

InventTransStaging(31 calculation)

Tuesday, May 6, 2025

11:25 AM

```
let
    Source = Sql.Databases("amgazsql.database.windows.net"),
    #"d365BYOD-Prod" = Source{[Name="d365BYOD-Prod"]}[Data],
    dbo_AMG_InventTransStaging = #"d365BYOD-Prod"{[Schema="dbo",Item="AMG_InventTransStaging"]}[Data],
    PurchPurchaseOrderHeaderV2Staging = #"d365BYOD-Prod"{[Schema="dbo",Item="PurchPurchaseOrderHeaderV2Staging"]}[Data],
    #"Duplicated Column" = Table.DuplicateColumn(dbo_AMG_InventTransStaging, "DATAAREAID", "DATAAREAID - Copy"),
    #"Duplicated Column2" = Table.DuplicateColumn(#"Duplicated Column", "DATAAREAID - Copy", "DATAAREAID - Copy - Copy"),
    #"Duplicated Column1" = Table.DuplicateColumn(#"Duplicated Column2", "INVENTLOCATIONID", "INVENTLOCATIONID - Copy"),
    #"Duplicated Column7" = Table.DuplicateColumn(#"Duplicated Column1", "INVENTLOCATIONID - Copy", "INVENTLOCATIONID - Copy - Copy"),
    #"Duplicated Column3" = Table.DuplicateColumn(#"Duplicated Column7", "INVENTSTYLEID", "INVENTSTYLEID - Copy"),
    #"Duplicated Column4" = Table.DuplicateColumn(#"Duplicated Column3", "INVENTSIZEID", "INVENTSIZEID - Copy"),
    #"Duplicated Column5" = Table.DuplicateColumn(#"Duplicated Column4", "INVENTCOLORID", "INVENTCOLORID - Copy"),
    #"Duplicated Column6" = Table.DuplicateColumn(#"Duplicated Column5", "ITEMID", "ITEMID - Copy"),
    #"Merged Columns1" = Table.CombineColumns(#"Duplicated Column6",{ "DATAAREAID - Copy - Copy", "INVENTCOLORID - Copy", "INVENTSIZEID - Copy", "INVENTSTYLEID - Copy", "ITEMID - Copy"},Combiner.CombineTextByDelimiter("-", QuoteStyle.None),"InventBarcode Key"),
    #"Merged Columns" = Table.CombineColumns(#"Merged Columns1",{ "DATAAREAID - Copy", "INVENTLOCATIONID - Copy"},Combiner.CombineTextByDelimiter("-", QuoteStyle.None),"WarehouseKey"),
    #"Changed Type" = Table.TransformColumnTypes(#"Merged Columns",{{"DATEPHYSICAL", type date}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type",{{"INVENTLOCATIONID - Copy - Copy", "Invent Location ID"}}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Renamed Columns",{{"DATEEXPECTED", type date}, {"DATEFINANCIAL", type date}, {"DATEINVENT", type date}}),
    #"Added Custom" = Table.AddColumn(#"Changed Type1", "Cost", each [COSTAMOUNTSTD] * [QTY]),
    #"Changed Type2" = Table.TransformColumnTypes(#"Added Custom",{{"Cost", type number}}),
    PurchHeaderKey = Table.SelectColumns(PurchPurchaseOrderHeaderV2Staging, {"PURCHASEORDERNUMBER"}),
    DistinctKeys = Table.Distinct(PurchHeaderKey),
    #"Filter Out PO Return" = Table.SelectRows(dbo_AMG_InventTransStaging, each not ([REFERENCECATEGORY] = "Purchase order" and not List.Contains(DistinctKeys[PURCHASEORDERNUMBER], [REFERENCEID]))),
    #"Changed Type3" = Table.TransformColumnTypes(#"Filter Out PO Return",{{"DATECLOSED", type date}, {"DATEEXPECTED", type date}, {"DATEFINANCIAL", type date}, {"DATEINVENT", type date}, {"DATEPHYSICAL", type date}, {"DATESTATUS", type date}}),
    #"Added Conditional Column" = Table.AddColumn(#"Changed Type3", "Date Physical (fixedblank)", each if [DATEPHYSICAL] = #date(1900, 1, 1) then "" else [DATEPHYSICAL]),
    #"Added Conditional Column1" = Table.AddColumn(#"Added Conditional Column", "Date Financial (fixedblanks)", each if [DATEFINANCIAL] = #date(1900, 1, 1) then "" else [DATEFINANCIAL]),
    #"Changed Type4" = Table.TransformColumnTypes(#"Added Conditional Column1",{{"Date Financial (fixedblanks)", type date}, {"Date Physical (fixedblank)", type date}})
in
    #"Changed Type4"
```

```

-----
AppliedFilters = VAR MaxFilters = 3 RETURN IF (
    ISFILTERED ( AMG_InventTransStaging[REFERENCECATEGORY] ),
    VAR ___f = FILTERS ( AMG_InventTransStaging[REFERENCECATEGORY] )
    VAR ___r = COUNTROWS ( ___f )
    VAR ___t = TOPN ( MaxFilters, ___f, AMG_InventTransStaging[REFERENCECATEGORY] )
    VAR ___d = CONCATENATEX ( ___t, AMG_InventTransStaging[REFERENCECATEGORY], ", " )
    VAR ___x = "Category = " & ___d
    & IF(___r > MaxFilters, ", ... [" & ___r & " items selected]") & " "
    RETURN ___x & UNICHAR(13) & UNICHAR(10))
) & IF (
    ISFILTERED ( PurchPurchaseOrderHeaderV2Staging[PO Status] ),
    VAR ___f = FILTERS ( PurchPurchaseOrderHeaderV2Staging[PO Status] )
    VAR ___r = COUNTROWS ( ___f )
    VAR ___t = TOPN ( MaxFilters, ___f, PurchPurchaseOrderHeaderV2Staging[PO Status] )
    VAR ___d = CONCATENATEX ( ___t, PurchPurchaseOrderHeaderV2Staging[PO Status], ", " )
    VAR ___x = "PO Status = " & ___d
    & IF(___r > MaxFilters, ", ... [" & ___r & " items selected]") & " "
    RETURN ___x & UNICHAR(13) & UNICHAR(10))
    & IF (
        ISFILTERED ( EcoResReleasedProductV2Staging[Business Unit] ),
        VAR ___f = FILTERS ( EcoResReleasedProductV2Staging[Business Unit] )
        VAR ___r = COUNTROWS ( ___f )
        VAR ___t = TOPN ( MaxFilters, ___f, EcoResReleasedProductV2Staging[Business Unit] )
        VAR ___d = CONCATENATEX ( ___t, EcoResReleasedProductV2Staging[Business Unit], ", " )
        VAR ___x = "Business Unit = " & ___d
        & IF(___r > MaxFilters, ", ... [" & ___r & " items selected]") & " "
        RETURN ___x & UNICHAR(13) & UNICHAR(10))
)
=====

```

```

Cancel Amount =
AMG_InventTransStaging[Cancel Qty] *
CALCULATE(
    MAX(PurchPurchaseOrderLineV2Staging[PURCHASEPRICE]),
    FILTER(
        PurchPurchaseOrderLineV2Staging,
        PurchPurchaseOrderLineV2Staging[ITEMNUMBER] = AMG_InventTransStaging[ITEMID]
    )
)
=====

```

```

Cancel Qty =
(AMG_InventTransStaging[Original Ordered Qty] -
( AMG_InventTransStaging[Delivered Qty] + AMG_InventTransStaging[Remaining Qty])) -
AMG_InventTransStaging[GIT Qty]
=====

```

```

Creation Date =
SWITCH(
    TRUE(),
    AMG_InventTransStaging[REFERENCECATEGORY] = "Purchase order",
    LOOKUPVALUE(PurchPurchaseOrderHeaderV2Staging[PURCHASEORDERCREATIONDATE],
        PurchPurchaseOrderHeaderV2Staging[PURCHASEORDERNUMBER],
        LOOKUPVALUE(PurchPurchaseOrderLineV2Staging[PURCHASEORDERNUMBER],
            PurchPurchaseOrderLineV2Staging[INVENTORYLOTID], AMG_InventTransStaging[INVENTTRANSID])),
    AMG_InventTransStaging[REFERENCECATEGORY] = "Consignment replenishment orde",
    LOOKUPVALUE(ConsignmentReplenishmentOrderHeaderStaging[CONSIGNMENTHEADERCREATEDDATETIME],
        ConsignmentReplenishmentOrderHeaderStaging[CONSIGNMENTREPLENISHMENTORDERNUMBER] ,
        LOOKUPVALUE(ConsignmentReplenishmentOrderLineStaging[CONSIGNMENTREPLENISHMENTORDERNUMBER],
            ConsignmentReplenishmentOrderLineStaging[INVENTTRANSID] , AMG_InventTransStaging[INVENTTRANSID]))
)
=====

```

```

Date Expected =
IF(AMG_InventTransStaging[DATEEXPECTED] = DATE(1900,1,1),
    BLANK(),
    AMG_InventTransStaging[DATEEXPECTED]
)

```

```

=====

Date Received =
VAR CalculatedDate =
    IF(
        AMG_InventTransStaging[REFERENCECATEGORY] = "Purchase Order" &&
        AMG_InventTransStaging[DisplayLOTid] <> AMG_InventTransStaging[InventTransID_of_RecID],
        COALESCE(
            CALCULATE(
                MIN(AMG_InventTransStaging[DATEPHYSICAL]),
                FILTER(
                    AMG_InventTransStaging,
                    AMG_InventTransStaging[OriginalLotID] = EARLIER(AMG_InventTransStaging[DisplayLOTid]) &&
                    AMG_InventTransStaging[MARKINGREFINVENTTRANSORIGIN] = 0
                )
            ),
            AMG_InventTransStaging[DATEPHYSICAL]
        ),
        AMG_InventTransStaging[DATEPHYSICAL]
    )
RETURN
IF(CalculatedDate = DATE(1900,1,1), BLANK(), CalculatedDate)
=====

```

```

Delivered Amount =
AMG_InventTransStaging[Delivered Qty] *
CALCULATE(
    MAX(PurchPurchaseOrderLineV2Staging[PURCHASEPRICE]),
    FILTER(
        PurchPurchaseOrderLineV2Staging,
        PurchPurchaseOrderLineV2Staging[ITEMNUMBER] = AMG_InventTransStaging[ITEMID]
    )
)
=====

```

```

Delivered Qty =
AMG_InventTransStaging[Original Ordered Qty] -(AMG_InventTransStaging[Remaining Qty] +
AMG_InventTransStaging[GIT Qty])
=====

```

```

DisplayLOTid =
IF(AMG_InventTransStaging[REFERENCECATEGORY] = "Goods in transit",
AMG_InventTransStaging[OriginalLotID], AMG_InventTransStaging[INVENTTRANSID])
=====

```

```

Expected Cost Price =
IF (AMG_InventTransStaging[Posted Cost Amount] > 0, AMG_InventTransStaging[Posted Cost Amount],
    RELATED(SkuCostVw[Price])
)
=====

```

```

Extended STD Cost =
IF(
    AMG_InventTransStaging[COSTAMOUNTSTD] = 0,
    AMG_InventTransStaging[QTY] * AMG_InventTransStaging[Expected Cost Price],
    AMG_InventTransStaging[COSTAMOUNTSTD]
)
=====

```

```

GIT Amount =
AMG_InventTransStaging[GIT Qty] *
CALCULATE(
    MAX(PurchPurchaseOrderLineV2Staging[PURCHASEPRICE]),
    FILTER(
        PurchPurchaseOrderLineV2Staging
    )
)

```

```

PurchPurchaseOrderLineV2Staging,
PurchPurchaseOrderLineV2Staging[ITEMNUMBER] = AMG_InventTransStaging[ITEMID]
)
)
=====

GIT Qty =
CALCULATE(
    SUM('AMG_InventTransStaging'[QTY]),
    FILTER(
        AMG_InventTransStaging,
        AMG_InventTransStaging[INVENTTRANSID] = EARLIER(AMG_InventTransStaging[InventTransID_of_RecID]) &&
        AMG_InventTransStaging[STATUSISSUE] = "Reserved Physical"
    )
)*-1
=====

GITLotID =
LOOKUPVALUE(AMG_InventTransStaging[InventTransID_of_RecID], AMG_InventTransStaging[INVENTTRANSID],
AMG_InventTransStaging[InventTransID_of_RecID], AMG_InventTransStaging[REFERENCECATEGORY] , "Purchase Order")
=====

Invent Key =
AMG_InventTransStaging[DATAAREAID] & "-" & AMG_InventTransStaging[INVENTCOLORID] & "-" &
AMG_InventTransStaging[INVENTSIZEID] & "-" & AMG_InventTransStaging[INVENTSTYLEID] & "-" &
AMG_InventTransStaging[ITEMID]
=====

InventTrans Purchase Amount =
AMG_InventTransStaging[QTY] * AMG_InventTransStaging[Purchase Price]
=====

InventTransID_of_RecID =
    LOOKUPVALUE( AMG_InventTransOriginStaging[INVENTTRANSID], AMG_InventTransOriginStaging[RECID],
AMG_InventTransStaging[MARKINGREFINVENTTRANSORIGIN])
=====

Invoice Vendor =
IF(AMG_InventTransStaging[REFERENCECATEGORY] = "Purchase order",
LOOKUPVALUE(PurchPurchaseOrderHeaderV2Staging[INVOICEVENDORACCOUNTNUMBER],
PurchPurchaseOrderHeaderV2Staging[PURCHASEORDERNUMBER],
LOOKUPVALUE(PurchPurchaseOrderLineV2Staging[PURCHASEORDERNUMBER],
PurchPurchaseOrderLineV2Staging[INVENTORYLOTID], AMG_InventTransStaging[INVENTTRANSID])),
LOOKUPVALUE(ConsignmentReplenishmentOrderHeaderStaging[CONSIGNMENTVENDORACCOUNTNUMBER],
ConsignmentReplenishmentOrderHeaderStaging[CONSIGNMENTREPLENISHMENTORDERNUMBER] ,
LOOKUPVALUE(ConsignmentReplenishmentOrderLineStaging[CONSIGNMENTREPLENISHMENTORDERNUMBER],
ConsignmentReplenishmentOrderLineStaging[INVENTTRANSID] , AMG_InventTransStaging[INVENTTRANSID]))
=====

Line Number =
SWITCH(
    TRUE(),
    AMG_InventTransStaging[REFERENCECATEGORY] = "Purchase order",
        LOOKUPVALUE(
            PurchPurchaseOrderLineV2Staging[LINENUMBER],
            PurchPurchaseOrderLineV2Staging[INVENTORYLOTID], AMG_InventTransStaging[INVENTTRANSID]
        ),
    AMG_InventTransStaging[REFERENCECATEGORY] = "Consignment replenishment orde",
        LOOKUPVALUE(
            ConsignmentReplenishmentOrderLineStaging[LINENUMBER],
            ConsignmentReplenishmentOrderLineStaging[INVENTTRANSID], AMG_InventTransStaging[INVENTTRANSID]
        ),
    BLANK()
)
=====

```

```

-----_:_-----
    LOOKUPVALUE( AMG_InventTransOriginStaging[INVENTTRANSID],
AMG_InventTransOriginStaging[RECID],
AMG_InventTransStaging[MARKINGREFINVENTTRANSORIGIN])
=====

Invoice Vendor =
IF(AMG_InventTransStaging[REFERENCECATEGORY] = "Purchase order",
LOOKUPVALUE(PurchPurchaseOrderHeaderV2Staging[INVOICEVENDORACCOUNTNUMBER],
PurchPurchaseOrderHeaderV2Staging[PURCHASEORDERNUMBER],
LOOKUPVALUE(PurchPurchaseOrderLineV2Staging[PURCHASEORDERNUMBER],
PurchPurchaseOrderLineV2Staging[INVENTORYLOTID],
AMG_InventTransStaging[INVENTTRANSID])),
LOOKUPVALUE(ConsignmentReplenishmentOrderHeaderStaging[CONSIGNMENTVENDORACCOUNTNUMBER],
ConsignmentReplenishmentOrderHeaderStaging[CONSIGNMENTREPLENISHMENTORDERNUMBER] ,
LOOKUPVALUE(ConsignmentReplenishmentOrderLineStaging[CONSIGNMENTREPLENISHMENTORDERNUMBE
R], ConsignmentReplenishmentOrderLineStaging[INVENTTRANSID] ,
AMG_InventTransStaging[INVENTTRANSID]))))
=====

Line Number =
SWITCH(
    TRUE(),
    AMG_InventTransStaging[REFERENCECATEGORY] = "Purchase order",
        LOOKUPVALUE(
            PurchPurchaseOrderLineV2Staging[LINENUMBER],
            PurchPurchaseOrderLineV2Staging[INVENTORYLOTID],
            AMG_InventTransStaging[INVENTTRANSID]
        ),
    AMG_InventTransStaging[REFERENCECATEGORY] = "Consignment replenishment orde",
        LOOKUPVALUE(
            ConsignmentReplenishmentOrderLineStaging[LINENUMBER],
            ConsignmentReplenishmentOrderLineStaging[INVENTTRANSID],
            AMG_InventTransStaging[INVENTTRANSID]
        ),
    BLANK()
)
=====

Number of Lines in InventTrans =
CALCULATE(
    COUNTROWS(AMG_InventTransStaging),
    ALLEXCEPT(AMG_InventTransStaging,AMG_InventTransStaging[INVENTTRANSID])
)
=====

Order Number =
SWITCH(
    TRUE(),
    AMG_InventTransStaging[REFERENCECATEGORY] = "Purchase order",
        LOOKUPVALUE(
            PurchPurchaseOrderLineV2Staging[PURCHASEORDERNUMBER],
            PurchPurchaseOrderLineV2Staging[INVENTORYLOTID],
            AMG_InventTransStaging[INVENTTRANSID]
        ),
    AMG_InventTransStaging[REFERENCECATEGORY] = "Consignment replenishment orde",
        LOOKUPVALUE(
            ConsignmentReplenishmentOrderLineStaging[CONSIGNMENTREPLENISHMENTORDERNUMBER
],
            ConsignmentReplenishmentOrderLineStaging[INVENTTRANSID],
            AMG_InventTransStaging[INVENTTRANSID]
        ),
    BLANK()
)
=====

Order Vendor =
IF(AMG_InventTransStaging[REFERENCECATEGORY] = "Purchase order",

```

```

LOOKUPVALUE(PurchPurchaseOrderHeaderV2Staging[ORDERVENDORACCOUNTNUMBER],
PurchPurchaseOrderHeaderV2Staging[PURCHASEORDERNUMBER],
LOOKUPVALUE(PurchPurchaseOrderLineV2Staging[PURCHASEORDERNUMBER],
PurchPurchaseOrderLineV2Staging[INVENTORYLOTID],
AMG_InventTransStaging[INVENTTRANSID])),
LOOKUPVALUE(ConsignmentReplenishmentOrderHeaderStaging[CONSIGNMENTVENDORACCOUNTNUMBER],
ConsignmentReplenishmentOrderHeaderStaging[CONSIGNMENTREPLENISHMENTORDERNUMBER] ,
LOOKUPVALUE(ConsignmentReplenishmentOrderLineStaging[CONSIGNMENTREPLENISHMENTORDERNUMBE
R], ConsignmentReplenishmentOrderLineStaging[INVENTTRANSID] ,
AMG_InventTransStaging[INVENTTRANSID]))))
=====

```

```

Original Ordered Amount =
AMG_InventTransStaging[Original Ordered Qty] *
CALCULATE(
    MAX(PurchPurchaseOrderLineV2Staging[PURCHASEPRICE]),
    FILTER(
        PurchPurchaseOrderLineV2Staging,
        PurchPurchaseOrderLineV2Staging[ITEMNUMBER] = AMG_InventTransStaging[ITEMID]
    )
)
=====

```

```

Original Ordered Qty =
SWITCH(
    TRUE(),
    AMG_InventTransStaging[REFERENCECATEGORY] = "Purchase order",
        LOOKUPVALUE(
            PurchPurchaseOrderLineV2Staging[ORDEREDPURCHASEQUANTITY],
            PurchPurchaseOrderLineV2Staging[INVENTORYLOTID],
            AMG_InventTransStaging[INVENTTRANSID]
        )
    /
    AMG_InventTransStaging[Number of Lines in InventTrans],
    AMG_InventTransStaging[REFERENCECATEGORY] = "Consignment replenishment orde",
        LOOKUPVALUE(
            ConsignmentReplenishmentOrderLineStaging[CONSIGNMENTREPLENISHMENTQUANTITY],
            ConsignmentReplenishmentOrderLineStaging[INVENTTRANSID],
            AMG_InventTransStaging[INVENTTRANSID]
        )
    /
    AMG_InventTransStaging[Number of Lines in InventTrans],
    BLANK()
)
=====

```

```

OriginalLotID =
LOOKUPVALUE(AMG_InventTransStaging[InventTransID_of_RecID],
AMG_InventTransStaging[GITLotID], AMG_InventTransStaging[INVENTTRANSID])
=====

```

```

Posted Cost Amount =
AMG_InventTransStaging[COSTAMOUNTSTD] / AMG_InventTransStaging[QTY]
=====

```

```

Purchase Price =
SWITCH (
    TRUE(),
    -- Case 1: Purchase Order → from PO Line Table
    AMG_InventTransStaging[REFERENCECATEGORY] = "Purchase order",
        LOOKUPVALUE (
            PurchPurchaseOrderLineV2Staging[PURCHASEPRICE],
            PurchPurchaseOrderLineV2Staging[INVENTORYLOTID],
            AMG_InventTransStaging[INVENTTRANSID]
        ),
        LOOKUPVALUE (
            EcoResReleasedProductV2Staging[PURCHASEPRICE],

```