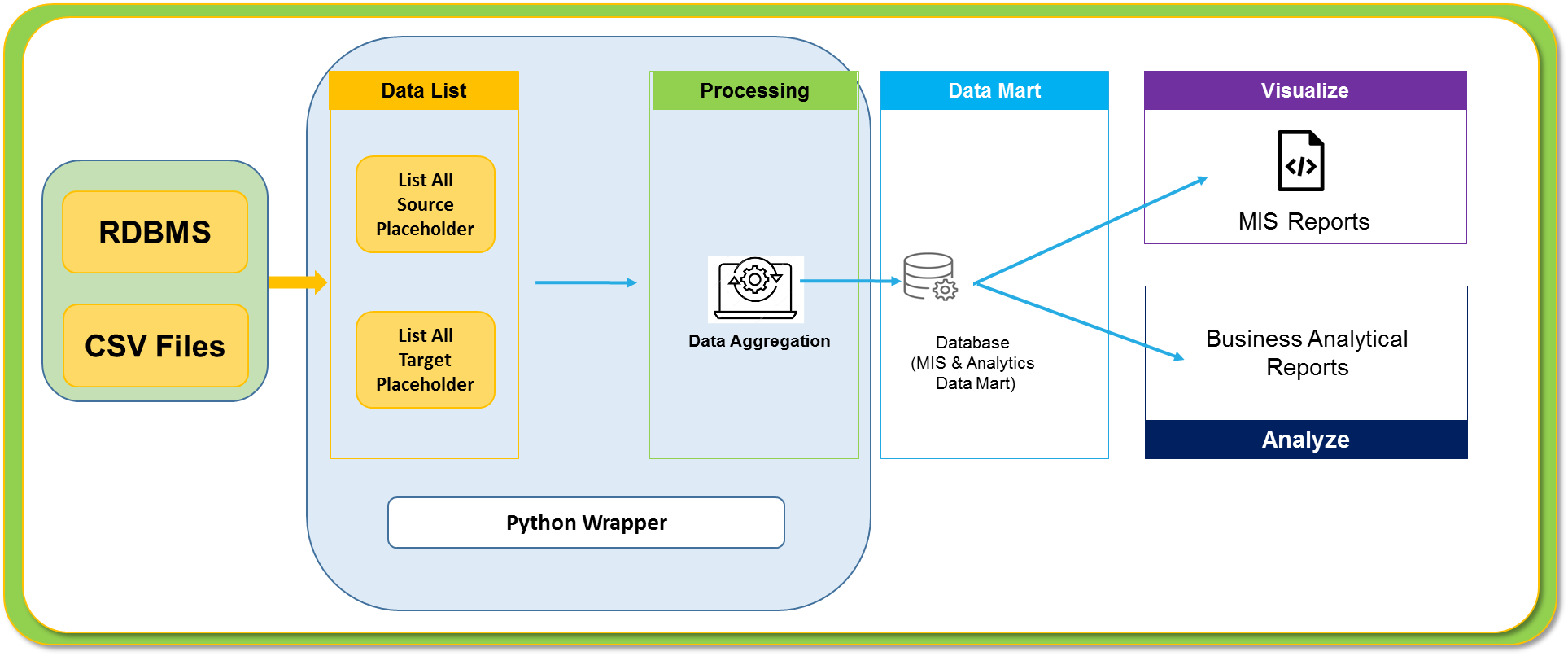
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Target Table Name | Target Column Name | Target Min Value | Target Max Value | Target Avg Value | Col Value Agg | Stage Table Name | Stage Col Name | Stage Min Value | Stage Max Value | Stage Avg Value | Col Value Agg | Cal  Date | Abnormality | Notify |
| Con A | Col 10 | 7230 | 22233 | 54566 | 56.23L | Table A | Col 22 | 7230 | 22233 | 54566 | 56.23L | Ddmmy | N | N |
| Con B | Col 23 | 1234 | 98765 | 23456 | 22.43L | Table B | Col 32 | 2341 | 78765 | 23456 | 20.43L | Ddmmy | Y | Y |

Table 4.6 Main ETL Process Success Notification Summary Audit Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Audit Table No: | Audit Table Name | Table Name | Table Type | Process Success |
| 4.0 | Source – Target Mapping List | Table A | Source | Yes |
| 4.2 | Source Data Extraction Audit Information | Table B | Source | No |
| 4.3 | Source Data Extraction Data Integrity & Data Accuracy Audit Table | Table B | Stage | No |
| 4.4 | Data Load Log & Audit Information | Table B | Target | No |

At every stage the Success List would be prepared, and a notification would be sent. The design is to try as much as modularity in the ETL Stream, so that Critical Data Loads would be separated and could be run Independently.

**Schematic Representation of Customized Python based ETL Tool:**



## Post ETL Process

|  |  |
| --- | --- |
|  | The Post ETL Process is performed mainly to do the house-keeping Job for the next ‘Main ETL Process’ execution. The important activities are to check the Table Space Extents for next Batch Run (part of Pre-ETL Process), clearing all caches, deleting any unwanted Temp Tables and archiving the old data. The Post ETL Process details will be available only after the Technical Design Document is ready. |