```
/* Freek Keijzer, myBrand, 17.09.2020
Based on standard SAP view P_MaterialStockTimeSeries1.
The standard view already had a union to combine
- "Stock Level before start of first time period" (Source = 'B') with
- "Stock Change from start of first time period" (Source = 'A').
Custom enhancements:
- “Stock Change within a time period” (Source = 'C')
- “Stock Change within a "rolling year" till end of the time period” (Source = 'D')
Source (A/B/C/D) added for testing purposes.
Underlying view ZP_P_MaterialStockPeriods contains an enhancement
for PerLFWIM.

Final result are four key figures:
MatlWrhsStkQtyInMatlBaseUnit - Stock Level Quantity at end of period
MatlStkChangeQtyInBaseUnit   - Stock Change Quantity in period
MatlStkChangeQtyRolYearInBU  - Stock Change Quantity in Rolling Year
CountMatDocPos               - No Material Document Items contributing to Stock Change in period

Only the first key figure was available in the standard SAP view.
The others are results of the custom enhancements.
--------------------------------------------------------
Change-log:
--------------------------------------------------------
--------------------------------------------------------*/
@EndUserText.label: 'MM: P_MaterialStockTimeSeries1 (enh.)'
@AbapCatalog.sqlViewName: 'ZPMATSTCKTIMESR1'

@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK

// The client field in the two JOIN conditions below is used as a pseudo condition
// because there is no real common field in both JOINed views
// but fields from both views are required in each result row
// (client can be used even if it is not declared in the DDL of the views itself but the generated SE11 view has a client field)


// Having the fields for PeriodType and PostingDates in the ON condition of the JOIN with the type and date parameter
// is quite faster than having the PeriodType and PostingDates in the WHERE clause

define view ZP_P_MaterialStockTimeSeries1
  with parameters
    P_StartDate  : vdm_v_start_date,
    P_EndDate    : vdm_v_end_date,
    P_PeriodType : nsdm_period_type

//--"Stock Change from start of first time period" (Source = 'A')
  as select from    ZP_I_MaterialStockPeriodsSingl(P_StartDate:  :P_StartDate,
                                                   P_EndDate:    :P_EndDate,
                                                   P_PeriodType: :P_PeriodType ) as a
    left outer join ZP_I_MaterialStock_Aggr       as b on  a.mandt                  =  b.mandt
                                                    and a.PeriodType             =  :P_PeriodType
                                                    and b.MatlDocLatestPostgDate >= :P_StartDate
                                                    and b.MatlDocLatestPostgDate <= a.EndDate
{
//-------------------------------------------------------------------------------------------------
//------Start of custom enhancement, part 1/7
//-------------------------------------------------------------------------------------------------
  key 'A' as Source,
//-------------------------------------------------------------------------------------------------
//------End of custom enhancement, part 1/7
//-------------------------------------------------------------------------------------------------
  key a.PeriodType,
  key a.StartDate,
  key a.EndDate,
  key a.YearPeriod,
      // Stock Identifier
  key b.Material,
  key b.Plant,
  key b.StorageLocation,
  key b.Batch,
  key b.Supplier,
  key b.SDDocument,
  key b.SDDocumentItem,
  key b.WBSElementInternalID,
  key b.Customer,
  key b.InventoryStockType,
  key b.InventorySpecialStockType,
  key b.CostEstimate,
      // Units
  key b.MaterialBaseUnit,
      // Stock Groups
  key b.CompanyCode,
  key a.FiscalYearVariant,
//-------------------------------------------------------------------------------------------------
//------Start of custom enhancement, part 2/7
//-------------------------------------------------------------------------------------------------
key b.StockChangeCategory,

      a.StartDateRolYear,
//-------------------------------------------------------------------------------------------------
//------End of custom enhancement, part 2/7
//-------------------------------------------------------------------------------------------------
      @Semantics.quantity.unitOfMeasure: 'MaterialBaseUnit'
      sum(b.MatlWrhsStkQtyInMatlBaseUnit) as MatlWrhsStkQtyInMatlBaseUnit,
//-------------------------------------------------------------------------------------------------
//------Start of custom enhancement, part 3/7
//-------------------------------------------------------------------------------------------------
      @Semantics.quantity.unitOfMeasure: 'MaterialBaseUnit'
      cast(0 as nsdm_stock_qty)           as MatlStkChangeQtyInBaseUnit,
      @Semantics.quantity.unitOfMeasure: 'MaterialBaseUnit'
      cast(0 as nsdm_stock_qty)           as MatlStkChangeQtyRolYearInBU,

      0 as CountMatDocPos
//-------------------------------------------------------------------------------------------------
//------End of custom enhancement, part 3/7
//-------------------------------------------------------------------------------------------------
}
group by
  a.PeriodType,
  a.StartDate,
  a.EndDate,
  a.YearPeriod,
  b.Material,
  b.Plant,
  b.StorageLocation,
  b.Batch,
  b.Supplier,
  b.SDDocument,
  b.SDDocumentItem,
  b.WBSElementInternalID,
  b.Customer,
  b.InventoryStockType,
  b.InventorySpecialStockType,
  b.CostEstimate,
  b.MaterialBaseUnit,
  b.CompanyCode,
  a.FiscalYearVariant,
  b.StockChangeCategory,
  a.StartDateRolYear

union all

//--"Stock Level before start of first time period" (Source = 'B')
select from       ZP_I_MaterialStockPeriodsSingl(P_StartDate:  :P_StartDate,
            P_EndDate:    :P_EndDate,
            P_PeriodType: :P_PeriodType )                             as a
  left outer join ZP_P_MaterialStockByKeyDate1(P_KeyDate:  :P_StartDate) as b on  a.mandt      = b.mandt
                                                                           and a.PeriodType = :P_PeriodType
{
//-------------------------------------------------------------------------------------------------
//------Start of custom enhancement, part 4/7
//-------------------------------------------------------------------------------------------------
  key 'B' as Source,
//-------------------------------------------------------------------------------------------------
//------End of custom enhancement, part 4/7
//-------------------------------------------------------------------------------------------------
  key a.PeriodType,
  key a.StartDate,
  key a.EndDate,
  key a.YearPeriod,
      // Stock Identifier
  key b.Material,
  key b.Plant,
  key b.StorageLocation,
  key b.Batch,
  key b.Supplier,
  key b.SDDocument,
  key b.SDDocumentItem,
  key b.WBSElementInternalID,
  key b.Customer,
  key b.InventoryStockType,
  key b.InventorySpecialStockType,
  key b.CostEstimate,
      // Units
  key b.MaterialBaseUnit,
      // Stock Groups
  key b.CompanyCode,
  key a.FiscalYearVariant,
//-------------------------------------------------------------------------------------------------
//------Start of custom enhancement, part 5/7
//-------------------------------------------------------------------------------------------------
key b.StockChangeCategory,

      a.StartDateRolYear,
//-------------------------------------------------------------------------------------------------
//------End of custom enhancement, part 5/7
```

```
//-------------------------------------------------------------------------------------------------
      @Semantics.quantity.unitOfMeasure: 'MaterialBaseUnit'
      sum(b.MatlWrhsStkQtyInMatlBaseUnit) as MatlWrhsStkQtyInMatlBaseUnit,
//-------------------------------------------------------------------------------------------------
//------Start of custom enhancement, part 6/7
//-------------------------------------------------------------------------------------------------
      @Semantics.quantity.unitOfMeasure: 'MaterialBaseUnit'
      0                                   as MatlStkChangeQtyInBaseUnit,
      @Semantics.quantity.unitOfMeasure: 'MaterialBaseUnit'
      0                                   as MatlStkChangeQtyRolYearInBU,

      0 as CountMatDocPos
//-------------------------------------------------------------------------------------------------
//------End of custom enhancement, part 6/7
//-------------------------------------------------------------------------------------------------
}
group by
  a.PeriodType,
  a.StartDate,
  a.EndDate,
  a.YearPeriod,
  b.Material,
  b.Plant,
  b.StorageLocation,
  b.Batch,
  b.Supplier,
  b.SDDocument,
  b.SDDocumentItem,
  b.WBSElementInternalID,
  b.Customer,
  b.InventoryStockType,
  b.InventorySpecialStockType,
  b.CostEstimate,
  b.MaterialBaseUnit,
  b.CompanyCode,
  a.FiscalYearVariant,
  b.StockChangeCategory,
  a.StartDateRolYear

//-------------------------------------------------------------------------------------------------
//------Start of custom enhancement, part 7/7
//-------------------------------------------------------------------------------------------------
union all

//--“Stock Change within a time period” (Source = 'C')
  select from    ZP_I_MaterialStockPeriodsSingl(P_StartDate:  :P_StartDate,
                                                   P_EndDate:    :P_EndDate,
                                                   P_PeriodType: :P_PeriodType ) as a
    left outer join ZP_I_MaterialStock_Aggr       as b on  a.mandt                  =  b.mandt
                                                    and a.PeriodType             =  :P_PeriodType
                                                    and b.MatlDocLatestPostgDate >= a.StartDate
                                                    and b.MatlDocLatestPostgDate <= a.EndDate

{
  key 'C' as Source,
  key a.PeriodType,
  key a.StartDate,
  key a.EndDate,
  key a.YearPeriod,
      // Stock Identifier
  key b.Material,
  key b.Plant,
  key b.StorageLocation,
  key b.Batch,
  key b.Supplier,
  key b.SDDocument,
  key b.SDDocumentItem,
  key b.WBSElementInternalID,
  key b.Customer,
  key b.InventoryStockType,
  key b.InventorySpecialStockType,
  key b.CostEstimate,
      // Units
  key b.MaterialBaseUnit,
      // Stock Groups
  key b.CompanyCode,
  key a.FiscalYearVariant,
  key b.StockChangeCategory,

      a.StartDateRolYear,

      // Quantity and Value
      @Semantics.quantity.unitOfMeasure: 'MaterialBaseUnit'
      0                                   as MatlWrhsStkQtyInMatlBaseUnit,
      @Semantics.quantity.unitOfMeasure: 'MaterialBaseUnit'
      sum(b.MatlWrhsStkQtyInMatlBaseUnit) as MatlStkChangeQtyInBaseUnit,
      @Semantics.quantity.unitOfMeasure: 'MaterialBaseUnit'
      0                                   as MatlStkChangeQtyRolYearInBU,

      sum(b.CountMatDocPos) as CountMatDocPos
}
group by
  a.PeriodType,
  a.StartDate,
  a.EndDate,
  a.YearPeriod,
  b.Material,
  b.Plant,
  b.StorageLocation,
  b.Batch,
  b.Supplier,
  b.SDDocument,
  b.SDDocumentItem,
  b.WBSElementInternalID,
  b.Customer,
  b.InventoryStockType,
  b.InventorySpecialStockType,
  b.CostEstimate,
  b.MaterialBaseUnit,
  b.CompanyCode,
  a.FiscalYearVariant,
  b.StockChangeCategory,
  a.StartDateRolYear

  union all

//--“Stock Change within a "Rolling Year" till end of the time period” (Source = 'D')
  select from    ZP_I_MaterialStockPeriodsSingl(P_StartDate:  :P_StartDate,
                                                  P_EndDate:    :P_EndDate,
                                                   P_PeriodType: :P_PeriodType ) as a
    left outer join ZP_I_MaterialStock_Aggr       as b on  a.mandt                  =  b.mandt
                                                    and a.PeriodType             =  :P_PeriodType
                                                    and b.MatlDocLatestPostgDate >= a.StartDateRolYear
                                                    and b.MatlDocLatestPostgDate <= a.EndDate
{
  key 'D' as Source,
  key a.PeriodType,
  key a.StartDate,
  key a.EndDate,
  key a.YearPeriod,
      // Stock Identifier
  key b.Material,
  key b.Plant,
  key b.StorageLocation,
  key b.Batch,
  key b.Supplier,
  key b.SDDocument,
  key b.SDDocumentItem,
  key b.WBSElementInternalID,
  key b.Customer,
  key b.InventoryStockType,
  key b.InventorySpecialStockType,
  key b.CostEstimate,
      // Units
  key b.MaterialBaseUnit,
      // Stock Groups
  key b.CompanyCode,
  key a.FiscalYearVariant,
  key b.StockChangeCategory,

      a.StartDateRolYear,                        //Start date of "Rolling Year" till end of time period, e.g. 25.09.2020 -> 26.09.2019

      // Quantity and Value
      @Semantics.quantity.unitOfMeasure: 'MaterialBaseUnit'
      0                                   as MatlWrhsStkQtyInMatlBaseUnit,
      @Semantics.quantity.unitOfMeasure: 'MaterialBaseUnit'
      0                                   as MatlStkChangeQtyInBaseUnit,
      @Semantics.quantity.unitOfMeasure: 'MaterialBaseUnit'
      sum(b.MatlWrhsStkQtyInMatlBaseUnit) as MatlStkChangeQtyRolYearInBU,

      sum(b.CountMatDocPos) as CountMatDocPos
}
group by
  a.PeriodType,
  a.StartDate,
  a.EndDate,
  a.YearPeriod,
  b.Material,
  b.Plant,
  b.StorageLocation,
  b.Batch,
  b.Supplier,
  b.SDDocument,
  b.SDDocumentItem,
  b.WBSElementInternalID,
  b.Customer,
  b.InventoryStockType,
  b.InventorySpecialStockType,
  b.CostEstimate,
  b.MaterialBaseUnit,
  b.CompanyCode,
  a.FiscalYearVariant,
  b.StockChangeCategory,
  a.StartDateRolYear
//-------------------------------------------------------------------------------------------------
//------End of custom enhancement, part 7/7
//-------------------------------------------------------------------------------------------------
```