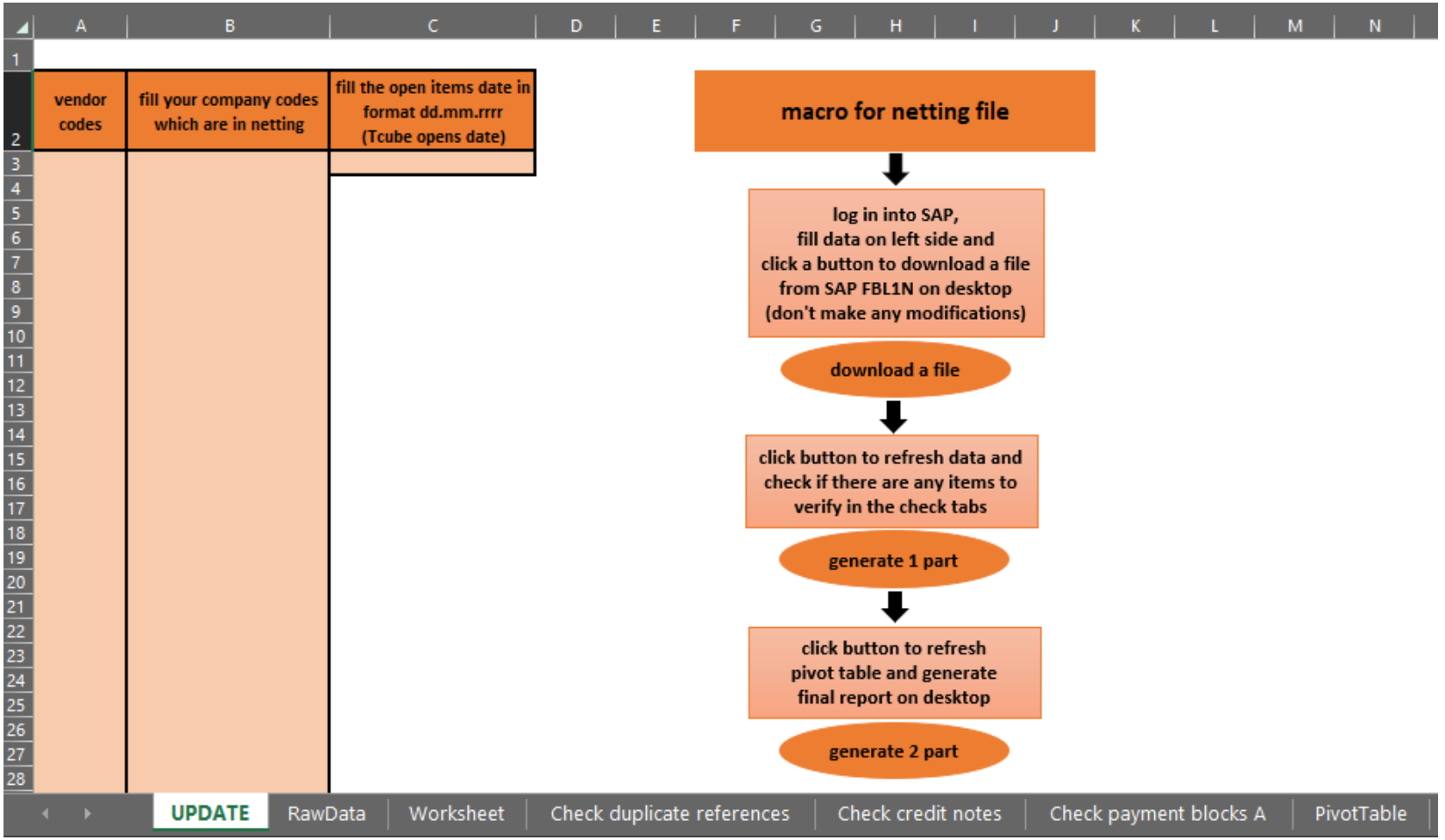
Power Query + Pivot Tables + Macros => Macro for netting file

*** Company data was hidden, among others: by blurring them, replacing them in the code with xxx**

The Excel file consists of the following tabs:

- UPDATE - which contains detailed instructions on what to do step by step, data for downloading the report and buttons with macros to refresh data in other tabs and generate the final report on the desktop;
- RawData - which contains raw data from SAP;
- Worksheet - which contains transformed data thanks to Power Query;
- tabs with the name Check - which were created thanks to Power Query, indicating possible manual items to be checked;
- PivotTable - contains a summary of data shown in a pivot table.

Finally, a final Excel file is created for the desktop without external connections to Power Query, containing tabs: RawData, Worksheet, PivotTable separated by company codes for easier further analysis.



Power Query editor:

The downloaded data from SAP was loaded into Power Query and named RawData.
Then the Raw Data table was duplicated and named Worksheet. This is where the data transformation was carried out, such as: removing unnecessary columns, filtering out empty and unnecessary rows based on specific criteria, defining data types.
Then three references to the Worksheet table were created, named:

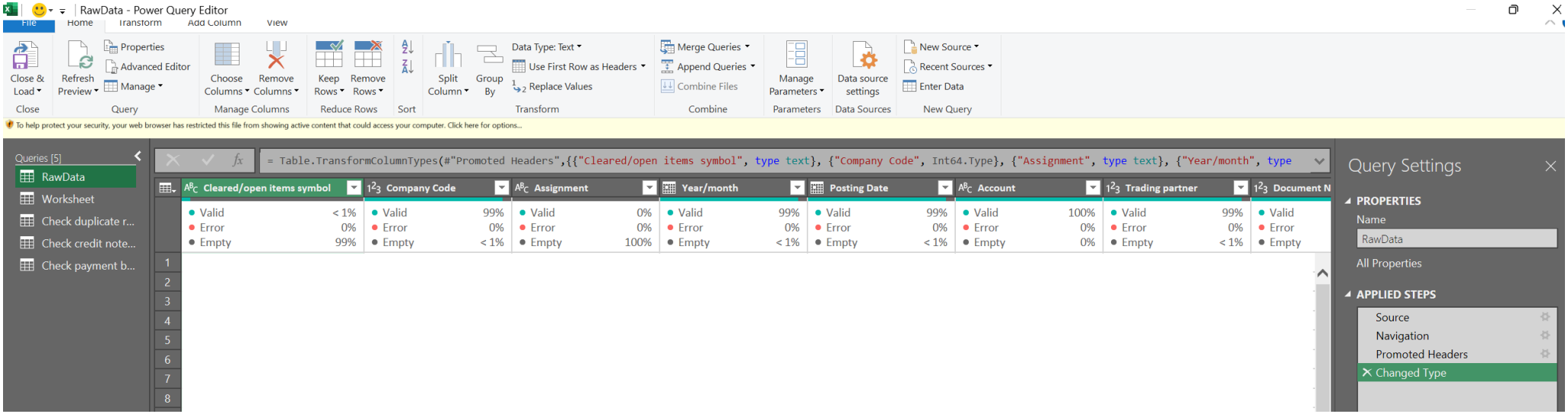
Check duplicate reference number in column - reference,

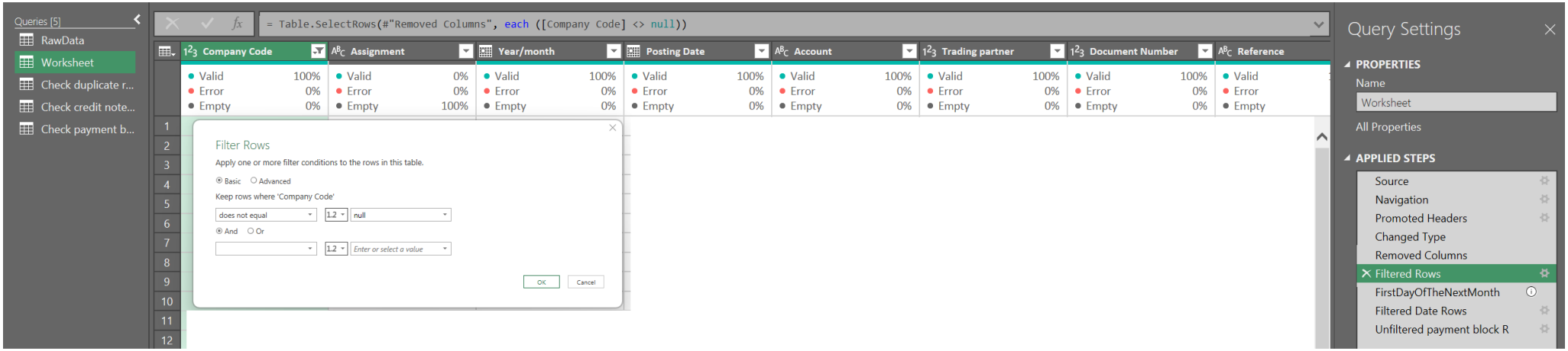
Check credit notes in column - amount in document currency,

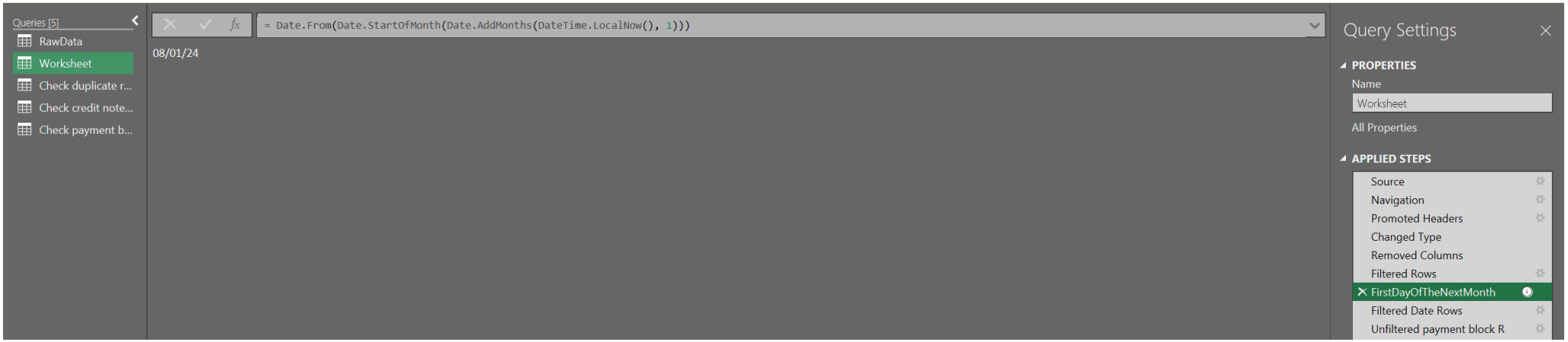
Check payment blocks A in column - payment block,

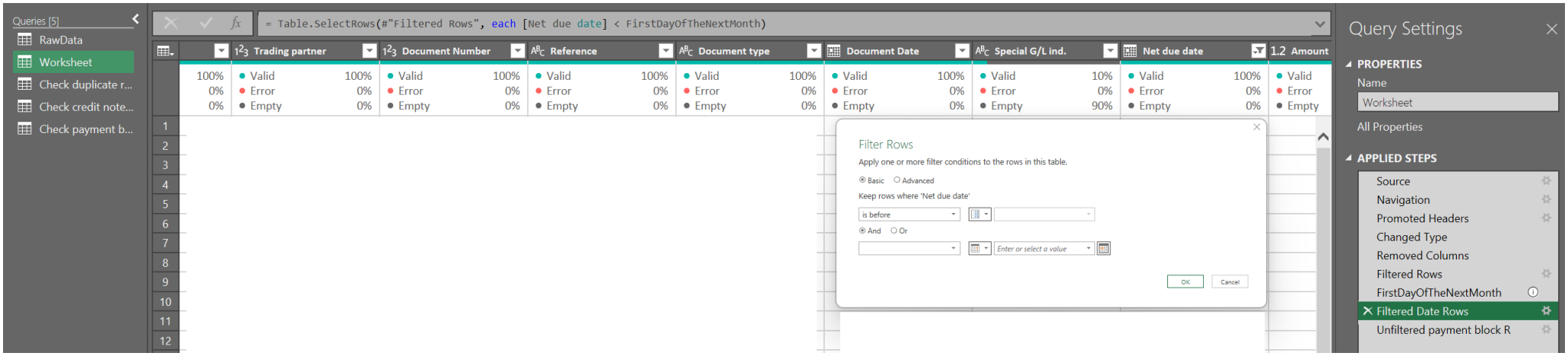
these tables are intended to check given conditions and show possible items for manual checking. They used such data transformations as: grouping, filtering, combining queries by join, adding a custom column, removing unnecessary columns.

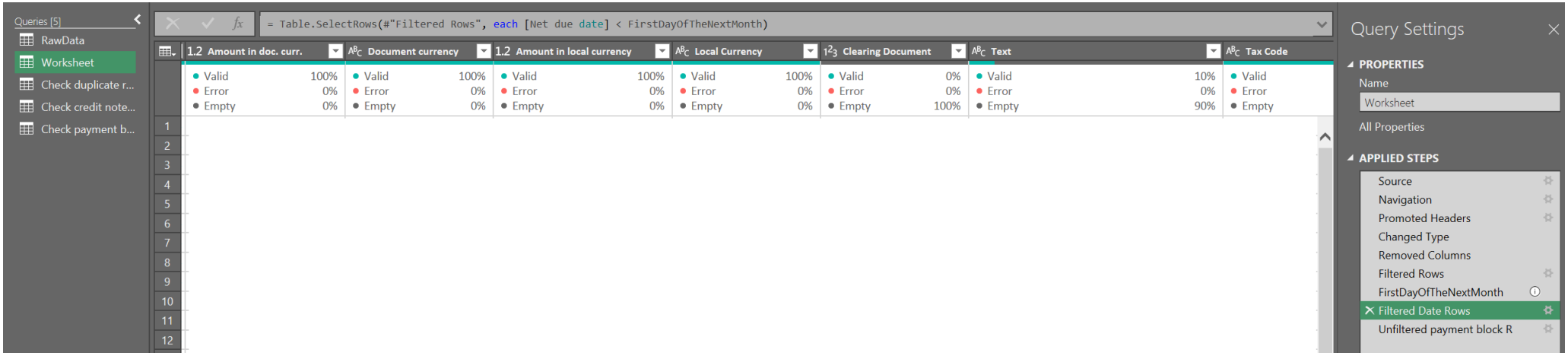
The screenshots below show the data transformations.

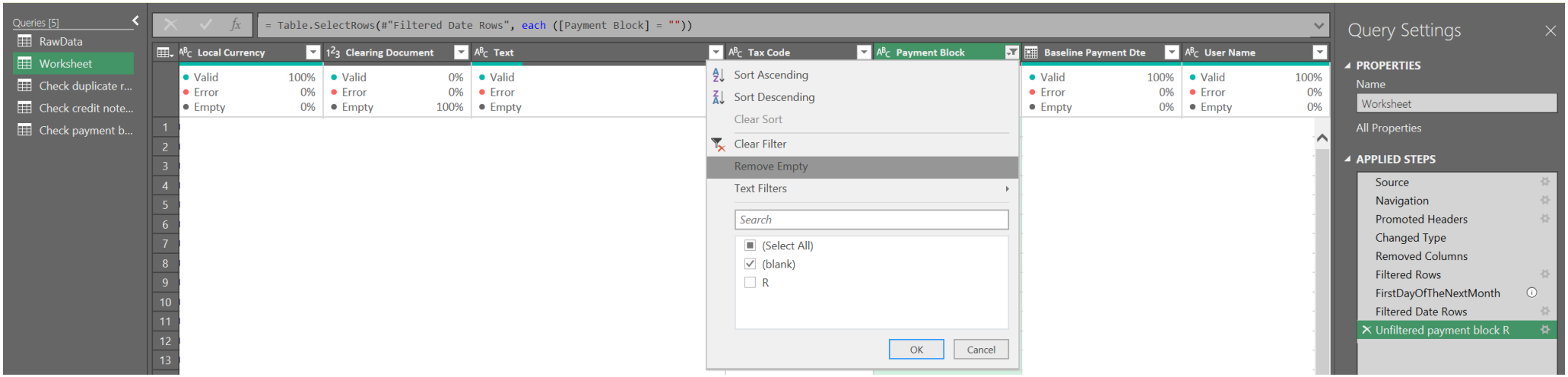


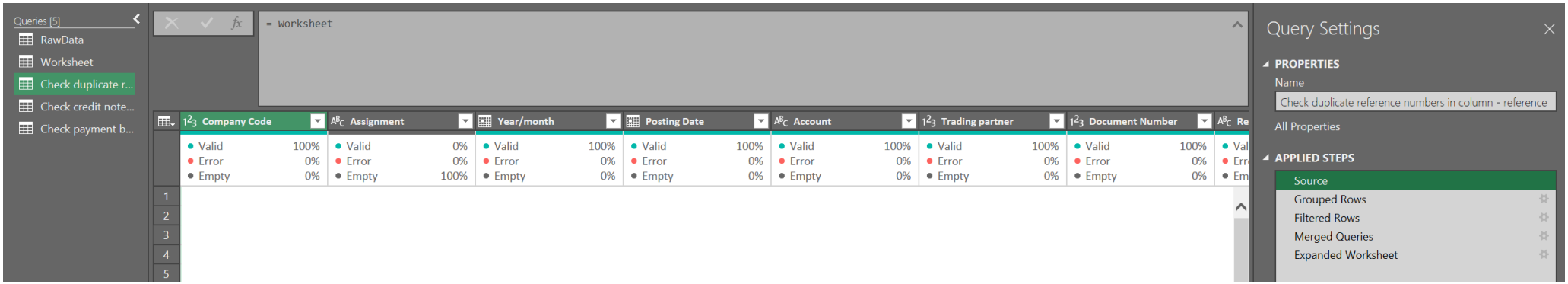


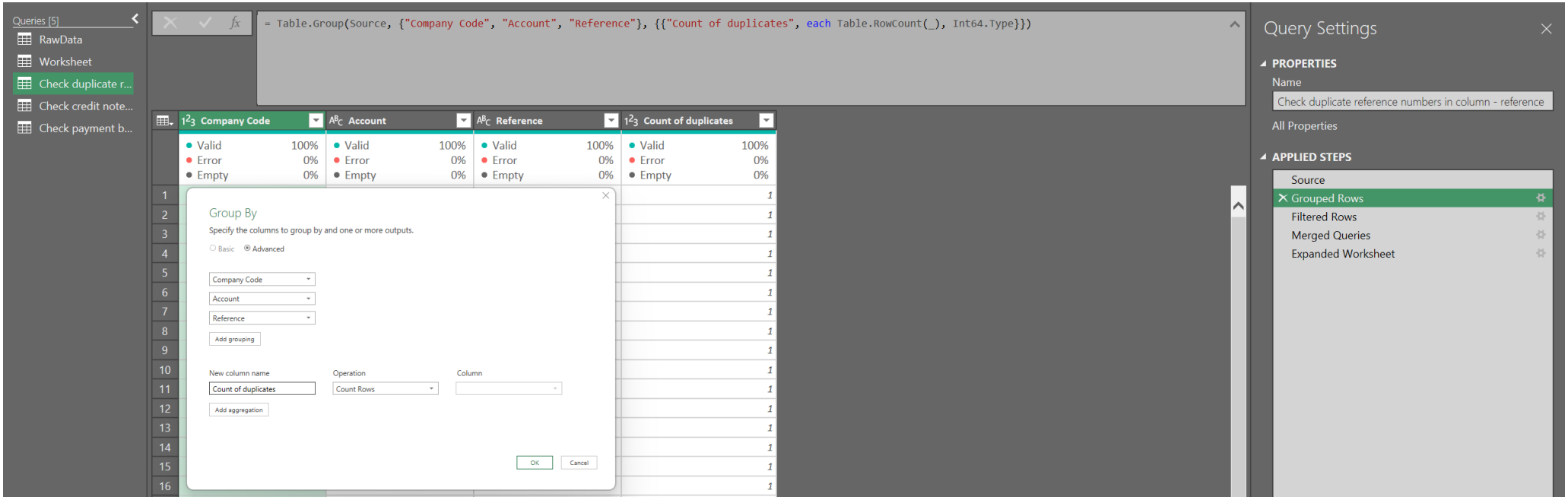

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Date.From(Date.StartOfMonth(Date.AddMonths(DateTime.LocalNow(), 1)))
08/01/24
Query Settings
PROPERTIES
Name
Worksheet
All Properties
APPLIED STEPS
Source
Navigation
Promoted Headers
Changed Type
Removed Columns
Filtered Rows
FirstDayOfTheNextMonth
Filtered Date Rows
Unfiltered payment block R

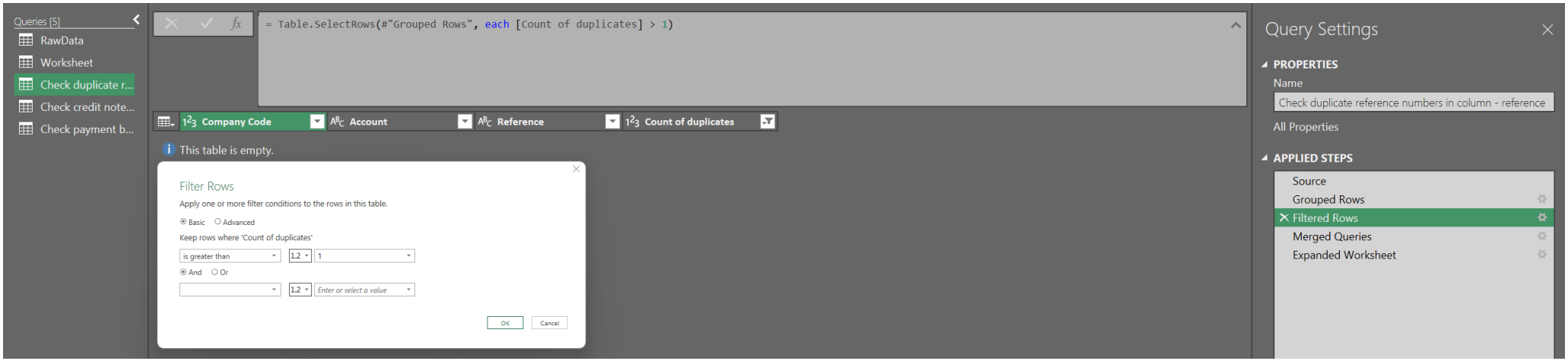

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Table.SelectRows(#"Filtered Rows", each [Net due date] < FirstDayOfTheNextMonth)
Trading partner
Document Number
Reference
Document type
Document Date
Special G/L ind.
Net due date
Amount
Valid
Error
Empty
Filter Rows
Apply one or more filter conditions to the rows in this table.
Basic
Advanced
Keep rows where 'Net due date'
is before
And
Or
Enter or select a value
OK
Cancel
Query Settings
PROPERTIES
Name
Worksheet
All Properties
APPLIED STEPS
Source
Navigation
Promoted Headers
Changed Type
Removed Columns
Filtered Rows
FirstDayOfTheNextMonth
Filtered Date Rows
Unfiltered payment block R

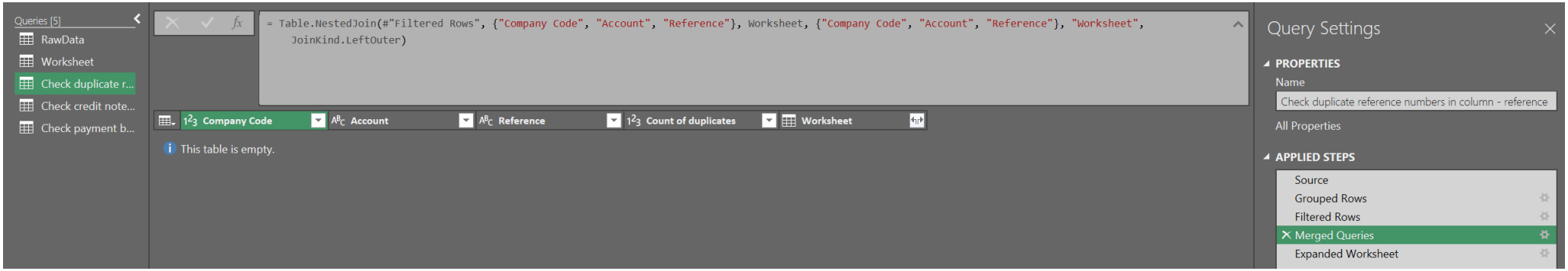

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Table.SelectRows(#"Filtered Rows", each [Net due date] < FirstDayOfTheNextMonth)
1.2 Amount in doc. curr.
Document currency
1.2 Amount in local currency
Local Currency
Clearing Document
Text
Tax Code
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Valid 0% Error 0% Empty 100%
Valid 10% Error 0% Empty 90%
Valid Error Empty
Query Settings
PROPERTIES
Name
Worksheet
All Properties
APPLIED STEPS
Source
Navigation
Promoted Headers
Changed Type
Removed Columns
Filtered Rows
FirstDayOfTheNextMonth
Filtered Date Rows
Unfiltered payment block R

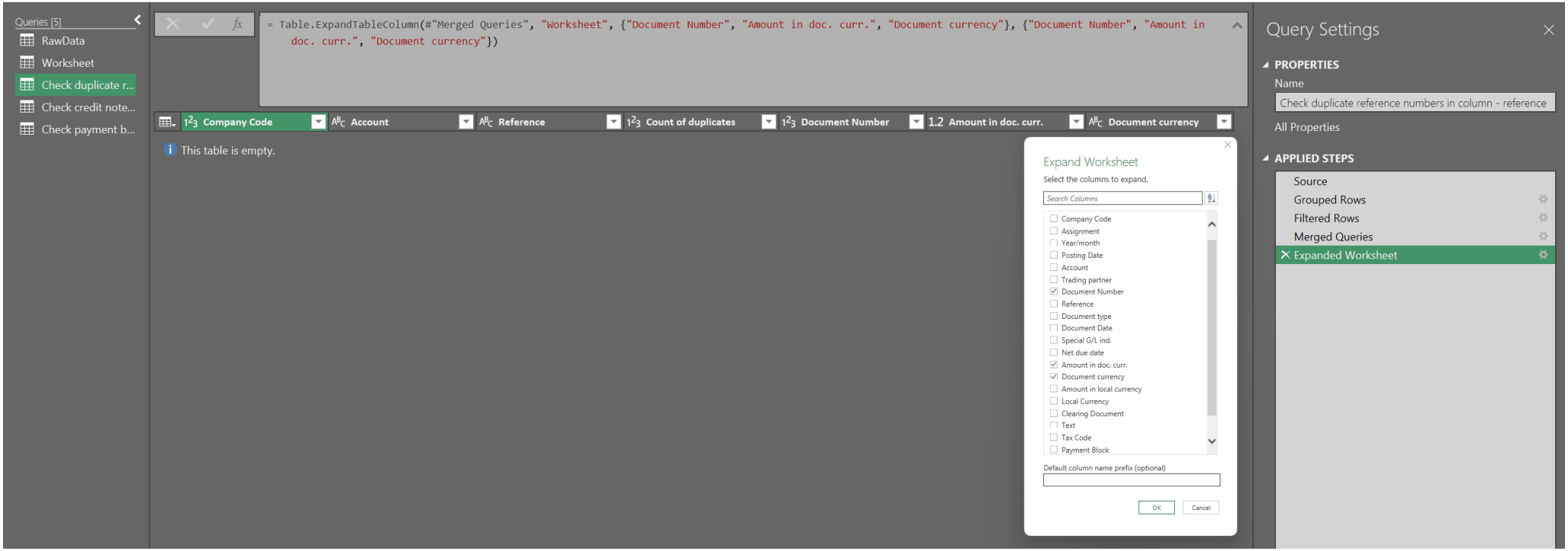

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Table.SelectRows(#"Filtered Date Rows", each ([Payment Block] = ""))
Local Currency
Clearing Document
Text
Tax Code
Payment Block
Baseline Payment Dte
User Name
Valid 100% Error 0% Empty 0%
Valid 0% Error 0% Empty 100%
Valid Error Empty
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Sort Ascending
Sort Descending
Clear Sort
Clear Filter
Remove Empty
Text Filters
Search
(Select All)
(blank)
R
OK
Cancel
Query Settings
PROPERTIES
Name
Worksheet
All Properties
APPLIED STEPS
Source
Navigation
Promoted Headers
Changed Type
Removed Columns
Filtered Rows
FirstDayOfTheNextMonth
Filtered Date Rows
Unfiltered payment block R

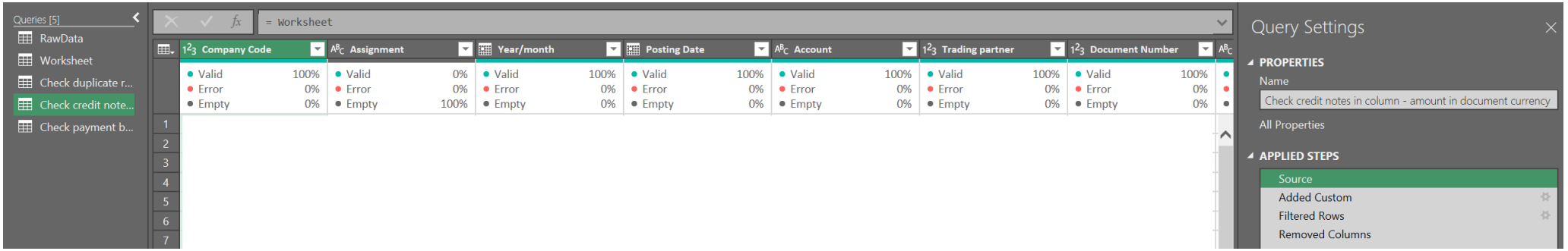

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Worksheet
Company Code
Assignment
Year/month
Posting Date
Account
Trading partner
Document Number
Valid 100% Error 0% Empty 0%
Valid 0% Error 0% Empty 100%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Query Settings
PROPERTIES
Name
Check duplicate reference numbers in column - reference
All Properties
APPLIED STEPS
Source
Grouped Rows
Filtered Rows
Merged Queries
Expanded Worksheet

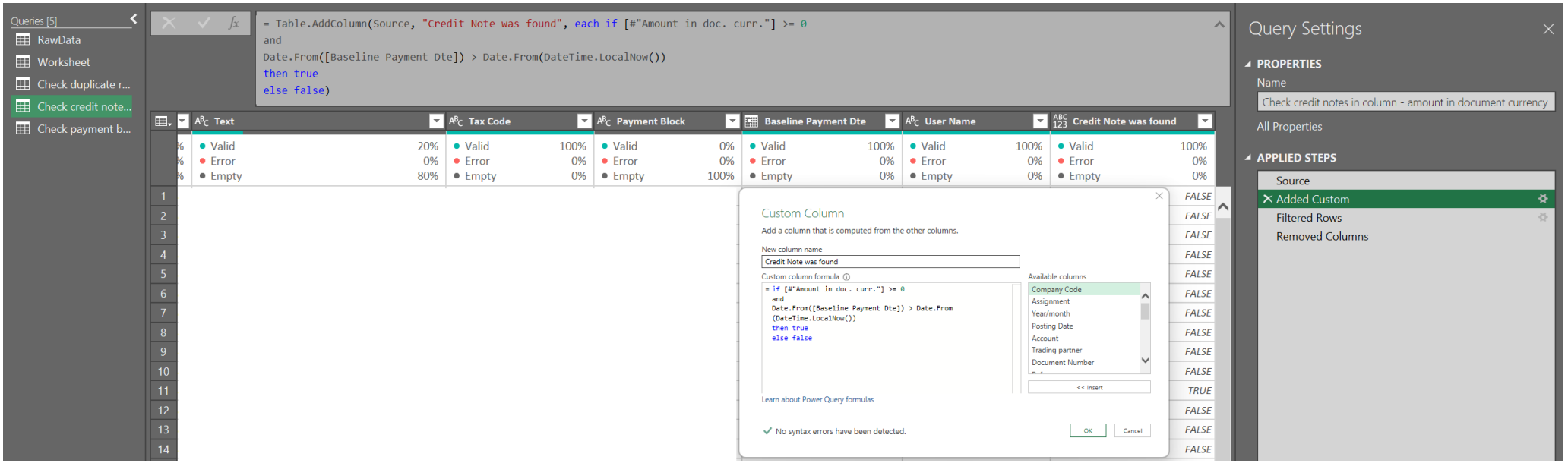

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Table.Group(Source, {"Company Code", "Account", "Reference"}, {{"Count of duplicates", each Table.RowCount(_), Int64.Type}})
Company Code
Account
Reference
Count of duplicates
Valid 100%
Error 0%
Empty 0%
Group By
Specify the columns to group by and one or more outputs.
Basic
Advanced
Company Code
Account
Reference
Add grouping
New column name
Operation
Column
Count of duplicates
Count Rows
Add aggregation
OK
Cancel
Query Settings
PROPERTIES
Name
Check duplicate reference numbers in column - reference
All Properties
APPLIED STEPS
Source
Grouped Rows
Filtered Rows
Merged Queries
Expanded Worksheet

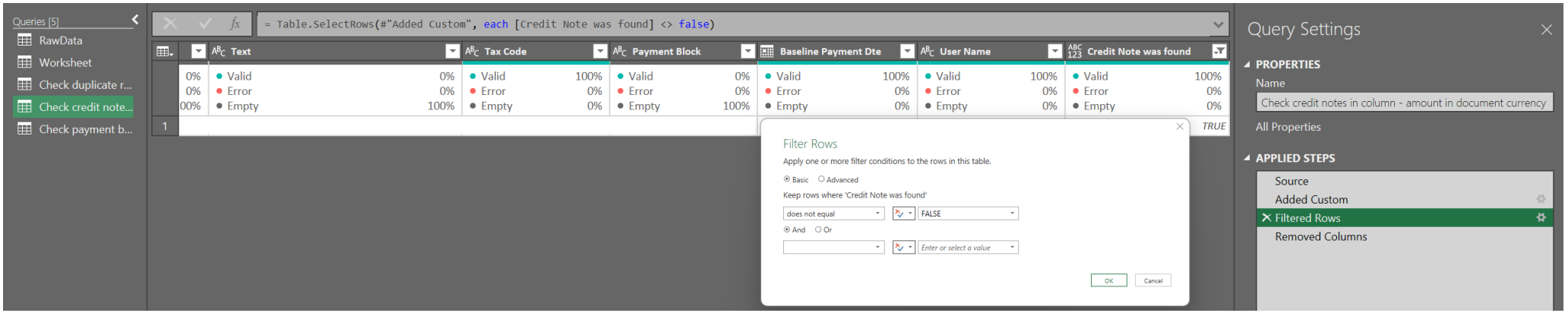

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Table.SelectRows(#"Grouped Rows", each [Count of duplicates] > 1)
Company Code
Account
Reference
Count of duplicates
This table is empty.
Filter Rows
Apply one or more filter conditions to the rows in this table.
Basic
Advanced
Keep rows where 'Count of duplicates'
is greater than
1
And
Or
Enter or select a value
OK
Cancel
Query Settings
PROPERTIES
Name
Check duplicate reference numbers in column - reference
All Properties
APPLIED STEPS
Source
Grouped Rows
Filtered Rows
Merged Queries
Expanded Worksheet

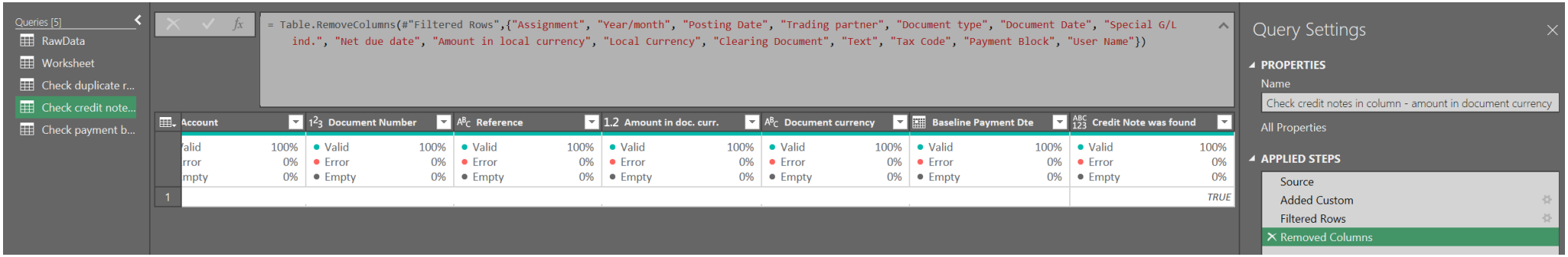

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Table.NestedJoin(#"Filtered Rows", {"Company Code", "Account", "Reference"}, Worksheet, {"Company Code", "Account", "Reference"}, "Worksheet", JoinKind.LeftOuter)
Company Code
Account
Reference
Count of duplicates
Worksheet
This table is empty.
Query Settings
PROPERTIES
Name
Check duplicate reference numbers in column - reference
All Properties
APPLIED STEPS
Source
Grouped Rows
Filtered Rows
Merged Queries
Expanded Worksheet

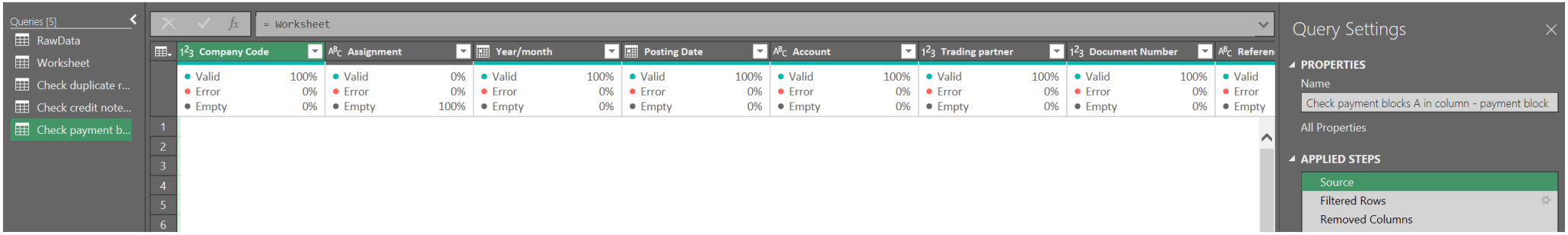

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Table.ExpandTableColumn(#"Merged Queries", "Worksheet", {"Document Number", "Amount in doc. curr.", "Document currency"}, {"Document Number", "Amount in doc. curr.", "Document currency"})
Company Code
Account
Reference
Count of duplicates
Document Number
Amount in doc. curr.
Document currency
This table is empty.
Expand Worksheet
Select the columns to expand.
Search Columns
Company Code
Assignment
Year/month
Posting Date
Account
Trading partner
Document Number
Reference
Document type
Document Date
Special G/L ind.
Net due date
Amount in doc. curr.
Document currency
Amount in local currency
Local Currency
Clearing Document
Text
Tax Code
Payment Block
Default column name prefix (optional)
OK
Cancel
Query Settings
PROPERTIES
Name
Check duplicate reference numbers in column - reference
All Properties
APPLIED STEPS
Source
Grouped Rows
Filtered Rows
Merged Queries
Expanded Worksheet

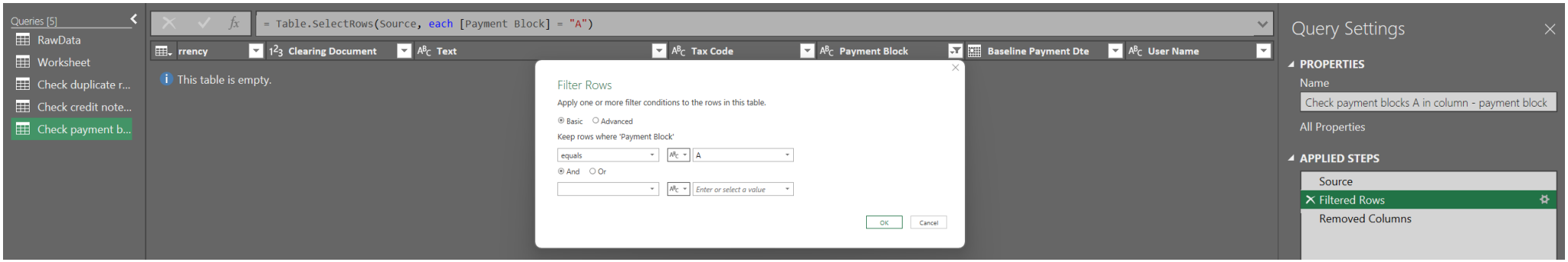

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Worksheet
Company Code
Assignment
Year/month
Posting Date
Account
Trading partner
Document Number
Valid 100% Error 0% Empty 0%
Valid 0% Error 0% Empty 100%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Query Settings
PROPERTIES
Name
Check credit notes in column - amount in document currency
All Properties
APPLIED STEPS
Source
Added Custom
Filtered Rows
Removed Columns

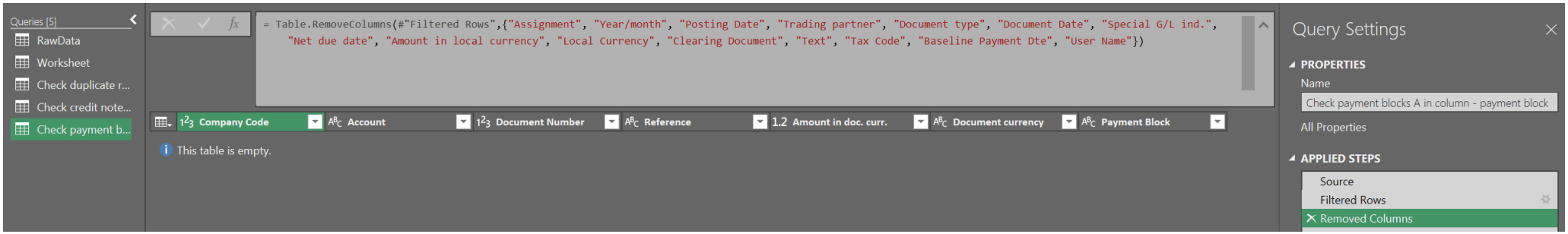

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Table.AddColumn(Source, "Credit Note was found", each if [#"Amount in doc. curr."] >= 0
and
Date.From([Baseline Payment Dte]) > Date.From(DateTime.LocalNow())
then true
else false)
Text
Tax Code
Payment Block
Baseline Payment Dte
User Name
Credit Note was found
Valid 20% Error 0% Empty 80%
Valid 100% Error 0% Empty 0%
Valid 0% Error 0% Empty 100%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
FALSE
TRUE
Custom Column
Add a column that is computed from the other columns.
New column name
Credit Note was found
Custom column formula
Available columns
Company Code
Assignment
Year/month
Posting Date
Account
Trading partner
Document Number
<< Insert
Learn about Power Query formulas
No syntax errors have been detected.
OK
Cancel
Query Settings
PROPERTIES
Name
Check credit notes in column - amount in document currency
All Properties
APPLIED STEPS
Source
Added Custom
Filtered Rows
Removed Columns


Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Table.SelectRows(#"Added Custom", each [Credit Note was found] <> false)
Text
Tax Code
Payment Block
Baseline Payment Dte
User Name
Credit Note was found
Valid 0% Error 0% Empty 100%
Valid 100% Error 0% Empty 0%
Valid 0% Error 0% Empty 100%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
Valid 100% Error 0% Empty 0%
TRUE
Filter Rows
Apply one or more filter conditions to the rows in this table.
Basic
Advanced
Keep rows where 'Credit Note was found'
does not equal
FALSE
And
Or
Enter or select a value
OK
Cancel
Query Settings
PROPERTIES
Name
Check credit notes in column - amount in document currency
All Properties
APPLIED STEPS
Source
Added Custom
Filtered Rows
Removed Columns


Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Table.RemoveColumns(#"Filtered Rows",{"Assignment", "Year/month", "Posting Date", "Trading partner", "Document type", "Document Date", "Special G/L ind.", "Net due date", "Amount in local currency", "Local Currency", "Clearing Document", "Text", "Tax Code", "Payment Block", "User Name"})
Account
Document Number
Reference
Amount in doc. curr.
Document currency
Baseline Payment Dte
Credit Note was found
Valid 100%
Error 0%
Empty 0%
TRUE
Query Settings
PROPERTIES
Name
Check credit notes in column - amount in document currency
All Properties
APPLIED STEPS
Source
Added Custom
Filtered Rows
Removed Columns

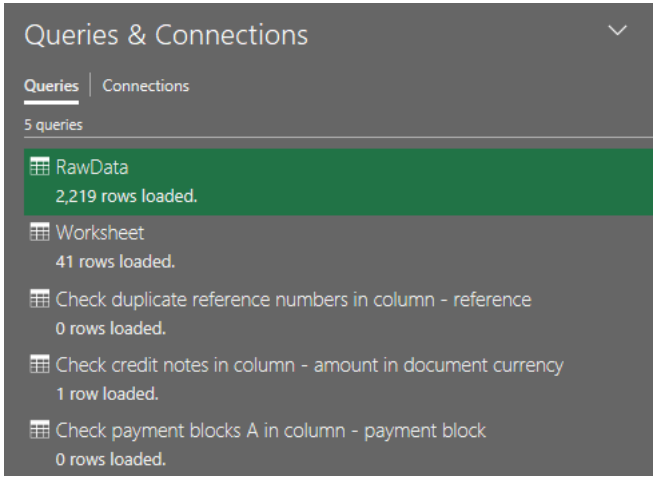

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Worksheet
Company Code
Assignment
Year/month
Posting Date
Account
Trading partner
Document Number
Valid 100%
Error 0%
Empty 0% / 100%
Query Settings
PROPERTIES
Name
Check payment blocks A in column - payment block
All Properties
APPLIED STEPS
Source
Filtered Rows
Removed Columns

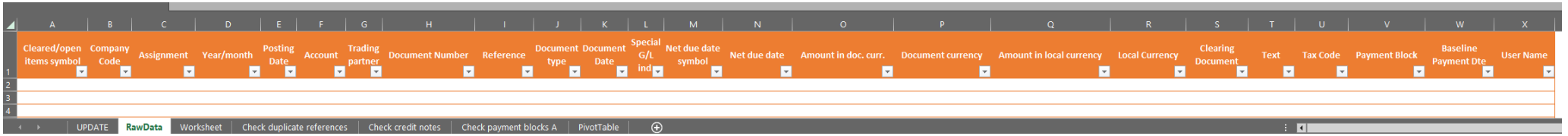

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Table.SelectRows(Source, each [Payment Block] = "A")
Clearing Document
Text
Tax Code
Payment Block
Baseline Payment Dte
User Name
This table is empty.
Filter Rows
Apply one or more filter conditions to the rows in this table.
Basic
Advanced
Keep rows where 'Payment Block'
equals
A
And
Or
Enter or select a value
OK
Cancel
Query Settings
PROPERTIES
Name
Check payment blocks A in column - payment block
All Properties
APPLIED STEPS
Source
Filtered Rows
Removed Columns

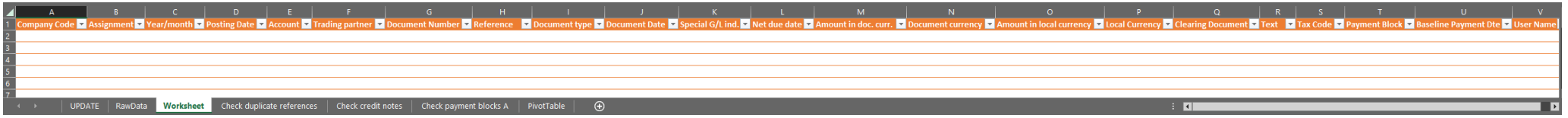

Queries [5]
RawData
Worksheet
Check duplicate r...
Check credit note...
Check payment b...
= Table.RemoveColumns(#"Filtered Rows",{"Assignment", "Year/month", "Posting Date", "Trading partner", "Document type", "Document Date", "Special G/L ind.", "Net due date", "Amount in local currency", "Local Currency", "Clearing Document", "Text", "Tax Code", "Baseline Payment Dte", "User Name"})
Company Code
Account
Document Number
Reference
Amount in doc. curr.
Document currency
Payment Block
This table is empty.
Query Settings
PROPERTIES
Name
Check payment blocks A in column - payment block
All Properties
APPLIED STEPS
Source
Filtered Rows
Removed Columns

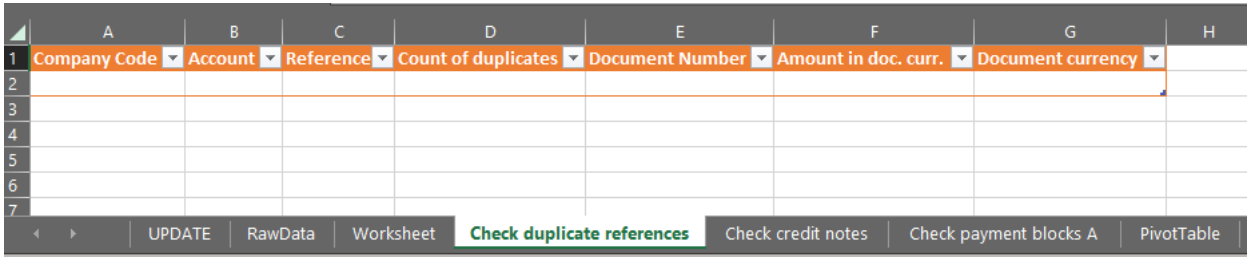
Excel:

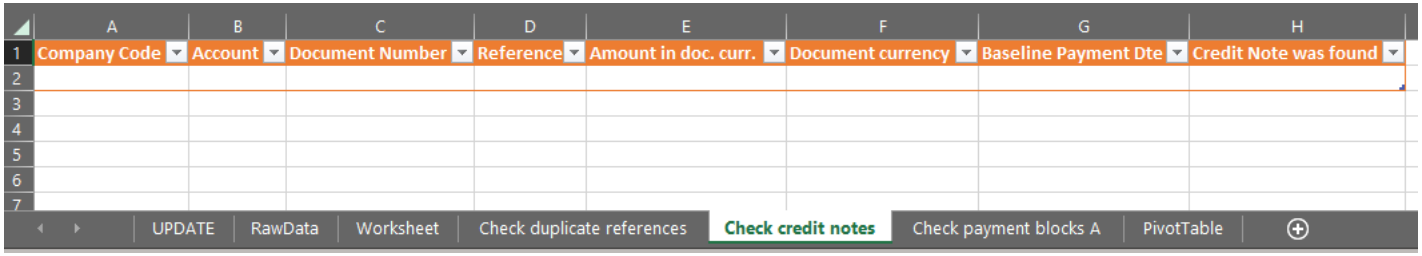
The above tables were loaded from Power Query to Excel.





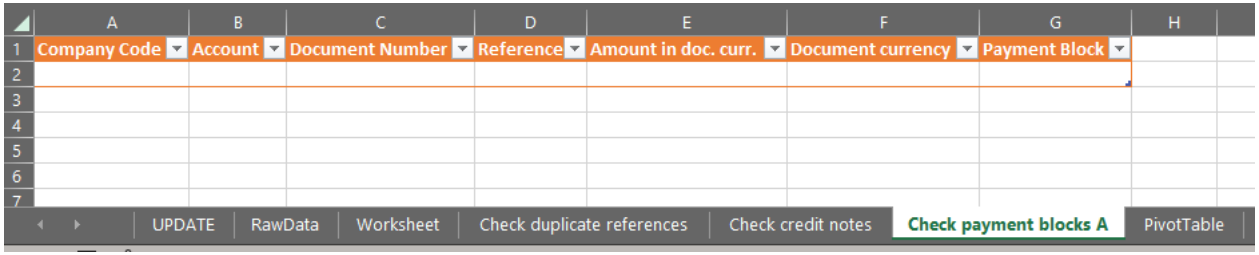






Company Code	Account	Document Number	Reference	Amount in doc. curr.	Document currency	Baseline Payment Dte	Credit Note was found

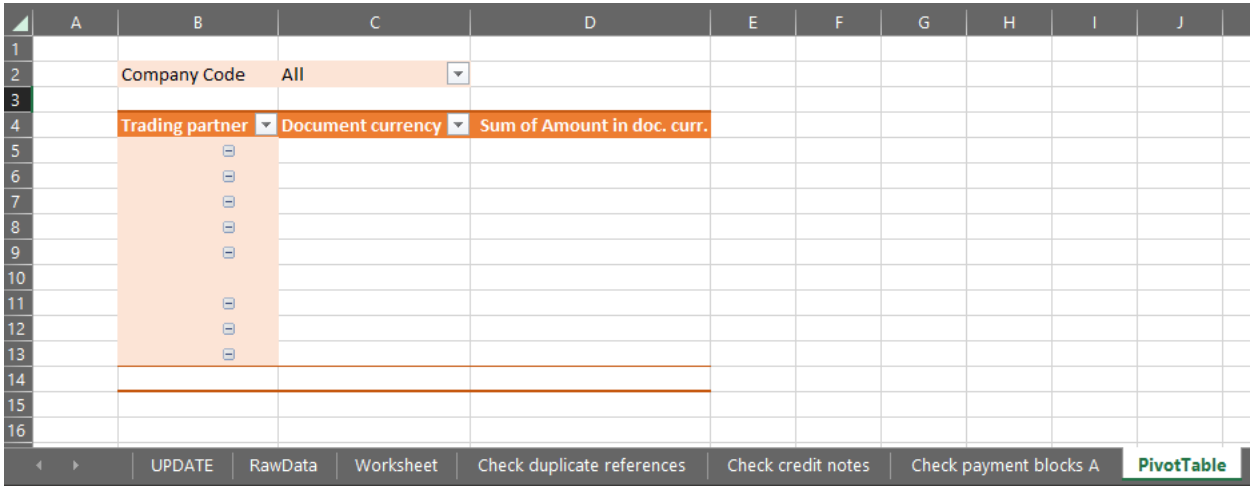
UPDATE | RawData | Worksheet | Check duplicate references | **Check credit notes** | Check payment blocks A | PivotTable



Company Code	Account	Document Number	Reference	Amount in doc. curr.	Document currency	Payment Block

UPDATE | RawData | Worksheet | Check duplicate references | Check credit notes | **Check payment blocks A** | PivotTable

A pivot table was created based on the data from the Worksheet.

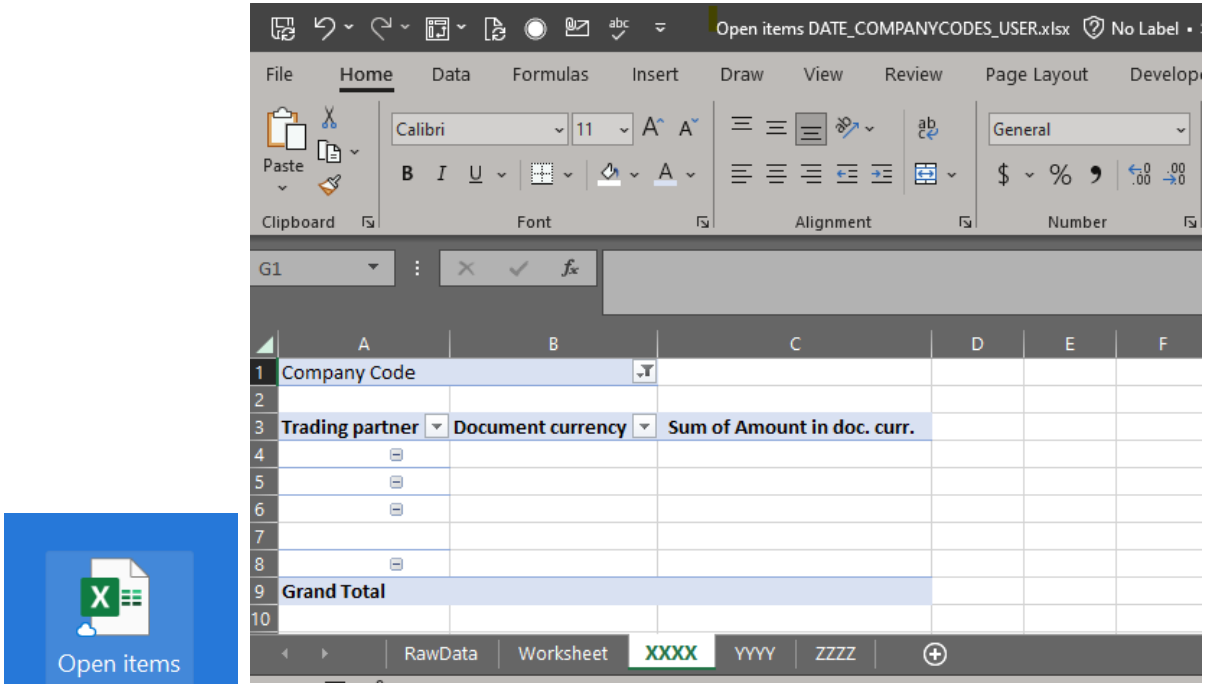


Company Code	All

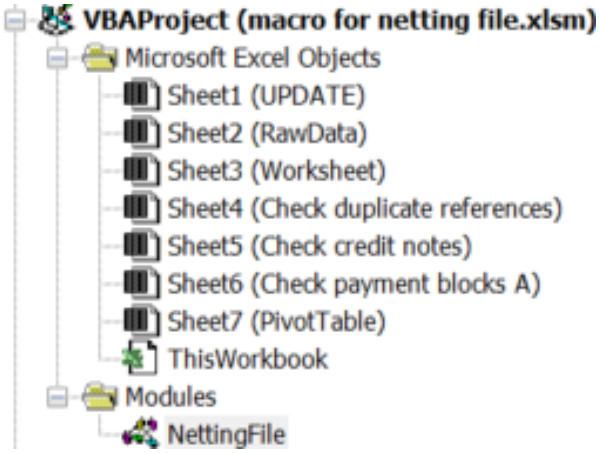
Trading partner	Document currency	Sum of Amount in doc. curr.

UPDATE | RawData | Worksheet | Check duplicate references | Check credit notes | Check payment blocks A | **PivotTable**

Generated final report:



VBA editor:



Code for button: download a file

(General) (Declarations)

```
Option Explicit

'created by Wiktoria Sliwinska
'macro for netting file


Sub downloading_a_file_from_SAP()

    'turn on the SAP library: C:\Program Files (x86)\SAP\FrontEnd\SAPgui\sapfewse.ocx
    'log into SAP before macro execution

    On Error GoTo ErrorHandler

    Dim SapGuiAuto As Object
    Dim objGui As Object
    Dim objConn As Object
    Dim objSess As Object

    Set SapGuiAuto = GetObject("SAPGUI")
    Set objGui = SapGuiAuto.GetScriptingEngine
    Set objConn = objGui.Children(0)
    Set objSess = objConn.Children(0)

    Dim wsMacro As Worksheet
    Set wsMacro = ThisWorkbook.Sheets("UPDATE")

    Dim rng As Range
    Dim lastRow As Long

    'enter transaction
    objSess.FindById("wnd[0]/tbar[0]/okcd").Text = "fbl1n"
    objSess.FindById("wnd[0]").SendVKey 0

    'enter vendor accounts
    lastRow = wsMacro.Cells(wsMacro.Rows.Count, "A").End(xlUp).Row
    Set rng = wsMacro.Range("A3:A" & lastRow)
    rng.Copy
    objSess.FindById("wnd[0]/usr/ctxtKD_LIFNR-LOW").Text = ""
    objSess.FindById("wnd[0]/usr/btn%_KD_LIFNR_%_APP_%-VALU_PUSH").Press
    Application.Wait (Now + TimeValue("0:00:01")) 'added delay
    objSess.FindById("wnd[1]").SendVKey 24
    Application.Wait (Now + TimeValue("0:00:01")) 'added delay
    objSess.FindById("wnd[1]").SendVKey 8
```

```
    'enter company codes
    lastRow = wsMacro.Cells(wsMacro.Rows.Count, "B").End(xlUp).Row
    Set rng = wsMacro.Range("B3:B" & lastRow)
    rng.Copy
    objSess.FindById("wnd[0]/usr/ctxtKD_BUKRS-LOW").Text = ""
    objSess.FindById("wnd[0]/usr/btn%_KD_BUKRS_%_APP_%-VALU_PUSH").Press
    Application.Wait (Now + TimeValue("0:00:01")) 'added delay
    objSess.FindById("wnd[1]").SendVKey 24
    Application.Wait (Now + TimeValue("0:00:01")) 'added delay
    objSess.FindById("wnd[1]").SendVKey 8

    'clean up the clipboard
    Application.CutCopyMode = False

    'open items
    objSess.FindById("wnd[0]/usr/ctxtPA_STIDA").Text = wsMacro.Range("C3").Value

    'check normal items and special G/L transactions
    objSess.FindById("wnd[0]/usr/chkX_NORM").Selected = True
    objSess.FindById("wnd[0]/usr/chkX_SHBV").Selected = True

    'execute
    objSess.FindById("wnd[0]/tbar[1]/btn[8]").Press

    'wait for the report to load
    Application.Wait (Now + TimeValue("0:00:06"))

    'save a file EXPORT.xlsx on desktop
    objSess.FindById("wnd[0]/mbar/menu[0]/menu[3]/menu[1]").Select
    Application.Wait (Now + TimeValue("0:00:01")) 'added delay
    objSess.FindById("wnd[1]/tbar[0]/btn[0]").Press
    Application.Wait (Now + TimeValue("0:00:01")) 'added delay
    objSess.FindById("wnd[1]/usr/ctxtDY_PATH").Text = Environ("USERPROFILE") & "\OneDrive - XXXX\Desktop"
    objSess.FindById("wnd[1]/tbar[0]/btn[0]").Press

    'exit transaction and go to homepage
    objSess.FindById("wnd[0]/tbar[0]/btn[3]").Press
    objSess.FindById("wnd[0]/tbar[0]/btn[3]").Press

    MsgBox "Done. Close the SAP report.", vbInformation, "Information"

ExitSub:
    Exit Sub
```

```
ExitSub:
    Exit Sub

ErrorHandler:
    MsgBox "Error: " & Err.Description, vbExclamation, "Error"
    Resume ExitSub


End Sub
```

Code for button: generate 1 part

```
Sub refresh_data()

    ActiveWorkbook.RefreshAll
    MsgBox "Verify check tabs."

End Sub
```

Code for button: generate 2 part

```vba
Sub refresh_pivot_table_and_generate_report()

    Application.ScreenUpdating = False 'turns off the visibility of windows
    Application.DisplayAlerts = False 'turns off the visibility of Excel alerts

    ActiveWorkbook.Sheets("PivotTable").PivotTables("PivotTableNetting").PivotCache.Refresh
    ActiveWorkbook.Sheets("PivotTable").Columns("D:D").Style = "Comma"

    'save the current excel file
    ActiveWorkbook.Save


    'create final file


    'define paths and file names
    Dim desktopPath As String
    desktopPath = Environ("USERPROFILE") & "\OneDrive - XXXX\Desktop\"
    Dim FinalReportName As String
    Dim FinalReportPath As String
    FinalReportName = "Open items DATE_COMPANYCODES_USER.xlsx"
    FinalReportPath = desktopPath & FinalReportName

    'create a new workbook
    Dim NewWorkbook As Workbook
    Set NewWorkbook = Workbooks.Add


    'copy data from the existing sheets to the new workbook
    With ThisWorkbook
        .Sheets("RawData").Copy Before:=NewWorkbook.Sheets(1)
        NewWorkbook.Sheets(1).Name = "RawData"
        .Sheets("Worksheet").Copy After:=NewWorkbook.Sheets("RawData")
        NewWorkbook.Sheets("Worksheet").Name = "Worksheet"
    End With

    'delete the default "Sheet1" sheet in the new workbook
    NewWorkbook.Sheets("Sheet1").Delete
```

```
'unlink the tables in the copied sheets to remove Power Query connections
NewWorkbook.Sheets("RawData").ListObjects("RawData").Unlink
NewWorkbook.Sheets("Worksheet").ListObjects("Worksheet").Unlink

'remove all Power Query connections in the new workbook
Dim pq As Object
For Each pq In NewWorkbook.Queries
    pq.Delete
Next pq
'remove all connections in the new workbook
Dim conn As WorkbookConnection
For Each conn In NewWorkbook.Connections
    On Error Resume Next
    conn.Delete
    On Error GoTo 0
Next conn


'creating pivot tables

'framework of a pivot table

Dim Destination, RangeData As Range

Sheets.Add(After:=Worksheets("Worksheet")).Name = "PivotTable"

Set RangeData = Worksheets("Worksheet").Range("A1").CurrentRegion
Set Destination = Worksheets("PivotTable").Range("B4")

ActiveWorkbook.PivotCaches.Create _
(SourceType:=xlDatabase, SourceData:=RangeData) _
.CreatePivotTable TableDestination:=Destination, TableName:="NettingPivotTable"

'fields collection of a pivot table

Dim PvtTable As PivotTable
Set PvtTable = Worksheets("PivotTable").PivotTables("NettingPivotTable")

Dim PageName, RowName1, RowName2, ColumnName, DataName As String
PageName = "Company Code"
RowName1 = "Trading partner"
RowName2 = "Document currency"
'ColumnName =
DataName = "Amount in doc. curr."
```

```
With PvtTable
    .PivotFields(PageName).Orientation = xlPageField
    .PivotFields(RowName1).Orientation = xlRowField 'Position = 1
    .PivotFields(RowName2).Orientation = xlRowField 'Position = 2
    '.PivotFields(ColumnName).Orientation = xlColumnField
    .PivotFields(DataName).Orientation = xlDataField 'making "Sum of Amount in doc. curr."
End With

'loop through all pivot fields and remove subtotals

Dim PvtField As PivotField

For Each PvtField In PvtTable.PivotFields
    PvtField.Subtotals(1) = True
    PvtField.Subtotals(1) = False
Next PvtField

'show pivot table in tabular form

PvtTable.RowAxisLayout xlTabularRow

'set comma style (format with a thousands separator)

Columns("D:D").Select
Selection.Style = "Comma"

'set autofit on columns

Columns("B:D").Select
Columns("B:D").EntireColumn.AutoFit

'set focus on cell A1

Range("A1").Select


'show for each company code separate tab

ActiveSheet.PivotTables("NettingPivotTable").ShowPages PageField:="Company Code"
```

```
    'check if the worksheet name is a four-digit number (company code) and apply the specified modifications to those worksheets

     Dim ws As Worksheet
     Dim i As Integer
     i = 1 'initial number of sheet
     NewWorkbook.Activate

     For Each ws In Sheets

     If Len(ws.Name) = 4 And IsNumeric(ws.Name) Then
     ws.Activate
        With ws
            .Columns("C:C").Style = "Comma"
            .Columns("A:C").EntireColumn.AutoFit
            .Range("A1").Select
        End With

     i = i + 1

     End If

     Next ws


    'hide PivotTable tab
    Sheets("PivotTable").Select
    ActiveWindow.SelectedSheets.Visible = False


    'save and close the final report
    NewWorkbook.SaveAs Filename:=FinalReportPath
    NewWorkbook.Close 'SaveChanges:=True


Application.ScreenUpdating = True  'turns on the visibility of windows
Application.DisplayAlerts = True  'turns on the visibility of Excel alerts


       'display message that file is ready
MsgBox "File is ready", vbInformation, "Information"


End Sub
```