

SIDI – VERITICO /STOCK Input Database Interface

Basic (Mandatory) Tables

WAREHOUSE_PRODUCT

Table of unique SKUs (warehouse-product combination) This is the master table for unique product-warehouse combination. All combinations listed in this table will be imported. WARNING: it is necessary to include only combinations suitable for import, all other must be excluded or labeled with warehouse_product.import=0. Combinations excluded from this table or labeled as import.=0 are not imported, even if they are recorded in other tables (promo actions, transactions).

Column	Description
warehouse_id	Warehouse identification (provided by the client)
product_id	Product identification (provided by the client)
producer_id	Reference to producer identification (provided by the client)
available_supply	Quantity On Hand. Customer orders are included; what is not included are for instance warranty claims or goods in a shop window. So, if you have say 5 items and a customer booked 2 items and in addition, there are 10 items claimed within the warranty claim, then the Quantity On Hand = 5 (bookings are deducted only for ordering, warranty claims are ignored).
physical_supply	Quantity Available in a warehouse. The term should be used ONLY if it is necessary to record the difference of Quantity On Hand and actual Quantity Available physically in a warehouse. In addition, warranty claims or goods in a shop window are included in the Quantity Available. So, using the example from the previous row, the number here would be 15. In STOCK, available_supply is 15, 10 items are blocked and 2 are customer bookings.
stock_price	Current stock unit price without VAT. It is the purchase price increased by additional costs (warehousing, transport, customs) and it should be an up-to-date figure. If the figure is not available, the current price based on market value could be used instead. A purchase price can be used as well, if the increased price is not available. However, if it is not added here, there will be zero stock on hand.

sale_price	Current unit selling price. This is the price used to multiply a non-promotional forecast. It should not be a promotion price, even if the item is in promotion now. If the price is zero, it will make the FORECAST zero as well. The price can be traced from transactions, for instance, as a weighted average of the last three month (average price, weighted by the quantity sold).
import	An auxiliary column: these rows are not to be added to the STOCK and an implementator doesn't need to delete them from the import database. It is helpful if you want, for instance, to calculate STOCK say in 2 stores and 3 categories only.
import_pwdata	Identifier of the products that are to be retained on the test instance *pw_data.
dont_order	Attribute: do not order. The quantity on hand drops to zero, the item is not going to be ordered under any circumstances, not even for a customer order.
new_product	New item (SKU). Item must have this attribute for the whole period, when it should be treated as New item.
new_product_order_amount	If the expiration reduces the order amount of a new product to 0, this quantity will be set. Additionally, if no order is suggested for the product, there are no future move (incoming orders), and available_supply <= new_product_ordering_supply, an order will be generated with this quantity.
new_product_ordering_supply	This is the threshold for generating order. Few conditions must be met: 1 - Quantity On Hand is smaller than this field (available_supply <= new_product_ordering_supply). 2 - There is no future move (incoming order) for this SKU. 3 - There is no order suggested for this SKU. 4 - This SKU does not have dont_order = 1. 5 - supplier is assigned.
selling_out	The product is under clearance: forecast is limited to the Quantity On Hand and it eventually reaches zero. The product is not to be ordered, under any circumstances.
zero_forecast	The NONPROMOTIONAL forecast should be set to zero. It is intended for products that should be ordered only under promotion or on a customer order.
zero_promo_forecast	The PROMOTIONAL forecast should be set to zero.
zero_safetystock	The Safety Stock should be set to zero
zero_promo_safetystock	The PROMOTIONAL Safety Stock should be set to zero
zero_basestock	The Base Stock should be set to zero
buyer	Name of the responsible person from our purchasing department for given product (used in some reports as well).

safetystock_min	Setting the minimum and maximum for SS and BS. Should -- after the computation -- the value of the BS or SS be outside of this interval, the value is set to one of these min/max values. Values equal to ZERO are ignored, anything else is accepted. If safetystock_min_increase/basestock_min_increase are set, the safetystock_min/basestock_min are ignored and the value of safetystock_min_increase/basestock_min_increase are added to the computed value. Safetystock_min is often used for items to display, or safetystock_min_increase is used if the display stock is not for sale.
safetystock_min_increase	see description of safetystock_min
safetystock_max	see description of safetystock_min
basestock_min	see description of safetystock_min
basestock_min_increase	see description of safetystock_min
basestock_max	see description of safetystock_min
basestock_maximum_min	see description of safetystock_min
basestock_maximum_max	see description of safetystock_min
similar_product_id	Predecessor - product similar to the current one. Fill either here or in warehouse_product_similar - in case you want to assign more than one predecessors.
similar_warehouse_id	If filled, a product similar to the current one could be taken from another warehouse. If not filled, then the product is taken from the same warehouse.
similar_percentage	If filled, a product similar to the current one is taken to X percent, following the value in the table.
similar_available_supply	If set to 1, for the purpose of orders calculation the product will sum the Stock on Hand of its similar product to its own Stock on hand.
similar_order_from_client	It there is Order from Client (in table Order_from_client) for the predecessor, can I cover this order with successor (this product). 1 - yes, 0 - no.
similarity_type_id	Similarity type ID (1=predecessor, 2=similar product)
similar_weight	If filled, a product similar to the current one is considered with the specified relative weight to other similar products.
virtual_parent_warehouse_id	If the warehouse_id of this entry is virtual, it is possible to set the column and thus determine that the parent warehouse of this virtual product is different from the parent warehouse defined by warehouse.parent_warehouse_id.

ordering_method	Ordering method. AUTO = method to be selected automatically, NORMAL = ordering based on forecast, BS = ordering based on basestock (ideal for sporadic products), OFC = ordering only when ordered by customer (no stock in warehouse)
vat_in_percent	VAT percentage rate - all other prices in the system must not include VAT.
disabled	It deactivates this product-warehouse combination in STOCK, but history of transactions is preserved. Unlike when import= 0, these plids are stored in warehouse_product_to_disable, imported into import_product_locator, and transactions into move tables (they, however, are deleted from product_locator and thus will not be displayed in STOCK). Subordinate items based on warehouse.parent_warehouse_id are also considered disabled.
mandatory	1 = mandatory item (must always be in stock). Prevents the order amount from being reduced to 0 due to expiration or creates a new order if none is proposed and the stock is below the mandatory_ordering_supply.
mandatory_ordering_amount	If the expiration reduces the order amount of a mandatory product to 0, mandatory_ordering_amount quantity will be set. Additionally, if no order is suggested for the mandatory product, there are no future move (incoming orders), and available_supply <= mandatory_ordering_supply, an order will be generated with this quantity. If NULL, one piece rounded up to package and MOQ will be ordered. If the item is new and mandatory at the same time, the higher value of mandatory_order_amount and mandatory_order_amount is used.
mandatory_ordering_supply	If a mandatory item, this value sets below which stock quantity ordering should be suggested (if less than or equal).
promo_only	1 = a promotion product only. In this case, non-promotional forecast, safetystock and basestock are set to zero.
expiration_in_days	Shelf life in days. If NULL, the value from the table product is used instead.
expiration_in_days_critical	Used to evaluate whether the quantity rounded per package will be consumed within the set time. If the pack exceeds the critical expiration time, nothing is ordered to prevent write-offs.
wacc	Percentage of costs for holding the stock in warehouse. Value from 0 to 1 (weighted average cost for capital).
inbound_cost	Cost for a receive warehouse transaction in value.
outbound_cost	Cost for a dispatch warehouse transaction in value.
stockout_level	Which level of stock is to be considered a stockout. If the value here is negative, the stockout is not to be evaluated.

category_management_group_id	The group within Category Management, to which the product belongs. See the table category_management_group.
abc_value	ABC value set for this combination product/warehouse (overwrites category from ABC analysis)
ordering_unit	Ordering unit used for new_product_ordering_supply and mandatory_ordering_amount. If specified, it must be filled in the product_measure_unit table.
use_in_mba	Determines, whether it should be imported and taken into consideration in Market Basket Analysis (MBA).
sale_set	Minimal sale set. It is used in a situation when a customer stores inventory and transactions in pieces in their ERP but sells them by whole packages, e.g., in boxes of 10 pieces that can not be unpacked. These values are used for the rounding of sales forecast and safety stock (must be enabled in the configuration)
product_locator_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

TRANSACTION

Record of all movements of goods that influence stock on hand. Including transactions such as sales, purchases, redistribution, inventory check corrections, countermovement etc. Selling to final customer must be recorded and separated from selling (moving) to child warehouse/location, which will further sell the product to final customer. Such transaction must be labeled as redistribution.

Column	Description
transaction_id	Unique transaction identifier.
transaction_type_id	For more information, see the sheet "PW Transactions".
transaction_header_id	Used to gather the transactions into a purchase blocs / invoice, etc. The transactions with the same transaction_header_id happened under the same invoice / purchase blocs. Important e.g. for Market Basket Analysis.
warehouse_id	Reference to warehouse identification
product_id	Reference to product identification
date_of_transaction	Actual date of the warehouse transaction.
amount	Quantity in the basic unit of measurement (um). For the +/- signs, see the PW transactions. In general, the +/- signs mark the goods flow in a warehouse (""+"" purchase / receipt, ""-"" sale / dispatch)

number_of_receipts	Number of transactions, if these transactions are combined/aggregated into days.
price	Unit price used for the given transaction. For purchases, export the purchase price, for sale the sales price, for transactions use the warehouse price. Use the system currency (defined in CURRENCY and CONFIGURATION.currency_id).
stock_price	The stock unit price for the transaction. Must be in the system currency.
vat_in_percent	Percentage of VAT (15 % = 0.15)
supplier_id	Mandatory for 'purchase' transactions.
customer_id	If not filled, 'Unknown customer' will be used.
cancellation_id	If you can track the cancellation to an original transaction, add the ID of the original transaction here.
promotion_id	Only in sales, if we can identify the promotion, in which the sale was made. It then facilitates evaluation of promotion sales / baseline identification. If the transaction is marked -1, it means that the sale happened in a promotion, but we fail to assign it to a specific promotion. Assignment to promotions can be turned on in recomputation.
extreme_sale_type_id	Extreme sales that should be excluded from forecast and other calculations. The extreme type list must be defined in the extreme_sale_type table. In principle, nothing excludes the possibility to use just one type everywhere. For example: a sale just before the expiration date at a reduced price, one-time sale for a event etc.
pw_order_id	For a purchase, unique order ID taken directly from STOCK, if the order was made based on STOCK suggestion. Not mandatory.
date_of_order	For a purchase, the date of the order.
move_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

WAREHOUSE

List of warehouses (in STOCK warehouse = shop)	
Column	Description
warehouse_id	Reference to warehouse identification

shortcut	For example, an original warehouse ID.
name	Name of the warehouse.
parent_warehouse_id	Deprecated from v2.19. For a central warehouse, fill NULL or 0 (there is no superordinate warehouse). You can specify parent warehouse in here (all products from this warehouse will be redistributed from parent_warehouse and you will not have any other attributes) or leave NULL and fill the table warehouse_product_redistribution.
is_virtual	1 = virtual warehouse, i.e., its parent warehouse (parent_warehouse_id) keeps inventory for all the products in the subordinate warehouse (exceptions can be set via the table warehouse_product_virtual)

PRODUCT

List of products, detailed attributes are set in warehouse_product.	
Column	Description
product_id	Reference to product identification
shortcut	Product abbreviation. The GUI will display the value that will be set when implementing the Settings refers to name or shortcut.
name	Product's full name. The GUI will display the value that will be set when implementing the Settings refers to name or shortcut.
weight	Product's weight in kg.
volume	Product's volume in m3.
expiration_in_days	Product's expiration (shelf life) in days. If filled in warehouse_product, value in this table is redundant.
rounding	How many decimals should be seen in the user interface. It could be deduced from the smallest value in transactions, for example.
base_measure_unit	Basic unit of measurement, all the data in ALL the tables must be converted into this unit.
minimal_set	The minimal quantity that can be sold. If NULL, the smallest quantity according to product.rounding is used.
product_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

SUPPLIER

List of suppliers, detailed information on suppliers are recorded in table warehouse_product_supplier etc. Raw client data often contain duplicates (e.g. two records with different VAST IDs). Input into this table must be clear of this duplicates, containing unique suppliers.

Column	Description
supplier_id	Reference to supplier identification
shortcut	Supplier's abbreviation.
name	Supplier's name.
registration_number	Supplier's Registration ID
currency_id	The currency must be filled and it is used for converting the price to the supplier's currency for orders.
client_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

CATEGORY_TYPE

Types of categories Base for building of the product-category-warehouse tree. Defines individual levels of categories, both on warehouse and product level. Each type must have only one parameter set as true (1). Example: category_type_id | name | is_product_structure | is_warehouse_structure | is_warehouse_level 1 | Shop type | 0 | 1 | 0 2 | Shop | 0 | 0 | 1 3 | Category 1 | 1 | 0 | 0 4 | Category 2 | 1 | 0 | 0 5 | Category 3 | 1 | 0 | 0 6 | Category 4 | 1 | 0 | 0

Column	Description
category_type_id	Category type identification
name	Category type (level) name
is_warehouse_structure	Categories to determine warehouses. Only one of these three can be filled, but there can be more lines for is_warehouse_structure = 1, e.g. division by country, area, type of warehouse, etc.

is_warehouse_level	Only one from (is_warehouse_structure, is_warehouse_level, is_product_structure) can be filled with 1 at one line. 1 = now dealing with individual warehouses. Only one line can have is_warehouse_level = 1
is_product_structure	Catetegories to determine products. Only one of these three can be filled, but there can be more lines for is_product_structure = 1. e.g. General product category (dairy products, bakery), Detailed product category (jogurts, cheese, butter) etc.

CATEGORY

List of product categories. This table contains complete warehouse and product structure (tree hierarchy). That means record of all warehouse types, warehouses, shop types, product categories and so on.

Column	Description
category_id	Category identification (provided by the client)
category_type_id	All stores (warehouses) must have identical category_type_id and that must equal import.configuration.warehouse_category_type_id.
name	Name of the category (e.g. Czech retail branch, Hypermarkets, Dairy products, Cheese, Blue cheese)
shortcut	Abbreviation of the customer's name. Display in the GUI depends on the settings in the configuration.
hidden	Tree menu settings. 1 = this category won't be displayed. Elements of this category will be displayed directly in the higher category.
category_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

PRODUCT_CATEGORY

Product listing under categories. Each product must be listed under product categories on all levels (only categories with is_product_structure = 1 type).

Column	Description
product_id	To a single product, categories at all levels (i.e., the levels in is_product_structure) must be defined. Thus, each product will have as many entries in the table, as many product category types there are.

category_type_id	Reference to category type. Forms a compound key with category_id, which may not be unique
category_id	Category identification (provided by the client)

CURRENCY

List of used currencies. Must be defined even if only one currency is used. Moreover, default currency must be defined using the CONFIGURATION.currency_id. All transactions in STOCK must be in the default currency. The different currencies only serves the purpose that you want to have STOCK GUI displayed in different currency.

Column	Description
currency_id	CZK,USD,EUR,...
default	Exactly one currency must be set as default = 1.

CONFIGURATION

Table for various configuration data. Contains two columns: key + value. For the definition of possible keys and their values see CONFIGURATION enumeration.

Column	Description
key	Key of configuration parameter, as listed in the CONFIGURATION enumeration
value	Value for the parameter specified in key field
description	--obsolete

WAREHOUSE_TEMPORARILY_CLOSED

Warehouses which were, or currently are, temporarily closed.

Column	Description
warehouse_id	Reference to warehouse identification.

date_from

Warehouse closing date.

date_to

Warehouse re-opening date.

pt_promo

Table for promotional actions

Column	Description
id	Primary key
pt_promo_id	Unique promo action code
pt_promo_type_id	Promo type ID
name	Name of the promo action
promo_date_from	Promo start date
promo_date_to	Promo end date
delivery_date_from	Delivery start date
delivery_date_to	Delivery end date
pt_wf_id	Workflow ID
pt_wf_date	Workflow date
note	Additional notes
commercial_costs_coverage	Commercial costs coverage
commercial_costs	Commercial costs
sent_in_sap	Sent in SAP
ignore_for_training	Ignore for training
currency_id	Currency ID

wf_pwuser_id	Workflow user ID
last_change_pwuser_id	Last change user ID
author_pwuser_id	Author user ID
owner_pwuser_id	Owner user ID
wf_note	Workflow note

pt_promo_product

Table for products associated with promotional actions

Column	Description
pt_promo_id	Unique promo action code
product_id	Product ID
amount	Amount
amount_real	Real amount
list_price	List price
ptr_promo	PTR promo
ptr_nonpromo	PTR non-promo
ptc_promo	PTC promo
ptc_nonpromo	PTC non-promo
vic	VIC
vlc	VLC
excise	Excise

list_price

Table for list prices of products	
Column	Description
product_id	Product ID
value	Value
value_excise	Value excise
value_vat	Value VAT
date_from	Date from
date_to	Date to
price_package	Price package

pt_promo_category

Table for linking promo actions to categories	
Column	Description
pt_promo_id	Unique promo action code
category_id	Category ID
category_type_id	Category type ID
valid	This tag is valid if we can find join in category

pt_nonpromo_discount

Table for non-promotional discounts linked to promo actions

Column	Description
pt_promo_id	Unique promo action code
product_id	Product ID
discount_type_id	Discount type ID
value_percent	Percentage value of the discount
value_absolute	Absolute value of the discount

pt_promo_placement_type

Table for promo placement types

Column	Description
id	Primary key auto-increment ID
name	Name of the promo placement type
description	Description of the promo placement type

pt_promo_placement

Table for promo placements linked to promo actions

Column	Description
pt_promo_id	Unique promo action code
pt_promo_placement_type_id	Promo placement type ID
date_from	Start date of the promo placement
date_to	End date of the promo placement

pt_promo_discount_type

Table for promo discount types	
Column	Description
id	Primary key auto-increment ID
name	Name of the promo discount type
shortcut	Shortcut for the promo discount type
default	Indicates if this is the default promo discount type
type	Type of the discount (discount or bonus)
value	Value type of the discount (percent or absolute)
gui_read_only	Flag indicating if this discount type is read-only in the GUI
export	Flag indicating if this discount type should be exported

pt_promo_discount

Table for promo discounts	
Column	Description
pt_promo_id	Unique promo action code
product_id	ID of the product
pt_promo_discount_type_id	ID of the promo discount type
value_percent	Percentage value of the discount
value_absolute	Absolute value of the discount

warehouse_product_attribute_date

Table for storing warehouse product attribute validity dates

Column	Description
warehouse_id	Warehouse identifier
product_id	Product identifier
valid_from	Start date of validity period
valid_to	End date of validity period (optional)
attribute_date_id	Attribute identifier
value	Value of the attribute

Extending Tables

CUSTOMER

List of customers, contains basic information on all customers if orders from clients (table ORDER_FROM_CLIENT) are in use or if clients are recorded within selling transactions.

Column	Description
customer_id	Reference to customer identification
shortcut	Abbreviation of customer name
name	Customer's name.
registration_number	Customer's Registration ID

client_id_in_pw

Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

PRODUCER

List of producers, this table is not used in STOCK unless client demanded some custom made reports.

Column	Description
producer_id	Producer identification as provided by the client
shortcut	Producer's name abbreviation
name	Producer's name
registration_number	Producer's Registration ID
producer_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

WAREHOUSE_CATEGORY

Warehouse listing under categories. Each warehouse must be listed under warehouse categories on all levels (only categories with is_warehouse_structure = 1 type). If there is no warehouse hierarchy, table should be empty.

Column	Description
warehouse_id	Reference to warehouse identification (provided by the client)
category_type_id	Category_type_id of a category with is_warehouse_structure=1; if no such exist, this table is empty.
category_id	Reference to category type. Forms a compound key with category_id, which may not be unique

FUTURE_DELIVERY

Stock in transit - All open orders from central warehouse to supplier or from shop to central warehouse. There are 2 purposes of this table:
 1) Calculation of lead times of suppliers (not calculated for warehouses)
 2) Keeping record of all open orders and thus stock in transit. This is critical for proper suggesting of orders, preventing duplicates and overstocking.

Column	Description
future_delivery_id	Unique identifier of Order in Transit, provided by the client.
product_id	Reference to product identification
warehouse_id	Reference to warehouse identification
supplier_id	If this is an order to a supplier, then fill in supplier's ID. If this is an order to a parent warehouse, fill in NULL.
parent_warehouse_id	If this is an order to a parent warehouse (intra-warehouse distribution), fill in the ID of the warehouse from which the product is distributed.
date_of_order	Order date.
estimated_date_of_delivery	Exepcted delivery date.
amount	Quantity ordered.
purchase_price	Purchase unit price. If 0, it will be calculated automatically.
accept_for_lt_first_shift	This field determines if I can have one - or more than one orders for particular product at a time (if I want to order, should I wait until the supplier deliveres the previous order). It can be set globally (order_configuration.enable-leadtime-first-shift-redistribution / provider / production), the options are YES, NO, Depending on the order. If depending on the order is chosen, than in this field is used. 0 - don't wait until this order is fullfilled, 1 - this order must be delivered first and than you can order a next order
shortcut	Code of the order that will be displayed in STOCK system in the list of Orders in Transit under Product Detail.
future_move_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

FUTURE_DELIVERY_HISTORY

History of created orders to suppliers and/or parent warehouses. Can contain delivered orders as well as orders in transit.

Column	Description
future_delivery_history_id	identification of order, provided by the client
product_id	Reference to product identification
warehouse_id	Reference to warehouse identification
supplier_id	If this is an order to a supplier, it must be filled.
parent_warehouse_id	If this is an order to a parent warehouse, it must be filled.
is_redistribution	1 = order to a parent warehouse, 0 = order to a supplier
date_of_order	Order date.
estimated_date_of_delivery	Expected delivery date.
amount	Quantity ordered.
pw_order_id	Unique order ID obtained from the STOCK system, if the order was created based on STOCK suggestion. Not mandatory.
real_order_history_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

ORDER_FROM_CLIENT

Active orders - All open orders from clients or child warehouses. Table Future_delivery records the products which I have ordered and are still not delivered. This table records products which someone else ordered from me and are still to be delivered. Redistribution between my warehouses should be kept in both tables as it was ordered by me and also not delivered.

Column	Description
order_from_client_id	OFC (Order-from-client) identification

product_id	Reference to product identification
warehouse_id	Reference to warehouse identification
customer_id	Reference to customer identification
date_of_order	Order date
estimated_date_of_delivery	Estimated date when this product will be shipped from my warehouse
amount	Ordered quantity
sale_price	Selling unit price
is_redistribution	1 = is redistribution
target_warehouse_id	If is_redistribution = 1, identifies the warehouse which ordered
shortcut	Order code, displayed in STOCK in the list of Orders From Customer on product detail page.
accept_only_ofc_bigger_than_forecast	0 = Increase forecast by the amount of this order, 1 = Update forecast just to cover this order (if amount > forecast), 2 = Replace forecast with the amount of this order.
order_from_client_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

DELIVERY_FOR_LEADTIME

Data for lead time calculation - Table contains historical data of purchase transactions. If this table is not filled, leadtime can be imported from warehouse_product_supplier or can be taken from GUI ""Periodical ordering settings - ordering attributes for suppliers"". If filled, the GUI settings has priority over the UDM. Lead time calculations is influenced by configuration stored in leadtime_calculation_method. Can contain all items (if order date cannot be matched to specific purchase transaction), or open transactions only.

Column	Description
delivery_for_leadtime_id	Delivery for leadtime identification as provided by the client
delivery_header_id	ID of the order

delivery_row_id	ID of line within the order
warehouse_id	Reference to warehouse identification
product_id	Reference to product identification
supplier_id	Reference to supplier identification
date_of_order	Order date
required_date_of_delivery	Required delivery date
confirmed_date_of_delivery	Delivery date, confirmed by the supplier
date_of_delivery	Actual delivery date, used for leadtime calculation
purchase_price	Unit price of the order, only for information
confirmed_amount	Ordered quantity, confirmed by the supplier
ordered_amount	Actual ordered quantity
delivered_amount	Actual delivered quantity. In case of negative value, the record is treated as cancellation and will be cleared afterwards.
delivery_for_leadtime_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)
delivery_for_leadtime_header_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)
delivery_for_leadtime_row_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

WAREHOUSE_PRODUCT_SIMILAR

Configuration of product predecessors - Enhancement of predecessor/successor functionality. Configuration enables setting up multiple predecessors (the new product replaces more old products). Can be also used for setting up a predecessor for a product on all warehouses/levels at once (warehouse_id and similar_warehouse_id will be left empty). If an entry for the same product is found both in warehouse_product and warehouse_product_similar, the system prioritizes information in warehouse_product table.

Column	Description
product_id	ID of Successor
warehouse_id	If left empty, predecessor for the given product is set on all warehouses where located.
similar_product_id	Another product similar to this product – to be treated as a predecessor.
similar_warehouse_id	If filled, a product similar to the current one is taken from another determined warehouse. If empty, the product is taken from the current warehouse.
similar_percentage	If filled, a product similar to the current one is taken up to X percent. If product has more predecessors the sum is advised to make 100% but not necessary. Also single predecessor can have more than 100%.
similar_available_supply	If set to 1, the product ordering includes the predecessor's stock (visible in GUI: Detail -> Graph of the daily course to the future). But the available supply is not changed.
similar_order_from_client	If set to 1, the product assumes customer orders and orders from daughter warehouses of its predecessor (from the table order_from_client).
similarity_type_id	Similarity type ID (1=predecessor, 2=similar product)
similar_weight	If filled, a product similar to the current one is considered with the specified relative weight to other similar products.

WAREHOUSE_PRODUCT_SUPPLIER

This table defines the supplier and ordering attributes for SKU. Most of the field related to ordering (minimal_order_quantity, minimal_order_set, minimal_period, minimal_period_bs, minimal_period_ss, leadtime, leadtime_bs) can be set in GUI ""Periodical ordering settings - ordering attributes for suppliers"" as well. If filled, the GUI settings has priority over the UDM

Column	Description
warehouse_id	Reference to warehouse identification
product_id	Reference to product identification
supplier_id	Reference to supplier identification
variant	Variant of ordering attributes for the same warehouse, product, and supplier. If any row is filled for a warehouse, product, and supplier combination, that combination must contain variant 1. Variant 1 is used by standard ordering; another variant can be selected by WAREHOUSE_PRODUCT_SUPPLY_SOURCE_INTERVAL. The same variant can be used in the supplier delivery and ordering set tables.
enabled	0 = SKU ordering from this supplier is inactive, cf. this SKU is not ordered from the supplier.
purchase_price	Price for which the product is bought from the supplier for this warehouse. Purchase unit price. If not specified, it can be calculated from transactions using the recomputeProductLocatorPricesAvg method.
minimal_order_quantity	Minimal order quantity. The value is not ignored even if warehouse_product_supplier_set is filled.
minimal_order_set	Minimal order set -- a packaging unit. If we want to use different sets for different results, fill in WAREHOUSE_PRODUCT_SUPPLIER_SET. This value is then ignored. The relation between minimal_order_quantity and minimal_order_set is the following equation: minimal_order_quantity: 100, minimal_order_set: 10, What can be ordered is 100, 110, 120, ...
min_set_fill_ratio	a minimal threshold in percent for an order to be rounded up to the next higher multiple of minimal_order_set. If the amount entering the rounding function exceeds the previous full set by a percentage smaller than this threshold, the order is rounded down to the nearest smaller multiple of minimal_order_set, but never to zero. E.g., with minimal_order_set = 100 and min_set_fill_ratio = 20, an amount of 215 is rounded down to 200 (only 15 % above), 220 is rounded up to 300 (20 % above), and 55 is rounded up to 100 (never to zero)
recompute_set	0= minimal_order_quantity and minimal_order_set is not recalculated; 1= it is recalculated only if the conditions recompute_set_history_length_in_days and recompute_set_percentage_of_lower_orders are met. Conditions are set in SIDI
minimal_period	If ordered, how long should the ordered quantity last. It strongly affects BS, SS and the resulting stock level. If not selected, use the closest possible date for delivery otherwise Safety Stock is calculated incorrectly.
minimal_period_bs	Value of period to calculate BS. If NULL, minimal_period will be used.
minimal_period_ss	Value of period to calculate SS. If NULL, minimal_period will be used.

leadtime	Count of days from order date to delivery date. It is used, if the order method is OD-LT (order date and leadtime), otherwise everything is governed by order and delivery dates. If table Delivery_for_leadtime is used this value wont be taken for ordering purposes but will be displayed in reports (comparison the promised LT and real LT). LT must not be 0, otherwise error will occure.
leadtime_std	If table Delivery_for_leadtime is not filled, and you have the data, you can specifiy standard deviaion of LT. As a result Safetystock due to lead-time error will be calculated.
leadtime_bs	Leadtime used to compute BS. If NULL, the usual leadtime will be used.
recompute_leadtime	1 = recompute, if possible. If there are not enough data, leadtime doesn't get recomputed. 0 = do not recompute.
recompute_leadtime_std	1 = recompute, if possible. If there are not enough data, leadtime doesn't get recomputed. 0 = do not recompute.
preferred	Preferred supplier, only 1 is allowed for a combination warehouse/product, if the supplier is preffered, the product will be ordered only from this supplier.
ordering_type_variant	Mode of ordering: OD-LT - ordered according to delivery date and time of delivery as defined by leadtime. OD-DD - ordered strictly by the date of order and delivery, leadtime is ignored. CALENDAR - Ordered strictly based on the specific dates provided by the user for order and delivery
ordering_number	The name or ID of the product used by the supplier.
ordering_unit	Ordering unit. If specified, it must be filled in the product_measure_unit table.
post_delivery_processing_days	Number of days after delivery before the stock is considered usable for ordering.

WAREHOUSE_PRODUCT_SUPPLIER_SET

Definition of advanced rules for ordering quantitties. This configuration enables setting up rules for ordering: e.g. when order quantity is smaller or bigger than a parameter, the system suggests ordering a unit different from basic setup in the table warehouse_product_supplier.

Column	Description
warehouse_id	Reference to warehouse identification
product_id	Reference to product identification
supplier_id	Reference to supplier identification

variant	Variant of the supplier ordering set. It must match WAREHOUSE_PRODUCT_SUPPLIER.variant.
priority	Rule priority for each combination warehouse_id, product_id a supplier_id. 1 = the highest.
type	Determines whether the from and to columns are in absolute quantity or percent; values are AMOUNT or PERCENT.
from	Lower limit, not compulsory.
to	Upper limit, not compulsory.
reference	Reference value to compute limits from percent to absolute value. Relevant only if type = PERCENT.
set	Number to which the product quantity is rounded
only_for_expiration	The set/packing is used only if the product expires when the regular set is used.
ordering_unit	Ordering unit. If specified, it must be filled in the product_measure_unit table.
min_fill_ratio	a minimal threshold in percent for an order to be rounded up to the next higher multiple of set. If the amount entering the rounding function exceeds the previous full set by a percentage smaller than this threshold, the order is rounded down to the nearest smaller multiple of set, but never to zero. E.g., with set= 100 and min_fill_ratio = 20, an amount of 215 is rounded down to 200 (only 15 % above), 220 is rounded up to 300 (20 % above), and 55 is rounded up to 100 (never to zero).

WAREHOUSE_PRODUCT_SUPPLIER_DELIVERY

Settings of exact ordering and delivery days (for a supplier). This table defines exact ordering and delivery days, alternatively, the table defines ordering days only while delivery is calculated based on lead time stored in warehouse_product_supplier. Instead of this table setting in GUI ""Periodical ordering settings - ordering attributes for suppliers"" can be used. If filled, the GUI settings has priority over the UDM.

Column	Description
warehouse_id	Reference to warehouse identification
product_id	If product_id is 0, it is the default setting for a store and a supplier. If another value, then this setting applies for a particular product.
supplier_id	Reference to supplier identification

variant	Variant of the supplier delivery settings. It must match WAREHOUSE_PRODUCT_SUPPLIER.variant.
delivery_id	If there is need for more rules/settings for a combination product, warehouse and supplier, this parameter distinguishes the rules.
even_odd	E = even, O = odd, B = both. If you are setting a calendar exception (exception_date_from != 0000-00-00), this value is ignored.
day_for_ordering	1 is Monday, 7 Sunday; the order day must be set, since it is used in OD-LT as well as in OD-DD.
day_for_delivery	1 or 8 is MONDAY, 2 or 9 is TUESDAY... 7 or 14 is SUNDAY. This is the closest possible delivery day with respect to the order day (day_for_ordering). For example, day_for_ordering = 3 and day_for_delivery = 1 means that when you order on Wednesday, 5 days later, on Wednesday, is the delivery day. If day_for_delivery = 8, then delivery actually happens only in 12 days on Monday. day_for_ordering = 1 and day_for_delivery = 1 indicates the same day delivery, if day_for_delivery = 8, the delivery happens only in a week, on Monday.
day_in_month	It determines the order day in a month. It could be combined with day_for_ordering. For example, day_for_ordering = 1 and day_in_month = 5, then order will happen on Monday closest to the 5th day in a month. If this column is filled (cf. it is not NULL), the columns day_for_delivery and even_odd are ignored. It cannot be combined with columns exception_*.
minimal_period	How long the stock should last. It is used when the delivery days are spread densely. If not selected, use the closest possible date for delivery otherwise Safety Stock is calculated incorrectly.

wp_supplier_delivery_date

Ordering calendar for plids and suppliers with the exact specification of ordering and delivery days in the format of a specific date	
Column	Description
warehouse_id	Reference to warehouse identification
product_id	If product_id is 0, it is the default setting for a store and a supplier. If another value, then this setting applies for a particular product.
supplier_id	Reference to supplier identification

date_for_ordering

Order date ? today

date_for_delivery

Delivery date ? date_for_ordering

wp_supplier_delivery_exception

Complementary table for table WAREHOUSE_PRODUCT_SUPPLIER_DELIVERY. This table specifies the exceptions for the given time period. Instead of this table setting in GUI ""Periodical ordering settings - ordering attributes for suppliers"" can be used.

Column	Description
warehouse_id	Reference to warehouse identification
product_id	If product_id is 0, it is the default setting for a store and a supplier. If another value, then this setting applies for a particular product.
supplier_id	Reference to supplier identification
delivery_id	If there is need for more rules/settings for a combination product, warehouse and supplier, this parameter distinguishes the rules.
day_for_ordering	1 is Monday, 7 Sunday; If order day and delivery day are NULL then exception functions as dont order. Otherwise the order day must be set, since it is used in OD-LT as well as in OD-DD.
day_for_delivery	1 or 8 is MONDAY, 2 or 9 is TUESDAY... 7 or 14 is SUNDAY. This is the closest possible delivery day with respect to the order day (day_for_ordering). For example, day_for_ordering = 3 and day_for_delivery = 1 means that when you order on Wednesday, 5 days later, on Wednesday, is the delivery day. If day_for_delivery = 8, then delivery actually happens only in 12 days on Monday. day_for_ordering = 1 and day_for_delivery = 1 indicates the same day delivery, if day_for_delivery = 8, the delivery happens only in a week, on Monday.
minimal_period	How long the stock should last. It is used when the delivery days are spread densely. If not selected, use the closest possible date for delivery otherwise Safety Stock is calculated incorrectly.
exception_date_from	If the date format is 0000-00-00, it is a usual setting for order days. If filled, then this is an exception that - in the given period (exception_date_from to exception_date_to) - fully replaces the delivery schedule.
exception_date_to	If the date format is 0000-00-00, it is a usual setting for order days. If filled, then this is an exception that - in the given period (exception_date_from to exception_date_to) - fully replaces the delivery schedule.
exception_type	If OD, then all the order days in a given period are replaced; if DD, all delivery days in the given period are replaced.

WAREHOUSE_PRODUCT_REDISTRIBUTION

Settings for replenishment by parent warehouse. For the SKU, internal replenishment from a parent warehouse is set up by this table. Such replenishment will be taken into account when ordering from supplier for the parent warehouse, so that the internal demand is also covered. Direct replenishment by suppliers is set up using the table warehouse_product_supplier. If both tables are filled for a SKU, this table has priority, but it is recommended to have only one of those tables filled for one particular SKU. The rules can be set in GUI ""Periodical ordering settings - ordering attributes for redistribution"" as well. If filled, the GUI settings has priority over the UDM

Column	Description
warehouse_id	Reference to warehouse identification
product_id	Reference to product identification
parent_warehouse_id	If NULL, then the one defined in the table WAREHOUSE is taken. So if you want to have redistribution set only for some products and the rest should be delivered by supplier, WAREHOUSE.parent_warehouse_id must be NULL.
variant	Variant of redistribution ordering attributes for the same warehouse and product. Variant 1 is used by standard ordering; another variant can be selected by WAREHOUSE_PRODUCT_SUPPLY_SOURCE_INTERVAL. Use a separate variant when switching to another source warehouse. The same variant can be used in the redistribution delivery and ordering set tables.
minimal_order_quantity	Minimal order quantity. The value is not ignored even if warehouse_product_supplier_set is filled.
minimal_order_set	Minimal order set -- a packaging unit. If we want to use different sets for different results, fill in WAREHOUSE_PRODUCT_SUPPLIER_SET. This value is then ignored. The relation between minimal_order_quantity and minimal_order_set is the following equation: minimal_order_quantity: 100, minimal_order_set: 10, What can be ordered is 100, 110, 120, ...
min_set_fill_ratio	a minimal threshold in percent for an order to be rounded up to the next higher multiple of minimal_order_set. If the amount entering the rounding function exceeds the previous full set by a percentage smaller than this threshold, the order is rounded down to the nearest smaller multiple of minimal_order_set, but never to zero. E.g., with minimal_order_set = 100 and min_set_fill_ratio = 20, an amount of 215 is rounded down to 200 (only 15 % above), 220 is rounded up to 300 (20 % above), and 55 is rounded up to 100 (never to zero).

recompute_set	0 = minimal_order_quantity and minimal_order_set is not recomputed at all; 1= recomputed only if conditions set in recompute_set_history_length_in_days and recompute_set_percentage_of_lower_orders are met
minimal_period	If ordered, how long should the ordered quantity last. If not selected, use the closest possible date for delivery otherwise Safety Stock is calculated incorrectly. It strongly affects BS, SS and the resulting stock level.
minimal_period_bs	Value of period to calculate BS. If NULL, minimal_period will be used.
minimal_period_ss	Value of period to calculate SS. If NULL, minimal_period will be used.
leadtime	Period of delivery counting from the ordering. It is used, if the order method is OD-LT, otherwise order and delivery dates are used. LT must not be 0, otherwise error will occur.
leadtime_std	LT deviation. It affects safety stock.
leadtime_bs	Leadtime used to compute BS. If NULL, usual leadtime will be used.
ordering_type_variant	Mode of ordering: OD-LT - ordered according to delivery date and time of delivery as defined by leadtime. OD-DD - ordered strictly by the date of order and delivery, leadtime is ignored. CALENDAR - Ordered strictly based on the specific dates provided by the user for order and delivery
ordering_unit	Ordering unit. If specified, it must be filled in the product_measure_unit table.

WAREHOUSE_PRODUCT_REDISTRIBUTION_SET

Definition of advanced rules for redistribution quantities. This configuration enables setting up rules for ordering: e.g. when order quantity is smaller or bigger than a parameter, the system suggests ordering a unit different from basic setup in the table warehouse_product_supplier.

Column	Description
warehouse_id	Reference to warehouse identification
product_id	Reference to product identification
parent_warehouse_id	Source warehouse for which the ordering set applies. Leave empty to use the row as a fallback for every source warehouse. A source-specific row has priority over the fallback row with the same variant and priority.

variant	Variant of the redistribution ordering set.
priority	Rule priority for each combination of warehouse_id and product_id. 1 = the highest
type	It determines whether the columns from and to are in absolute quantity or percent; values are AMOUNT or PERCENT.
from	Lower limit, not compulsory.
to	Upper limit, not compulsory.
reference	Reference value for calculating limits from percentage to absolute value. Relevant only if type = PERCENT. Percentages are taken as ratio
set	Number to which the product quantity is rounded.
only_for_expiration	The set/packing is used only in when there is a danger of expiration.
ordering_unit	Ordering unit. If specified, it must be filled in the product_measure_unit table.
min_fill_ratio	a minimal threshold in percent for an order to be rounded up to the next higher multiple of set. If the amount entering the rounding function exceeds the previous full set by a percentage smaller than this threshold, the order is rounded down to the nearest smaller multiple of set, but never to zero. E.g., with set= 100 and min_fill_ratio = 20, an amount of 215 is rounded down to 200 (only 15 % above), 220 is rounded up to 300 (20 % above), and 55 is rounded up to 100 (never to zero).

WAREHOUSE_PRODUCT_REDISTRIBUTION_DELIVERY

Settings of exact ordering and delivery days (for a redistribution). This table defines exact ordering and delivery days, alternatively, the table defines ordering days only while delivery is calculated based on lead time stored in warehouse_product_supplier. Instead of this table setting in GUI ""Periodical ordering settings - ordering attributes for redistribution"" can be used. If filled, the GUI settings has priority over the UDM.

Column	Description
warehouse_id	Reference to warehouse identification
product_id	If product_id is 0, it is the default setting for a store and a supplier. If another value, then this setting applies for a particular product.

parent_warehouse_id	Source warehouse for which the delivery settings apply. Leave empty to use the row as a fallback for every source warehouse. A source-specific row has priority over the fallback row with the same variant and delivery_id.
variant	Variant of the redistribution delivery settings.
delivery_id	Sequence number of the record for the combination warehouse and product.
even_odd	E = even, O = odd, B = both
day_for_ordering	1 is Monday, 7 Sunday
day_for_delivery	1 or 8 is MONDAY, 2 or 9 is TUESDAY... 7 or 14 is SUNDAY. This is the closest possible delivery day with respect to the order day (day_for_ordering). For example, day_for_ordering = 3 and day_for_delivery = 1 means that when you order on Wednesday, 5 days later, on Wednesday, is the delivery day. If day_for_delivery = 8, then delivery actually happens only in 12 days on Monday. day_for_ordering = 1 and day_for_delivery = 1 indicates the same day delivery, if day_for_delivery = 8, the delivery happens only in a week, on Monday.
day_in_month	It determines the order day in a month. It could be combined with day_for_ordering. For example, day_for_ordering = 1 and day_in_month = 5, then order will happen on Monday closest to the 5th day in a month. If this column is filled (cf. it is not NULL), the columns day_for_delivery and even_odd are ignored. It cannot be combined with columns exception_*.
minimal_period	How long the stock should last. It is used when the delivery days are spread densely. If not selected, use the closest possible date for delivery otherwise Safety Stock is calculated incorrectly.

wp_redistribution_delivery_date

Ordering calendar for redistribution with the exact specification of ordering and delivery days in the format of a specific date

Column	Description
warehouse_id	Reference to warehouse identification

product_id	If product_id is 0, it is the default setting for a store and a supplier. If another value, then this setting applies for a particular product.
date_for_ordering	Order date ? today
date_for_delivery	Delivery date ? date_for_ordering

wp_redistribution_delivery_exception

Complementary table for table WAREHOUSE_PRODUCT_REDISTRIBUTION_DELIVERY. This table specifies the exceptions for the given time period. Instead of this table setting in GUI ""Periodical ordering settings - ordering attributes for suppliers"" can be used.

Column	Description
warehouse_id	Reference to warehouse identification
product_id	If product_id is 0, it is the default setting for a store and a supplier. If another value, then this setting applies for a particular product.
delivery_id	Sequence number of the record for the combination warehouse and product.
day_for_ordering	1 is Monday, 7 Sunday
day_for_delivery	1 or 8 is MONDAY, 2 or 9 is TUESDAY... 7 or 14 is SUNDAY. This is the closest possible delivery day with respect to the order day (day_for_ordering). For example, day_for_ordering = 3 and day_for_delivery = 1 means that when you order on Wednesday, 5 days later, on Wednesday, is the delivery day. If day_for_delivery = 8, then delivery actually happens only in 12 days on Monday. day_for_ordering = 1 and day_for_delivery = 1 indicates the same day delivery, if day_for_delivery = 8, the delivery happens only in a week, on Monday.
minimal_period	How long the stock should last. It is used when the delivery days are spread densely. If not selected, use the closest possible date for delivery otherwise Safety Stock is calculated incorrectly.
exception_date_from	If the date format is 0000-00-00, it is a usual setting for order days. If filled, then this is an exception that - in the given period (exception_date_from to exception_date_to) - fully replaces the delivery schedule.
exception_date_to	see description of exception_date_from
exception_type	If OD, then all the order days in a month are replaced; if DD, all delivery days in the given period are replaced.

WAREHOUSE_PRODUCT_PRODUCTION_COMPONENT

Bill of Materials (or set of components of a product). Component (component_product_id) from the warehouse (component_warehouse_id) is used to produce a product (product_id) in a warehouse/plant (warehouse_id)

Column	Description
warehouse_id	At this warehouse, there is a finished product with product_id, made from the component_product_id.
product_id	This product is made from the product defined in column component_product_id.
component_warehouse_id	From this warehouse, components component_product_id to make product_id are taken.
component_product_id	The product defined in column product_id is made from this product.
amount	Quantity of product component_product_id needed to produce product product_id.
leadtime	How many days before the final product is completed does this component have to be available. The value must be at most equal to the leadtime of the final product. If NULL, the final product leadtime is used. LT must not be 0, otherwise error will occur.

WAREHOUSE_PRODUCT_PRODUCTION

Setting of manufacturing specifics. In this table, you can specify batch sizes, production time etc.

Column	Description
warehouse_id	Reference to warehouse identification
product_id	Reference to product identification
leadtime	Production time. LT must not be 0, otherwise error will occur.
leadtime_std	Standard deviation of the production time.

leadtime_bs	Leadtime used to compute BS. If NULL, usual leadtime will be used.
minimal_period	For how many days the final product quantity (one batch) should lasts. It determines the quantity (batch size) for production if the batch sizes are not strictly given.
minimal_production_quantity	Minimal quntity of the product that must be produced.
minimal_production_set	How many unists are produced at the same time.
min_set_fill_ratio	a minimal threshold in percent for an order to be rounded up to the next higher multiple of minimal_order_set. If the amount entering the rounding function exceeds the previous full set by a percentage smaller than this threshold, the order is rounded down to the nearest smaller multiple of minimal_order_set, but never to zero. E.g., with minimal_order_set = 100 and min_set_fill_ratio = 20, an amount of 215 is rounded down to 200 (only 15 % above), 220 is rounded up to 300 (20 % above), and 55 is rounded up to 100 (never to zero).
zero_supply	1 = there is no stock for the final product, stock is held only for components.
ordering_type_variant	Mode of ordering from production: OD-LT - ordered according to delivery date and production time defined by leadtime. OD-DD - ordered strictly by the date of order and delivery date (finished date), production time is ignored. CALENDAR - Ordered strictly based on the specific dates provided by the user for order and delivery
ordering_unit	Ordering unit. If specified, it must be filled in the product_measure_unit table.

WAREHOUSE_PRODUCT_PRODUCTION_SET

Definition of advanced rules for manufacturing quantities (batch sizes). Similar to warehouse_product_supplier_set, but defined for the case of production.
https://docs.google.com/document/d/1Vh_Of0zy2t7sWNoTVTIJfeVUjn7QKgzn_c3T11th3JI/edit?usp=sharing

Column	Description
warehouse_id	Reference to warehouse identification
product_id	Reference to product identification
priority	Rule priority for each combinaiton warehouse_id, product_id a supplier_id. 1 = the highest.
type	It determines whether the from and to columns are in absolute quantity or percent; values are AMOUNT or PERCENT.

from	Lower limit, not compulsory.
to	Upper limit, not compulsory.
reference	Reference value to recompute limits from percent to absolute value. Relevant only if type = PERCENT.
set	Number to which the product quantity is rounded.
only_for_expiration	The set/packing is used only in when there is a danger of expiration.
ordering_unit	Ordering unit. If specified, it must be filled in the product_measure_unit table.
min_fill_ratio	a minimal threshold in percent for an order to be rounded up to the next higher multiple of set. If the amount entering the rounding function exceeds the previous full set by a percentage smaller than this threshold, the order is rounded down to the nearest smaller multiple of set, but never to zero. E.g., with set= 100 and min_fill_ratio = 20, an amount of 215 is rounded down to 200 (only 15 % above), 220 is rounded up to 300 (20 % above), and 55 is rounded up to 100 (never to zero).

WAREHOUSE_PRODUCT_PRODUCTION_DELIVERY

Settings of exact ordering and manufacturing dates. There are two ways of setting up ordering days for manufacturing. This table defines exact ordering and manufacturing days, or ordering days only while lead time is added from warehouse_product_production table (in that case warehouse_product_production.ordering_type_variant should be set to OD-LT)

Column	Description
warehouse_id	Reference to warehouse identification
product_id	If product_id is 0, it is the default setting for a store and a supplier. If another value, then this setting applies for a particular product.
delivery_id	Sequence number of the record for the combination warehouse and product.
even_odd	E = even, O = odd, B = both
day_for_ordering	1 is Monday, 7 Sunday

day_for_delivery

1 or 8 is MONDAY, 2 or 9 is TUESDAY... 7 or 14 is SUNDAY. This is the closest possible delivery day with respect to the order day (day_for_ordering). For example, day_for_ordering = 3 and day_for_delivery = 1 means that when you order on Wednesday, 5 days later, on Wednesday, is the delivery day. If day_for_delivery = 8, then delivery actually happens only in 12 days on Monday. day_for_ordering = 1 and day_for_delivery = 1 indicates the same day delivery, if day_for_delivery = 8, the delivery happens only in a week, on Monday.

day_in_month

It determines the order day in a month. It could be combined with day_for_ordering. For example, day_for_ordering = 1 and day_in_month = 5, then order will happen on Monday closest to the 5th day in a month. If this column is filled (cf. it is not NULL), the columns day_for_delivery and even_odd are ignored. It cannot be combined with columns exception_*

minimal_period

For how many days the final product quantity (one batch) should lasts. It determines the quantity (batch size) for production if the batch sizes are not strictly given.

wp_production_delivery_date

Ordering calendar for production with the exact specification of ordering and delivery days in the format of a specific date

Column	Description
warehouse_id	Reference to warehouse identification
product_id	If product_id is 0, it is the default setting for a store and a supplier. If another value, then this setting applies for a particular product.
date_for_ordering	Order date ? today
date_for_delivery	Delivery date ? date_for_ordering

wp_production_delivery_exception

Complementary table for table WAREHOUSE_PRODUCT_PRODUCTION_DELIVERY. This table specifies the exceptions for the given time period. Instead of this table setting in GUI ""Periodical ordering settings - ordering attributes for suppliers"" can be used.

Column	Description
warehouse_id	Reference to warehouse identification
product_id	If product_id is 0, it is the default setting for a store and a supplier. If another value, then this setting applies for a particular product.
delivery_id	Sequence number of the record for the combination warehouse and product.
day_for_ordering	1 is Monday, 7 Sunday; If order day and delivery day are NULL then exception functions as dont order.
day_for_delivery	1 or 8 is MONDAY, 2 or 9 is TUESDAY... 7 or 14 is SUNDAY. This is the closest possible delivery day with respect to the order day (day_for_ordering). For example, day_for_ordering = 3 and day_for_delivery = 1 means that when you order on Wednesday, 5 days later, on Wednesday, is the delivery day. If day_for_delivery = 8, then delivery actually happens only in 12 days on Monday. day_for_ordering = 1 and day_for_delivery = 1 indicates the same day delivery, if day_for_delivery = 8, the delivery happens only in a week, on Monday.
minimal_period	For how many days the final product quantity (one batch) should lasts. It determines the quantity (batch size) for production if the batch sizes are not strictly given.
exception_date_from	If the date format is 0000-00-00, it is a usual setting for order days. If filled, then this is an exception that - in the given period (exception_date_from to exception_date_to) - fully replaces the delivery schedule.
exception_date_to	If the date format is 0000-00-00, it is a usual setting for order days. If filled, then this is an exception that - in the given period (exception_date_from to exception_date_to) - fully replaces the delivery schedule.
exception_type	If OD, then all the order days in a given period are replaced; if DD, all delivery days in the given period are replaced.

PROMOTION

List of promo actions. This table links promotions within promotion type/group.

Column	Description
promotion_id	Promotion action identification, as provided by the client
shortcut	Promotion abbreviation - unique identifier used for grouping products within one promotion.

name	Promotion name.
promotion_type_id	Refernce to promotion type
promo_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

PROMOTION_PRODUCT_WAREHOUSE

Details of the promo action for the SKU. For most of the usual cases, this table in combination with table promotion is sufficient.

Column	Description
promotion_id	Reference to promotion identification
product_id	Reference to product identification
warehouse_id	Reference to warehouse identification
delivery_date_from	Delivery of goods for promotion start date.
delivery_date_to	Delivery of goods for promotion end date.
promotion_date_from	Promotion start date.
promotion_date_to	Promotion end date.
base_price	Selling price per unit, not discounted. For the purpose of promotion forecast, the price doesn't need to be real, the percentage of discount is calculated from the ration between base_price and discounted_price.
distcounted_price	Selling price per unit, discounted. For the purpose of promotion forecast, the price doesn't need to be real, the percentage of discount is calculated from the ration between base_price and discounted_price.
base_purchase_price	Non-promotion purchase price per unit.
discounted_purchase_price	Promotion purchase price per unit.
estimated_amount	Expected volume - must be filled for all records with the same promotion_product_id, or none.
maximal_amount	Maximal volume - must be filled for all records with the same promotion_product_id, or none.

compute_promo_history	If you turn this off, you will disable cleaning of the promo sales from the history.
compute_promo_prediction	If you turn this off, you will disable prediction for promotion. Pre-order (special_preorder*) and speculative purchase (loading*) will still be suggested for the promotion if they are enabled in the columns intended for this.
special_preorder	1 = redistribution for promotion will take place, reflecting settings in the following special_preorder_* columns
special_preorder_date_of_order	Mandatory if special_preorder = 1. If NULL, delivery day will be set according to special_preorder_date_of_delivery.
special_preorder_date_of_delivery	Mandatory if special_preorder = 1. If NULL, considered as last delivery before promotion.
special_preorder_days_to_cover	Mandatory if special_preorder = 1. Number of days covered by a pre-order. If NULL, interpreted as 1 period before delivery, by minimum the nearest delivery_date.
special_preorder_minimal_quantity	Minimal quantity that needs to be ordered in a pre-order for a promotion. If NULL, considered as 1 package.
special_preorder_delivery_variant	BEFORE - within the last possible delivery before special_preorder_date_of_delivery, EXACT - delivery on the date of special_preorder_date_of_delivery, AFTER - delivery will be suggested on the first delivery after the special_preorder_date_of_delivery. When using this settings, the special_preorder_date_of_order date must be set.
compute_date_from	Transactions within this time period are considered as part of promotion. They will be taken into account in promotion analysis and forecast. For retail, period of promotion, for wholesale delivering period for promotion should be entered.
compute_date_to	Transactions within this time period are considered as part of promotion. They will be taken into account in promotion analysis and forecast. For retail, period of promotion, for wholesale delivering period for promotion should be entered.
loading_max_days_to_cover	Maximal number of day that can be covered by a speculative purchase. 0 or NULL = do not suggest speculative purchase.
loading_min_days_before_end	Minimal number of days before end of delivery period for promotion, when speculative purchase can be delivered.
safetystock_before_promotion_days	Number of days before the start of promotion when safety stock should be increased.
safetystock_before_promotion_percent	Safety stock increase before the start of promotion will be X per cent from the difference between promotion and non-promotion safety stock.

estimated_increment	A value that corresponds to the effects of promotion. It is the value that will be sold as a result of this promotion. The difference between estimated_amount and estimated_baseline. The sum of estimated_increment + estimated_baseline must come out the same value as in estimated_amount. Example: sales in the promotion (estimated_amount) is 500. Sales without promotion (estimated_baseline) is 100. Estimated_increment is 400
estimated_baseline	Quantity sold without promotional effects. See example in estimated_increment
estimated_error	This is a percentage value that is used as a safety stock for graduation. The do is stored in result_amount_error_one_sided_computed to be used for calculating the safetystock
estimated_post_promo_length	Post-promo is the period after a promotion in which sales are reduced due to the post-promo effect. For historical promotions, the baseline is lowered during this period . For future promotions, the forecast is lowered during this period. We set here how many days after graduation the postpromo effect should operate
safetystock_up_limiter	Maximal ratio of safety stock in relation to forecast increment, or NULL if global configuration should be used.
safetystock_down_limiter	Minimal ratio of safety stock in relation to forecast increment, or NULL if global configuration should be used.
estimated_post_promo_amount	A negative value that indicates how much the sales will decrease in the post-promo period. Can be a set negative quantity or a negative percentage value

PROMOTION_TYPE

List of promotion types.	
Column	Description
promotion_type_id	Identification of ? promotion type
shortcut	Promotion type abbreviation.
name	Promotion type name.
promotion_variant	Unused.
promo_type_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

PROMOTION_PRODUCT_WAREHOUSE_ATTRIBUTE

Additional boolean (on/off) attributes to be added to the promotion for SKU (atributes are defined in PROMOTION_ATTRIBUTE table).

Column	Description
promotion_id	Reference to promotion identification
product_id	If 0 or empty, applies for all products in promotion.
warehouse_id	If 0 or empty, applies for all warehouses in promotion.
promotion_attribute_id	Reference to identification of promotion attribute
value	Define if this attribute will apply to this promotion for the SKU

PROMOTION_ATTRIBUTE

List of promotion attributes. A custom set of attributes can be added to a promotion using PROMOTION_PRODUCT_WAREHOUSE_ATTRIBUTE table. Besides custom attributes, it is possible to use several pre-defined (attribute name as listed): - leaflet - secondary_placement - display_stand - media_plan - hostess - leaflet_title

Column	Description
promotion_attribute_id	Promotion attribute identification
name	Promotion attribute name, e.g. leaflet, secondary_placement, display_stand, media_plan, hostess, leaflet_title.
promo_attribute_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

WAREHOUSE_PRODUCT_CANNIBALIZATION_GROUP

Groups of products which are likely to be cannibalised within each other during promo action. Setting up a promotion for one product means expected decrease in sales of other products within the same group. If there is not a better way, It is usual to set these groups according to product tree categories. Each product can be in two groups at the same time, one using this table, second group might be defined in warehouse_product_cannibalization table.

Column	Description
warehouse_id	Reference to warehouse identification
product_id	Reference to product identification
cannibalization_group_id	Group identification.

VIEW

List of product views (different product-tree hierarchy used only for reporting). Not mandatory. If not set, views will be created reflecting category structure defined by category_type table.

Column	Description
view_id	Product view identification
name	Product view name
is_default	If no view set as default, the view over all category_type will be used as default. The default view will be transferred into STOCK only once or when cleanup = 1.

VIEW_DEFINITION

Product views (different product-tree hierarchy used only for reporting) definition. Each view is created by hierarchy of category_type, the hierarchy is defined by parent_category_type_id sequence.

Column	Description
view_id	Reference to view identification
category_type_id	Reference to category type identification
parent_category_type_id	NULL = level 0 category, category level -1 will be linked to the 0 category by the category_type_id of the 0 level, level -2 will be some of the category_type_id from the -1 level categories etc.

ABC_GROUP

List of groups for independent ABC calculation. ABC can be calculated independently for some groups, for example within shops or a shop sizes. See description of WAREHOUSE_PRODUCT_ABC_GROUP table.

Column	Description
abc_group_id	ABC group identification
name	Name of product group for which ABC analysis should be calculated independently.
abc_group_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

WAREHOUSE_PRODUCT_ABC_GROUP

It is possible to define separate groups of products for the purpose of separate ABC analysis. ABC can be calculated independently for some groups, for example within shops or a shop sizes. The groups are defined in ABC_GROUP table, this table links each SKU to a group. If this table and ABC_GROUP is left empty, products will not be divided into groups for the purpose of ABC analysis. Otherwise, every SKU must be linked to exactly one group.

Column	Description
warehouse_id	Reference to warehouse identification
product_id	Reference to product identification
abc_group_id	Reference to ABC group identification

ORDER_LIMIT

Definition of the limit groups. This table defines SKUs which are in one group together. The group can consist of SKUs from several warehouses, in case supplier delivers to multiple warehouses.

Column	Description
order_limit_group_id	Limit group identification

warehouse_id	Reference to warehouse identification
product_id	Reference to product identification
max_days_to_cover	To comply with ordering limits, suggestions of future orders can be moved to current day. The value specifies how many days forward should be considered for the group fulfilling. The value must be greater than 0. If you fill 30 it means that STOCK can look month ahead and move the orders from that period to current day.
custom_unit_ratio	Custom unit ratio

ORDER_LIMIT_GROUP

Definition of the limit for a group. The group is not dependant on supplier eg. one supplier can have several groups (in case supplier has several warehouses and each has different limit). Limit can be defined only by one of the attributes (either amount or value), if both are filled, the limit will be calculated for amount. It is possible to have amount_min and value_max and vice versa.

Column	Description
order_limit_group_id	Identification of order limit group
supplier_id	Reference to the supplier of the group
amount_min	The minimal ordered quantity for the whole group. It must be greater than 0 or NULL. Preferred value to value_min if both are filled.
value_min	The minimal value of the order for the whole group. It must be greater than 0 or NULL.
amount_max	The ordered quantity for the whole group must match as closely as possible this number (but can not be bigger) or it's multiple (in case the original order is already over the amount_max). It must be greater than 0 or NULL. Preferred value to value_max if both are filled.
value_max	The ordered value for the whole group must match as closely as possible this number (but can not be bigger) or it's multiple (in case the original order is already over the value_max). It must be greater than 0 or NULL.
ratio_max	If amount_max or value_max are set, then this parameter must ≥ 0 and < 1 (otherwise NULL). It determines percentually minimal sufficient quantity, if amount_max or value_max are defined. The algorithm tries to fit the order into the range delimited by this minimal level and defined maximum. For example, ratio_max is 0.75 and amount_max is 100, then the algorithm tries to fit the resulting order within the range 75 - 100 mu, 175 - 200 mu, 275 - 300 mu, etc.

period_to_cover	To fit the limitation rules, suggested orders get shifted from the future to current day. Order_limit.max_day_to_cover defines the max lenght of the period. This parameter defines how many days are added within each iteration. If set to 1 it adds day by day. If set to eg. 7 it adds weeks, in a week ahead there will be more orders, so the alorythm can choose the size of the orders which he really needs to fill the limit as precisly as possible (I need to add 5 pcs to fill the amount_min, and tomorrow there is order for 100 pcs, but day after tomorrow there is order for 6 pcs). The parameter must be greater than 0.
custom_unit	Name of limit unit. For description, see ORDER_LIMIT.custom_unit_ratio. If NULL, STOCK supposes that all products in this group share the same basic measurement unit. E.g. if all products are measured in kgs, then custom_unit = NULL means that the limit is also set in kgs.
client_order_limit_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

CATEGORY_MANAGEMENT_GROUP

List of groups used in Category Management. STOCK can fit group of product into shop/warehouse limits. It Shortens the period or omits some products from the group. Individual products are linked to a group using a record in WAREHOUSE_PRODUCT.category_management_group_id. For more info, see CATEGORY_MANAGEMENT_GROUP_WAREHOUSE

Column	Description
category_management_group_id	Group for category management identification
name	Group name
category_management_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

CATEGORY_MANAGEMENT_GROUP_WAREHOUSE

Definition of the rules for Category Management groups. The table defines maximum/minimum types of products to be hold at a warehouse/shop within one group/category. In case that a warehouse holds more products than set for a group, the STOCK system excludes some products automatically by not ordering them anymore. In case that a warehouse/shop holds less products than set for a group, the STOCK system suggests including some products that report sales on another warehouse/shop.

Column	Description
--------	-------------

category_management_group_id	Reference to category management group
warehouse_id	Reference to warehouse identification
min_product_count	Minimal number of various products within a category kept in a warehouse. If the number is not reached, it is supplemented by other products (from the same Category Management group) based on their popularity in sale from other warehouses.
max_product_count	Maximal number of various products within a category kept in a warehouse.
recompute_max_product_count	1 = the maximal number of various products within a category at a warehouse will be automatically computed based on 3 month history, in this case, no need to supply max_product_count.
max_amount	Maximal amount of various products within a category kept in a warehouse.
max_value	Maximal value of various products within a category kept in a warehouse.

CURRENCY_RATE

Exchange rate for used currencies. Related to the default currency set using currency_id in the CONFIGURATION table. The currency value (rate) is defined in default currency. E.g, if the default rate is CZK, then the rate value for EUR will be aproximatelly 25.

Column	Description
target_currency_id	CZK,USD,EUR,...
date	The date on which the exchange rate is valid. Cannot be NULL.
rate	Value of the currency expressed in the default currency. Cf., if the default is CZK, then for EUR it is circa 25.

EXTREME_SALE_TYPE

List of extreme sales types. A sale transaction can be labeled as extreme for different reasons. This table lists the reasons. The reason has no influence over forecasting calculations.

Column	Description
extreme_sale_type_id	Identification of extreme sale (outlier) type
name	Type of extreme sale.
shortcut	Abbreviation of the extreme sale type.
is_expiration	1 = extreme sale for a lowered price due to approaching expiration date

ATTRIBUTE

Attributes can be displayed grafically in GUI as a icon.

Column	Description
attribute_id	Identification of the attribute
shortcut	If there is no defined icon, this text appears. If there is an icon, then the text is ignored. If icon_on=0 and there is no defined icon at the same time, the text appears in grey, otherwise it appears in bold and in expected color.
name	After hoovering over the shortcut, this appears as intro to the value from WAREHOUSE_PRODUCT_ATTRIBUTE.value
icon_on	It lists the attribute dial, it has nothing to do with toggling, but stores the icon dial. Example: the dont order icon here will only hold a number of which image to use when dont_order is on (not defining when it should be on, but what icon should be displayed when the attribute is active)
icon_off	It lists the attribute dial, it has nothing to do with toggling, but stores the icon dial. Example: the dont order icon here will only hold a number of which image to use when dont_order is on (not defining when it should be on, but what icon should be displayed when the attribute is active)

WAREHOUSE_PRODUCT_ATTRIBUTE

Collection of attributes assigned to a SKU. Attributes assigned to products in this table serve for information only and do not enter any calculation. They are used for explanation of actions performed upon a product or as complementary values. In the STOCK GUI, attributes are displayed in product detail section, can take form of an icon. Examples: ""Mandatory item"", ""Optional item"", ""Do not order due to low demand"".

Column	Description
warehouse_id	Reference to warehouse identification
product_id	Reference to product identification
attribute_id	Reference to attribute identification
value	A value that the user is going to see after hovering over an icon or a shortcut. It could be in HTML format.
icon_on	Each attribute has 2 states: 1 = on, 0 = off. If there is an icon to the attribute, the icon appears in accordance with the attribute's state. If there is no icon, the ATTRIBUTE.shortcut appears in grey if 0, in standard color if 1.

PRODUCT_MEASURE_UNIT

Conversion rates to base measure unit. This table does not affect any calculations in a core. Can be used for custom features.

Column	Description
product_id	Reference to product identification
measure_unit	measurement unit code
conversion_to_base_measure_unit	If base_measure_unit = KG and measure_unit in this table is G, then conversion_to_base_measure_unit is 1000.
rounding_after_conversion	Rounding applied after conversion from/to this measure unit.

PRODUCT_EAN

Product EAN groups. If one or more EANs are to be recorded for a product, this table keeps the record.

Column	Description
product_id	Reference to product identification
ean	EAN Code
primary_ean	1 = indication of the primary EAN code, only one within the product_id can be primary

HOLIDAYS

Holidays settings. Settings of holidays, defining whether ordering/delivery is available on certain date or not. Does not have any effect on forecast.

Column	Description
date	Date of holiday
type	Holiday 0=Saturday, 1=Sunday, 3=National Holiday, 3=Easter, 4=Christmas, 6=Other. May be modified, depending on the project.
is_order	Is it possible to order on this day?
is_delivery	Is it possible to deliver on this day?

CONFIGURATION_PW

Table enabling insertion of any value into STOCK configurations tables.

Column	Description
table	Name of a configuration table in STOCK database, ending *_configuration (eg.: forecast_configuration)
key	Name of a value in the configuration table (eg.: history-days)

value

Value to be inserted into the configuration table (eg.: 28)

VERSION_DB

Contains information on data map version. Not critical for data import per se, only maintains correct functioning of the data map. Filled automatically using UDM updater.

Column	Description
version	Migration script (diff file) version
user_name	Author of a migration script (diff file)
created	Date of migration script (diff file)
updated	Time the migration script (diff file) was executed
last_cmd	The last executed SQL query - important, if the script was terminated by an error, it is good to know whether the SQL query that caused the error got saved
state	UDM version, with each difference a new version is created.
error_msg	Description of the error during the script execution

WAREHOUSE_PRODUCT_STOCKOUT_LEVEL

Definition of stockout level for SKU.

Column	Description
warehouse_id	For which warehouse_id the setting should be used
product_id	Reference to product identification
year	Year
month	Month
stockout_level	Stockout level

WAREHOUSE_PRODUCT_CANNIBALIZATION

Table of warehouse product cannibalization pairs

Column	Description
product_id	Reference to the identification of the cannibalising product
warehouse_id	Link to identify the cannibalizing warehouse
cannibalized_product_id	Reference to cannibalized product
cannibalized_warehouse_id	Reference to cannibalized warehouse
date_from	Cannibalization period from
date_to	Cannibalization period to
cannibalization_ratio	Ratio for calculation of cannibalization from baseline. Either ratio or amount should be used
amount	Cannibalized amount, absolute. Either ratio or amount should be used.

PROMOTION_PRODUCT_WAREHOUSE_DAY

Externally provided promotion prediction data

Column	Description
promotion_id	Reference to promotion identification
product_id	Reference to product identification
warehouse_id	Reference to warehouse identification
date	Date on which the change will be applied
total	Expected volume

increment	A value that corresponds to the effects of promotion. It is the value that will be sold as a result of this promotion. The difference between estimated_amount and estimated_baseline. The sum of estimated_increment + estimated_baseline must come out the same value as in estimated_amount. Example: sales in the promotion (estimated_amount) is 500. Sales without promotion (estimated_baseline) is 100. Estimated_increment is 400.
baseline	Quantity sold without promotional effects. See example in estimated_increment

WAREHOUSE_PRODUCT_CONFIGURATION_PW

Table for configuring settings per product/warehouse item (SKU). Keys are defined in the WAREHOUSE_PRODUCT_CONFIGURATION enumeration.	
Column	Description
warehouse_id	Identification of the referenced warehouse
product_id	Identification of the referenced product
table	Name of a the configuration table
key	Name of a value in the configuration table
value	Value to be inserted into the configuration table (eg.: 28)

WAREHOUSE_PRODUCT_PARTIAL_EXPIRATION

Table for specification of stock production batch expiration (end of shelf life). Expiration date set to past is treated as today. The consumption is calculated as in FIFO mode, oldest stock would be sold first.	
Column	Description
warehouse_id	Warehouse ID
product_id	Product ID
partial_expiration_id	Production batch ID
name	Production batch name

expiration_date

Shelf life expiration

amount

Quantity in this batch

WAREHOUSE_PRODUCT_SHELF_CAPACITY

Table for specification of promo/non shelves capacity. It allows precisely control minimal supply level. Should replace missuse of safetystock_min or safetystock_min_increase settings.

Column	Description
warehouse_id	Warehouse ID
product_id	Product ID
shelf_id	(Promo) Shelf id
name	Shelf name
date_from	Shelf life validity from
date_to	Shelf life validity to
width	Shelf product width (amount of products in shelf, that can fit next to each other)
height	Shelf product height (amount of products in shelf, that can fit on each other)
depth	Shelf product depth (amount of products in shelf, that can fit behind each other)
lower_capacity_days_before_end	Days before validity end when starts lowering shelf capacity
increase_safetystock	Should add shelf capacity to safetyStock or calculate diff with MAX(safetyStock, shelfCapacity)
promotion_id	Promotion ID

WAREHOUSE_PRODUCT_SUPPLY_SOURCE_INTERVAL

Time-valid replenishment source for a warehouse product. The source can be an external supplier, another warehouse/DC, or production. This table is the date-aware routing model; detailed ordering parameters remain in the source-specific supplier, redistribution, or production configuration. Validity is matched against the order date of each replenishment order: an order created within the interval will use that source and its ordering parameters.

Column	Description
warehouse_id	Target warehouse identifier.
product_id	Product identifier.
source_type	Type of replenishment source. SUPPLIER maps to source_type_id 3, WAREHOUSE to source_type_id 2, and PRODUCTION to source_type_id 1.
source_id	Supplier identifier for SUPPLIER rows or source warehouse identifier for WAREHOUSE rows. Must be empty for PRODUCTION rows.
variant	Variant of the matching WAREHOUSE_PRODUCT_SUPPLIER or WAREHOUSE_PRODUCT_REDISTRIBUTION row. Use 1 for PRODUCTION.
valid_from	First order date from which this source becomes active.
valid_to	Last order date for which this source is active.
enabled	Whether this source interval is active.
primary_source	Marks the primary routing source used by date-aware replenishment planning.

tune_holiday

List of holidays to which various rules can be linked

Column	Description
--------	-------------

holiday_id	ID of the holiday
name	Name of the holiday
date_from	Start date of the holiday
date_to	End date of the holiday
affected_days_before	How many days before the holiday start date will be affected by the rules
affected_days_after	How many days after the holiday end date will be affected by the rules

tune_holiday_rule

List of rules for holidays	
Column	Description
rule_id	ID of the rule
rule_type	Type of the rule
holiday_id	ID of the holiday
name	Name of the rule
priority	Priority of the rule
data_version	Version of the JSON data
data	Rule-specific data in JSON format (content depends on rule type and data version)

BALANCING_GROUP

List of balancing groups	
Column	Description

group_id	Balancing group identification (provided by the client)
name	Balancing group name
group_id_in_pw	Unique ID that will be used in STOCK. If NULL, id will be generated in STOCK during import. Must be filled in all rows of the table, or in none. (Suitable for Clustered Solution)

BALANCING_GROUP_SETTINGS

Balancing groups settings by for each day of week	
Column	Description
group_id	Balancing group identification (provided by the client)
day_of_week	Day of week
minimum_lines	Minimum lines of orders
maximum_lines	Maximum lines of orders

BALANCING_GROUP_WAREHOUSE_PRODUCT

Assignment of SKUs to balancing groups	
Column	Description
warehouse_id	Warehouse identification (provided by the client)
product_id	Product identification (provided by the client)
group_id	Balancing group identification (provided by the client)
is_ordering_mandatory	Ordering of this product is mandatory (not affected by BALANCING_GROUP_SETTINGS.maximum_lines)
max_items_per_line	Max number of items on each line of order (e.g. if suggested amount is 5 and max_items_per_line is 1, then this order will occupy 5 lines of orders in balancing)

max_stock_duration

Max stock duration