EliteSeries Analytics

Release 9.4

Data Mart Schema and Mapping Guide

TECSYS®

Copyrights and Trademarks

Printed in Canada.
June 2016
TE-GT940

Corporate Headquarters:

TECSYS Inc.
1 Place Alexis Nihon
Suite 800
Montreal, Quebec
Canada H3Z 3B8


1-514-866-0001
1-800-922-8649


1-514-866-1805


documentation@tecsys.com

Table of Contents

1 Introduction

1.1 About this Guide

This document is a comprehensive guide to the construct of the various data marts that are part of EliteSeries Analytics. This guide includes the database schema design, information on the various database tables, and mapping information that describes the source and calculations (if applicable) used to produce the data item.

1.2 Intended Audience

Since this document focuses specifically on data marts, it is recommended that the reader of this document be familiar with the contents of the EliteSeries Analytics User Guide.

1.3 Conventions

This topic describes the conventions dictating the use of certain text elements and symbols within this document.

Convention	Definition
Text that appears in this Times New Roman font	The default font that is used for regular body text.
Text that appears in this Arial font	Titles of chapters, headings and subheadings, captions, and static elements that define the structural organization of this document.
Text That Appears in Title Text	View names, table names, and resource names. Page names, page section titles, subview titles, related resource titles, and tab titles from dialog pages. Field names and valid field values and/or descriptions, when these are in the process of being described or when direct cross references are made.
Text that is underlined	Topic or document cross references, either internal or external.
TEXT THAT IS IN SMALL CAPS	Keys that you press on a keyboard.
Text that is bolded	Actions (generally buttons) and links that you click to select. Textual equivalent of certain icons (e.g. the Graph icon) that must be clicked to perform a given action.
Text that is italicized	Terms or expressions on which you have placed emphasis.

Table 1-1: Typographical conventions

Symbol	Definition
	Note
	Warning
	Start of a procedure

Table 1-2: Symbol conventions

1.4 Getting Customer Support

Should you need assistance beyond what this guide can provide, contact our Software Assistance Help Desk. TECSYS trained professionals may be reached at 1-888-TECSYS5. Before you call TECSYS Software Assistance, please perform the following steps:

- Check the documentation to ensure that the information you require has not already been supplied.
- Write down a detailed description of your problem or query, to assist you in describing the issue over the phone. Please note any error messages that are displayed on your screen.

2 Overview

Data marts are used to capture data that is used to produce performance metrics. Each data mart focuses on a specific area of the supply chain, such as warehouse performance, sales, shipping, and so on.

All EliteSeries data marts are based on a star schema design. The data is stored in a central fact table, with one or more tables holding information on each dimension. The central fact tables hold measures (e.g. sales amounts, quantities, and so on) and foreign key references. Foreign key references on a fact table are hidden and the dimension tables hold attributes (e.g. customer name and address, item description, and so on). Dimensions have levels, and all levels are usually displayed as columns in each dimension table. For example, the time dimension may contain the year, period, and day levels, while the customer dimension contains the S/A customer, customer, and ship-to levels.

All data marts consist of data derived from a transactional system or from third-party data sources (i.e. benchmark data). In order to populate the data marts with transactional data, a series of extraction, transformation, and load (ETL) scripts are used.

Refreshing a data mart can be done by using BI Tools, via scheduled jobs that can be defined on the BI server (refer to the EliteSeries Analytics User Guide for more information) or via a queue timer available in EliteSeries.

All data marts are designed to refresh based on new or changed transactional data; therefore, a data mart refresh is usually an incremental update (the GL data mart is recreated each time the data mart is refreshed). In the event that a data mart needs to be rebuilt, an SQL purge script is provided.

2.1 ETL - DMS General Ledger

BiDmrGeneralLedgerUpdate class(TecsysBI\src\com\tecsys\bi\tools\ext)

Update the bi_gl's dimensions and fact tables.

- bi_gl_time:
  - Select from the glperiod where the start_date <= today(current)
- bi_gl_hist:
  - Select from the gl_chart table all gl that are not defined as a retained gl code.
  - Select all period/year from the glperiod where the start date is before or equal to today's date.
  - Select from gl_hist and/or the gl budget table.
  - The retained gl code is processed at the end.

The General Ledger data mart is recreated each time the refresh is executed.

Each of the dimension and fact tables includes placeholder fields that can be used for custom purposes.

If no budget record exists, a record in the bi_gl_bdgt will be created with the gl_bdgt_key and gl_budget_code as '-' and the budget_desc_1 as 'no records defined in DMS Budgets.'

2.2 ETL - DMS Item Demand

BiDmrDmsItemDemandUpdate class(TecsysBI\src\com\tecsys\bi\tools\ext)

Update the bi_dm's dimensions and fact tables.

- bi_dm_time:
    - Select from the st_cal_p_date where the cal_code match the global.sa_cal_code.
- bi_dm_dmnd:
    - The latest item_w_demand.demand_date is stored in the bi_dmr_att table as the dmr_name = 'item-demand' and the dmr_att_key as '*bi_dm_dmnd!last_datetime_processed'*.
    - Since while refreshing, new transaction might come in, the server system date is retrieved and a special method is call to wait until no transaction.
    - Select from item_w_demand where the mod_stamp or create_stamp from item_w_demand > last demand date processed and mod_stamp or create_stamp <= to the server system date.
    - Each of the dimension and fact tables includes placeholder fields that can be used for custom purposes.

The Item Demand data mart may be rebuilt using the provided SQL statement located under TecsysBI/app/eliteseries/dmr/schema /item-demand/*database*/purge.sql.

2.3 ETL - DMS Sales Analysis

BiDmrSalesAnalysisUpdate class(TecsysBI\src\com\tecsys\bi\tools\ext)

Update the bi_id's dimensions and fact tables.

- bi_sa_time:
    - Select from the st_cal_p_date where the cal_code match the global.sa_cal_code and the global.master_cal_code.
- bi_sa_sale:
    - The last sa.sa_id is stored in the bi_dmr_att table as the dmr_name ='sales-analysis' and the dmr_att_key as 'last_sa_id'.
    - Since while refreshing, new transaction might come in, the maximum sa_id is selected from sa and a special method is call to wait until no transaction.
    - Select from sa where sa.sa_id > last sa_id processed and sa.sa_id <= to the maximum sa.sa_id.

Each of the dimension and fact tables includes placeholder fields that can be used for custom purposes.

The Sales Analysis data mart may be rebuilt using the provided SQL statement located under TecsysBI/app/eliteseries/dmr/schema/sales-analysis/database/purge.sql.

2.4 ETL - FCST Demand and Forecasting

BiDmrFcstDemandForecastUpdate class(TecsysBI\src\com\tecsys\bi\tools\ext)

Update the bi_id's dimensions and fact tables.

- bi_id_time:
  - Select from the calendar_day.
- bi_id_dmnd:
  - The latest demand.demand_date is stored in the bi_dmr_att table as the dmr_name = 'demand-forecast' and the dmr_att_key as '*bi_id_dmnd!last_datetime_processed'*.
  - Since while refreshing, new transaction might come in, the server system date is retrieved and a special method is call to wait until no transaction.
  - Select from demand where the mod_stamp or create_stamp from demand > last demand date processed and mod_stamp or create_stamp <= to the server system date.
- bi_id_fcdm:
  - Select from forecast where the mod_stamp or create_stamp from forecast > mod_stamp or create_stamp <= to the server system date.
  - The latest forecast.mod_stamp/forecast.create_stamp is stored in the bi_dmr_att table as the dmr_name = 'demand-forecast' and the dmr_att_key as 'bi_id_fcdm!last_datetime_processed'.
  - Each of the dimension and fact tables includes placeholder fields that can be used for custom purposes.

The Demand and Forecasting data mart may be rebuilt using the provided SQL statement located under TecsysBI/app/eliteseries/dmr/schema /demand-forecast/*database*/purge.sql.

2.5 ETL - SMS Supply Performance

BiDmrSupplyPerformanceUpdate class(TecsysBI\src\com\tecsys\bi\tools\ext)

Update the bi_sp's dimensions and fact tables.

- bi_sp_time:
  - The start year is determined by using the earliest movement create stamp.
  - The end year is determined by using the latest movement create stamp.
- bi_sp_cons:
  - The last movement.movement_id is stored in the bi_dmr_att table as the dmr_name = 'supply-performance' and the dmr_att_key as 'bi_sp_cons!last_movement_id!*site_code(hospital)*.

- Since while refreshing, new transaction might come in, the server system date is retrieved and a special method is call to wait until no transaction.
- Select from movement where the location type is 0 and the movement id > last movement id processed and movement id <= to the max movement id for each site.

- bi_sp_invt:
  - Select from movement where the movement id > last movement id processed and movement id <= to the max movement id for each site.
  - The last movement.movement_id is stored in the bi_dmr_att table as the dmr_name = 'supply-performance' and the dmr_att_key as 'bi_sp_invt!last_movement_id!s*ite_code(hospital)*.
  - Each of the dimension and fact tables includes placeholder fields that can be used for custom purposes.

The Supply Performance data mart may be rebuilt using the provided SQL statement located under TecsysBI/app/eliteseries/dmr/schema /supply-performance/*database*/purge.sql.

2.6 ETL - TMS Shipping Performance

BiDmrCarrierPerformanceUpdate class(TecsysBI\src\com\tecsys\bi\tools\ext)

Update the bi_cp's dimensions and fact tables.

- bi_cp_time:
  - The start year is determined by using the earliest manifest create stamp where the closeout status is 999.
  - The end year is determined by using the latest manifest create stamp where the closeout status is 999.

- bi_cp_ship:
  - The date that the Carrier Performance was refreshed is stored in the bi_dmr_att table as the dmr_name = 'carrier-performance' and the dmr_att_key as 'last_datetime_processed'.
  - Since it is possible that, while refreshing, new transactions might come in, the server system date is retrieved and a special method is called to wait until there are no transactions.
  - Select from manifest joining to shipment_package, the shipment where the manifest mod stamp is after the last time it was processed and before or equal to today's datetime. Only includes records for which the closeout status is 999.

Each of the dimension and fact tables includes placeholder fields that can be used for custom purposes.

The Carrier Performance data mart may be rebuilt using the provided SQL statement located under TecsysBI/app/eliteseries/dmr/schema/carrier-performance/database/purge.sql.

2.7 ETL - WMS Warehouse Performance

BiDmrWarehousePerformanceWercImport class(TecsysBI\src\com\tecsys\bi\tools\ext)

Import into the werc's tables from the werc-import.txt file.

- werc-import.txt is located under the following folder: TecsysBI\app\eliteseries\dmr\schema\warehouse-performance.
- The werc-import.txt file stamp is store into the bi_import_file (import_file_name is werc-import.txt.
- The bi_wp_werc_... tables will be re-generated(deleted 1st) only when there is a new werc-import file. The new werc-import file stamp is compared to the file stamp value stored in bi_import_file.

BiDmrWarehousePerformanceUpdate class(TecsysBI\src\com\tecsys\bi\tools\ext)

Update the bi_wp's dimensions and fact tables.

- bi_wp_time:
  - The start year is determined by using the earliest order date from om_h where the it_f.transact is SHIP, by using the earliest it_f.end_time where the it_f.transact is CYCC or by using the earliest werc year.
  - The end year is determined by using the latest order date from om_h where the it_f.transact is SHIP, by using the latest it_f.end_time where the it_f.transact is CYCC or by using the latest werc year.
- bi_wp_ordl:
  - For the order line fact table, selecting all the it_f where the it_f.transact is SHIP and linking to the om_h, od_h to extract the value, if the concatenation of the ob_oid, ob_type and the ob_lno do not exist already it will insert into the order line fact table.
  - The last it_f.it_rid is stored in the bi_dmr_att table as the dmr_name ='warehouse-performance' and the dmr_att_key as 'bi_wp_ordl!last_it_rid!*warehouse code*'.
  - Since it is possible that, while refreshing, new transaction might come in, the maximum it_f.it_rid is selected from it_f and a special method is called to wait until there are no transactions.
  - Select from it_f where it_f.it_rid > last it_rid processed and it_f.it_rid <= to the maximum it_f.it_rid.
- bi_wp_coun:
  - For the cycle count fact table, selecting all the it_f where the it_f.transact is CYCC and the it_f.rsn_code is CYCC, it will first try to update the bi_wp_coun; if it does not exist, it will then insert into it.
  - The last it_f.it_rid is stored in the bi_dmr_att table as the dmr_name ='warehouse-performance' and the dmr_att_key as 'bi_wp_coun!last_it_rid!*warehouse code*'.
  - Since while refreshing, new transaction might come in, the maximum it_f.it_rid is selected from it_f and a special method is call to wait until no transaction.
  - Select from it_f where it_f.it_rid > last it_rid processed and it_f.it_rid <= to the maximum it_f.it_rid.

- bi_wp_rcpl:
    - For the receipt line fact table, selecting all the it_f where the it_f.transact is RCPT, it will first try to update the bi_wp_rcpl; if it does not exist, it will then insert into it.
    - The last it_f.it_rid is stored in the bi_dmr_att table as the dmr_name =’warehouse-performance’ and the dmr_att_key as ‘bi_wp_rcpl!last_it_rid!*warehouse code*’.
    - Since while refreshing, new transaction might come in, the maximum it_f.it_rid is selected from it_f and a special method is call to wait until no transaction.
    - Select from it_f where it_f.it_rid > last it_rid processed and it_f.it_rid <= to the maximum it_f.it_rid.
- bi_wp_capa:
    - For the warehouse capacity fact table, selecting all the iv_f records, this will need to be refresh daily, in the case for any reason some days were skipped, it will take it into consideration the number of days skipped, i.e.:

      Iv_f.qty * number of days (today’s date – last date processed) / number of days + 1
    - The Today’s date is stored in the bi_dmr_att table as dmr_name=’warehouse-performance’ and the bi_dmr_att_key as ‘bi_wp_capa!last_date_processed!*warehouse code*’.

Each of the dimension and fact tables includes placeholder fields that can be used for custom purposes.

The Warehouse Performance data mart may be rebuilt using the provided SQL statement located under TecsysBI/app/eliteseries/dmr/schema/warehouse-performance/database/purge.sql.

2.8 ETL - WMS Warehouse Transaction History

BiDmrWarehousePerformanceUpdate class(TecsysBI\src\com\tecsys\bi\tools\ext)

Update the bi_th’s dimensions and fact tables.

- bi_th_time:
    - The start year is determined by using the earliest it_f.start_time.
    - The end year is determined by using the latest it_f.start_time.
- bi_th_tran:
    - The last it_f.it_rid is stored in the bi_dmr_att table as the dmr_name =’warehouse-transaction-history’ and the dmr_att_key as ‘last_it_rid!*warehouse code*’.
    - Since while refreshing, new transaction might come in, the maximum it_f.it_rid is selected from it_f and a special method is called to wait until no transaction.
    - Select from it_f where it_f.it_rid > last it_rid processed and it_f.it_rid <= to the maximum it_f.it_rid.

Each of the dimension and fact tables include placeholder fields that can be used for custom purposes.

The Warehouse Transaction History data mart may be rebuilt using the provided SQL statement located under TecsysBI/app/eliteseries/dmr/schema/warehouse-transaction-history/database/purge.sql.

2.9 Analysis Models

Each of the data marts listed in this document have a corresponding analysis model defined using the Cognos Framework Manager. This model represents metadata that describes the relationships between tables and is used by Cognos to produce the SQL and MDX queries to retrieve information from the data mart. These models are required in order to expose a data mart to an end user.

The source file where these models are maintained is located under TecsysBI/app/eliteseries/model/Analysis Model/ on the BI server.

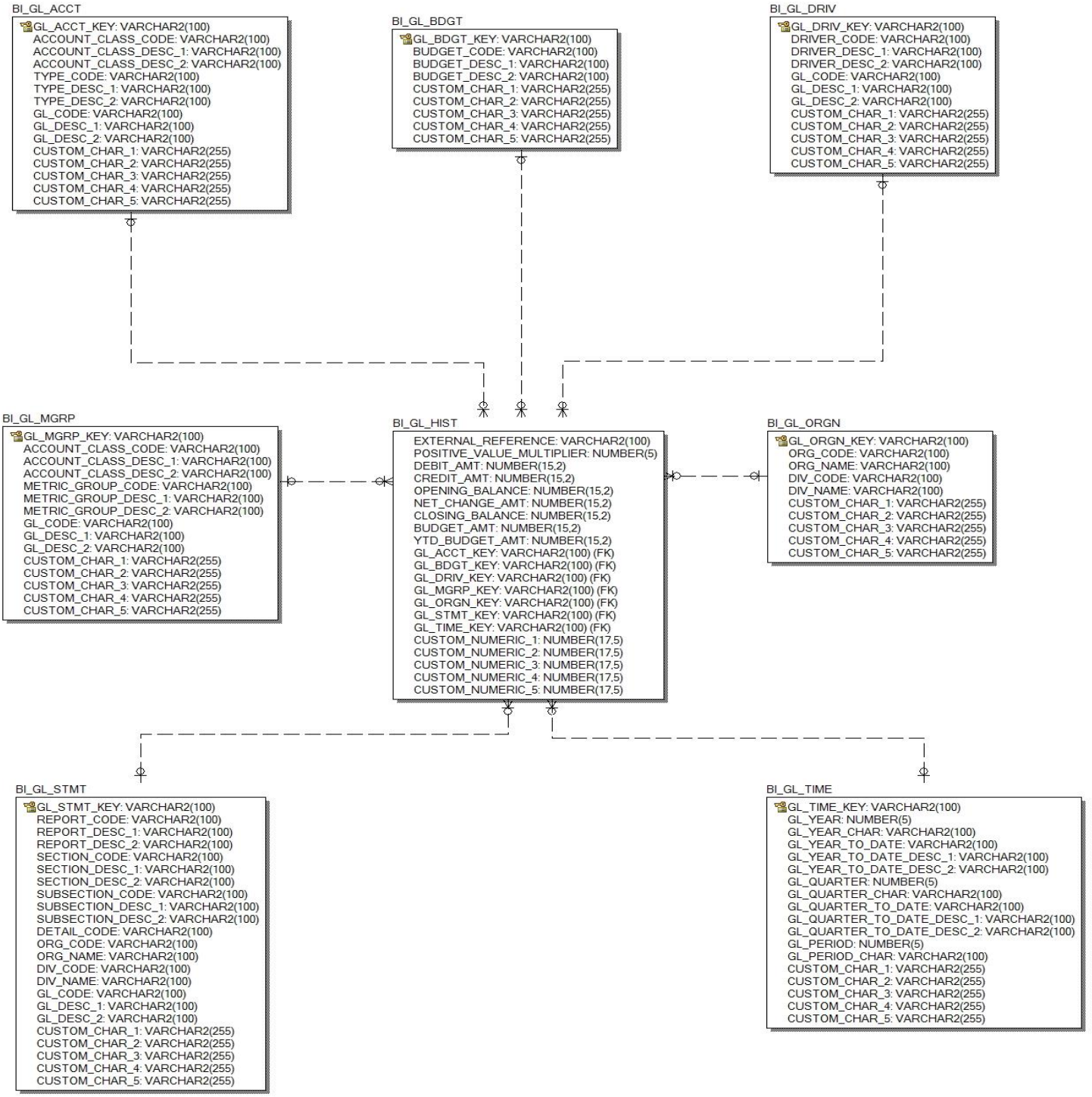
The models are published into the Cognos content store, and are thus accessible from within the Cognos portal. Whenever a user chooses to create new content, they will always be prompted to select a model.

Whenever a change is made to a data mart, such as new measures or new dimensions, the corresponding analysis model would require a change.

For more information on Analysis Models, please refer to the EliteSeries Analytics User Guide.

3 DMS General Ledger

3.1 Structure



3.2 Mapping

3.2.1 Columns for the BI_GL_HIST Table

Name	Value Coming From	Fact
EXTERNAL_REFERENCE	null	
GL_ACCT_KEY	gl_chart.org_code '!' gl_chart.gl_code	
GL_BDGT_KEY	gl_budget '-' gl_budget.rev_no	
GL_DRIV_KEY	gl_chart.org_code '!' gl_chart.gl_code	
GL_MGRP_KEY	gl_chart.org_code '!' gl_chart.gl_code	
GL_ORGN_KEY	gl_chart.org_code '!' gl_hist.div_code	
GL_STMT_KEY	gl_chart.org_code '!' gl_hist.div_code '!' gl_chart.gl_code	
GL_TIME_KEY	glperiod.yr '-' glperiod.period	[YYYY-PP]
POSITIVE_VALUE_MULTIPLIER	gl_chart.dr_cr_mult	
DEBIT_AMT	gl_hist.debit_amt	
CREDIT_AMT	gl_hist.credit_amt	
OPENING_BALANCE	gl_hist.net_chg_amt of period 0 CLOSING_BALANCE of previous period	When processing period 1 the opening balance = gl_hist.net_chg_amt of period 0
NET_CHANGE_AMT	gl_hist.net_chg_amt	
CLOSING_BALANCE	OPENING_BALANCE + gl_hist.net_chg_amt	
BUDGET_AMT	gl_budget_d.budget_amt	
YTD_BUDGET_AMT	sum(gl_budget_d.budget_amt) for the year	
CUSTOM_NUMERIC_1		
CUSTOM_NUMERIC_2		
CUSTOM_NUMERIC_3		
CUSTOM_NUMERIC_4		
CUSTOM_NUMERIC_5		

The BI general ledger tables are initialized each time there is a refresh.

- Flow for populating the fact table is as follows:
  Foreach gl_chart where gl_chart.gl_code != org_gl.retained_code
  Foreach div_code from gl_hist union all div_code from gl_budget_d
  Foreach glperiod where glperiod.start_date <= today's
  Select from gl_hist if there is otherwise select from gl_budget_d
  Retained codes are processed separately.

3.2.2 Columns for the BI_GL_ACCT Table

Name	Value Coming From	Dimension
GL_ACCT_KEY (PK)	gl_chart.org_code '!' gl_chart.gl_code	
ACCOUNT_CLASS_CODE	gl_type.acct_class	
ACCOUNT_CLASS_DESC_1	md_locale_text.lit_text***{locale1}***	**Note 1**
ACCOUNT_CLASS_DESC_2	md_locale_text.lit_text***{locale2}***	**Note 1**
TYPE_CODE	gl_type.type_code	
TYPE_DESC_1	gl_type.desc_1	
TYPE_DESC_2	gl_type.desc_2	
GL_CODE	gl_chart.gl_code	
GL_DESC_1	gl_chart.desc_1	
GL_DESC_2	gl_chart.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Note 1
Select md_locale_text.lit_text from md_locale_text, md_domain_value, md_column
where md_column.database_name = ***'dms'*** and md_column.table_name = ***'gl_type'***
and md_column.column_name = ***'acct_class'***
and md_domain_value.database_name = md_column.database_name
and md_domain_value.domain_name = md_column.domain_name
and md_domain_value.accepted_value = ***bi_gl_acct.account_class_code***
and md_locale_text.lit_key = md_domain_value.desc_key "
and md_locale_text.locale_key = ***{locale1}/{locale2}***

3.2.3 Columns for the BI_GL_BDGT Table

Name	Value Coming From	Dimension
GL_BDGT_KEY (PK)	gl_budget.budget_code '-' gl_budget.rev_no	
BUDGET_CODE	gl_budget.budget_code '-' gl_budget.rev_no	
BUDGET_DESC_1	gl_budget.desc_1 '-'gl_budget.rev_no gl_budget.budget_code '-' gl_budget.rev_no	gl_budget.desc_1 not null gl_budget.desc_1 is null
BUDGET_DESC_2	gl_budget.desc_2 '-'gl_budget.rev_no gl_budget.budget_code '-' gl_budget.rev_no	gl_budget.desc_2 not null gl_budget.desc_2 is null
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Select all records from gl_budget where gl_budget.rev_no < 1000.

3.2.4 Columns for the BI_GL_DRIV Table

Name	Value Coming From	Dimension
GL_DRIV_KEY (PK)	gl_chart.org_code '!' gl_chart.gl_code	
DRIVER_CODE	gl_chart.driver_code	
DRIVER_DESC_1	gl_chart.driver_code	
DRIVER_DESC_2	gl_chart.driver_code	
GL_CODE	gl_chart.gl_code	
GL_DESC_1	gl_chart.desc_1	
GL_DESC_2	gl_chart.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

3.2.5 Columns for the BI_GL_MGRP Table

Name	Value Coming From	Dimension
GL_MGRP_KEY (PK)	gl_chart.org_code '!' gl_chart.gl_code	
ACCOUNT_CLASS_CODE	gl_type.acct_class	
ACCOUNT_CLASS_DESC_1	md_locale_text.lit_text***{locale1}***	**Note 1**
ACCOUNT_CLASS_DESC_2	md_locale_text.lit_text***{locale2}***	**Note 1**
METRIC_GROUP_CODE	gl_chart.metric_group	
METRIC_GROUP_DESC_1	md_locale_text.lit_text***{locale1}***	**Note 2**
METRIC_GROUP_DESC_2	md_locale_text.lit_text***{locale2}***	**Note 2**
GL_CODE	gl_chart.gl_code	
GL_DESC_1	gl_chart.desc_1	
GL_DESC_2	gl_chart.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Note 1

Select md_locale_text.lit_text from md_locale_text, md_domain_value, md_column
where md_column.database_name = ***'dms'*** and md_column.table_name = ***'gl_type'***
and md_column.column_name = ***'acct_class'***
and md_domain_value.database_name = md_column.database_name
and md_domain_value.domain_name = md_column.domain_name
and md_domain_value.accepted_value = ***bi_gl_acct.account_class_code***
and md_locale_text.lit_key = md_domain_value.desc_key "
and md_locale_text.locale_key = ***{locale1}/{locale2}***

Note 2

Select md_locale_text.lit_text from md_locale_text, md_domain_value, md_column
where md_column.database_name = ***'dms'*** and md_column.table_name = ***'gl_chart'***
and md_column.column_name = ***'metric_group'***
and md_domain_value.database_name = md_column.database_name
and md_domain_value.domain_name = md_column.domain_name
and md_domain_value.accepted_value = ***bi_gl_mgrp.metric_group_code***
and md_locale_text.lit_key = md_domain_value.desc_key "
and md_locale_text.locale_key = ***{locale1}/{locale2}***

3.2.6 Columns for the BI_GL_ORGN Table

Name	Value Coming From	Dimension
GL_ORGN_KEY (PK)	org.org_code '!' division.div_code	
ORG_CODE	org.org_code	
ORG_NAME	org.name_1	
DIV_CODE	division.div_code	
DIV_NAME	division.name_1	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

3.2.7 Columns for the BI_GL_STMT Table

Name	Value Coming From	Dimension
GL_STMT_KEY (PK)	org.org_code '!' division.div_code '!' gl_chart.gl_code	
REPORT_CODE	'1' '2'	gl_chart.metric_group< 6000 gl_chart.metric_group>= 6000
REPORT_DESC_1	md_locale_text.lit_text ***{locale1}***	**Note 1**
REPORT_DESC_2	md_locale_text.lit_text ***{locale2}***	**Note 1**
SECTION_CODE	'11' '12' '13' '14' '15' '21' '22' '23' '24' '25'	gl_chart.metric_group< 2000 gl_chart.metric_group< 3000 gl_chart.metric_group< 4000 gl_chart.metric_group< 5000 gl_chart.metric_group< 6000 gl_chart.metric_group< 6000 gl_chart.metric_group< 7000 (gl_chart.metric_group>=8000 & gl_chart.metric_group<=8003) || gl_chart.metric_group=8006 gl_chart.metric_group>=6001 & gl_chart.metric_group<=6002 otherwise
SECTION_DESC_1	md_locale_text.lit_text ***{locale1}***	**Note 1**
SECTION_DESC_2	md_locale_text.lit_text ***{locale2}***	**Note 1**

Name	Value Coming From	Dimension
SUBSECTION_CODE	gl_chart.metric_group	
SUBSECTION_DESC_1	md_locale_text.lit_text ***{locale1}***	**Note 2**
SUBSECTION_DESC_2	md_locale_text.lit_text ***{locale2}***	**Note 2**
DETAIL_CODE	org.org_code ‘ ’ division.div_code ‘ ’ gl_chart.gl_code	
ORG_CODE	org.org_code	
ORG_NAME	org.org_name	
DIV_CODE	division.div_code	
DIV_NAME	division.name_1	
GL_CODE	gl_chart.gl_code	
GL_DESC_1	gl_chart.desc_1	
GL_DESC_2	gl_chart.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Note 1

Select md_locale_text.lit_text desc_1 from md_locale_text where

where md_locale_text.lit_key = ***‘balance_sheet’*** when bi_gl_stmt.report_code = ‘1’
md_locale_text.lit_key = ***‘income_statement’*** when bi_gl_stmt.report_code = ‘2’
md_locale_text.lit_key = ***‘current_asset’*** when bi_gl_stmt.section_code = ‘11’
md_locale_text.lit_key = ***‘non_current_asset’*** when bi_gl_stmt.section_code = ‘12’
md_locale_text.lit_key = ***‘current_liability’*** when bi_gl_stmt.section_code = ‘13’
md_locale_text.lit_key = ***‘long_term_liability’*** when bi_gl_stmt.section_code = ‘14’
md_locale_text.lit_key = ***‘owner_equity’*** when bi_gl_stmt.section_code = ‘15’
md_locale_text.lit_key = ***‘revenue’*** when bi_gl_stmt.section_code = ‘21’
md_locale_text.lit_key = ***‘cost_goods_sold’*** when bi_gl_stmt.section_code = ‘22’
md_locale_text.lit_key = ***‘expense’*** when bi_gl_stmt.section_code = ‘23’
md_locale_text.lit_key = ***‘other_revenue’*** when bi_gl_stmt.section_code = ‘24’
md_locale_text.lit_key = ***‘other_expense’*** when bi_gl_stmt.section_code = ‘25’
and md_locale_text.locale_key = ***{locale1}/{locale2}***

Note 2

Select md_locale_text.lit_text from md_locale_text, md_domain_value, md_column
where md_column.database_name = ***'dms'*** and md_column.table_name = ***'gl_chart'***
and md_column.column_name = ***'metric_group'***
and md_domain_value.database_name = md_column.database_name
and md_domain_value.domain_name = md_column.domain_name
and md_domain_value.accepted_value = ***bi_gl_mgrp.metric_group_code***
and md_locale_text.lit_key = md_domain_value.desc_key "
and md_locale_text.locale_key = ***{locale1}/{locale2}***

3.2.8 Columns for the BI_GL_TIME Table

Name	Value Coming From	Dimension
GL_TIME_KEY (PK)	glperiod.yr '-' glperiod.period	[YYYY-PP] Select all records from glperiod where glperiod.start_date <= today.
GL_YEAR	glperiod.yr	
GL_YEAR_CHAR	glperiod.yr	
GL_YEAR_TO_DATE	'1' '2'	GL_YEAR = glperiod.yr - 1 & GL_PERIOD <= glperiod.period GL_YEAR = glperiod.yr & GL_PERIOD <= glperiod.period
GL_YEAR_TO_DATE_DESC_1	'Last Year-To-Date' 'Year-To-Date'	GL_YEAR = glperiod.yr - 1 & GL_PERIOD <= glperiod.period GL_YEAR = glperiod.yr & GL_PERIOD <= glperiod.period
GL_YEAR_TO_DATE_DESC_2	'Last Year-To-Date' 'Year-To-Date'	GL_YEAR = glperiod.yr - 1 & GL_PERIOD <= glperiod.period GL_YEAR = glperiod.yr & GL_PERIOD <= glperiod.period
GL_QUARTER	'1' '2' '3' '4'	glperiod.period < 4 glperiod.period >= 4 and < 7 glperiod.period >=7 and < 10 glperiod.period >= 10
GL_QUARTER_CHAR	glperiod.yr '-' GL_QUARTER	[YYYY-Q]

Name	Value Coming From	Dimension
GL_QUARTER_TO_DATE	'1' '2' '3'	GL_YEAR = glperiod.yr - 1 & GL_QUARTER = current GL_QUARTER & GL_PERIOD <= glperiod.period GL_YEAR = glperiod.yr & GL_QUARTER = previous GL_QUARTER & GL_PERIOD <= previous GL_QUARTER period GL_YEAR = glperiod.yr & GL_QUARTER = current GL_QUARTER & GL_PERIOD <= previous period
GL_QUARTER_TO_DATE_DESC_1	'Last Year Quarter-To-Date' 'Last Quarter-To-Date' 'Quarter-To-Date'	GL_YEAR = glperiod.yr - 1 & GL_QUARTER = current GL_QUARTER & GL_PERIOD <= glperiod.period GL_YEAR = glperiod.yr & GL_QUARTER = previous GL_QUARTER & GL_PERIOD <= previous GL_QUARTER period GL_YEAR = glperiod.yr & GL_QUARTER = current GL_QUARTER GL_PERIOD <= glperiod.period
GL_QUARTER_TO_DATE_DESC_2	'Last Year Quarter-To-Date' 'Last Quarter-To-Date' 'Quarter-To-Date'	GL_YEAR = glperiod.yr - 1 & GL_QUARTER = current GL_QUARTER & GL_PERIOD <= glperiod.period GL_YEAR = glperiod.yr & GL_QUARTER = previous GL_QUARTER & GL_PERIOD <= previous GL_QUARTER period GL_YEAR = glperiod.yr & GL_QUARTER = current GL_QUARTER GL_PERIOD <= glperiod.period
GL_PERIOD	glperiod.period	
GL_PERIOD_CHAR	glperiod.yr '-' glperiod.period	[YYYY-PP]
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

3.2.9 DMS General Ledger Measures Definition

3.2.9.1 bi_gl_hist Fact Table

Name	Value	ETL/Aggregate
	1- from gl_chart, org_gl where gl_chart.org_code = org_gl.org_code and gl_chart.gl_code != org_gl.retained_code order by org_code, gl_code 2- from gl_hist where org_code = *{gl_chart.org_code}* and gl_code = *{gl_chart.gl_code}* group by div_code union from gl_budget_d where org_code = *{gl_chart.org_code}* and gl_code = *{gl_chart.gl_code}* group by div_code; 3- from glperiod where glperiod.org_code = *{gl_chart.org_code}* and glperiod.start_date <= *{today's date}* order by glperiod.yr, glperiod.period 4- from gl_hist where org_code = *{gl_chart.org_code}* and gl_code = *{gl_chart.gl_code}* and div_code = *{gl_hist.div_code or gl_budget_d.div_code}* and yr = *{glperiod.yr}* and period = *{glperiod.period}* or from gl_budget_d where org_code = *{gl_chart.org_code}* and gl_code = *{gl_chart.gl_code}* and div_code = *{gl_hist.div_code or gl_budget_d.div_code}* and yr = *{glperiod.yr}* and period = *{glperiod.period}*	
External Reference	null	ETL
Account Class Code	gl_type.acct_class from gl_chart join gl_type on gl_chart.type_code = gl_type.type_code	ETL
Positive Value Multiplier	gl_chart.dr_cr_mult	ETL/Sum
Debit Amount	gl_hist.debit_amt	ETL/Sum
Credit Amount	gl_hist.credit_amt	ETL/Sum

Name	Value	ETL/Aggregate
Opening Balance	when glperiod.period = 1 select debit_amt, credit_amt, net_chg_amt from gl_hist where org_code = *{gl_chart.org_code}* and gl_code = *{gl_chart.gl_code}* and div_code = *{gl_hist.div_code or* *gl_budget_d.div_code}* and yr = *{glperiod.yr}* and period = 0 otherwise Closing Balance of previous year, period	ETL/Sum
Net Change Amount	gl_hist.net_chg_amt	ETL/Sum
Closing Balance	Opening Balance(period 1) + foreach period cumulate gl_hist.net_chg_amt	ETL/Sum
Budget Amount	gl_budget_d.budget_amt where org_code = *{gl_chart.org_code}* and gl_code = *{gl_chart.gl_code}* and div_code = *{gl_hist.div_code or* *gl_budget_d.div_code}* and yr = *{glperiod.yr}* and period = *{glperiod.period}* and budget_code = *{gl_budget.budget_code}* and rev_no = *{gl_budget.rev_no}*	ETL/Sum
Year-To-Date Budget Amount	Cumulate Budget Amount	
Opening Balance (Positive)	Positive Value Multiplier * Opening Balance	ETL/Sum
Net Change Amount (Positive)	Relational Model: Positive Value Multiplier * Net Change Amount	ETL/Sum
Net Change Amount (Positive)	Dimensional Model: Case When Account Class Code in ('7','8') Then Net Change Amount When Account Class Code in ('6') and Positive Value Multiplier = 1 Then Net Change Amount * -1 Else Net Change Amount (Positive)	Sum
Closing Balance (Positive)	Relational Model: Positive Value Multiplier * Closing Balance	ETL/Sum

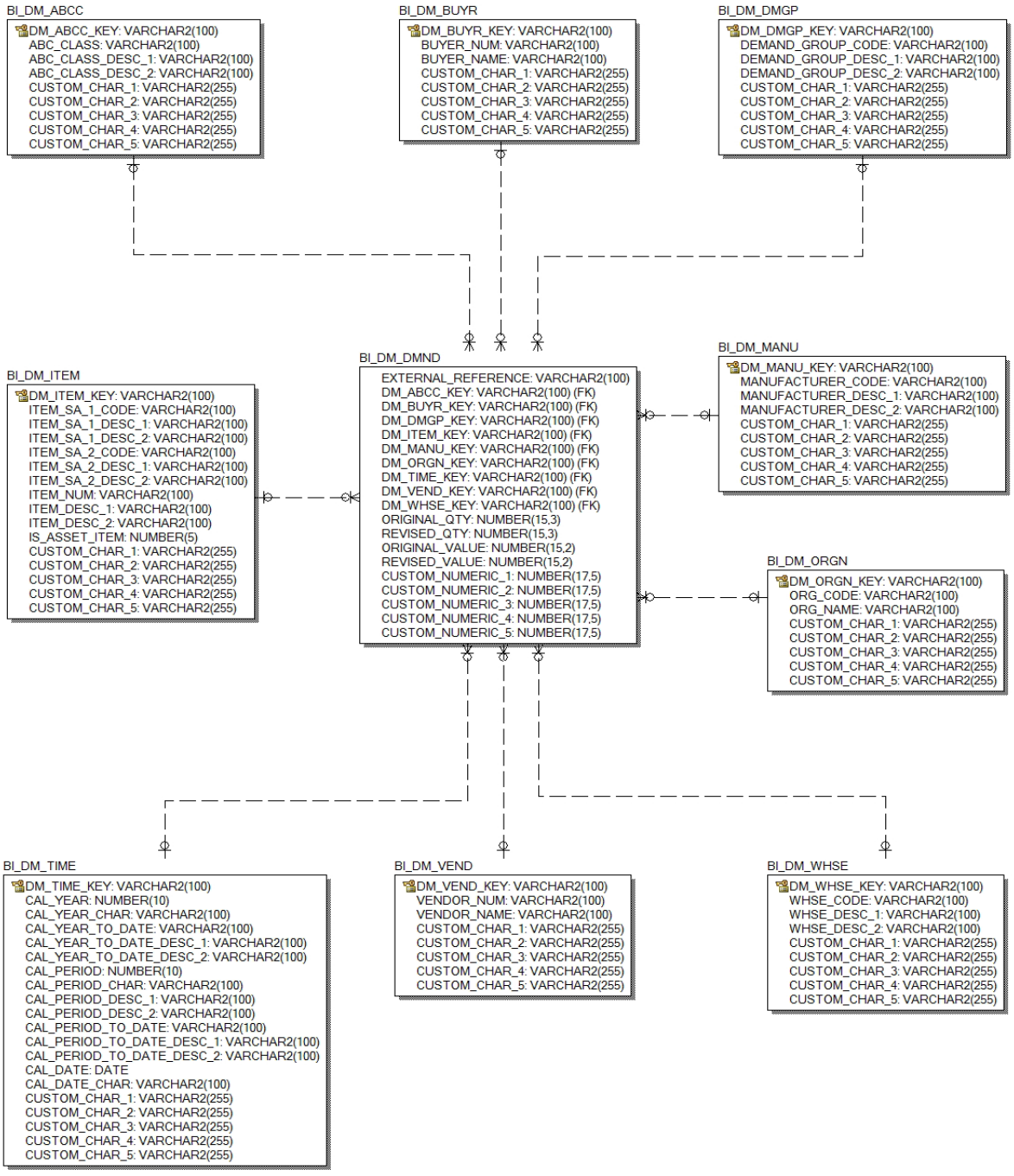
Name	Value	ETL/Aggregate
Closing Balance (Positive)	Dimensional Model: Case When Account Class Code in ('1','2') Then Closing Balance When Account Class Code in ('3','4','5') and Positive Value Multiplier = 1 Then Closing Balance * -1 Else Closing Balance (Positive)	Sum
Variance Amount	Net Change Amount (Positive) - Budget Amount	ETL/Sum

3.2.10 DMS General Ledger Dimension Names

- Time
- Account
- Organization
- Metric Group
- Driver
- Financial Statement
- Budget

4 DMS Item Demand

4.1 Structure - Demand



4.2 Mapping

4.2.1 Columns for the BI_DM_DMND Table

Name	Value Coming From	Fact
EXTERNAL_REFERENCE	item_w.whse_code '!' item_w_demand.item_num '!' item_w_demand.demand_date '!' item_w_demand.cust_fcst_grp	
DM_ABCC_KEY	item_w.abc_class	
DM_BUYR_KEY	item_w.buyer_num	
DM_DMGP_KEY	item_w_demand.cust_fcst_grp	
DM_ITEM_KEY	item_w_demand.item_num	
DM_MANU_KEY	item.manu_code	
DM_ORGN_KEY	item_w.org_code	
DM_TIME_KEY	item_w_demand.demand_date	[YYYY-MM-DD]
DM_VEND_KEY	null item_w_demand.item_num '!' item_w.whse_code	item_w.repl_code is null item_w.repl_code not null
DM_WHSE_KEY	item_w.whse_code	
ORIGINAL_QTY	item_w_demand.original_qty	
REVISED_QTY	item_w_demand.revised_qty	
ORIGINAL_VALUE	item_w_demand.original_qty * item_w.last_cost	
REVISED_VALUE	item_w_demand.revised_qty * item_w.last_cost	
CUSTOM_NUMERIC_1		
CUSTOM_NUMERIC_2		
CUSTOM_NUMERIC_3		
CUSTOM_NUMERIC_4		
CUSTOM_NUMERIC_5		

Read records from item_w_demand, item, item_w where
item_w_demand.mod_stamp > (bi_dmr_att.dmr_att_value where dmr_name =
'item_demand' and dmr_att_key = 'bi_id_dmnd!last_datetime_processed) and mod_stamp <= Today
or
item_w_demand.create_stamp > (bi_dmr_att.dmr_att_value where dmr_name = 'item_demand' and dmr_att_key =
'bi_id_dmnd!last_datetime_processed``) and create_stamp<= Today

4.2.2 Columns for the BI_DM_ABCC Table

Name	Value Coming From	Dimension
DM_ABCC_KEY (PK)	abc_class.abc_class	
ABC_CLASS	abc_class.abc_class	
ABC_CLASS_DESC_1	abc_class.abc_class	
ABC_CLASS_DESC_2		
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

4.2.3 Columns for the BI_DM_BUYR Table

Name	Value Coming From	Dimension
DM_BUYR_KEY (PK)	buyer.buyer_num	
BUYER_NUM	buyer.buyer_num	
BUYER_NAME	md_user.first_name ' ' md_user.last_name	select md_user.first_name, md_user.last_name from md_user where user_name = ?
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

4.2.4 Columns for the BI_DM_DMGPTable

Name	Value Coming From	Dimension
DM_DMGP_KEY (PK)	cust_fcst_grp.cust_fcst_grp	
DEMAND_GROUP_CODE	cust_fcst_grp.cust_fcst_grp	
DEMAND_GROUP_DESC_1	cust_fcst_grp.desc_1	
DEMAND_GROUP_DESC_2	cust_fcst_grp.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

4.2.5 Columns for the BI_DM_ITEM Table

Name	Value Coming From	Fact
DM_ITEM_KEY (PK)	item.item_num	
ITEM_SA_1_CODE	item.sa_item	
ITEM_SA_1_DESC_1	item_sa.desc_1 item.sa_item	where item_sa.item_num = item.sa_item when item_sa.desc_1 is null
ITEM_SA_1_DESC_2	item_sa.desc_2	where item_sa.item_num = item.sa_item
ITEM_SA_2_CODE	item.sa_item2	
ITEM_SA_2_DESC_1	item_sa.desc_1 item.sa_item2	where item_sa.item_num = item.sa_item2 when item_sa.desc_1 is null
ITEM_SA_2_DESC_2	item_sa.desc_2	where item_sa.item_num = item.sa_item2
ITEM_NUM	item.item_num	
ITEM_DESC_1	item.desc_1	
ITEM_DESC_2	item.desc_2	
IS_ASSET_ITEM	item.asset	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		

Name	Value Coming From	Fact
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

4.2.6 Columns for the BI_DM_MANU Table

Name	Value Coming From	Dimension
DM_MANU_KEY (PK)	manu.manu_code	
MANUFACTURER_CODE	manu.manu_code	
MANUFACTURER_DESC_1	manu.desc_1	
MANUFACTURER_DESC_2	manu.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

4.2.7 Columns for the BI_DM_ORGN Table

Name	Value Coming From	Dimension
DM_ORGN_KEY (PK)	org.org_code	
ORG_CODE	org.org_code	
ORG_NAME	org.name_1	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

4.2.8 Columns for the BI_DM_TIME Table

Name	Value Coming From	Dimension
DM_TIME_KEY (PK)	st_cal_p_date.cal_date	[YYYY-MM-DD]
CAL_YEAR	st_cal_p_date.cal_yr	
CAL_YEAR_CHAR	st_cal_p_date.cal_yr	
CAL_YEAR_TO_DATE	'1' '2'	CAL_YEAR = st_cal_p_date.cal_yr - 1 & CAL_DATE <= last year TODAY CAL_YEAR = st_cal_p_date.cal_yr & CAL_DATE <= TODAY
CAL_YEAR_TO_DATE_DESC_1	'Last Year-To-Date' 'Year-To-Date'	CAL_YEAR = st_cal_p_date.cal_yr - 1 & CAL_DATE <= last year TODAY CAL_YEAR = st_cal_p_date.cal_yr & CAL_DATE <= TODAY
CAL_YEAR_TO_DATE_DESC_2	'Last Year-To-Date' 'Year-To-Date'	CAL_YEAR = st_cal_p_date.cal_yr - 1 & CAL_DATE <= last year TODAY CAL_YEAR = st_cal_p_date.cal_yr & CAL_DATE <= TODAY
CAL_PERIOD	st_cal_p_date.cal_period	
CAL_PERIOD_CHAR	st_cal_p_date.cal_period	
CAL_PERIOD_DESC_1	st_cal_p_date.period_desc_1	
CAL_PERIOD_DESC_2	st_cal_p_date.period_desc_2	
CAL_PERIOD_TO_DATE	'1' '2' '3'	CAL_YEAR = st_cal_p_date.cal_yr - 1 & CAL_PERIOD = st_cal_p_date.cal_period & CAL_DATE <= last year TODAY CAL_YEAR = st_cal_p_date.cal_yr & CAL_PERIOD = st_cal_p_date.cal_period - 1& CAL_DATE <= last period end date CAL_YEAR = st_cal_p_date.cal_yr & CAL_PERIOD = st_cal_p_date.cal_period & CAL_DATE <= TODAY
CAL_PERIOD_TO_DATE_DESC_1	'Last Year Period-To-Date' 'Last Period-To-Date' 'Period-To-Date'	CAL_YEAR = st_cal_p_date.cal_yr - 1 & CAL_PERIOD = st_cal_p_date.cal_period & CAL_DATE <= last year TODAY CAL_YEAR = st_cal_p_date.cal_yr & CAL_PERIOD = st_cal_p_date.cal_period - 1& CAL_DATE <= last period end date CAL_YEAR = st_cal_p_date.cal_yr & CAL_PERIOD = st_cal_p_date.cal_period & CAL_DATE <= TODAY

Name	Value Coming From	Dimension
CAL_PERIOD_TO_DATE_DESC_2	'Last Year Period-To-Date' 'Last Period-To-Date' 'Period-To-Date'	CAL_YEAR = st_cal_p_date.cal_yr - 1 & CAL_PERIOD = st_cal_p_date.cal_period & CAL_DATE <= last year TODAY CAL_YEAR = st_cal_p_date.cal_yr & CAL_PERIOD = st_cal_p_date.cal_period - 1& CAL_DATE <= last period end date CAL_YEAR = st_cal_p_date.cal_yr & CAL_PERIOD = st_cal_p_date.cal_period & CAL_DATE <= TODAY
CAL_DATE	st_cal_p_date.cal_date	[YYYY-MM-DD]
CAL_DATE_CHAR	st_cal_p_date.cal_date	[YYYY-MM-DD]
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

4.2.9 Columns for the BI_DM_VEND Table

Name	Value Coming From	Dimension
DM_VEND_KEY (PK)	item_w.item_num '!' item_w.whse_code	
VENDOR_NUM	item_w_p.repl_path	**Note 1**
VENDOR_NAME	vendor.vend_name	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Select from item_w where item_w.repl_code is not null.

Note 1

Select the item_w_p.repl_type, item_w_p.repl_path using

if item_w_p.repl_type is 1 use *{item_w_p.repl_path}* as VENDOR_NUM

else

select item_w_p.repl_type, *{item_w_p.repl_path}* and use the new item_w_p.repl_path value as VENDOR_NUM

4.2.10 Columns for the BI_DM_WHSE Table

Name	Value Coming From	Dimension
DM_WHSE_KEY (PK)	whse.whse_code	
WHSE_CODE	whse.whse_code	
WHSE_DESC_1	whse.desc_1	
WHSE_DESC_2	whse.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

4.2.11 DMS Item Demand - Demand Measures Definition

4.2.11.1 bi_id_dmnd Fact Table

Name	Value	ETL/Aggregate

	from item_w_demand
	  join item on item_w_demand.item_num = item.item_num
	  join item_w on item_w_demand.item_num = item_w.item_num
	  and item_w_demand.whse_code = item_w.whse_code
	where (item_w_demand.mod_stamp > *{last datetime processed}*
	  and item_w_demand.mod_stamp <= *{today's date}*)
	  or (item_w_demand.create_stamp > *{last datetime processed}*
	  and item_w_demand.create_stamp <= *{today's date}*)	

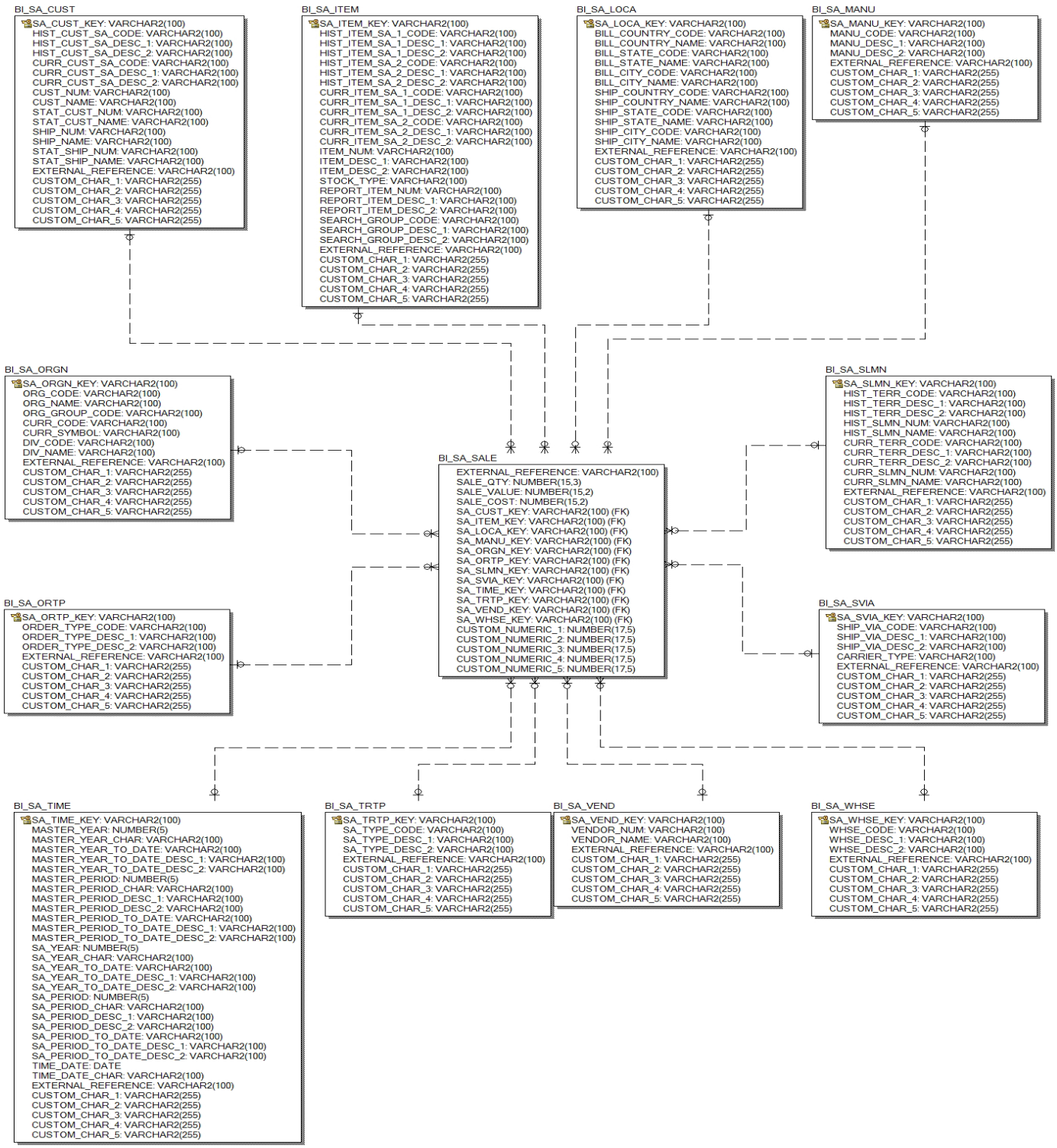
External Reference	item_w.whse_code + "!" + item_w_demand.item_num + "!" + item_w_demand.demand_date + "!" + item_w_demand.cust_fcst_grp	ETL
Original Quantity	item_w_demand.original_qty	ETL/Sum
Revised Quantity	item_w_demand.revised_qty	ETL/Sum
Original Value (Last Cost)	Original Quantity * item_w.last_cost	ETL/Sum
Revised Value (Last Cost)	Revised Quantity * item_w.last_cost	ETL/Sum

4.2.12 DMS Item Demand Dimensions Name

- Time
- Item
- Warehouse
- Item Demand Group
- Vendor
- Manufacturer
- Organization
- ABC Class
- Buyer

5 DMS Sales Analysis

5.1 Structure



5.2 Mapping

5.2.1 Columns for the BI_SA_SALE Table

Name	Value Coming From	Fact
EXTERNAL_REFERENCE	sa_sale.sa_sale_id	
SA_CUST_KEY	sa.sa_cust '!' sa.cust_num '!' sa.sa_cust '!' sa.cust_num '!' sa.ship_num	when sa.ship_num is null when sa.ship_num is not null
SA_ITEM_KEY	sa.sa_item '!' sa.sa_item2 '!' sa.item_num '!' sa.vendor_num '!' sa.stock_type	
SA_LOCA_KEY	sa.cust_num '!' sa.cust_num '!' sa.ship_num	when sa.ship_num is null when sa.ship_num is not null
SA_MANU_KEY	item.manu_code	
SA_ORGN_KEY	sa.org_code '!'sa.div_code	
SA_ORTP_KEY	sa.order_type_code	
SA_SLMN_KEY	inv_shp_l.slmn_num customer.slmn_num cust_shp.slmn_num	inv_shp_l.slmn_num is null & sa.ship_num is null inv_shp_l.slmn_num is null & sa.ship_num is not null
SA_SVIA_KEY	inv_shp.ship_via_code	
SA_TIME_KEY	sa_sale.inv_date	[YYYY-MM-DD]
SA_TRTP_KEY	sa_sale.sa_type	
SA_VEND_KEY	sa.vendor_num	
SA_WHSE_KEY	sa.whse_code	
SALE_QTY	sa_sale.sale_qty	
SALE_VALUE	sa_sale.sale_value	
SALE_COST	sa_sale.sale_cost	
CUSTOM_NUMERIC_1		
CUSTOM_NUMERIC_2		
CUSTOM_NUMERIC_3		
CUSTOM_NUMERIC_4		
CUSTOM_NUMERIC_5		

Read records from sa, sa_sale where sa.sa_id > (bi_dmr_att.dmr_att_value where dmr_name = 'sales-analysis' and dmr_att_key = 'last_sale_id') and sa.sa_id <= max(sa.sa_id)

5.2.2 Columns for the BI_SA_CUST Table

Name	Value Coming From	Dimension
SA_CUST_KEY (PK)	sa.sa_cust '!' sa.cust_num '!' sa.sa_cust '!' sa.cust_num '!' sa.ship_num	when sa.ship_num is null when sa.ship_num is not null
HIST_CUST_SA_CODE	sa.sa_cust	
HIST_CUST_SA_DESC_1	cust_sa.desc_1	
HIST_CUST_SA_DESC_2	cust_sa.desc_2	
CURR_CUST_SA_CODE	customer.sa_cust	when sa.ship_num is null when sa.ship_num is not null
CURR_CUST_SA_DESC_1	cust_shp.sa_cust	
CURR_CUST_SA_DESC_2	cust_sa.desc_1	
CUST_NUM	cust_sa.desc_2	
CUST_NAME	sa.cust_num	
STAT_CUST_NUM	customer.cust_name	
STAT_CUST_NAME	customer.stat_cust_num	using customer.stat_cust_num
SHIP_NUM	customer.cust_name	
SHIP_NAME	sa.ship_num	
STAT_SHIP_NUM	cust_shp.ship_name	
STAT_SHIP_NAME	cust_shp.stat_ship_num	using cust_shp.stat_ship_num
EXTERNAL_REFERENCE	cust_shp.ship_name	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Read records from sa where sa.sa_id > (bi_dmr_att.dmr_att_value where dmr_name = 'sales-analysis' and dmr_att_key = 'last_sale_id') and sa.sa_id <= max(sa.sa_id)

5.2.3 Columns for the BI_SA_ITEM Table

Name	Value Coming From	Dimension
SA_ITEM_KEY (PK)	sa.sa_item '!' sa.sa_item2 '!' sa.item_num '!' sa.vendor_num '!' sa.stock_type	
HIST_ITEM_SA_1_CODE	sa.sa_item	
HIST_ITEM_SA_1_DESC_1	item_sa.desc_1 sa.sa_item	when item_sa.desc_1 is null
HIST_ITEM_SA_1_DESC_2	item_sa.desc_2	
HIST_ITEM_SA_2_CODE	sa.sa_item2	
HIST_ITEM_SA_2_DESC_1	item_sa.desc_1 sa.sa_item2	when item_sa.desc_1 is null
HIST_ITEM_SA_2_DESC_2	item_sa.desc_2	
CURR_ITEM_SA_1_CODE	item.sa_item	
CURR_ITEM_SA_1_DESC_1	item_sa.desc_1	
CURR_ITEM_SA_1_DESC_2	item_sa.desc_2	
CURR_ITEM_SA_2_CODE	item.sa_item2	
CURR_ITEM_SA_2_DESC_1	item_sa.desc_1	
CURR_ITEM_SA_2_DESC_2	item_sa.desc_2	
ITEM_NUM	sa.item_num	
ITEM_DESC_1	item.desc_1	
ITEM_DESC_2	item.desc_2	
STOCK_TYPE	sa.stock_type	
REPORT_ITEM_NUM	item.report_item_num	
REPORT_ITEM_DESC_1	item.desc_1	using item.report_item_num
REPORT_ITEM_DESC_2	item.desc_2	using item.report_item_num
SEARCH_GROUP_CODE	item_srch_grp.search_group	
SEARCH_GROUP_DESC_1	item_srch_grp.desc_1	
SEARCH_GROUP_DESC_2	item_srch_grp.desc_2	
EXTERNAL_REFERENCE	sa.stock_type '!' sa.item_num '!' sa.vendor_num	
CUSTOM_CHAR_1		

Name	Value Coming From	Dimension
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Read records from sa where sa.sa_id > (bi_dmr_att.dmr_att_value where dmr_name = 'sales-analysis' and dmr_att_key = 'last_sale_id') and sa.sa_id <= max(sa.sa_id)

5.2.4 Columns for the BI_SA_LOCA Table

Name	Value Coming From	Dimension
SA_LOCA_KEY (PK)	customer.cust_num '!' cust_shp.cust_num '!' cust_shp.ship_num	read from customer and cust_shp
BILL_COUNTRY_CODE	customer.country	
BILL_COUNTRY_NAME	customer.country	
BILL_STATE_CODE	customer.province	
BILL_STATE_NAME	customer.province	
BILL_CITY_CODE	customer.city	
BILL_CITY_NAME	customer.city	
SHIP_COUNTRY_CODE	customer.country cust_shp.country	
SHIP_COUNTRY_NAME	customer.country cust_shp.country	
SHIP_STATE_CODE	customer.province cust_shp.province	
SHIP_STATE_NAME	customer.province cust_shp.province	
SHIP_CITY_CODE	customer.city cust_shp.city	
SHIP_CITY_NAME	customer.city cust_shp.city	
EXTERNAL_REFERENCE	null	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		

Name	Value Coming From	Dimension
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

5.2.5 Columns for the BI_SA_MANU Table

Name	Value Coming From	Dimension
SA_MANU_KEY (PK)	manu.manu_code	
MANU_CODE	manu.manu_code	
MANU_DESC_1	manu.desc_1	
MANU_DESC_2	manu.desc_2	
EXTERNAL_REFERENCE	null	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

5.2.6 Columns for the BI_SA_ORGN Table

Name	Value Coming From	Dimension
SA_ORGN_KEY (PK)	org.org_code '!' division.div_code	
ORG_CODE	org.org_code	
ORG_NAME	org.name_1	
ORG_GROUP_CODE	org.org_group_code	
CURR_CODE	org_group.curr_code	
CURR_SYMBOL	curr.symbol	
DIV_CODE	division.div_code	
DIV_NAME	division.name_1	
EXTERNAL_REFERENCE	null	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		

Name	Value Coming From	Dimension
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

5.2.7 Columns for the BI_SA_ORTP Table

Name	Value Coming From	Dimension
SA_ORTP_KEY (PK)	order_type.order_type_code	
ORDER_TYPE_CODE	order_type.order_type_code	
ORDER_TYPE_DESC_1	order_type.desc_1	
ORDER_TYPE_DESC_2	order_type.desc_2	
EXTERNAL_REFERENCE	null	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

5.2.8 Columns for the BI_SA_SLMN Table

Name	Value Coming From	Dimension
SA_SLMN_KEY (PK)	sa.cust_num '!' sa.ship_num	
HIST_TERR_CODE	sa.terr_code	
HIST_TERR_DESC_1	terr.desc_1	
HIST_TERR_DESC_2	terr.desc_2	
HIST_SLMN_NUM	inv_shp_l.slmn_num	
HIST_SLMN_NAME	slmn.slmn_name	
CURR_TERR_CODE	slmn.terr_code	
CURR_TERR_DESC_1	terr.desc_1	using slmn.terr_code
CURR_TERR_DESC_2	terr.desc_2	using slmn.terr_code

Name	Value Coming From	Dimension
CURR_SLMN_NUM	customer.slmn_num cust_shp.slmn_num	read from customer and cust_shp
CURR_SLMN_NAME	slmn.slmn_name	
EXTERNAL_REFERENCE	NULL	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Read records from sa where sa.sa_id > (bi_dmr_att.dmr_att_value where dmr_name = 'sales-analysis' and dmr_att_key = 'last_sale_id') and sa.sa_id <= max(sa.sa_id)

5.2.9 Columns for the BI_SA_SVIA Table

Name	Value Coming From	Dimension
SA_SVIA_KEY (PK)	ship_via.ship_via_code	
SHIP_VIA_CODE	ship_via.ship_via_code	
SHIP_VIA_DESC_1	ship_via.desc_1	
SHIP_VIA_DESC_2	ship_via.desc_2	
CARRIER_TYPE	ship_via.carrier_type	
EXTERNAL_REFERENCE	null	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

5.2.10 Columns for the BI_SA_TIME Table

Name	Value Coming From	Dimension
SA_TIME_KEY (PK)	st_cal_p_date.cal_date	[YYYY-MM-DD] using global.master_cal_code

Name	Value Coming From	Dimension
MASTER_YEAR	st_cal_p_date.cal_yr	using global.master_cal_code
MASTER_YEAR_CHAR	st_cal_p_date.cal_yr	using global.master_cal_code
MASTER_YEAR_TO_DATE	'1' '2'	master_yr = st_cal_p_date.cal_yr - 1 & time_date <= last year today's master_yr = st_cal_p_date.cal_yr & time_date <= today's
MASTER_YEAR_TO_DATE_DESC_1	'Master Calendar Last Year-To-Date' 'Master Calendar Year-To-Date'	master_year = st_cal_p_date.cal_yr - 1 & time_date <= last year today's master_year = st_cal_p_date.cal_yr & time_date <= today's
MASTER_YEAR_TO_DATE_DESC_2	'Master Calendar Last Year-To-Date' 'Master Calendar Year-To-Date'	master_year = st_cal_p_date.cal_yr - 1 & time_date <= last year today's master_year = st_cal_p_date.cal_yr & time_date <= today's
MASTER_PERIOD	st_cal_p_date.cal_period	using global.master_cal_code
MASTER_PERIOD_CHAR	st_cal_p_date.cal_period	using global.master_cal_code
MASTER_PERIOD_DESC_1	st_cal_p_desc.period_desc_1	using global.master_cal_code
MASTER_PERIOD_DESC_2	st_cal_p_desc.period_desc_2	using global.master_cal_code
MASTER_PERIOD_TO_DATE	'1' '2' '3'	master_year = st_cal_p_date.cal_yr - 1 & master_period = current master_period & time_date <= last year today's master_year = st_cal_p_date.cal_yr & master_period = current master_period -1 & time_date <= end date for master_period -1 master_year = st_cal_p_date.cal_yr & master_period = current master_period & time_date <= today's
MASTER_PERIOD_TO_DATE_DESC_1	'Master Calendar Last Year Period-To-Date' 'Master Calendar Last Period-To-Date' 'Master Calendar Period-To-Date'	master_year = st_cal_p_date.cal_yr - 1 & master_period = current master_period & time_date <= last year today's master_year = st_cal_p_date.cal_yr & master_period = current master_period -1 & time_date <= end date for master_period -1 master_year = st_cal_p_date.cal_yr & master_period = current master_period & time_date <= today's
MASTER_PERIOD_TO_DATE_DESC_2	'Master Calendar Last Year Period-To-Date' 'Master Calendar Last Period-To-Date' 'Master Calendar Period-To-Date'	master_year = st_cal_p_date.cal_yr - 1 & master_period = current master_period & time_date <= last year today's master_year = st_cal_p_date.cal_yr & master_period = current master_period -1 & time_date <= end date for master_period -1 master_year = st_cal_p_date.cal_yr & master_period = current master_period & time_date <= today's
SA_YEAR	st_cal_p_date.cal_yr	using global.sa_cal_code

Name	Value Coming From	Dimension
SA_YEAR_CHAR	st_cal_p_date.cal_yr	using global.sa_cal_code
SA_YEAR_TO_DATE	'1' '2'	sa_year = st_cal_p_date.cal_yr - 1 & time_date <= last year today's sa_year = st_cal_p_date.cal_yr & time_date <= today's
SA_YEAR_TO_DATE_DESC_1	'S/A Calendar Last Year-To-Date' 'S/A Calendar Year-To-Date'	sa_year = st_cal_p_date.cal_yr - 1 & time_date <= last year today's sa_year = st_cal_p_date.cal_yr & time_date <= today's
SA_YEAR_TO_DATE_DESC_2	'S/A Calendar Last Year-To-Date' 'S/A Calendar Year-To-Date'	sa_year = st_cal_p_date.cal_yr - 1 & time_date <= last year today's sa_year = st_cal_p_date.cal_yr & time_date <= today's
SA_PERIOD	st_cal_p_date.cal_period	using global.sa_cal_code
SA_PERIOD_CHAR	st_cal_p_date.cal_period	using global.sa_cal_code
SA_PERIOD_DESC_1	st_cal_p_desc.period_desc_1	using global.sa_cal_code
SA_PERIOD_DESC_2	st_cal_p_desc.period_desc_2	using global.sa_cal_code
SA_PERIOD_TO_DATE	'1' '2' '3'	sa_year = st_cal_p_date.cal_yr - 1 & sa_period = current sa_period & time_date <= last year today's sa_year = st_cal_p_date.cal_yr & sa_period = current sa_period -1 & time_date <= end date for sa_period -1 sa_year = st_cal_p_date.cal_yr & sa_period = current sa_period & time_date <= today's
SA_PERIOD_TO_DATE_DESC_1	'S/A Calendar Last Year Period-To-Date' 'S/A Calendar Last Period-To-Date' 'S/A Calendar Period-To-Date'	sa_year = st_cal_p_date.cal_yr - 1 & sa_period = current sa_period & time_date <= last year today's sa_year = st_cal_p_date.cal_yr & sa_period = current sa_period -1 & time_date <= end date for sa_period -1 sa_year = st_cal_p_date.cal_yr & sa_period = current sa_period & time_date <= today's
SA_PERIOD_TO_DATE_DESC_2	'S/A Calendar Last Year Period-To-Date' 'S/A Calendar Last Period-To-Date' 'S/A Calendar Period-To-Date'	sa_year = st_cal_p_date.cal_yr - 1 & sa_period = current sa_period & time_date <= last year today's sa_year = st_cal_p_date.cal_yr & sa_period = current sa_period -1 & time_date <= end date for sa_period -1 sa_year = st_cal_p_date.cal_yr & sa_period = current sa_period & time_date <= today's
TIME_DATE	st_cal_p_date.cal_date	[YYYY-MM-DD] using global.master_cal_code

Name	Value Coming From	Dimension
TIME_DATE_CHAR	st_cal_p_date.cal_date	[YYYY-MM-DD] using global.master_cal_code
EXTERNAL_REFERENCE	null	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

5.2.11 Columns for the BI_SA_TRTP Table

Name	Value Coming From	Dimension
SA_TRTP_KEY (PK)	md_domain_value.accepted_value	where md_column.database_name = ‘dms’ & md_column.table_name = ‘sa_sale’ & md_column.column_name = ‘sa_type’
SA_TYPE_CODE	md_domain_value.accepted_value	where md_column.database_name = ‘dms’ & md_column.table_name = ‘sa_sale’ & md_column.column_name = ‘sa_type’
SA_TYPE_DESC_1	md_locale_text.lit_text***{locale1}***	using md_domain_value.accepted_value
SA_TYPE_DESC_2	md_locale_text.lit_text***{locale2}***	using md_domain_value.accepted_value
EXTERNAL_REFERENCE	null	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

5.2.12 Columns for the BI_SA_VEND Table

Name	Value Coming From	Dimension
SA_VEND_KEY (PK)	vendor.vendor_num	
VENDOR_NUM	vendor.vendor_num	
VENDOR_NAME	vendor.vend_name	
EXTERNAL_REFERENCE	null	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

5.2.13 Columns for the BI_SA_WHSE Table

Name	Value Coming From	Dimension
SA_WHSE_KEY (PK)	whse.whse_code	
WHSE_CODE	whse.whse_code	
WHSE_DESC_1	whse.desc_1	
WHSE_DESC_2	whse.desc_2	
EXTERNAL_REFERENCE	null	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

5.2.14 DMS Sales Analysis Measures Definition

5.2.14.1 bi_sa_sale Fact Table

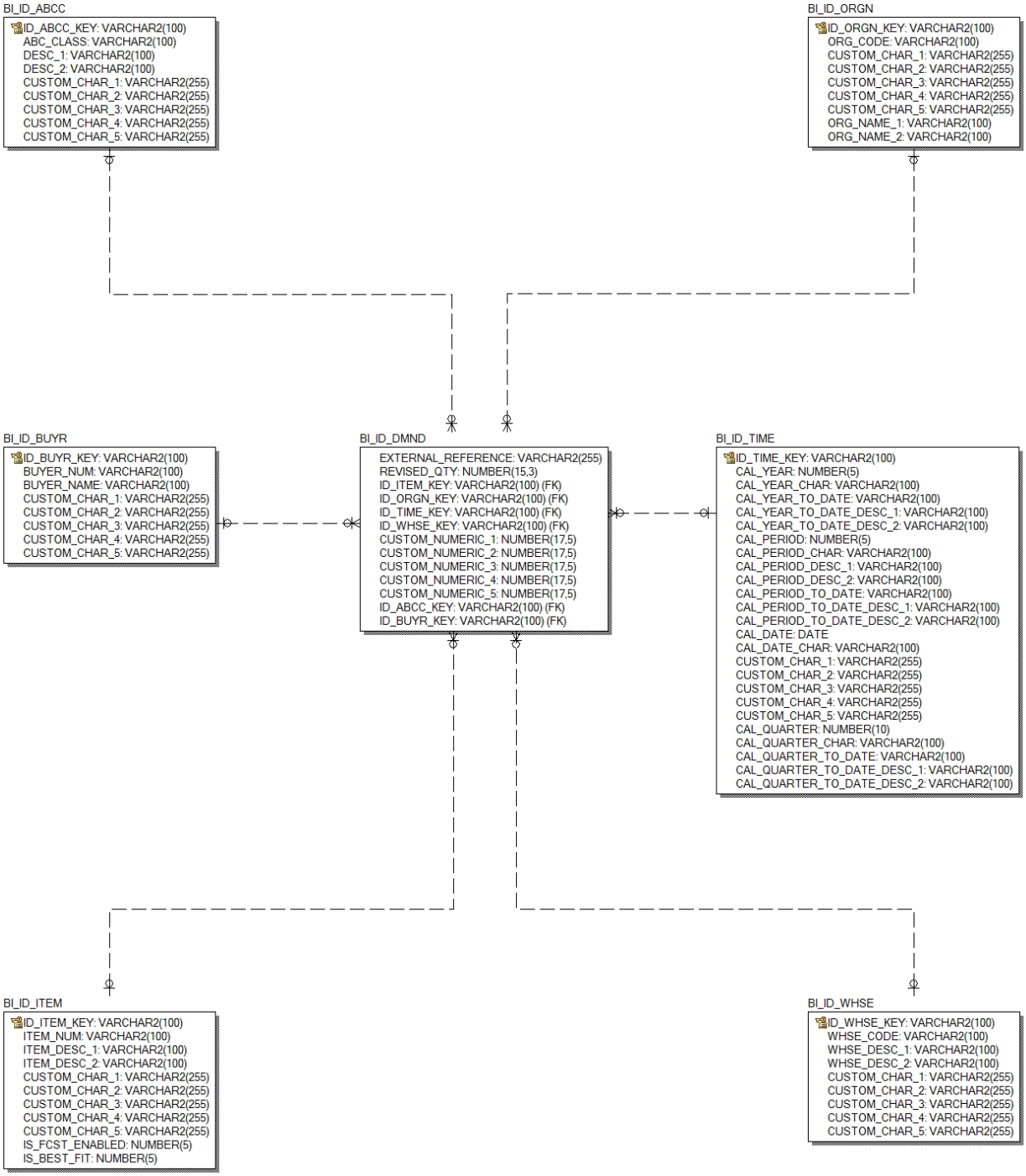
Name	Value	ETL/Aggregate
	from sa sa join sa_sale sa_sale on sa.sa_id = sa_sale.sa_id join customer on customer.cust_num = sa.cust_num left outer join cust_shp on cust_shp.cust_num = sa.cust_num and cust_shp.ship_num = sa.ship_num left outer join inv_shp_l on sa_sale.inv_shp_l_id = inv_shp_l.inv_shp_l_id left outer join inv_shp on inv_shp_l.inv_shp_id = inv_shp.inv_shp_id left outer join item on sa.item_num = item.item_num left outer join manu on item.manu_code = manu.manu_code where sa.sa_id > *{last sa.sa_id processed}* and sa.sa_id <= *{max sa_sa_id}* order by sa.sa_id	
External Reference	sa_sale.sa_sale_id	ETL
Sales Quantity	sa_sale.sale_qty	ETL/Sum
Sale Value	sa_sale.sale_value	ETL/Sum
Sale Cost	sa_sale.sale_cost	ETL/Sum
Gross Margin	Sales Value - Sale Cost	Sum
Margin % (Sale)	Gross Margin / Sale Value	Calculated
Margin % (Cost)	Gross Margin / Sale Cost	Calculated

5.2.15 DMS Sales Analysis Dimensions Name

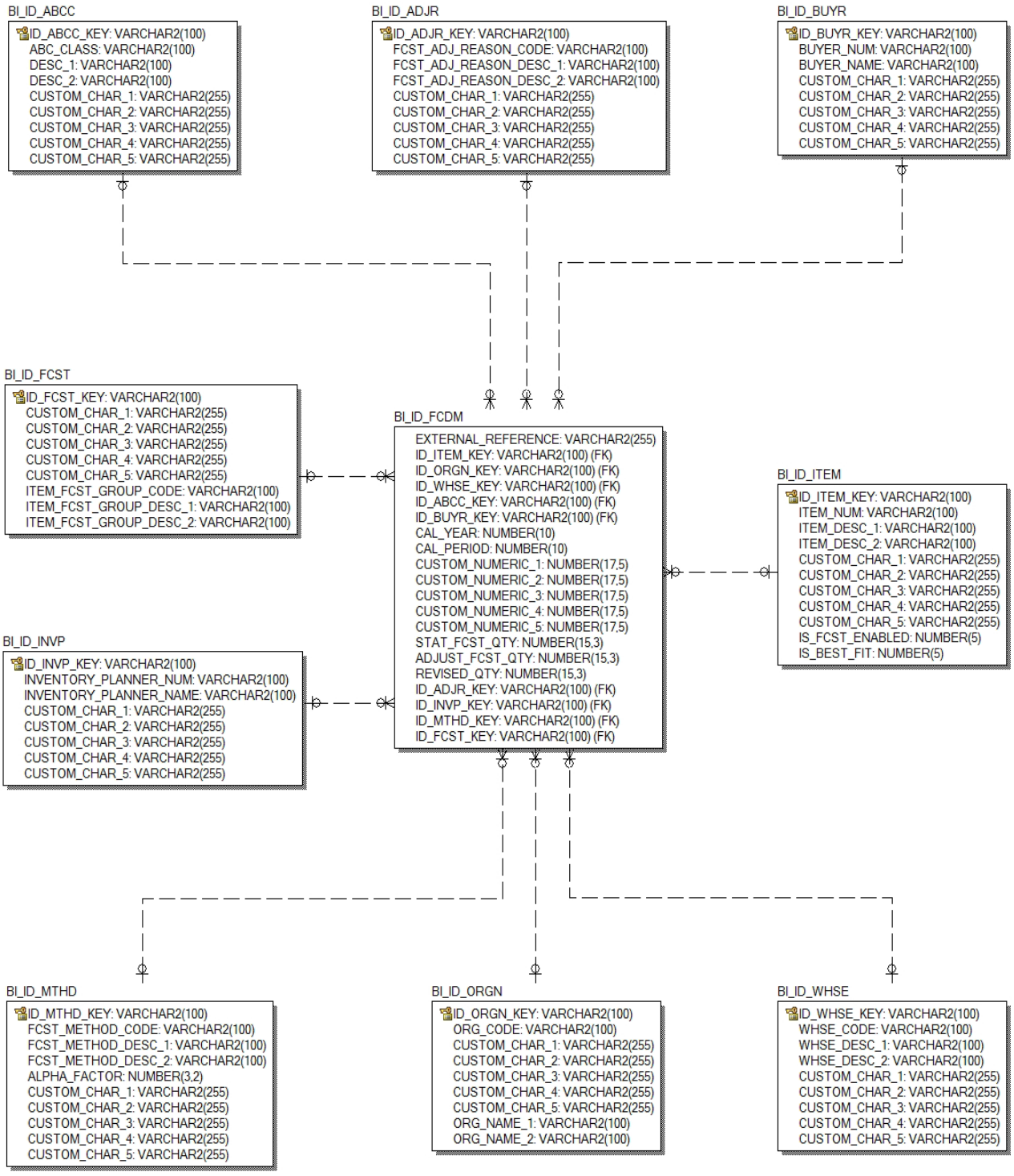
- Time
- Item
- Manufacturer
- Vendor
- Customer
- Location
- Sales Representative
- Warehouse
- Organization
- Ship Via
- Order Type
- Transaction Type

6 FCST Demand and Forecasting

6.1 Structure - Actual



6.2 Structure - Forecast



6.3 Mapping

6.3.1 Columns for the BI_ID_DMND Table

Name	Value Coming From	Fact
EXTERNAL_REFERENCE	demand.warehouse_code '!' demand.item_num '!' organization.calendar_code "!" demand.demand_date '!' warehouse_item.abc_class "!" warehouse_item.buyer_user_name	
ID_ITEM_KEY	demand.item_num	
ID_ORGN_KEY	warehouse_item.organization_code	
ID_TIME_KEY	organization.calendar_code "!" demand.demand_date	[YYYY-MM-DD]
ID_WHSE_KEY	demand.warehouse_code	
REVISED_QTY	demand.demand_qty	
CUSTOM_NUMERIC_1		
CUSTOM_NUMERIC_2		
CUSTOM_NUMERIC_3		
CUSTOM_NUMERIC_4		
CUSTOM_NUMERIC_5		
ID_ABCC_KEY	warehouse_item.abc_class	
ID_BUYR_KEY	warehouse_item.buyer_user_name	

Read records from demand, warehouse_item, organization

where demand.mod_stamp > (bi_dmr_att.dmr_att_value where dmr_name = 'demand-forecast' and dmr_att_key = 'bi_id_dmnd!last_datetime_processed) and mod_stamp <= Today
or
demand.create_stamp > (bi_dmr_att.dmr_att_value where dmr_name = 'demand-forecast' and dmr_att_key = 'bi_id_dmnd!last_datetime_processed``) and create_stamp<= Today

6.3.2 Columns for the BI_ID_FCDM Table

Name	Value Coming From	Fact
EXTERNAL_REFERENCE	forecast.warehouse_code '!' forecast.item_num '!' warehouse_item.item_fcst_group_code '!' warehouse_item.abc_class "!" warehouse_item.buyer_user_name "!" forecast.fcst_adj_reason_code "!" warehouse_item.inv_planner_user_name "!" forecast.fcst_mthd "!" forecast.alpha_factor "!" forecast.calendar_yr + "!" forecast.calendar_period or demand.warehouse_code "!" demand.item_num "!" warehouse_item.item_fcst_group_code '!' warehouse_item.abc_class "!" warehouse_item.buyer_user_name "!" 'UNASSIGNED' "!" warehouse_item.inv_planner_user_name "!" 'UNASSIGNED' "!" demand.demand_date(year portion) + "!" demand.demand_date(month + 1 portion)	
ID_ITEM_KEY	forecast.item_num or demand.item_num	
ID_ORGN_KEY	warehouse_item.org_code	
ID_WHSE_KEY	forecast.warehouse_code or demand.warehouse_code	
ID_ABCC_KEY	warehouse_item.abc_class	
ID_BUYR_KEY	warehouse_item.buyer_user_name	
CAL_YEAR	forecast.calendar_yr or demand.demand_date(year portion)	
CAL_PERIOD	forecast.calendar_period or demand.demand_date(month + 1 portion)	
CUSTOM_NUMERIC_1		
CUSTOM_NUMERIC_2		
CUSTOM_NUMERIC_3		
CUSTOM_NUMERIC_4		
CUSTOM_NUMERIC_5		

Name	Value Coming From	Fact
ID_ADJR_KEY	forecast.fcst_adj_reason_code or 'UNASSIGNED'	
ID_INVP_KEY	warehouse_item.inv_planner_user_name	
ID_MTHD_KEY	forecast.fcst_mthd "!" forecast.alpha_factor "!" or 'UNASSIGNED'	
ID_FCST_KEY	warehouse_item.item_fcst_group_code	
stat_fcst_qty	forecast.stat_fcst_qty	
adjust_fcst_qty	forecast.adjust_fcst_qty	
revised_qty	demand.demand_qty	

Read records from forecast, warehouse_item where union all demand, warehouse_item
forecast.mod_stamp > (bi_dmr_att.dmr_att_value where dmr_name =
'demand-forecast' and dmr_att_key = 'bi_id_fcdm!last_datetime_processed) and mod_stamp <= Today
Or forecast.create_stamp > (bi_dmr_att.dmr_att_value where dmr_name = 'demand-forecast' and dmr_att_key =
'bi_id_fcdm!last_datetime_processed") and create_stamp<= Today
note: The union all on demand is to make sure to capture all demand in case there is no corresponding forecast.

6.3.3 Columns for the BI_ID_ABCC Table

Name	Value Coming From	Dimension
ID_ABCC_KEY (PK)	warehouse_item.abc_class if null 'UNASSIGNED'	
ABC_CLASS	warehouse_item.abc_class if null 'UNASSIGNED'	
DESC_1	warehouse_item.abc_class if null 'UNASSIGNED'	
DESC_2	warehouse_item.abc_class if null 'UNASSIGNED'	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

6.3.4 Columns for the BI_ID_ADJR Table

Name	Value Coming From	Dimension
ID_ADJR_KEY (PK)	fcst_adj_reason.fcst_adj_reason_code	
FCST_ADJ_REASON_CODE	fcst_adj_reason.fcst_adj_reason_code	
FCST_ADJ_REASON _DESC_1	fcst_adj_reason.desc_1	
FCST_ADJ_REASON _DESC_2	fcst_adj_reason.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

6.3.5 Columns for the BI_ID_BUYR Table

Name	Value Coming From	Dimension
ID_BUYR_KEY (PK)	warehouse_item.buyer_user_name	
BUYER_NUM	warehouse_item.buyer_user_name	
BUYER_NAME	md_user.first_name ‘ ‘ md_user.last_name	select md_user.first_name, md_user.last_name from md_user where user_name = ?
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

6.3.6 Columns for the BI_ID_FCST Table

Name	Value Coming From	Dimension
ID_FCST_KEY (PK)	item_fcst_grp.item_fcst_group_code	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		

Name	Value Coming From	Dimension
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		
ITEM_FCST_GROUP_CODE	item_fcst_grp.item_fcst_group_code	
ITEM_FCST_GROUP_DESC_1	item_fcst_grp.desc_1	
ITEM_FCST_GROUP_DESC_2	item_fcst_grp.desc_2	

6.3.7 Columns for the BI_ID_INVP Table

Name	Value Coming From	Dimension
ID_INVP_KEY (PK)	warehouse_item.inv_planner_user_name if null 'UNASSIGNED'	
INVENTORY_PLANNER_NUM	warehouse_item.inv_planner_user_name if null 'UNASSIGNED'	
INVENTORY_PLANNER_NAME	md_user.first_name ' ' md_user.last_name	select md_user.first_name, md_user.last_name from md_user where user_name = ?
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

6.3.8 Columns for the BI_ID_ITEM Table

Name	Value Coming From	Fact
ID_ITEM_KEY (PK)	warehouse_item.item_num	
ITEM_NUM	warehouse_item.item_num	
ITEM_DESC_1	warehouse_item.desc_1	
ITEM_DESC_2	warehouse_item.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		

Name	Value Coming From	Fact
CUSTOM_CHAR_5		
IS_FCST_ENABLED	warehouse_item.is_fcst_enabled	
IS_BEST_FIT	warehouse_item.is_best_fit	

6.3.9 Columns for the BI_ID_MTHD Table

Name	Value Coming From	Fact
ID_MTHD_KEY (PK)	forecast.fcst_mthd "!" forecast.alpha_factor "!" or 'UNASSIGNED'	
FCST_METHOD_CODE	forecast.fcst_mthd "!" forecast.alpha_factor "!" or 'UNASSIGNED'	
FCST_METHOD _DESC_1	md_locale_text.lit_text for locale_1 where lit_key correspond to the desc_key of md_domain_value for table forecast and column fcst_mthd and forecast.fcst_mthd or 'UNASSIGNED'	
FCST_METHOD _DESC_2	md_locale_text.lit_text for locale_2 where lit_key correspond to the desc_key of md_domain_value for table forecast and column fcst_mthd and forecast.fcst_mthd or 'Non Assigné'	
ALPHA_FACTOR	forecast.alpha_factor or 0	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

6.3.10 Columns for the BI_ID_ORGN Table

Name	Value Coming From	Dimension
ID_ORGN_KEY (PK)	organization.organization_code	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		
ORG_NAME_1	organization.name_1	
ORG_NAME_ 2	organization.name_2	

6.3.11 Columns for the BI_ID_TIME Table

Name	Value Coming From	Dimension
ID_TIME_KEY (PK)	calendar_day.calendar_code '!' calendar_day.calendar_date	[YYYY-MM-DD]
CAL_YEAR	calendar_day.calendar_yr	
CAL_YEAR_CHAR	calendar_day.calendar_period	
CAL_YEAR_TO_DATE	'1' '2'	CAL_YEAR = calendar_day.calendar_yr-1 & CAL_DATE <= last year TODAY CAL_YEAR = calendar_day.calendar_yr & CAL_DATE <= TODAY
CAL_YEAR_TO_DATE_DESC_1	'Last Year-To-Date' 'Year-To-Date'	CAL_YEAR = calendar_day.calendar_yr-1 & CAL_DATE <= last year TODAY CAL_YEAR = calendar_day.calendar_yr & CAL_DATE <= TODAY
CAL_YEAR_TO_DATE_DESC_2	'Last Year-To-Date' 'Year-To-Date'	CAL_YEAR = calendar_day.calendar_yr-1 & CAL_DATE <= last year TODAY CAL_YEAR = calendar_day.calendar_yr & CAL_DATE <= TODAY
CAL_PERIOD	calendar_day.calendar_period	
CAL_PERIOD_CHAR	calendar_day.calendar_yr '-' calendar_day.calendar_period	
CAL_PERIOD_DESC_1	calendar_day.calendar_period	
CAL_PERIOD_DESC_2	calendar_day.calendar_period	

Name	Value Coming From	Dimension
CAL_PERIOD_TO_DATE	'1' '2' '3'	CAL_YEAR = calendar_day.calendar_yr - 1 & CAL_PERIOD = calendar_day.calendar_period & CAL_DATE <= last year TODAY CAL_YEAR = calendar_day.calendar_yr & CAL_PERIOD = calendar_day.calendar_period – 1 & CAL_DATE <= last period end date CAL_YEAR = calendar_day.calendar_yr & CAL_PERIOD = calendar_day.calendar_period & CAL_DATE <= TODAY
CAL_PERIOD_TO_DATE_DESC_1	'Last Year Period-To-Date' 'Last Period-To-Date' 'Period-To-Date'	CAL_YEAR = calendar_day.calendar_yr- 1 & CAL_PERIOD = calendar_day.calendar_period & CAL_DATE <= last year TODAY CAL_YEAR = calendar_day.calendar_yr & CAL_PERIOD = calendar_day.calendar_period - 1& CAL_DATE <= last period end date CAL_YEAR = calendar_day.calendar_yr & CAL_PERIOD = calendar_day.calendar_period & CAL_DATE <= TODAY
CAL_PERIOD_TO_DATE_DESC_2	'Last Year Period-To-Date' 'Last Period-To-Date' 'Period-To-Date'	CAL_YEAR = calendar_day.calendar_yr- 1 & CAL_PERIOD = calendar_day.calendar_period & CAL_DATE <= last year TODAY CAL_YEAR = calendar_day.calendar_yr & CAL_PERIOD = calendar_day.calendar_period- 1 & CAL_DATE <= last period end date CAL_YEAR = calendar_day.calendar_yr & CAL_PERIOD = calendar_day.calendar_period & CAL_DATE <= TODAY
CAL_DATE	calendar_day.calendar_date	[YYYY-MM-DD]
CAL_DATE_CHAR	calendar_day.calendar_date	[YYYY-MM-DD]
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Name	Value Coming From	Dimension

CAL_QUARTER	if (calendar_day.calendar_period < 4) {
	"1";
	}
	else if (calendar_day.calendar_period < 7) {
	"2";
	}
	else if (calendar_day.calendar_period < 10) {
	"3";
	}
	else {
	"4";
	}	

CAL_QUARTER_CHAR	calendar_day.calendar_yr '-'
	if (calendar_day.calendar_period < 4) {
	"1";
	}
	else if (calendar_day.calendar_period < 7) {
	"2";
	}
	else if (calendar_day.calendar_period < 10) {
	"3";
	}
	else {
	"4";
	}	

Name	Value Coming From	Dimension

CAL_QUARTER_TO_DATE	‘1’
	
	‘2’
	
	‘3’
		CAL_YEAR = calendar_day.calendar_yr - 1
		&
		CAL_QUARTER = if
		(calendar_day.calendar_period < 4) {
		"1";
		}
		else if (calendar_day.calendar_period < 7) {
		"2";
		}
		else if (calendar_day.calendar_period < 10) {
		"3";
		}
		else {
		"4";
		} &
		CAL_PERIOD =
		calendar_day.calendar_period &
		CAL_DATE <= last year TODAY
		CAL_YEAR = calendar_day.calendar_yr &
		CAL_QUARTER = if
		(calendar_day.calendar_period < 4) {
		"1";
		}
		else if (calendar_day.calendar_period < 7) {
		"2";
		}
		else if (calendar_day.calendar_period < 10) {
		"3";
		}
		else {
		"4";
		} &
		CAL_PERIOD =
		calendar_day.calendar_period – 1 &
		CAL_DATE <= last period end date
		CAL_YEAR = calendar_day.calendar_yr &
		CAL_QUARTER = if
		(calendar_day.calendar_period < 4) {
		"1";
		}
		else if (calendar_day.calendar_period < 7) {
		"2";
		}
		else if (calendar_day.calendar_period < 10) {
		"3";
		}
		else {
		"4";
		} &
		CAL_PERIOD =
		calendar_day.calendar_period &
		CAL_DATE <= TODAY

CAL_QUARTER_TO_DATE_DESC_1	‘Last Year Quarter-To-Date’ ‘Last Quarter-To-Date’ ‘Quarter-To-Date’	Same as CAL_QUARTER_TO_DATE

Name	Value Coming From	Dimension
CAL_QUARTER_TO_DATE_DESC_2	Same as CAL_QUARTER_TO_DATE_DESC_1	Same as CAL_QUARTER_TO_DATE

6.3.12 Columns for the BI_ID_WHSE Table

Name	Value Coming From	Dimension
ID_WHSE_KEY (PK)	warehouse.warehouse_code	
WHSE_CODE	warehouse.warehouse_code	
WHSE_DESC_1	warehouse.desc_1	
WHSE_DESC_2	warehouse.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

6.3.13 FCST Demand and Forecasting Measures Definition

6.3.13.1 bi_id_fcdm Fact Table

Name	Value	ETL/Aggregate
	from forecast, warehouse_item where forecast.warehouse_code = warehouse_item.warehouse_code " and forecast.item_num = warehouse_item.item_num union all " from demand, warehouse_item where demand.warehouse_code = warehouse_item.warehouse_code and demand.item_num = warehouse_item.item_num and not exists (select 'found' from forecast where demand.item_num = forecast.item_num and demand.warehouse_code = forecast.warehouse_code and forecast.calendar_yr = to_char(demand.demand_date,'YYYY') and forecast.calendar_period = to_char(demand.demand_date, 'MM') and ((forecast.mod_stamp > {last datetime processed} and forecast.mod_stamp <= {today's date}) or (forecast.create_stamp > {last datetime processed} and forecast.create_stamp <= {today's date}))	

6.3.13.2 bi_id_dmnd Fact Table

Name	Value	ETL/Aggregate
	from demand, warehouse_item, organization where demand.warehouse_code = warehouse_item.warehouse_code and demand.item_num = warehouse_item.item_num and warehouse_item.organization_code = organization.organization_code and ((demand.mod_stamp > *{last datetime processed}* and demand.mod_stamp <= *{today's date}*) or (demand.create_stamp > *{last datetime processed}* and demand.create_stamp <= *{today's date}*))	
External Reference(bi_id_fcdm)	forecast.whse_code+ "!" + forecast.item_num + "!" + warehouse_item.item_fcst_group_code + "!" + warehouse_item.abc_class + "!" + warehouse_item.buyer_user_name + "!" + forecast.fcst_adj_reason_code + "!" + warehouse_item.inv_planner_user_name + "!" + forecast.fcst_method + "!" + forecast.calendar_yr + "!" + forecast.calendar_period	ETL
External Reference(bi_id_dmnd)	demand.whse_code+ "!" + demand.item_num + "!" + organization.calendar_code + "!" + demand.demand_date + "!" + warehouse_item.abc_class + "!" + warehouse_item.buyer_user_name	ETL
Demand Quantity	demand.demand_qty	ETL/Sum
Statistical Forecast Quantity	forecast.stat_fcst_qty	ETL/Sum
Adjusted Forecast Quantity	forecast.adjust_fcst_qty	ETL/Sum
Statistical MAPE	abs((Demand Quantity - Statistical Forecast Quantity) / Demand Quantity]	Average
Statistical SMAPE	abs((Statistical Forecast Quantity - Demand Quantity) / (Demand Quantity + Statistical Forecast Quantity]))	Average
Statistical MAD	abs(Demand Quantity - Statistical Forecast Quantity)	Average

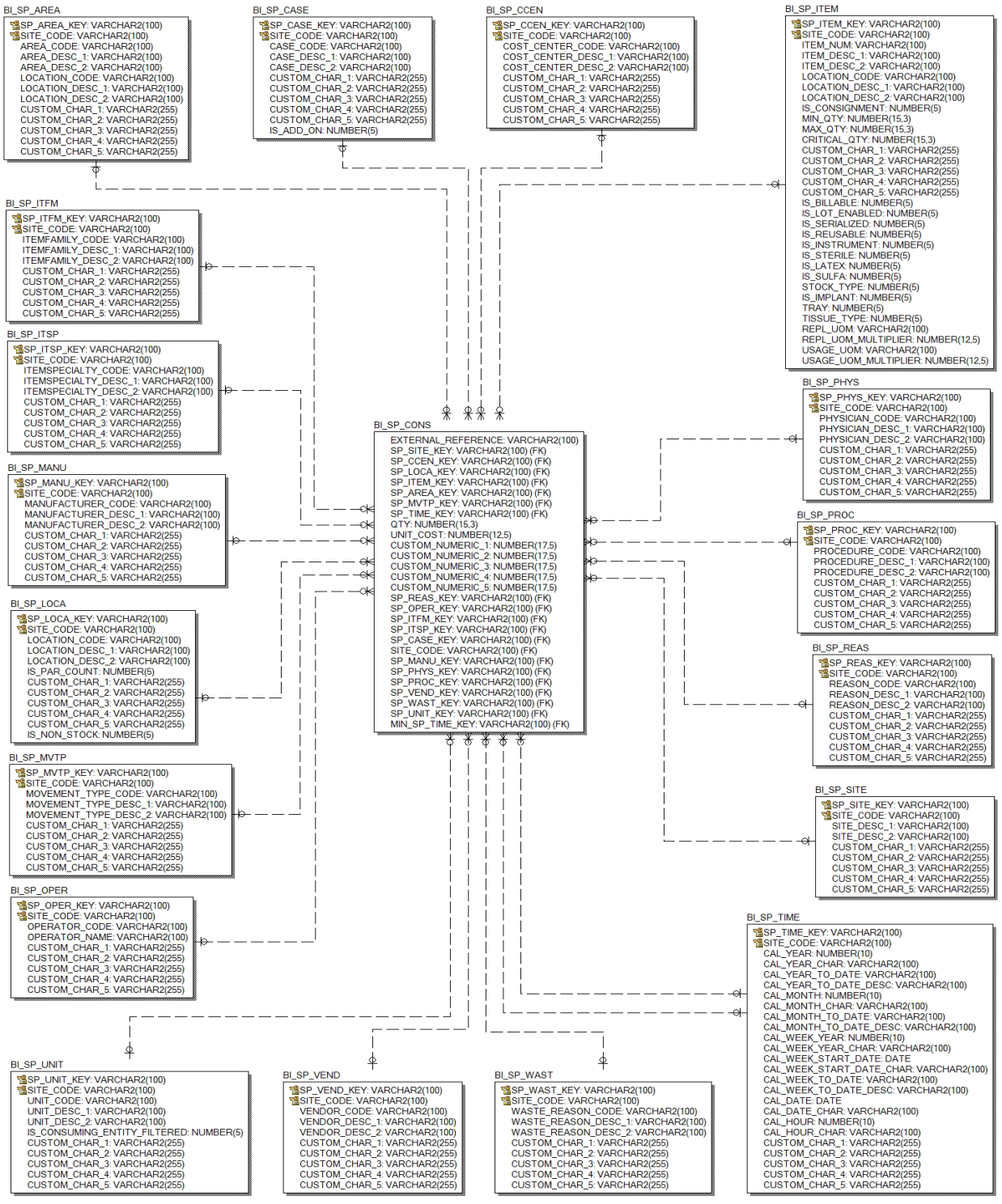
Name	Value	ETL/Aggregate
Statistical (Actual - Forecast)	Demand Quantity - Statistical Forecast Quantity	Sum(hidden)
Running Statistical Forecast Error	running-total(Statistical (Actual - Forecast)])	Calculated(hidden)
Statistical Tracking Signal	Running Statistical Forecast Error / Statistical MAD	Calculated
Adjusted MAPE	abs((Demand Quantity - Adjusted Forecast Quantity) / Demand Quantity]	Average
Adjusted SMAPE	abs((Adjusted Forecast Quantity - Demand Quantity) / (Demand Quantity + Adjusted Forecast Quantity]))	Average
Adjusted MAD	abs(Demand Quantity - Adjusted Forecast Quantity)	Average
Adjusted (Actual - Forecast)	Demand Quantity - Adjusted Forecast Quantity	Sum(hidden)
Running Adjusted Forecast Error	running-total(Adjusted (Actual - Forecast)])	Calculated(hidden)
Adjusted Tracking Signal	Running Adjusted Forecast Error / Adjusted MAD	Calculated

6.3.14 FCST Demand and Forecasting Dimensions Name

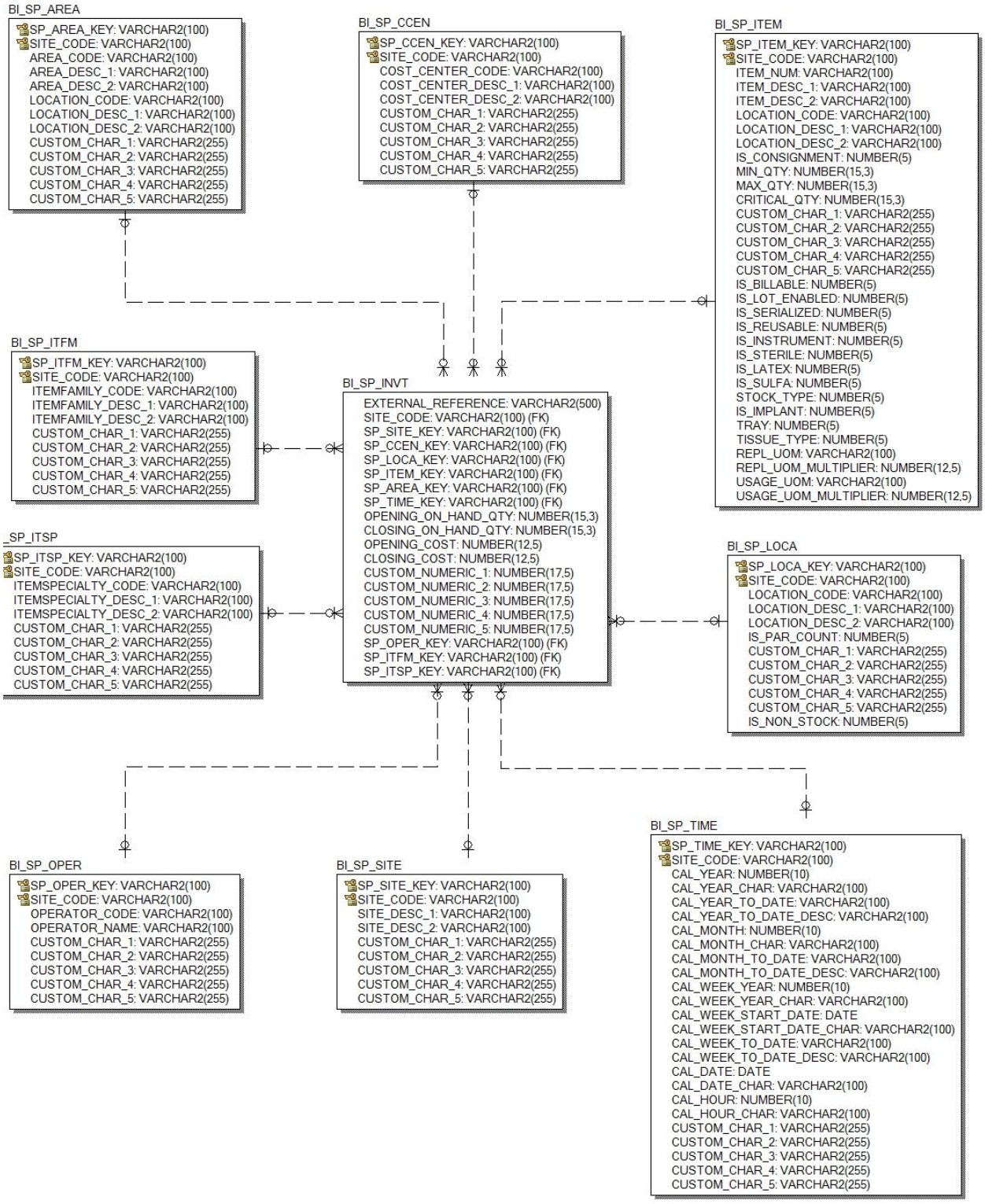
- Time
- ABC Class
- Adjustment Reason
- Buyer
- Item Forecast Group
- Inventory Planner
- Item
- Forecast Method
- Organization
- Warehouse

7 SMS Supply Performance

7.1 Structure - Consumption



7.2 Structure - Inventory



7.3 Mapping

7.3.1 Columns for the BI_SP_CONS Table

Name	Value Coming From	Fact
EXTERNAL_REFERENCE	'1!' movement.movement_id or 'See Note' '!' usage_order.order_num '!' usage_order_detail.order_detail_line_num '!' usage_order_detail.location_code '!" usage_order_detail.item_num '!' usage_order_detail.location_code Note: MovementType correspond to: Movement Type-Case Pick Add-On(22) Movement Type-Case Pick(12) Movement Type-Usage(4) Movement Type-Waste(20) Movement Type-Soiled(21) Movement Type-Putaway(13) Movement Type-Return(14) Movement Type-Inventory Adj(8)	
SITE_CODE	movement.site_code or usage_order.site_code	
SP_SITE_KEY	movement.site_code or usage_order.site_code	
SP_CCEN_KEY	movement.cost_center_code or placegroup.cost_center_code	
SP_LOCA_KEY	movement.location_code or usage_order_detail.location_code	
SP_ITEM_KEY	movement.item_num + '!' + movement.location_code or usage_order_detail.item_num + '!' + usage_order_detail.location_code	
SP_AREA_KEY	movement.location_code or usage_order_detail.location_code	

Name	Value Coming From	Fact
SP_MVTP_KEY	movement.movement_type or calculated off the usage_order_detail quantities 1-In Transit 2-Replenishment 3-Usage Floor 4-Usage OR 5-Usage Count 6-Transfer Out 7-Adjustment Cost 8-Adjustment Inventory 9-Transfer In 10-Serial Number Assignment 11-Count 12-Case Pick 13-Case Putaway 14-Return 15-Waste 16-Case Move 17-Delivery Putaway 18-Lost in Transit 19-Lost During Putaway 20-Waste OR 21-Soiled OR 22-Case Pick Add-On	
SP_REAS_KEY	movement.invadjust_reason_code or usage_order_detail.waste_reason_code	
SP_TIME_KEY	movement.create_stamp or usage_order.create_stamp	[YYYY-MM-DD HH:00:00]

Name	Value Coming From	Fact
QTY	movement.movement_qty or qtyAddOn(22) = usage_order_detail.picked_qty – usage_order_detail.requested_qty (preference_card_qty) qtyPicked(12) = usage_order_detail.requested_qty(pr eference_card_qty) qtyUsage(4) = usage_order_detail.usage_qty qtyWasted(20) = usage_order_detail.wasted_qty qtySoiled(21) = usage_order_detail.soiled_qty qtyReturn(putaway) qtyUnusedInventory = qtyPicked - (qtyUsage + qtyWasted + qtySoiled) bonusQty = qtyReturn(putaway) – qtyUnusedInventory If BonusQty > 0 then Movement Putaway = Qty Return(putaway) – BonusQty (if result 0 no putaway movement) Movement Return = BonusQty If BonusQty = 0 if Qty Return > 0 Movement Putaway = Qty Return(putaway) (if result 0 no putaway movement) If BonusQty < 0 Movement Putaway = Qty Return(putaway) (if result 0 no putaway movement) Movement Inventory Adjustment = Qty Return(putaway) - UnusedInventoryQty (if result 0 no inventory adjustment movement) If qtySoiled > 0 Movement Putaway = qtySoiled If qtyWasted > 0 Movement Inventory Adjustment = qtyWasted	
UNIT_COST	movement.item_cost or usage_order_detail.item_cost or location_item.replacement_cost	
CUSTOM_NUMERIC_1		
CUSTOM_NUMERIC_2		

Name	Value Coming From	Fact
CUSTOM_NUMERIC_3		
CUSTOM_NUMERIC_4		
CUSTOM_NUMERIC_5		
SP_OPER_KEY	movement.create_user or usage_order.create_user	
SP_ITFM_KEY	item.itemfamily_code	
SP_ITSP_KEY	item.itemspecialty_code	
SP_CASE_KEY	usage_order.order_num '!' usage_order_detail.order_detail_line_num	
SP_MANU_KEY	manuitem.manufacturer_code	
SP_PHYS_KEY	usage_order.physician_code	
SP_PROC_KEY	usage_order.operating_proc_code	
SP_UNIT_KEY	usage_order.placegroup_code	
SP_VEND_KEY	location.vendor_code	
SP_WAST_KEY	usage_order_detail.waste_reason_code	
MIN_SP_TIME_KEY	min(create_stamp) from movement or min(create_stamp) from usage_order	

Read records from movement, location_item, location

where the location type is 0 and the movement id > last movement id processed

and movement id <= to the max movement id for each site.

7.3.2 Columns for the BI_SP_INVT Table

Name	Value Coming From	Fact
EXTERNAL_REFERENCE	movement.create_stamp(YYYY-MM-DD) "!" movement.site_code "!" movement.cost_center_code "!" movement.location_code "!" movement.item_num "!" movement.location_code "!" movement.create_user	
SITE_CODE	movement.site_code	
SP_SITE_KEY	movement.site_code	
SP_CCEN_KEY	movement.cost_center_code	

Name	Value Coming From	Fact
SP_LOCA_KEY	movement.location_code	
SP_ITEM_KEY	movement.item_num	
SP_AREA_KEY	movement.location_code	
SP_TIME_KEY	movement.create_stamp	[YYYY-MM-DD 00:00:00]
OPENING_ON_HAND_QTY	select top 1 movement.old_inventory_qty from movement where movement.site_code = {movement.site_code} and movement.cost_center_code = {movement.cost_center_code} and movement.location_code = {movement.location_code} and movement.item_num = {movement.item_num} and movement.movement_type != '1' and movement.movement_type != '8' and (old_inventory_qty != new_inventory_qty or (old_inventory_qty = new_inventory_qty and movement_qty = 0)) and movement.create_stamp between {ts '{create_date} 00:00:00'} and {ts '{create_date} 23:59:59'} order by movement.create_stamp	
CLOSING_ON_HAND_QTY	select top 1 movement.new_inventory_qty from movement where movement.site_code = {movement.site_code} and movement.cost_center_code = {movement.cost_center_code} and movement.location_code = {movement.location_code} and movement.item_num = {movement.item_num} and movement.movement_type != '1' and movement.movement_type != '8' and (old_inventory_qty != new_inventory_qty or (old_inventory_qty = new_inventory_qty and movement_qty = 0)) and movement.create_stamp between {ts '{create_date} 00:00:00'} and {ts '{create_date} 23:59:59'} order by movement.create_stamp desc	

Name	Value Coming From	Fact
CLOSING_COST	select top 1 movement.item_cost from movement where movement.site_code = {movement.site_code} and movement.cost_center_code = {movement.cost_center_code} and movement.location_code = {movement.location_code} and movement.item_num = {movement.item_num} and movement.movement_type != '1' and movement.movement_type != '8' and (old_inventory_qty != new_inventory_qty or (old_inventory_qty = new_inventory_qty and movement_qty = 0)) and movement.create_stamp between {ts '{create_date} 00:00:00'} and {ts '{create_date} 23:59:59'} order by movement.create_stamp desc	
CUSTOM_NUMERIC_1		
CUSTOM_NUMERIC_2		
CUSTOM_NUMERIC_3		
CUSTOM_NUMERIC_4		
CUSTOM_NUMERIC_5		
SP_OPER_KEY	movement.create_user	
SP_ITFM_KEY	item.itemfamily_code	
SP_ITSP_KEY	item.itemspecialty_code	

Read records from movement, location_item, location
where the movement id > last movement id processed

and movement id <= to the max movement id for each site.

7.3.3 Columns for the BI_SP_AREA Table

Name	Value Coming From	Dimension
SP_AREA_KEY (PK)	supply_area.supply_area_code + "!" + location.location_code	
SITE_CODE	site.site_code	
AREA_CODE	supply_area.supply_area_code	
AREA_DESC_1	supply_area.desc_1	
AREA_DESC_2	supply_area.desc_2	
LOCATION_CODE	location.location_code	
LOCATION_DESC_1	location.desc_1	
LOCAT|ION_DESC_2	location.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.4 Columns for the BI_SP_CASE Table

Name	Value Coming From	Dimension
SP_CASE_KEY (PK)	usage_order.order_num '!' usage_order_detail.order_detail_line_num	
SITE_CODE	site.site_code	
CASE_CODE	usage_order.order_num '!' usage_order_detail.order_detail_line_num	
CASE_DESC_1	usage_order.order_num '!' usage_order_detail.order_detail_line_num	
CASE_DESC_2		
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		
IS_ADD_ON	0 or if usage_order_detail.qty_picked – usage_order_detail.requested_qty(preference_ card_qty) > 0 then 1	

7.3.5 Columns for the BI_SP_CCEN Table

Name	Value Coming From	Dimension
SP_CCEN_KEY (PK)	location_cost_center.cost_center_code	
SITE_CODE	site.site_code	
COST_CENTER_CODE	location_cost_center.cost_center_code	
COST_CENTER _DESC_1	cost_center.desc_1	
COST_CENTER _DESC_2	cost_center.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.6 Columns for the BI_SP_ITEM Table

Name	Value Coming From	Fact
SP_ITEM_KEY (PK)	location_item.item_num '!' location_item.location_code	
SITE_CODE	site.site_code	
ITEM_NUM	location_item.item_num	
ITEM_DESC_1	item.desc_1	
ITEM_DESC_2	item.desc_2	
LOCATION_CODE	location_item.location_code	
LOCATION_DESC_1	location.desc_1	
LOCATION_DESC_2	location.desc_2	
IS_CONSIGNMENT	item.is_consignment	
MIN_QTY	location_item.min_qty	
MAX_QTY	location_item.max_qty	
CRITICAL_QTY	location_item.critical_qty	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		
IS_BILLABLE	item.is_billable	
IS_LOT_ENABLED	location_item.is_lot_enabled	
IS_SERIALIZED	item.is_serialized	
IS_REUSABLE	item.is_reusable	
IS_INSTRUMENT	item.is_instrument	
IS_STERILE	item.is_sterile	
IS_LATEX	item.is_latex	
IS_SULFA	item.is_sulfa	
STOCK_TYPE	location_item.stock_type	

Name	Value Coming From	Fact
IS_IMPLANT	item.is_implant	
TRAY	item.tray	
TISSUE_TYPE	item.tissue_type	
REPL_UOM	location_item.repl_uom_code	
REPL_UOM_MULTIPLIER	item_uom.uom_multiplier where item_uom.uom_code = location_item.repl_uom_code and item_uom.item_num = usage_order_detail.item_num	
USAGE_UOM	location_item.usage_uom_code	
USAGE_UOM_MULTIPLIER	item_uom.uom_multiplier where item_uom.uom_code = location_item.usage_uom_code and item_uom.item_num = usage_order_detail.item_num	

7.3.7 Columns for the BI_SP_ITFM Table

Name	Value Coming From	Fact
SP_ITFM_KEY (PK)	itemfamily.itemfamily_code	
SITE_CODE	site.site_code	
ITEMFAMILY_CODE	itemfamily. itemfamily_code	
ITEMFAMILY_DESC_1	itemfamily.desc_1	
ITEMFAMILY_DESC_2	itemfamily.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.8 Columns for the BI_SP_ITSP Table

Name	Value Coming From	Fact
SP_ITSP_KEY (PK)	itemspecialty.itemspecialty_code	
SITE_CODE	site.site_code	

Name	Value Coming From	Fact
ITEMSPECIALTY_CODE	itemspecialty.itemspecialty_code	
ITEMSPECIALTY_DESC_1	itemspecialty.desc_1	
ITEMSPECIALTY_DESC_2	itemspecialty.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.9 Columns for the BI_SP_LOCA Table

Name	Value Coming From	Fact
SP_LOCA_KEY (PK)	location_item.location_code	
SITE_CODE	site.site_code	
LOCATION_CODE	location_item.location_code	
LOCATION_DESC_1	location.desc_1	
LOCATION_DESC_2	location.desc_2	
IS_PAR_COUNT	location.is_par_count	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		
IS_NON_STOCK	location.is_non_stock	

7.3.10 Columns for the BI_SP_MANU Table

Name	Value Coming From	Fact
SP_MANU_KEY (PK)	manufacturer.manufacturer_code	
SITE_CODE	site.site_code	

Name	Value Coming From	Fact
MANUFACTURER_CODE	manufacturer.manufacturer_code	
MANUFACTURER_DESC_1	manufacturer.desc_1	
MANUFACTURER_DESC_2	manufacturer.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.11 Columns for the BI_SP_MVTP Table

Name	Value Coming From	Fact
SP_MVTP_KEY (PK)	md_domain_value.accepted_value 1-In Transit 2-Replenishment 3-Usage Floor 4-Usage OR 5-Usage Count 6-Transfer Out 7-Adjustment Cost 8-Adjustment Inventory 9-Transfer In 10-Serial Number Assignment 11-Count 12-Case Pick 13-Case Putaway 14-Return 15-Waste 16-Case Move 17-Delivery Putaway 18-Lost in Transit 19-Lost During Putaway 20-Waste OR 21-Soiled OR 22-Case Pick Add-On	Database bi, Domain_name analytic_movement_type
SITE_CODE	site.site_code	
MOVEMENT_TYPE_CODE	md_domain_value.accepted_value	
MOVEMENT_TYPE_DESC_1	md_locale_text.lit_text for locale_1 where lit_key correspond to the desc_key of md_domain_value of sp_mvtp_key	

Name	Value Coming From	Fact
MOVEMENT_TYPE_DESC_2	md_locale_text.lit_text for locale_2 where lit_key correspond to the desc_key of md_domain_value of sp_mvtp_key	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.12 Columns for the BI_SP_OPER Table

Name	Value Coming From	Dimension
SP_OPER_KEY (PK)	movement.create_user or usage_order.create_user	
SITE_CODE	site..site_code	
OPERATOR_CODE	movement.create_user or usage_order.create_user	
OPERATOR_NAME	movement.create_user or usage_order.create_user	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.13 Columns for the BI_SP_PHYS Table

Name	Value Coming From	Dimension
SP_PHYS_KEY (PK)	physician.physician_code	
SITE_CODE	site..site_code	
PHYSICIAN_CODE	physician.physician_code	
PHYSICIAN_DESC_1	physician.desc_1	
PHYSICIAN_DESC_2	physician.desc_2	
CUSTOM_CHAR_1		

Name	Value Coming From	Dimension
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.14 Columns for the BI_SP_PROC Table

Name	Value Coming From	Dimension
SP_PROC_KEY (PK)	operating_proc.operating_proc_code	
SITE_CODE	site..site_code	
PROCEDURE_CODE	operating_proc.operating_proc_code	
PROCEDURE_DESC_1	operating_proc.desc_1	
PROCEDURE_DESC_2	operating_proc.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.15 Columns for the BI_SP_REAS Table

Name	Value Coming From	Dimension
SP_REAS_KEY (PK)	invadjust_reason.invadjust_reason_code	
SITE_CODE	site.site_code	
REASON_CODE	invadjust_reason.invadjust_reason_code	
REASON_DESC_1	invadjust_reason.desc_1	
REASON_DESC_2	invadjust_reason.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		

Name	Value Coming From	Dimension
CUSTOM_CHAR_5		

7.3.16 Columns for the BI_SP_SITE Table

Name	Value Coming From	Dimension
SP_SITE_KEY (PK)	site.site_code	
SITE_CODE	site.site_code	
SITE_DESC_1	site.desc_1	
SITE_DESC_2	site.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.17 Columns for the BI_SP_TIME Table

Name	Value Coming From	Dimension
SP_TIME_KEY (PK)	movement.create_stamp	[YYYY-MM-DD 00:00:00]
SITE_CODE	site.site_code	
CAL_YEAR	movement.create_stamp(YYYY)	
CAL_YEAR_CHAR	movement.create_stamp(YYYY)	
CAL_YEAR_TO_DATE	'1' '2'	CAL_YEAR = movement.create_stamp(YYYY)-1 & CAL_DATE <= last year TODAY CAL_YEAR = movement.create_stamp(YYYY) & CAL_DATE <= TODAY
CAL_YEAR_TO_DATE_DESC	'Last Year-To-Date' 'Year-To-Date'	CAL_YEAR = movement.create_stamp(YYYY)-1 & CAL_DATE <= last year TODAY CAL_YEAR = movement.create_stamp(YYYY) & CAL_DATE <= TODAY
CAL_MONTH	movement.create_stamp(MM)	
CAL_MONTH_CHAR	movement.create_stamp(YYYY-MM)	

Name	Value Coming From	Dimension
CAL_MONTH_TO_DATE	'1' '2' '3'	CAL_YEAR = movement.create_stamp(YYYY) - 1 & CAL_MONTH = movement.create_stamp(MM) & CAL_DATE <= last year TODAY CAL_YEAR = movement.create_stamp(YYYY) & CAL_MONTH = movement.create_stamp(MM) – 1 & CAL_DATE <= last period end date CAL_YEAR = movement.create_stamp(YYYY) & CAL_MONTH = movement.create_stamp(MM) & CAL_DATE <= TODAY
CAL_MONTH_TO_DATE_DESC	'Last Year Month-To-Date' 'Last Month-To-Date' 'Month-To-Date'	CAL_YEAR = movement.create_stamp(YYYY) - 1 & CAL_MONTH = movement.create_stamp(MM) & CAL_DATE <= last year TODAY CAL_YEAR = movement.create_stamp(YYYY) & CAL_MONTH = movement.create_stamp(MM) – 1 & CAL_DATE <= last period end date CAL_YEAR = movement.create_stamp(YYYY) & CAL_MONTH = movement.create_stamp(MM) & CAL_DATE <= TODAY
CAL_WEEK_YEAR	movement.create_stamp(YYYY)	use movement.create_stamp in order to determine the week start date and grab the year portion
CAL_WEEK_YEAR_CHAR	movement.create_stamp(YYYY)	
CAL_WEEK_START_DATE	movement.create_stamp	use movement.create_stamp in order to determine the week start date [YYYY-MM-DD 00:00:00]
CAL_WEEK_START_DATE_CHAR	movement.create_stamp	use movement.create_stamp in order to determine the week start date [YYYY-MM-DD]
CALL_WEEK_TO_DATE	'1' '2'	
CAL_WEEK_TO_DATE_DESC	'Last Week-To-Date' 'Week-To-Date'	
CAL_DATE	movement.create_stamp	[YYYY-MM-DD 00:00:00]
CAL_DATE_CHAR	movement.create_stamp	[YYYY-MM-DD]

Name	Value Coming From	Dimension
CAL_HOUR	‘0’ through ‘23’	
CAL_HOUR_CHAR	‘00’ through ‘23’	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.18 Columns for the BI_SP_UNIT Table

Name	Value Coming From	Dimension
SP_UNIT_KEY (PK)	placegroup.placegroup_code	
SITE_CODE	site..site_code	
UNIT_CODE	placegroup.placegroup_code	
UNIT_DESC_1	placegroup.desc_1	
UNIT_DESC_2	placegroup.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.19 Columns for the BI_SP_VEND Table

Name	Value Coming From	Dimension
SP_VEND_KEY (PK)	vendor.vendor_code	
SITE_CODE	site..site_code	
VENDOR_CODE	vendor.vendor_code	
VENDOR_DESC_1	vendor.desc_1	
VENDOR_DESC_2	vendor.desc_2	

Name	Value Coming From	Dimension
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.20 Columns for the BI_SP_WAST Table

Name	Value Coming From	Dimension
SP_WAST_KEY (PK)	waste_reason.waste_reason_code	
SITE_CODE	site..site_code	
WASTE_REASON_CODE	waste_reason.waste_reason_code	
WASTE_REASON_DESC_1	waste_reason.desc_1	
WASTE_REASON_DESC_2	waste_reason.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

7.3.21 SMS Supply Performance Measures Definition

7.3.21.1 Consumption - bi_sp_cons Fact Table

Name	Value	ETL/Aggregate
	from movement, location_item, location where movement.location_code = location_item.location_code and movement.item_num = location_item.item_num and movement.location_code = location.location_code and location.location_type = 0 and movement.movement_id > {last movement.movement_id processed} and movement.movement_id <= {maximum movement.movement_id} and movement.site_code = {site.site_code} order by movement.movement_id or from usage_order, usage_order_detail, placegroup, location_item, location, item where usage_order.order_status = 9 and usage_order.site_code = usage_order_detail.site_code and usage_order.order_num = usage_order_detail.order_num and usage_order.site_code = placegroup.site_code and usage_order.placegroup_code = placegroup.placegroup_code and usage_order_detail.location_code = location_item.location_code and usage_order_detail.item_num = location_item.item_num and usage_order_detail.location_code = location.location_code and usage_order_detail.item_num = item.item_num	
External Reference	'1!' movement.movement_id or 'See Note' '!' usage_order.order_num '!' usage_order_detail.order_detail_line_num '!' usage_order_detail.location_code '!" usage_order_detail.item_num '!' usage_order_detail.location_code Note: MovementType correspond to: Movement Type-Case Pick Add-On(22) Movement Type-Case Pick(12) Movement Type-Usage(4) Movement Type-Waste(20)	ETL

Name	Value	ETL/Aggregate
	Movement Type-Soiled(21) Movement Type-Putaway(13) Movement Type-Return(14) Movement Type-Inventory Adj(8)	
Movement Type Code	movement.movement_type or calculated off the usage_order_detail quantities 1-In Transit 2-Replenishment 3-Usage Floor 4-Usage OR 5-Usage Count 6-Transfer Out 7-Adjustment Cost 8-Adjustment Inventory 9-Transfer In 10-Serial Number Assignment 11-Count 12-Case Pick 13-Case Putaway 14-Return 15-Waste 16-Case Move 17-Delivery Putaway 18-Lost in Transit 19-Lost During Putaway 20-Waste OR 21-Soiled OR 22-Case Pick Add-On	ETL
Quantity	movement.movement_qty or qtyAddOn(22) = usage_order_detail.picked_qty – usage_order_detail.requested_qty (preference_card_qty) qtyPicked(12) = usage_order_detail.requested_qty(preference_card_qty) qtyUsage(4) = usage_order_detail.usage_qty qtyWasted(20) = usage_order_detail.wasted_qty qtySoiled(21) = usage_order_detail.soiled_qty qtyReturn(putaway) qtyUnusedInventory = qtyPicked - (qtyUsage + qtyWasted + qtySoiled) bonusQty = qtyReturn(putaway) – qtyUnusedInventory If BonusQty > 0 then Movement Putaway = Qty Return(putaway) – BonusQty (if result 0 no putaway movement) Movement Return = BonusQty	ETL/Sum

Name	Value	ETL/Aggregate
	If BonusQty = 0 if Qty Return > 0 Movement Putaway = Qty Return(putaway) (if result 0 no putaway movement) If BonusQty < 0 Movement Putaway = Qty Return(putaway) (if result 0 no putaway movement) Movement Inventory Adjustment = Qty Return(putaway) - UnusedInventoryQty (if result 0 no inventory adjustment movement) If qtySoiled > 0 Movement Putaway = qtySoiled If qtyWasted > 0 Movement Inventory Adjustment = qtyWasted	
Unit Cost	movement.item_cost or usage_order_detail.item_cost or location_item.replacement_cost	ETL/Average
Total Quantity	Quantity	Sum
Total Cost	Quantity * Unit Cost	Sum
Total Usage	if ((Movement Type Code = '3' or Movement Type Code = '4' or Movement Type Code = '5')) THEN (Quantity) else (null)	Calculated
Total Usage Cost	if ((Movement Type Code = '3' or Movement Type Code = '4' or Movement Type Code = '5')) THEN (Quantity * Unit Cost) else (null)	Calculated
Net Count Variation	if (Movement Type Code = '11') THEN (Quantity) ELSE (null)	Calculated
Net Count Variation Cost	if (Movement Type Code = '11') THEN (Quantity * Unit Cost) ELSE (null)	Calculated
Net Waste	if (Movement Type Code =’15’) THEN (Quantity) ELSE (null)	Calculated
Net Waste Cost	if (Movement Type Code = ‘15’) THEN (Quantity * Unit Cost) ELSE (null)	Calculated
Net Returns	if (Movement Type Code = '14') THEN (Quantity) else (null)	Calculated
Net Returns Cost	if (Movement Type Code = '14') THEN	Calculated

Name	Value	ETL/Aggregate
	(Quantity * Unit Cost) else (null)	
Total Adjusted - Incoming	if (Movement Type Code = '8' and Quantity > 0) THEN (Quantity) else (null)	Caluculated
Total Adjusted – Incoming Cost	if (Movement Type Code = '8' and Quantity > 0) THEN (Quantity * Unit Cost) else (null)	Caluculated
Total Adjusted - Outgoing	if (Movement Type Code = '8' and Quantity < 0) THEN (Quantity) else (null)	Caluculated
Total Adjusted – Outgoing Cost	if (Movement Type Code = '8' and Quantity < 0) THEN (Quantity * Unit Cost) else (null)	Caluculated
Number of Cases	count distinct Case Key(1, pos of ! -1)	Sum
Total Case Quantity	if (Movement Type Code = '4' or Movement Type Code = '20' or Movement Type Code = '21') THEN (Quantity) else (null)	Sum
Average Case Quantity	Total Case Quantity	Average
Total Case Cost	if (Movement Type Code = '4' or Movement Type Code = '20' or Movement Type Code = '21') THEN (Quantity * Unit Cost) else (null)	Calculated
Average Case Cost	Total Case Cost	Average
Total Case – Picked	if (Movement Type Code = '12') THEN (Quantity) else (null)	Sum
Total Case – Picked Cost	if (Movement Type Code = '12') THEN (Quantity * Unit Cost) else (null)	Calculated
Total Case – Used	if (Movement Type Code = '4') THEN (Quantity) else (null)	Sum
Total Case – Used Cost	if (Movement Type Code = '4') THEN (Quantity * Unit Cost) else (null)	Calculated
Total Case – Wasted	if (Movement Type Code = '20') THEN (Quantity) else (null)	Sum
Total Case – Waster Cost	if (Movement Type Code = '20') THEN (Quantity * Unit Cost) else (null)	Calculated
Total Case – Returned	if (Movement Type Code = '14') THEN (Quantity) else (null)	Sum
Total Case – Returned	if (Movement Type Code = '14') THEN (Quantity * Unit Cost) else (null)	Calculated
Total Case – Add-On	if (Movement Type Code = '22') THEN (Quantity) else (null)	Sum
Total Case – Add-On	if (Movement Type Code = '22') THEN	Calculated

Name	Value	ETL/Aggregate
	(Quantity * Unit Cost) else (null)	

7.3.21.2 Inventory - bi_sp_invt Fact Table

Name	Value	ETL/Aggregate
External Reference	movement.create_stamp[YYYY-MM-DD] + "!" + movement.site_code + "!" + movement.cost_center_code + "!" + movement.location_code[sp_loca_key] + "!" + movement.item_num + "!" + movement.location_code[sp_item_key] + "!" + movement.location_code[sp_area_key]	ETL
Opening On-Hand Quantity	select top 1 old_inventory_qty, item_cost opening_cost from movement where movement.site_code = {movement.site_code} and movement.cost_center_code = {movement.cost_center_code} and movement.location_code = {movement.location_code} and movement.item_num = {movement.item_num} and movement.movement_type != '1' and movement.movement_type != '8' and (old_inventory_qty != new_inventory_qty or (old_inventory_qty = new_inventory_qty and movement_qty = 0)) and movement.create_stamp between {ts '{create_date} 00:00:00'} and {ts '{create_date} 23:59:59'} order by movement.create_stamp	ETL/Sum
Closing On-Hand Quantity	select top 1 new_inventory_qty, item_cost closing_cost from movement where movement.site_code = {movement.site_code} and movement.cost_center_code = {movement.cost_center_code} and movement.location_code = {movement.location_code} and movement.item_num = {movement.item_num} and movement.movement_type != '1' and movement.movement_type != '8' and (old_inventory_qty != new_inventory_qty or (old_inventory_qty = new_inventory_qty and movement_qty = 0)) and movement.create_stamp between {ts '{create_date} 00:00:00'} and {ts '{create_date} 23:59:59'} order by movement.create_stamp desc	ETL/Sum

Name	Value	ETL/Aggregate
Opening Cost	select top 1 old_inventory_qty, item_cost opening_cost from movement where movement.site_code = {movement.site_code} and movement.cost_center_code = {movement.cost_center_code} and movement.location_code = {movement.location_code} and movement.item_num = {movement.item_num} and movement.movement_type != '1' and movement.movement_type != '8' and (old_inventory_qty != new_inventory_qty or (old_inventory_qty = new_inventory_qty and movement_qty = 0)) and movement.create_stamp between {ts '{create_date} 00:00:00'} and {ts '{create_date} 23:59:59'} order by movement.create_stamp	ETL/Sum
Closing Cost	select top 1 new_inventory_qty, item_cost closing_cost from movement where movement.site_code = {movement.site_code} and movement.cost_center_code = {movement.cost_center_code} and movement.location_code = {movement.location_code} and movement.item_num = {movement.item_num} and movement.movement_type != '1' and movement.movement_type != '8' and (old_inventory_qty != new_inventory_qty or (old_inventory_qty = new_inventory_qty and movement_qty = 0)) and movement.create_stamp between {ts '{create_date} 00:00:00'} and {ts '{create_date} 23:59:59'} order by movement.create_stamp desc	ETL/Sum
Average Closing Cost	Closing Cost	Average
Total Usage	if ((Movement Type Code = '3' or Movement Type Code = '4' or Movement Type Code = '5')) THEN (Quantity) else (null)	Calculated
Quantity Usage Rate	abs(Total Usage) / (Closing On-Hand Quantity + abs(Total Usage))	
Total Usage Cost	if ((Movement Type Code = '3' or Movement Type Code = '4' or Movement Type Code = '5')) THEN (Quantity * Unit Cost) else (null)	Calculated
Turnover	abs(Total Usage Cost) /(Average Closing Cost)	Calculated

Name	Value	ETL/Aggregate
Volatility	Closing On-Hand Quantity	Variance

7.3.22 SMS Supply Performance Dimension Names

- Time
- Area
- Case
- Cost Center
- Item
- Item Family
- Item Specialty
- Location
- Manufacturer
- Movement Type
- Operator
- Physician
- Procedure
- Inventory Adjustment Reason
- Site
- Unit
- Vendor
- Waste Reason

8 TMS Shipping Performance

8.1 Structure

8.2 Mapping

8.2.1 Columns for the BI_CP_SHIP Table

Name	Value Coming From	Fact
EXTERNAL_REFERENCE	shipment.shipment_id	
SITE_CODE	shipment.site_code	
CP_CARR_KEY	shipment.carrier_code '!' shipment.service_code	
CP_CCEN_KEY	shipment.cost_center_code	can be null
CP_LOCA_KEY	shipment.consignee_country_code '!' shipment.consignee_stat_prov_code '!' shipment.consignee_city '!' shipment.consignee_zip_postal '!' shipment.consignee_num	consignee_num can be null
CP_SITE_KEY	shipment.site_code	
CP_TIME_KEY	manifest.create_stamp[1,14] + '00 :00'	[YYYY-MM-DD HH:00:00]
CP_OPER_KEY	shipment.create_user	
CP_SVTP_KEY	null	To be developed.
PACKAGE_COUNT	shipment.total_packages	
WEIGHT_LBS	shipment.total_weight_lbs	
WEIGHT_KG	shipment.total_weight_lbs	
VOLUME_CUBIC_FEET	shipment.total_volume_cubic_feet	
VOLUME_CUBIC_METERS	shipment.total_volume_cubic_meters	
PRE_CONSOL_BASE_RATE	shipment.pre_consol_base_rate	
PRE_CONSOL_FUEL_SURCHARGE	shipment.pre_consol_fuel_surcharge	
PRE_CONSOL_SURCHARGES	shipment.pre_consol_surcharges	
PRE_CONSOL_FEE_AMT	shipment.pre_consol_fee_amt	
PRE_CONSOL_DISCOUNT_AMT	shipment.pre_consol_discount_amt	
PRE_CONSOL_TAX_VALUE	shipment.pre_consol_tax_value	
PRE_CONSOL_TOTAL_RATE	shipment.pre_consol_total_rate	

Name	Value Coming From	Fact
CONSOL_BASE_RATE	shipment.consol_base_rate	
CONSOL_FUEL_SURCHARGE	shipment.consol_fuel_surcharge	
CONSOL_SURCHARGES	shipment.consol_surcharges	
CONSOL_FEE_AMT	shipment.consol_fee_amt	
CONSOL_DISCOUNT_AMT	shipment.consol_discount_amt	
CONSOL_TAX_VALUE	shipment.consol_tax_value	
CONSOL_TOTAL_RATE	shipment.consol_total_rate	
APPOINTMENT_DELIVERY_FEE	sum(shipment_fee.pre_consol_fee_amt) + sum(shipment_package_fee. pre_consol_fee_amt)	fee.associated_option_code **APPOINTMENTDELIVERY** **Note 1**
CHAIN_OF_SIGNATURE_FEE	sum(shipment_fee.pre_consol_fee_amt) + sum(shipment_package_fee. pre_consol_fee_amt)	fee.associated_option_code **CHAINOFSIGNATURE** **Note 1**
COD_FEE	sum(shipment_fee.pre_consol_fee_amt) + sum(shipment_package_fee. pre_consol_fee_amt)	fee.associated_option_code COD **Note 1**
DECLARED_VALUE_FEE	sum(shipment_fee.pre_consol_fee_amt) + sum(shipment_package_fee. pre_consol_fee_amt)	fee.associated_option_code **DECLAREDVALUE** **Note 1**
DELIVERY_CONF_FEE	sum(shipment_fee.pre_consol_fee_amt) + sum(shipment_package_fee. pre_consol_fee_amt)	fee.associated_option_code **DELIVERYCONFIRMATION** **Note 1**
HOLD_LOCATION_FEE	sum(shipment_fee.pre_consol_fee_amt) + sum(shipment_package_fee.pre_consol_fee_amt)	fee.associated_option_code **HOLDATLOCATION** **Note 1**
INSIDE_DELIVERY_FEE	sum(shipment_fee.pre_consol_fee_amt) + sum(shipment_package_fee.pre_consol_fee_amt)	fee.associated_option_code **INSIDEDELIVERY** **Note 1**
LIFTGATE_DELIVERY_FEE	sum(shipment_fee.pre_consol_fee_amt) + sum(shipment_package_fee. pre_consol_fee_amt)	fee.associated_option_code **LIFTGATEDELIVERY** **Note 1**
RETURN_DELIVERY_FEE	sum(shipment_fee.pre_consol_fee_amt) + sum(shipment_package_fee. pre_consol_fee_amt)	fee.associated_option_code **RETURNDELIVERY** **Note 1**

Name	Value Coming From	Fact
SIGNATURE_RELEASE_FEE	sum(shipment_fee.pre_consol_fee_amt) + sum(shipment_package_fee. pre_consol_fee_amt)	fee.associated_option_code **SIGNATURERELEASE** **Note 1**
SHIP_HOURS	(manifest.create_stamp - shipment.required_ship_date) * 24	
INTERNAL_CYCLE_TIME_MINUTES	(manifest.create_stamp - shipment.create_stamp)	
EXTERNAL_CYCLE_TIME_MINUTES	0	To be developed.
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		
CP_CONS_KEY	shipment.billto_num + "!" + shipment.consignee_num	

Note 1

Select sum(pre_consol_fee_amt) shipment_fee_amt from shipment_fee
where shipment_fee.shipment_id = ? and fee_code in
(select fee_code from fee where shipment_fee.site_code = fee.site_code
and fee.associated_option_code = **?**)
union all "
select sum(pre_consol_fee_amt) shipment_package_fee_amt from shipment_package_fee
where shipment_package_fee.shipment_id = ? and fee_code in
(select fee_code from fee where shipment_package_fee.site_code = fee.site_code
and fee.associated_option_code = **?**)"

Read records from manifest, shipment where

```
from manifest, shipment package, service, site shipper, shipment
     left outer join dev_trunkbi2_m.md_country
                on shipment.consignee_country_code = md_country.country_code
     left outer join dev_trunkbi2_m.md_country_state_prov
                on shipment.consignee state prov code =
                   md_country_state_prov.state_prov_code
                 and shipment.consignee country code =
                   md_country_state_prov.country_code
where manifest.closeout_status = 999
  and manifest.carrier_code = shipment_package.carrier_code
  and manifest.carrier module identifier =
      shipment_package.carrier_module_identifier
  and manifest.manifest date = shipment package.manifest date
  and manifest.manifest_seq = shipment_package.manifest_seq
  and shipment_package.shipment_id = shipment.shipment_id
  and shipment.site_code = site_shipper.site_code
  and shipment.shipper code = site shipper.shipper code
  and site_shipper.is_default = 1
  and manifest.mod stamp <=TIMESTAMP '2015-02-04 13:31:54.0'
  and not exists
```

```
    (select bi_cp_ship.external_reference from dev_trunkbi2_a.bi_cp_ship
      where bi cp ship.external reference = shipment.shipment id)
union
from shipment package, service, site shipper, shipment
     left outer join dev_trunkbi2_m.md_country
                  on shipment.consignee_country_code = md_country.country_code
     left outer join dev_trunkbi2_m.md_country_state_prov
     on shipment.consignee_state_prov_code =
        md_country_state_prov.state_prov_code
      and shipment.consignee country code = md country state prov.country code
where shipment.shipment_status = 1
  and shipment_package.shipment_id = shipment.shipment_id
  and shipment.site_code = site_shipper.site_code
  and shipment.shipper_code = site_shipper.shipper_code
  and site shipper.is default = 1
  and service.site_code = shipment.site_code
  and service.service_code = shipment.service_code
  and (service.is_ltl = 1 or service.is_truckload = 1)
  and shipment.mod_stamp <=TIMESTAMP '2015-02-04 13:31:54.0'
  and not exists (select bi_cp_ship.external_reference from
                         dev trunkbi2 a.bi cp ship
                   where bi_cp_ship.external_reference = shipment.shipment_id)
order by shipment_id
```

8.2.2 Columns for the BI_CP_CARR Table

Name	Value Coming From	Dimension
CP_CARR_KEY (PK)	service.carrier_code '!' service.service_code	
SITE_CODE (PK)	service.site_code	
CARRIER_CODE	service.carrier_code	
CARRIER_DESC_1	carrier.desc_1	
CARRIER_DESC_2	carrier.desc_2	
SERVICE_CODE	service.service_code	
SERVICE_DESC_1	service.desc_1	
SERVICE_DESC_2	service.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

8.2.3 Columns for the BI_CP_CCEN Table

Name	Value Coming From	Dimension
CP_CCEN_KEY (PK)	cost_center.cost_center_code	
SITE_CODE (PK)	cost_center.site_code	
COST_CENTER_CODE	cost_center.cost_center_code	
COST_CENTER_DESC_1	cost_center.desc_1	
COST_CENTER_DESC_2	cost_center.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

8.2.4 Columns for the BI_CP_CONS Table

Name	Value Coming From	Dimension
CP_CONS_KEY (PK)	shipment.billto_num + "!" + shipment.consignee_num	
SITE_CODE (PK)	shipment.site_code	
BILL_TO_CODE	shipment.billto_num	
BILL_TO_NAME	billto.billto_name	
CONSIGNEE_CODE	shipment.consignee_num	
CONSIGNEE_NAME	consignee.consignee_name	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Read records from manifest, shipment where

- manifest.mod_stamp > (bi_dmr_att.dmr_att_value and dmr_name = 'carrier-performance' and dmr_att_key = 'last_datetime_processed') and manifest.mod_stamp <= Today
- shipment.closeout_status = '999'
- site_shipper.is_default = 1

8.2.5 Columns for the BI_CP_LOCA Table

Name	Value Coming From	Dimension
CP_LOCA_KEY (PK)	tms_shipment.consignee_country_code '!' tms_shipment.consignee_stat_prov_code '!' tms_shipment.consignee_city '!' tms_shipment.consignee_zip_postal	
SITE_CODE (PK)	tms_shipment.site_code	
COUNTRY_CODE	tms_shipment.consignee_country_code	
COUNTRY_DESC_1	md_country.desc_key***{locale1}***	**Note 1**
COUNTRY_DESC_2	md_country.desc_key***{locale 2}***	**Note 1**
STATE_CODE	tms_shipment.consignee_stat_prov_code	
STATE_DESC_1	md_country_state_prov.desc_key***{locale1}***	**Note 2**
STATE_DESC_2	md_country_state_prov.desc_key***{locale2}***	**Note 2**

Name	Value Coming From	Dimension
CITY_CODE	tms_shipment.consignee_city	
CITY_DESC_1	tms_shipment.consignee_city	
CITY_DESC_2	tms_shipment.consignee_city	
POSTAL_CODE	tms_shipment.consignee_zip_postal	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Note 1

- Select shipment.consignee_country_code, ***{md_country.desc_key}***
  from shipment
  left outer join md_country on shipment.consignee_country_code =
  md_country.country_code

- Select lit_text from md_locale_text where lit_key = ***{md_country.desc_key}***
  and locale_key = ***{locale1} / {locale2}***

Note 2

- Select shipment.consignee_state_prov_code, ***{md_country_state_prov.desc_key}***
  from shipment
  left outer join md_country_state_prov on shipment.consignee_state_prov_code =
  md_country_state_prov.state_prov_code
  and shipment.consignee_country_code =
  md_country_state_prov.country_code
- Select lit_text from md_locale_text where lit_key = ***{md_country_state_prov.desc_key}***
  and locale_key = ***{locale1} / {locale2}***

8.2.6 Columns for the BI_CP_OPER Table

Name	Value Coming From	Dimension
CP_OPER_KEY (PK)	tms_shipment.create_user	
SITE_CODE (PK)	tms_shipment.site_code	
OPERATOR_CODE	tms_shipment.create_user	
OPERATOR_NAME	tms_shipment.create_user	
CUSTOM_CHAR_1		

Name	Value Coming From	Dimension
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

8.2.7 Columns for the BI_CP_SITE Table

Name	Value Coming From	Dimension
CP_SITE_KEY (PK)	tms_site.site_code	
SITE_CODE (PK)	tms_site.site_code	
SITE_DESC_1	tms_site.desc_1	
SITE_DESC_2	tms_site.desc_2	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

8.2.8 Columns for the BI_CP_SVTP Table

Name	Value Coming From	Dimension
CP_SVTP_KEY (PK)		
SITE_CODE (PK)		
SERVICE_TYPE_CODE		
SERVICE_TYPE_DESC_1		
SERVICE_TYPE_DESC_2		
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

8.2.9 Columns for the BI_CP_TIME Table

Name	Value Coming From	Dimension
CP_TIME_KEY (PK)	tms_manifest.create_stamp	**Note 1** [YYYY-MM-DD HH:00:00]
SITE_CODE (PK)	tms_shipment.site_code	
CAL_YEAR	cp_time_key[1,4]	[YYYY]
CAL_YEAR_CHAR	cp_time_key[1,4]	[YYYY]
CAL_YEAR_TO_DATE	'1' '2'	where cal_date >= lytd Start Date and >= lytd End Date where cal_date >= ytd Start Date and >= ytd End Date
CAL_YEAR_TO_DATE_DESC	'Last Year-To-Date' 'Year-To-Date'	where cal_date >= lytd Start Date and >= lytd End Date where cal_date >= ytd Start Date and >= ytd End Date
CAL_MONTH	cp_time_key[6,2]	[MM]
CAL_MONTH_CHAR	cp_time_key[6,2]	[MM]
CAL_MONTH_TO_DATE	'1' '2' '3'	where cal_date >= lymtd Start Date and >= lymtd End Date where cal_date >= lmtd Start Date and >= lmtd End Date where cal_date >= mtd Start Date and >= mtd End Date
CAL_MONTH_TO_DATE_DESC	'Last Year Month-To-Date' 'Last Month-To-Date' 'Month-To-Date'	where cal_date >= lymtd Start Date and >= lymtd End Date where cal_date >= lmtd Start Date and >= lmtd End Date where cal_date >= mtd Start Date and >= mtd End Date
CAL_WEEK_YEAR	Week Start Date[1,4]	**Note 2** [YYYY]
CAL_WEEK_YEAR_CHAR	Week Start Date[1,4]	**Note 2** [YYYY]
CAL_WEEK_START_DATE	Week Start Date	**Note 2**
CAL_WEEK_START_DATE_CHAR	Week Start Date	**Note 2**
CAL_WEEK_TO_DATE	'1' '2'	where cal_date >= lwtd Start Date and >= lwtd End Date where cal_date >= wtd Start Date and >= wtd End Date

Name	Value Coming From	Dimension
CAL_WEEK_TO_DATE_DESC	'Last-Week-To-Date' 'Week-To-Date'	where cal_date >= lwtd Start Date and >= lwtd End Date where cal_date >= wtd Start Date and >= wtd End Date
CAL_DATE	cp_time_key[1,10]	[YYYY-MM-DD]
CAL_DATE_CHAR	cp_time_key[1,10]	[YYYY-MM-DD]
CAL_HOUR	cp_time_key[12,2]	[HH]
CAL_HOUR_CHAR	cp_time_key[12,2]	[HH]
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Note 1

- Select min(manifest.create_stamp) from manifest where manifest.closeout_status = '999'
- Select max(manifest.create_stamp) from manifest where manifest.closeout_status = '999'
- For loop on the earliest create stamp(min) year starting on the 1st of January to the latest create stamp(max) year ending on the 31st of December
- Each date is defined per 24 hours.
- If the min year = max year will start 1 year earlier.

Note 2

Determine the start date of the week of the current cp_time_key assuming Monday as the Start Day in Week.

8.2.10 TMS Shipping Measures Definition

8.2.10.1 bi_cp_ship Fact Table

Name	Value	ETL/Aggregate
	from manifest, shipment_package, site_shipper, shipment left outer join *metadb*.dbo.md_country on shipment.consignee_country_code = md_country.country_code left outer join *metadb*.dbo.md_country_state_prov on shipment.consignee_state_prov_code = md_country_state_prov.state_prov_code and shipment.consignee_country_code = md_country_state_prov.country_code where manifest.closeout_status = 999 and manifest.carrier_code = shipment_package.carrier_code and manifest.carrier_module_identifier = shipment_package.carrier_module_identifier and manifest.manifest_date = shipment_package.manifest_date and manifest.manifest_seq = shipment_package.manifest_seq and shipment_package.shipment_id = shipment.shipment_id and shipment.site_code = site_shipper.site_code and shipment.shipper_code = site_shipper.shipper_code and site_shipper.is_default = 1 and manifest.mod_stamp > {last time processed} and manifest.mod_stamp <= {Today's Timestamp} and not exists (select bi_cp_ship.external_reference from *bidb*.dbo.bi_cp_ship where bi_cp_ship.external_reference = shipment.shipment_id) order by shipment.shipment_id	

Name	Value	ETL/Aggregate
External Reference	shipment.shipment_id	ETL
Package Count	shipment.total_packages	ETL/Sum
Weight - lb	shipment.total_weight_lbs	ETL/Sum
Weight - kg	shipment.total_weight_kg	ETL/Sum
Volume - ft3	shipment.total_volume_cubic_feet	ETL/Sum
Volume - m3	shipment.total_volume_cubic_meters	ETL/Sum
Base Rate	shipment.pre_consol_base_rate	ETL/Sum
Surcharge - Fuel	shipment.pre_consol_fuel_surcharge	ETL/Sum
Surcharge - Other	shipment.pre_consol_surcharges	ETL/Sum
Package Fees	shipment.pre_consol_fee_amt	ETL/Sum
Discount Amount	shipment.pre_consol_discount_amt	ETL/Sum
Tax Value	shipment.pre_consol_tax_value	ETL/Sum
Total Package Rate	shipment.pre_consol_total_rate	ETL/Sum
Consolidated Base Rate	shipment.consol_base_rate	ETL/Sum
Consolidated Surcharge - Fuel	shipment.consol_fuel_surcharge	ETL/Sum
Consolidated Surcharge - Other	shipment.consol_surcharges	ETL/Sum
Consolidated Fee Amount	shipment.consol_fee_amt	ETL/Sum
Consolidated Discount Amount	shipment.consol_discount_amt	ETL/Sum
Consolidated Tax Value	shipment.consol_tax_value	ETL/Sum
Consolidated Total Rate	shipment.consol_total_rate	ETL/Sum
Appointment Delivery Fee	sum(shipment_fee.pre_consol_fee_amt) where shipment_fee.fee_code in (select fee_code from fee where shipment_fee.site_code = fee.site_code and shipment_fee.site_code = fee.site_code and fee.associated_option_code = 'APPOINTMENTDELIVERY' + sum(shipment_package_fee.pre_consol_fee_amt) where shipment_package_fee.fee_code in (select fee_code from fee where shipment_package_fee.site_code = fee.site_code and shipment_package_fee.site_code = fee.site_code and fee.associated_option_code = 'APPOINTMENTDELIVERY'	ETL/Sum

Name	Value	ETL/Aggregate
Chain of Signature Fee	sum(shipment_fee.pre_consol_fee_amt) where shipment_fee.fee_code in (select fee_code from fee where shipment_fee.site_code = fee.site_code and shipment_fee.site_code = fee.site_code and fee.associated_option_code = 'CHAINOFSIGNATURE' + sum(shipment_package_fee.pre_consol_fee_amt) where shipment_package_fee.fee_code in (select fee_code from fee where shipment_package_fee.site_code = fee.site_code and shipment_package_fee.site_code = fee.site_code and fee.associated_option_code = 'CHAINOFSIGNATURE'	ETL/Sum
COD Fee	sum(shipment_fee.pre_consol_fee_amt) where shipment_fee.fee_code in (select fee_code from fee where shipment_fee.site_code = fee.site_code and shipment_fee.site_code = fee.site_code and fee.associated_option_code = 'COD' + sum(shipment_package_fee.pre_consol_fee_amt) where shipment_package_fee.fee_code in (select fee_code from fee where shipment_package_fee.site_code = fee.site_code and shipment_package_fee.site_code = fee.site_code and fee.associated_option_code = 'COD'	ETL/Sum
Declared Value Fee	sum(shipment_fee.pre_consol_fee_amt) where shipment_fee.fee_code in (select fee_code from fee where shipment_fee.site_code = fee.site_code and shipment_fee.site_code = fee.site_code and fee.associated_option_code = 'DECLAREDVALUE' + sum(shipment_package_fee.pre_consol_fee_amt) where shipment_package_fee.fee_code in (select fee_code from fee where shipment_package_fee.site_code = fee.site_code and shipment_package_fee.site_code = fee.site_code and fee.associated_option_code = 'DECLAREDVALUE'	ETL/Sum

Name	Value	ETL/Aggregate
Delivery Confirmation Fee	sum(shipment_fee.pre_consol_fee_amt) where shipment_fee.fee_code in (select fee_code from fee where shipment_fee.site_code = fee.site_code and shipment_fee.site_code = fee.site_code and fee.associated_option_code = 'DELIVERYCONFIRMATION' + sum(shipment_package_fee.pre_consol_fee_amt) where shipment_package_fee.fee_code in (select fee_code from fee where shipment_package_fee.site_code = fee.site_code and shipment_package_fee.site_code = fee.site_code and fee.associated_option_code = 'DELIVERYCONFIRMATION'	ETL/Sum
Hold at Location Fee	sum(shipment_fee.pre_consol_fee_amt) where shipment_fee.fee_code in (select fee_code from fee where shipment_fee.site_code = fee.site_code and shipment_fee.site_code = fee.site_code and fee.associated_option_code = 'HOLDATLOCATION' + sum(shipment_package_fee.pre_consol_fee_amt) where shipment_package_fee.fee_code in (select fee_code from fee where shipment_package_fee.site_code = fee.site_code and shipment_package_fee.site_code = fee.site_code and fee.associated_option_code = 'HOLDATLOCATION'	ETL/Sum
Inside Delivery Fee	sum(shipment_fee.pre_consol_fee_amt) where shipment_fee.fee_code in (select fee_code from fee where shipment_fee.site_code = fee.site_code and shipment_fee.site_code = fee.site_code and fee.associated_option_code = 'INSIDEDELIVERY' + sum(shipment_package_fee.pre_consol_fee_amt) where shipment_package_fee.fee_code in (select fee_code from fee where shipment_package_fee.site_code = fee.site_code and shipment_package_fee.site_code = fee.site_code and fee.associated_option_code =	ETL/Sum

Name	Value	ETL/Aggregate
	'INSIDEDELIVERY'	
Liftgate Delivery Fee	sum(shipment_fee.pre_consol_fee_amt) where shipment_fee.fee_code in (select fee_code from fee where shipment_fee.site_code = fee.site_code and shipment_fee.site_code = fee.site_code and fee.associated_option_code = 'LIFTGATEDELIVERY' + sum(shipment_package_fee.pre_consol_fee_amt) where shipment_package_fee.fee_code in (select fee_code from fee where shipment_package_fee.site_code = fee.site_code and shipment_package_fee.site_code = fee.site_code and fee.associated_option_code = 'LIFTGATEDELIVERY'	ETL/Sum
Return Delivery Fee	sum(shipment_fee.pre_consol_fee_amt) where shipment_fee.fee_code in (select fee_code from fee where shipment_fee.site_code = fee.site_code and shipment_fee.site_code = fee.site_code and fee.associated_option_code = 'RETURNDELIVERY' + sum(shipment_package_fee.pre_consol_fee_amt) where shipment_package_fee.fee_code in (select fee_code from fee where shipment_package_fee.site_code = fee.site_code and shipment_package_fee.site_code = fee.site_code and fee.associated_option_code = 'RETURNDELIVERY'	ETL/Sum

Name	Value	ETL/Aggregate

Signature Release Fee	sum(shipment_fee.pre_consol_fee_amt) where shipment_fee.fee_code in
	(select fee_code from fee
	  where shipment_fee.site_code = fee.site_code
	  and shipment_fee.site_code = fee.site_code
	  and fee.associated_option_code = 'SIGNATURERELEASE'
	+
	sum(shipment_package_fee.pre_consol_fee_amt) where shipment_package_fee.fee_code in
	(select fee_code from fee
	  where shipment_package_fee.site_code = fee.site_code
	  and shipment_package_fee.site_code = fee.site_code
	  and fee.associated_option_code = 'SIGNATURERELEASE'
		ETL/Sum

Shipment Time in Hours	(manifest.create_stamp - shipment.required_ship_date) * 24	ETL/Sum
Internal Cycle Time in Minutes	(manifest.create_stamp - shipment.create_stamp) / 1000/ 60	ETL/Sum
External Cycle Time in Minutes	0	ETL/Sum
Shipped on Time	if (Shipment Time in Hours > 0) then 0(false) else 1(true)	Calculated
Number of Shipments	Count(distinct(External Reference))	Calculated
Shipped on Time Rate	Shipped on Time / Number of Shipments	Calculated
Packages per Shipment	Package Count / Number of Shipments	Calculated
Internal Cycle Time in Hours	Internal Cycle Time in Minutes / 60	Average
External Cycle Time in Hours	External Cycle Time in Minutes / 60	Average
Total Weight - lb	Total(Weight - lb)	Calculated
Cost per Weight - lb	Consolidated Total Rate / Weight - lb)	Calculated
Total Weight - kg	Total(Weight - kg)	Calculated
Cost Per Weight - kg	Consolidated Total Rate / Weight - kg)	Calculated
Cost per Shipment	Consolidated Total Rate / Number of Shipments	Calculated

8.2.11 TMS Shipping Performance Dimension Names

- Time
- Carrier
- Cost Center
- Service Type
- Geography
- Operator
- Site
- Consignee

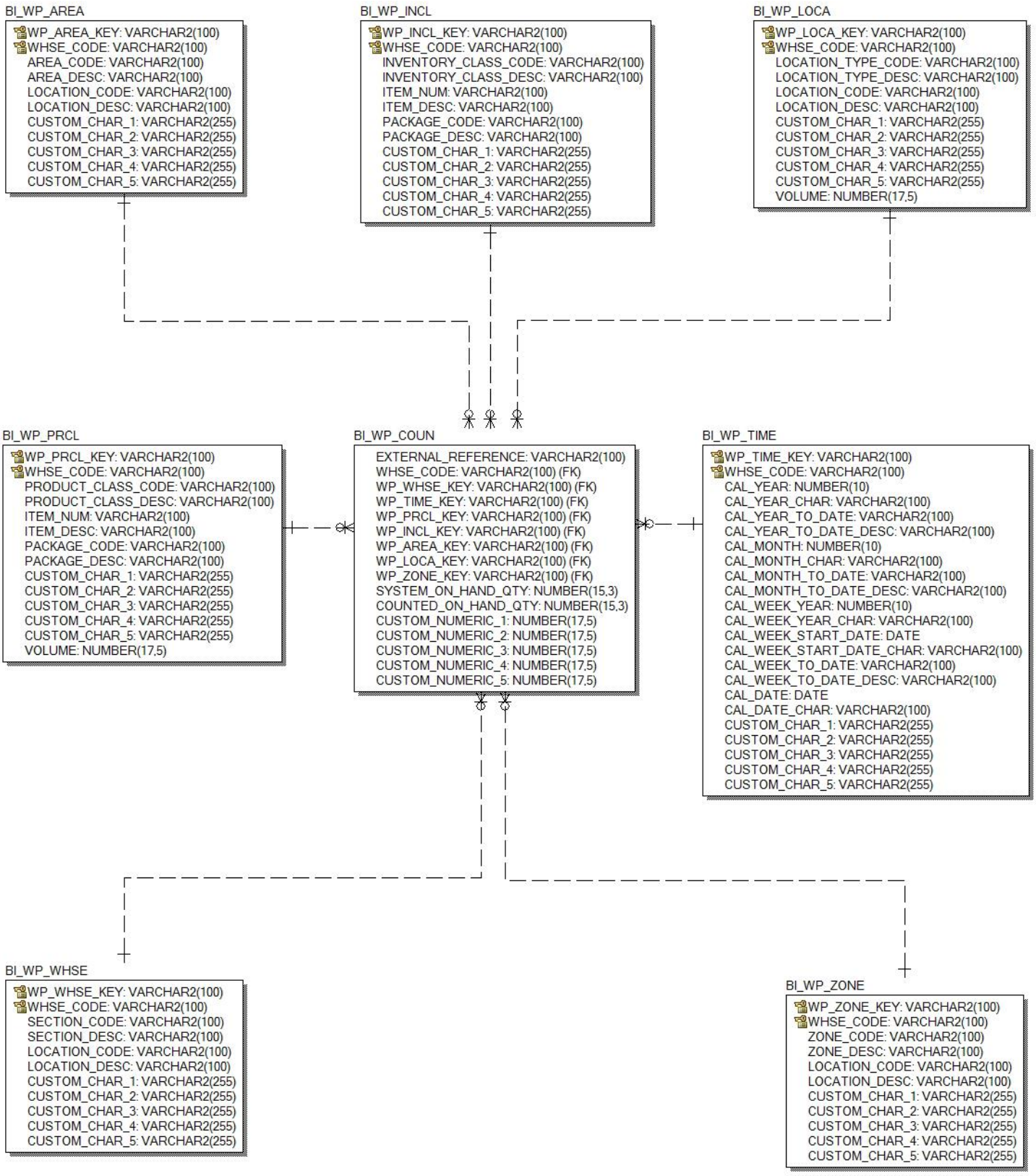
9 WMS Warehouse Performance

9.1 Structure - WERC Benchmarks

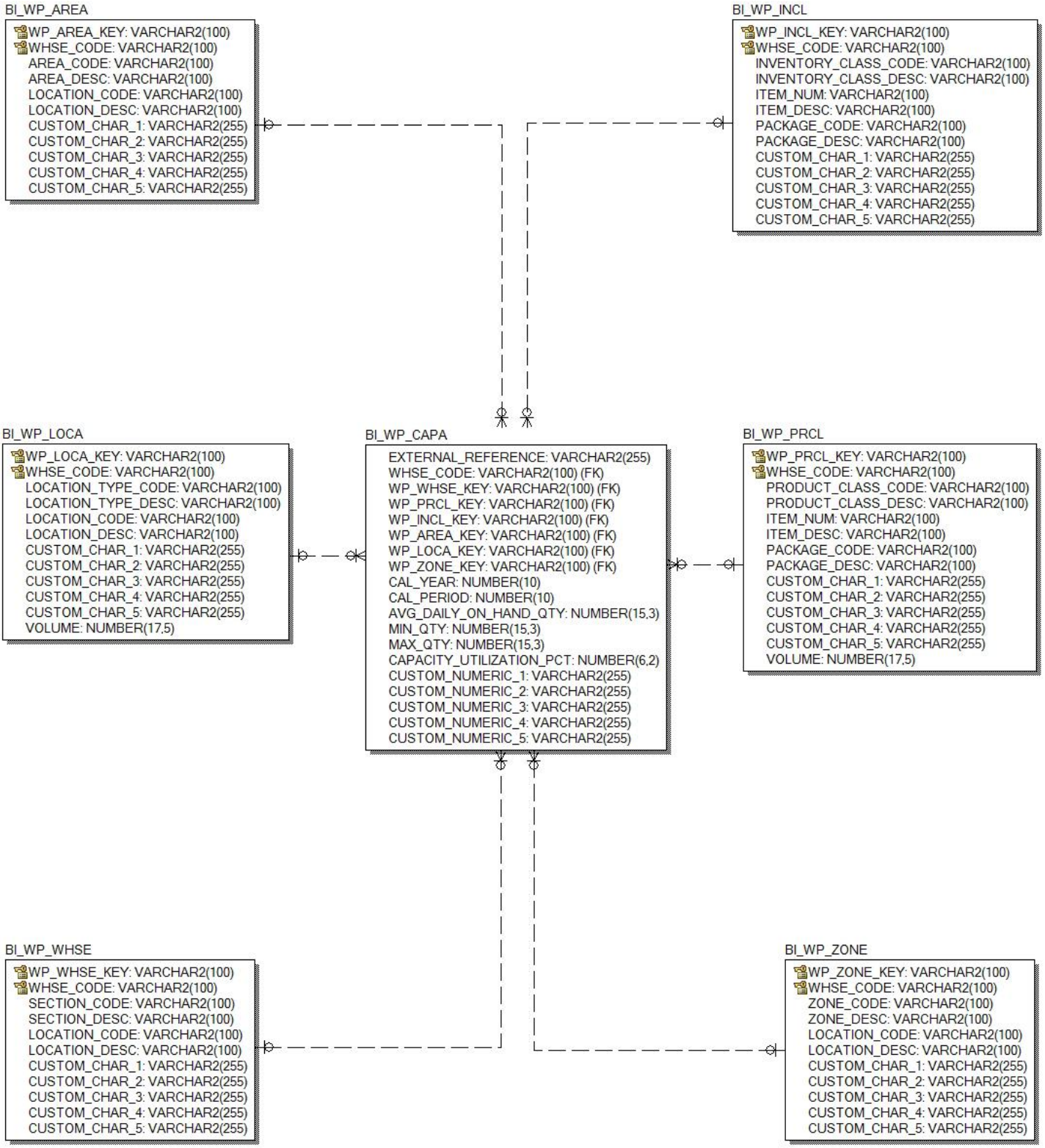
9.2 Structure - Order Lines

9.3 Structure - Receipt Lines

9.4 Structure - Cycle Counts



9.5 Structure - Capacity



9.6 MAPPING

9.6.1 Columns for the BI_WP_COUN Table

Name	Value Coming From	Fact
EXTERNAL_REFERENCE	it_f.it_rid '!' it_f.cycc_oid	
WHSE_CODE	so_f.whse_code	
WP_WHSE_KEY	it_f.from_loc	
WP_TIME_KEY	it_f.end_time[1,10]	[YYYY-MM-DD]
WP_PRCL_KEY	it_f.sku '!' it_f.pkg	
WP_INCL_KEY	it_f.sku '!' it_f.pkg	
WP_AREA_KEY	it_f.from_loc	
WP_LOCA_KEY	it_f.from_loc	
WP_ZONE_KEY	it_f.from_loc	
SYSTEM_ON_HAND_QTY	it_f.qty	
COUNTED_ON_HAND_QTY	it_f.act_qty	
CUSTOM_NUMERIC_1		
CUSTOM_NUMERIC_2		
CUSTOM_NUMERIC_3		
CUSTOM_NUMERIC_4		
CUSTOM_NUMERIC_5		

Read records from it_f where

- it_f.it_rid > (bi_dmr_att.dmr_att_value where dmr_name = 'warehouse-performance' and dmr_att_key = 'bi_wp_coun!last_it_rid!' + *warehouse code*) and it_f.it_rid <= max(it_rid)
- transact = 'CYCC' and rsn_code = 'CYCC'

9.6.2 Columns for the BI_WP_ORDL Table

Name	Value Coming From	
EXTERNAL_REFERENCE	it_f.ob_oid '!' it_f.ob_type '!' it_f.ob_lno	
WHSE_CODE	so_f.whse_code	
WP_WHSE_KEY	it_f.from_loc	if from_loc blank than '_'

Name	Value Coming From	
WP_TIME_KEY	it_f.end_time[1,10]	[YYYY-MM-DD]
WP_PRCL_KEY	it_f.sku '!' it_f.pkg	
WP_INCL_KEY	it_f.sku '!' it_f.pkg	
WP_ORDH_KEY	it_f.ob_oid '!' it_f.ob_type	
TOTAL_ORDER_CYCLE_MINUTES	it_f.end_time - om_h.ord_date - number of days off	**Note 1**
INTERNAL_ORDER_CYCLE_MINUTES	it_f.end_time - om_h.wms_ord_dtimecre - number of days off	**Note 1**
ORDER_QTY	od_h.ord_qty	
FILLED_QTY	od_h.cmp_qty	
SHIPPED_ON_TIME	it_f.end_time > om_h.request_date than 0 else 1	
WP_SHIP_KEY	om_h.bill_custnum ' !' om_h.ship_custnum	
CUSTOM_NUMERIC_1		
CUSTOM_NUMERIC_2		
CUSTOM_NUMERIC_3		
CUSTOM_NUMERIC_4		
CUSTOM_NUMERIC_5		
PICK_SHIP_MINUTES	max(end_time) from it_f where transact = 'SHIP' – min(it_f.start_time) for transact = 'PICK' - number of days off	

Note 1
The number of days off correspond to the number of md_calendar_day.is_work_day is 0 for the so_f.operations_calendar_code

Read records from it_f, om_h, od_h where

- it_f.it_rid > (bi_dmr_att.dmr_att_value where dmr_name = 'warehouse-performance' and dmr_att_key = 'bi_wp_ordl!last_it_rid!' + *warehouse code*) and it_f.it_rid <= max(it_rid)

transact = 'SHIP'

9.6.3 Columns for the BI_WP_RCPL Table

Name	Value Coming From	
EXTERNAL_REFERENCE	it_f.tag "!" it_f.supp_num	
WHSE_CODE	so_f.whse_code	
WP_WHSE_KEY	max(it_f.to_loc)	if to_loc blank than '_'
WP_TIME_KEY	max(it_f.end_time[1,10])	[YYYY-MM-DD]
WP_PRCL_KEY	it_f.sku '!' it_f.pkg	
WP_INCL_KEY	it_f.sku '!' it_f.pkg	
WP_AREA_KEY	max(it_f.to_loc)	if to_loc blank than '_'
WP_LOCA_KEY	max(it_f.to_loc)	if to_loc blank than '_'
WP_ZONE_KEY	max(it_f.to_loc)	if to_loc blank than '_'
WP_RCPH_KEY	it_f.receipt	
DOCK_TO_STOCK_MINUTES	max(it_f.end_time[1,10])[STOR transaction] – min(it_f.start_time)[RCPT transaction] – number of days off	**Note 1**
ON_TIME_RECEIPT	min(it_f.dt_rcvd)[RCPT transaction] > ibod_h.expect_date	True or false
CUSTOM_NUMERIC_1		
CUSTOM_NUMERIC_2		
CUSTOM_NUMERIC_3		
CUSTOM_NUMERIC_4		
CUSTOM_NUMERIC_5		
WP_SUPP_KEY	it_f.supp_num	

Note 1
The number of days off correspond to the number of md_calendar_day.is_work_day is 0 for the so_f.operations_calendar_code

Read records from it_f

- it_f.it_rid > (bi_dmr_att.dmr_att_value where dmr_name = 'warehouse-performance' and dmr_att_key = 'bi_wp_rcpl!last_it_rid!' + *warehouse code*) and it_f.it_rid <= max(it_rid)
- transact = 'RCPT'

9.6.4 Columns for the BI_WP_CAPA Table

Column Name	Value Coming From	Description/Literal/Key
external_reference	iv_f.whse_code + "!" + iv_f.loc + "!" + iv_f.sku + "!" + iv_f.pkg + "!" + getYear(sysdate) + "!" + getMonth(sysdate)	External Reference
whse_code	so_f.whse_code	Warehouse
wp_whse_key	iv_f.whse_code	
wp_prcl_key	iv_f.sku + "!"+ iv_f.pkg	
wp_incl_key	iv_f.sku + "!" + iv_f.pkg	
wp_area_key	lc_f.area	
wp_loca_key	iv_f.loc	
wp_zone_key	lc_f.zone	
cal_year	getYear(sysdate)	
cal_period	getMonth(sysdate)	
avg_daily_on_hand_qty	bi_wp_capa.avg_dail y_on_hand_qty + (sum(iv_f.qty) group by iv_f.sku, iv_f.pkg, iv_f.loc, iv_f.whse_code/ bi_dmr_att.dmr_att_ value bi_wp_capa!daysPro cessed + 1	
min_qty	iv_f.load_min_qty	
max_qty	iv_f.load_max_qty	
capacity_utilization_pct	lc_f.dpth * lc_f.wid * lc_f.hgt / (avg_daily_on_hand _qty * iv_f.dpth1 * iv_f.wid1 * iv_f.hgt1)	
custom_numeric_1		
custom_numeric_2		
custom_numeric_3		

Column Name	Value Coming From	Description/Literal/Key
custom_numeric_4		
custom_numeric_5		

9.6.5 Columns for the BI_WP_AREA Table

Name	Value Coming From	Dimension
WP_AREA_KEY (PK)	lc_f.loc	
WHSE_CODE (PK)	so_f.whse_code	
AREA_CODE	lc_f.area	
AREA_DESC	lc_f.area	
LOCATION_CODE	lc_f.loc	
LOCATION_DESC	lc_f.loc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.6 Columns for the BI_WP_INCL Table

Name	Value Coming From	Dimension
WP_INCL_KEY (PK)	pm_f.sku '!' pm_f.pkg	
WHSE_CODE (PK)	so_f.whse_code	
INVENTORY_CLASS_CODE	pm_f.inv_class	
INVENTORY_CLASS_DESC	pm_f.inv_class	
ITEM_NUM	pm_f.sku	
ITEM_DESC	pm_f.sku_desc	
PACKAGE_CODE	pm_f.pkg	
PACKAGE_DESC	pm_f.pkg_desc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		

Name	Value Coming From	Dimension
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.7 Columns for the BI_WP_LOCA Table

Name	Value Coming From	Dimension
WP_LOCA_KEY (PK)	lc_f.loc	
WHSE_CODE (PK)	so_f.whse_code	
LOCATION_TYPE_CODE	lc_f.loc_type	
LOCATION_TYPE_DESC	lc_f.loc_type	
LOCATION_CODE	lc_f.loc	
LOCATION_DESC	lc_f.loc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.8 Columns for the BI_WP_ORDH Table

Name	Value Coming From	Dimension
WP_ORDH_KEY (PK)	it_f.ob_oid '!' it_f.ob_type	
WHSE_CODE (PK)	so_f.whse_code	
ORDER_REF	it_f.ob_oid '!' it_f.ob_type	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.9 Columns for the BI_WP_PRCL Table

Name	Value Coming From	Dimension
WP_PRCL_KEY (PK)	pm_f.sku '!' pm_f.pkg	
WHSE_CODE (PK)	so_f.whse_code	
PRODUCT_CLASS_CODE	pm_f.prod_class	
PRODUCT_CLASS_DESC	pm_f.prod_class	
ITEM_NUM	pm_f.sku	
ITEM_DESC	pm_f.sku_desc	
PACKAGE_CODE	pm_f.pkg	
PACKAGE_DESC	pm_f.pkg_desc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.10 Columns for the BI_WP_RCPH Table

Name	Value Coming From	Dimension
WP_RCPH_KEY (PK)	it_f.receipt_num	
WHSE_CODE (PK)	so_f.whse_code	
RECEIPT_NUM	it_f.receipt_num	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.11 Columns for the BI_WP_SHIP Table

Name	Value Coming From	Dimension
WP_SHIP_KEY (PK)	om_h.bill_custnum '!' om_h.ship_custnum	**Note 1**

Name	Value Coming From	Dimension
WHSE_CODE (PK)	so_f.whse_code	
COUNTRY_CODE	cust_ship.country or om_h.ship_cntry	**Note 2**
COUNTRY_NAME	cust_ship.country or om_h.ship_cntry	
STATE_CODE	cust_ship.state_province or om_h.ship_state	**Note 2**
STATE_NAME	cust_ship.state_province or om_h.ship_state	
CITY_CODE	cust_ship.city or om_h.ship_city	**Note 2**
CITY_NAME	cust_ship.city or om_h.ship_city	
CUST_NUM	cust_ship.state_province or	
CUST_NAME	cust.cust_name or om_h.bill_name	**Note 2**
SHIP_NUM	om_h.ship_custnum	
SHIP_NAME	cust_ship.ship_name or om_h.ship_name	**Note 2**
SHIP_ZIP_CODE	cust_ship.zip_postal or om_h.ship_zip	**Note 2**
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Note 1

Select om_h.bill_cust_num, om_h.ship_custnum from om_h, it_f
where it_f.transact = 'SHIP'

Note 2

Select from cust, cust_ship and if exist set information from it otherwise take it from om_h.

9.6.12 Columns for the BI_WP_SUPP Table

Name	Value Coming From	Dimension
WP_SUPP_KEY (PK)	it_f.supp_num	
WHSE_CODE (PK)	so_f.whse_code	
SUPPLIER_NUM	it_f.supp_num	
SUPPLIER_NAME	it_f.supp_num	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Read records from it_f

- it_f.it_rid > (bi_dmr_att.dmr_att_value where dmr_name = 'warehouse-performance' and dmr_att_key = 'bi_wp_rcpl!last_it_rid!' + *warehouse code*) and it_f.it_rid <= max(it_rid)
- transact = 'RCPT'

9.6.13 Columns for the BI_WP_TIME Table

Name	Value Coming From	Dimension
WP_TIME_KEY (PK)	om_h.ord_date it_f.end_time	**Note 1** [YYYY-MM-DD]
WHSE_CODE (PK)	so_f.whse_code	so_f.so_rid = 1
CAL_YEAR	wp_time_key[1,4]	[YYYY]
CAL_YEAR_CHAR	wp_time_key[1,4]	[YYYY]
CAL_YEAR_TO_DATE	'1' '2'	where cal_date >= lytd Start Date and >= lytd End Date where cal_date >= ytd Start Date and >= ytd End Date
CAL_YEAR_TO_DATE_DESC	'Last Year-To-Date' 'Year-To-Date'	where cal_date >= lytd Start Date and >= lytd End Date where cal_date >= ytd Start Date and >= ytd End Date
CAL_MONTH	wp_time_key[6,2]	[MM]
CAL_MONTH_CHAR	wp_time_key[6,2]	[MM]

Name	Value Coming From	Dimension
CAL_MONTH_TO_DATE	'1' '2' '3'	where cal_date >= lymtd Start Date and >= lymtd End Date where cal_date >= lmtd Start Date and >= lmtd End Date where cal_date >= mtd Start Date and >= mtd End Date
CAL_MONTH_TO_DATE_DESC	'Last Year Month-To-Date' 'Last Month-To-Date' 'Month-To-Date'	where cal_date >= lymtd Start Date and >= lymtd End Date where cal_date >= lmtd Start Date and >= lmtd End Date where cal_date >= mtd Start Date and >= mtd End Date
CAL_WEEK_YEAR	Week Start Date[1,4]	**Note 2** [YYYY]
CAL_WEEK_YEAR_CHAR	Week Start Date[1,4]	**Note 2** [YYYY]
CAL_WEEK_START_DATE	Week Start Date	**Note 2**
CAL_WEEK_START_DATE_CHAR	Week Start Date	**Note 2**
CAL_WEEK_TO_DATE	'1' '2'	where cal_date >= lwtd Start Date and >= lwtd End Date where cal_date >= wtd Start Date and >= wtd End Date
CAL_WEEK_TO_DATE_DESC	'Last-Week-To-Date' 'Week-To-Date'	where cal_date >= lwtd Start Date and >= lwtd End Date where cal_date >= wtd Start Date and >= wtd End Date
CAL_DATE	wp_time_key[1,10]	[YYYY-MM-DD]
CAL_DATE_CHAR	wp_time_key[1,10]	[YYYY-MM-DD]
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Note 1

- Take the earliest(min) date between om_h.ord_date where it_f.transact = 'SHIP' and it_f.end_time where it_f = 'CYCC' and min(bi_wp_werc_year.werc year)
- Take the latest(max) date between om_h.ord_date where it_f.transact = 'SHIP' and it_f.end_time where it_f = 'CYCC' and max(bi_wp_werc_year.werc year)

- For loop on the earliest date year starting on the 1st of January to
  the latest date year ending on the 31st of December.
- If the min year = max year will start 1 year earlier.

Note 2
Determine the start date of the week of the current wp_time_key using the
Start Day in Week defined in the md_calendar for the so_f.operations_calendar_code

9.6.14 Columns for the BI_WP_WHSE Table

Name	Value Coming From	Dimension
WP_WHSE_KEY (PK)	lc_f.loc	A dummy '_' wp_whse_key is inserted in the case the it_f.from_loc is null(due toforeign key).
WHSE_CODE (PK)	so_f.whse_code	so_f.so_rid = 1
SECTION_CODE	lc_f.whse_section	
SECTION_DESC	lc_f.whse_section	
LOCATION_CODE	lc_f.loc	
LOCATION_DESC	lc_f.loc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.15 Columns for the BI_WP_ZONE Table

Name	Value Coming From	Dimension
WP_ZONE_KEY (PK)	lc_f.loc	
WHSE_CODE (PK)	so_f.whse_code	so_f.so_rid = 1
ZONE_CODE	lc_f.zone	
ZONE_DESC	lc_f.zone	
LOCATION_CODE	lc_f.loc	
LOCATION_DESC	lc_f.loc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		

Name	Value Coming From	Dimension
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.16 Columns for the BI_WP_WERC_METRIC Table

Name	Value Coming From	Benchmark
EXTERNAL_REFERENCE	werc-import.txt-ID	splitarray[0]
WP_WERC_CUSTOMERTYPE_KEY	werc-import.txt-CUSTOMER	splitarray[7]
WP_WERC_INDUSTRY_KEY	werc-import.txt-INDUSTRY	splitarray[5]
WP_WERC_SALES_KEY	werc-import.txt-SALES	splitarray[9]
WP_WERC_STRATEGY_KEY	werc-import.txt-STRATEGY	splitarray[3]
WP_WERC_YEAR_KEY	werc-import.txt-Year	splitarray[1]
PARTIAL_CASE_PICK_PCT	werc-import.PICK_PARTCASE	splitarray[11]
FULL_CASE_PICK_PCT	werc-import.PICK_FULLCASE	splitarray[12]
PARTIAL_PALLET_PICK_PCT	werc-import.PICK_PARTPALLET	splitarray[13]
FULL_PALLET_PICK_PCT	werc-import.PICK_FULLPALLET	splitarray[14]
ON_TIME_READY_TO_SHIP_PCT	werc-import.ON_TIME_RDY_SHIP	splitarray[15]
TOTAL_ORDER_CYCLE_HOURS	werc-import.TOT_ORD_CYCLE_TIME	splitarray[16]
INTERNAL_ORDER_CYCLE_HOURS	werc-import.INT_ORD_CYCLE_TIME	splitarray[17]
ORDER_FILL_RATE_PCT	werc-import.FILL_RATE_ORD	splitarray[18]
ORDER_LINE_FILL_RATE_PCT	werc-import.FILL_RATE_LINE	splitarray[19]
CYCLE_COUNT_ACCURACY_PCT	werc-import.CYCLE_COUNT_LOC	splitarray[20]
REFERENCE_YEAR	werc-import.txt-Reference Year	splitarray[2]
DOCK_TO_STOCK_HOURS	werc-import.DOCK_STOCK	splitarray[21]
ON_TIME_RECEIPTS_PCT	werc-import.ON_TIME_RCPTS	splitarray[22]
ORDER_PICK_SHIP_HOURS	werc-import.ORD_PICK_SHIP	splitarray[23]
ORDER_LINE_PICK_SHIP_HOURS	werc-import.ORD_L_PICK_SHIP	splitarray[24]
WHSE_CAPACITY_USED_PCT	werc-import.WHSE_CAP_USED	splitarray[25]

Name	Value Coming From	Benchmark
HONEYCOMB_PCT	werc-import.HONEYCOMB	splitarray[26]
CUSTOM_NUMERIC_1		
CUSTOM_NUMERIC_2		
CUSTOM_NUMERIC_3		
CUSTOM_NUMERIC_4		
CUSTOM_NUMERIC_5		

9.6.17 Columns for the BI_WP_WERC_CUSTOMERTYPE Table

Name	Value Coming From	Benchmark
WP_WERC_CUSTOMERTYPE_KEY (PK)	werc-import.txt-CUSTOMER	splitarray[7]
CUSTOMERTYPE_DESC	werc-import.txt-CUSTOMER_TYPE	splitarray[8]
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.18 Columns for the BI_WP_WERC_INDUSTRY Table

Name	Value Coming From	Benchmark
WP_WERC_INDUSTRY_KEY (PK)	werc-import.txt-INDUSTRY	splitarray[5]
INDUSTRY_DESC	werc-import.txt-INDUSTRY_DESC	splitarray[6]
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.19 Columns for the BI_WP_WERC_SALES Table

Name	Value Coming From	Benchmark
WP_WERC_SALES_KEY (PK)	werc-import.txt-SALES	splitarray[9]
SALES_DESC	werc-import.txt-SALES_DESC	splitarray[10]
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.20 Columns for the BI_WP_WERC_STRATEGY Table

Name	Value Coming From	Benchmark
WP_WERC_STRATEGY_KEY (PK)	werc-import.txt-STRATEGY	splitarray[3]
STRATEGY_DESC	werc-import.txt-STRATEGY_DESC	splitarray[4]
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.21 Columns for the BI_WP_WERC_YEAR Table

Name	Value Coming From	Benchmark
WP_WERC_YEAR_KEY (PK)	werc-import.txt-Year	splitarray[1]
CAL_YEAR	werc-import.txt-Year	splitarray[1]
CAL_YEAR_CHAR	werc-import.txt-Year	splitarray[1]
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

9.6.22 WMS Warehouse Performance - WERC Metric Measures Definition

9.6.22.1 bi_wp_werc_metric Fact Table

Name	Value	ETL/Aggregate

	// Open the file that is the first command line parameter
	FileInputStream fstream = new FileInputStream(fileName);
	// Get the object of DataInputStream
	in = new DataInputStream(fstream);
	BufferedReader br = new BufferedReader(new InputStreamReader(in));
	String strLine;
	boolean skipFirstLine = true;
	
	//Read File Line By Line
	while ((strLine = br.readLine()) != null) {
	if (skipFirstLine == true) {
	skipFirstLine = false;
	continue;
	}
	
	String splitarray[] = strLine.split("\t");
	int splitArrayLength = strLine.split("\t").length;
	if (splitArrayLength == 0) {
	continue;
	}	

External Reference	splitarray[0]	ETL
Reference Year	splitarray[2]	ETL
WERC Customer Type Key	splitarray[7]	ETL
WERC Industry Type Key	splitarray[5]	ETL
WERC Sales Key	splitarray[9]	ETL
WERC Strategy Key	splitarray[3]	ETL
WERC Year Key	splitarray[1]	ETL
Partial Case Pick Percent WERC	Relational Model: splitarray[11]	ETL/Median
Partial Case Pick Percent WERC	Dimensional Model: Partial Case Pick Percent WERC / 100	Median
Full Case Pick Percent WERC	Relational Model: splitarray[12]	ETL/Median
Full Case Pick Percent WERC	Dimensional Model: Full Case Pick Percent WERC / 100	Median

Name	Value	ETL/Aggregate
Partial Pallet Pick Percent WERC	Relational Model: splitarray[13]	ETL/Median
Partial Pallet Pick Percent WERC	Dimensional Model: Partial Pallet Pick Percent WERC / 100	Median
Full Pallet Pick Percent WERC	Relational Model: splitarray[14]	ETL/Median
Full Pallet Pick Percent WERC	Dimensional Model: Full Pallet Pick Percent WERC / 100	Median
On Time Ready to Ship Percent WERC	Relational Model: splitarray[15]	ETL/Median
On Time Ready to Ship Percent WERC	Dimensional Model: On Time Ready to Ship Percent WERC / 100	Median
Internal Order Cycle Time in Hours WERC	splitarray[17]	ETL/Median
Total Order Cycle Time in Hours WERC	splitarray[16]	ETL/Median
Order Fill Rate Percent WERC	Relational Model: splitarray[18]	ETL/Median
Order Fill Rate Percent WERC	Dimensional Model: Order Fill Rate Percent WERC / 100	Median
Order Line Fill Rate Percent WERC	Relational Model: splitarray[19]	ETL/Median
Order Line Fill Rate Percent WERC	Dimensional Model: Order Line Fill Rate Percent WERC / 100	Median
Cycle Count Accuracy Percent WERC	Relational Model: splitarray[20]	ETL/Median
Cycle Count Accuracy Percent WERC	Dimensional Model: Cycle Count Accuracy Percent WERC / 100	Median
Dock to Stock in Hours WERC	splitarray[21]	ETL/Median
On Time Receipts Percent WERC	Relational Model: splitarray[22]	ETL/Median
On Time Receipts Percent WERC	Dimensional Model: On Time Receipts Percent WERC / 100	Median
Orders Picked and Shipped in Hours WERC	splitarray[23]	ETL/Median
Order Lines Picked and Shipped in Hours WERC	splitarray[24]	ETL/Median

Name	Value	ETL/Aggregate
Warehouse Capacity Used Percent WERC	Relational Model: splitarray[25]	ETL/Median
Warehouse Capacity Used Percent WERC	Dimensional Model: Warehouse Capacity Used Percent WERC / 100	Median
Honeycomb Percent WERC	Relational Model: splitarray[26]	ETL/Median
Honeycomb Percent WERC	Dimensional Model: Honeycomb Percent WERC	Median

9.6.23 WMS Warehouse Performance - Cycle Count Measures Definition

9.6.23.1 bi_wp_coun Fact Table

Name	Value	ETL/Aggregate
	from it_f where it_f.transact = 'CYCC' and it_f.rsn_code = 'CYCC' and it_f.whse_code = *{whse}* and it_f.it_rid > *{last it_rid processed}* and it_f.it_rid <= *{max it_f.it_rid}* order by it_f.it_rid	
External Reference	it_f.it_rid + "!" + it_f.cycc_oid	ETL
Time Key	it_f.end_time[YYYY-MM-DD]	ETL
System On Hand Quantity	it_f.qty	ETL/Sum
Counted On Hand Quantity	it_f.act_qty - (it_f.act_qty for it_f.cycc_oid = {current cycc order} and for it_f.transact = 'CEXR')	ETL/Sum
Number of Lines Correct	if (total(System On Hand Quantity for External Reference, Time Key) <> total(Counted On Hand Quantity for External Reference, Time Key)) then (0) else (1)	Calculated
Number of Lines Counted	COUNT (DISTINCT External Reference)	Calculated
Cycle Count Accuracy Percent	Number of Lines Correct / Number of Lines Counted	Calculated

9.6.24 WMS Warehouse Performance - Order Line Measures Definition

9.6.24.1 bi_wp_ordl Fact Table

Name	Value	ETL/Aggregate
	from it_f, om_h, od_h where it_f.ob_oid = om_h.ob_oid and it_f.ob_type = om_h.ob_type and it_f.ob_oid = od_h.ob_oid and it_f.ob_type = od_h.ob_type and it_f.ob_lno = od_h.ob_lno and it_f.transact = 'SHIP' and it_f.whse_code = *{whse}* and it_f.it_rid > *{last it_rid processed}* and it_f.it_rid <= *{max it_f.it_rid}*	
External Reference	it_f.ob_oid + "!" + it_f.ob_type + "!" + if_f.ob_lno	ETL
Order Key	it_f.ob_oid + "!" + it_f.ob_type	ETL
Time Key	it_f.end_time[YYYY-MM-DD]	ETL
Number of Orders	COUNT (DISTINCT Order Key)	Calculated
Order Quantity	od_h.ord_qty	ETL/Sum
Filled Quantity	od_h.cmp_qty	ETL/Sum
Number of Orders Filled	if (total(Order Quantity for Order Key) <> total(Filled Quantity for Order Key)) then (0) else (1)	Calculated
Order Fill Rate Percent	Number of Orders Filled / Number of Orders	Calculated
Number of Order Lines	COUNT (DISTINCT External Reference)	Calculated
Number of Order Lines Filled	if (total(Order Quantity for External Reference, Time Key) <> total(Filled Quantity for External Reference, Time Key)) then (0) else (1)	Calculated
Order Line Fill Rate Percent	Number of Order Lines Filled / Number of Order Lines	Calculated
On Time Ready to Ship	it_f.end_time <= om_h.request_date then 1 else 0	ETL
On Time Ready to Ship Order	if (total(On Time Ready to Ship for Order Key) = 0) then (0) else (1)	Calculated
On Time Ready to Ship Rate Percent	On Time Ready to Ship Order / Number of Orders	Calculated

Name	Value	ETL/Aggregate
Total Order Cycle Time in Minutes	it_f.end_time - om_h.ord_date - number of Non working days(defined in md_calendar using so_f.operations_calendar_code)	ETL/Average
Total Order Cycle Time in Hours	Total Order Cycle Time in Minutes / 60	Average
Internal Order Cycle Time in Minutes	it_f.end_time - om_h.wms_ord_create_stamp - number of Non working days(defined in md_calendar using so_f.operations_calendar_code)	ETL/Average
Internal Order Cycle Time in Hours	Internal Order Cycle Time in Minutes / 60	Average
Order Lines Picked and Shipped per Minute	min(it_f.start_time) where transact = 'PICK' for it_f.ob_oid and for it_f.ob_type and for it_f.ob_lno and whse_code = {whse} - max(it_f.end_time) where transact = 'PICK' for it_f.ob_oid and for it_f.ob_type and for it_f.ob_lno and whse_code = {whse} - number of Non working days(defined in md_calendar using so_f.operations_calendar_code)	ETL/Average
Orders Picked and Shipped per Minute	Order Lines Picked and Shipped per Minute / Number of Orders	Average
Order Lines Picked and Shipped per Hour	Order Lines Picked and Shipped per Minute /60	Average
Orders Picked and Shipped per Hour	Orders Picked and Shipped per Minute /60	
On Time Ready to Ship Line	On Time Ready to Ship	Sum

9.6.25 WMS Warehouse Performance - Receipt Line Measures Definition

9.6.25.1 bi_wp_rcpl Fact Table

Name	Value	ETL/Aggregate
	from it_f where it_f.transact = 'RCPT' and it_f.receipt is not null and it_f.receipt != ' ' and it_f.whse_code = {whse} and it_f.it_rid > {last it_rid processed} and it_f.it_rid <= {max it_f.it_rid} group by it_f.tag, it_f.receipt, it_f.sku, it_f.pkg, it_f.supp_num order by max_it_rid	
External Reference	it_f.tag + "!" + it_f.supp_num	ETL
Receipt Key	it_f.receipt	ETL
Dock to Stock in Minutes	max(it_f.end_time) where it_f.transact = 'STOR' and for it_f.tag and whse_code = {whse} - min(it_f.start_time) where it_f.transact = 'RCPT' and for it_f.receipt = and whse_code = {whse} - number of days off (none working days defined in md_calendar using so_f.operations_calendar_code)	ETL/Average
Dock to Stock in Hours	Dock to Stock in Minutes /60	Average
On Time Receipts	min(it_f.dt_recv) where it_f.transact = 'RCPT' and for it_f.receipt and whse_code = {whse} > ibod_h.expect_date where for ibod_h.ib_oid and for ibod_h.ib_type and for ibod_h.ib_lno and whse_code = {whse} order by mod_stamp desc then 0 else 1	ETL
Number of On Time Receipts	if (total(On Time Receipts for Receipt Key) = 1) then (1) else (0)	Calculated
Number of Receipts	COUNT (DISTINCT Receipt Key)	Sum
On Time Receipts Percent	(Number of On Time Receipts / Number of Receipts)	Calculated

9.6.26 WMS Warehouse Performance - Warehouse Capacity Measures Definition

9.6.26.1 bi_wp_capa Fact Table

Name	Value	ETL/Aggregate
	from iv_f join lc_f on lc_f.whse_code = iv_f.whse_code and lc_f.loc = (case when iv_f.cont = ' ' then iv_f.loc else (select cn_f.loc from cn_f where cn_f.whse_code = iv_f.whse_code and cn_f.cont = iv_f.cont) end) where iv_f.whse_code = {whse} group by iv_f.whse_code, iv_f.sku, iv_f.pkg, lc_f.loc order by whse_code, sku, pkg, lc_f.loc	
External Reference	iv_f.whse_code + "!" + lc_f.loc + "!" + iv_f.sku + "!" + iv_f.pkg + "!" + Today's Year + "!" + Today's Period	ETL
Location Type Code	lc_f.loc_type	ETL
Average Daily On-Hand Quantity	(sum(iv_f.qty) + (bi_wp_capa.Average Daily On-Hand Quantity * number of days (today's date – last date processed))) / number of days + 1	ETL/Average
Minimum Quantity	iv_f.qty when bi_cp_capa do not exist else bi_cp_capa.min_qty	ETL/Sum
Maximum Quantity	iv_f.qty > bi_cp_capa.max_qty then iv_f.qty	ETL/Sum
Honeycomb Percentage	(((Average Daily On-Hand Quantity + iv_f.qty).multiply(iv_f.dpth1 * iv_f.wid1 * iv_f.hgt1)).divide((lc_f.dpth * lc_f.wid1 * lc_f.hgt), 3)).multiply/100	ETL/Average
Total Volume	lc_f.dpth * lc_f.wid1 * lc_f.hgt	ETL/Sum

Name	Value	ETL/Aggregate

Warehouse Storage Capacity	IF (Location Type Code in ('FP','FIFO','LIFO','BULK')) THEN
	(lc_f.dpth * lc_f.wid1 * lc_f.hgt)
	ELSE
	(null)
		Sum

Average Warehouse Capacity Used Percentage	Warehouse Storage Capacity / Total Volume	Calculated

9.6.27 WMS Warehouse Performance Dimension Names

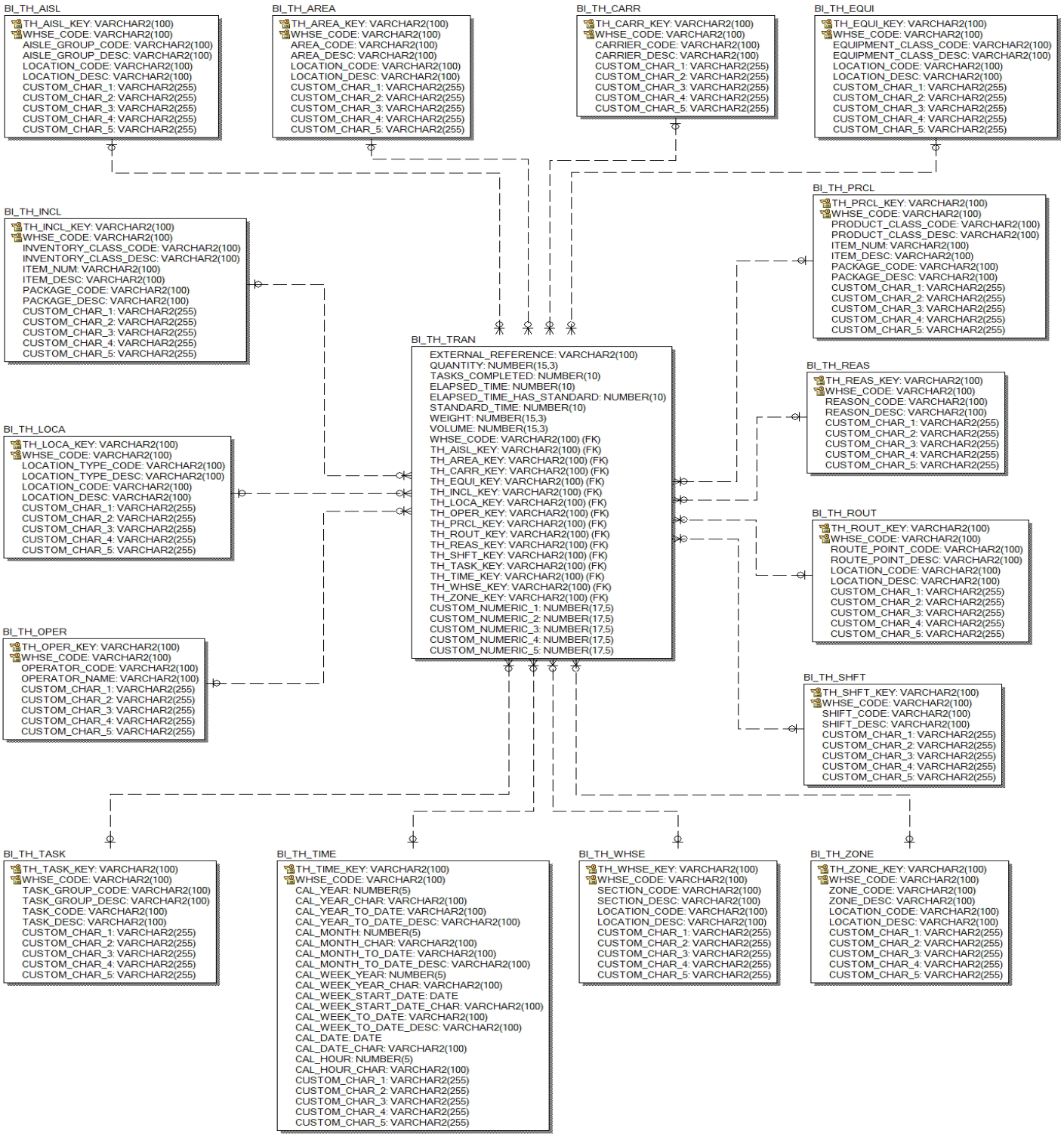
- Time
- Warehouse
- Area
- Zone
- Location Type
- Inventory Class
- Product Class
- Order
- Receipt
- Item
- Customer
- Geography
- Cycle Count
- Supplier

9.6.27.1 Werc Dimensions

- WERC Customer Type
- WERC Industry Type
- WERC Sales
- WERC Strategy
- WERC Year

10 WMS Warehouse Transaction History

10.1 Structure



10.2 Mapping

10.2.1 Columns for the BI_TH_TRAN Table

Name	Value Coming From	Fact
EXTERNAL_REFERENCE	it_f.it_rid	
WHSE_CODE	so_f.whse_code	
TH_AISL_KEY	it_f.from_loc it_f.to_loc	it_f.transact != 'REPL' it_f.transact = 'REPL'
TH_AREA_KEY	same as th_aisl_key	
TH_CARR_KEY	it_f.carrier_service	
TH_EQUI_KEY	same as th_aisl_key	
TH_INCL_KEY	it_f.sku '!' it_f.pkg	
TH_LOCA_KEY	same as th_aisl_key	
TH_OPER_KEY	it_f.opr	
TH_PRCL_KEY	it_f.sku '!' it_f.pkg	
TH_ROUT_KEY	same as th_aisl_key	
TH_REAS_KEY	rs_f.rsn_code	
TH_SHFT_KEY	1 2 3	th_time_key[12,2][HH]<10 th_time_key[12,2][HH]> =10 and < 14 th_time_key[12,2][HH]> = 14
TH_TASK_KEY	it_f.transact	
TH_TIME_KEY	it_f.start_time[1,14] + '00 :00'	[YYYY-MM-DD HH:00:00]
TH_WHSE_KEY	same as th_aisl_key	
TH_ZONE_KEY	same as th_aisl_key	
QUANTITY	it_f.act_qty	
TASK_COMPLETED	1	true
ELAPSED_TIME	it_f.end_time - om_h.ord_date - number of days off	**Note 1**
ELAPSED_TIME_HAS_STANDARD	elapsed_time	**Note 2**
STANDARD_TIME	null	**Note 3**

Name	Value Coming From	Fact
WEIGHT	it_f.act_qty * pm_f.wgt1	
VOLUME	it_f.act_qty * pm_f.wid1 * pm_f.hgt1* pm_f.dpth1	
CUSTOM_NUMERIC_1		
CUSTOM_NUMERIC_2		
CUSTOM_NUMERIC_3		
CUSTOM_NUMERIC_4		
CUSTOM_NUMERIC_5		

Note 1
The number of days off correspond to the number of md_calendar_day.is_work_day is 0 for the so_f.operations_calendar_code where so_f.so_rid = 1

Note 2
- The default is elapsed_time if does not fall to the below condition:
- If the sd_f.std_time for sd_f.full_part 'F','P' is null then
  ELAPSED_TIME_HAS_STANDARD = null
- If STANDARD_TIME is null then ELAPSED_TIME_HAS_STANDARD = 0

Note 3
The default is null if does not fall to the below condition:
If pm_f.load_min_qty is not null and compare to the it_f.act_qty >= 0 then
STANDARD_TIME = sd_f.std_time for sd_f.full_part 'F' if it is not null otherwise
STANDARD_TIME = sd_f.std_time for sd_f.full_part 'P' if it is not null

Read records from it_f where
it_f.it_rid > (bi_dmr_att.dmr_att_value where dmr_name = 'warehouse-transaction-history' and dmr_att_key = 'last_it_rid!' + *warehouse code*) and it_f.it_rid <= max(it_rid)

10.2.2 Columns for the BI_TH_AISL Table

Name	Value Coming From	Dimension
TH_AISL_KEY (PK)	lc_f.loc	
WHSE_CODE (PK)	so_f.whse_code	
AISLE_GROUP_CODE	lc_f.aisle_grp	
AISLE_GROUP_DESC	lc_f.aisle_grp	
LOCATION_CODE	lc_f.loc	
LOCATION_DESC	lc_f.loc	

Name	Value Coming From	Dimension
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

10.2.3 Columns for the BI_TH_AREA Table

Name	Value Coming From	Dimension
TH_AREA_KEY (PK)	lc_f.loc	
WHSE_CODE (PK)	so_f.whse_code	
AREA_CODE	lc_f.area	
AREA_DESC	lc_f.area	
LOCATION_CODE	lc_f.loc	
LOCATION_DESC	lc_f.loc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

10.2.4 Columns for the BI_TH_CARR Table

Name	Value Coming From	Dimension
TH_CARR_KEY (PK)	ca_f.carrier_service	
WHSE_CODE (PK)	so_f.whse_code	
CARRIER_CODE	ca_f.carrier_service	
CARRIER_DESC	ca_f.carrier_service_name	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		

Name	Value Coming From	Dimension
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

10.2.5 Columns for the BI_TH_EQUI Table

Name	Value Coming From	Dimension
TH_EQUI_KEY (PK)	lc_f.loc	
WHSE_CODE (PK)	so_f.whse_code	
EQUIPMENT_CLASS_CODE	lc_f.eq_class	
EQUIPMENT_CLASS_DESC	lc_f.eq_class	
LOCATION_CODE	lc_f.loc	
LOCATION_DESC	lc_f.loc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

10.2.6 Columns for the BI_TH_INCL Table

Name	Value Coming From	Dimension
TH_INCL_KEY (PK)	pm_f.sku '!' pm_f.pkg	
WHSE_CODE (PK)	so_f.whse_code	
INVENTORY_CLASS_CODE	pm_f.inv_class	
INVENTORY_CLASS_DESC	pm_f.inv_class	
ITEM_NUM	pm_f.sku	
ITEM_DESC	pm_f.sku_desc	
PACKAGE_CODE	pm_f.pkg	
PACKAGE_DESC	pm_f.pkg_desc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		

Name	Value Coming From	Dimension
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

10.2.7 Columns for the BI_TH_LOCA Table

Name	Value Coming From	Dimension
TH_LOCA_KEY (PK)	lc_f.loc	
WHSE_CODE (PK)	so_f.whse_code	
LOCATION_TYPE_CODE	lc_f.loc_type	LOCATION_TYPE_CODE
LOCATION_TYPE_DESC	lc_f.loc_type	
LOCATION_CODE	lc_f.loc	
LOCATION_DESC	lc_f.loc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

10.2.8 Columns for the BI_TH_OPER Table

Name	Value Coming From	Dimension
TH_OPER_KEY (PK)	us_f.opr	
WHSE_CODE (PK)	so_f.whse_code	
OPERATOR_CODE	us_f.opr	
OPERATOR_NAME	us_f.opr_name	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

10.2.9 Columns for the BI_TH_PRCL Table

Name	Value Coming From	Dimension
TH_PRCL_KEY (PK)	pm_f.sku '!' pm_f.pkg	
WHSE_CODE (PK)	so_f.whse_code	
PRODUCT_CLASS_CODE	pm_f.prod_class	
PRODUCT_CLASS_DESC	pm_f.prod_class	
ITEM_NUM	pm_f.sku	
ITEM_DESC	pm_f.sku_desc	
PACKAGE_CODE	pm_f.pkg	
PACKAGE_DESC	pm_f.pkg_desc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

10.2.10 Columns for the BI_TH_REAS Table

Name	Value Coming From	Dimension
TH_REAS_KEY (PK)	rs_f.rsn_code	
WHSE_CODE (PK)	so_f.whse_code	
REASON_CODE	rs_f.rsn_code	
REASON_DESC	rs_f.rsn_code_desc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

10.2.11 Columns for the BI_TH_ROUT Table

Name	Value Coming From	Dimension
TH_LOCA_KEY (PK)	lc_f.loc	
WHSE_CODE (PK)	so_f.whse_code	
ROUTE_POINT_CODE	lc_f.route_point	
ROUTE_POINT_DESC	lc_f.route_point	
LOCATION_CODE	lc_f.loc	
LOCATION_DESC	lc_f.loc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

10.2.12 Columns for the BI_TH_SHFT Table

Name	Value Coming From	Dimension
TH_SHFT_KEY (PK)	'1' '2' '3'	
WHSE_CODE (PK)	so_f.whse_code	
SHIFT_CODE	'1' '2' '3'	
SHIFT_DESC	1-'BEFORE 10:00' 2-'BETWEEN 10:00 AND 14:00' 3-'AFTER 14:00'	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		

Name	Value Coming From	Dimension
CUSTOM_CHAR_5		

10.2.13 Columns for the BI_TH_TASK Table

Name	Value Coming From	Dimension
TH_TASK_KEY (PK)	transact.transact	
WHSE_CODE (PK)	so_f.whse_code	
TASK_GROUP_CODE	transact.rpt_group	
TASK_GROUP_DESC	locale_text.lit_text	**Note 1**
TASK_CODE	transact.transact	
TASK_DESC	transact.desc_1	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Note 1

Select md_locale_text.lit_text from md_locale_text, md_domain_value, md_column
where md_column.database_name = 'wms' and md_column.table_name = 'transact'
and md_column.column_name = 'rpt_group'
and md_domain_value.database_name = md_column.database_name
and md_domain_value.domain_name = md_column.domain_name
and md_domain_value.accepted_value = *bi_th_task.task_group_code*
and md_locale_text.lit_key = md_domain_value.desc_key "
and md_locale_text.locale_key = ***{locale1}/{locale2}***

10.2.14 Columns for the BI_TH_TIME Table

Name	Value Coming From	Dimension
TH_TIME_KEY (PK)	it_f.start_time	**Note 1** [YYYY-MM-DD HH:00:00]
WHSE_CODE (PK)	so_f.whse_code	
CAL_YEAR	th_time_key[1,4]	[YYYY]
CAL_YEAR_CHAR	th_time_key[1,4]	[YYYY]

Name	Value Coming From	Dimension
CAL_YEAR_TO_DATE	'1' '2'	where cal_date >= lytd Start Date and >= lytd End Date where cal_date >= ytd Start Date and >= ytd End Date
CAL_YEAR_TO_DATE_DESC	'Last Year-To-Date' 'Year-To-Date'	where cal_date >= lytd Start Date and >= lytd End Date where cal_date >= ytd Start Date and >= ytd End Date
CAL_MONTH	th_time_key[6,2]	[MM]
CAL_MONTH_CHAR	th_time_key[6,2]	[MM]
CAL_MONTH_TO_DATE	'1' '2' '3'	where cal_date >= lymtd Start Date and >= lymtd End Date where cal_date >= lmtd Start Date and >= lmtd End Date where cal_date >= mtd Start Date and >= mtd End Date
CAL_MONTH_TO_DATE_DESC	'Last Year Month-To-Date' 'Last Month-To-Date' 'Month-To-Date'	where cal_date >= lymtd Start Date and >= lymtd End Date where cal_date >= lmtd Start Date and >= lmtd End Date where cal_date >= mtd Start Date and >= mtd End Date
CAL_WEEK_YEAR	Week Start Date[1,4]	**Note 2** [YYYY]
CAL_WEEK_YEAR_CHAR	Week Start Date[1,4]	**Note 2** [YYYY]
CAL_WEEK_START_DATE	Week Start Date	**Note 2**
CAL_WEEK_START_DATE_CHAR	Week Start Date	**Note 2**
CAL_WEEK_TO_DATE	'1' '2'	where cal_date >= lwtd Start Date and >= lwtd End Date where cal_date >= wtd Start Date and >= wtd End Date
CAL_WEEK_TO_DATE_DESC	'Last-Week-To-Date' 'Week-To-Date'	where cal_date >= lwtd Start Date and >= lwtd End Date where cal_date >= wtd Start Date and >= wtd End Date
CAL_DATE	th_time_key[1,10]	[YYYY-MM-DD]
CAL_DATE_CHAR	th_time_key[1,10]	[YYYY-MM-DD]
CAL_HOUR	th_time_key[12,2]	[HH]
CAL_HOUR_CHAR	th_time_key[12,2]	[HH]
CUSTOM_CHAR_1		

Name	Value Coming From	Dimension
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

Note 1

- Select min(it_f.start_time) from it_f
- Select max(it_f.start_time) from it_f
- For loop on the earliest start_time(min) year starting on the 1st of January to the latest start_time(max) year ending on the 31st of December
- Each date is defined per 24 hours.
- If the min year = max year will start 1 year earlier.

Note 2

Determine the start date of the week of the current th_time_key using the Start Day in Week defined in the md_calendar for the so_f.operations_calendar_code.

10.2.15 Columns for the BI_TH_WHSE Table

Name	Value Coming From	Dimension
TH_WHSE_KEY (PK)	lc_f.loc	
WHSE_CODE (PK)	so_f.whse_code	
SECTION_CODE	lc_f.whse_section	
SECTION_DESC	lc_f.whse_section	
LOCATION_CODE	lc_f.loc	
LOCATION_DESC	lc_f.loc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

10.2.16 Columns for the BI_TH_ZONE Table

Name	Value Coming From	Dimension
TH_ZONE_KEY (PK)	lc_f.loc	
WHSE_CODE (PK)	so_f.whse_code	

Name	Value Coming From	Dimension
ZONE_CODE	lc_f.zone	
ZONE_DESC	lc_f.zone	
LOCATION_CODE	lc_f.loc	
LOCATION_DESC	lc_f.loc	
CUSTOM_CHAR_1		
CUSTOM_CHAR_2		
CUSTOM_CHAR_3		
CUSTOM_CHAR_4		
CUSTOM_CHAR_5		

10.2.17 WMS Warehouse Transaction History Measures Definition

10.2.17.1 bi_th_tran Fact Table

Name	Value	Aggregate/ETL
	from it_f join transact on it_f.transact = transact.transact and transact.rpt_group != 0 left outer join pm_f on it_f.sku = pm_f.sku and it_f.pkg = pm_f.pkg left outer join lc_f on it_f.from_loc = lc_f.loc left outer join sd_f f_sd_f on lc_f.area = f_sd_f.area and lc_f.eq_class = f_sd_f.eq_class and it_f.transact = f_sd_f.task_code and f_sd_f.full_part = 'F' left outer join sd_f p_sd_f on lc_f.area = p_sd_f.area and lc_f.eq_class = p_sd_f.eq_class and it_f.transact = p_sd_f.task_code and p_sd_f.full_part = 'P' left outer join rs_f on it_f.rsn_code = rs_f.rsn_code and it_f.pkg = pm_f.pkg where it_f.transact != 'REPL' and transact.rpt_group !=0 and it_f.whse_code = {whse} and it_f.it_rid > {last it_f.it_rid processed} and it_f.it_rid <= {max it_f.it_rid} union from it_f join transact on it_f.transact = transact.transact and transact.rpt_group != 0 left outer join pm_f on it_f.sku = pm_f.sku and it_f.pkg = pm_f.pkg left outer join lc_f on it_f.to_loc = lc_f.loc left outer join sd_f f_sd_f on lc_f.area = f_sd_f.area and lc_f.eq_class = f_sd_f.eq_class and it_f.transact = f_sd_f.task_code and f_sd_f.full_part = 'F' left outer join sd_f p_sd_f on lc_f.area = p_sd_f.area and lc_f.eq_class = p_sd_f.eq_class and it_f.transact = p_sd_f.task_code and p_sd_f.full_part = 'P' left outer join rs_f on it_f.rsn_code = rs_f.rsn_code and it_f.pkg = pm_f.pkg where it_f.transact = 'REPL' and transact.rpt_group !=0 and it_f.whse_code = {whse} and it_f.it_rid > {last it_f.it_rid processed} and it_f.it_rid <= {max it_f.it_rid} order by f_it_rid	
External Reference	it_f.it_rid	ETL
Quantity	it_f.act_qty	ETL/Sum
Quantity per Hour	Quantity / Elapsed Time in Seconds * 3600	Calculated
Tasks Completed	1 (true)	ETL/Sum
Tasks Completed per Hour	Tasks Completed / Elapsed Time in Seconds * 3600	Calculated

Name	Value	Aggregate/ETL
Elapsed Time in Seconds	(it_f.end_time - it_f.start_time) - number of days off(none working days defined in md_calendar using so_f.operations_calendar_code) between it_f.start_time and it_f.end_time inclusively	ETL/Sum
Elapsed Time in Minutes	Elapsed Time in Seconds / 60	Sum
Elapsed Time in Hours	Elapsed Time in Seconds / 3600	Sum
Elapsed Time in Seconds (Has Standard)	> default to Elapsed Time > case if f_sd_f.std_time is null and if p_sd_f.std_time is null then set to null > case if Standard Time == null then set to 0.	ETL/Sum

Standard Time in Seconds	if (pm_f.load_min_qty != null) {
	if ((f_sd_f.std_time != null) && (it_f.act_qty.compareTo(pm_f.load_min_qty) >= 0))
	{
	standardTime = new Integer(f_sd_f.std_time);
	}
	else if ((pStandardTime != null) && (quantity.compareTo(loadMinQty) < 0)) {
	standardTime = new Integer(pStandardTime);
	}
	}
		ETL/Sum

Standard Time in Minutes	Standard Time in Seconds / 60	Sum
Standard Time in Hours	Standard Time in Seconds / 3600	Sum
Time Over the Standard in Seconds	(IF Standard Time in Seconds is not null THEN Elapsed Time in Seconds - Standard Time in Seconds ELSE null)	Sum
Time Over the Standard in Minutes	(IF Standard Time in Seconds is not null THEN (Elapsed Time in Seconds - Standard Time in Seconds) / 60 ELSE null)	Sum
Time Over the Standard in Hours	(IF Standard Time in Seconds is not null THEN (Elapsed Time in Seconds - Standard Time in Seconds) / 3600 ELSE null)	Sum
Weight	pm_f.wgt1 * it_f.act_qty	ETL/Sum
Weight per Hour	Weight / Elapsed Time in Seconds * 3600	Calculated
Volume	pm_f.wid1.multiply(pm_f.wgt1).multiply(pm_f.dpth1).multiply(it_f.act_qty)	ETL/Sum
Volume per Hour	Volume / Elapsed Time in Seconds * 3600	Calculated
Productivity Ratio	Standard Time in Seconds / Elapsed Time in Seconds (Has Standard)	Calculated

10.2.18 WMS Warehouse Performance Dimensions Name

- Time
- Shift
- Task
- Operator
- Warehouse
- Area
- Zone
- Aisle Group
- Route Point
- Equipment Class
- Location Type
- Carrier
- Product Class
- Inventory Class
- Reason

11 Appendix

The purpose of this appendix is to explain how to set up a rule in order to send an email when a DMR refresh contains an error.

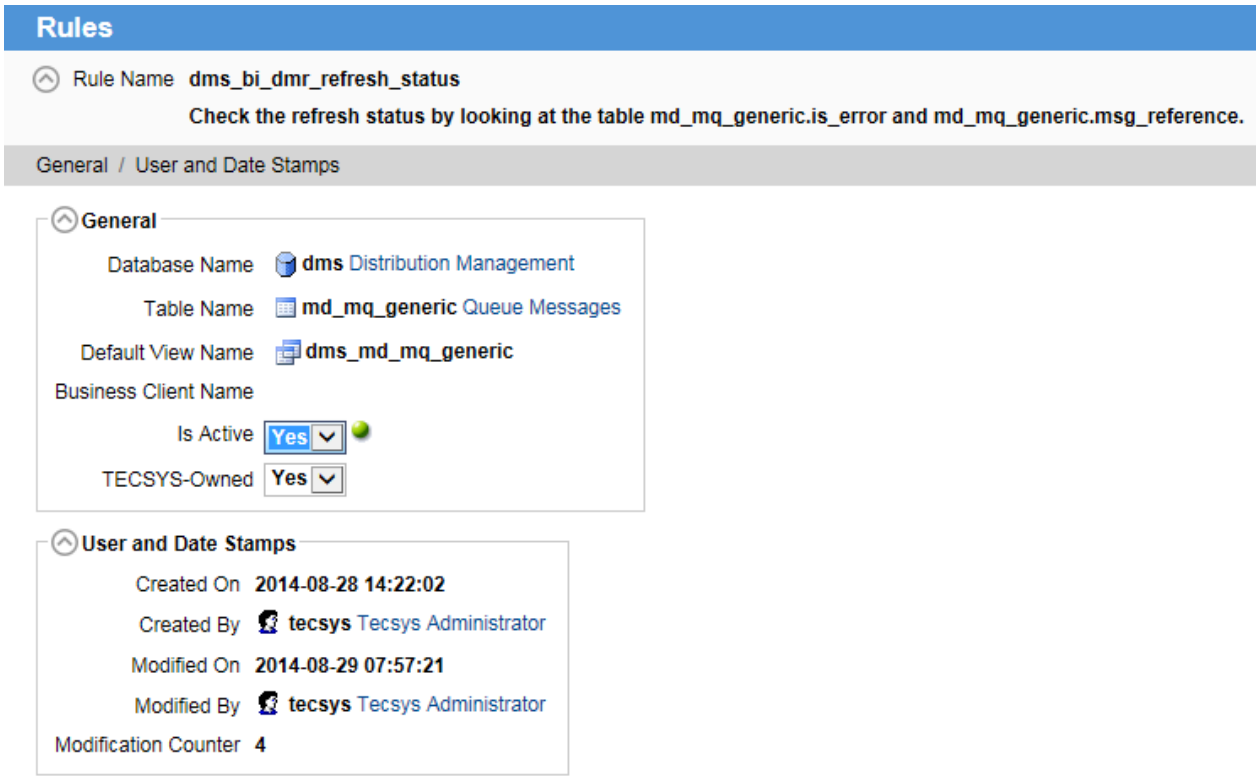
The following conditions need to be fulfilled in order it to work:

- A rule must be created.
- A table action must be created.
- A task must be created.
- A timer that will execute the task must be created.
- The Message Reference of the Bi Dmr Refresh timer needs to be set to be the same as the WHEN condition on the Rule of Message Reference.

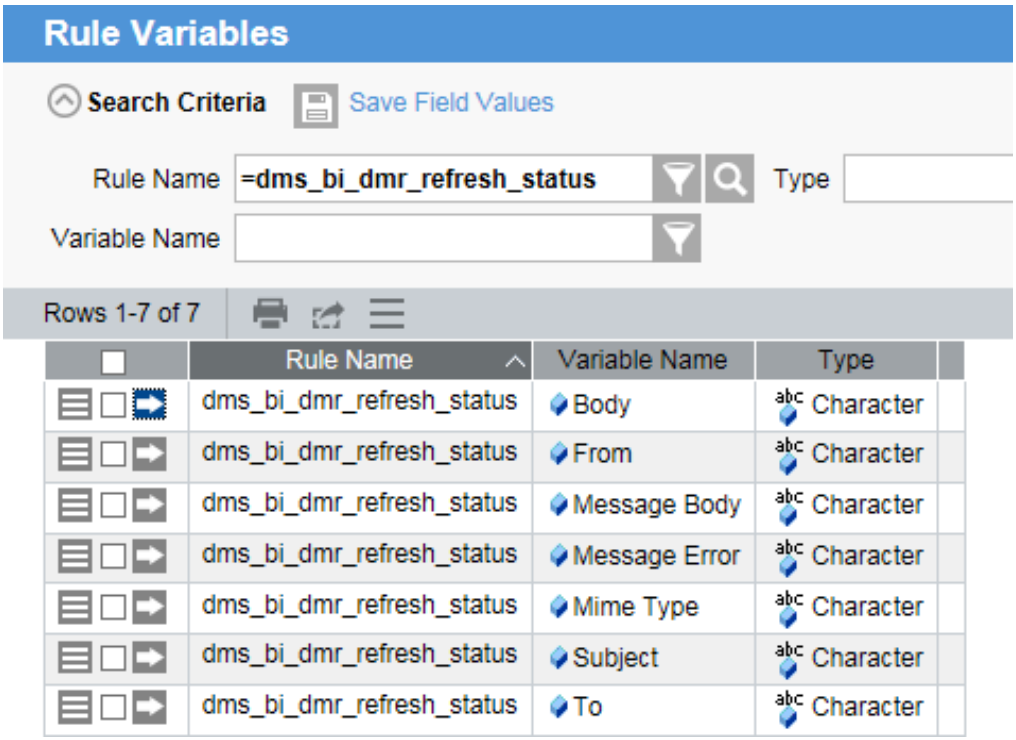
11.1 Creating a Rule

To create a rule, do this:

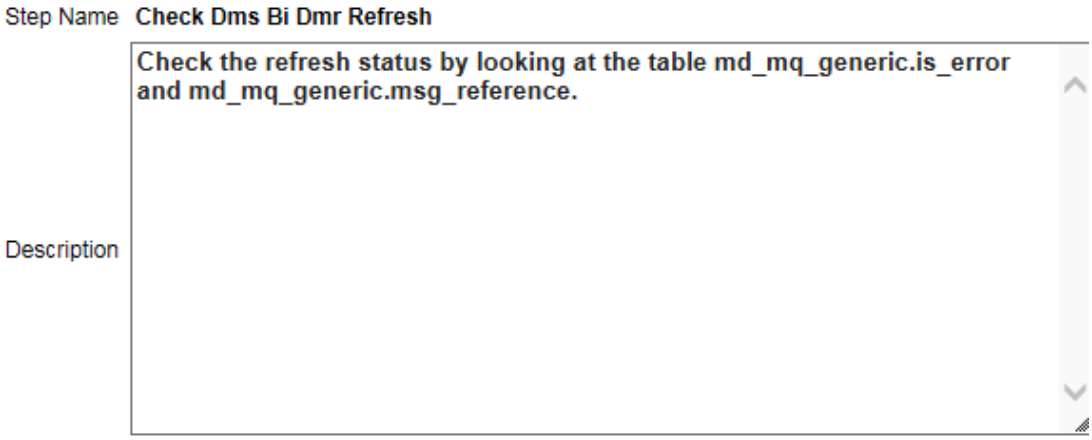
1. Launch the Rules resource and click **Create**.
2. Define the rule, as displayed below:



3. Define the variables:
   - Maintain rule; and,
   - Create a variable (create as many as needed).

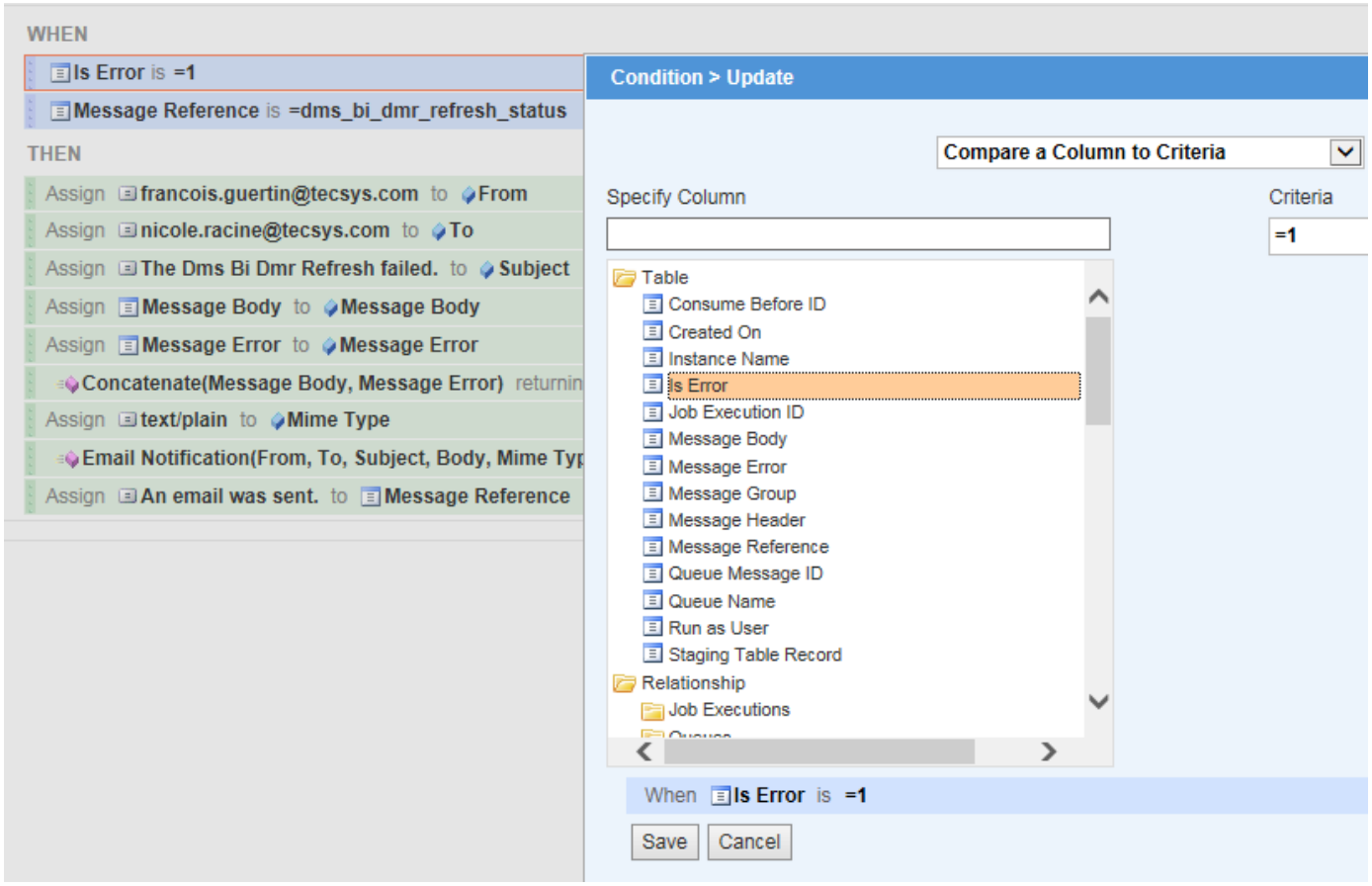


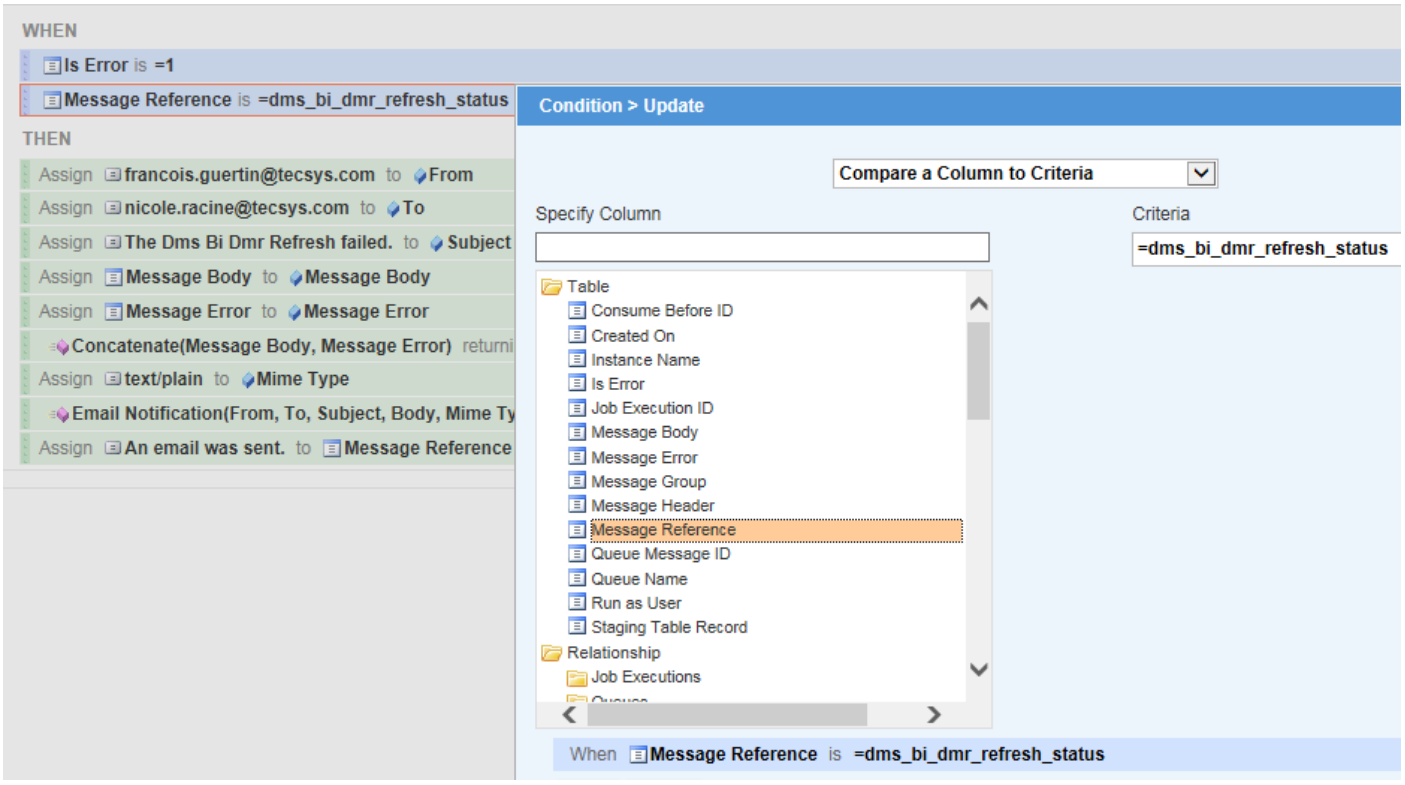
4. Define the step as displayed in the screenshot below:
   - Maintain rule; and,
   - Create a step.



5. Create the When condition as displayed below:

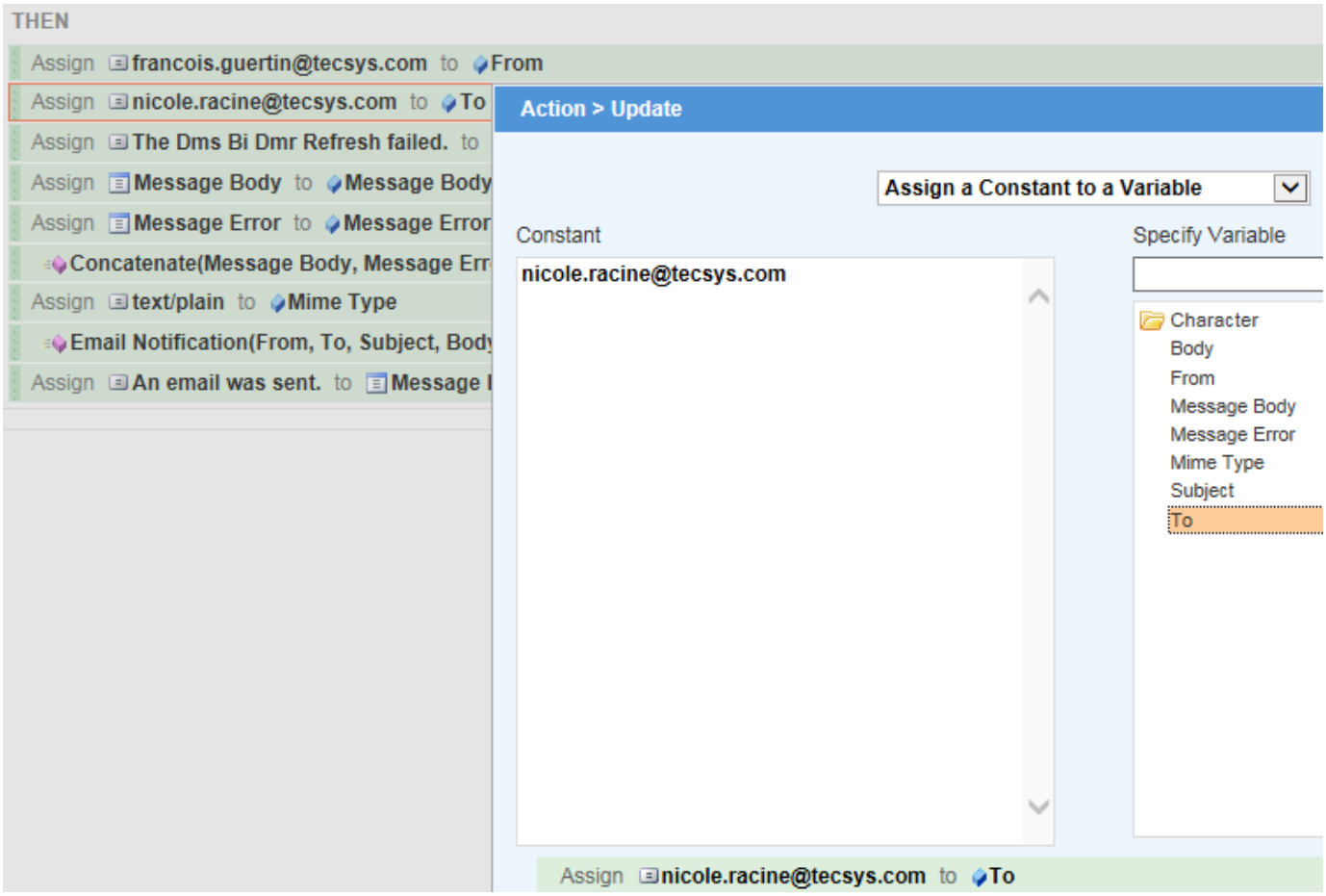
   - Maintain rule; and,
   - Create a WHEN condition (create as many as needed).

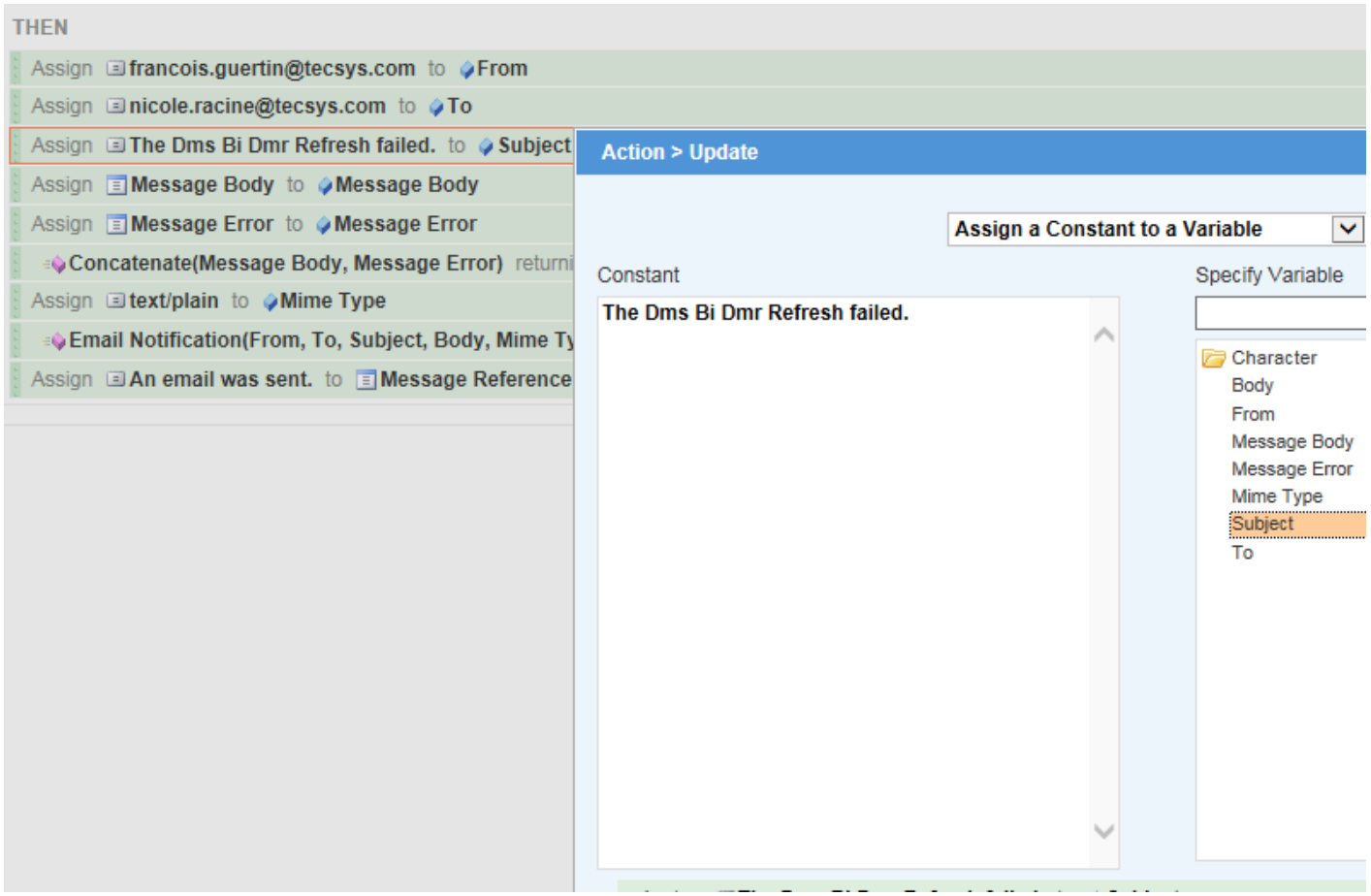


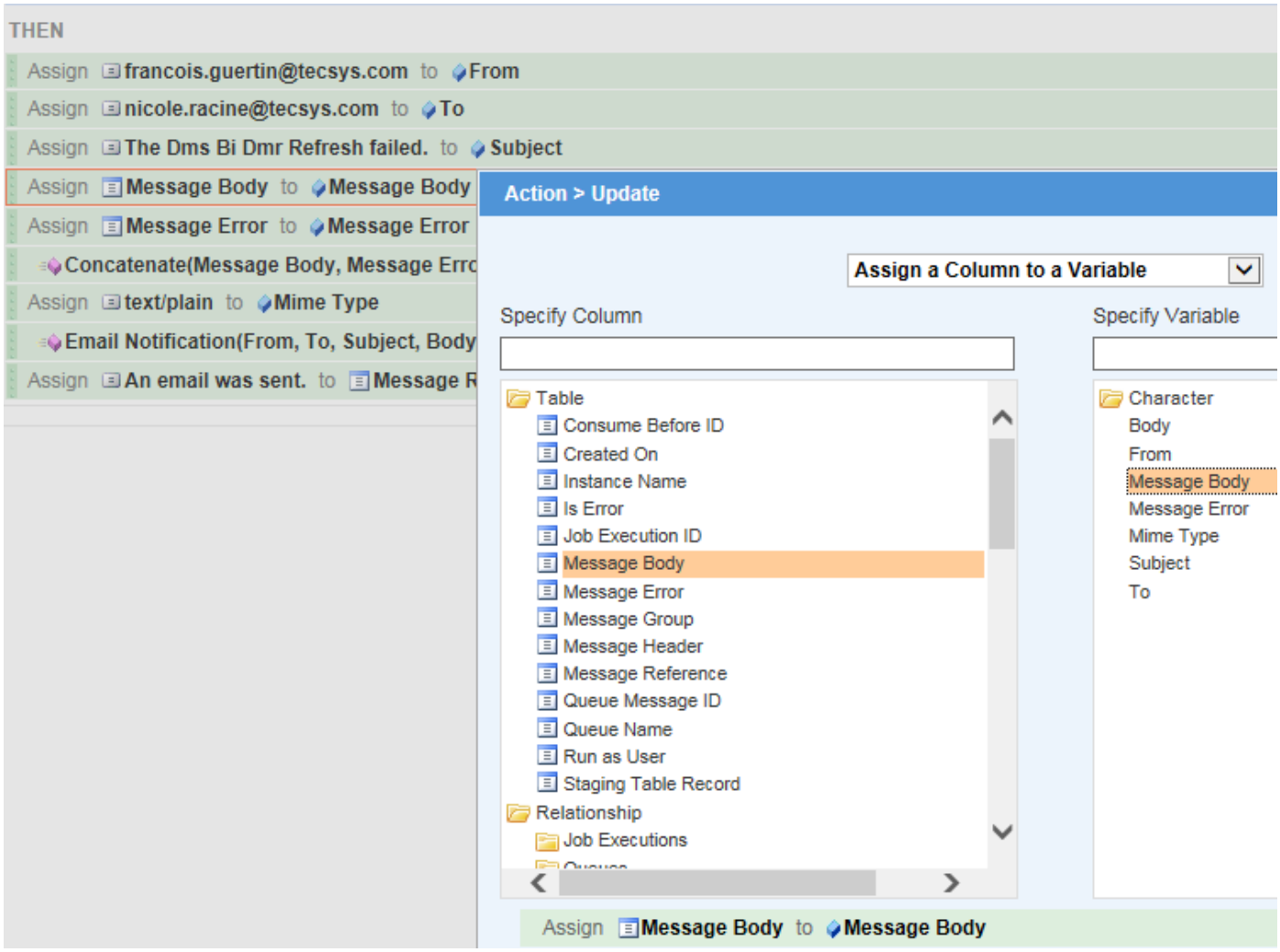


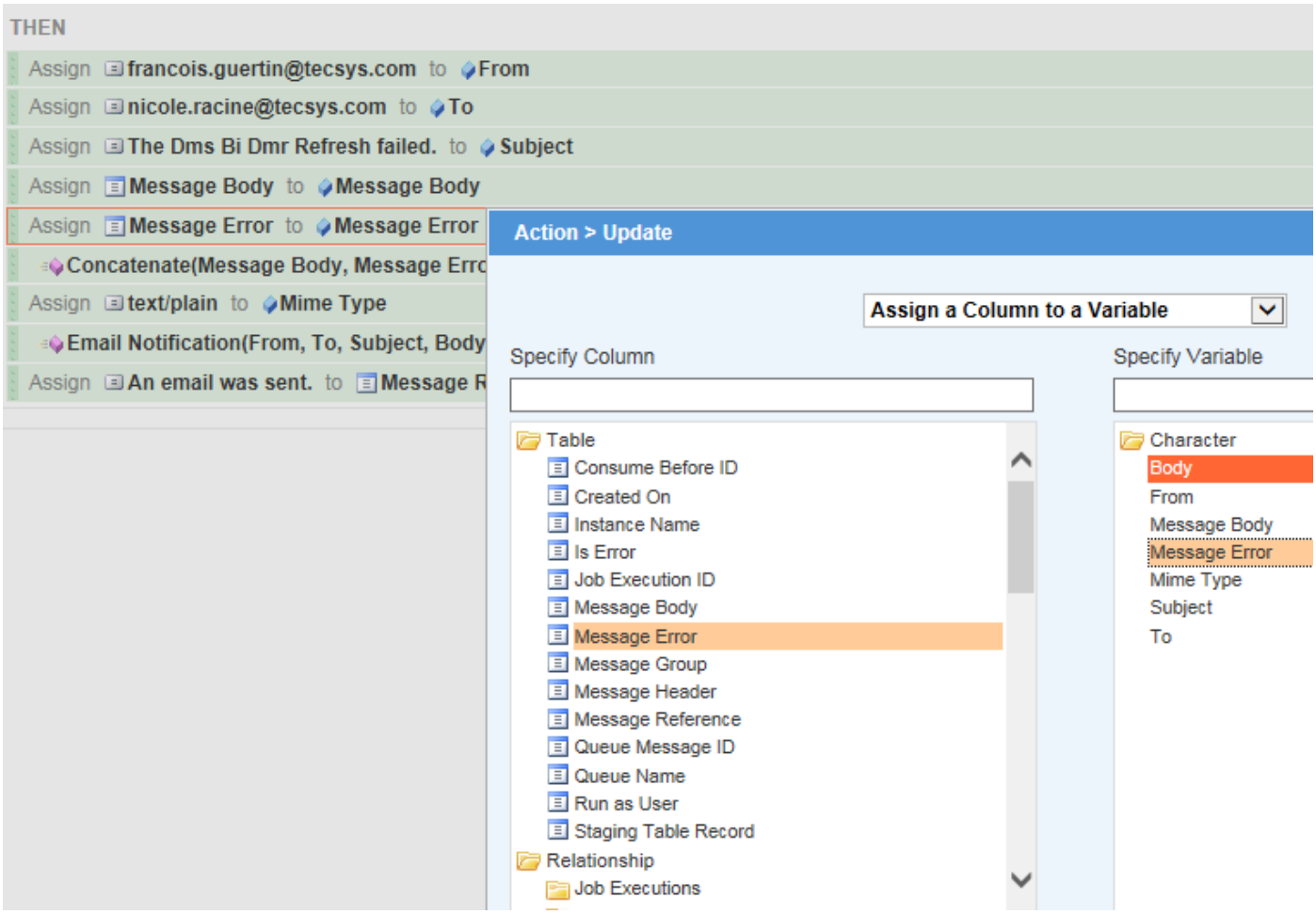
The md_timer.msg_reference need to match the above criteria. Refer to the snapshot at the end of the document.

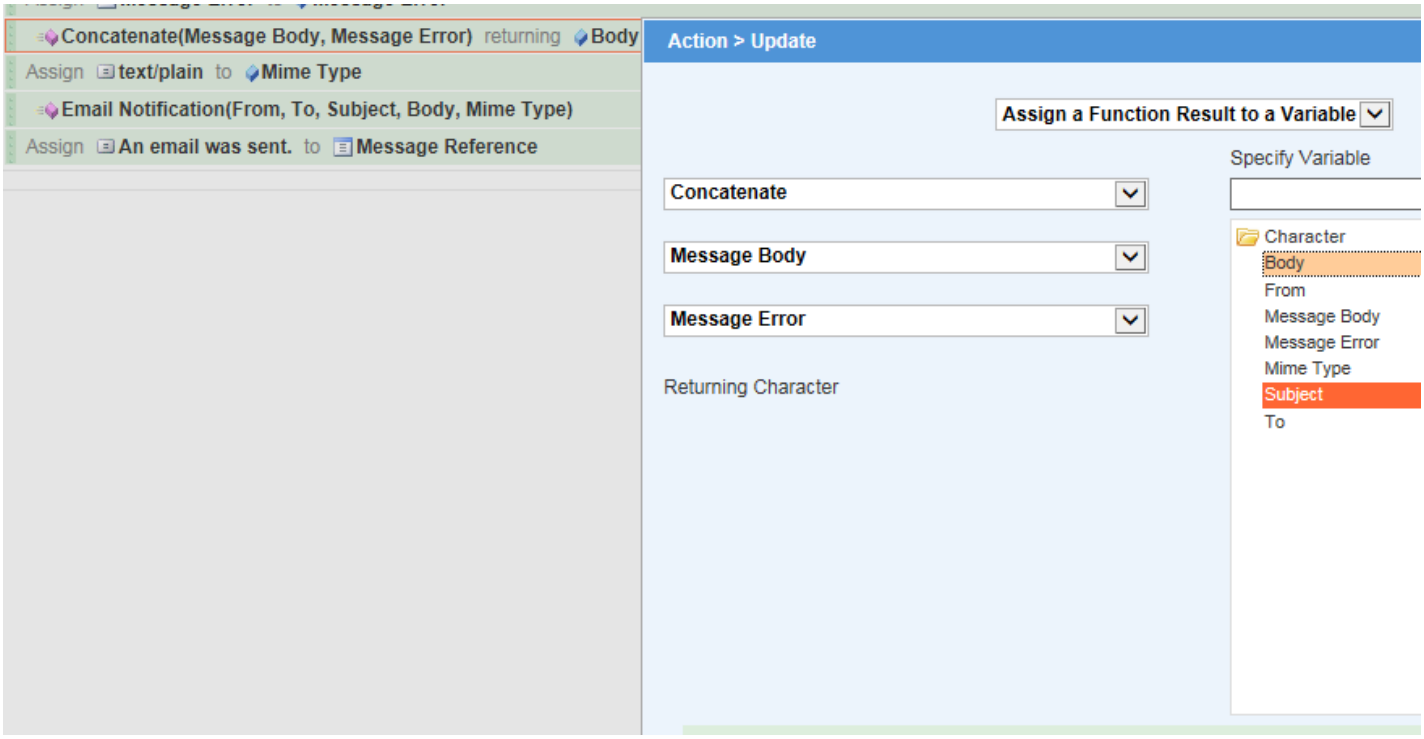
6. Create the Then action as displayed below:
   - Maintain rule; and,
   - Create a THEN action (create as many as needed).

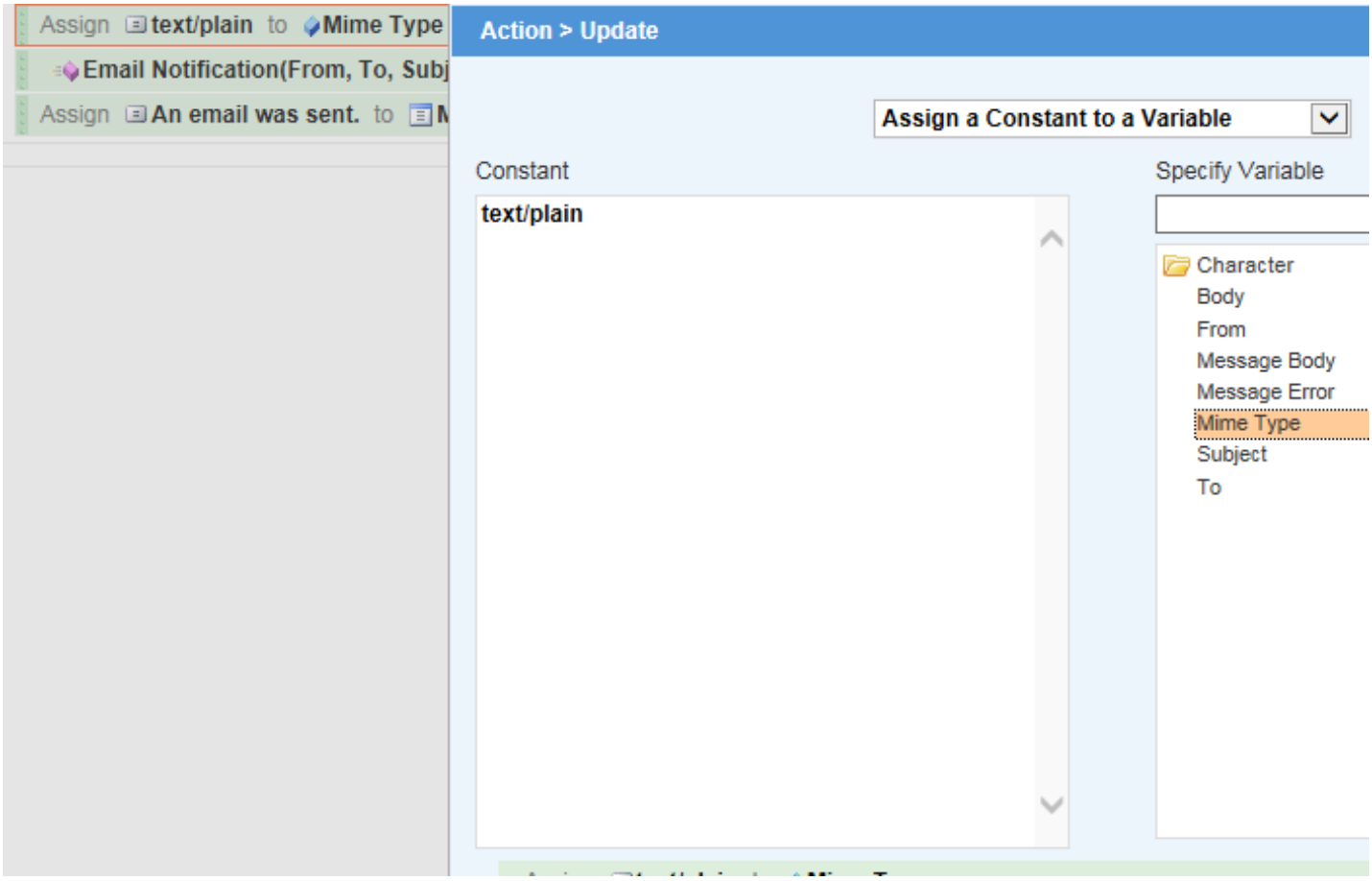

THEN
Assign francois.guertin@tecsys.com to From
Assign nicole.racine@tecsys.com to To
Assign The Dms Bi Dmr Refresh failed. to
Assign Message Body to Message Body
Assign Message Error to Message Error
Concatenate(Message Body, Message Err
Assign text/plain to Mime Type
Email Notification(From, To, Subject, Body
Assign An email was sent. to Message
Action > Update
Assign a Constant to a Variable
Constant
nicole.racine@tecsys.com
Specify Variable
Character
Body
From
Message Body
Message Error
Mime Type
Subject
To
Assign nicole.racine@tecsys.com to To

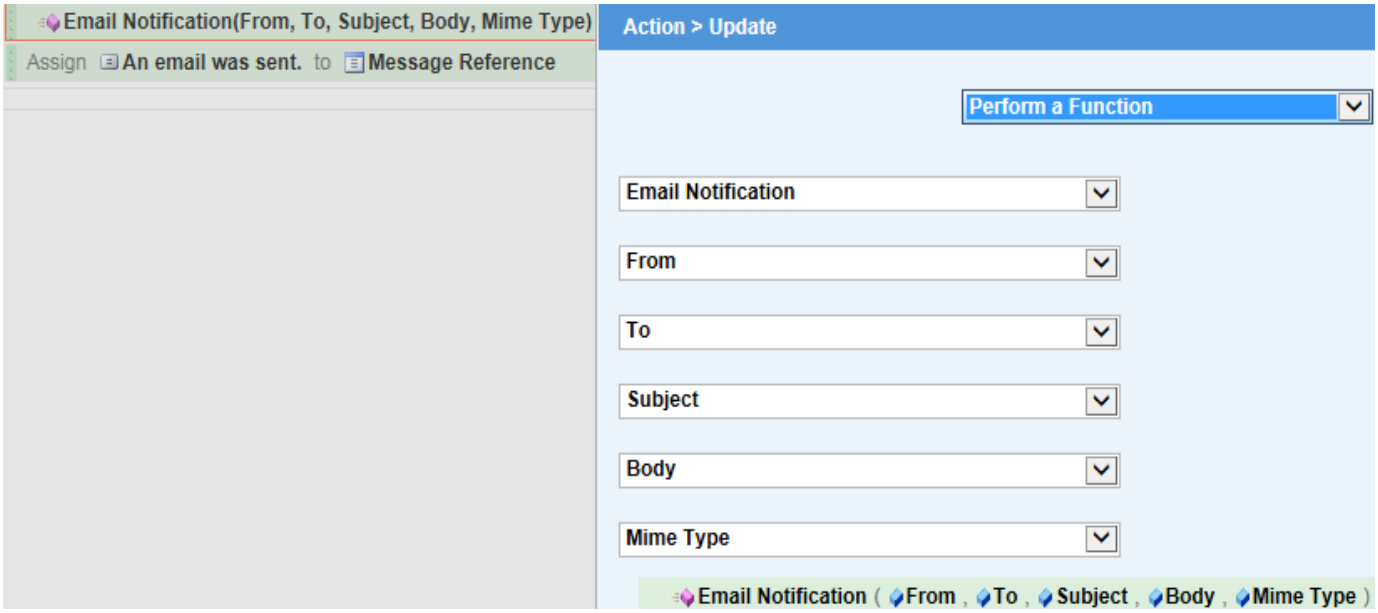

THEN
Assign francois.guertin@tecsys.com to From
Assign nicole.racine@tecsys.com to To
Assign The Dms Bi Dmr Refresh failed. to Subject
Assign Message Body to Message Body
Assign Message Error to Message Error
Concatenate(Message Body, Message Error) returni
Assign text/plain to Mime Type
Email Notification(From, To, Subject, Body, Mime Ty
Assign An email was sent. to Message Reference
Action > Update
Assign a Constant to a Variable
Constant
The Dms Bi Dmr Refresh failed.
Specify Variable
Character
Body
From
Message Body
Message Error
Mime Type
Subject
To

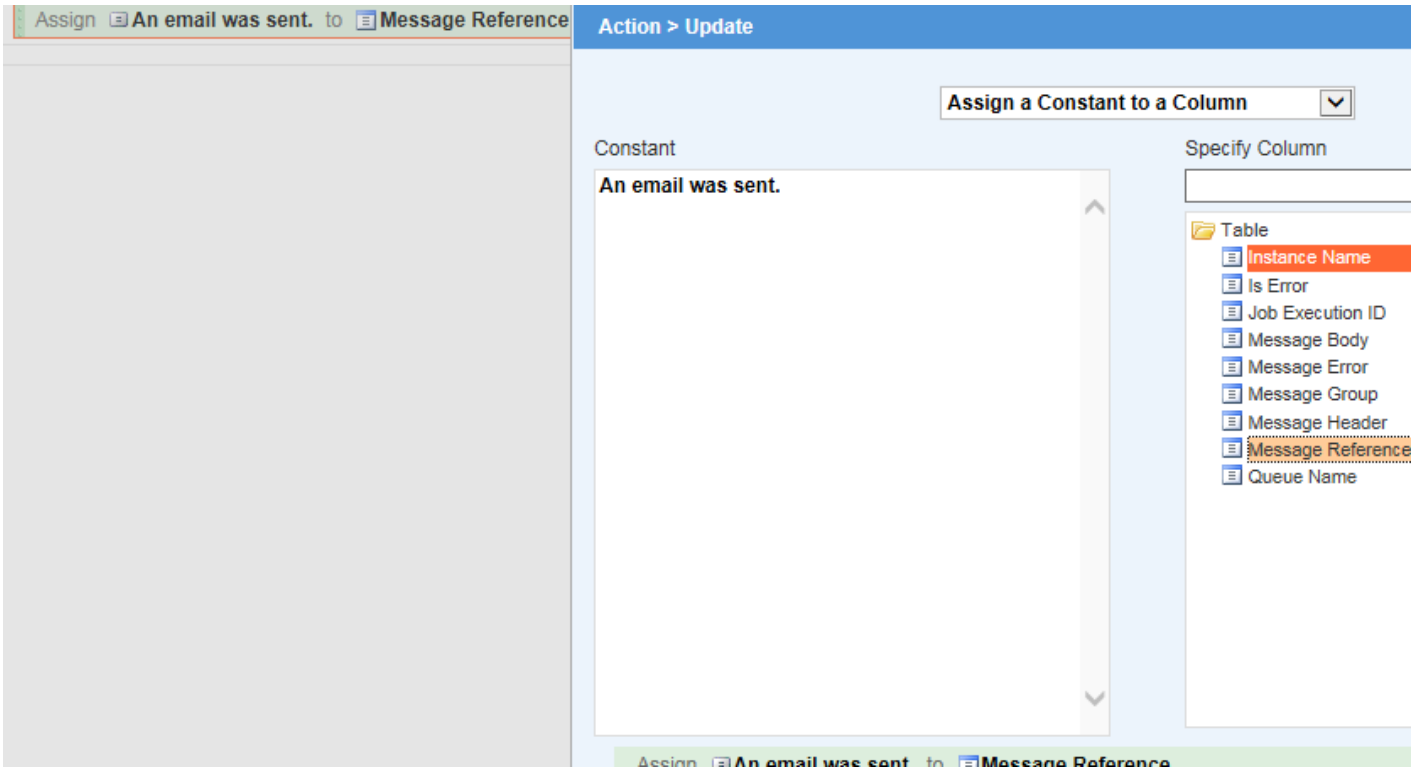

THEN
Assign francois.guertin@tecsys.com to From
Assign nicole.racine@tecsys.com to To
Assign The Dms Bi Dmr Refresh failed. to Subject
Assign Message Body to Message Body
Assign Message Error to Message Error
Concatenate(Message Body, Message Erro
Assign text/plain to Mime Type
Email Notification(From, To, Subject, Body
Assign An email was sent. to Message R
Action > Update
Assign a Column to a Variable
Specify Column
Specify Variable
Table
Consume Before ID
Created On
Instance Name
Is Error
Job Execution ID
Message Body
Message Error
Message Group
Message Header
Message Reference
Queue Message ID
Queue Name
Run as User
Staging Table Record
Relationship
Job Executions
Character
Body
From
Message Body
Message Error
Mime Type
Subject
To
Assign Message Body to Message Body


THEN
Assign francois.guertin@tecsys.com to From
Assign nicole.racine@tecsys.com to To
Assign The Dms Bi Dmr Refresh failed. to Subject
Assign Message Body to Message Body
Assign Message Error to Message Error
Concatenate(Message Body, Message Erro
Assign text/plain to Mime Type
Email Notification(From, To, Subject, Body
Assign An email was sent. to Message R
Action > Update
Assign a Column to a Variable
Specify Column
Specify Variable
Table
Consume Before ID
Created On
Instance Name
Is Error
Job Execution ID
Message Body
Message Error
Message Group
Message Header
Message Reference
Queue Message ID
Queue Name
Run as User
Staging Table Record
Relationship
Job Executions
Character
Body
From
Message Body
Message Error
Mime Type
Subject
To


Concatenate(Message Body, Message Error) returning Body
Assign text/plain to Mime Type
Email Notification(From, To, Subject, Body, Mime Type)
Assign An email was sent. to Message Reference
Action > Update
Assign a Function Result to a Variable
Specify Variable
Concatenate
Message Body
Message Error
Returning Character
Character
Body
From
Message Body
Message Error
Mime Type
Subject
To


Assign text/plain to Mime Type
Email Notification(From, To, Subj
Assign An email was sent. to M
Action > Update
Assign a Constant to a Variable
Constant
text/plain
Specify Variable
Character
Body
From
Message Body
Message Error
Mime Type
Subject
To





7. Click **Submit** to apply your changes.

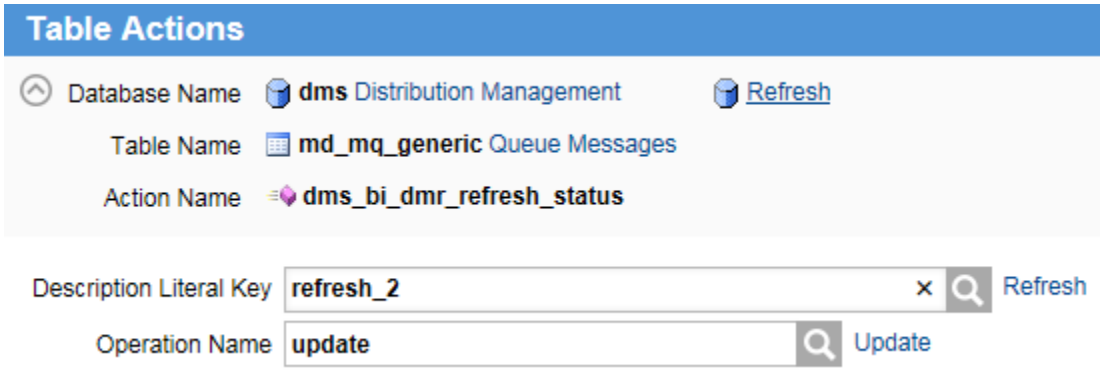
The above examples demonstrate a setup that prevents an email from being sent more than once.

11.2 Creating a Table Action

The table action is required for the task purpose.

To create a table action, do this:

1. Launch the Table Actions resource and click **Create**.
2. Enter an action name that is the same as the rule name, as displayed below.

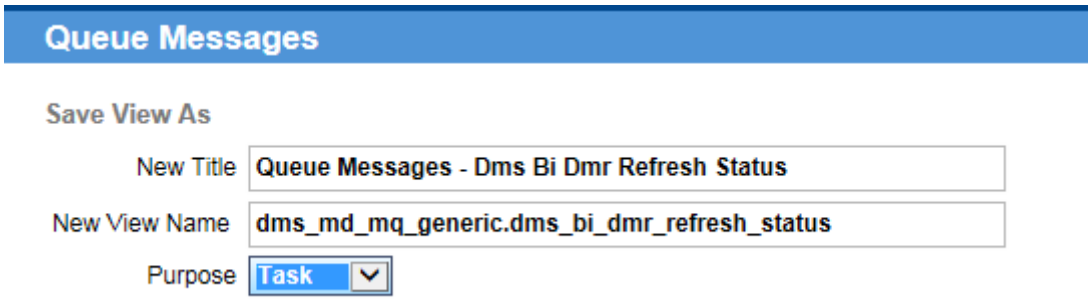


3. Enter the operation name and click **Submit**.
4. Launch the Refresh Metadata and Literals resource to refresh the metadata in order to be able to create a task.

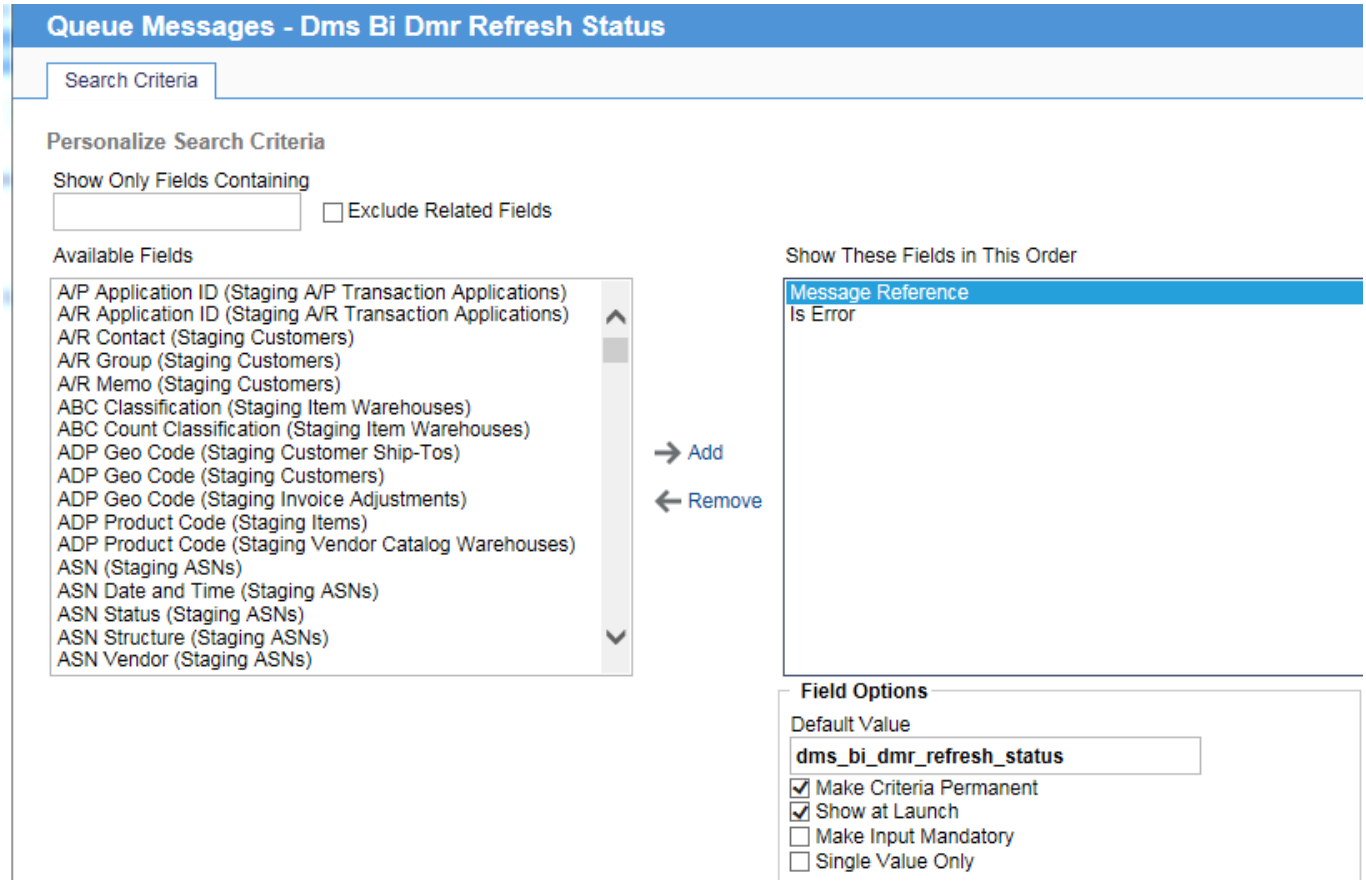
11.3 Creating a Task

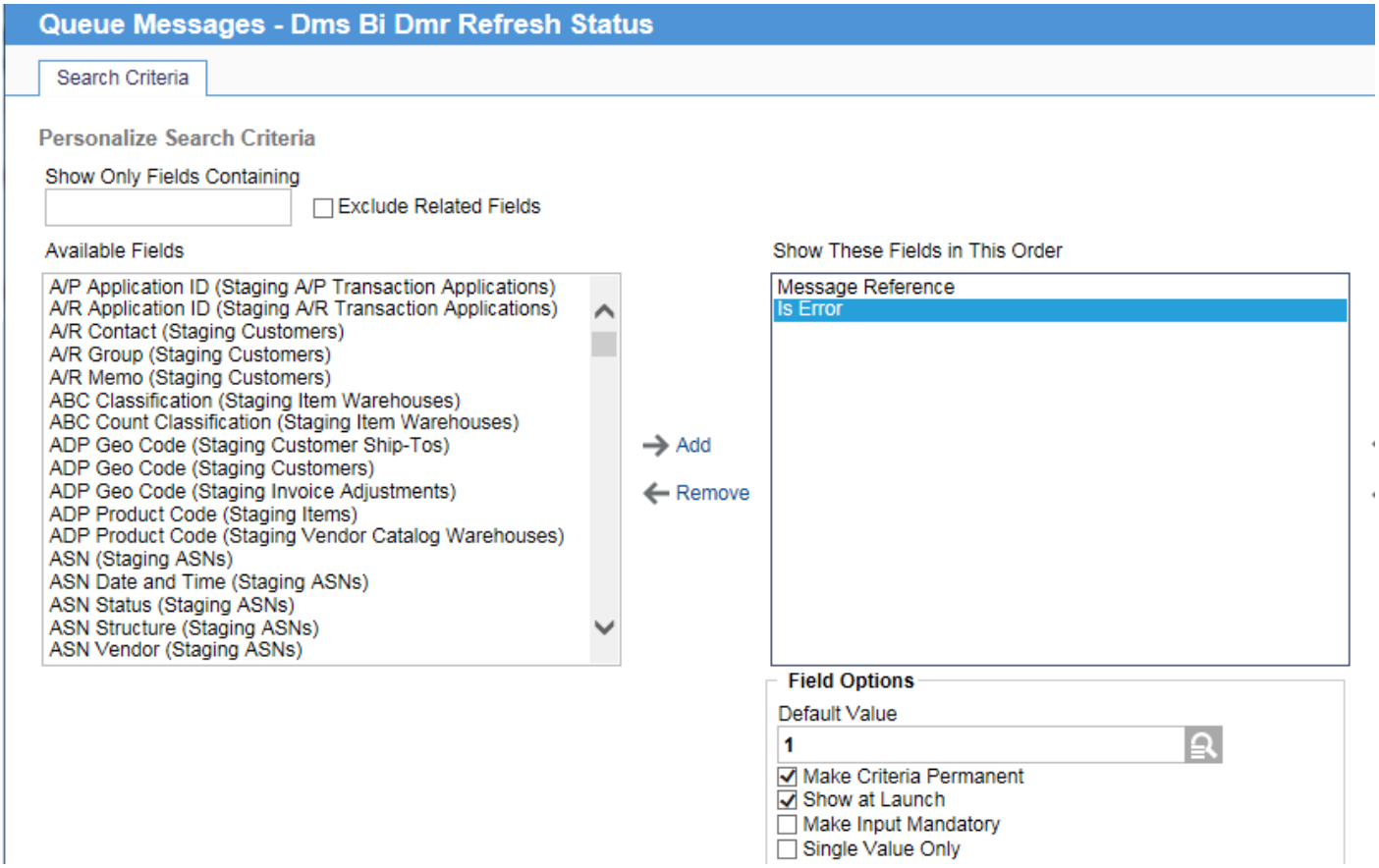
To create a task, do this:

1. Launch the view of the table defined in the Table Actions resource (e.g. dms_md_mq_generic).
2. Select the **Personalize** action and click **Save As** and enter the new view name, as displayed below.



3. Set the Purpose to Task and click **Submit**.
4. In the Search Criteria tab, only include the column defined as the When condition in the rule (e.g. is_error and msg_reference) and check the Make Criteria Permanent option, as displayed below.


Queue Messages - Dms Bi Dmr Refresh Status
Search Criteria
Personalize Search Criteria
Show Only Fields Containing
Exclude Related Fields
Available Fields
A/P Application ID (Staging A/P Transaction Applications)
A/R Application ID (Staging A/R Transaction Applications)
A/R Contact (Staging Customers)
A/R Group (Staging Customers)
A/R Memo (Staging Customers)
ABC Classification (Staging Item Warehouses)
ABC Count Classification (Staging Item Warehouses)
ADP Geo Code (Staging Customer Ship-Tos)
ADP Geo Code (Staging Customers)
ADP Geo Code (Staging Invoice Adjustments)
ADP Product Code (Staging Items)
ADP Product Code (Staging Vendor Catalog Warehouses)
ASN (Staging ASNs)
ASN Date and Time (Staging ASNs)
ASN Status (Staging ASNs)
ASN Structure (Staging ASNs)
ASN Vendor (Staging ASNs)
Add
Remove
Show These Fields in This Order
Message Reference
Is Error
Field Options
Default Value
dms_bi_dmr_refresh_status
Make Criteria Permanent
Show at Launch
Make Input Mandatory
Single Value Only

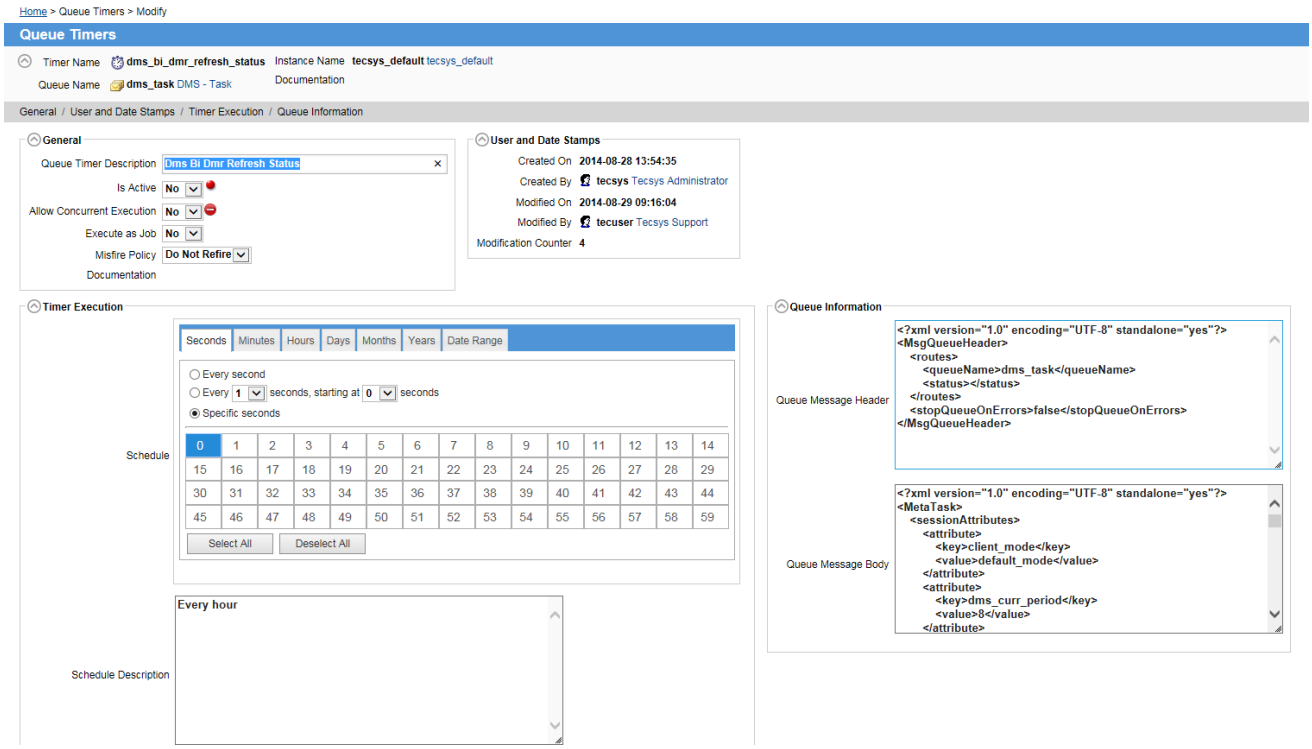


5. Click **Submit** to apply your changes.

11.4 Creating a Timer

To create a timer, do this:

1. Launch the Queue Timers resource and click **Create**.
2. Enter the required criteria and ensure the timer is created for the *tecsys_default* instance, as displayed below.



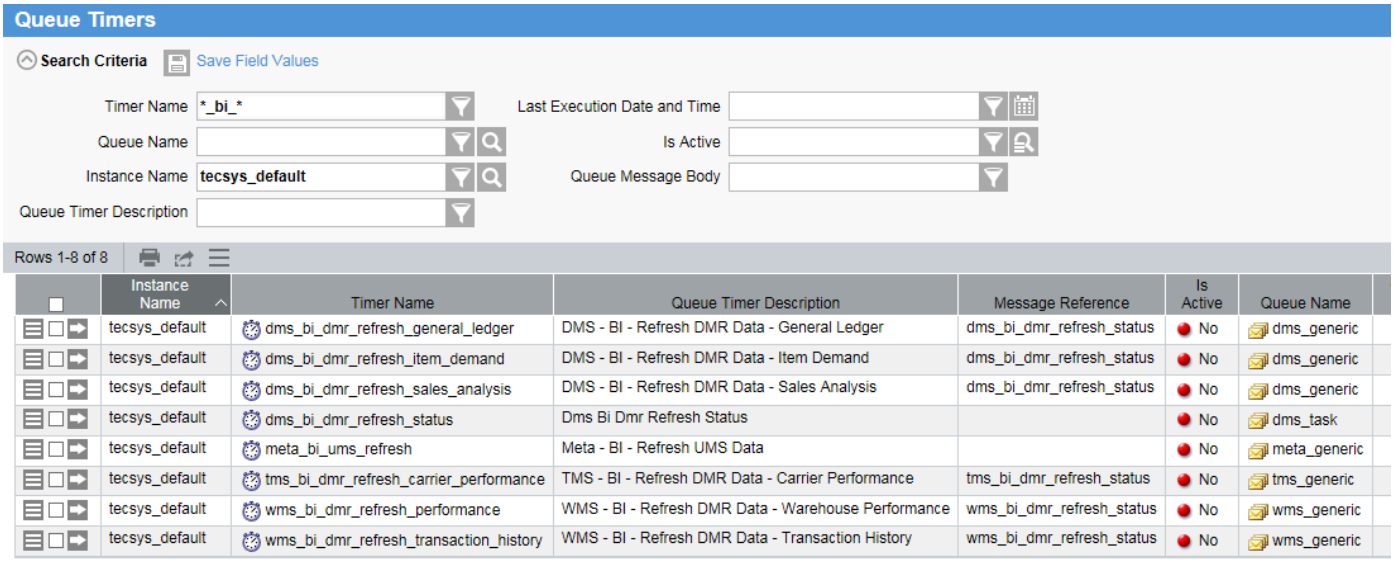
3. The Queue Message Header should be filled as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MsgQueueHeader>
    <debugKey></debugKey>
    <reference></reference>
    <routes>
        <queueName>dms_generic</queueName>
        <status></status>
    </routes>
    <stopQueueOnErrors>false</stopQueueOnErrors>
</MsgQueueHeader>
```

4. The Queue Message Body should be filled as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MetaGeneric>
    <args>general-ledger</args>
    <className>com.tecsys.bi.api.BiRefreshDmrDataExecutable</className>
</MetaGeneric>
```

5. Click **Submit** to apply your changes.



11.5 Testing

In order to ensure the previous steps were successful, do this:

1. Copy the tecsys_default timer *dms_bi_dmr_refresh_general_ledger*, and a new timer will be defined for your current instance.
2. Set the Active flag to Yes for that instance.
3. Select the **Execute Now** action (This must error out in the dms_md_mq_generic).
4. Launch *dms_bi_dmr_refresh_status*, and click Schedule Task, set up as wish the other properties.
5. Go to the md_timer view and find the newly created task created (step 4) and set the Active flag to Yes.
6.

Instance Name	Timer Name	Queue Timer Description	Message Reference	Is Active
lx6rnd.tecsys.corp_dev_91xfixes_o	dms_md_mq_generic.dms_bi_dmr_refresh_status_tecsys_dmsbidmr.task	Scheduled Task: dmsbidmr		No
lx6rnd.tecsys.corp_dev_91xfixes_o	wms_bi_dmr_refresh_performance	WMS - BI - Refresh DMR Data - Warehouse Performance	wms_bi_dmr_refresh_status	Yes
lx6rnd.tecsys.corp_dev_91xfixes_o	wms_bi_dmr_refresh_transaction_history	WMS - BI - Refresh DMR Data - Transaction History	wms_bi_dmr_refresh_status	No
lx6rnd.tecsys.corp_dev_91xfixes_o	wms_md_mq_generic.wms_bi_dmr_refresh_status_tecsys_wmsbidmr.task	Scheduled Task: wmsbidmr		Yes

7. Select the **Execute Now** action. The email is now sent.