

# EESys Stock Movements System

EESys includes a set of data-objects to manage the tracking of stock items with full traceability. Items may be purchased and moved several times prior to sale. In these movements they may be incorporated into new products, so a set of boxes are purchased, delivered to a factory and used to make a finished product. There are also occasions when some of the boxes might be unused, and could be returned to a store, or moved to a new location.

The systems built into EESys cope with these processes. This process is not simple, particularly once the conversion of "ingredient" products into finished-goods is added, together with the potential for part-consignments to be unused, moved or returned.

## How the data is marshalled

EESys includes a data-table "StockCounts" into which data is periodically added for a "Full Physical" check of the items that are **actually present** in different locations. This is done on a **date**, and records **location** and **StockPurchaseItem-ID (SPI-ID)** for every item.

After a Full Physical stock check, all computations in EESys are reset to start from the new date. In other words, all computations start by listing all data present in the full physical stock check and only add data from all other data-objects for **later** dates.

All movements and processes that involve stock are recorded in EESys, and can be queried for management analysis, but for day-to-day management of actual stocks all reporting and systems work from the "Last Full Physical" date.

### Data-Objects in EESys

The details of the contents of all these tables is included below. Here the basic purpose of each object is given.

1. **StockCounts.** A table which lists actual stock on shelves at a "DateDone". This generates the "StartingStock" a dataset of Location, Product, Quantity and Stock-ID/BatchCode.
2. **StockAdjustments.** A table which holds data on exceptional events that happen to stock. For example stock damaged in transit, requiring disposal, failing quality checks etc. This dataset includes a Date, Location, Product, Quantity and SPI-ID/BatchCode. This dataset is **deducted** from StockCount as part of the "CurrentStock" computation.
3. **StockPurchases.** A table which holds not only purchases of stock, but also stock-movements. Rather than have multiple tables, the data in the Stock Purchases table includes a **Source** and **Destination**, as well as Date, Product Quantity and Stock-ID/BatchCode. The Source and Destination are Locations. When "CurrentStock" is computed items are **added** to stock for the Destination, and **deducted** from stock at the Source.
4. **SalesInvoices.** A table which holds all sales data. Each Sales-Item includes a Source and SPI-ID. When CurrentStock is computed item are **deducted** from stock at the source.
5. **Production and IngredientsUsed.** These two tables store data about the production of semi-finished goods (which may be held as stock). The Production table includes a Date, Location, a Product (produced, which is assigned a unique Batch-Code) and Quantity. The IngredientsUsed table lists the materials used to produce the semi-finished product. Production data is **added** to "Current Stock", IngredientsUsed data is **deducted**.
6. **Packaging and PackagingUsed.** These two tables store data about the processing of semi-finished products into finished goods. Non-contact packaging items are part of the PackagingUsed, as well as the multiple of semi-finished items included in the finished good. Exactly as with Production, Packaging data **adds** to "CurrentStock", and PackagingUsed **deducts**.

## StockCounts, Adjustments, Purchases and Sales



		Locations				
		A	B	C	D	E
Products	1					
	2		X	Y		
	3			Z		
	4					
	5					

Product Location Computation

EESys must arrive at the stock-figures for "X, Y and Z" shown in the above image, and must enable "drilling down" so that staff can examine "X, Y or Z" and view the SPI-IDs and residual quantities for each SPI-ID. This computation is achieved by bringing together all the data in the tables detailed above.

For each SPI-ID a "part of X" Quantity is arrived at for each Location. Each of these items takes as its start date the date of the last physical stock count.

Part of X = StockCount

- StockAdjustment
- + StockPurchase.Dest
- StockPurchase.Source
- SalesInvoice.Source
- IngredientsUsed
- PackagingUsed.

"X" = the Sum of all "Part of X", ie the sum for every product of all SPI-IDs.

## Example SQL

Code will be added here as examples of how each part of the "Part of X" computation above is arrived at. Including these here will be a useful reference as it is regularly necessary to use just parts of the whole computation.