



Prepared by: LTI

Dated: 16August 2021

Version 1

**Low Level Design**

CNOS & GC Europe - Azure Blueprint2

**Contents**

1. [**Pre-reading & Version History**](#_Pre-reading_&_Version)
2. [Overview of CNOS & GC Data Service](#_Overview_of_CNOS)
   1. [Understanding of CNOS & GC](#_Understanding_of_CNOS)
   2. [Data components in scope of CNOS & GC Data service](#_Data_components_in)
3. [High Level Architecture](#_High_Level_Architecture)
   1. [Technical Components](#_Technical_Components)
   2. [System architecture components](#_System_architecture_components)
   3. [Data Sources / Ingestion](#_Data_Sources_/)
   4. [Processing Layer](#_Processing_Layer)
   5. [Consumption Layer](#_Consumption_Layer)
   6. [Data Flow](#_Data_flow)
4. [*Orchestration*](#_Orchestration)
   1. [Metadata approach](#_Metadata_approach)
   2. [ADF pipelines](#_ADF_pipelines)
      1. [Notebook\_Sequence\_Execution](#_Notebook_Sequence_Execution)
      2. [PreviousRunValidation](#_PreviousRunValidation)
      3. [SMO\_Parallel\_Notebook\_Execution](#_SMO_Parallel_Notebook_Execution)
      4. [Workday\_Orchestration](#_Workday_Orchestration)
5. [***External Interfaces***](#_External_Interfaces)
   1. [Service now](#_Service_now)
6. [***Data Processing***](#_Data_Processing)
   1. [Data Ingestion](#_Data_Ingestion_Patterns)
      1. [Data Ingestion Patterns](#_Data_Ingestion_Patterns)
   2. [Master Data](#_Master_Data)
   3. [FX Rates](#_FX_Rates)
   4. [Direct Shipments](#_Direct_Shipments)
      1. [350\_CREATE\_COMMON\_SHIP\_FACTS\_EU](#_350_CREATE_COMMON_SHIP_FACTS_EU)
      2. [450\_EU\_DIR\_IND\_SHIP\_Extraction](#_405_EU_DIR_IND_SHIP_Extraction)
      3. [405\_EU\_DIR\_ISHIP\_LC](#_405_EU_DIR_ISHIP_LC)
   5. [Indirect Shipments](#_Indirect_Shipments)
   6. [Unallocated](#_Unallocated)
      1. [01\_Create\_Common\_Tables](#_01_Create_Common_Tables)
      2. [CMF SD](#_CMF_SD)
      3. [NATSD](#_NATSD)
      4. [Trade Terms](#_Trade_Terms)
      5. [Optima](#_Optima_1)
         1. [For CE](#_For_CE)
         2. [For Nordics](#_For_Nordics)
   7. [Optima](#_Optima)
   8. [Anaplan](#_Anaplan)
      1. [02\_FRANCE\_ANAPLAN\_SHIPMENTS](#_02_FRANCE_ANAPLAN_SHIPMENTS)
      2. [03\_FRANCE\_ANAPLAN\_FPC\_GTIN\_PROC](#_03_FRANCE_ANAPLAN_FPC_GTIN_PROC)
      3. [04\_FRANCE\_ANAPLAN\_TDC\_IYA](#_04_FRANCE_ANAPLAN_TDC_IYA)
      4. [05\_FRANCE\_ANAPLAN\_TDC\_ROI](#_05_FRANCE_ANAPLAN_TDC_ROI)
      5. [06\_FRANCE\_ANAPLAN\_TDC\_OUTPUT](#_06_FRANCE_ANAPLAN_TDC_OUTPUT)
      6. [07\_FRANCE\_ANAPLAN\_COEFFICIENTS](#_07_FRANCE_ANAPLAN_COEFFICIENTS)
      7. [08\_FRANCE\_ANAPLAN\_COEFFICIENTS\_OUTPUT](#_08_FRANCE_ANAPLAN_COEFFICIENTS_OUTP)
      8. [01\_UKI\_ANAPLAN\_SHIPMENT](#_01_UKI_ANAPLAN_SHIPMENT)
      9. [02\_UKI\_ANAPLAN\_FPC\_GTIN\_PROC](#_02_UKI_ANAPLAN_FPC_GTIN_PROC)
      10. [03\_UKI\_ANAPLAN\_TDC\_IYA](#_03_UKI_ANAPLAN_TDC_IYA)
      11. [04\_UKI\_ANAPLAN\_TDC\_ROI](#_04_UKI_ANAPLAN_TDC_ROI)
      12. [05\_UKI\_ANAPLAN\_TDC\_OUTPUTS](#_05_UKI_ANAPLAN_TDC_OUTPUTS)
      13. [06\_UKI\_ANAPLAN\_COEFFICIENTS](#_06_UKI_ANAPLAN_COEFFICIENTS)
      14. [07\_UKI\_ANAPLAN\_COEFFICIENTS\_OUTPUTS](#_07_UKI_ANAPLAN_COEFFICIENTS_OUTPUTS)
   9. [TDC Smoothing](#_TDC_Smoothing)
      1. [110\_EUROPE\_TDC\_FACT\_PROCESS](#_110_EUROPE_TDC_FACT_PROCESS)
      2. [560\_EUR\_PEA\_FACT\_PROCESS](#_560_EUR_PEA_FACT_PROCESS)
   10. [PEA Sources](#_PEA_Sources:)
       1. [PEA\_ALB](#_PEA_ALB)
          1. [01\_eur\_pea\_and\_customer\_final](#_01_eur_pea_and_customer_final)
          2. [02\_eur\_pea\_fpc\_to\_gtin](#_02_eur_pea_fpc_to_gtin:)
          3. [03\_eur\_pea\_gtin\_fy](#_03_eur_pea_gtin_fy)
          4. [04\_eur\_pea\_tdc\_join](#_04_eur_pea_tdc_join)
          5. [05\_eur\_pea\_avg](#_05_eur_pea_avg)
          6. [06\_eur\_pea\_output](#_06_eur_pea_output:)
       2. [PEA\_AUS](#_PEA_AUS)
          1. [01\_eur\_pea\_aus\_customer\_final](#_01_eur_pea_aus_customer_final)
          2. [02\_eur\_pea\_aus\_tdc\_join](#_02_eur_pea_aus_tdc_join)
          3. [03\_eur\_pea\_aus\_fpc\_to\_gtin](#_03_eur_pea_aus_fpc_to_gtin)
          4. [04\_eur\_pea\_aus\_gtin\_fy](#_04_eur_pea_aus_gtin_fy)
          5. [05\_eur\_pea\_aus\_gtin\_ml\_fy](#_05_eur_pea_aus_gtin_ml_fy)
          6. [06\_eur\_pea\_aus\_avg](#_06_eur_pea_aus_avg)
          7. [07\_eur\_pea\_aus\_agg\_ml](#_07_eur_pea_aus_agg_ml)
          8. [08\_eur\_pea\_aus\_output](#_08_eur_pea_aus_output)
       3. [PEA\_BEL](#_PEA_BEL)
          1. [01\_eur\_pea\_bel\_customer\_final](#_01_eur_pea_bel_customer_final)
          2. [02\_eur\_pea\_bel\_tdc\_join](#_02_eur_pea_bel_tdc_join)
          3. [03\_eur\_pea\_bel\_fpc\_to\_gtin](#_03_eur_pea_bel_fpc_to_gtin)
          4. [04\_eur\_pea\_bel\_gtin\_fy](#_04_eur_pea_bel_gtin_fy)
          5. [05\_eur\_pea\_bel\_gtin\_ml\_fy](#_05_eur_pea_bel_gtin_ml_fy)
          6. [06\_eur\_pea\_bel\_data\_join](#_06_eur_pea_bel_data_join)
          7. [07\_eur\_pea\_bel\_avg](#_07_eur_pea_bel_avg)
          8. [08\_eur\_pea\_bel\_output](#_08_eur_pea_bel_output)
       4. [PEA\_TUR](#_PEA_TUR)
          1. [01\_eur\_pea\_tur\_customer\_final](#_01_eur_pea_tur_customer_final)
          2. [02\_eur\_pea\_tur\_fpc\_to\_gtin](#_02_eur_pea_tur_fpc_to_gtin)
          3. [03\_eur\_pea\_tur\_avg](#_03_eur_pea_tur_avg)
          4. [04\_eur\_pea\_tur\_output](#_04_eur_pea_tur_output)
7. [***Output Generation***](#_Output_Generation)
   1. [BNL OUTPUT CREATION](#_BNL_OUTPUT_CREATION)
   2. [CCAR\_OUTPUT\_CREATION](#_CAU_OUTPUT_CREATION)
   3. [CE\_OUTPUT CREATION](#_CE_OUTPUT_CREATION)
   4. [DACH\_OUTPUT\_CREATION](#_DACH_OUTPUT_CREATION)
   5. [EE\_OUTPUT\_CREATION](#_EE_OUTPUT_CREATION)
   6. FRANCE\_OUTPUT CREATION
   7. [IBERIA OUTPUT CREATION](#_IBERIA_OUTPUT_CREATION)
   8. [ITALY\_OUTPUT CREATION](#_ITALY_OUTPUT_CREATION)
   9. [NORDICS OUTPUT CREATION](#_NORDICS_OUTPUT_CREATION)
   10. [SEE OUTPUT CREATION](#_SEE_OUTPUT_CREATION)
   11. [TURKEY\_OUTPUT CREATION](#_TURKEY_OUTPUT_CREATION)
   12. [UKI\_OUTPUT\_CREATION](#_UKI_OUTPUT_CREATION)
8. [***DATA\_INTEGRATION***](#_Data_Ingestion)
   1. [01\_CUSTOMER\_REGIONAL\_MANUAL\_MAPPING](#_01_CUSTOMER_REGIONAL_MANUAL_MAPPING)
   2. [02\_DIMS AND FACT](#_02_DIMS_AND_FACT)
9. [***Automated Data Validation***](#_Automated_Data_Validation)
   1. [Output Data Validation](#_Output_data_validation)
   2. [Anaplan Output Automated Data Validation](#_Anaplan_Output_Automated)
   3. [Input Automated Data Validation](#_Input_Automated_Data)
10. [***Data Dependency Flow***](#_Data_Dependency_Flow)
11. [***Output Export***](#_Output_Export)
12. ***Data Visualization***

# Pre-reading & Version History

Document Revision History

| Ver. No. | Ver. Date | Prepared By | Reviewed By | Approved By | Affected Section & Summary of Change |
| --- | --- | --- | --- | --- | --- |
| 1.0 | 27/05/2021 | Sudheer Bachu, Ashutosh Singh Chauhan |  |  | Initial LLD document creation. |
| 1.1 | 04/10/2021 | Ashutosh Singh Chauhan |  |  | Included details of Data visualization. Changes related to renaming of CAU to CCAR. |
| 1.2 | 18/10/2021 | Ashutosh Singh Chauhan |  |  | Included details of AD\_HOC ingestion of file for RCM and creation of Exception reports. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Overview of CNOS & GC Data Service

## Understanding of CNOS & GC

* Implement a new data feed from C-NOS & GC data service, in a way that:
  + Re-designs existing inputs/transformation/outputs with newly agreed formulae (more details below)
  + Uses MDM & MMR in tandem for customer and product master data
  + Ensures Data Governance & security of last mile calculation of IYAs and Sales KPIs
* Enable automated connections to source systems where possible and use manual input if automated is not feasible, technically, or process-wise
* Continue strong data governance of TDC and GC raw values in line with RE tax model
* Ensure data model does not deviate within countries, and provides one number region-wide and in local SMOs

## Data components in scope of CNOS & GC Data service

# High Level Architecture

High Level Solution Architecture

Graphical user interface, application, website

Description automatically generated

## Technical Components

1. Azure Logic app - To create, develop and deploy cloud-based integrations and workflows.
2. Azure ADF – To read data and orchestrate complete flow
3. Azure Blob – To store the data
4. Databrics – To write or apply business logics for CNOS & GC
5. SQL Server – To store dimension and facts
6. Key vault – To securely storing and accessing secrets
7. Azure Monitor - Collect, analyze, and act on telemetry data from your Azure and on-premises environments
8. On-prem Gateway – To install in an on-premises network
9. Active Directory - for centralized domain management
10. AAS – To host semantic data models.

## System architecture components

1. Data Sources / Ingestion
2. Processing / Transformation
3. Consumption

## Data Sources / Ingestion

The below are the list of data sources

* RDS
* DIRECT SHIPMENTS
* INDIRECT SHIPMENTS
* OPTIMA
* MMR
* SharePoint

## Processing Layer

For Processing layer, ADF will act as an orchestration layer to trigger various ETL jobs.

Based on the pipeline design, ADF will invoke the Azure Databricks notebooks. Databricks will execute the data processing logic based on available system resources of execution worker.

## Consumption Layer

In consumption layer, dimensional and facts data will be stored.

This data will be moved from SQL DB to Azure Analysis Service (AAS) based on configured frequency. The data will be stored in AAS cubes in-memory to provide better performance.

## Data flow

Diagram

Description automatically generated

# Orchestration

In this complete process ADF is used as orchestrator tool and SQL DB tables as control tables for complete orchestration.

## Metadata approach

**[cngc\_operations].[CNOS\_GC\_ETL\_Notebooks]:**

[cngc\_operations].[CNOS\_GC\_ETL\_Notebooks] is a table where the list of notebooks which are active (Execution Enabled as “Y”) and to be executed are present with details like Execution Day,SMO,Subject Area, Notebook Name ,Notebook Path,Service Now Location and Execution Enabled Flag.

A screenshot of a computer

Description automatically generated

For any newly created notebook to be considered for execution, there should be an entry corresponding to that notebook in this table with mandatory information for Execution\_Day,SMO,Subject\_Area,Sequence,Notebook\_Name,Notebook\_Path,Execution\_Enabled fields

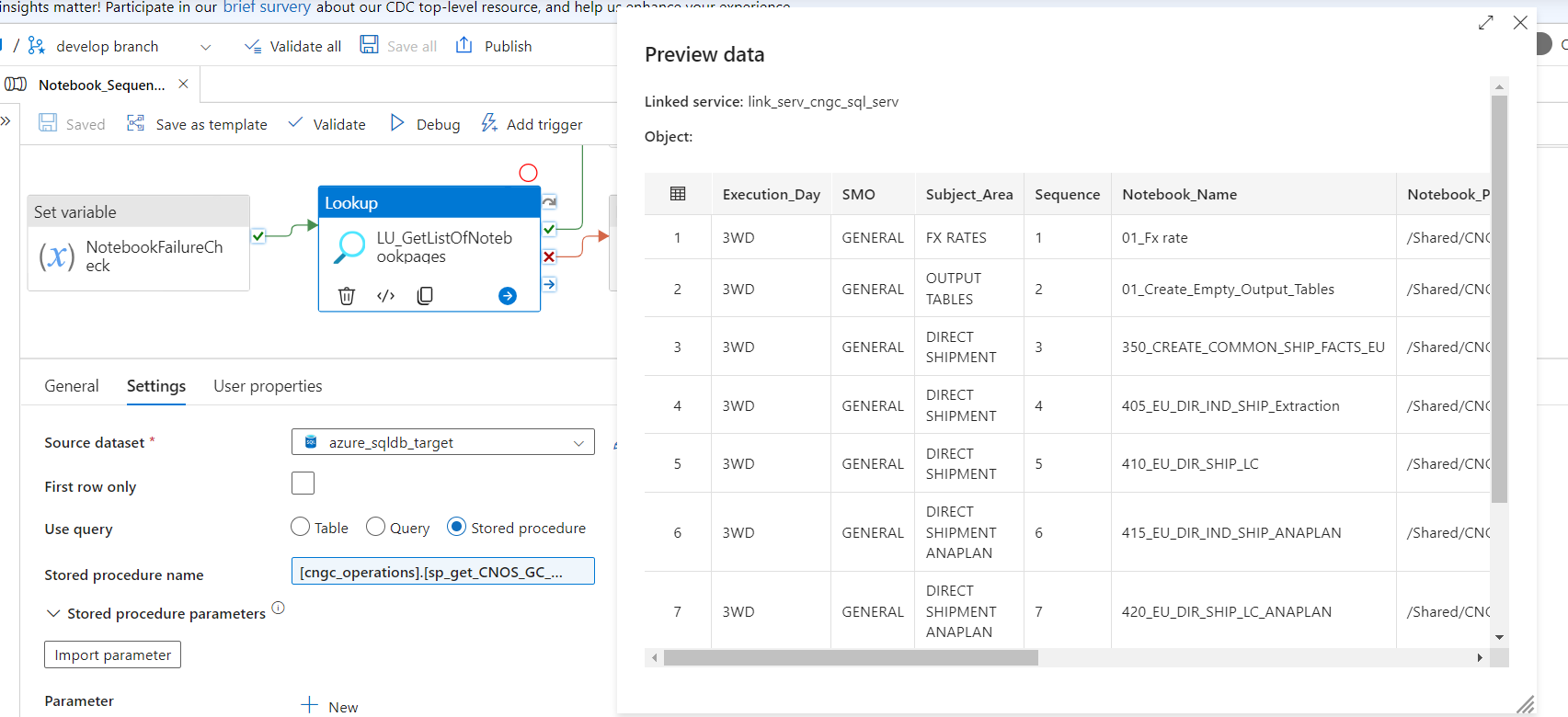
Insert Query:

INSERT INTO [cngc\_operations].[CNOS\_GC\_ETL\_Notebooks]

([Execution\_Day],[SMO],[Subject\_Area],[Sequence],[Notebook\_Name],[Notebook\_Path],[Execution\_Enabled],[Last\_Inserted\_Date])  
SELECT '3WD','FRANCE','OUTPUT','5','05\_eur\_rep\_agg\_mth\_fra','/EUROPE/OUTPUT CREATION/FRANCE OUTPUT CREATION/05\_eur\_rep\_agg\_mth\_fra','Y',GETDATE();

[cngc\_operations].[sp\_get\_CNOS\_GC\_ETL\_Notebooks]

This stored procedure is used to fetch the list of notebooks to be executed from the table [cngc\_operations].[CNOS\_GC\_ETL\_Notebooks] with Execution Enabled flag as “Y”.



[cngc\_operations].[cngc\_output\_email\_list]:

This table stores information about list of SMO which are having Blob to SharePoint Enabled as ‘Y’ or ‘N’ for various output types for different SMO’s. The enabled column signifies whether Blob to Sharepoint output generation is enabled or disabled.

A computer screen shot of a computer screen

Description automatically generated

[cngc\_stg\_eur].[spUtil\_Log\_ETL\_Job]

‘[cngc\_stg\_eur].[spUtil\_Log\_ETL\_Job]’ is used to log the run entries of Notebooks with status as START,END and ERROR in ‘cngc\_stg\_eur.work\_ETL\_Audit\_Header’ table.

Graphical user interface, application

Description automatically generated

Input parameters to the procedure includes Pipeline Name,Trigger Name,Trigger Type,Trigger Time etc.

‘cngc\_operations.V\_CNOS\_GC\_Data\_Validation\_Status’:

‘cngc\_operations.V\_CNOS\_GC\_Data\_Validation\_Status’ view provides the information of data validation status of input data for each SMO.

A screenshot of a computer

Description automatically generated

## ADF pipelines

**List of Pipelines:**

1. Pipeline\_EUROPE\_Main\_Processing
2. Workday\_Orchestration
3. Notebook\_Sequence\_Execution
4. PreviousRunValidation
5. SMO\_Parallel\_Notebook\_Execution
6. cngc\_eu\_dim\_fact\_aas\_notification

## Pipeline\_EUROPE\_Main\_Processing

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Description**:

This “Pipeline\_EUROPE\_Main\_Processing” pipeline is used to call all the other activities based on the workday parameter.

1. **Input Parameters:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Data Type | Sample Input Data |
| SMO\_GENERAL | String | BNL |
| WorkDay | String | 5WD |
| CNGC\_EUROPE\_Path | String | /Shared/CNGC\_EUROPE/ |
| ADV\_Flag | String | 1 |

1. **Input Data:**

Input for the pipeline is WorkDay parameter which is fetched by using the stored procedure [cngc\_operations].[get\_pwd\_number] or can be given as input parameter.

1. **Execution Flow:**

The pipeline “Pipeline\_EUROPE\_Main\_Processing” begins by logging ETL activities.The Set Workday activity sets the workday in case of input parameter else result obtained from the previous lookup. Then A switch case within a forEach loop is used, each case within switch case corresponds to a particular workday, when value is matched with a case it invokes ‘Workday\_Orchestration’ pipeline and performs all other activities.

## Workday\_Orchestration

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Description:

This pipeline is used to process all the input SMO’s parallelly by running all notebooks in sequence, which is defined in control table, for the respective SMO and this can be used in the below mentioned scenarios,

* To process the single/Multiple SMO’s notebooks parallelly based on the Execution day and Subject Area.
* To Process all the SMO’s specific notebooks for a given working day.

Input Data:

Input for the pipeline is working day which is fetched from the stored procedure [cngc\_operations].[get\_pwd\_number] by passing the current date.

Graphical user interface, application

Description automatically generated

Execution Flow:

Once the Pipeline execution is started, an entry will be made in the table [cngc\_operations].[ETL\_PROCESS\_EXECUTION] by executing the stored procedure [cngc\_operations].[usp\_ins\_ETL\_Execution] and next activity Lookup\_stored\_procedure\_for\_wdp will be started.

Up on completion of Lookup\_stored\_procedure\_for\_wdp, working day is passed to SMO\_Parallel\_Execution pipeline, which will trigger PreviousRunValidation and Notebook\_Sequence\_Execution sequentially.

Example:

Run date – 2021-03-05

For the given run date, output from [cngc\_operations].[usp\_ins\_ETL\_Execution] is 5WD, which will be passed to SMO\_Parallel\_Execution to execute all the SMO’s configured to run on 5WD parallelly and the corresponding notebooks sequentially.

## Notebook\_Sequence\_Execution

A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

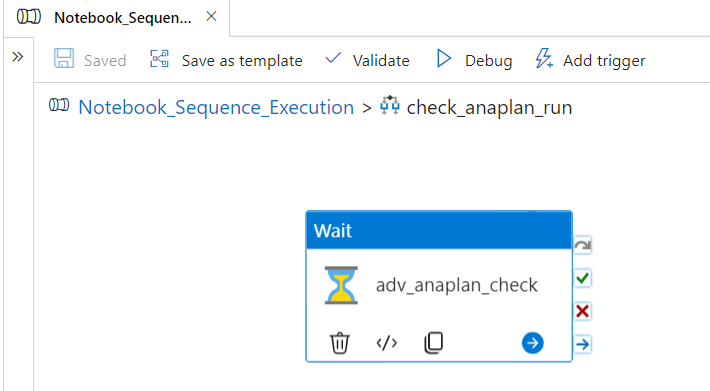
Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

****

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Description:**

This pipeline is used to run the notebooks in sequence based on the input parameters given and this can be used in the below mentioned scenarios,

* To restart from the point of a failure (Sequence) for a specific SMO, Execution Day and Subject Area.
* To perform an Ad-hoc run for a specific SMO, Execution Day and Subject Area.

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Data Type | Sample Input Data |
| SMO | String | BNL |
| Execution\_Day | String | 5WD |
| Subject\_Area | String | OUTPUT |
| Sequence | String | 1 |

**Input Data:**

Input for the pipeline is list of notebooks which are fetched by using the stored procedure [cngc\_operations].[sp\_get\_CNOS\_GC\_ETL\_Notebooks] from activity LU\_GetListOfNotebookpages.

Graphical user interface, text, application

Description automatically generated

Execution Flow:

Once the input data (list of notebooks) is fetched, it is filtered based on the input parameters by the filter activities and passed to foreach activity NotebookExecution.

Once the control enters into foreach activity, notebook execution starts after logging the status in audit table [cngc\_operations].[sp\_log\_CNOS\_GC\_ETL\_Step\_Hist] with the help of [cngc\_stg\_eur].[spUtil\_Log\_ETL\_JOB] and same will happened post success/Failure of notebook execution.

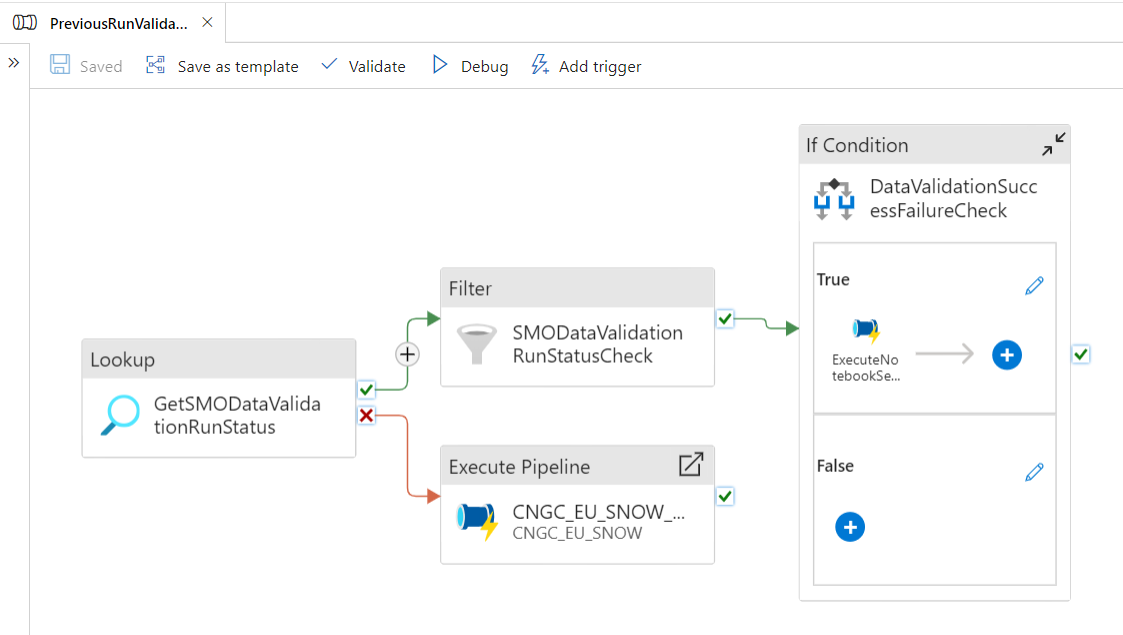
In case of Notebook failure for each is broken with the help of Web Activity BreakForEach for which the URL is fetched from the set variable activity PipelineRunId.

**Example:**

If the input parameters are as below, then the pipeline will execute the notebooks of 5th working day BNL with subject Area OUTPUT and Sequence number equal or greater than 1.

|  |  |
| --- | --- |
| Parameter Name | Sample Input Data |
| SMO | BNL |
| Execution\_Day | 5WD |
| Subject\_Area | OUTPUT |
| Sequence | 1 |

## PreviousRunValidation

****

**A screenshot of a computer

Description automatically generated**

Description:

This Pipeline is used to check the data validation status of the SMO given in parameters and this can be used in the below mentioned scenarios,

* To check the status of data validation of given SMO and execute the notebooks corresponding to the given SMO sequentially by calling the pipeline Notebook\_Sequence\_Execution with respect to the other given parameters.

Input Parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Data Type | Sample Input Data |
| SMO | String | BNL |
| Execution\_Day | String | 5WD |
| Subject\_Area | String | OUTPUT |
| Sequence | String | 1 |

Input Data:

Input for the pipeline is status of data validation failure happened for the SMO’s on the run date, which is fetched from the view [cngc\_operations].[V\_CNOS\_GC\_Data\_Validation\_Status].

Execution Flow:

Once the data is fetched, it is compared with the SMO passed as parameter for the pipeline.

If there is no data validation failure then the process will start Notebook\_Sequence\_Execution and run the notebooks in sequence which are matching the input parameter criteria, but, If there is any data validation failure then the process will stop and will not go ahead with processing of the notebooks.

Example:

If the input parameters are as below, then the pipeline will validate if there is any data validation failure entry for the “BNL” SMO on run date and proceed ahead if there are no failures or stop if there is any failure.

|  |  |
| --- | --- |
| Parameter Name | Sample Input Data |
| SMO | BNL |
| Execution\_Day | 5WD |
| Subject\_Area | OUTPUT |
| Sequence | 1 |

## SMO\_Parallel\_Notebook\_Execution

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Description:

This pipeline is used to process all the input SMO’s sequentially by running all notebooks in sequence, which is defined in control table, for the respective SMO and this can be used in the below mentioned scenarios,

* To process the single/Multiple SMO’s notebooks sequentially based on the Execution day and Subject Area.
* To Process all the SMO’s specific notebooks for a given working day.

Input Parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Data Type | Sample Input Data |
| SMO | Array | [“BNL”,”France”] |
| Execution\_Day | String | 5WD |
| Subject\_Area | String | OUTPUT |
| Sequence | String | 1 |

Input Data:

Input for the pipeline is list of SMO’s which are fetched from the table ‘[cngc\_operations].[CNOS\_GC\_ETL\_Notebooks]’.

Graphical user interface

Description automatically generated

Execution Flow:

Once the input data (list of SMO’s) is fetched, it is filtered based on the input parameters(SMO, Execution Day) by the filter activity and passed to foreach activity SMO\_Parallel\_Execution .

Once the input is received to for each activity, all parallel executions will start for all the SMO’s and will trigger PreviousRunValidation for checking the data validation status of the respective SMO’s.

If the data validation is successful, then Notebook\_Sequence\_Execution pipeline will be triggered and the notebooks matching the input criteria will be executed sequentially else the process will step for that SMO alone.

Example:

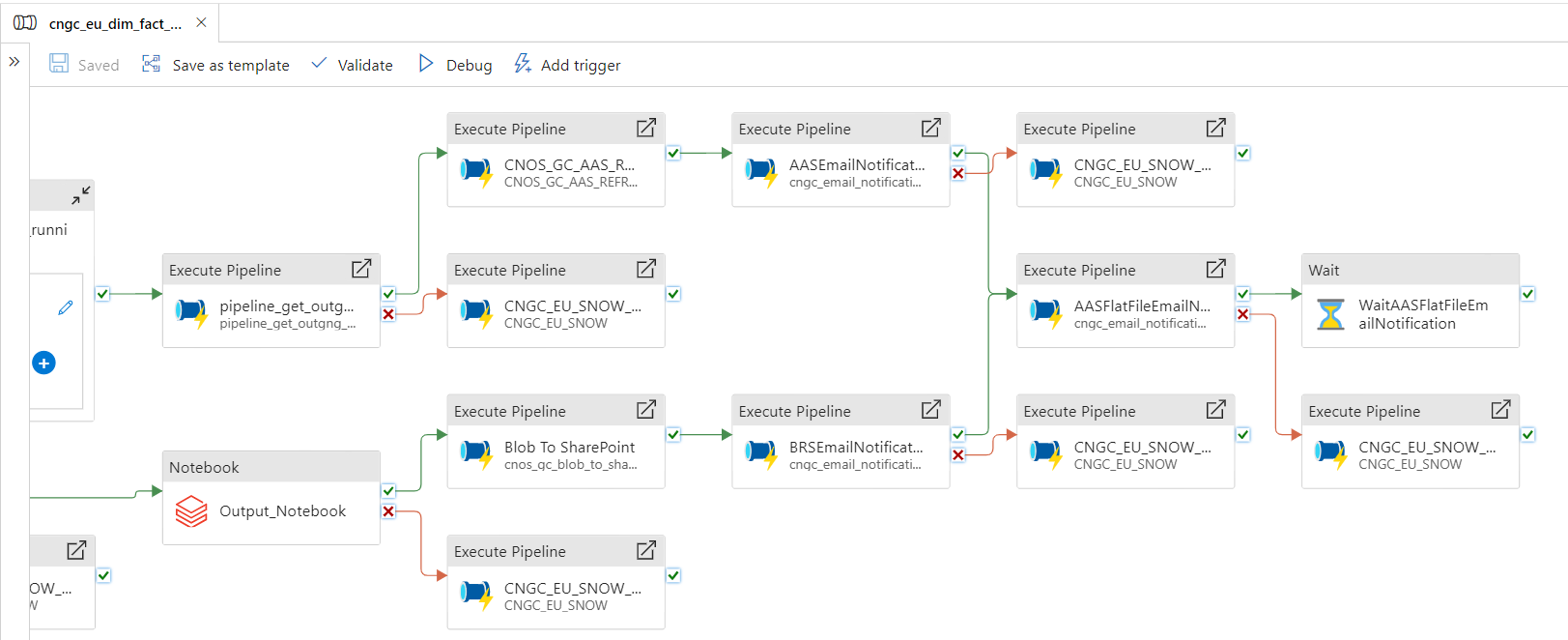
|  |  |
| --- | --- |
| Parameter Name | Sample Input Data |
| SMO | [“BNL”,”FRANCE”] |
| Execution\_Day | 5WD |
| Subject\_Area | OUTPUT |
| Sequence | 1 |

If the input parameters are as below, then the pipeline will start parallel processing for BNL and FRANCE . First data validation failure is checked and upon successful completion, notebooks of BNL and FRANCE of 5th working day with Subject Area OUTPUT are executed sequentially.

## cngc\_eu\_dim\_fact\_aas\_notification

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

Description:

In this pipeline all the ‘Dims and Fact’ notebooks and output notebooks are being processed and AAS notifications are being sent.

Input Parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Data Type | Sample Input Data |
| SMO | String | BNL |
| NotebookPath | String | /SHARED/ |
| WorkDay | String | 5WD |

Input Parameters:

Input for the pipeline is list of notebooks which are fetched from the table **‘**cngc\_meta.cdl\_light\_refined\_EU’.

Execution Flow:

This pipeline processes notebooks ‘01\_CUSTOMER\_REGIONAL\_MANUAL\_MAPPING’ and’ 02\_DIMS AND FACT’ and then fetches DIM, FCT table list for AMA and refreshes data using ‘CNOS\_GC\_AAS\_REFRESH’ activity and sends the email notificatons using ‘cngc\_email\_notification’ activity.

# External Interfaces

As external interface used below components

## Service now

Service now:

**Prerequisites:**

* Service Now end point URL.
* ServiceNow Username
* ServiceNow Password
* Service Now end point URL payload/body.

In Azure ADF, we can use the Web activity to send the request to service now end point URL by passing the body, username and password as shown below

Graphical user interface, text, application

Description automatically generated

**Body :**

Graphical user interface, text

Description automatically generated

# Data Processing

## Data Ingestion Patterns

(a) Data is ingested automatically from various sources through pipelines as indicated below:

* **CDL**

Unrefined Data from CDL is extracted for Direct Shipments on 3WD using pipeline ‘pipeline\_get\_cdl\_dir\_ship\_table\_list\_cp’ which is carried out by DXC team and Unrefined data for Indirect Shipments is extracted on 5 and 6 WD using pipeline ‘pipeline\_get\_cdl\_ind\_ship\_table\_list\_cp\_europe’

* **MMR MD (RDS)**

Data from MMR is extracted on 2 WD using pipeline ‘pipeline\_get\_mmr\_table\_list\_cp\_EU’.

* **Optima**

Data from Optima is extracted on daily basis using pipeline ‘pipeline\_get\_cdl\_unr\_optima\_data’.

(b) Data from Share-point is extracted for Customer Manual Mapping and Regional mapping

## Master Data

* In Notebook ‘MMR MD 2WD’ Customer mapping, Geo mapping, Product mapping and Customer hierarchy tables are created using parquet files from Blob storage.
* New structures consisting of required columns from Customer mapping, Geo mapping, Product mapping and Customer hierarchy are created.
* A Unified view for GEO MAPPING is created to provide unified list of Geo IDs irrespective of Geo ID Level.

## FX Rates

* FX Rate for Argentina and Kyrgyzstan is provided as manual input via MasterMinder Application, while for other countries it’s being extracted from CDL.
* In notebook ‘00\_Exchange\_rate’ Local currency code is mapped on the basis of geo levels from Geo mapping table as geo Id to create currency geo ID table.
* C$ and R$ calculation is done separately for Direct and Indirect Shipments.

## Direct Shipments

Data processing of Unrefined Data of Direct Shipment is processed in following Notebooks: -

350\_CREATE\_COMMON\_SHIP\_FACTS\_EU, 405\_EU\_DIR\_IND\_SHIP\_Extraction and 405\_EU\_DIR\_ISHIP\_LC. Details of the notebooks are described below.

## 350\_CREATE\_COMMON\_SHIP\_FACTS\_EU

* Table ‘ship\_step\_100\_sode\_hist\_star\_work\_eu’ from unrefined direct shipments having join condition with Customer hierarchy is created.
* In the table ‘ship\_step\_100\_sode\_hist\_star\_work\_eu’ a select statement with alias FCT where FACT\_TYPE\_CODE is not equal to 'TR' is applied in which following operations and conditions are applied:

(a) Geo Ids '056', '442' which refers to the countries Belgium and Luxembourg respectively from, ‘unrefined\_direct\_shipments\_v’ are classified under ‘918’ in order to form a singular cluster and Geo Ids '336', '380', '674' which refers to the countries are classified under ‘991’ which is referred to as Italy cluster.

(b) STAT\_UNIT\_QTY, GROSS\_TC\_AMT and NET\_TC\_AMT from unrefined\_direct\_shipments\_v are summed up into STAT\_UNIT\_QTY, GROSS\_TC\_AMT and NET\_TC\_AMT respectively to aggregate it at group by level.

(c)When SRCE\_SYS\_ID is among (2965,2966,3074,3075,3076,3077,3078,

3079,3080,3183,825,829,830,840,850,854,855,856,859,873,876,878,879,880,884,887,890) and Fact\_Type\_code is ‘BR’ and ‘ATTR\_VAL\_7’ is null or 0 then Sum of ENV\_FEE is 0 otherwise it’s taken as difference of ‘NET\_TC\_AMT’ and ‘ATTR\_VAL\_7’.

* A case statement in the table ‘ship\_step\_100\_sode\_hist\_star\_work\_eu’ is inserted to classify year into four quarters as described below:

1. When month number is among ‘7’,’8’ and ‘9’ then Q1 (Quarter 1)
2. When months number is among ’10’,’11’ and ‘12’ then Q2 (Quarter 2)
3. When months number is among ‘1’,’2’ and ‘3’ the Q3 (Quarter 3)
4. When months number is among ‘4’,’5’ and ‘6’ then Q4 (Quarter 4)

* Another case statement is inserted to classify year into two halves as when the month number is among '1', '2', '3', '4', '5', '6' then it’s classified as HY2 (Half-Year 2) and when month number is among ‘7’,’8’,’9’,’10’,’11’,’12’ then HY1 (Half-Year 1)
* Following left Joins are performed on table ‘ship\_step\_100\_sode\_hist\_star\_work\_eu’ with:

1. SODE\_PC064\_FDIM (Profit Centre hierarchy) On PROFT\_CTR\_ID (Profit Centre ID)
2. SODE\_CAL\_FDIM (Calendar Dimension) On MTH\_NUM (Month Number)
3. SODE\_O817\_FDIM (Organization Hierarchy) On ORG\_ID (Organization ID)
4. SODE\_C898\_FDIM (Customer Hierarchy) On CUST\_ID (Customer ID)
5. SODE\_P5005\_FDIM (Product Hierarchy) On PROD\_ID (Product ID)
6. SODE\_G707\_FDIM (GEO Hierarchy) On GEO\_ID (Geo ID)
7. SODE\_CI\_FDIM On SHPMT\_CLASS\_IND\_ID

## 405\_EU\_DIR\_IND\_SHIP\_Extraction

* Tables ‘ship\_step\_100\_sode\_hist\_star\_work\_eu’ and ‘mnr\_geo\_mapng\_dim\_vw\_v’ are refreshed to update the data within them.
* Table ‘dir\_ship\_step\_120\_sode\_hist\_star\_lvl\_4\_filtered\_eu’ is created in which fact table ‘ship\_step\_100\_sode\_hist\_star\_work\_eu’ is mapped with Product Dimension, Geo dimension and Customer Dimension using join for the geo level 4 and customer regional name as Europe.
* Table ‘dir\_ship\_step\_120\_sode\_hist\_star\_lvl\_5\_filtered\_eu’ is created in which fact table ‘ship\_step\_100\_sode\_hist\_star\_work\_eu’ is mapped with Product Dimension, Geo dimension and Customer Dimension using join for the geo level 5 and customer regional name as Europe.
* Table ‘dir\_ship\_step\_120\_sode\_hist\_star\_lvl\_6\_filtered\_eu’ is created in which fact table ‘ship\_step\_100\_sode\_hist\_star\_work\_eu’ is mapped with Product Dimension, Geo dimension and Customer Dimension using join for the geo level 6 and customer regional name as Europe.
* Table ‘dir\_ship\_step\_120\_sode\_hist\_star\_lvl\_4\_5\_6\_filtered\_eu’ is created by taking union of tables created for Geo level 4,5 and 6 where geo ID is not equal to ‘991’ (Italy cluster).

## 405\_EU\_DIR\_ISHIP\_LC

* The table ‘dir\_ship\_step\_120\_sode\_hist\_star\_lvl\_4\_5\_6\_filtered\_eu’ is refreshed to update the data within it before creating ‘step\_100\_fx\_rates\_dir’ table.
* A left join is applied on table ‘dir\_ship\_step\_120\_sode\_hist\_star\_lvl\_4\_5\_6\_filtered\_eu’ with ‘FX\_mth\_multiple\_currencies\_v’ on Target Currency Code (described as Currency\_ID in ‘dir\_ship\_step\_120\_sode\_hist\_star\_lvl\_4\_5\_6\_filtered\_eu’ and TRGT\_ISO\_CRNCY\_CODE in table ‘FX\_mth\_multiple\_currencies\_v’ ) and Source Currency code ( described as SOURCE\_CRNCY\_CODE in ‘dir\_ship\_step\_120\_sode\_hist\_star\_lvl\_4\_5\_6\_filtered\_eu’ and SRCE\_ISO\_CRNCY\_CODE in ‘FX\_mth\_multiple\_currencies\_v’ ) which fetches the data
* NIV2 is calculated as the difference of NIV (Net Input Volume) and ENV\_FEE (Environmental Fee).
* GIV\_LC (Gross Input Volume in Local Currency) is calculated as the product of GIV(Gross Input Volume) and EXCHG\_RATE (Exchange Rate).
* NIV\_LC (Net Input Volume in Local Currency) is calculated as the product of NIV(Net Input Volume) and EXCHG\_RATE (Exchange Rate).
* ENV\_FEE\_LC (Environmental Fee in Local Currency) is calculated as the product of ENV\_FEE (Environmental Fee) and EXCHG\_RATE (Exchange Rate).
* NIV2\_LC (Net Input Volume in Local Currency) is calculated as the difference in products of NIV (Net Input Volume), EXCHG\_RATE (Exchange Rate) and of ENV\_FEE (Environmental Fee), EXCHG\_RATE (Exchange Rate).

## Indirect Shipments

* For Indirect Shipments data is processed separately for other clusters and SEE cluster as data for SEE cluster is ingested on 5 WD and for others it is ingested on 6 WD.

In Notebook ‘03\_eu\_ind\_ship\_mth\_agg’ base table ‘ISIS\_EM\_PR\_SALES\_DAY\_DFCT’ is created using parquet file from blob storage.

* Another table ‘ISIS\_EM\_PR\_SALES\_MTH\_DFCT’ is created to refine and filter the data obtained from base table. Values such as ‘su\_vol\_qty’ ,’case\_vol\_qty’, ‘item\_vol\_qty’, ‘net\_val\_tc\_amt’, ‘net\_val\_usd\_amt’, ‘gross\_val\_tc\_amt’ and ‘gross\_val\_usd\_amt’ are summed up so that they can be aggregated at group level. The data is not extracted for countries whose geo Ids is among ('642', 'E04', '807', '008', '070', '688', '498', '100', '499', '300') as they all belong to SEE cluster. A separate table is created in notebook ‘03\_eu\_ind\_ship\_mth\_agg\_see’ as ‘ISIS\_EM\_PR\_SALES\_MTH\_DFCT\_SEE’ in which the same logic is used for countries whose geo Ids belongs to SEE Cluster as mentioned above.
* In notebook ‘01\_eu\_ind\_cust\_map\_stg’ the table ‘eu\_ind\_cust\_map\_stg’ is created from ‘mnr\_cust\_mapng\_dim\_vw’ Customer Mapping Table which is joined with Customer Hierarchy ‘SODE\_C898\_FDIM’ on Customer ID excluding the geo IDs of countries belonging to SEE. In notebook ‘01\_eu\_ind\_cust\_map\_stg\_see’ same logic is followed and table ‘eu\_ind\_cust\_map\_stg\_see’ is created for countries belonging to SEE.
* In notebook ‘02\_eu\_ind\_cust\_map’ customer mapping table for indirect shipments is created with name ‘eu\_ind\_cust\_map’ from base table ‘eu\_ind\_cust\_map\_stg’ where Source of Indirect is true (SRC\_INDIR\_VAL = 'Y') and countries not belonging to geo Ids for SEE cluster. Similar logic is followed in notebook ‘02\_eu\_ind\_cust\_map\_see’ and table ‘eu\_ind\_cust\_map\_see’ is created from base table ‘eu\_ind\_cust\_map\_\_stg\_see’ for countries belonging to SEE cluster.
* In notebook ‘04\_eu\_mth\_star\_no\_rds’ a table named ‘eu\_mth\_star\_no\_rds’ is created in which is mapped with Customer Hierarchy, Product Hierarchy, Calendar Dimension, Geo Hierarchy and excluding the countries from SEE cluster with tables using join conditions as mentioned below:
  + SODE\_C898\_FDIM (Customer Hierarchy) On CUST\_ID (Customer ID)
  + SODE\_CAL\_FDIM (Calendar Dimension) On MTH\_END\_DATE (Month End Date)
  + SODE\_IC1000\_FDIM On
  + SODE\_P5005\_FDIM (Product Hierarchy) On PROD\_ID (Product ID)
  + SODE\_G707\_FDIM (GEO Hierarchy) On GEO\_ID (Geo ID)
* In a separate notebook ’04\_eu\_mth\_star\_no\_rds\_see’ table ‘eu\_mth\_star\_no\_rds\_see’ is created using same logic for countries belonging to SEE cluster. ONE ADDITIONAL CONDITON\*\*
* In notebook ‘05\_eu\_ind\_sh’ table ‘eu\_ind\_sh’ is created for all countries except countries with GEO Ids belonging to SEE cluster. The following operations are carried out in the notebook:

a) Fiscal Year ID of current Fiscal year and previous Fiscal year is defined based on maximum Month Number from Calendar Dimension ’SODE\_CAL\_FDIM’, both are selected using Union statement.

b) A case statement in the table ‘eu\_ind\_sh’ is inserted to classify year into four quarters as described below:

* + - When month number is among ‘7’,’8’ and ‘9’ then Q1 (Quarter 1)
    - When months number is among ’10’,’11’ and ‘12’ then Q2 (Quarter 2)
    - When months number is among ‘1’,’2’ and ‘3’ the Q3 (Quarter 3)
    - When months number is among ‘4’,’5’ and ‘6’ then Q4 (Quarter 4)

c) Another case statement is inserted to classify year into two halves as when the month number is among '1', '2', '3', '4', '5', '6' then it’s classified as HY2 (Half-Year 2) and when month number is among ‘7’,’8’,’9’,’10’,’11’,’12’ then HY1 (Half-Year 1)

d) A join condition is applied to table ‘eu\_mth\_star\_no\_rds’ to map it with ‘mnr\_prod\_mapng\_dim\_vw\_v’ on Product ID (described as ‘PROD\_6\_ID’ in table ‘eu\_mth\_star\_no\_rds’ and ‘BRAND\_ID’ in ‘mnr\_prod\_mapng\_dim\_vw\_v’) and Geo ID only where EXCDD\_IND is not present.

A separate notebook ‘05\_eu\_ind\_sh\_see’ is used with same logic for countries belonging to SEE cluster.

* In notebook ‘06\_eu\_ind\_customer\_data’ view ‘eu\_ind\_customer\_data’ is created in which Name of Reporting Customer L4 ‘RPTNG\_CUST\_L4\_NAME’ is mapped at different levels by using left join with ‘eu\_ind\_cust\_map’ table where ‘SRC\_CUST\_ID‘ (Source Customer ID) from customer mapping at respective levels is equal to Customer Id from table’ eu\_ind\_sh’ at respective levels for all the countries except countries belonging to SEE cluster. To gather data of Customer at their respective levels. For Ex- ‘RPTNG\_CUST\_L4\_NAME’ is mapped at LEV= ‘3’ (Level 3) using join condition on tables ‘CUST\_MAP3’ and ‘eu\_ind\_sh’ where ‘SRC\_CUST\_ID ‘(Source Customer ID) from ‘CUST\_MAP3’ is equal to ‘CUST\_3\_ID’ from ‘eu\_ind\_sh’.
* It also consists of a logic for aggregating ‘RPTNG\_CUST\_L4\_NAME’ for all other customers which do not belong to any Customer level (1-12).
* In a separate notebook ‘06\_eu\_ind\_customer\_data\_see’ view ‘eu\_ind\_customer\_data\_see’ is created with same logic for countries whose Geo Ids belongs to SEE cluster.

## Unallocated

Data which is not mapped to any customer or product but still has values for coefficients CMF SD, National SD (NATSD), Optima and Trade Terms (TT) is processed as Unallocated. It is processed separately for each cluster for Direct and Indirect shipments.

## 01\_Create\_Common\_Tables

* Table ‘TST\_Optima\_Switch’ is created to map optima data with geo mapping table for Direct and Indirect Customers. It is defined for geo level 5 and 6. For direct shipments customer is defined as ‘Direct’ and Optima switch as ‘OPT\_SWITCH\_DATE\_DIR’ where SHPMT\_DIR\_IND='Y', for indirect shipments customer type is defined as ‘Indirect’ and Optima switch as ‘OPT\_SWITCH\_DATE\_IND’ where SHPMT\_INDIR\_IND='Y'. Records for direct and indirect are merged using a union clause.
* Table ‘eur\_rep\_form\_optima\_switch’ is created to contain specific columns from table ‘TST\_Optima\_Switch’. It includes SMO name, Country name, Cluster name, Geo\_ID, optima switch date and Customer type.
* Table ‘DIRECT\_SH\_GIV\_BRAND\_CE\_NORDICS’ is created to generate GIV\_LC data at geo ID, month, customer ID and Brand ID. For countries belonging to Nordics cluster.
* Tables ‘DR\_CUSTOMER\_MAP\_UNALLOCATED’ and ‘IND\_CUSTOMER\_MAP\_UNALLOCATED’ are created as customer mapping table for Direct and Indirect shipments respectively.
* Tables ‘CAL\_MASTR\_DIM’ and ‘DR\_PROD\_HIER’ are created to extract required columns from Calendar Dimension (SODE\_CAL\_FDIM) and Product hierarchy (SODE\_P5005\_FDIM\_WORK) respectively.
* Tables ‘CEEX\_OPT\_DIRECT’ and ‘WEEX\_OPT\_DIRECT’ are created to calculate SD KPI for direct customers of Central and Western Europe respectively. SD is calculated as sum of (ESTMT\_FIXED\_MDA\_LOR) and (ESTMT\_VARIABLE\_MDA\_LOR). They are joined with table ‘wk\_day\_date’ table to extract data only for workdays. They are mapped with customer mapping table (mnr\_cust\_mapng\_dim\_vw\_opt) and calendar dimension based on Optima ID and Month name respectively.
* Table ‘OPTIMA\_DIRECT’ is created to merge the data for Central & Western Europe for direct customers.
* Tables ‘CEEX\_OPT\_INDIRECT’ and ‘WEEX\_OPT\_INDIRECT’ are created to calculate SD KPI for indirect customers of Central and Western Europe respectively. SD is calculated as sum of (ESTMT\_FIXED\_MDA\_LOR) and (ESTMT\_VARIABLE\_MDA\_LOR). They are joined with table ‘wk\_day\_date’ table to extract data only for workdays. They are mapped with customer mapping table (mnr\_cust\_mapng\_dim\_vw\_opt\_ind) and calendar dimension based on Optima ID and Month name respectively.
* Table ‘OPTIMA\_INDIRECT’ is created to merge the data for Central & Western Europe for indirect customers.

## CMF SD

* A raw table is created in which data is filtered out on basis of Geo IDs and KPI columns which are not null having source CMF SD.
* Another ‘other’ table is created to deduct named customers from total customers.
* Data from the raw table and ‘other’ table is merged into Final table.
* Another table for each cluster is created which filters the records for Customer\_L4 as ‘Total’ and where all the measures from SD are either null or 0. Records are extracted from where GIV\_LC is null or 0 to get only unallocated records.

## NATSD

* A raw table is created in which data is filtered out based on Geo IDs and KPI columns which are not null having source Trade Terms.
* Another ‘other’ table is created to deduct named customers from total customers.
* Data from the raw table and ‘other’ table is merged into Final table.
* Another table for each cluster is created which filters the records for Customer\_L4 as ‘Total’ and where all the measures for NATSD are either null or 0. Records are extracted from where GIV\_LC is null or 0 to get only unallocated records.

## Trade Terms

* A raw table is created in which data is filtered out based on Geo IDs and KPI columns which are not null having source Trade Terms.
* Another ‘other’ table is created to deduct named customers from total customers.
* Data from the raw table and ‘other’ table is merged into Final table.
* Another table for each cluster is created which filters the records for Customer\_L4 as ‘Total’ and where all the measures for TT are either null or 0. Records are extracted from where GIV\_LC is null or 0 to get only unallocated records.

## Optima

## For CE

* Tables ‘CEEX\_OPT\_CE\_UKR’ and ‘WEEX\_OPT\_CE\_UKR’ are created for Central and Western Europe respectively to calculate SD KPI and join it with customer mapping table (mnr\_cust\_mapng\_dim\_vw\_opt) and Calendar dimension (SODE\_CAL\_FDIM). SD is calculated as the sum of (ESTMT\_FIXED\_MDA\_LOR) and (ESTMT\_VARIABLE\_MDA\_LOR).
* Another table ‘OPTIMA\_UNION\_CE\_UKR’ is created to merge the records of Central and Western Europe for Optima.
* Table ‘OPTIMA\_DR\_SD\_CE\_UKR’ is created to extract the optima data for CE and Ukraine.
* Table ‘OPTIMA\_DR\_FJOIN\_SD\_CE\_UKR’ is created to extract the records where GIV\_LC is null to get only unallocated records. The data is then joined with the shipment data for respective Geo IDs.
* Table ‘OPTIMA\_DR\_FILTERED\_SD\_CE\_UKR’ is created for filtering out the data based on Month date where Month\_Start\_Date is greater than or equal to switch date for respective geo IDs .
* Table ‘OPTIMA\_SD\_DIRECT\_MERGE\_CE\_UKR’ is created to merge the optima data of SEE and CE region.

## For Nordics

* Tables ‘CEEX\_OPT\_NORDICS’ and ‘WEEX\_OPT\_ NORDICS’ are created for Central and Western Europe respectively to calculate SD KPI and join it with customer mapping table (mnr\_cust\_mapng\_dim\_vw\_opt) and Calendar dimension (SODE\_CAL\_FDIM). SD is calculated as the sum of (ESTMT\_FIXED\_MDA\_LOR) and (ESTMT\_VARIABLE\_MDA\_LOR).
* Another table ‘OPTIMA\_UNION\_ NORDICS’ is created to merge the records of Central and Western Europe for Optima.
* Table ‘OPTIMA\_DR\_SD\_ NORDICS’ is created to extract the optima data for Nordics.
* Table ‘OPTIMA\_DR\_FJOIN\_SD\_ NORDICS’ is created to extract the records where GIV\_LC is null to get only unallocated records. The data is then joined with the shipment data for respective Geo IDs.
* Table ‘OPTIMA\_DR\_FILTERED\_SD\_ NORDICS’ is created for filtering out the data based on Month date where Month\_Start\_Date is greater than or equal to switch date for respective geo IDs .
* Table ‘OPTIMA\_SD\_DIRECT\_MERGE\_ NORDICS’ is created to merge the optima data of Nordics region.

## For other clusters

* Table is created for each cluster to extract the Optima data from CE and Nordics.
* Another table (FJOIN) is created to extract the records where GIV\_LC is null to get only unallocated records. The data is then joined with the shipment data for respective Geo IDs.
* Table (Filtered) for filtering out the data based on Month date where Month\_Start\_Date is greater than or equal to switch date for respective geo IDs is created.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Source | Cluster Name | CMF\_SD | NATSD | Optima | TT |
| Direct | CCAR | 1 | 0 | 1 | 0 |
| Direct | CE\_UKR | 1 | 1 | 1 | 1 |
| Direct | DACH | 1 | 1 | 1 | 1 |
| Direct | EE | 1 | 1 | 0 | 0 |
| Direct | France | 1 | 1 | 1 | 0 |
| Direct | Iberia | 1 | 1 | 1 | 1 |
| Direct | Italy | 1 | 1 | 1 | 1 |
| Direct | Nordics | 1 | 1 | 1 | 1 |
| Direct | SEE | 1 | 1 | 1 | 1 |
| Direct | Turkey | 1 | 0 | 1 | 0 |
| Direct | UKI | 1 | 1 | 1 | 0 |
| Indirect | CAU | 1 | 0 | 1 | 1 |
| Indirect | EE | 1 | 1 | 0 | 0 |
| Indirect | SEE | 1 | 1 | 1 | 1 |
| Indirect | Turkey | 1 | 0 | 1 | 1 |

## Optima

* Optima is defined for UKI, France and Turkey
* In notebook ‘00\_EUR\_OPTIMA\_COMMON’ two load tables ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_LOAD’ and ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_LOAD’ are created using parquet file from the blob storage for Central Europe and Western Europe respectively. As data for Optima is ingested on daily basis a separate table to specify data based on working day table ‘wk\_day\_date’ is created using a csv file from blob storage.
* Two empty staging tables ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_STG’ and ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_STG’ are created in which data is inserted in the form of snapshots on daily basis from the load tables with current timestamp as ‘RECORD\_LOAD\_TIMESTAMP’ to differentiate snapshot based on time and date.
* Table ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT’ is created in which latest data is selected from staging table using inner join condition on staging table in which ‘RECORD\_LOAD\_TIMESTAMP’ is described as Load Date column (To fetch the date at which snapshot was created) and Max\_TS column (To select the latest snapshot).
* Similarly, table ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT’ is created for Western Europe in which latest data in form of snapshot is selected from staging table using same logic.
* In notebook ‘01\_EUR\_OPTIMA\_DIRECT\_PROCESSING’ data belonging to direct shipments is processed, the table ‘FISCAL\_YEAR’ is created for selecting current and previous fiscal year.
* Tables ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_WORK’ and ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_WORK’ are created to extract the data for previous two fiscal years from ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT’ and ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT’ respectively.
* Table ‘mnr\_cust\_mapng\_dim\_vw\_opt’ is created in which filters are applied to Direct shipment customers. Customers are defined at levels L1, L2, L3 and L4 as CUSTOMER\_L1, CUSTOMER\_L2, CUSTOMER\_L3 and CUSTOMER\_L4. The records of customers not belonging to any above-mentioned levels are coalesced as ‘All Other’. For selecting data belonging to only Direct customers condition SRC\_DIRCT\_VAL= 'Y' is used.
* Tables ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD’ and ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD’ are created from ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_WORK’ and ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_WORK’ respectively to calculate SD KPI. The table is joined with Customer mapping table (mnr\_cust\_mapng\_dim\_vw\_opt) and Calendar dimension (SODE\_CAL\_FDIM CAL) on Optima ID and Month number respectively. Bus units 'ROMANIA', 'BULGARIA', 'FRANCE FOOD', 'FRANCE NON-FOOD/PGP' and 'UK' are omitted out. SD is calculated as sum of ‘ESTMT\_FIXED\_MDA\_LOR’ and ‘ESTMT\_VARIABLE\_MDA\_LOR’.
* Table ‘OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD\_UNION’ is created to merge the data for Central and Western Europe of tables ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD’ and ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD’.
* Table ‘DIRECT\_SH\_GIV\_BRAND’ is created to allocate GIV (GIV\_LC) from direct shipment data on month, Customer (Customer\_L4) and Brand ID basis, which is termed as GIV\_LC\_BRAND from ‘step\_100\_fx\_rates\_dir’ table for all geo IDs except '250', '642', '100' and '826'.
* Table ‘DIRECT\_SH\_GIV\_BRAND\_ALLOC’ is created to calculate Brand optima key (BRAND\_GIV\_KEY) for direct shipments from ‘step\_100\_fx\_rates\_dir’. Brand optima key (BRAND\_GIV\_KEY) is defined as ratio of GIV\_LC and GIV\_LC\_BRAND.
* Table ‘DIRECT\_SH\_OPTIMA\_ALL’ is created to allocate optima to shipment data. Table is joined with table ‘DIRECT\_SH\_GIV\_BRAND\_ALLOC’ on month number (Month\_ID), product id (Form\_ID) and customer (CUSTOMER\_L4). It is also joined with table ‘OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD\_UNION’ on month start date (MTH\_START\_DATE), product id (Brand\_ID) and customer (CUSTOMER\_L4). Optima SD (Optima\_SD) is calculated as the product of Brand optima key (BRAND\_GIV\_KEY) and SD KPI. Countries belonging to geo IDs (Geo\_ID) '250', '642', '100' and '826' are excluded.
* In notebook ‘02\_EUR\_OPTIMA\_INDIRECT\_PROCESSING’ data belonging to indirect shipments is processed.
* Table ‘mnr\_cust\_mapng\_dim\_vw\_opt\_ind’ is created in which filters are applied to Direct shipment customers. Customers are defined at levels L1, L2, L3 and L4 as CUSTOMER\_L1, CUSTOMER\_L2, CUSTOMER\_L3 and CUSTOMER\_L4. The records of customers not belonging to any above-mentioned levels are coalesced as ‘All Other’. For selecting data belonging to only Indirect customers condition SRC\_INDIR\_VAL= 'Y' is used.
* Tables ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_IND\_SD’ and ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_IND\_SD’ are created from ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_WORK’ and ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_WORK’ respectively to calculate SD KPI. The table is joined with Customer mapping table (mnr\_cust\_mapng\_dim\_vw\_opt) and Calendar dimension (SODE\_CAL\_FDIM CAL) on Optima ID and Month number respectively. Bus units (countries) 'ROMANIA', 'BULGARIA', 'FRANCE FOOD', 'FRANCE NON-FOOD/PGP' and 'UK' are omitted out. SD is calculated as sum of ‘ESTMT\_FIXED\_MDA\_LOR’ and ‘ESTMT\_VARIABLE\_MDA\_LOR’.
* Table ‘OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_IND\_SD\_UNION’ is created to merge the data for Central and Western Europe of tables ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_IND\_SD’ and ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_IND\_SD’.
* Table ‘INDIRECT\_SH\_GIV\_BRAND’ is created to allocate GIV (GIV\_LC) from indirect shipment data on month, Customer (Customer\_L4) and Brand ID basis, which is termed as GIV\_LC\_BRAND from ‘eu\_ind\_customer\_final’ table for all geo IDs except '250', '642', '100' and '826'.
* Table ‘INDIRECT\_SH\_GIV\_BRAND\_ALLOC’ is created to calculate Brand optima key (BRAND\_GIV\_KEY) for direct shipments from ‘eu\_ind\_customer\_final’. Brand optima key (BRAND\_GIV\_KEY) is defined as ratio of GIV\_LC and GIV\_LC\_BRAND.
* Table ‘INDIRECT\_SH\_OPTIMA\_ALL’ is created to allocate optima to shipment data. Table is joined with table ‘INDIRECT\_SH\_GIV\_BRAND\_ALLOC’ on month number (Month\_ID), product id (Form\_ID) and customer (CUSTOMER\_L4). It is also joined with table ‘OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_IND\_SD\_UNION’ on month start date (MTH\_START\_DATE), product id (Brand\_ID) and customer (CUSTOMER\_L4). Optima SD (Optima\_SD) is calculated as the product of Brand optima key (BRAND\_GIV\_KEY) and SD KPI. Countries belonging to geo IDs (Geo\_ID) '250', '642', '100' and '826' are excluded.
* In notebook ‘03\_EUR\_OPTIMA\_DIRECT\_PROCESS\_TURKEY’ optima data of direct shipments for TURKEY is processed.
* Table ‘EUR\_REP\_TUR\_DIR\_OPTIMA’ is created using table ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FC’. It is joined internally with table ‘wk\_day\_date’ to extract records only for working days. Other joins are implemented with Calendar (SODE\_CAL\_FDIM) dimension on month name (MTH\_NAME) and with Customer mapping table (‘mnr\_cust\_mapng\_dim\_vw\_opt’) for Turkey to map the data based on optima ID (OPTIMA\_ID). Another join with Product mapping table (‘mnr\_prod\_mapng\_dim\_vw\_v’) is implemented to map data based on Brand ID (BRAND\_ID). The country name (BUS\_UNIT\_NAME), geo id (Geo\_ID) and workday (wd) is hardcoded as ‘Turkey’, ‘792’ and ‘15’ respectively.
* Table ‘eur\_rep\_dir\_co\_brd\_agg\_cache\_turkey’ is created to allocate GIV from shipment data on month, Customer\_L4 and Brand\_ID from table from table ‘eu\_dir\_turkey\_customer\_final’ for Turkey (GEO\_ID = '792’).
* Table ‘eur\_rep\_dir\_opt\_key\_turkey’ is created in which Brand optima key (BRAND\_GIV\_KEY) is defined as ratio of GIV\_LC and GIV\_LC\_BRAND for Turkey.
* Table ‘eur\_rep\_dir\_opt\_all\_turkey’ is created using table ‘eu\_dir\_turkey\_customer\_final’ in which Optima SD (Optima\_SD) is calculated as the product of Brand optima key (BRAND\_GIV\_KEY) and SD KPI.
* Table ‘eur\_rep\_dir\_crncy\_turkey’is created using table ‘eur\_rep\_dir\_opt\_all\_turkey’ which is joined with currency geo ID table ‘eur\_rep\_dir\_crncy\_geoid’ to calculate C$ (OPTIMA\_SD\_USD\_CST) and R$ (OPTIMA\_SD\_USD\_REP) as product of Optima SD and Exchange rate (REP\_EXCHG\_RATE for R$ and CST\_EXCHG\_RATE for C$).
* In ‘04\_EUR\_OPTIMA\_INDIRECT\_PROCESS\_TURKEY’ optima data of indirect shipments for TURKEY is processed suing similar logic as that of direct Shipments.
* Table ‘EUR\_REP\_TUR\_IND\_OPTIMA’ is created using table ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT’. It is joined internally with table ‘wk\_day\_date’ to extract records only for working days. Other joins are implemented with Calendar (SODE\_CAL\_FDIM) dimension on month name (MTH\_NAME) and with Customer mapping table (‘mnr\_cust\_mapng\_dim\_vw\_opt’) for Turkey to map the data based on optima ID (OPTIMA\_ID). Another join with Product mapping table (‘mnr\_prod\_mapng\_dim\_vw\_v’) is implemented to map data based on Brand ID (BRAND\_ID). The country name (BUS\_UNIT\_NAME), geo id (Geo\_ID) and workday (wd) is hardcoded as ‘Turkey’, ‘792’ and ‘15’ respectively.
* Table ‘eur\_rep\_ind\_co\_brd\_agg\_cache\_turkey’ is created to allocate GIV from shipment data on month, Customer\_L4 and Brand\_ID from table from table ‘eu\_ind\_customer\_final\_tur’ for Turkey (GEO\_ID = '792’).
* Table ‘eur\_rep\_ind\_opt\_key\_turkey’ is created in which Brand optima key (BRAND\_GIV\_KEY) is defined as ratio of GIV\_LC and GIV\_LC\_BRAND for Turkey.
* Table ‘eur\_rep\_ind\_opt\_all\_turkey’ is created using table ‘eu\_ind\_customer\_final\_tur’ in which Optima SD (Optima\_SD) is calculated as the product of Brand optima key (BRAND\_GIV\_KEY) and SD KPI.
* Table ‘eur\_rep\_ind\_crncy\_turkey’ is created using table ‘eur\_rep\_ind\_opt\_all\_turkey’ which is joined with currency geo ID table ‘eur\_rep\_indir\_crncy\_geoid’ to calculate C$ (OPTIMA\_SD\_USD\_CST) and R$ (OPTIMA\_SD\_USD\_REP) as product of Optima SD and Exchange rate (REP\_EXCHG\_RATE for R$ and CST\_EXCHG\_RATE for C$).
* In notebook ‘05\_EUR\_OPTIMA\_DIRECT\_PROCESS\_UK’ optima data of direct shipments for UK is processed.
* Tables ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD\_UK’ for central Europe and ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD\_UK’ for western Europe is created using table ‘CEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_WORK’ and ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD\_UK’ respectively. These tables are used to calculate SD KPI as sum of ‘ESTMT\_FIXED\_MDA\_LOR’ and ‘ESTMT\_FIXED\_MDA\_LOR’.It is joined internally with table ‘wk\_day\_date’ to extract records only for working days. Other joins are implemented with Calendar (SODE\_CAL\_FDIM) dimension on month name (MTH\_NAME) and with Customer mapping table (‘mnr\_cust\_mapng\_dim\_vw\_opt’) for UK to map the data based on optima ID (OPTIMA\_ID). Records for all Brand IDs from Product mapping table ‘mnr\_prod\_mapng\_dim\_vw\_v’ are selected except the records in which ‘PROD\_HIER\_EFF\_DATE’ is either null or less than the current date. The country name (BUS\_UNIT\_NAME), geo id (Geo\_ID) and workday (wd) is hardcoded as ‘UK’, ‘826’ and ‘15’ respectively.
* Table ‘OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD\_UNION\_UK’ is created to merge the records for Central and Western Europe selected in previous step for further processing. SD is further classified as SD\_OI If ACTVY\_TYPE is equal to “LTOI” or “STOI” it is being treated as “SD On Invoice” and SD\_nOI, if ACTVY\_TYPE is not equal to “LTOI” or “STOI” it is being treated as “SD not On Invoice”
* Table ‘DIRECT\_SH\_GIV\_BRAND\_UK’ is created to allocate GIV from shipment data on month, Customer\_L4 and Brand\_ID from table ‘step\_100\_fx\_rates\_dir’ for UK (GEO\_ID = '826’).
* Table ‘DIRECT\_SH\_GIV\_BRAND\_ALLOC\_UK’ is created in which Brand optima key (BRAND\_GIV\_KEY) is defined as ratio of GIV\_LC and GIV\_LC\_BRAND for UK.
* Table ‘DIRECT\_SH\_OPTIMA\_ALL\_UK’ is created using table ‘step\_100\_fx\_rates\_dir’ to allocate final optima to shipment data of UK in which Optima SD (Optima\_SD) is calculated as the product of Brand optima key (BRAND\_GIV\_KEY) and SD, Optima\_SD\_OI is calculated as product of SD\_OI and Brand optima key (BRAND\_GIV\_KEY) and Optima\_SD\_nOI is calculated as product of SD\_nOI and Brand optima key (BRAND\_GIV\_KEY).
* Table ‘eur\_rep\_dir\_crncy\_uk’ is created using table ‘DIRECT\_SH\_OPTIMA\_ALL\_UK’ which is joined with currency geo ID table ‘eur\_rep\_dir\_crncy\_geoid’ to calculate C$ (OPTIMA\_SD\_USD\_CST) and R$ (OPTIMA\_SD\_USD\_REP) as product of Optima SD and Exchange rate (REP\_EXCHG\_RATE for R$ and CST\_EXCHG\_RATE for C$).
* In notebook ‘07\_EUR\_OPTIMA\_DIRECT\_PROCESS\_IRELAND’ optima data of direct shipments for IRELAND is processed using the same logic as for UK with tables specified for Ireland and Geo\_ID ‘372’.
* In notebook ‘06\_EUR\_OPTIMA\_DIRECT\_PROCESS\_FR’ optima data of direct shipments for France is processed.
* Optima data for France is divided for into Food and Non-food, thus separate tables for Food ‘mnr\_cust\_mapng\_dim\_vw\_FF\_FR’ and non-food ‘mnr\_cust\_mapng\_dim\_vw\_FNF\_FR’ are created in which filters are applied for Direct shipment customers of France (Geo\_ID =’250’).
* Table ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD\_OI\_FR’ is created to calculate SD\_OI KPI and to join it with customer mapping table and calendar dimension. SD\_OI is calculated as sum of (ESTMT\_FIXED\_MDA\_LOR) and (ESTMT\_VARIABLE\_MDA\_LOR). An inner join with table ‘wk\_day\_date’ is implemented to extract data only for working days. Further it is joined with calendar dimension based on month name (MTH\_NAME) and with customer mapping table ‘mnr\_cust\_mapng\_dim\_vw\_FF\_FR’ based on Optima ID for (BUS\_UNIT\_NAME) as France Food for 15th workday, where activity type (ACTVY\_TYPE) is among 'RSF Promo', 'TPR Promo' or 'RSF Long Term'. Records for all Brand IDs from Product mapping table ‘mnr\_prod\_mapng\_dim\_vw\_v’ are selected except the records in which ‘PROD\_HIER\_EFF\_DATE’ is either null or less than the current date.
* Similarly it is joined with table ’ mnr\_cust\_mapng\_dim\_vw\_FNF\_FR’ based on Optima ID for (BUS\_UNIT\_NAME) as France Non-Food.
* Same logic is used to calculate SD\_nOI in table ‘WEEX\_OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD\_NOI\_FR’.
* Table ‘OPT\_ACCT\_BRAND\_ACTVY\_SPDNG\_FCT\_SD\_UNION\_FR’ is created to merge the data for SD\_OI and SD\_nOI for further processing. SD is taken as sum of SD\_I and SD\_nOI.
* Table ‘DIRECT\_SH\_GIV\_BRAND\_FR’ is created to allocate GIV from shipment data on month, Customer\_L4 and Brand\_ID from table ‘step\_100\_fx\_rates\_dir’ for France (GEO\_ID = '250’).
* Table ‘DIRECT\_SH\_GIV\_BRAND\_ALLOC\_FR’ is created in which Brand optima key (BRAND\_GIV\_KEY) is defined as ratio of GIV\_LC and GIV\_LC\_BRAND for France.
* Table ‘DIRECT\_SH\_OPTIMA\_ALL\_FR’ is created using table ‘step\_100\_fx\_rates\_dir’ to allocate final optima to shipment data of France in which Optima SD (Optima\_SD) is calculated as the product of Brand optima key (BRAND\_GIV\_KEY) and SD, Optima\_SD\_OI is calculated as product of SD\_OI and Brand optima key (BRAND\_GIV\_KEY) and Optima\_SD\_nOI is calculated as product of SD\_nOI and Brand optima key (BRAND\_GIV\_KEY).
* Table ‘eur\_rep\_dir\_crncy\_fra’ is created using table ‘DIRECT\_SH\_OPTIMA\_ALL\_FR’ which is joined with currency geo ID table ‘eur\_rep\_dir\_crncy\_geoid’ to calculate C$ (OPTIMA\_SD\_USD\_CST) and R$ (OPTIMA\_SD\_USD\_REP) as product of Optima SD and Exchange rate (REP\_EXCHG\_RATE for R$ and CST\_EXCHG\_RATE for C$).

## Anaplan

Anaplan is defined for France and UKI.

For France following notebooks and their description is as follows :

## 02\_FRANCE\_ANAPLAN\_SHIPMENTS

* Table ‘step\_100\_fx\_rates\_dir\_anaplan\_shipment’ is refreshed to refresh the data of Anaplan for direct shipments.
* Table ‘eur\_anaplan\_shipments’ is created using direct shipment table ‘step\_100\_fx\_rates\_dir\_anaplan\_shipment’ to select specified columns from direct shipment table and thereafter map it with customer mapping to obtain Market type and aggregate Vol , GIV , NIV, NIV2\_DISC and NIV2 based on it and product ID, product name.Scope of customer and product are limited based on customer mapping and product mapping. Customer mapping table (mnr\_cust\_mapng\_dim\_vw\_v) is joined on Customer L4 except ‘Total’ and ‘All Other’ (RPTNG\_CUST\_L4\_NAME in customer mapping and CUSTOMER\_L4 in direct shipment table) along with organization ID for Geo ID belonging to France (250). Organization ID “90096433” is defined as FF (France Food) and all others are defined as FNF (France Non-Food).
* Table ‘eur\_ana\_fra\_shipments\_fx’ is created to calculate shipment KPIs of Local currency GIV\_LC, NIV\_LC and NIV2\_LC. GIV\_LC is calculated as the sum of product of GIV and Final exchange rate (FINAL\_EXCHG\_RATE). NIV\_LC is calculated as sum of product of NIV and Final exchange rate (FINAL\_EXCHG\_RATE). NIV2\_LC is calculated as sum of product of NIV2 and Final exchange rate (FINAL\_EXCHG\_RATE). They are aggregated based on customer ID, customer name, geo ID, geo name, financial year ID, month, segment ID, segment name, category ID, category name, brand ID, brand name, subsector ID, subsector name, FPC ID and FPC name.

## 03\_FRANCE\_ANAPLAN\_FPC\_GTIN\_PROC

* A customer mapping view (eur\_ana\_fra\_customer\_map\_vw) is created from customer mapping dimension (mnr\_cust\_mapng\_dim) to create a separate customer mapping for France Anaplan. The market is hardcoded as ‘FF’ and ‘FNF’ (France food and France non-food) separately and merged using Union clause with geo id ‘250’ (France) and organization ID being neither ‘90096437’ nor ‘90096433’. Source belongs to Direct shipment only.
* A table ‘eur\_ana\_fra\_cu\_gtin\_mapping’ is created to get the data at CU\_GTIN level from the common table ‘pea\_fpc\_to\_cugtin\_mapping’. Subsector ID, Category ID, Brand ID, and Segment ID are selected from product hierarchy (PROD\_HIER\_DIM) and a join is performed with CU\_GTIN table (CU\_GTIN\_NON\_OPTIMA) to map them at cu\_gtin level. A filter is applied to select ISO\_CNTRY\_CODE (geo id) from France Anaplan customer mapping view (eur\_ana\_fra\_customer\_map\_vw). Product hierarchy ID (PROD\_HIER\_ID) is hardcoded as ‘5005’ where CURR\_IND = 'Y' and Stage ID does not belong to ‘06’ or ‘07’.
* A Table of GTINs master list is created as ‘eur\_ana\_fra\_cu\_gtin\_ML’ to get the data at CU\_GTIN level from the common table ‘pea\_fpc\_to\_cugtin\_mapping’. Subsector ID, Category ID, Brand ID, and Segment ID are selected from product hierarchy (PROD\_HIER\_DIM) and a join is performed with CU\_GTIN table (CU\_GTIN\_NON\_OPTIMA) to map them at cu\_gtin level. A filter is applied to select ISO\_CNTRY\_CODE for cu gtin table from geo hierarchy where geo hierarchy ID (GEO\_HIER\_ID) is set to be ‘705’ and GEO\_11\_ISO\_CNTRY\_NUM is selected from France Anaplan customer mapping view (eur\_ana\_fra\_customer\_map\_vw). Product hierarchy ID (PROD\_HIER\_ID) and GEO\_ID are hardcoded as ‘5005’ and ‘250’ repectively where CURR\_IND = 'Y'.
* Table ‘eur\_ana\_fra\_cu\_gtin\_ML\_FY’ is created to join each GTIN from master list to financial year.
* Table ‘eur\_ana\_fra\_shipments\_cu\_gtin’ to add VOL to FPC-GTIN mapping and complementation of product hierarchy.

## 04\_FRANCE\_ANAPLAN\_TDC\_IYA

* Table ‘eur\_ana\_fra\_IYA\_TDC\_join’ is created using table ‘eur\_ana\_fra\_shipments\_fx’to calculate TDC\_SU as the ratio of ‘TDC\_LC’ and ‘VOL’ from eur\_ana\_fra\_shipments\_fx.. All records from table ‘eur\_rep\_for\_tdc\_su’ is selected for France (Geo\_ID = ‘250’) and DIRECT customer type and is defined with alias ‘eur\_rep\_for\_tdc\_su\_fbnl’ which is joined with ‘eur\_ana\_fra\_shipments\_fx’ on Geo ID, FPC ID ( Product\_Id for ‘eur\_rep\_for\_tdc\_su\_fbnl’), customer name and time\_id. TDC\_LC is selected from ‘eur\_rep\_for\_tdc\_su\_fbnl’.
* Table ‘eur\_ana\_fra\_IYA\_TDC\_seg\_avg’ is created to calculate TDC average (TDC\_SU) based on segment ID. TDC\_SU average is calculated as the ratio of sum of product of TDC\_SU (eur\_ana\_fra\_IYA\_TDC\_join) and VOL (eur\_ana\_fra\_IYA\_TDC\_join) and sum of all VOL from table (eur\_ana\_fra\_IYA\_TDC\_join).
* Table ‘eur\_ana\_fra\_IYA\_TDC\_brd\_avg’ is created to calculate TDC average (TDC\_SU) based on brand ID. TDC\_SU average is calculated as the ratio of sum of product of TDC\_SU (eur\_ana\_fra\_IYA\_TDC\_join) and VOL (eur\_ana\_fra\_IYA\_TDC\_join) and sum of all VOL from table (eur\_ana\_fra\_IYA\_TDC\_join).
* Table ‘eur\_ana\_fra\_IYA\_TDC\_cat\_avg’ is created to calculate TDC average (TDC\_SU) based on Category ID. TDC\_SU average is calculated as the ratio of sum of product of TDC\_SU (eur\_ana\_fra\_IYA\_TDC\_join) and VOL (eur\_ana\_fra\_IYA\_TDC\_join) and sum of all VOL from table (eur\_ana\_fra\_IYA\_TDC\_join).
* Table ‘eur\_ana\_fra\_IYA\_TDC\_all\_avg’ is created using table ‘eur\_ana\_fra\_IYA\_TDC\_join’ to accommodate all values of TDC\_SU based on FPC, segment, Brand or category. If value of TDC\_SU is zero for FPC then it’s taken for segment. If TDC\_SU is zero at both FPC and segment level then it’s taken based on brand otherwise it’s taken for category. Table ‘eur\_ana\_fra\_IYA\_TDC\_join’ is joined with ‘eur\_ana\_fra\_IYA\_TDC\_seg\_avg’, ‘eur\_ana\_fra\_IYA\_TDC\_brd\_avg’ and ‘eur\_ana\_fra\_IYA\_TDC\_cat\_avg’ on Segment\_ID , Brand\_ID and Category\_ID respectively along with FY\_ID.

## 05\_FRANCE\_ANAPLAN\_TDC\_ROI

* Table ‘eur\_ana\_fra\_ROI\_TDC\_join’ is created using table ‘eur\_ana\_fra\_shipments\_cu\_gtin’ to calculate TDC\_SU at GTIN level. A join is performed with shipment table for France ‘eur\_ana\_fra\_shipments\_fx’ on Geo ID, FY ID, Subsector ID, Category ID, Brand ID, Segment ID and FPC ID.
* Table ‘eur\_ana\_fra\_ROI\_TDC\_seg\_avg’ is created to calculate TDC average (TDC\_SU) based on segment ID. TDC\_SU average is calculated as the ratio of sum of product of TDC\_SU (eur\_ana\_fra\_IYA\_TDC\_join) and VOL (eur\_ana\_fra\_IYA\_TDC\_join) and sum of all VOL from table (eur\_ana\_fra\_IYA\_TDC\_join).
* Table ‘eur\_ana\_fra\_ROI\_TDC\_brd\_avg’ is created to calculate TDC average (TDC\_SU) based on brand ID. TDC\_SU average is calculated as the ratio of sum of product of TDC\_SU (eur\_ana\_fra\_IYA\_TDC\_join) and VOL (eur\_ana\_fra\_IYA\_TDC\_join) and sum of all VOL from table (eur\_ana\_fra\_IYA\_TDC\_join).
* Table ‘eur\_ana\_fra\_ROI\_TDC\_cat\_avg’ is created to calculate TDC average (TDC\_SU) based on Category ID. TDC\_SU average is calculated as the ratio of sum of product of TDC\_SU (eur\_ana\_fra\_IYA\_TDC\_join) and VOL (eur\_ana\_fra\_IYA\_TDC\_join) and sum of all VOL from table (eur\_ana\_fra\_IYA\_TDC\_join).
* Table ‘eur\_fra\_ROI\_TDC\_all\_avg’ is created using table ‘eur\_ana\_fra\_cu\_gtin\_ML\_FY’ to accommodate all values of TDC\_SU based on GTIN, segment, Brand, or category. If value of TDC\_SU is zero or null for GTIn then it’s taken for segment. If TDC\_SU is zero at both GTIN and segment, then it’s taken based on brand otherwise it’s taken for category.Table ‘eur\_ana\_fra\_ROI\_TDC\_join’ is joined with ‘eur\_ana\_fra\_IYA\_TDC\_seg\_avg’, ‘eur\_ana\_fra\_IYA\_TDC\_brd\_avg’ and ‘eur\_ana\_fra\_IYA\_TDC\_cat\_avg’ on Segment\_ID , Brand\_ID and Category\_ID respectively along with FY\_ID.

## 06\_FRANCE\_ANAPLAN\_TDC\_OUTPUT

* Table ‘eur\_ana\_fra\_iya\_BRD\_C1’ is created using ‘eur\_ana\_fra\_IYA\_TDC\_all\_avg’ in which Customer ID, absolute value of TDC and a code is defined. They are defined as:
* Customer\_ID – Brand ID, Market type and customer ID are concatenated to get.
* TDC\_absolute\_value- sum of product of TDC\_SU and Vol.
* Code – Brand\_ID, market, customer\_ID, Month, ‘FY’, FY\_ID and ‘Local Currency’ are concatenated to form code.
* Table ‘eur\_ana\_fra\_ROI\_BRD\_C1’ is created using ‘eur\_fra\_ROI\_TDC\_all\_avg’ in which Code based on brand is defined for ‘FF’ (France Food) and ‘FnF’ (France non-food) separately and is merged using Union clause. Code is defined as concatenation of FF/FnF, Brand ID, ‘FY’, FY\_ID and ‘Local Currency’
* Table ‘eur\_ana\_fra\_ROI\_CAT\_C1’ is created using ‘eur\_fra\_ROI\_TDC\_all\_avg’ in which Code based on category is defined for ‘FF’ (France Food) and ‘FnF’ (France non-food) separately and is merged using Union clause. Code is defined as concatenation of FF/FnF, category ID, ‘FY’, FY\_ID and ‘Local Currency’.
* Table ‘eur\_ana\_fra\_ROI\_GTIN\_C1’ is created using ‘eur\_fra\_ROI\_TDC\_all\_avg’ in which Code based on GTIN is defined for ‘FF’ (France Food) and ‘FnF’ (France non-food) separately and is merged using Union clause. Code is defined as concatenation of FF/FnF, CU\_GTIN, ‘FY’, FY\_ID and ‘Local Currency’.
* Table ‘eur\_ana\_fra\_ROI\_SEG\_C1’ is created using ‘eur\_fra\_ROI\_TDC\_all\_avg’ in which Code based on segment is defined for ‘FF’ (France Food) and ‘FnF’ (France non-food) separately and is merged using Union clause. Code is defined as concatenation of FF/FnF, Segment ID, ‘FY’, FY\_ID and ‘Local Currency’.

## 07\_FRANCE\_ANAPLAN\_COEFFICIENTS

* Table ‘eur\_ana\_fra\_sh\_act\_cat\_agg’ is created to aggregate actual shipments to category level.
* Table ‘eur\_ana\_fra\_coef\_natsd\_act\_perc’ is created to convert National SD from absolute to % GIV. NATSD\_PERC (National SD Percentage) is calculated as ration of NAT\_SD\_VAL and GIV\_LC. A join is performed with table ‘mnr\_sd\_natl\_dirct\_L4\_dim\_vw’ to select records for France (geo\_id 250).
* Table ‘eur\_ana\_fra\_coef\_regsd\_act\_giv\_perc’ is created to convert Regional SD absolutes from %NIV2 to %GIV. REGNL\_SD\_GIV\_PERC (Regional SD Percentage) is calculated as ratio of REGNL\_SD\_ABS and GIV\_LC. Regional SD absolute values are calculated as product of REGNL\_SD\_VAL and NIV2\_LC from table ‘mnr\_sd\_regnl\_dirct\_std\_dim\_vw’. A join is performed with table ‘mnr\_sd\_regnl\_dirct\_std\_dim\_vw’ to select records for France (geo\_id 250).
* Table ‘eur\_ana\_fra\_sh\_fcst\_avg’ is created to calculate forecast of shipments.
* Table ‘eur\_ana\_fra\_sh\_fcst\_cat\_agg’ is created to aggregate forecast based on category.
* Table ‘eur\_ana\_fra\_coef\_regsd\_fcst\_giv\_perc’ is created to convert regional SD forecast from %NIV2 to %GIV. REGNL\_SD\_GIV\_PERC (Regional SD Percentage) is calculated as ratio of REGNL\_SD\_ABS and GIV\_LC. Regional SD absolute values are calculated as product of REGNL\_SD\_VAL and NIV2\_LC from table ‘eur\_ana\_fra\_sh\_fcst\_cat\_agg’. A join is performed with table ‘mnr\_sd\_regnl\_dirct\_fcst\_dim\_vw’ to select records for France (geo\_id 250).
* Table ‘eur\_ana\_fra\_coef\_act\_all’ is created to allocate actual values of coefficients.
* Table ‘eur\_ana\_fra\_coef\_fcst\_all’ is created to allocate forecast values of coefficients.

## 08\_FRANCE\_ANAPLAN\_COEFFICIENTS\_OUTPUT

* Table ‘eur\_ana\_fra\_SD\_TT\_C1’ is created to allocate actual and forecast percentages for Trade terms on invoice and Trade Terms not on invoice.
* They are calculated as:
  + Trade\_terms\_On\_Invoice\_Pct = TT\_OI\_PERC \* 100.
  + Trade\_terms\_Off\_Invoice\_Pct = TT\_NOI\_PERC \* 100
  + National\_SD\_nOI\_Pct = NATSD\_PERC \* 100
  + Regional\_SD\_nOI\_Pct= REGNL\_SD\_GIV\_PERC \* 100
  + Cash\_Discount\_OI\_Pct = TT\_CASH\_DISC\_PERC \* 100.
* GIV\_LC, NIV2\_LC, TT\_NOI\_PERC, TT\_OI\_PERC, TT\_CASH\_DISC\_PERC, NATSD\_PERC, REGNL\_SD\_GIV\_PERC are aggregated based on category, Customer ID, geo name, month, and fiscal year for both actual and forecast values.

For UKI notebooks and their description is as follows:

## 01\_UKI\_ANAPLAN\_SHIPMENT

* Table ‘eur\_ana\_uki\_shipments’ is created to extract shipment data for UKI from table ‘step\_100\_fx\_rates\_dir\_anaplan\_shipment’ and scope of customer and product are limited based on customer mapping and product mapping. Customer mapping table (mnr\_cust\_mapng\_dim\_vw\_v) is joined on Customer L4 except ‘Total’ and ‘All Other’ (RPTNG\_CUST\_L4\_NAME in customer mapping and CUSTOMER\_L4 in direct shipment table) along with organization ID for Geo ID belonging to UKI region i.e., ‘826’ and ‘372’.
* Table ‘eur\_ana\_uki\_shipments\_fx’ is created to calculate shipment KPIs of Local currency GIV\_LC, NIV\_LC and NIV2\_LC. GIV\_LC is calculated as the sum of product of GIV and Final exchange rate (FINAL\_EXCHG\_RATE). NIV\_LC is calculated as sum of product of NIV and Final exchange rate (FINAL\_EXCHG\_RATE). NIV2\_LC is calculated as sum of product of NIV2 and Final exchange rate (FINAL\_EXCHG\_RATE). They are aggregated based on customer ID, customer name, geo ID, geo name, financial year ID, month, segment ID, segment name, category ID, category name, brand ID, brand name, subsector ID, subsector name, FPC ID and FPC name.

## 02\_UKI\_ANAPLAN\_FPC\_GTIN\_PROC

* A customer mapping view (eur\_ana\_uki\_customer\_map\_vw) is created from customer mapping dimension (mnr\_cust\_mapng\_dim) to create a separate customer mapping for UKI Anaplan. Customer ID, customer name, optima ID, optima name, organization ID, organization name, Geo ID, geo name, and customer levels (L1, L2, L3 and L4) are selected for Geo Ids ‘826’ and ‘372’. Source belongs to Direct shipment only.
* Table ‘eur\_ana\_uki\_cu\_gtin\_mapping’ is created to get the data at CU\_GTIN level from the common table ‘pea\_fpc\_to\_cugtin\_mapping’. Subsector ID, Category ID, Brand ID, and Segment ID are selected from product hierarchy (PROD\_HIER\_DIM) and a join is performed with CU\_GTIN table (CU\_GTIN\_NON\_OPTIMA) to map them at cu\_gtin level. A filter is applied to select ISO\_CNTRY\_CODE (geo id) from UKI Anaplan customer mapping view (eur\_ana\_uki\_customer\_map\_vw). Product hierarchy ID (PROD\_HIER\_ID) is hardcoded as ‘5005’ where CURR\_IND = 'Y' and Stage ID does not belong to ‘06’ or ‘07’.
* A Table of GTINs master list is created as ‘eur\_ana\_uki\_cu\_gtin\_ML’ to get the data at CU\_GTIN level from the common table ‘pea\_fpc\_to\_cugtin\_mapping’. Subsector ID, Category ID, Brand ID, and Segment ID are selected from product hierarchy (PROD\_HIER\_DIM) and a join is performed with CU\_GTIN table (CU\_GTIN\_NON\_OPTIMA) to map them at cu\_gtin level. A filter is applied to select ISO\_CNTRY\_CODE for cu gtin table from geo hierarchy where geo hierarchy ID (GEO\_HIER\_ID) and GEO\_11\_ISO\_CNTRY\_NUM is selected from UKI Anaplan customer mapping view (eur\_ana\_uki\_customer\_map\_vw). Product hierarchy ID (PROD\_HIER\_ID) where CURR\_IND = 'Y'.
* Table ‘eur\_ana\_uki\_cu\_gtin\_ML\_FY’ is created to join each GTIN from master list to financial year.
* Table ‘eur\_ana\_uki\_shipments\_cu\_gtin’ to add VOL to FPC-GTIN mapping and complementation of product hierarchy.

## 03\_UKI\_ANAPLAN\_TDC\_IYA

* Table ‘eur\_ana\_uki\_IYA\_TDC\_join’ is created using table ‘eur\_ana\_uki\_shipments\_fx’to calculate TDC\_SU as the ratio of ‘TDC\_LC’ and ‘VOL’ from eur\_ana\_uki\_shipments\_fx.. All records from table ‘eur\_rep\_for\_tdc\_su’ is selected for UKI (Geo\_ID = ‘826’ and ‘372’) and DIRECT customer type and is defined with alias ‘eur\_rep\_for\_tdc\_su\_fbnl’ which is joined with ‘eur\_ana\_fra\_shipments\_fx’ on Geo ID, FPC ID ( Product\_Id for ‘eur\_rep\_for\_tdc\_su\_fbnl’), customer name and time\_id. TDC\_LC is selected from ‘eur\_rep\_for\_tdc\_su\_fbnl’.
* Table ‘eur\_ana\_uki\_IYA\_TDC\_seg\_avg’ is created to calculate TDC average (TDC\_SU) based on segment ID. TDC\_SU average is calculated as the ratio of sum of product of TDC\_SU (eur\_ana\_uki\_IYA\_TDC\_join) and VOL (eur\_ana\_uki\_IYA\_TDC\_join) and sum of all VOL from table (eur\_ana\_uki\_IYA\_TDC\_join).
* Table ‘eur\_ana\_uki\_IYA\_TDC\_brd\_avg’ is created to calculate TDC average (TDC\_SU) based on brand ID. TDC\_SU average is calculated as the ratio of sum of product of TDC\_SU (eur\_ana\_uki\_IYA\_TDC\_join) and VOL (eur\_ana\_uki\_IYA\_TDC\_join) and sum of all VOL from table (eur\_ana\_uki\_IYA\_TDC\_join).
* Table ‘eur\_ana\_uki\_IYA\_TDC\_cat\_avg’ is created to calculate TDC average (TDC\_SU) based on Category ID. TDC\_SU average is calculated as the ratio of sum of product of TDC\_SU (eur\_ana\_uki\_IYA\_TDC\_join) and VOL (eur\_ana\_uki\_IYA\_TDC\_join) and sum of all VOL from table (eur\_ana\_uki\_IYA\_TDC\_join).
* Table ‘eur\_ana\_uki\_IYA\_TDC\_all\_avg’ is created using table ‘eur\_ana\_uki\_IYA\_TDC\_join’ to accommodate all values of TDC\_SU based on FPC, segment, Brand, or category. If value of TDC\_SU is zero for FPC then it’s taken for segment. If TDC\_SU is zero at both FPC and segment level, then it’s taken based on brand otherwise it’s taken for category. Table ‘eur\_ana\_uki\_IYA\_TDC\_join’ is joined with ‘eur\_ana\_uki\_IYA\_TDC\_seg\_avg’, ‘eur\_ana\_uki\_IYA\_TDC\_brd\_avg’ and ‘eur\_ana\_uki\_IYA\_TDC\_cat\_avg’ on Segment\_ID , Brand\_ID and Category\_ID respectively along with FY\_ID.

## 04\_UKI\_ANAPLAN\_TDC\_ROI

* Table ‘eur\_ana\_uki\_ROI\_TDC\_join’ is created using table ‘eur\_ana\_uki\_shipments\_cu\_gtin’ to calculate TDC\_SU at GTIN level. A join is performed with shipment table for UKI ‘eur\_ana\_uki\_shipments\_fx’ on Geo ID, FY ID, Subsector ID, Category ID, Brand ID, Segment ID and FPC ID.
* Table ‘eur\_ana\_uki\_ROI\_TDC\_seg\_avg’ is created to calculate TDC average (TDC\_SU) based on segment ID. TDC\_SU average is calculated as the ratio of sum of product of TDC\_SU (eur\_ana\_uki\_IYA\_TDC\_join) and VOL (eur\_ana\_uki\_IYA\_TDC\_join) and sum of all VOL from table (eur\_ana\_uki\_IYA\_TDC\_join).
* Table ‘eur\_ana\_uki\_ROI\_TDC\_brd\_avg’ is created to calculate TDC average (TDC\_SU) based on brand ID. TDC\_SU average is calculated as the ratio of sum of product of TDC\_SU (eur\_ana\_uki\_IYA\_TDC\_join) and VOL (eur\_ana\_uki\_IYA\_TDC\_join) and sum of all VOL from table (eur\_ana\_uki\_IYA\_TDC\_join).
* Table ‘eur\_ana\_uki\_ROI\_TDC\_cat\_avg’ is created to calculate TDC average (TDC\_SU) based on Category ID. TDC\_SU average is calculated as the ratio of sum of product of TDC\_SU (eur\_ana\_uki\_IYA\_TDC\_join) and VOL (eur\_ana\_uki\_IYA\_TDC\_join) and sum of all VOL from table (eur\_ana\_uki\_IYA\_TDC\_join).
* Table ‘eur\_uki\_ROI\_TDC\_all\_avg’ is created using table ‘eur\_ana\_uki\_cu\_gtin\_ML\_FY’ to accommodate all values of TDC\_SU based on GTIN, segment, Brand, or category. If value of TDC\_SU is zero or null for GTIn then it’s taken for segment. If TDC\_SU is zero at both GTIN and segment, then it’s taken based on brand otherwise it’s taken for category.Table ‘eur\_ana\_uki\_ROI\_TDC\_join’ is joined with ‘eur\_ana\_uki\_IYA\_TDC\_seg\_avg’, ‘eur\_ana\_uki\_IYA\_TDC\_brd\_avg’ and ‘eur\_ana\_uki\_IYA\_TDC\_cat\_avg’ on Segment\_ID , Brand\_ID and Category\_ID respectively along with FY\_ID.

## 05\_UKI\_ANAPLAN\_TDC\_OUTPUTS

* Table ‘eur\_ana\_uki\_IYA\_BRD\_C1’ is created using ‘eur\_ana\_uki\_IYA\_TDC\_all\_avg’ in which Customer ID, absolute value of TDC and a code is defined. They are defined as:
* Customer\_ID – For Great Britain (geo ID = ‘826’) it is defined as concatenation of ‘GB\_C1\_’ and Customer ID. For Ireland (geo ID = ‘372’) it is defined as concatenation of ‘IE\_C1\_’ and Customer ID.
* TDC\_absolute\_value- sum of product of TDC\_SU and Vol.
* Code – For Great Britain (geo ID = ‘826’) Brand\_ID, ‘GB\_C1\_’, customer\_ID, Month, ‘FY’, FY\_ID and ‘Local Currency’ are concatenated to form code.
* For Ireland (geo ID = ‘372’) Brand\_ID, ‘IE\_C1\_’, customer\_ID, Month, ‘FY’, FY\_ID and ‘Local Currency’ are concatenated to form code.
* Table ‘eur\_ana\_uki\_IYA\_BRD\_C2’ is created using ‘eur\_ana\_uki\_IYA\_TDC\_all\_avg’ in which Customer ID, absolute value of TDC and a code is defined. They are defined as:
* Customer\_ID – For Great Britain (geo ID = ‘826’) it is defined as concatenation of ‘GB\_C2\_’ and Customer ID. For Ireland (geo ID = ‘372’) it is defined as concatenation of ‘IE\_C2\_’ and Customer ID.
* TDC\_absolute\_value- sum of product of TDC\_SU and Vol.
* Code – For Great Britain (geo ID = ‘826’) Brand\_ID, ‘GB\_C2\_’, customer\_ID, Month, ‘FY’, FY\_ID and ‘Local Currency’ are concatenated to form code.
* For Ireland (geo ID = ‘372’) Brand\_ID, ‘IE\_C2\_’, customer\_ID, Month, ‘FY’, FY\_ID and ‘Local Currency’ are concatenated to form code.
* Table ‘eur\_ana\_uki\_IYA\_CAT\_C1’ is created using ‘eur\_uki\_IYA\_TDC\_all\_avg’ in which Customer ID, absolute value of TDC and a code is defined. They are defined as:
* Customer\_ID – For Great Britain (geo ID = ‘826’) it is defined as concatenation of ‘GB\_C1\_’ and Customer ID. For Ireland (geo ID = ‘372’) it is defined as concatenation of ‘IE\_C1\_’ and Customer ID.
* TDC\_absolute\_value- sum of product of TDC\_SU and Vol.
* Code – For Great Britain (geo ID = ‘826’) Brand\_ID, ‘GB\_C1\_’, customer\_ID, Month, ‘FY’, FY\_ID and ‘Local Currency’ are concatenated to form code.
* For Ireland (geo ID = ‘372’) Brand\_ID, ‘IE\_C1\_’, customer\_ID, Month, ‘FY’, FY\_ID and ‘Local Currency’ are concatenated to form code.
* Table ‘eur\_ana\_uki\_IYA\_CAT\_C2’ is created using ‘eur\_uki\_IYA\_TDC\_all\_avg’ in which Customer ID, absolute value of TDC and a code is defined. They are defined as:
* Customer\_ID – For Great Britain (geo ID = ‘826’) it is defined as concatenation of ‘GB\_C2\_’ and Customer ID. For Ireland (geo ID = ‘372’) it is defined as concatenation of ‘IE\_C2\_’ and Customer ID.
* TDC\_absolute\_value- sum of product of TDC\_SU and Vol.
* Code – For Great Britain (geo ID = ‘826’) Brand\_ID, ‘GB\_C2\_’, customer\_ID, Month, ‘FY’, FY\_ID and ‘Local Currency’ are concatenated to form code.
* For Ireland (geo ID = ‘372’) Brand\_ID, ‘IE\_C2\_’, customer\_ID, Month, ‘FY’, FY\_ID and ‘Local Currency’ are concatenated to form code.
* Table ‘eur\_ana\_uki\_ROI\_BRD\_C1’ is created using ‘eur\_uki\_ROI\_TDC\_all\_avg’ in which Code based on brand is defined for ‘GB’ (Great Britain) and ‘IE’ (Ireland). Code is defined as concatenation of ‘GB\_C1\_’/’[IE\_C1\_’, Brand ID, ‘FY’, FY\_ID and ‘Local Currency’
* Table ‘eur\_ana\_uki\_ROI\_BRD\_C2’ is created using ‘eur\_uki\_ROI\_TDC\_all\_avg’ in which Code based on brand is defined for ‘GB’ (Great Britain) and ‘IE’ (Ireland). Code is defined as concatenation of ‘GB\_C2\_’/’IE\_C2\_’, Brand ID, ‘FY’, FY\_ID and ‘Local Currency’
* Table ‘eur\_ana\_uki\_ROI\_CAT\_C1’ is created using ‘eur\_uki\_ROI\_TDC\_all\_avg’ in which Code based on category is defined for ‘GB’ (Great Britain) and ‘IE’ (Ireland). Code is defined as concatenation of ‘GB\_C1\_’/’IE\_C1\_’, Brand ID, ‘FY’, FY\_ID and ‘Local Currency’
* Table ‘eur\_ana\_uki\_ROI\_CAT\_C2’ is created using ‘eur\_uki\_ROI\_TDC\_all\_avg’ in which Code based on category is defined for ‘GB’ (Great Britain) and ‘IE’ (Ireland). Code is defined as concatenation of ‘GB\_C2\_’/’IE\_C2\_’, Brand ID, ‘FY’, FY\_ID and ‘Local Currency’
* Table ‘eur\_ana\_uki\_ROI\_GTIN\_C1’ is created using ‘eur\_uki\_ROI\_TDC\_all\_avg’ in which Code based on GTIN is defined for ‘GB’ (Great Britain) and ‘IE’ (Ireland). Code is defined as concatenation of ‘GB\_C1\_’/’IE\_C1\_’, Brand ID, ‘FY’, FY\_ID and ‘Local Currency’
* Table ‘eur\_ana\_uki\_ROI\_GTIN\_C2’ is created using ‘eur\_uki\_ROI\_TDC\_all\_avg’ in which Code based on GTIN is defined for ‘GB’ (Great Britain) and ‘IE’ (Ireland). Code is defined as concatenation of ‘GB\_C2\_’/’IE\_C2\_’, Brand ID, ‘FY’, FY\_ID and ‘Local Currency’
* Table ‘eur\_ana\_uki\_ROI\_SEG\_C1’ is created using ‘eur\_uki\_ROI\_TDC\_all\_avg’ in which Code based on segment is defined for ‘GB’ (Great Britain) and ‘IE’ (Ireland). Code is defined as concatenation of ‘GB\_C1\_’/’IE\_C1\_’, Brand ID, ‘FY’, FY\_ID and ‘Local Currency’
* Table ‘eur\_ana\_uki\_ROI\_SEG\_C2’ is created using ‘eur\_uki\_ROI\_TDC\_all\_avg’ in which Code based on segment is defined for ‘GB’ (Great Britain) and ‘IE’ (Ireland). Code is defined as concatenation of ‘GB\_C2\_’/’IE\_C2\_’, Brand ID, ‘FY’, FY\_ID and ‘Local Currency’

## 06\_UKI\_ANAPLAN\_COEFFICIENTS

* Table ‘eur\_ana\_uki\_sh\_act\_cat\_agg’ is created to aggregate actual shipments to category level.
* Table ‘eur\_ana\_uki\_coef\_natsd\_act\_perc’ is created to convert National SD from absolute to % GIV. NATSD\_PERC (National SD Percentage) is calculated as ratio of NAT\_SD\_VAL and GIV\_LC. A join is performed with table ‘mnr\_sd\_natl\_dirct\_L4\_dim\_vw’ to select records for UKI (geo\_id 826 or 372).
* Table ‘eur\_ana\_uki\_coef\_regsd\_act\_giv\_perc’ is created to convert Regional SD absolutes from %NIV2 to %GIV. REGNL\_SD\_GIV\_PERC (Regional SD Percentage) is calculated as ratio of REGNL\_SD\_ABS and GIV\_LC. Regional SD absolute values are calculated as product of REGNL\_SD\_VAL and NIV2\_LC from table ‘mnr\_sd\_regnl\_dirct\_std\_dim\_vw’. A join is performed with table ‘mnr\_sd\_regnl\_dirct\_std\_dim\_vw’ to select records for UKI (geo\_id 826 or 372).
* Table ‘eur\_ana\_uki\_sh\_fcst\_avg’ is created to calculate forecast of shipments.
* Table ‘eur\_ana\_uki\_sh\_fcst\_cat\_agg’ is created to aggregate forecast based on category.
* Table ‘eur\_ana\_uki\_coef\_regsd\_fcst\_giv\_perc’ is created to convert regional SD forecast from %NIV2 to %GIV. REGNL\_SD\_GIV\_PERC (Regional SD Percentage) is calculated as ratio of REGNL\_SD\_ABS and GIV\_LC. Regional SD absolute values are calculated as product of REGNL\_SD\_VAL and NIV2\_LC from table ‘eur\_ana\_uki\_sh\_fcst\_cat\_agg’. A join is performed with table ‘mnr\_sd\_regnl\_dirct\_fcst\_dim\_vw’ to select records for UKI (geo\_id 826 or 372).
* Table ‘eur\_ana\_uki\_coef\_act\_all’ is created to allocate actual values of coefficients.
* Table ‘eur\_ana\_uki\_coef\_fcst\_all’ is created to allocate forecast values of coefficients.

## 07\_UKI\_ANAPLAN\_COEFFICIENTS\_OUTPUTS

* Table ‘eur\_ana\_gbr\_SD\_TT\_C1’ is created to allocate actual and forecast percentages for Trade terms on invoice and Trade Terms not on invoice for Great Britain.
* They are calculated as:
  + Trade\_terms\_On\_Invoice\_Pct = TT\_OI\_PERC \* 100.
  + Trade\_terms\_Off\_Invoice\_Pct = TT\_NOI\_PERC \* 100.
  + National\_SD\_nOI\_Pct = NATSD\_PERC \* 100.
  + Regional\_SD\_nOI\_Pct= REGNL\_SD\_GIV\_PERC \* 100
  + Cash\_Discount\_OI\_Pct = TT\_CASH\_DISC\_PERC \* 100.
* GIV\_LC, NIV2\_LC, TT\_NOI\_PERC, TT\_OI\_PERC, TT\_CASH\_DISC\_PERC, NATSD\_PERC, REGNL\_SD\_GIV\_PERC are aggregated based on category, Customer ID, geo name, month, and fiscal year for both actual and forecast values.
* Table ‘eur\_ana\_gbr\_SD\_TT\_C1’ is created to allocate actual and forecast percentages for Trade terms on invoice and Trade Terms not on invoice for Great Britain.
* They are calculated as:
  + Trade\_terms\_On\_Invoice\_Pct = TT\_OI\_PERC \* 100.
  + Trade\_terms\_Off\_Invoice\_Pct = TT\_NOI\_PERC \* 100.
  + National\_SD\_nOI\_Pct = NATSD\_PERC \* 100.
  + Regional\_SD\_nOI\_Pct= REGNL\_SD\_GIV\_PERC \* 100
  + Cash\_Discount\_OI\_Pct = TT\_CASH\_DISC\_PERC \* 100.
* GIV\_LC, NIV2\_LC, TT\_NOI\_PERC, TT\_OI\_PERC, TT\_CASH\_DISC\_PERC, NATSD\_PERC, REGNL\_SD\_GIV\_PERC are aggregated based on category, Customer ID, geo name, month, and fiscal year for both actual and forecast values.
* Table ‘eur\_ana\_irl\_SD\_TT\_C1’ is created to allocate actual and forecast percentages for Trade terms on invoice and Trade Terms not on invoice for Ireland.
* They are calculated as:
  + Trade\_terms\_On\_Invoice\_Pct = TT\_OI\_PERC \* 100.
  + Trade\_terms\_Off\_Invoice\_Pct = TT\_NOI\_PERC \* 100.
  + National\_SD\_nOI\_Pct = NATSD\_PERC \* 100.
  + Regional\_SD\_nOI\_Pct= REGNL\_SD\_GIV\_PERC \* 100
  + Cash\_Discount\_OI\_Pct = TT\_CASH\_DISC\_PERC \* 100.
* GIV\_LC, NIV2\_LC, TT\_NOI\_PERC, TT\_OI\_PERC, TT\_CASH\_DISC\_PERC, NATSD\_PERC, REGNL\_SD\_GIV\_PERC are aggregated based on category, Customer ID, geo name, month, and fiscal year for both actual and forecast values.
* Table ‘eur\_ana\_irl\_SD\_TT\_C2’ is created to allocate actual and forecast percentages for Trade terms on invoice and Trade Terms not on invoice for Ireland.
* They are calculated as:
  + Trade\_terms\_On\_Invoice\_Pct = TT\_OI\_PERC \* 100.
  + Trade\_terms\_Off\_Invoice\_Pct = TT\_NOI\_PERC \* 100.
  + National\_SD\_nOI\_Pct = NATSD\_PERC \* 100.
  + Regional\_SD\_nOI\_Pct= REGNL\_SD\_GIV\_PERC \* 100
  + Cash\_Discount\_OI\_Pct = TT\_CASH\_DISC\_PERC \* 100.
* GIV\_LC, NIV2\_LC, TT\_NOI\_PERC, TT\_OI\_PERC, TT\_CASH\_DISC\_PERC, NATSD\_PERC, REGNL\_SD\_GIV\_PERC are aggregated based on category, Customer ID, geo name, month, and fiscal year for both actual and forecast values.

## TDC Smoothing

## 110\_EUROPE\_TDC\_FACT\_PROCESS

* Aggregated sum(vol) from stage europe.step\_100\_fx\_rates\_dir direct shipment by using below group by columns

Group by : FY\_ID

GEO\_ID

CURRENCY\_ID

PROD\_4\_ID\_5005

PROD\_5\_ID\_5005

PROD\_6\_ID\_5005

PROD\_7\_ID\_5005

PROD\_8\_ID\_5005

PRODUCT\_ID

* Cretated stage\_europe.europe\_shipment\_agg table by combining europe\_shipment\_agg & direct\_shipment\_agg
* Created stage\_europe.europe\_mnr\_tdc\_su as a TDC source from stage\_mmr.mnr\_tdc\_su\_dim\_vw and filtered only EUROPE countries by doing inner join with stage\_mmr.mnr\_geo\_mapng\_dim\_vw\_unified\_v
* Created stage\_europe.europe\_tdc\_smoothing\_step10 table by using stage\_europe.europe\_shipment\_agg and stage\_europe.europe\_mnr\_tdc\_su by taking max(vol) from shipment and MAX(TDC\_PER\_SU) from tdc source table by taking below columns as group by
  + FY\_ID, FY\_ORDER, GEO\_ID, FPC\_ID, SUBSECTOR\_ID,CATEGORY\_ID, BRAND\_ID, .SEGMENT\_ID,

BRAND\_FORM\_ID

* Created stage\_europe.europe\_tdc\_smoothing\_step20 by doing pivot of stage\_europe.europe\_tdc\_smoothing\_step10 and also created nominator and denominator of average TDC by applying below logic
  + NVL(CFY\_TDC, 0) > 0 THEN CFY\_TDC \* CFY\_VOL ELSE null END as WAVG\_NOM
  + NVL(CFY\_TDC, 0) > 0 THEN CFY\_VOL ELSE null END as WAVG\_DEN
* Created stage\_europe.europe\_tdc\_smoothing\_step30 by using stage\_europe.europe\_tdc\_smoothing\_step20 by calculating weighted average from shipment for every hierarchy level like..
* brand\_form\_sub, segment\_sub, brand\_sub, category\_sub, subsector\_sub based on required group by for each hierarchy levels
* Created stage\_europe.europe\_tdc\_smoothing\_step20 by using stage\_europe.europe\_tdc\_smoothing\_step30 by calculating missing CFY\_TDC based on weighted average
* Updated europe.europe\_tdc\_smoothing\_step20 by update PFY TDC based on 10% variance between CFY\_TDC and PFY\_TDC and Update FY\_ID for calculated missing source rows
* Created stage\_mmr.europe\_mnr\_fx\_rate\_tdc\_dim\_vw\_v by creating TDC exchange rate fact to convert USD to Local Currency and for Europe June month exchange rates are considered for the current and previous financial year
* Created stage\_europe.europe\_tdc\_amt\_lc by using stage\_europe.europe\_tdc\_smoothing\_step20 with shipment data by creating TDC\_AMT by multiplying volume with TDC\_PER\_SU
* Created stage\_europe.europe\_tdc\_amt\_rusd\_v by multiplying TDC\_AMT with reporting dollar value along with multiplier value for local currency
* Created stage\_europe. europe\_tdc\_amt\_cusd\_v\_v by multiplying TDC\_AMT with constant dollar value
* Combined all constant TDC\_AMT, reporting TDC\_AMT with TDC\_AMT tables and created final stage\_europe.tdc\_europe\_fct as output of TDC smoothing.

## 560\_EUR\_PEA\_FACT\_PROCESS

* Aggregated sum(vol) from stage europe.step\_100\_fx\_rates\_dir direct shipment by using below group by columns
  + Group by : FY\_ID
  + GEO\_ID
  + CURRENCY\_ID
  + PROD\_4\_ID\_5005
  + PROD\_5\_ID\_5005
  + PROD\_6\_ID\_5005
  + PROD\_7\_ID\_5005
  + PROD\_8\_ID\_5005
  + PRODUCT\_ID
* Created base for FPC to CUGTIN mapping by using shipment and stage\_common.pea\_fpc\_to\_cugtin\_mapping by naming table as stage\_europe.prod\_fpc\_gtin\_assoc\_geo\_4\_5\_6 and made CU\_GTIN column with 14 digits,
* Created stage\_europe.prod\_fpc\_gtin\_assoc\_filtered by using stage\_europe.prod\_fpc\_gtin\_assoc\_geo\_4\_5\_6, stage\_common.SODE\_P5005\_FDIM and stage\_mmr.mnr\_geo\_mapng\_dim\_vw\_v along with shipment data in this based in geo level created each geo\_id
* Created stage\_europe.europe\_mnr\_tdc\_su as a TDC source from stage\_mmr.mnr\_tdc\_su\_dim\_vw and filtered only EUROPE countries by doing inner join with stage\_mmr.mnr\_geo\_mapng\_dim\_vw\_unified\_v
* Created stage\_europe.europe\_tdc\_smoothing\_step10 table by using stage\_europe.europe\_shipment\_agg and stage\_europe.europe\_mnr\_tdc\_su by taking max(vol) from shipment and MAX(TDC\_PER\_SU) from tdc source table by taking below columns as group by
  + FY\_ID, FY\_ORDER, GEO\_ID, FPC\_ID, SUBSECTOR\_ID,CATEGORY\_ID, BRAND\_ID, .SEGMENT\_ID,

BRAND\_FORM\_ID

* Created stage\_europe.europe\_tdc\_smoothing\_step20 by doing pivot of stage\_europe.europe\_tdc\_smoothing\_step10 and also created nominator and denominator of average TDC by applying below logic
  + NVL(CFY\_TDC, 0) > 0 THEN CFY\_TDC \* CFY\_VOL ELSE null END as WAVG\_NOM
  + NVL(CFY\_TDC, 0) > 0 THEN CFY\_VOL ELSE null END as WAVG\_DEN
* Created stage\_europe.europe\_tdc\_smoothing\_step30 by using stage\_europe.europe\_tdc\_smoothing\_step20 by calculating weighted average from shipment for every hierarchy level like..
* brand\_form\_sub, segment\_sub, brand\_sub, category\_sub, subsector\_sub based on required group by for each hierarchy levels
* Created stage\_europe.europe\_tdc\_smoothing\_step20 by using stage\_europe.europe\_tdc\_smoothing\_step30 by calculating missing CFY\_TDC based on weighted average
* Updated europe.europe\_tdc\_smoothing\_step20 by update PFY TDC based on 10% variance between
* CFY\_TDC and PFY\_TDC and Update FY\_ID for calculated missing source rows
* Created stage\_europe.eur\_pea\_data\_join\_5005 by joining GTIN, shipment and TDC data
* Created stage\_europe.eur\_pea\_gtin\_avg by calculating average TDC per GTIN,
* Created stage\_europe.eur\_pea\_gtin\_level\_tdc\_avg by applying enrich logic for all hierarchy level,
  + when TDC\_CU from CU\_GTIN level is positive :) - apply
  + when TDC\_SU from CU\_GTIN level is NULL, equal to zero or negative - apply Brand Form average
  + when TDC\_SU average from Brand Form level is NULL, equal to zero or negative - apply Segment average
  + when TDC\_SU average from Segment level is NULL, equal to zero or negative - apply Brand average
  + when TDC\_SU average from Brand level is NULL, equal to zero or negative - apply Category average
  + when TDC\_SU average from Category level is equal to zero or negative - put NULL value as TDC\_SU
* Created stage\_europe.eur\_pea\_product\_levels\_tdc\_avg CATEGORY/BRAND/SEGMENT/BRAND FORM level by applying below enrich logic
* for TDC\_SU for a given product level
  + when TDC\_SU for a given product level is NULL, negative or zero, apply average from higher level, repeat untill first positive (and non-nullable by this) average is found
  + if TDC\_SU average for a category (highest level considered) is negative 0 - apply NULL
* Created stage\_europe.eur\_pea\_tdc\_crncy\_code by combining gtin level tdc average and product level tdc average for each geo level for shipment data
* Created stage\_europe.eur\_pea\_tdc\_fct\_before\_geo\_filter for updating timestamp for each load
* Created final stage\_europe.eur\_pea\_tdc\_fct for PEA TDC smooting logic

## PEA Sources

For PEA we have splitted based on common logic based on geo\_id levels:

## PEA\_ALB

## 01\_eur\_pea\_and\_customer\_final

Created stage\_europe.eur\_pea\_and\_customer\_final table by combining direct shipment and indirect shipment for geo specific, here filtered below countries for both shipments:

Indirect: '008', '051', '031', '070', '100', '268', 'E04', '807','498', '499','688'

Direct: '020', '208', '233', '246', '372', '428', '440', '578', '620', '703', '705', '756', '008', '051', '031', '070', '100', '268', 'E04', '807', '498', '499', '688', '991'

## 02\_eur\_pea\_fpc\_to\_gtin:

Created stage\_europe.eur\_pea\_fpc\_to\_gtin by using stage\_common.pea\_fpc\_to\_cugtin\_mapping and stage\_europe.PROD\_HIER\_DIM based on PROD\_19\_ID and prod\_id and joined stage\_europe.GEO\_HIER\_707 based on ISO\_CNTRY\_CODE and applied below where condition

PROD\_HIER\_DIM.PROD\_HIER\_ID = '5005'

CURR\_IND = 'Y'

STAGE\_ID <> '07'

STAGE\_ID <> '06'

## 03\_eur\_pea\_gtin\_fy

Created stage\_europe.eur\_pea\_gtin\_fy by joining shipment data with eur\_pea\_fpc\_to\_gtin table for below geo\_id’s

'020', '208', '233', '246', '372', '428', '440', '578', '620', '703', '705', '756', '008', '051', '031', '070', '100', '268', 'E04', '807', '498', '499', '688', '991'

## 04\_eur\_pea\_tdc\_join

Created stage\_europe.eur\_pea\_tdc\_join by joining TDC smoothing values for fpc to gtin table.

stage\_europe.eur\_pea\_gtin\_fy with stage\_europe.eur\_pea\_tdc\_fct

ON eur\_rep\_for\_tdc\_su\_see.geo\_id = eur\_pea\_fpc\_to\_gtin.GEO\_ID

AND eur\_rep\_for\_tdc\_su\_see.FPC\_ID = eur\_pea\_fpc\_to\_gtin.FPC\_ID

AND eur\_rep\_for\_tdc\_su\_see.FY\_ID = eur\_pea\_fpc\_to\_gtin.FY\_ID

And stage\_europe.eur\_pea\_and\_customer\_final based on r\_pea\_fpc\_to\_gtin.FPC\_ID = eur\_pea\_customer\_final.FPC\_ID

AND eur\_pea\_fpc\_to\_gtin.FY\_ID = eur\_pea\_customer\_final.FY\_ID

AND eur\_pea\_fpc\_to\_gtin.GEO\_ID = eur\_pea\_customer\_final.GEO\_ID

## 05\_eur\_pea\_avg

* Created stage\_europe.eur\_pea\_brd\_avg by calculating brand average where volume is not equal to ‘0’
* Created stage\_europe.eur\_pea\_cat\_avg by calculating category average where volume is not equal to ‘0’
* Created stage\_europe.eur\_pea\_sbstr\_avg by calculating sub sector average where volume is not equal to ‘0’
* Created stage\_europe.eur\_pea\_avg based on above created average tables and allocated tdc\_lc based on tdc value

## 06\_eur\_pea\_output:

Created final output as stage\_europe.eur\_pea\_output for above geo\_id’s by created start\_date and end\_date based on fiscal year values by joining stage\_europe.eur\_pea\_tdc\_fct table.

## PEA\_AUS

## 01\_eur\_pea\_aus\_customer\_final

Created stage\_europe.eur\_pea\_aus\_customer\_final by combining direct shipment by using stage\_europe.step\_100\_fx\_rates\_dir and for indirect shipment from stage\_europe.eu\_ind\_customer\_final and stage\_europe.eu\_ind\_customer\_final\_see tables

|  |  |  |  |
| --- | --- | --- | --- |
| 40, | 616, | 276, | 826, |
| 191, | 642, | 348, | 804, |
| 203, | 643, | 991, | 398, |
| 250, | 752, | 528, | 417, |
|  |  |  |  |

## 02\_eur\_pea\_aus\_tdc\_join

Creating stage\_europe.eur\_pea\_aus\_tdc\_join by getting tdc smoothed values from stage\_europe.eur\_pea\_tdc\_fct and combined with shipment data based on FPC\_ID, FY\_ID, CRNCY\_CODE and geo\_id

## 03\_eur\_pea\_aus\_fpc\_to\_gtin

Created stage\_europe. 03\_eur\_pea\_aus\_fpc\_to\_gtin by using stage\_common.pea\_fpc\_to\_cugtin\_mapping and stage\_europe.PROD\_HIER\_DIM based on PROD\_19\_ID and prod\_id and joined stage\_europe.GEO\_HIER\_707 based on ISO\_CNTRY\_CODE and applied below where condition

PROD\_HIER\_DIM.PROD\_HIER\_ID = '5005'

CURR\_IND = 'Y'

STAGE\_ID <> '07'

STAGE\_ID <> '06'

## 04\_eur\_pea\_aus\_gtin\_fy

Created stage\_europe. 04\_eur\_pea\_aus\_gtin\_fy by joining shipment data with eur\_pea\_fpc\_to\_gtin table for below geo\_id’s

|  |  |  |  |
| --- | --- | --- | --- |
| 40, | 616, | 276, | 826, |
| 191, | 642, | 348, | 804, |
| 203, | 643, | 991, | 398, |
| 250, | 752, | 528, | 417, |
|  |  |  |  |
|  |  |  |  |

## 05\_eur\_pea\_aus\_gtin\_ml\_fy

* Assigining all fysical year with shipment data and created stage\_europe.eur\_pea\_aus\_gtin\_ml\_fy
* Assigned TDC smoothed value in fpc\_id level and created stage\_europe.eur\_pea\_aus\_data\_join

## 06\_eur\_pea\_aus\_avg

* Created stage\_europe.eur\_pea\_aus\_gtin\_avg when vol <> 0 then UM( eur\_pea\_aus\_data\_join.TDC\_LC \* eur\_pea\_aus\_data\_join.VOL) / SUM(eur\_pea\_aus\_data\_join.VOL)
* Created stage\_europe.eur\_pea\_aus\_bform\_avg by calculating brand average where volume is not equal to ‘0’
* Created stage\_europe.eur\_pea\_aus\_brand\_avg by calculating brand average where volume is not equal to ‘0’
* Created stage\_europe.eur\_pea\_aus\_categ\_avg by calculating brand average where volume is not equal to ‘
* Created stage\_europe.eur\_pea\_brd\_avg by calculating brand average where volume is not equal to ‘0’
* Created stage\_europe.eur\_pea\_cat\_avg by calculating category average where volume is not equal to ‘0’
* Created stage\_europe.eur\_pea\_sbstr\_avg by calculating sub sector average where volume is not equal to ‘0’
* Created stage\_europe. eur\_pea\_aus\_segmn\_avg by calculating sub sector average where volume is not equal to ‘0’
* Created stage\_europe. eur\_pea\_aus\_avg based on above created average tables and allocated tdc\_lc based on tdc value

## 07\_eur\_pea\_aus\_agg\_ml

Created stage\_europe.eur\_pea\_aus\_agg\_ml table by doing union stage\_europe.eur\_pea\_aus\_avg for segment average tdc value for segment level, stage\_europe.eur\_pea\_aus\_avg BRAND level average and stage\_europe.eur\_pea\_aus\_avg CATEGORY level tdc value

## 08\_eur\_pea\_aus\_output

Created final output as stage\_europe. eur\_pea\_aus\_output for above geo\_id’s by created start\_date and end\_date based on fiscal year values by joining stage\_europe.eur\_pea\_tdc\_fct table.

## PEA\_BEL

## 01\_eur\_pea\_bel\_customer\_final

Create stage\_europe.eur\_pea\_bel\_customer\_final from direct shipment for geo\_id 918 only

## 02\_eur\_pea\_bel\_tdc\_join

Above created table has been joined with stage\_europe.eur\_pea\_tdc\_fct to get TDC\_PER\_SU as tdm smoothed value and created stage\_europe.eur\_pea\_bel\_tdc\_join

## 03\_eur\_pea\_bel\_fpc\_to\_gtin

Created stage\_europe. pea\_fpc\_to\_cugtin\_mapping by using stage\_common.pea\_fpc\_to\_cugtin\_mapping and stage\_europe.PROD\_HIER\_DIM based on PROD\_19\_ID and prod\_id and joined stage\_europe.GEO\_HIER\_707 based on ISO\_CNTRY\_CODE and applied below where condition

PROD\_HIER\_DIM.PROD\_HIER\_ID = '5005'

CURR\_IND = 'Y'

STAGE\_ID <> '07'

STAGE\_ID <> '06'

## 04\_eur\_pea\_bel\_gtin\_fy

Created stage\_europe. eur\_pea\_bel\_gtin\_fy by joining shipment data with eur\_pea\_fpc\_to\_gtin table

## 05\_eur\_pea\_bel\_gtin\_ml\_fy

Created stage\_europe. eur\_pea\_bel\_gtin\_ml\_fy by adding all fiscal year values stage\_europe.eur\_pea\_bel\_customer\_final

## 06\_eur\_pea\_bel\_data\_join

* Created tage\_europe.eur\_pea\_bel\_data\_join table by getting tdc smoothed value from stage\_europe.eur\_pea\_bel\_tdc\_join based on geo\_id, fy\_id and fpc\_id
* Created stage\_europe.eur\_pea\_bel\_gtin\_avg table where vol <> 0 then SUMeur\_pea\_bel\_data\_join.TDC\_LC \* eur\_pea\_bel\_data\_join.VOL ) / SUM(eur\_pea\_bel\_data\_join.VOL) as TDC\_SU

## 07\_eur\_pea\_bel\_avg

* Created stage\_europe.eur\_pea\_bel\_bform\_avg by calculating brand average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_bel\_gtin\_avg
* Created stage\_europe. eur\_pea\_bel\_brand\_avg by calculating category average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_bel\_gtin\_avg
* Created stage\_europe. eur\_pea\_bel\_brd\_avg by calculating sub sector average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_bel\_tdc\_join
* Created stage\_europe.eur\_pea\_bel\_cat\_ by calculating sub sector average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_bel\_tdc\_join
* Created stage\_europe.eur\_pea\_bel\_categ\_avg by calculating sub sector average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_bel\_gtin\_avg
* Created stage\_europe.eur\_pea\_bel\_sbstr\_avg by calculating sub sector average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_bel\_tdc\_join
* Created stage\_europe.eur\_pea\_bel\_segmn\_avg by calculating sub sector average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_bel\_gtin\_avg
* Created stage\_europe. eur\_pea\_bel\_avg based on above created average tables and allocated tdc\_lc based on tdc value from stage\_europe.eur\_pea\_bel\_gtin\_avg

## 08\_eur\_pea\_bel\_output

* Created stage\_europe.eur\_pea\_bel\_agg\_ml table by doing union stage\_europe.eur\_pea\_bel\_avg, stage\_europe.eur\_pea\_bel\_avg and stage\_europe.eur\_pea\_bel\_avg by taking TDC value based on product level
* Created final output as stage\_europe.eur\_pea\_bel\_output for above geo\_id’s by created start\_date and end\_date based on fiscal year values by joining stage\_europe.eur\_pea\_tdc\_fct table.

## PEA\_TUR

## 01\_eur\_pea\_tur\_customer\_final

Created stage\_europe.eur\_pea\_tur\_customer\_final by combining direct and indirect shipment data for '792', '300', '724' geo\_id’s

## 02\_eur\_pea\_tur\_fpc\_to\_gtin

* Created stage\_europe.eur\_pea\_tur\_tdc\_join by joining shipment data with has been joined with stage\_europe.eur\_pea\_tdc\_fct for tdc smoothed value
* Created stage\_europe.eur\_pea\_tur\_fpc\_to\_gtin by using stage\_common.pea\_fpc\_to\_cugtin\_mapping and stage\_europe.PROD\_HIER\_DIM based on PROD\_19\_ID and prod\_id and joined stage\_europe.GEO\_HIER\_707 based on ISO\_CNTRY\_CODE and applied below where condition

PROD\_HIER\_DIM.PROD\_HIER\_ID = '5005'

CURR\_IND = 'Y'

STAGE\_ID <> '07'

STAGE\_ID <> '06'

* Created stage\_europe.eur\_pea\_tur\_gtin\_fy by joining shipment data with eur\_pea\_fpc\_to\_gtin table
* Created stage\_europe.eur\_pea\_tur\_gtin\_ml\_fy by adding all fiscal year values stage\_europe.eur\_pea\_bel\_customer\_final
* Created stage\_europe.eur\_pea\_tur\_data\_join table by getting tdc smoothed value from stage\_europe.eur\_pea\_bel\_tdc\_join based on geo\_id, fy\_id and fpc\_id
* Created stage\_europe.eur\_pea\_tur\_gtin\_avg table where vol <> 0 then SUMeur\_pea\_bel\_data\_join.TDC\_LC \* eur\_pea\_bel\_data\_join.VOL ) / SUM(eur\_pea\_bel\_data\_join.VOL) as TDC\_SU

## 03\_eur\_pea\_tur\_avg

* Created stage\_europe.eur\_pea\_tur\_bform\_avg by calculating brand average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_tur\_gtin\_avg
* Created stage\_europe. eur\_pea\_tur\_brand\_avg by calculating category average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_tur\_gtin\_avg
* Created stage\_europe. eur\_pea\_tur\_brd\_avg by calculating sub sector average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_tur\_tdc\_join
* Created stage\_europe.eur\_pea\_tur\_cat\_ by calculating sub sector average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_tur\_tdc\_join
* Created stage\_europe.eur\_pea\_tur\_categ\_avg by calculating sub sector average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_tur\_gtin\_avg
* Created stage\_europe.eur\_pea\_tur\_sbstr\_avg by calculating sub sector average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_tur\_tdc\_join
* Created stage\_europe.eur\_pea\_tur\_segmn\_avg by calculating sub sector average where volume is not equal to ‘0’ from stage\_europe.eur\_pea\_tur\_gtin\_avg
* Created stage\_europe. eur\_pea\_tur\_avg based on above created average tables and allocated tdc\_lc based on tdc value from stage\_europe.eur\_pea\_tur\_gtin\_avg

## 04\_eur\_pea\_tur\_output

* Created stage\_europe.eur\_pea\_tur\_agg\_ml table by doing union stage\_europe.eur\_pea\_tur\_avg, stage\_europe.eur\_pea\_tur\_avg and stage\_europe.eur\_pea\_tur\_avg by taking TDC value based on product level
* Created final output as stage\_europe.eur\_pea\_tur\_output for above geo\_id’s by created start\_date and end\_date based on fiscal year values by joining stage\_europe.eur\_pea\_tdc\_fct table.

## Output Generation

## BNL OUTPUT CREATION

Please find attached individual document for BNL output creation



## CCAR\_OUTPUT\_CREATION

Please find attached individual document for CCAR output creation



## CE\_OUTPUT CREATION

Please find attached individual document for CE output creation



## DACH\_OUTPUT\_CREATION

Please find attached individual document for DACH output creation



## EE\_OUTPUT\_CREATION

Please find attached individual document for EE output creation



## FRANCE\_OUTPUT CREATION

Please find attached individual document for FRANCE output creation



## IBERIA OUTPUT CREATION

Please find attached individual document for IBERIA output creation



## ITALY\_OUTPUT CREATION

Please find attached individual document for ITALY output creation



## NORDICS OUTPUT CREATION

Please find attached individual document for NORDICS output creation



## SEE OUTPUT CREATION

Please find attached individual document for SEE output creation



## TURKEY\_OUTPUT CREATION

Please find attached individual document for TURKEY output creation



## UKI\_OUTPUT\_CREATION

Please find attached individual document for UKI output creation



## Data Ingestion

## 01\_CUSTOMER\_REGIONAL\_MANUAL\_MAPPING

Created stage\_europe.CUSTOMER\_REGIONAL\_DIM table and created in the for of parquet file in below location dbfs:/mnt/cngc-storage/europe/output/CUSTOMER\_REGIONAL\_DIM

Random row numbers has been created based on Geo\_ID,CNOSGC\_Reporting\_Customer,Customer\_Type columns

Selected below columns from stage\_europe.EU\_Manual\_Customer\_Mapping and do the union with other tables,

`GeoID` AS Geo\_ID,

`Country` AS Geo\_Name ,

`Customer Level 1` AS CNOSGC\_Customer\_Group\_1,

`Customer Level 2` AS CNOSGC\_Customer\_Group\_2,

`Customer Level 3` AS CNOSGC\_Customer\_Group\_3,

`Customer Level 4` AS CNOSGC\_Reporting\_Customer,

`Customer Type` AS Customer\_Type,

`Regional Customer Name 1` AS Reginal\_Customer\_name1,

`Regional Customer Name 2` AS Reginal\_Customer\_name2,

`Customer Name 1` AS Customer\_Name1,

`Channel` AS Channel ,

`DistributorEBG Name 2` AS DistributorEBG,

`DistributorEBG Type` AS DistributorEBG\_Type,

`Unofficial Channel Name` AS Unofficial\_Channel,

Current\_timestamp() SYS\_INSERTED\_DT

Created temp like below

'Unallocated Customer L1' as CNOSGC\_Customer\_Group\_1,

'Unallocated Customer L2' as CNOSGC\_Customer\_Group\_2,

'Unallocated Customer L3' as CNOSGC\_Customer\_Group\_3,

'Unallocated Customer L4' as CNOSGC\_Reporting\_Customer,

'Direct' as Customer\_Type

'Unallocated Customer L1' as CNOSGC\_Customer\_Group\_1,

'Unallocated Customer L2' as CNOSGC\_Customer\_Group\_2,

'Unallocated Customer L3' as CNOSGC\_Customer\_Group\_3,

'Unallocated Customer L4' as CNOSGC\_Reporting\_Customer,

'Indirect' as Customer\_Type

Cross joined with stage\_mmr.mnr\_geo\_mapng\_dim\_vw table

## 02\_DIMS AND FACT

* Created stage\_europe.CUSTOMER\_DIM table and created in the for of parquet file in below location

dbfs:/mnt/cngc-storage/europe/output/CUSTOMER\_DIM

Selected distinct RPTNG customer L1 – L4 from stage\_mmr.mnr\_cust\_mapng\_dim\_vw\_v for below where condition

* + mmr.SMO\_NAME IN ('EE','NE','Iberia','TCCAR','CE','FBNL','SEE','DACH','ITALY', 'SE')

and RPTNG\_CUST\_L4\_NAME not in (SELECT DISTINCT RPTNG\_CUST\_L4\_NAME FROM s tage\_europe.CUSTOMER\_SPLIT)

* Did the union with above data for joining stage\_mmr.mnr\_cust\_mapng\_dim\_vw\_v and stage\_europe.CUSTOMER\_SPLIT based on RPTNG\_CUST\_L4\_NAME for below ‘SMO\_NAME’ \

'EE','NE','Iberia','TCCAR','CE','FBNL','SEE','DACH','ITALY', 'SE'

Created some dummy columns as below and did union with above data

* + 'All Other' RPTNG\_CUST\_L1\_NAME,
  + 'All Other' RPTNG\_CUST\_L2\_NAME,
  + 'All Other' RPTNG\_CUST\_L3\_NAME,
  + 'All Other' RPTNG\_CUST\_L4\_NAME,
  + CURRENT\_TIMESTAMP() SYS\_INSERTED\_DT
  + 'Unallocated Customer L1' RPTNG\_CUST\_L1\_NAME,
  + 'Unallocated Customer L2' RPTNG\_CUST\_L2\_NAME,
  + 'Unallocated Customer L3' RPTNG\_CUST\_L3\_NAME,
  + 'Unallocated Customer L4' RPTNG\_CUST\_L4\_NAME,
  + CURRENT\_TIMESTAMP() SYS\_INSERTED\_DT
* Created stage\_europe.CUST\_REG\_CHNL\_DIM and loaded as parquet file in below location

path='dbfs:/mnt/cngc-storage/europe/output/CUST\_REG\_CHNL\_DIM' by using stage\_mmr.mnr\_cust\_mapng\_dim\_vw\_v created customertype column based on below logic

WHEN SRC\_DIRCT\_VAL='Y' and SRC\_INDIR\_VAL ='N' THEN 'DIRECT'

WHEN SRC\_DIRCT\_VAL='N' and SRC\_INDIR\_VAL ='Y' THEN 'INDIRECT'

Created below dummy columns and appended to above table

* + 'Unallocated Channel' AS CHANNEL,
  + 'DIRECT' AS CUSTOMERTYPE,
  + 'Unallocated Customer L4' AS CUSTOMERLEVEL4,
  + CURRENT\_TIMESTAMP() SYS\_INSERTED\_DT
  + 'All Other' AS CHANNEL,
  + 'DIRECT' AS CUSTOMERTYPE ,
  + 'All Other' AS CUSTOMERLEVEL4
* Created stage\_europe.CRNCY\_DIM from stage\_common.SODE\_CRNCY\_FDIM by using below filter condition

CURRENCY\_NAME IS NOT NULL and CURRENCY\_NAME NOT like('unknown%')

* Created stage\_europe.GEOGRAPHY\_DIM from stage\_mmr.mnr\_geo\_mapng\_dim\_vw where SMO\_NAME NOT IN ('AMA','LA','Japan') created below columns

Case when GEO\_ID\_LVL=4 then GEO\_4\_ID

when GEO\_ID\_LVL=5 then GEO\_5\_ID

when GEO\_ID\_LVL=6 then GEO\_6\_ID

End GEO\_ID,

Case when GEO\_ID\_LVL=4 then GEO\_4\_NAME

when GEO\_ID\_LVL=5 then GEO\_5\_NAME

when GEO\_ID\_LVL=6 then GEO\_6\_NAME

End GEO\_NAME

* Created stage\_europe.PRODUCT\_DIM from stage\_europe.eur\_rep\_output\_vw and created row number as product\_id
* Created stage\_europe.flat\_output\_fct table and created as parquet file in below location dbfs:/mnt/cngc-storage/europe/output/eur\_rep\_output\_vw/ from stage\_europe.eur\_rep\_output\_vw by joining stage\_europe.CUST\_REG\_CHNL\_DIM and stage\_europe.TIME\_DIM and stage\_mmr.mnr\_cust\_mapng\_dim\_vw\_v and stage\_europe.CUSTOMER\_REGIONAL\_DIM and also selected all columns from stage\_europe.eur\_rep\_output\_vw
* Created stage\_europe.TIME\_DIM and also loaded in the form of parquest in dbfs:/mnt/cngc-storage/europe/output/SODE\_CAL\_FDIM by joining tage\_common.SODE\_CAL\_FDIM with stage\_europe.flat\_output\_fct based on time\_id
* Created stage\_europe.TIME\_DIM\_ROLLING and also created as parquest file in dbfs:/mnt/cngc-storage/europe/output/TIME\_DIM\_ROLLING by using stage\_common.SODE\_CAL\_FDIM

## Automated Data Validation

## Output data validation

Post Processing GC Output Automated Data Validation:

Automated Data Validations for all relevant output KPIs has been implemented. These output data validations are triggered as part of the main ADF Orchestration pipeline. Validations for different KPIs are set to run on different workdays based on the schedule provided.

Development has been done for multiple test cases for all SMOs. Exception reports are created after completion of all the test cases for RegSD, CMFSD, NatSD, TTSD, GC OUTPUT, TDC, FX Rate, CM and PM respectively.

The list of all test cases for Automated Data Validation of outputs can be found below.

|  |  |
| --- | --- |
| No. | Test Case |
| 1 | No Duplicate rows in Output. |
| 2 | All Geos available in output are from Europe |
| 3 | Sum of GIV in output <> 'NULL' and <> '0' |
| 4 | 'There are no 'NULL' Values in 'NOS' (all currency types) and 'GC' (all currency types) |
| 5 | 'There are no 'NULL' Values in 'TDC' (all currency types) |
| 6 | 'There are no 'NULL' Values in Output Master Data (Geo-Customer-Product-Time) columns have values in all rows |
| 7 | 'There are no 'NULL' Values in Output Master Data (Geo-Customer-Product-Time) columns have values in all rows |
| 8 | 'There are no 'NULL' Values in 'Total CMF Calc' (all currency types) |
| 7 | 'There are no 'NULL' Values in 'Total TT' (all currency types) |
| 8 | Relevant KPIs for Geo are available across all months from PFY and CFY until reporting month are available in output |
| 9 | All months from PFY and CFY until reporting month are available in output |
| 10 | No duplicates in Output Master Data (Geo-Customer-Product-Time) columns combination |
| 11 | Correct FX Rate conversion is applied in all relevant KPIs |
| 12 | Subsector Unknown (ID: 1000021140) is not available in Output |
| 13 | Sum of all relevant KPIs after aggregation to monthly granularity available in input are equal input vs output |
| 14 | 'All other' customer exists for Geo-Month combination when Total Shipment Vol from Upstream > Total shipment value for all mapped customers |
| 15 | All L4 Customers From customer mapping with Vol or GIV <> 0 in SHEMEA/RTDC are available in output |
| 16 | Sum of KPI value = '0' on Subsector-Month aggregation in output |
| 17 | All ingested values should be equal in GC output in Azure vs. MMR ingested inputs at Geo - Customer L4 - Category - Month level aggregation |
| 18 | All ingested values should be equal in GC output in Azure vs. MMR ingested input at Geo - Customer L4 - Lowest Possible Product Level - Month level aggregation |
| 19 | All ingested values should be equal in GC Output in Azure vs. MMR ingested input at Geo - Customer L4 (if available) - Category - Month level aggregation |
| 20 | All ingested values should be equal in GC Output in Azure vs. MMR ingested inputs at Geo - Customer L4 - Category - Month level aggregation |
| 21 | No duplicate rows in all Anaplan output |

**Description:**

**Duplicate check**

Test case ID: 130010

Expected Result: No duplicate rows in output.

Implementation: In order to check for duplicate records, a logic is implemented where a join is performed on the final Output table (stage\_europe.eur\_rep\_output\_vw) with itself and the count of number of individual records is checked. If the count is greater than one, then it implies that there are duplicate records present in the output. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation of Geo Output contains correct geo data**

Test case ID: 130022

Expected Result: All Geos available in Output are from Europe.

Implementation: In order to check if the Output contains only Geos specific to Europe, a logic is implemented where all records are selected from the final Output table (stage\_europe.eur\_rep\_output\_vw) where geo\_id is NOT part of Geo Mapping. In case such records exist, it implies that there are Geos present which are not part of Europe. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation if GIV data is available in Output**

Test case ID: 130248

Expected Result: Sum of GIV in Output <>’0’.

Implementation: In order to check if GIV data in Output contains 0 values, a logic is implemented where all records are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) where GIV\_LC=0 or GIV\_R$=0 OR GIV\_C$=0. In case such records exist, it implies that there are records where GIV is 0. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation if NOS and GC columns have no ‘NULL’ values**

Test case ID: 130250

Expected Result: There are no NULL values in NOS (All currency types) and GC (All currency types).

Implementation: In order to check if NOS and GC data in Output contains NULL values, a logic is implemented where all records are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) where NOS\_LC is NULL or NOS\_R$ is NULL or NOS\_C$ is NULL or GC\_LC is NULL or GC\_R$ is NULL or GC\_C$ is NULL. In case such records exist, it implies that there are records where NOS or GC is NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation if TDC Column has no NULL values**

Test case ID: 130251

Expected Result: There are no NULL values in TDC (All currency types).

Implementation: In order to check if TDC data in Output contains NULL values, a logic is implemented where all records are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) where TDC \_LC is NULL or TDC \_R$ is NULL or NOS\_C$ is NULL. In case such records exist, it implies that there are records where TDC is NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation if Output Master Data have no NULL values**

Test case ID: 130252

Expected Result: There are no NULL values in Output Master Data (Geo-Product-Customer-Time) columns have values in all rows.

Implementation: In order to check if Master Data in Output contains NULL values, a logic is implemented where all records are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) where GEO\_ID or SMO or CLUSTER or COUNTRY or CUSTOMER\_TYPE or CUSTOMER\_LEVEL\_1 or CUSTOMER\_LEVEL\_2 or CUSTOMER\_LEVEL\_3 or CUSTOMER\_LEVEL\_4 or SECTOR\_ID or SUBSECTOR\_ID or CATEGORY\_ID or BRAND\_ID or BRAND\_FORM\_ID or MONTH or QUARTER or SEMESTER or FY\_ID is NULL. In case such records exist, it implies that there are records where Master Data is NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation if Total CMF column have no NULL values**

Test case ID: 130253

Expected Result: There are no NULL values in Total\_CMF\_Calc (All currency types).

Implementation: In order to check if Total\_CMF\_Calc data in Output contains NULL values, a logic is implemented where all records are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) where Total\_CMF\_Calc\_LC is NULL or Total\_CMF\_Calc\_R$ is NULL or Total\_CMF\_Calc\_C$ is NULL. In case such records exist, it implies that there are records where TDC is NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation if Total TT column have no NULL values**

Test case ID: 130254

Expected Result: There are no NULL values in ‘Total TT’ (All currency types).

Implementation: In order to check if Total TT data in Output contains NULL values, a logic is implemented where all records are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) where Total\_TT\_LC is NULL or Total\_TT\_R$ is NULL or Total\_TT\_C$ is NULL. In case such records exist, it implies that there are records where Total TT is NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation if all relevant KPIs for specific Geo are present across all reporting months**

Test case ID: 130256

Expected Result: Relevant KPIs for Geo are available across all months from PFY and CFY until reporting month are available in Output.

Implementation: In order to check if all relevant KPIs for any specified Geo are available across all months (PFY and CFY), a logic has been implemented where a join is performed on the final output table (stage\_europe.eur\_rep\_output\_vw) with all the Geo specific MMR Coefficient tables on Month and Customer Type and all records are selected where the respective coefficient KPIs are NULL and count of the KPIs<>’0’. If any such records exist, it implies that there are records which are not available across all months (PFY and CFY). This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation if all reporting months are available**

Test case ID: 130258

Expected Result: All months from CFY and PFY until reporting month are available in Output.

Implementation: In order to check if all months from CFY and PFY until reporting month are available in Output, a logic has been implemented where a join is performed on the final output table (stage\_europe.eur\_rep\_output\_vw) with the month count table (stage\_europe\_adv.cfy\_month\_count) on the FY\_ID and all records are selected where FY\_ID of final output table = FY\_ID of month count table and count(months) in output table <> count(months) in month count table. In case such records exist, it implies that there are records where month is not part of CFY, PFY till reporting month. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation if there are no duplicates in Output Masterdata**

Test case ID: 130259

Expected Result: No duplicates in Output Master Data (Geo-Customer-Product-Time) columns combination.

Implementation: In order to check if Output Master data contains duplicate records, a logic has been implemented where a join is performed with the final output table (stage\_europe.eur\_rep\_output\_vw) with itself and all records are selected where the count is greater than 1. In case such records exist, it implies that duplicate records are present in the Output. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation to check if FX-Rate conversion is applied**

Test case ID: 130280

Expected Result: Correct FX-Rate conversion is applied in all relevant KPIs.

Implementation: In order to check if correct FX-Rate conversion has been applied across all relevant KPIs, a logic has been implemented where a join is performed on the final output table (stage\_europe.eur\_rep\_output\_vw) with the fx-rate table (stage\_europe.eur\_rep\_dir\_crncy\_geoid) and a comparison is performed between the fx-rates coming from the (stage\_europe.eur\_rep\_dir\_crncy\_geoid) and the calculated values (LC/C$ and LC/R$) from (stage\_europe.eur\_rep\_output\_vw). If a difference greater than 0.001 is found between the two, then it implies that fx rate conversion is happening incorrectly. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation is Subsector “Unknown” is excluded from Output**

Test case ID: 130287

Expected Result: Subsector “Unknown” (ID: 1000021140) is not available in output.

Implementation: In order to check if the subsector “Unknown” is present in the Output, a logic has been implemented where all records are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) where SUBSECTOR\_ID=1000021140 or SUBSECTOR\_NAME=’UNKNOWN’. If such records exist, then it implies that there are records where subsector is “UNKNOWN” in the final Output. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation if sum of input KPIs is equal to sum of output KPIs**

Test case ID: 130291

Expected Result: Sum of all relevant KPIs after aggregation to monthly granularity available in input are equal vs the output.

Implementation: In order to check if Input KPIs are matching with Output KPIs, a logic has been implemented where a join is performed on the final output table (stage\_europe.eur\_rep\_output\_vw) with the respective MMR coefficient table on geo-month-customer-product level and all records are selected where the difference of the Output KPI and Input KPIs is greater than 0.01. If such records exist, it implies that the input KPIs and output KPIs have a mismatch. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation to check for availability of “All Other” customer**

Test case ID: 130298

Expected Result: “All Other” customer exists for Geo-Month combination when Total shipments VOL from upstream > Total shipments VOL from mapped customers.

Implementation: In order to check if “All Other” customer exists for all geo-months combination, a logic has been implemented where all records are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) where CUSTOMER\_LEVEL\_4<>’All Other’. If such records exist, it implies that the “All Other” customer is not available for a specified geo-month combination. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validate if customers are available in latest customer mapping input vs customers available in the cngc raw output to verify if data is propagated properly**

Test case ID: 130302

Expected Result: All L4 customers from Customer Mapping with VOL or GIV<>0 in SHEMEA/RTDC are available in output.

Implementation: In order to check if all L4 customers are available in output, a logic has been implemented where a comparison is performed between customers in final Output table and customers from customer mapping. In case records exist, it implies that there are customers which are present in customer mapping but not in final Output. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation of KPIs equals to 0 in the output for validation**

Test case ID: 130303

Expected Result: Sum of KPI value = 0 on subsector-month aggregation in output.

Implementation: In order to check if output KPIs are 0, a logic has been implemented where all relevant KPIs are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) where KPI=0. If such records exist, it implies that there are records in the final Output table where value of relevant KPIs is 0. This is a non-hard stop test case that will not stop further validations.

**Validate if all customers have correct values in relevant breakdowns of the TT input**

Test case ID: 129604

Expected Result: All ingested values should be equal in GC Output in Azure vs MMR ingested inputs at Geo-Customer-Category-Month level aggregation.

Implementation: In order to check if the input TT KPIs are matching with the final Output KPIs, a logic has been implemented where the relevant TT KPIs are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) and they are compared with the data from the input MMR TT table. If the difference between the input KPI and output KPI is greater than 0.01, then it is considered as an exception. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validate if all customers have correct values in CMF input**

Test case ID: 129607

Expected result: All ingested values should be equal in GC Output in Azure vs MMR at Geo-Customer-Product-Month level.

Implementation: In order to check if the input CMF KPIs are matching with the final Output KPIs, a logic has been implemented where the relevant CMF KPIs are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) and they are compared with the data from the input MMR CMF (SD) table. If the difference between the input KPI and output KPI is greater than 0.01, then it is considered as an exception. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validate if all customers have correct values in NatSD input**

Test case ID: 129611

Expected result: All ingested values should be equal in GC Output in Azure vs MMR at Geo-Customer-Product-Month level.

Implementation: In order to check if the input NatSD KPIs are matching with the final Output KPIs, a logic has been implemented where the relevant NatSD KPIs are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) and they are compared with the data from the input MMR NatSD table. If the difference between the input KPI and output KPI is greater than 0.01, then it is considered as an exception. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validate if all customers have correct values in RegSD input**

Test case ID: 129611

Expected result: All ingested values should be equal in GC Output in Azure vs MMR at Geo-Customer-Product-Month level.

Implementation: In order to check if the input RegSD KPIs are matching with the final Output KPIs, a logic has been implemented where the relevant RegSD KPIs are selected from the final output table (stage\_europe.eur\_rep\_output\_vw) and they are compared with the data from the input MMR RegSD table. If the difference between the input KPI and output KPI is greater than 0.01, then it is considered as an exception. This is a hard stop test case and will stop further data validations until such errors are resolved.

## Anaplan Output Automated Data Validation

|  |  |
| --- | --- |
| No. | Test Case |
| 1 | No Duplicate rows in Anaplan Output. |
| 2 | NULL validation in relevant outputs. |
| 3 | ‘0’ values validation in relevant outputs. |
| 4 | Validation if KPIs are within defined range |
| 5 | Validation for KPI availability |
|  |  |

**Description:**

**Duplicates Check**

Test case ID: 130324

Expected Result: Data is available in output as per relevant inputs.

Implementation: In order to check for duplicate records, a logic is implemented where all records are selected where the count of the individual record is greater than 1 from each Anaplan Coefficient Output table (stage\_europe.eur\_ana\_fra\_SD\_TT\_C1, stage\_europe.eur\_ana\_gbr\_SD\_TT\_C1, stage\_europe.eur\_ana\_gbr\_SD\_TT\_C2,

stage\_europe.eur\_ana\_irl\_SD\_TT\_C1,

stage\_europe.eur\_ana\_irl\_SD\_TT\_C2). This implies that there are records in the Anaplan Coefficient Outputs are duplicated. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation of NULL values in pre-defined columns in pre-defined Outputs**

Test case ID: 130329

Expected Result: All rows have values in relevant columns.

Implementation: In order to check if all relevant KPIs of Anaplan Outputs have data within them, a logic has been implemented where all records are selected from the Anaplan output tables (stage\_europe.eur\_ana\_fra\_SD\_TT\_C1, stage\_europe.eur\_ana\_gbr\_SD\_TT\_C1, stage\_europe.eur\_ana\_gbr\_SD\_TT\_C2,

stage\_europe.eur\_ana\_irl\_SD\_TT\_C1,

stage\_europe.eur\_ana\_irl\_SD\_TT\_C2) where the relevant KPIs are NULL. In case any such records exist, it implies that there are records where the value for one or more KPIs is NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation of ‘0’ values in pre-defined columns in pre-defined outputs**

Test case ID: 130331

Expected Result: All rows are not equal to 0 in all relevant columns.

Implementation: In order to check if all relevant KPIs of Anaplan Outputs do not have ‘0’ data within them, a logic has been implemented where all records are selected from the Anaplan output tables (stage\_europe.eur\_ana\_fra\_SD\_TT\_C1, stage\_europe.eur\_ana\_gbr\_SD\_TT\_C1, stage\_europe.eur\_ana\_gbr\_SD\_TT\_C2,

stage\_europe.eur\_ana\_irl\_SD\_TT\_C1,

stage\_europe.eur\_ana\_irl\_SD\_TT\_C2) where the relevant KPIs are ‘0’. In case any such records exist, it implies that there are records where the value for one or more KPIs is ‘0’. This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation for Value Range Check**

Test case ID: 130337

Expected Result: Validation of data in Trade\_Terms\_On\_Invoice\_%, Cash\_Discount\_OI\_%, Cash\_Discount\_NOI\_% columns within defined range.

Implementation: In order to check if the required columns lie within the defined range, a logic has been implemented where all records are selected from the Anaplan Output tables (stage\_europe.eur\_ana\_fra\_SD\_TT\_C1, stage\_europe.eur\_ana\_gbr\_SD\_TT\_C1, stage\_europe.eur\_ana\_gbr\_SD\_TT\_C2,

stage\_europe.eur\_ana\_irl\_SD\_TT\_C1,

stage\_europe.eur\_ana\_irl\_SD\_TT\_C2) where Trade\_Terms\_On\_Invoice\_% is <-5 or >11.5,

Cash\_Discount\_OI\_% is not in (0,1,2),

Cash\_Discount\_NOI\_% is not in (0,1,2). If such records exist, it implies that there are some records where the relevant KPIs are out of range.

This is a hard stop test case and will stop further data validations until such errors are resolved.

**Validation for KPI availability**

Test case ID: 130347

Expected Result: Data is available in output as per relevant inputs.

Implementation: In order to check if all relevant data is present in Anaplan Output tables, a logic has been implemented where all records are selected from the Anaplan Output tables (stage\_europe.eur\_ana\_fra\_SD\_TT\_C1, stage\_europe.eur\_ana\_gbr\_SD\_TT\_C1, stage\_europe.eur\_ana\_gbr\_SD\_TT\_C2,

stage\_europe.eur\_ana\_irl\_SD\_TT\_C1,

stage\_europe.eur\_ana\_irl\_SD\_TT\_C2) where the total number of months for each record is not equal to 12. If such records exist, it implies that there are records present which are not available throughout every month of the Fiscal Year. This is a hard stop test case and will stop further data validations until such errors are resolved.

## Input Automated Data Validation

|  |  |
| --- | --- |
| No. | Test Case |
| 1 | MMR-RegSD Data Completeness |
| 2 | MMR-RegSD Duplicate check |
| 3 | MMR-RegSD DQ Data Availability Report |
| 4 | MMR-NatSD Data Completeness |
| 5 | MMR-NatSD Duplicate check |
| 6 | MMR-NatSD DQ Data Availability Report |
| 7 | MMR-CMF Data Completeness |
| 8 | MMR-CMF Duplicate check |
| 9 | MMR-CMF DQ Data Availability Report |
| 10 | MMR-CMF DQ ‘All Other’ calculation report |
| 11 | MMR-TT Data Completeness |
| 12 | MMR-TT Duplicate check |
| 13 | MMR-TT DQ Data Availability Report |
| 14 | MMR-TT DQ ‘All Other’ calculation report |
| 15 | MMR-FX Data Completeness |
| 16 | MMR-FX Duplicates |
| 17 | MMR-FX Key column duplicates |
| 18 | MMR-TDC Data Completeness |
| 19 | MMR-TDC Duplicates |
| 20 | MMR-TDC Key column duplicates |
| 21 | MMR-PM Data Completeness |
| 22 | MMR-PM Duplicates |
| 23 | MMR-PM Product Hierarchy |
| 24 | MMR-PM Valid Product ID |
| 25 | MMR-PM DQ Input Delta Report |
| 26 | MMR-CM Data Completeness |
| 27 | MMR-CM Duplicates |
| 28 | MMR-CM Geo Hierarchy |
| 29 | MMR-CM L4 to Geo Mapping |
| 30 | MMR-CM L4 to L3 Mapping |
| 31 | MMR-CM L4 to L2 Mapping |
| 32 | MMR-CM L4 to L1 Mapping |
| 33 | MMR-CM Optima to L4 Mapping |
| 34 | MMR-CM ShipID to L$ Mapping |
| 35 | MMR-CM Key columns duplication |
| 36 | MMR-CM Geo ID to Europe Mapping |
|  |  |

**Description:**

**MMR-RegSD Data Completeness**

Test case ID: 129612

Expected Result: No NULL values in ingested data.

Implementation: In order to check if Input data consists of NULL values, a logic has been implemented where all records are selected from the Input RegSD table (stage\_mmr.MNR\_SD\_REGNL\_DIRCT\_STD\_DIM\_VW) where the relevant KPIs (Geo-Category-Customer-Month-REGSD\_VAL) are NULL. In case such records exist, it implies that there are some records where the relevant Input KPIs are NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-RegSD Duplicate Check**

Test case ID: 129613

Expected Result: No duplicate rows in ingested data.

Implementation: In order to check if ingested data has duplicate records, a logic has been implemented where all records are selected from the input RegSD table (stage\_mmr.MNR\_SD\_REGNL\_DIRCT\_STD\_DIM\_VW) where the count of the individual record is greater than 1. In case such records exist, it implies that the ingested data consists of duplicated records. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-RegSd DQ Data Availability Report**

Test case ID: 130362

Expected Result: Data is available (<>0) for all Subsector, reporting months for relevant breakdowns on geo level.

Implementation: In order to check for Data availability, a logic has been implemented where all records are selected from the input RegSD table (stage\_mmr.MNR\_SD\_REGNL\_DIRCT\_STD\_DIM\_VW) at Geo-Subsector-Month level where REGNL\_SD\_VAL=0. In case such records exist, it implies that the ingested data consists of records where relevant KPIs =0. This is a non-hard stop test case and will not stop further processing of data.

**MMR-NatSD Data Completeness**

Test case ID: 129608

Expected Result: No NULL values in ingested data.

Implementation: In order to check if Input data consists of NULL values, a logic has been implemented where all records are selected from the Input NatSD table (stage\_mmr.mnr\_sd\_natl\_dirct\_nstd\_dim\_vw) where the relevant KPIs (Geo-Category-Customer-Month-NATSD\_VAL) are NULL. In case such records exist, it implies that there are some records where the relevant Input KPIs are NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-NatSD Duplicate Check**

Test case ID: 129610

Expected Result: No duplicate rows in ingested data.

Implementation: In order to check if ingested data has duplicate records, a logic has been implemented where all records are selected from the input NatSD table (stage\_mmr.mnr\_sd\_natl\_dirct\_nstd\_dim\_vw) where the count of the individual record is greater than 1. In case such records exist, it implies that the ingested data consists of duplicated records. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-NatSd DQ Data Availability Report**

Test case ID: 130361

Expected Result: Data is available (<>0) for all Subsector, reporting months for relevant breakdowns on geo level.

Implementation: In order to check for Data availability, a logic has been implemented where all records are selected from the input NatSD table (stage\_mmr.mnr\_sd\_natl\_dirct\_nstd\_dim\_vw) at Geo-Subsector-Month level where NATSD\_VAL=0. In case such records exist, it implies that the ingested data consists of records where relevant KPIs =0. This is a non-hard stop test case and will not stop further processing of data.

**MMR-CMF Data Completeness**

Test case ID: 129605

Expected Result: No NULL values in ingested data.

Implementation: In order to check if Input data consists of NULL values, a logic has been implemented where all records are selected from the Input CMF table (stage\_mmr.mnr\_sd\_dirct\_nstd\_dim\_vw) where the relevant KPIs (Geo-Category-Customer-Month-NO\_BRKDW\_VAL) are NULL. In case such records exist, it implies that there are some records where the relevant Input KPIs are NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-CMF Duplicate Check**

Test case ID: 129606

Expected Result: No duplicate rows in ingested data.

Implementation: In order to check if ingested data has duplicate records, a logic has been implemented where all records are selected from the input CMF table (stage\_mmr.mnr\_sd\_dirct\_nstd\_dim\_vw) where the count of the individual record is greater than 1. In case such records exist, it implies that the ingested data consists of duplicated records. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-CMF DQ Data Availability Report**

Test case ID: 130359

Expected Result: Data is available (<>0) for all Subsector, reporting months for relevant breakdowns on geo level.

Implementation: In order to check for Data availability, a logic has been implemented where all records are selected from the input CMF table (stage\_mmr.mnr\_sd\_dirct\_nstd\_dim\_vw) at Geo-Subsector-Month level where NO\_BRKDW\_VAL=0. In case such records exist, it implies that the ingested data consists of records where relevant KPIs =0. This is a non-hard stop test case and will not stop further processing of data.

**MMR-CMF DQ ‘All Other’ Calculation Report**

Test case ID: 130360

Expected Result: Sum of Inputs for all mapped L4 customers for all relevant breakdowns < value for TOTAL customer for lowest possible product granularity.

Implementation: In order to check for correct calculation of ‘All Other’ customer, a logic has been implemented where a union is applied to the input CMF table (stage\_mmr.mnr\_sd\_dirct\_nstd\_dim\_vw) with itself. Sum of all input KPIs are selected in the first part of the union query where customer=’TOTAL’. In the second part, the sum of all input KPIs \* -1 is selected where customer<>’TOTAL’. Therefore, by applying the union, the difference is calculated and stored in the ‘All Other’ customer. Records that have values in relevant KPIs for ‘All Other’ > 0 are considered as exceptions. This is a non-hard stop test case and will not stop further processing of data.

**MMR-TT Data Completeness**

Test case ID: 129602

Expected Result: No NULL values in ingested data.

Implementation: In order to check if Input data consists of NULL values, a logic has been implemented where all records are selected from the Input TT table (stage\_mmr.mnr\_tt\_dirct\_nstd\_dim\_vw) where the relevant KPIs (Geo-Category-Customer-Month-NO\_BRKDW\_VAL) are NULL. In case such records exist, it implies that there are some records where the relevant Input KPIs are NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-TT Duplicate Check**

Test case ID: 129603

Expected Result: No duplicate rows in ingested data.

Implementation: In order to check if ingested data has duplicate records, a logic has been implemented where all records are selected from the input TT table (stage\_mmr.mnr\_tt\_dirct\_nstd\_dim\_vw) where the count of the individual record is greater than 1. In case such records exist, it implies that the ingested data consists of duplicated records. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-TT DQ Data Availability Report**

Test case ID: 130353

Expected Result: Data is available (<>0) for all Subsector, reporting months for relevant breakdowns on geo level.

Implementation: In order to check for Data availability, a logic has been implemented where all records are selected from the input TT table (stage\_mmr.mnr\_tt\_dirct\_nstd\_dim\_vw) at Geo-Subsector-Month level where NO\_BRKDW\_VAL=0. In case such records exist, it implies that the ingested data consists of records where relevant KPIs =0. This is a non-hard stop test case and will not stop further processing of data.

**MMR-TT DQ ‘All Other’ Calculation Report**

Test case ID: 130354

Expected Result: Sum of Inputs for all mapped L4 customers for all relevant breakdowns < value for TOTAL customer for lowest possible product granularity.

Implementation: In order to check for correct calculation of ‘All Other’ customer, a logic has been implemented where a union is applied to the input TT table (stage\_mmr.mnr\_tt\_dirct\_nstd\_dim\_vw) with itself. Sum of all input KPIs are selected in the first part of the union query where customer=’TOTAL’. In the second part, the sum of all input KPIs \* -1 is selected where customer<>’TOTAL’. Therefore, by applying the union, the difference is calculated and stored in the ‘All Other’ customer. Records that have values in relevant KPIs for ‘All Other’ > 0 are considered as exceptions. This is a non-hard stop test case and will not stop further processing of data.

**MMR-FX Data Completeness**

Test case ID: 129599

Expected Result: No NULL values in ingested data.

Implementation: In order to check if Input data consists of NULL values, a logic has been implemented where all records are selected from the Input FX Rate table (stage\_mmr.MNR\_FRGN\_EXCHG\_RATE\_DIM\_VW) where the relevant KPIs are NULL. In case such records exist, it implies that there are some records where the relevant Input KPIs are NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-FX Duplicates check**

Test case ID: 129600

Expected Result: No duplicate rows in ingested data.

Implementation: In order to check if ingested data has duplicate records, a logic has been implemented where all records are selected from the input FX Rate table (stage\_mmr.MNR\_FRGN\_EXCHG\_RATE\_DIM\_VW) where the count of the individual record is greater than 1. In case such records exist, it implies that the ingested data consists of duplicated records. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-FX Key Column Duplicates**

Test case ID: 129601

Expected Result: No duplicates for currency/target currency/month column combinations.

Implementation: In order to check if relevant KPIs have duplicate records, a logic has been implemented where relevant KPIs are selected from the input FX Rate table (stage\_mmr.MNR\_FRGN\_EXCHG\_RATE\_DIM\_VW) where the count of the individual record is greater than 1. In case such records exist, it implies that the relevant KPIs consist of duplicated records. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-TDC Data Completeness**

Test case ID: 129596

Expected Result: No NULL values in ingested data.

Implementation: In order to check if Input data consists of NULL values, a logic has been implemented where all records are selected from the Input TDC table (stage\_mmr.mnr\_tdc\_su\_dim\_vw) where the relevant KPIs are NULL. In case such records exist, it implies that there are some records where the relevant Input KPIs are NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-TDC Duplicates check**

Test case ID: 129597

Expected Result: No duplicate rows in ingested data.

Implementation: In order to check if ingested data has duplicate records, a logic has been implemented where all records are selected from the input TDC table (stage\_mmr.mnr\_tdc\_su\_dim\_vw) where the count of the individual record is greater than 1. In case such records exist, it implies that the ingested data consists of duplicated records. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-TDC Key Column Duplicates**

Test case ID: 129598

Expected Result: No duplicates for currency/target currency/month column combinations.

Implementation: In order to check if relevant KPIs have duplicate records, a logic has been implemented where relevant KPIs are selected from the input TDC table (stage\_mmr.mnr\_tdc\_su\_dim\_vw) where the count of the individual record is greater than 1. In case such records exist, it implies that the relevant KPIs consist of duplicated records. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-PM Data Completeness**

Test case ID: 129581

Expected Result: No NULL values in ingested values.

Implementation: In order to check if Input data consists of NULL values, a logic has been implemented where all records are selected from the Input PM table (stage\_mmr.MNR\_PROD\_MAPNG\_DIM\_VW) where the relevant KPIs are NULL. In case such records exist, it implies that there are some records where the relevant Input KPIs are NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-PM Duplicate check**

Test case ID: 129582

Expected Result: No duplicate rows in ingested data.

Implementation: In order to check if ingested data has duplicate records, a logic has been implemented where all records are selected from the input PM table (stage\_mmr.MNR\_PROD\_MAPNG\_DIM\_VW) where the count of the individual record is greater than 1. In case such records exist, it implies that the ingested data consists of duplicated records. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-PM Product Hierarchy**

Test case ID: 129583

Expected Result: One to many (higher to lower level) relationships for all product dimensions across all available levels.

Implementation: In order to check for one to many relationships in PM, a logic has been implemented where all records are selected from the Product Mapping Table (stage\_mmr.MNR\_PROD\_MAPNG\_DIM\_VW) where Brand ID is selected from all records where count of Brand ID is greater than 1. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-PM Valid Product ID**

Test case ID: 129586

Expected Result: All ingested Product IDs should be a valid RDS Product ID.

Implementation: In order to check if all ingested values are also valid RDS Product IDs, a logic has been implemented where a join is performed between the Product mapping table (stage\_mmr.MNR\_PROD\_MAPNG\_DIM\_VW) and the Product Hierarchy table from RDS (stage\_common.sode\_p5005\_product\_dim) and all records are selected where Brand ID from Product mapping table is not a part of the Product ID from RDS table. If such records exist, then they are considered exceptions. This is a hard stop test case and will stop further validations until the issue is resolved.

**MMR-PM DQ Input Delta Report**

Test case ID: 130349

Expected Result: Report with visible changes current submission vs last submission.

Implementation: In order to check if visible changes between current submission and last submission have been applied, a logic has been implemented where all records are selected from the Product mapping table (stage\_mmr.MNR\_PROD\_MAPNG\_DIM\_VW) which were not present in the previous extract of Product mapping. In case such records exist, it implies that new data has been included in the current Product Mapping data. This is a non-hard stop test case that will not stop further validations and processing.

**MMR-CM Data Completeness**

Test case ID: 129275

Expected Result: No NULL values in ingested data.

Implementation: In order to check if Input data consists of NULL values, a logic has been implemented where all records are selected from the Input CM table (stage\_mmr.MNR\_CUST\_MAPNG\_DIM\_VW) where the relevant KPIs are NULL. In case such records exist, it implies that there are some records where the relevant Input KPIs are NULL. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-CM Duplicate Check**

Test case ID: 129559

Expected Result: No duplicate records in ingested values.

Implementation: In order to check if ingested data has duplicate records, a logic has been implemented where all records are selected from the input CM table (stage\_mmr.MNR\_CUST\_MAPNG\_DIM\_VW) where the count of the individual record is greater than 1. In case such records exist, it implies that the ingested data consists of duplicated records. This is a hard stop test case and will stop further data validations until such errors are resolved.

**MMR-CM Geo Hierarchy**

Test case ID: 129560

Expected Result: No many to one relationship in ingested geo hierarchy.

Implementation: In order to check for many to one relationship in ingested data, a logic has been implemented where all records are selected from the CM table (stage\_mmr.MNR\_CUST\_MAPNG\_DIM\_VW) where the count of the Geo\_ID is greater than 1. This is a hard stop test case and will stop further processing of data until it is resolved.

**MMR-CM L4 to Geo Mapping**

Test case ID: 129562

Expected Result: No many to many relationships in ingested customer L4 to Geo.

Implementation: In order to check for many to many relationships, a logic has been implemented where records are selected from the customer mapping table (stage\_mmr.MNR\_CUST\_MAPNG\_DIM\_VW) where the count of records at customer L4-L3-L2-L1-Geo level is greater than 1. If such records exist, it implies that there are records in the ingested data where many to many relationships exist. This is a hard stop test case and will stop further processing of data until it is resolved.

**MMR-CM L4 to L3 Mapping**

Test case ID: 129564

Expected Result: One L4 customer cannot have many L3 customer nodes in hierarchy.

Implementation: In order to check if one customer\_L4 has many L3 customers, a logic has been implemented where records are selected from the customer mapping table (stage\_mmr.MNR\_CUST\_MAPNG\_DIM\_VW) where the count of records at customer L4-Geo-customer L3 level is greater than 1. If such records exist, it implies that there are records in the ingested data where one customer L4 has more than 1 L3 customer in the hierarchy. This is a hard stop test case and will stop further processing of data until it is resolved.

**MMR-CM L4 to L2 Mapping**

Test case ID: 129566

Expected Result: One L4 customer cannot have many L2 customer nodes in hierarchy.

Implementation: In order to check if one customer\_L4 has many L2 customers, a logic has been implemented where records are selected from the customer mapping table (stage\_mmr.MNR\_CUST\_MAPNG\_DIM\_VW) where the count of records at customer L4-Geo-customer L2 level is greater than 1. If such records exist, it implies that there are records in the ingested data where one customer L4 has more than 1 L2 customer in the hierarchy. This is a hard stop test case and will stop further processing of data until it is resolved.

**MMR-CM L4 to L1 Mapping**

Test case ID: 129567

Expected Result: One L4 customer cannot have many L1 customer nodes in hierarchy.

Implementation: In order to check if one customer\_L4 has many L1 customers, a logic has been implemented where records are selected from the customer mapping table (stage\_mmr.MNR\_CUST\_MAPNG\_DIM\_VW) where the count of records at customer L4-Geo-customer L1 level is greater than 1. If such records exist, it implies that there are records in the ingested data where one customer L4 has more than 1 L1 customer in the hierarchy. This is a hard stop test case and will stop further processing of data until it is resolved.

**MMR-CM Optima to L4 Mapping**

Test case ID: 129569

Expected Result: One optima ID cannot be mapped with different customer L4s in customer hierarchy. Exception 1: France – Different L4 customers with same Org ID cannot be mapped with same Optima ID. Exception 2: Indirect markets (Turkey, CCAR, SEE, Ukraine, EE) – Different L4 customers with same shipment type cannot be mapped with same Optima ID.

Implementation: In order to check for correct Optima to L4 mapping, a logic has been implemented where all records are selected from the CM table (stage\_mmr.MNR\_CUST\_MAPNG\_DIM\_VW) where the count of Optima ID at Customer\_L4 level is greater than 1 and SMOs are NOT in Turkey, France, CCAR and Ukraine. If such records exist, they are considered exceptions. This is a hard stop test case and will stop further processing of data until it is resolved.

**MMR-CM ShipID to L4 Mapping**

Test case ID: 129573

Expected Result: One Shipment ID cannot be mapped with different L4 customers in hierarchy. Exception in France where Customer\_L4 – Shipment\_ID – Org\_ID combination must be unique.

Implementation: In order to check for correct ShipID to L4 mapping, a logic has been implemented where all records are selected from the CM table (stage\_mmr.MNR\_CUST\_MAPNG\_DIM\_VW) where the count of records at Customer\_ID – Org\_ID – Geo\_ID is greater than 1. If such recods exist, they are considered as exceptions. This is a hard stop test case and will stop further processing of data until it is resolved.

**MMR-CM Key columns duplications**

Test case ID: 129576

Expected result: No duplicates in Country – Customer\_L4 – Customer\_L3 – Customer\_L2 – Customer\_L1 – Shipment\_ID – Optima\_ID – Org\_ID (France only) column combination values.

Implementation: In order to check for key column duplicates, a logic has been implemented where a join has been performed on the CM table (stage\_mmr.MNR\_CUST\_MAPNG\_DIM\_VW) with itself at Country – Customer\_L4 – Customer\_L3 – Customer\_L2 – Customer\_L1 – Shipment\_ID – Optima\_ID – Org\_ID level and all records are selected where the count of records is greater than 1. This is a hard stop test case and will stop further processing of data until it is resolved.

**MMR-CM Geo\_ID to Europe Mapping**

Test case ID: 129579

Expected Result: All Geo\_IDs should come from Europe.

Implementation: In order to check if all Geos are coming from Europe, a logic has been implemented where all records are selected from the CM table (stage\_mmr.MNR\_CUST\_MAPNG\_DIM\_VW) where Geo\_ID is not part of the Geo Hierarchy table (stage\_common.SODE\_G707\_FDIM). If such records exist, it implies that the ingested data consists of Geos that are not from Europe. This is a hard stop test case and will stop further processing of data until it is resolved.

**AD-HOC ingestion of RCM (Regional Customer mapping)**

* Whenever there is requirement to ingest RCM file on Ad-hoc basis please trigger the below mentioned pipeline.

In folder 63\_EU\_Orchestration pipeline named ‘cngc\_eu\_regional\_customer\_mapping\_ingestion’

## Data Dependency Flow

Please find below attached data flow/dependency for individual SMO level in each sheet.



## Output Export

* Please find the output export details for individual SMO level below.





1. **Data Visualization**

After processing the data and generating outputs, the validation test cases are represented using Microsoft Power BI in the form of report.

Following are the components of the report :

**Column Chart:** A column chart is used to depict the daily execution status.

**Pie Chart:** A pie chart is used to depict the overall status of the validation test cases.

**Filters:** Two filters are used to filter out the data on basis of month and year.

**Table:** A table is used to represent the validation test cases for each SMO and sources along with the run counts and results. The result is represented as: -

Success: - It represents the completion of the runs successfully without any failure.

Fail: - It represents the failed status of the validation test cases.

Resolved: - It implies the successful re-run of the validation test cases after a failure has been observed and resolved.

Count: - It represents the no of runs for a particular validation.

The components of this table can be drilled down to see detailed information of a particular run.

**Cards:-** Cards are used to represent the overall count of test cases. There are 6 cards which are :

Total Case: This card represents the total no of cases.

DAX query used :

Executed: This card represents the total no cases executed till a particular date of a month.

Not run: This card represents the test cases yet to be executed for the monthly cycle.

Success: This card represents the total no of test cases which resulted in success.

Fail: This card represents the total no of failed test cases.

Resolved: This card represents the total no of test cases that were failed at first run but after fixation resulted in success.

The data is sourced from SQL server “cngc-sqlserver.database.windows.net” from table ‘[cngc\_operations].[V\_Data\_Quality\_Master\_Transaction]’ and table ‘[cngc\_operations].[V\_Data\_Quality\_Master\_Transaction\_Final]’

Graphical user interface, application

Description automatically generated