

Mecano - Talend Detail

- MECANO
 - Site
 - Main Data Flow
 - A. F_MAIN_MECANO
 - A.1 F100_MEC_EXTRACT_BW_QUERY_TO_GSC
 - A.2 J201_STG_TO_ODS
 - A.3 F300_MEC_LOAD_ALL_DIM_TABLE_PART_1
 - A.4 F300_MEC_LOAD_ALL_DIM_TABLE_PART_2
 - A.5 F310_MEC_LOAD_FACT_MAINTENANCE_STRATEGIC
 - A.6 F320_MEC_LOAD_FACT_MAINTENANCE_EFFECTIVENESS
 - A.7 F330_MEC_LOAD_FACT_MAINTENANCE EFFICIENCY
 - A. Additional Development on project 13129
 - A1. GTBU (not yet available)
 - A2. Notification Planning
 - B Loading to Data Ocean Industrial
 - B.1 F002_BWH_MPR_PM05_003_to_ODS (not yet available)
 - B.2 F010_FACT_notification_planning_to DM
 - C. F_500_REFRESH_MAINT_DASH_VIEW
 - D. F100_REFRESH_ACESS_RIGHTS
 - E. Refresh Tableau
 - Loading Data
 - A. MECANO (Talend project MECANO but using Data Ocean Industrial)
 - B. Data Ocean Industrial (Talend project DATA_OCEAN_DOMAIN_INDUSTRIAL)
 - B.1 F002_BWH_MPR_PM05_003_to_ODS
 - B.2 F010_FACT_notification_planning_to DM
 - Example Case
- Logging

Talend project = MECANO

GCP Project : prj-data-dm-industrial-[env]

GCP Product: prj-data-maintenance-dash-[env]

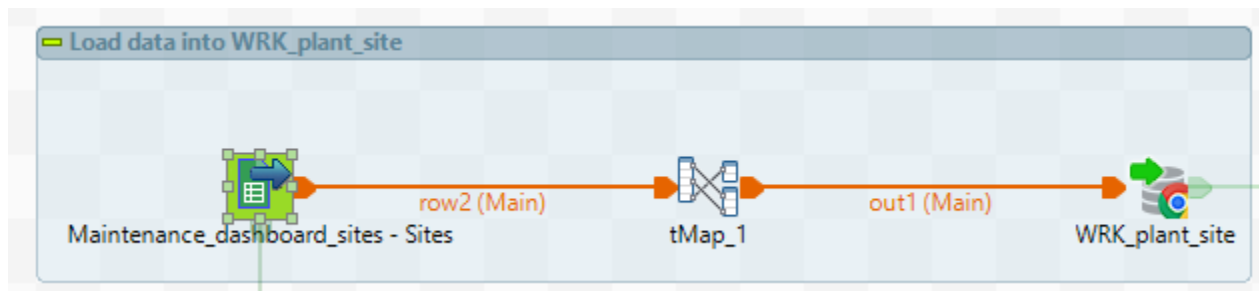
Bucket = cs-ew1-prj-data-dm-industrial-[env]-staging/MECANO/

MECANO

Site

To assign the right site codes and filter the data into final fact table

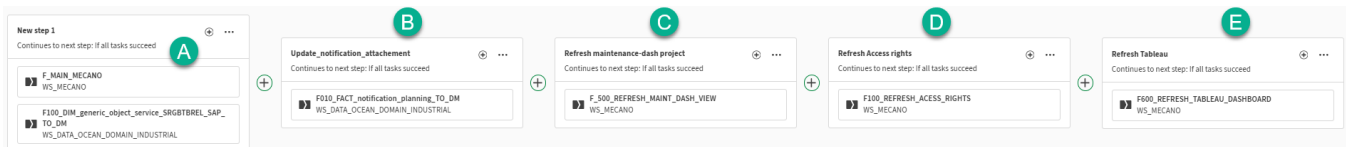
Talend job F001_MEC_LOAD_MAINTENANCE_DASH_SITE



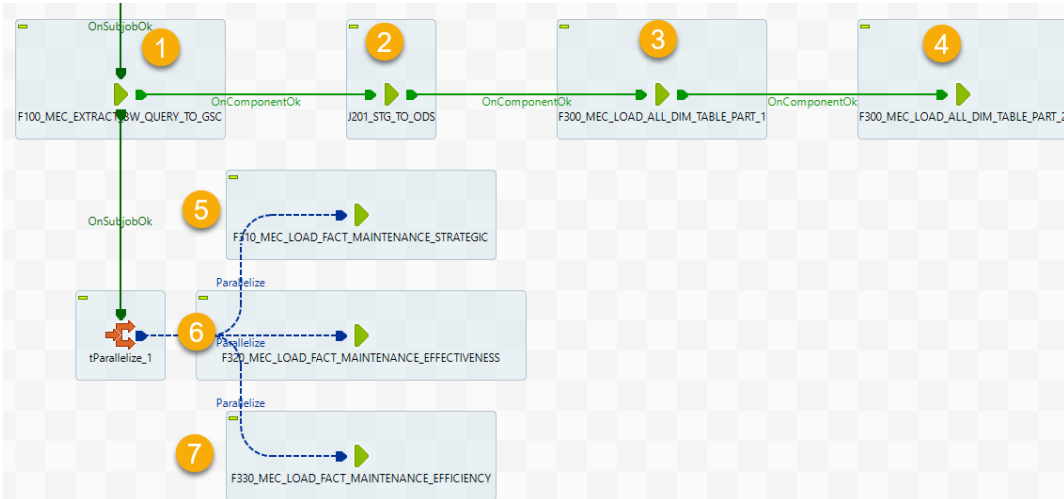
Load from this gsheets (sheet name = "SITE") to prj-data-maintenance-dash-[env].WDL.WRK_plant_site. There is no STG/ODS table

It is full load with truncate table and it is schedule by PL_MECANO_REFRESH_SITES

Main Data Flow

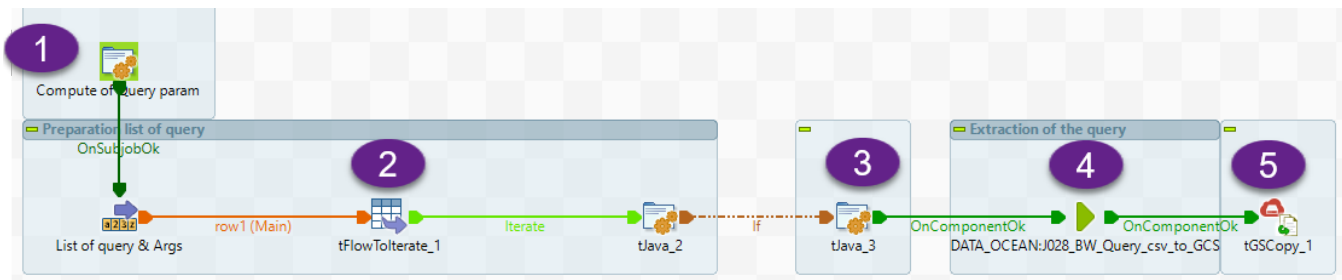


A. F_MAIN_MECANO



A.1 F100_MEC_EXTRACT_BW_QUERY_TO_GSC

To extract data from Xtract to Bucket



1. Compute of query parameter.

To check it is incremental or reload the data by using `I_LOCAL_VAR_RECOVERY_MODE` (true = reload, false = incremental). More detail check on reload session

There are 3 main parameter groups for BW queries.

There are 3 groups of loading variables

Day(yyyyMMdd,yyyyMMdd) : QVMECANOBW_QRY_MVPMOR01_0002

- `I_LOCAL_VAR_MVPMOR01_YYYYMMDD_FROM` eg. 20241001
- `I_LOCAL_VAR_MVPMOR01_YYYYMMDD_TO` ***Must NOT over current Monday

Week (yyyymmww) : The rest of queries

- `I_LOCAL_VAR_HISTO_YEAR` eg. 2024
- `I_LOCAL_VAR_HISTO_WEEK_END` eg. 01

Month(yyyy0MM,yyyy0MM) : QVMECANO_BW_QRY_MVPMCO01_0001

- `I_LOCAL_VAR_MVPMCO01_0001_HIST_K4YYYY0MM_PREV` eg. K42024009
- `I_LOCAL_VAR_MVPMCO01_0001_HIST_K4YYYY0MM_CURR` eg. K42024010

If the `I_LOCAL_VAR_RECOVERY_MODE` = false (normal case that load weekly on Monday)

Day : Previous Monday to current Monday

Week : Last week (last Monday to Sunday)

Month : Load last month to current month

2. List of query to extract

List of query Args(tFixedFlowInput_1)

Basic settings

Advanced settings

Dynamic settings

View

Documentation

Validation Rules

Schema

Built-In

Edit schema

Number of rows

1

Mode

Use Single Table

Use Inline Table

Inline Table

L_var_xtract_job	L_var_file	valid	stg_filename
context.I_VAR_mecano_xtract_DBPMMD01_0001...	context.I_VAR_mecano_xtract_DBPMMD01_0001...	context.I_VAR_mecano_xtract_DBPMMD01_000...	"MEC_IT_0000_0000_F001_" + context.Business_d...
context.I_VAR_mecano_xtract_DBPMMD02_0001...	context.I_VAR_mecano_xtract_DBPMMD02_0001...	context.I_VAR_mecano_xtract_DBPMMD02_000...	"MEC_IT_0000_0000_F002_" + context.Business_d...
context.I_VAR_mecano_xtract_DBPMMD05_0001...	context.I_VAR_mecano_xtract_DBPMMD05_0001...	context.I_VAR_mecano_xtract_DBPMMD05_000...	"MEC_IT_0000_0000_F003_" + context.Business_d...
context.I_VAR_mecano_xtract_mvpmco01_job+...	context.I_VAR_mecano_xtract_mvpmco01_job+...	context.I_VAR_mecano_xtract_mvpmco01_valid	"MEC_IT_0000_0000_F016_" + context.Business_d...
context.I_VAR_mecano_xtract_MVPMNO04_000...	context.I_VAR_mecano_xtract_MVPMNO04_000...	context.I_VAR_mecano_xtract_MVPMNO04_000...	"MEC_IT_0000_0000_F004_" + context.Business_d...
context.I_VAR_mecano_xtract_MVPMNO04_000...	context.I_VAR_mecano_xtract_MVPMNO04_000...	context.I_VAR_mecano_xtract_MVPMNO04_000...	"MEC_IT_0000_0000_F005_" + context.Business_d...
context.I_VAR_mecano_xtract_MVPMCL01_0001...	context.I_VAR_mecano_xtract_MVPMCL01_0001...	context.I_VAR_mecano_xtract_MVPMCL01_0001...	"MEC_IT_0000_0000_F006_" + context.Business_d...
context.I_VAR_mecano_xtract_MVPMOP02_0010...	context.I_VAR_mecano_xtract_MVPMOP02_0010...	context.I_VAR_mecano_xtract_MVPMOP02_001...	"MEC_IT_0000_0000_F015_" + context.Business_d...
context.I_VAR_mecano_xtract_MVPMOP04_0001...	context.I_VAR_mecano_xtract_MVPMOP04_0001...	context.I_VAR_mecano_xtract_MVPMOP04_000...	"MEC_IT_0000_0000_F007_" + context.Business_d...
context.I_VAR_mecano_xtract_MVPMOP04_0002...	context.I_VAR_mecano_xtract_MVPMOP04_0002...	context.I_VAR_mecano_xtract_MVPMOP04_000...	"MEC_IT_0000_0000_F008_" + context.Business_d...
context.I_VAR_mecano_xtract_MVPMOP04_0003...	context.I_VAR_mecano_xtract_MVPMOP04_0003...	context.I_VAR_mecano_xtract_MVPMOP04_000...	"MEC_IT_0000_0000_F009_" + context.Business_d...
context.I_VAR_mecano_xtract_MVPMOP04_0005...	context.I_VAR_mecano_xtract_MVPMOP04_0005...	context.I_VAR_mecano_xtract_MVPMOP04_000...	"MEC_IT_0000_0000_F011_" + context.Business_d...
context.I_VAR_mecano_xtract_MVPMOP04_0006...	context.I_VAR_mecano_xtract_MVPMOP04_0006...	context.I_VAR_mecano_xtract_MVPMOP04_000...	"MEC_IT_0000_0000_F012_" + context.Business_d...
context.I_VAR_mecano_xtract_MVPMOR01_0002...	context.I_VAR_mecano_xtract_MVPMOR01_0002...	context.I_VAR_mecano_xtract_MVPMOR01_000...	"MEC_IT_0000_0000_F014_" + context.Business_d...
context.I_VAR_mecano_xtract_MVPMOR04_0001...	context.I_VAR_mecano_xtract_MVPMOR04_0001...	context.I_VAR_mecano_xtract_MVPMOR04_000...	"MEC_IT_0000_0000_F013_" + context.Business_d...

It control by set of parameter

Xtract Job = I_VAR_mecano_xtract_[query name]_job = Xtract job name

Valid = I_VAR_mecano_xtract_[query name]_valid. If it is 1, it will extract the data (check condition on "if" line)

tJava2 does not have anything

3. tJava3 just print what is the "Current Query" in order to see on the log in TMC

4. Use reference job to call BW query and save the file to GCS

5. Move from staging bucket to folder MECANO

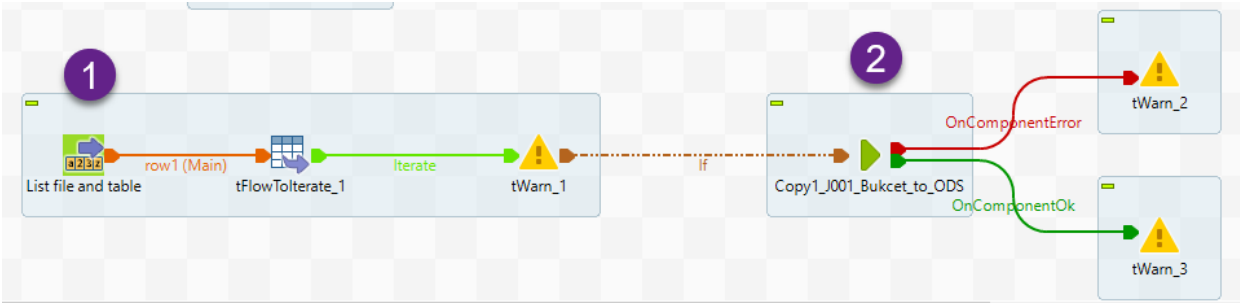
i

Xtract information

The separator in Xtract must use "%" since the data inside BW has all the characters (\$,|,% and etc) and it will split data incorrectly. It is required to use 2 chars

A.2 J201_STG_TO_ODS

To load the file from GCS to ODS



1. Control filename, STG, ODS table name, valid and separator

List file and table(tFixedFlowInput_1)

Schema: Built-In Edit schema

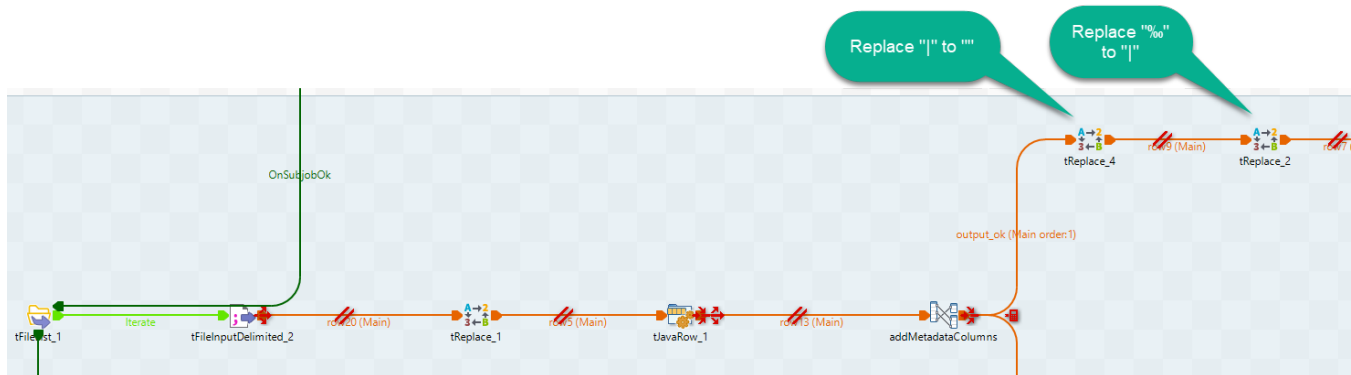
Number of rows: 1

Mode:
☐ Use Single Table
☒ Use Inline Table

File_name	Table_name	ODS_table_name	Valid	Separator	Escape_separator
"MEC_IT_0000_0000_F016_..."	"STG_BWH_0000_0000_F016..."	"ODS_BWH_0000_F016_F_W..."	context.LVAR_mecano_xtra...	"%"	"\"
"MEC_IT_0000_0000_F015_..."	"STG_BWH_0000_0000_F015..."	"ODS_BWH_0000_F015_F_W..."	context.LVAR_mecano_xtra...	"%"	"\"
"MEC_IT_0000_0000_F014_..."	"STG_BWH_0000_0000_F014..."	"ODS_BWH_0000_F014_F_W..."	context.LVAR_mecano_xtra...	"%"	"\"
"MEC_IT_0000_0000_F013_..."	"STG_BWH_0000_0000_F013..."	"ODS_BWH_0000_F013_F_W..."	context.LVAR_mecano_xtra...	"%"	"\"
"MEC_IT_0000_0000_F012_..."	"STG_BWH_0000_0000_F012..."	"ODS_BWH_0000_F012_F_W..."	context.LVAR_mecano_xtra...	"%"	"\"
"MEC_IT_0000_0000_F011_..."	"STG_BWH_0000_0000_F011..."	"ODS_BWH_0000_F011_F_W..."	context.LVAR_mecano_xtra...	"%"	"\"
"MEC_IT_0000_0000_F009_..."	"STG_BWH_0000_0000_F009..."	"ODS_BWH_0000_F009_F_W..."	context.LVAR_mecano_xtra...	"%"	"\"
"MEC_IT_0000_0000_F008_..."	"STG_BWH_0000_0000_F008..."	"ODS_BWH_0000_F008_F_W..."	context.LVAR_mecano_xtra...	"%"	"\"

It is use the same context as job F100_MEC_EXTRACT_BW_QUERY_TO_GSC and hardcode the separator to %

2. Up load the file from GCS to ODS table, which copy from reference job but only this part is different



A.3 F300_MEC_LOAD_ALL_DIM_TABLE_PART_1

To load ODS to DM layer for:

- DIM_priority
- DIM_user_status
- DIM_system_status

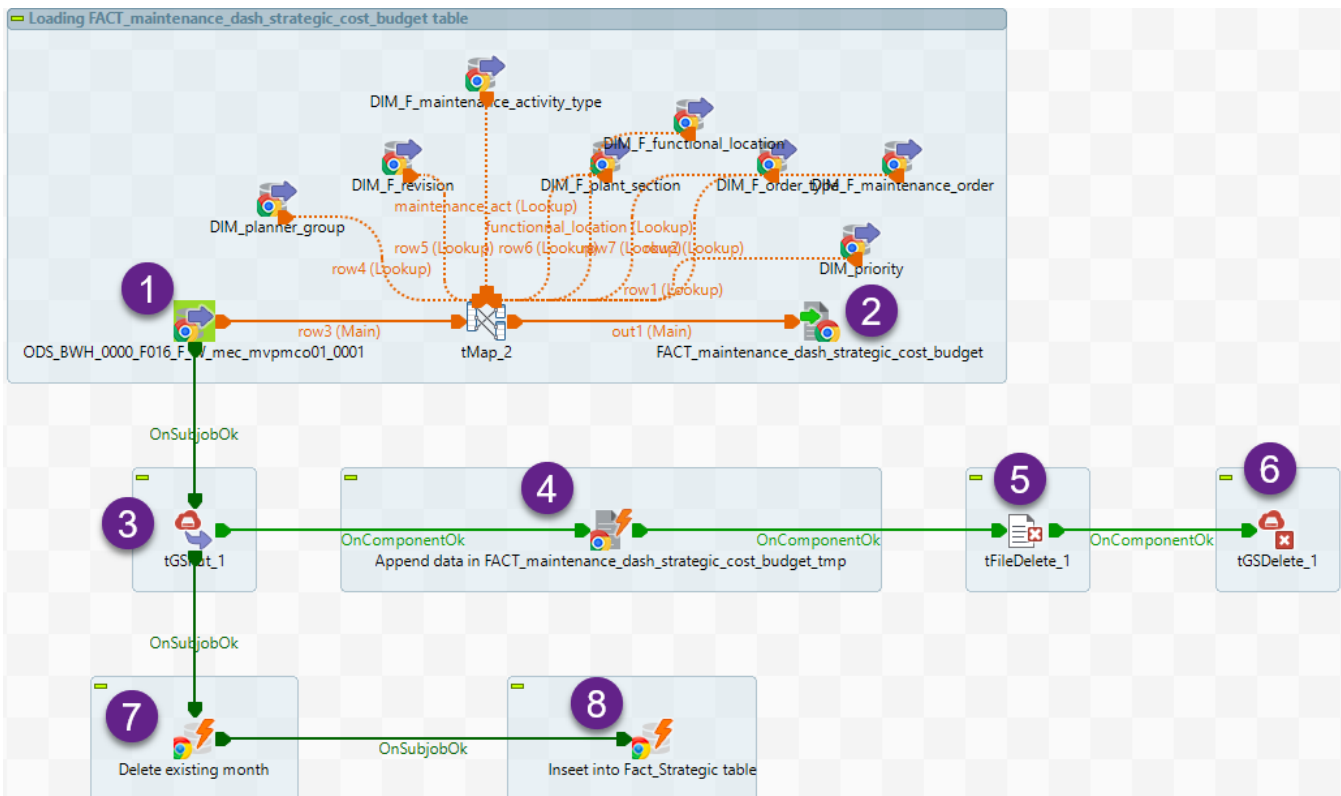
A.4 F300_MEC_LOAD_ALL_DIM_TABLE_PART_2

To load ODS to DM layer for:

1. DIM_F_maintenance_activity_type : MaintActivityType_Key , MaintActivityType (only for PF1)
2. DIM_F_equipment : Equipment_Key, Equipment
3. DIM_F_maintenance_order: Maintenance_Order_Key, Maintenance_Order
4. DIM_F_planner_group: Planner_Group_Key, Planner_Group, Source_System_Key
5. DIM_F_order_type: Order_Type_Key, Order_Type (only for WP1)
6. DIM_F_functional_location: Functional_Location_Key, Functional_Location, Source_System_Key
7. DIM_F_revision: Revision_Key, Revision
8. DIM_F_work_center: Work_Center_Key, Work_Center
9. DIM_F_plant_section : Plant_Section_Key, Plant_Section, Source_System_Key

A.5 F310_MEC_LOAD_FACT_MAINTENANCE_STRATEGIC

To load ODS to DM layer for FACT_maintenance_strategic. There is only 1 query ODS_BWH_0000_F016_F_W_mec_mvpmco01_0001



1. Select data from ODS with condition

```
from "+context.I_CNX_GOOGLE_MECANO_ProjectID+"."+context.I_LOCAL_GOOGLE_MECANO_DATASET_ODS+"
ODS_BWH_0000_F016_F_W_mec_mvpmco01_0001
inner join "+context.I_CNX_GOOGLE_MECANO_ProjectID+"."+context.I_LOCAL_GOOGLE_MECANO_DATASET_STAGING+"
ODS_BWH_0000_F016_F_W_mec_mvpmco01_0001.meta_run_id=log_files.meta_run_id and log_files.meta_file_name LIKE
'MEC_IT_0000_0000_F016_'+context.Business_date+'_0000_F_W_MEC_MVPMCO01_0001%.csv' and log_files.meta_status in ('OK','NOK')
```

2. Save the result of the FACT join with dimension tables to context.I_LOCAL_PATHDIR_MECANO_DATA+context.I_LOCAL_PATHDIR_MECANO_DM+"FACT_maintenance_strategic.csv"

3. Put the file to buckets context.I_LOCAL_PATHDIR_MECANO_DATA+context.I_LOCAL_PATHDIR_MECANO_DM+"FACT_maintenance_strategic.csv"

4. Upload the file to WDL.FACT_maintenance_dash_strategic_cost_budget_tmp

5. Delete file from 2

6. Delete file from 4

7. Delete data of existing month from the selection of ODS on DM.FACT_maintenance_strategic

```
DELETE FROM DM.FACT_maintenance_strategic
WHERE ref_date IN (SELECT DISTINCT CAST(ref_date as Date format 'MON YYYY') as ref_date FROM WDL
FACT_maintenance_dash_strategic_cost_budget_tmp)
8. Insert the data from ODS to DM.FACT_maintenance_strategic
```

⚠ There is no requirement to delete data manually when reload, which different from efficiency or effectiveness

A.6 F320_MEC_LOAD_FACT_MAINTENANCE_EFFECTIVENESS

To load ODS to DM layer for FACT_maintenance_effectiveness. There is only 2 queries

Special case: F12_QVMECANO_BW_QRY_MVPMOP04_0006 will select only max reference date from ODS > fact table

A.7 F330_MEC_LOAD_FACT_MAINTENANCE EFFICIENCY

To load ODS to DM layer for FACT_maintenance_efficiency. There is only 9 queries

Summary

Group / KPI	Tables / Dimension	Measure	1	2	3	4	5	6	7	8	9	Pri	US	SS
Strategic	ODS_BWH_0000_F016_F_W_mec_mvpmco01_0001	- actual_amount - plan_amount	x		x	x	x	x	x		x	x		
1.10	** [202410] add priority to Strategic													
Effectiveness	ODS_BWH_0000_F006_F_W_mec_mvpmcl01_0001	- planned_activity_nb_of_calls - planned_activity_executed_nb_of_calls - ongoing_activity_nb_of_calls - activity_delayed_nb_of_calls - activity_delayed_1_30_days_nb_of_calls - activity_delayed_31_90_days_nb_of_calls - activity_delayed_more_than_90_days_Nb_of_Calls - planned_activity_nb_of_hours - Planned_activity_executed_nb_of_hours	x	x	x	x	x	x	x	x	x			
3.02														
Effectiveness	ODS_BWH_0000_F014_F_W_mec_mvpmor01_0002	work_orders	x	x		x	x	x	x	x	x	x		
3.03	** [202410]only this query is mapping created_on with reference_date to Effectiveness													
Efficiency	ODS_BWH_0000_F013_F_W_mec_mvpmor04_0001	- nb_of_orders_in_progress - kpi_2_11_nb_of_orders_executed_with_a_required_end_date - kpi_2_11_nb_of_orders_executed_before_required_date - kpi_2_11_nb_of_orders_executed_without_a_required_end_date - kpi_2_11_number_of_orders_executed	x			x	x	x	x	x	x	x	x	x
2.03/2.12/2.14														
Efficiency	ODS_BWH_0000_F015_F_W_mec_mvpmop02_0010	- barcode_scan_rate - nb_operations_well_scanned - nb_operations_completed	x			x	x	x	x	x	x	-		
2.11														
Efficiency	ODS_BWH_0000_F007_F_W_mec_mvpmop04_0001	- nb_operation_scheduled_kpi6 - nb_operation_well_confirmed_kpi6 - nb_operations_in_delay - remaining_work_backlog_h - remaining_work_backlog_man_days	x			x	x	x	x	x	x	x	xx	xx
2.06/2.07/2.09														
Efficiency	ODS_BWH_0000_F008_F_W_mec_mvpmop04_0002	- nb_operation_planned - nb_operation_planned_good_day - nb_operation_reactives	x			x	x	x	x	x	x	x	x	x
2.04														
Efficiency	ODS_BWH_0000_F009_F_W_mec_mvpmop04_0003	- nb_operation_unscheduled - nb_operation_planned - nb_operation_planned_good_day - nb_operation_planned_good_week - nb_operation_reactives - hours_of_operations_planned - hours_of_operations_planned_good_day - hours_of_operations_planned_good_week - hours_of_operations_reactives - nb_operations_with_planned_hr - nb_operation_with_actual_hr - hours_of_operation_actual - hours_of_operation_actual_for_cal_delta - hours_of_operation_planned_for_cal_delta - nb_completed_operation_with_actual_hr - nb_completed_operation - posted_file_delay(V_FACT_gtbu_reactivity) - technical_validation_delay(V_FACT_gtbu_reactivity) - nb_rejection(V_FACT_gtbu_reactivity)	x		x	x	x	x	x	x	x	x	xx	xx
2.06														
Efficiency	ODS_BWH_0000_F011_F_W_mec_mvpmop04_0005	- remaining_work_hours - available_capacity - available_capacity_unit	x			x	x		x	x	x	x		
2.08														
Efficiency	ODS_BWH_0000_F012_F_W_mec_mvpmop04_0006	- nb_operation_planned - nb_operation_planned_good_day - nb_operation_reactives	x			x	x	x	x	x	x	x	xx	xx
2.13	(daily load)													
Efficiency	ODS_BWH_0000_F004_F_W_mec_mvpmno04_0001	- nb_of_notification_created - nb_of_notif_in_progress - nb_of_notif_in_progress_without_work_order - nb_of_notif_in_progress_with_delay - nb_of_notif_in_progress_delay_1_to_30_days - nb_of_notif_in_progress_delay_31_to_90_days - nb_of_notif_in_progress_delay_more_than_91_days - nb_of_notif_in_progress_no_required_end_date - nb_of_notif_with_attachement (V_FACT_notification_planning)				x		x		x	x	x	x	x
2.01/2.02/2.05_hist														
Efficiency	ODS_BWH_0000_F005_F_W_mec_mvpmno04_0002	- nb_of_notification_created - nb_of_notif_in_progress - nb_of_notif_in_progress_without_work_order - nb_of_notif_in_progress_with_delay - nb_of_notif_in_progress_delay_1_to_30_days - nb_of_notif_in_progress_delay_31_to_90_days - nb_of_notif_in_progress_delay_more_than_91_days - nb_of_notif_in_progress_no_required_end_date - nb_of_notif_with_attachement (V_FACT_notification_planning)				x		x		x	x	x	x	x
2.01/2.02/2.05_det														

Note: number of 1 - 9 are the dimension and Pri = priority, US = user status and SS = system status

Data Dictionary

Naming convention on BW provider such as MVMPMNO04

1-2 Data layer	3-4 Domain	5-6 Sub domain	7-8 only for mecano
----------------	------------	----------------	---------------------

CP = Composite provider	PM = Plant Maintenance	CL = Plan call	01 Detail up to date
MV = Multi provider	FI = Financial	CO = Cost	02 Detail Snapshot
CR = Reporting layer	SD = Sales and distribution	NO = Notification	03 Aggregate up to date
DB = Business layer		OP = Operation	04 Aggregate Snapshot
DP = Propagation layer		OR = Order	

Query ODS_BWH_0000_F012_F_W_mec_mvpmop04_0006 will be loaded on Daily flow as well and this query has a special component to select max (reference_date) from the DM.FACT_maintenance_efficiency WHERE kpi_no='2.13', which will be related only this query.

Then the selection from ODS will have condition to select only the data that more than max reference_date from FACT_maintenance_efficiency

WHERE PARSE_DATE('%d.%m.%Y', SnapShot__Date)>" + TalendDate.formatDate("yyyy-MM-dd",((java.util.Date)globalMap.get("max_ref_date")))+""

A. Additional Development on project 13129

A1. CTBU (not yet available)

To get the information about shce_creation_date(C_SEDAT), date_file_posted(C_FPDAT), end_real_date(e_PM_OPE2_C_FINREEL) and date_rejection(C_SPDAT) of the PM orders in order to calculate "posted file delay", technical validation delay" and "percentage of validation rejected" by loading from BW query DO_BW_QRY_MPR_PM05_003 via Xtract job TALEND_DEV_DO_BW_QRY_MPR_PM05_003

Talend job on DATA_OCEAN_DOMAIN INDUSTRIAL project

F002_BWH_MPR_PM05_003_to_ODS-

This job use reference job and follow the standard and there is no job to load to DM layer

Context parameter-

~~I_VAR_XTRACT_FILE_TALEND_DO_BW_QRY_MPR_PM05_003 = from to read on Xtract server
I_VAR_XTRACT_JOB_TALEND_DO_BW_QRY_MPR_PM05_003 = Xtract job name
I_LOCAL_VAR_STG_TABLE_TALEND_DO_BW_QRY_MPR_PM05_003 = STG table
I_LOCAL_VAR_ODS_TABLE_TALEND_DO_BW_QRY_MPR_PM05_003 = ODS table
I_VAR_XTRACT_PARA_TALEND_DO_BW_QRY_MPR_PM05_003 = "currentmonth" for loading last month to current month OR
"&YYYYMM_Start=202407&YYYYMM_End=202407" to reload~~

A2. Notification Planning

This will load the table SRGBTBREL (Generic Object Services) from PF1 and WP1 to get the number of the notification that has attachment

Generic Object Services (GOS): provides functionality for handling attachments, notes, and links to various objects in SAP transactions. Relationships in this environment might refer to how different objects (e.g., documents, business objects) are linked or related within the system.

- TYPEID_A : The type of the source object (e.g., notification, work order) filter on BUS2038 in DM, which is notification.
- TYPEID_B : The type of the target object (e.g., attached document, file)
- INSTID_A: The id of the object, in this case it will be notification id
- INSTID_B: If it is not empty, it means that we have attached file.
- RELTEYP: If it has value ATTA, it means that that object has attachement

There are many objects in this table. However, the notification object is

TYPEID_A(type_a_of_objects_in_persistent_object_references) = 'BUS2038'. It will be filter on DM layer

To get the information about number of attachment on the PM orders in order to calculate kpi "Notification Created with Picture" by loading from ECC table SRGBTBREL on PF1/WP1 via Talend job on DATA_OCEAN_DOMAIN INDUSTRIAL project. There are 2 main flow

2.1 To get the attachment

F100_DIM_generic_object_service_SRGBTBREL_SAP_TO_DM main flow job that call 3 sub jobs

F001_SPF_F001_I_D_SRGBTBREL_TO_BQ F001_SPW_F001_I_D_SRGBTBREL_TO_BQ
F010_DIM_generic_object_service_SRGBTBREL_TO_DM

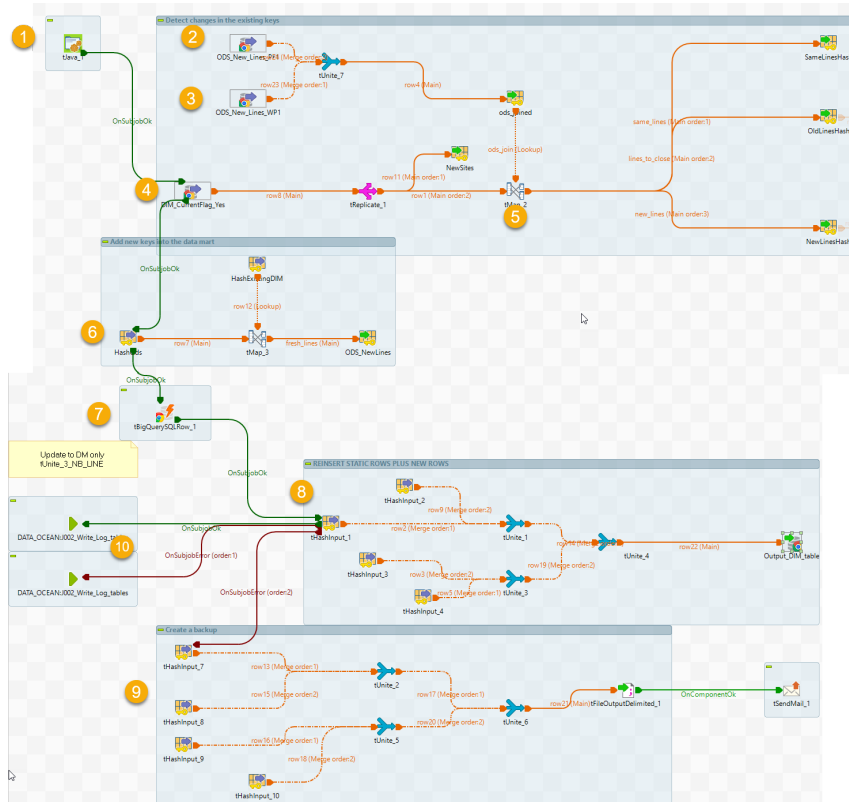
F001_SPF_F001_I_D_SRGBTBREL_TO_BQ and F001_SPW_F001_I_D_SRGBTBREL_TO_BQ

It is from reference job to load with incremental load by using field UTCTIME to update incremental_loading table with value PF1_SRGBTBREL and WP1_SRGBTBREL

I_VAR_[system]_SRGBTBREL_BUCKET = bucket location
 I_VAR_[system]_SRGBTBREL_BQ_Table_ODS = ODS table
 I_VAR_[system]_SRGBTBREL_BQ_Table_STG = STG table
 I_VAR_[system]_SRGBTBREL_INC_LOAD = incremental load field value PF1_SRGBTBREL or WP1_SRGBTBREL
 I_VAR_[system]_SRGBTBREL_additional_filter = to filter in case of reloading
 I_VAR_[system]_SRGBTBREL_email_flag = yes then Talend will inform when incremental load time is equal or less than time stamp in incremental_loading table
 I_VAR_[system]_SRGBTBREL_email_recipient = email address to inform
 I_VAR_[system]_SRGBTBREL_dm_reload_condition = to set reload condition from ODS to DM

F010_DIM_generic_object_service_SRGBTBREL_TO_DM

Detail job = J020_DIM_SRGBTBREL_SCD_Type2



2.2 To get the notification data

F100_MEC_EXTRACT_BW_QUERY_TO_GSC J201_STG_TO_ODS F010_FACT_notification_planning_TO_DM

To load source prj-data-dm-industrial-dev.ODS.ODS_BWH_0000_F005_F_W_mec_mvpmno04_0002 fact_notification_planning

B Loading to Data Ocean Industrial

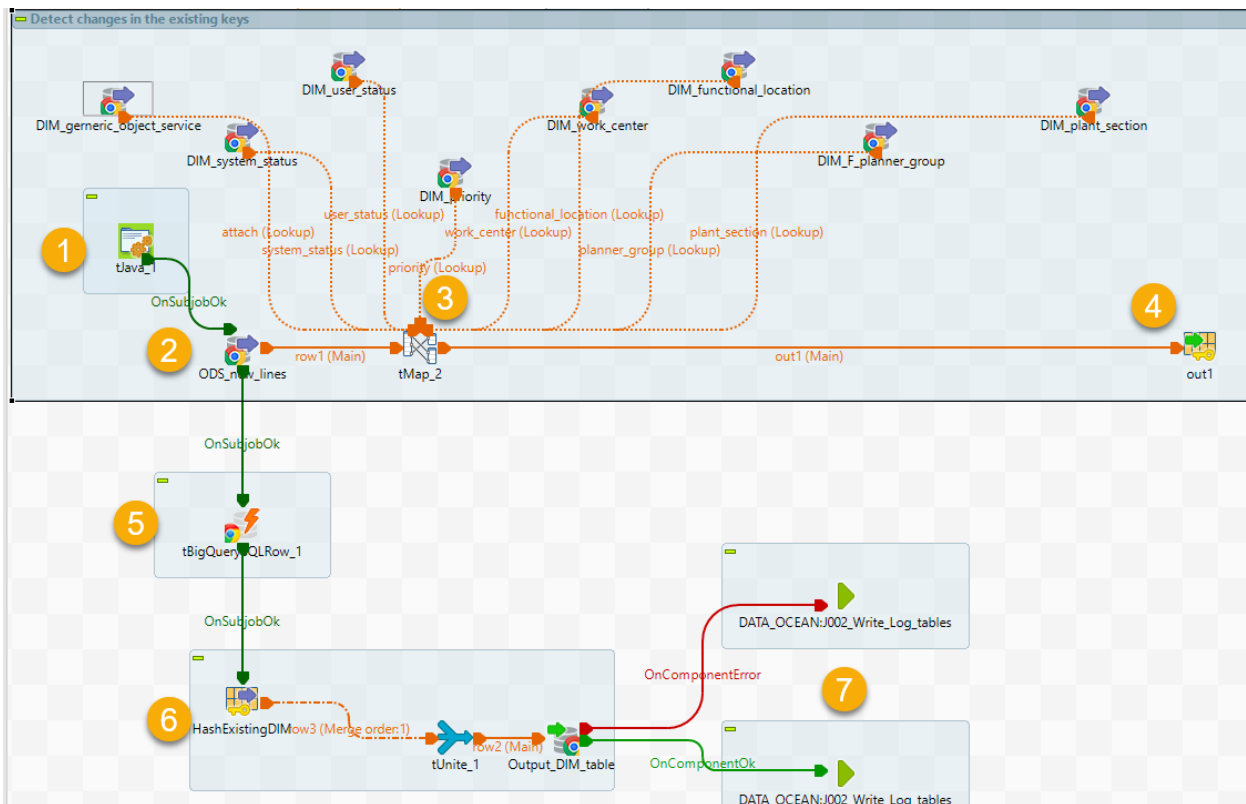
The new development in 2025 to add new KPI

~~B.1 F002_BWH_MPR_PM05_003_to_ODS (not yet available)~~

This will load the maintenance operation order with the important date such as date_file_posted(C_FPDAT), end_real_date(e_PM_OPE2_C_FINREEL), shce_creation_date(C_SEDAT), and date_rejection(C_SPDAT) from BW query. These dates can calculate the KPI Posted File Delay, Technical Validation Delay, Validation Rejected%

B.2 F010_FACT_notification_planning_to DM

Detail job is J020_FACT_notification_planning



1. Check reload on DM.FACT_notification_planning condition for incremental or reload
2. Select data from ODS_BWH_0000_F005_F_W_mec_mvpmno04_0002 only the max meta_business_date for incremental and status loading is OK
3. Join with dimension (the attachment will be on dimension DIM_generic_object_service)
 - a. nb_of_notif_with_attachment = Relational.ISNULL(attach.generic_object_services_key)?0:1
 - b. selection from DIM_generic_object_service
 SELECT generic_object_services_key,
 concat(upper(substr(meta_source_system,1,3)),'_','client','/',REGEXP_REPLACE
 (instance_ident_a_in_bor_compat_persistent_object_references,'^0+', ''))
 instance_ident_a_in_bor_compat_persistent_object_references,
 FROM DM.DIM_generic_object_services
 where type_a_of_objects_in_persistent_object_references = 'BUS2038'
 and relationship_type = 'ATTA'
 and current_flag
 QUALIFY ROW_NUMBER() OVER (PARTITION BY meta_source_system,
 instance_ident_a_in_bor_compat_persistent_object_references,start_date ORDER BY inserted_date DESC) = 1
4. Keep output in memory
5. Delete the data of the fact table that have the same key in the ODS
6. Update the new records to the fact table
7. Update log

The new DM.FACT_notification_planning will join in the prj-data-maintenance-dash-dev.DM.FACT_maintenance_dash (part of V_FACT_maintenance_efficiency)

C. F_500_REFRESH_MAINT_DASH_VIEW

This job will run **sql query** to delete prj-data-maintenance-dash-[env].DM.FACT_maintenance_dash and fill the table again with the view prj-data-maintenance-dash-[env].DM.FACT_maintenance_dash with parameter I_LOCAL_VAR_FACT_HISTORY_NUM_MONTH in order to control how many months that need for the final table for dashboard. Normally it is 6 month

D. F100_REFRESH_ACESS_RIGHTS

To update authorization from **gsheet** (sheet name = "for training sessions and acces") : save to prj-data-dm-industrial-[env].WDL.
 WRK_maintenance_dash_access to assign permissions to end users

E. Refresh Tableau

To trigger Tableau to load data from GCP

Loading Data

A. MECANO (Talend project MECANO but using Data Ocean Industrial)

Normally it will be incremental load every week on PL_MECANO_DASH every Monday at 04:30 AM CET

and PL_MECANO_DAILY_LOAD will load with delete data on xtract TALEND_QVMECANO_BW_QRY_MVPMOP04_0006 only on Tue, Wed, Thur, Fri at 04:30 CET

[Reloading data detail in this link](#)

B. Data Ocean Industrial (Talend project DATA_OCEAN_DOMAIN INDUSTRIAL)

~~B.1 F002_BWH_MPR_PM05_003 to ODS~~

~~This data flow will be incremental and it is added to the plan PL_MECANO_DASH~~

~~Reload data~~

~~I_VAR_XTRACT_PARA_TALEND_DO_BW_QRY_MPR_PM05_003 = "&YYYYMM_Start=202407&YYYYMM_End=202407" (in case reload July 2024)~~

~~If it is normal load this variable should be "currentmonth"~~

B.2 F010_FACT_notification_planning_to DM

This job is also added to PL_MECANO_DASH

Reload data

ODS: it is in [the mecano loading MVPMNO04_0002](#)

ODS: SRGBTBREL tables

Just change value WP1_SRGBTBREL, PF1_SRGBTBREL on table prj-data-dm-industrial-[evn].STG.incremental_loading

[UPDATE](#) STG.incremental_loading

[SET](#) meta_last_process_date = '2024-06-06 00:00:00'

[where](#) meta_file_name = ' WP1_SRGBTBREL '

DM: DIM_generic_object_service

Control the reload by I_VAR_SPF_SRGBTBREL_dm_reload_condition and I_VAR_SPF_SRGBTBREL_dm_reload_condition such as QUALIFY ROW_NUMBER() OVER (PARTITION BY CONCAT(meta_source_system,CLIENT,BRELGUID) ORDER BY meta_ods_insert_date DESC) = 1 in order to reload everything from ODS.

DM : maintain I_VAR_BWH_notification_planning_dm_reload_condition. The query to select ODS will be:

```
select xxx FROM ODS."+context.I_VAR_ODS_TABLE+" o
JOIN STG.log_files log_files ON o.meta_run_id = log_files.meta_run_id and lower(log_files.meta_file_name) like '%mvpmno04_0002%'
WHERE log_files.meta_status = 'OK' "+ context.I_VAR_BWH_notification_planning_dm_reload_condition
```

Therefore, it can enter where condition in order to specific to reload such as

"and SnapShot__Week = '20.2024'

If it is normal load this variable should be "incremental"

Note: Reload of this fact table will cause the selected week to update the number of attachment is the latest one based on DIM_generic_object_services. It is better to NOT reload this fact table

Example Case

The number of attachment is too low on week 11.2025

```

select reference_week , sum ( nb_of_notification_created ) nb_of_created , sum ( nb_of_notif_with_attachement ) nb_of_attachement
from `prj-data-dm-industrial-dev.DM.FACT_notification_planning`
where reference_week in ( '10.2025' , '11.2025' , '12.2025' )
and nb_of_notification_created = 1
group by reference_week
order by reference_week ;

```

reference_week ▾	nb_of_created ▾	nb_of_attachement ▾
10.2025	8633	576
11.2025	8934	76
12.2025	9352	705

Random validate with table SRGBTBREL

```

select instance_ident_a_in_bor_compat_persistent_object_references from `prj-data-dm-industrial-dev.DM.DIM_generic_object_services`
where instance_ident_a_in_bor_compat_persistent_object_references like '%1102765340'

```

=> No data

```

select RELTYPE , INSTID_A , TYPEID_A from prj-data-dm-industrial-dev.ODS.ODS_SPF_0000_F001_I_D_SRGBTBREL
where INSTID_A like '%1102765340'

```

RELTYPE ▾	INSTID_A ▾	TYPEID_A ▾
ATTA	001102765340	BUS2038
ATTA	001102765340	BUS2038
ATTA	001102765340	BUS2038
ATTA	001102765340	BUS2038
ATTA	001102765340	BUS2038

It means that data in DIM_generic_object_services has something wrong.

Solution:

1. Reload DIM_generic_object_services by change following parameter:

I_VAR_SPF_SRGBTBREL_dm_reload_condition = "where UTCTIME > 20250301000000"

I_VAR_SPW_SRGBTBREL_dm_reload_condition = "where UTCTIME > 20250301000000"

To reload all data that more than 1 Mar 2025 from ODS both PF1 and WP1 to DIM_generic_object_services by running job F100_DIM_generic_object_service_SRGBTBREL_SAP_TO_DM.

Note: it may need to change process to have more memory in TMC in order to reload this.

2. Reload FACT_notification_planning by changing parameter:

I_VAR_BWH_notification_planning_dm_reload_condition = "and SnapShot_Week = '11.2025'"

To reload ODS_BWH_0000_F005_F_W_mec_mvpmno04_0002 only week 11.2025 to mapping with DIM_generic_object_services to find the number of attachment again.

Run job F010_FACT_notification_planning_TO_DM

3. Change parameters back to 'incremental'

4. Test with script above again, the number of attachment should be increased on week 11.2025

reference_week ▼	nb_of_created ▼	nb_of_attachement
10.2025	8633	576
11.2025	8934	710
12.2025	9352	705

5. Run job F_500_REFRESH_MAINT_DASH_VIEW to update prj-data-maintenance-dash-dev.DM.FACT_maintenace_dash

Logging

Loading job

in `prj-data-dm-industrial-[environment].STG.[table]`

```
select job.job_name , job.meta_start_date , logs.meta_run_id , logs.meta_source_system , logs.meta_step , logs.meta_status , logs.meta_num_lines , logs
.meta_error_lines from STG.log_tables logs join STG.run_jobs job on logs.meta_run_id = job.meta_run_id
```

```
where logs.meta_run_id in ( SELECT meta_run_id FROM STG.run_jobs order by meta_start_date desc limit 1000 )
```

```
and ( ( meta_source_system like '%MVP%' or meta_source_system like '%DBPMMD%' ) or
```

```
lower ( job_name ) like '%srgbtbre%' or lower ( job_name ) like '%do_bw_qry_mpr_pm05_003%')
```

```
and meta_step = 'Bucket to Staging'
```

```
and meta_start_date > DATE_SUB ( CURRENT_TIMESTAMP () , INTERVAL 7 DAY )
```

```
order by job.meta_start_date desc , meta_source_system
```

⚠ Loading on Monday must have 17 lines

Check loading from Xtract in detail

```
select count ( distinct kpi_no ) , DATE ( inserted_date ) ins_date , meta_run_id from `DM.FACT_maintenance_efficiency`
```

```
where EXTRACT ( DAYOFWEEK FROM inserted_date ) = 2
```

```
and kpi_no != '2.13'
```

```
group by ins_date , meta_run_id
```

```
order by ins_date desc
```

f0_ ▼	ins_date ▼	meta_run_id ▼
8	2025-03-24	6e60caf1-f995-48e7-816f-19aa...
7	2025-03-17	7f325782-3e1d-4553-b429-317...
7	2025-03-10	3d18b3fd-44a2-4511-b113-618...
7	2025-03-03	ab5fbf0a-aec3-4e9d-aa17-b7a...
7	2025-02-24	50ad526f-98ea-4085-bedc-c59...
7	2025-02-17	4302b82c-1df5-4e47-8e4f-28d...
7	2025-02-10	5e4b470e-bd26-49e3-82cb-4a0...

Current Monday must have 8 , Previous Monday must have 7 (exclude 2.13, which daily load and it can be miss on Monday). In case, there is missing kpi_no. Check further

```
select distinct kpi_no , DATE ( inserted_date ) from `DM.FACT_maintenance_efficiency`
```

```
where meta_run_id = meta_run_id from above
```

order by kpi_no