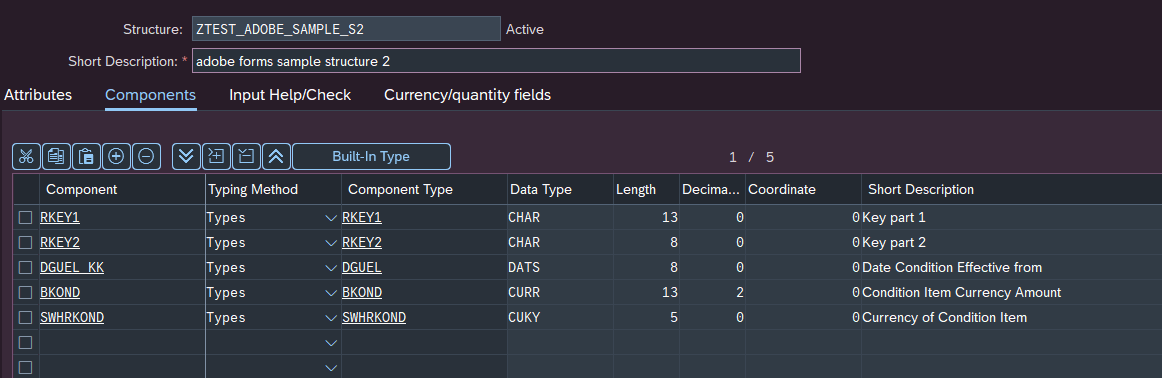
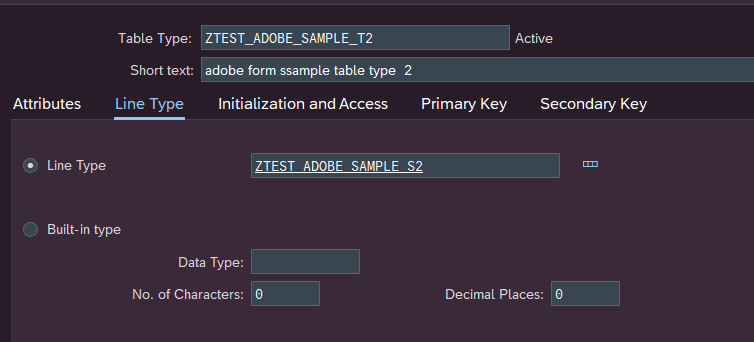
1.Prepare adobe forms interface structure header and item in **SE11**

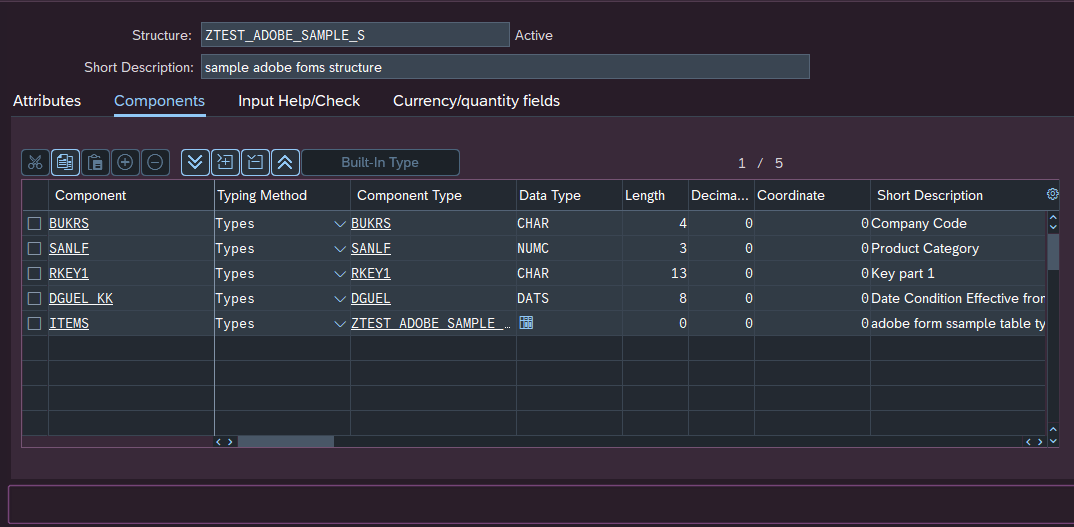
Item structure



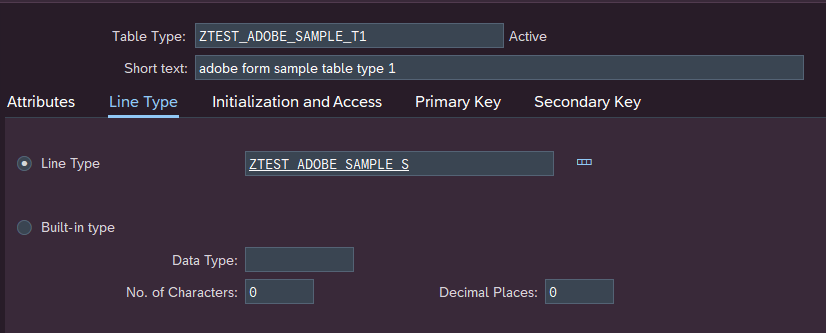
Create Item table type



**Header structure with ITEM TABLE TYPE ASSIGNED**

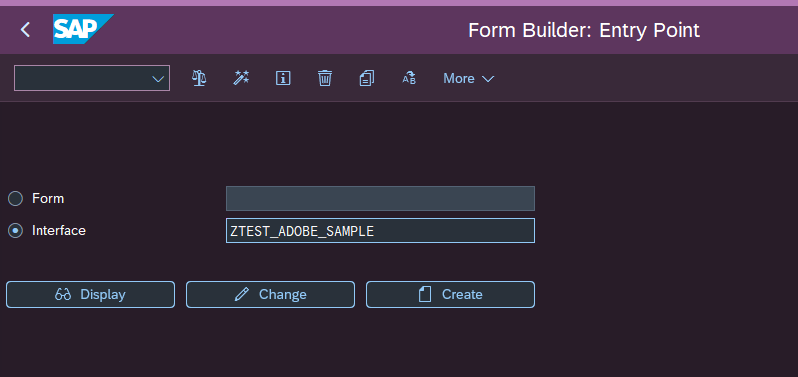


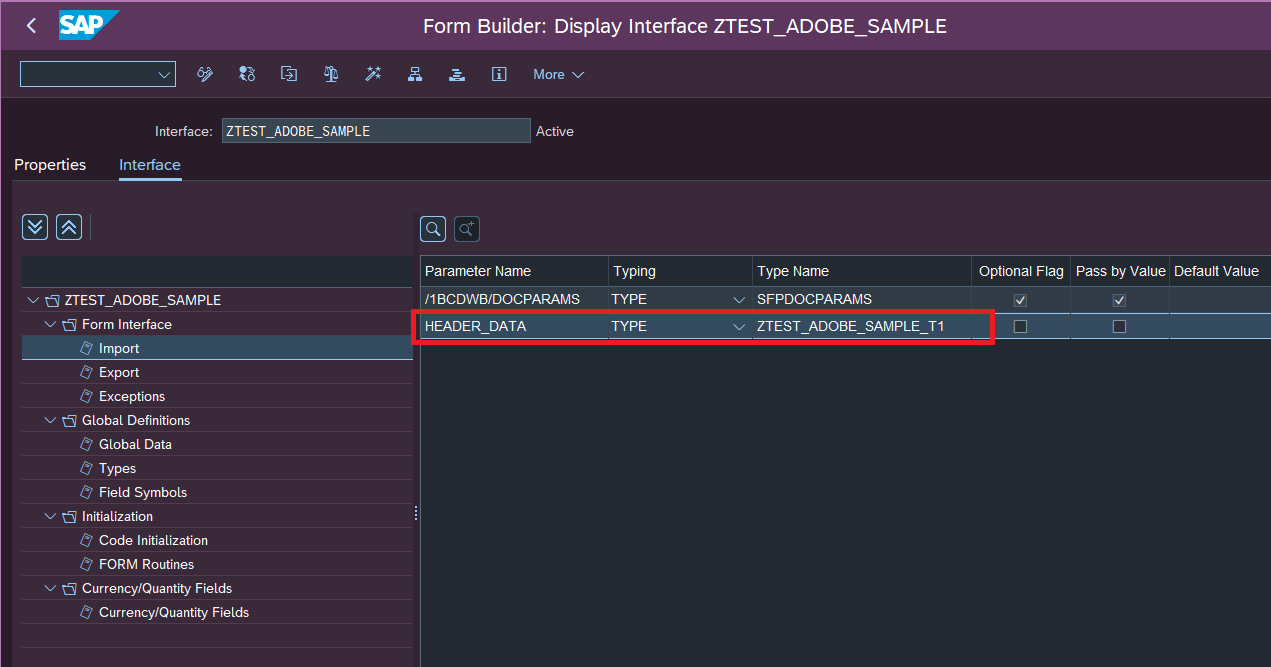
Create Header Table type



2. Create Adobe form Interface and assign the created Table type

Goto SFP transaction and create interface

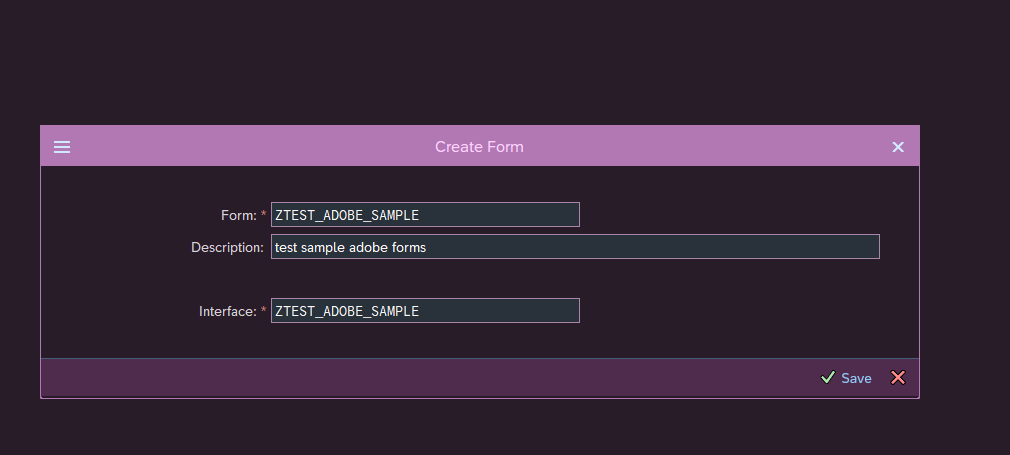


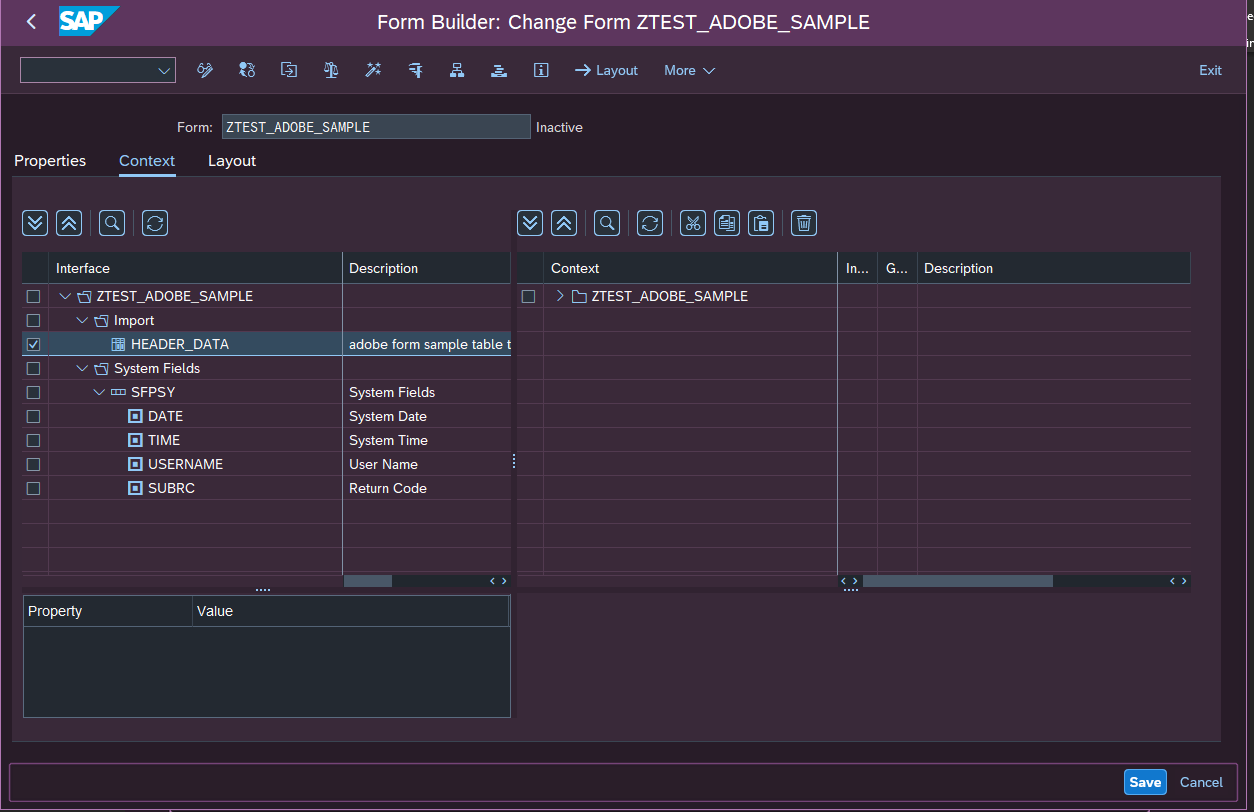


3. Create ADOBE Forms Driver Program with appropriate logic and sample code as follows

\*&---------------------------------------------------------------------\*  
\*& Report ZTEST\_ADOBE\_SAMPLE  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ztest\_adobe\_sample.  
  
DATA: gd\_path TYPE string.  
  
DATA : gt\_header TYPE STANDARD TABLE OF ztest\_adobe\_sample\_s,  
       gt\_item   TYPE STANDARD TABLE OF ztest\_adobe\_sample\_s2,  
       gs\_header TYPE ztest\_adobe\_sample\_s,  
       gs\_item   TYPE ztest\_adobe\_sample\_s2.  
  
DATA : gv\_conn TYPE rfcdest.  
  
PARAMETERS : p\_rkey1 TYPE rkey1 OBLIGATORY.  
  
  
SELECTION-SCREEN: BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-001.  
  
  PARAMETERS : " ADOBE FORM PREVIEW  
    rad1   RADIOBUTTON GROUP r1 DEFAULT 'X' USER-COMMAND ccc,  
  
    " SAVE DIALOG DIRECTORY POPUP  
    rad2   RADIOBUTTON GROUP r1,  
  
    " ENTER DIRECOTRY IN SELECTION SCREEN TO SAVE PDF FILES TO THE DIRECTORY  
    rad3   RADIOBUTTON GROUP r1,  
  
    p\_path TYPE rlgrap-filename OBLIGATORY MODIF ID ccc.  
SELECTION-SCREEN: END OF BLOCK b1.  
  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_path.  
  
  CALL METHOD cl\_gui\_frontend\_services=>directory\_browse  
    EXPORTING  
      window\_title    = 'File Directory'  
      initial\_folder  = 'C:'  
    CHANGING  
      selected\_folder = gd\_path.  
  CALL METHOD cl\_gui\_cfw=>flush.  
  
  CONCATENATE gd\_path '' INTO p\_path.  
  
AT SELECTION-SCREEN OUTPUT.  
  
  IF rad1 = 'X'.  
    LOOP AT SCREEN.  
      IF screen-group1 = 'CCC'.  
        screen-active = '0'.  
        MODIFY SCREEN.  
      ENDIF.  
    ENDLOOP.  
  
  ELSEIF rad2 = 'X'.  
    LOOP AT SCREEN.  
      IF screen-group1 = 'CCC'.  
        screen-active = '0'.  
        MODIFY SCREEN.  
      ENDIF.  
    ENDLOOP.  
  
  ELSEIF rad3 = 'X'.  
    LOOP AT SCREEN.  
      IF screen-group1 = 'CCC'.  
        screen-active = '1'.  
        MODIFY SCREEN.  
      ENDIF.  
    ENDLOOP.  
  ENDIF.  
  
  
INITIALIZATION.  
  gv\_conn = 'ADS'.  
  MOVE cl\_fp=>get\_ads\_connection( ) TO gv\_conn.  
  
START-OF-SELECTION.  
  
  SELECT \* FROM vzzkoko INTO TABLE @DATA(lt\_koko)  
    WHERE rkey1 EQ @p\_rkey1.  
  IF sy-subrc EQ 0.  
    SELECT \* FROM vzzkopo INTO TABLE @DATA(lt\_kopo)  
      FOR ALL ENTRIES IN @lt\_koko WHERE rkey1 EQ @lt\_koko-rkey1.  
  ENDIF.  
  
  IF lt\_kopo IS NOT INITIAL.  
    LOOP AT lt\_kopo INTO DATA(ls\_kopo).  
      MOVE-CORRESPONDING ls\_kopo TO gs\_item.  
      gs\_item-rkey2 = sy-tabix.  
      APPEND gs\_item TO gt\_item.  
      CLEAR : gs\_item, ls\_kopo.  
    ENDLOOP.  
  
    READ TABLE lt\_koko INTO DATA(ls\_koko) INDEX 1.  
    IF sy-subrc EQ 0.  
      MOVE-CORRESPONDING ls\_koko TO gs\_header.  
      gs\_header-items = gt\_item.  
      APPEND gs\_header TO gt\_header.  
    ENDIF.  
  ENDIF.  
  
\*\*\*\*\*\* ADOBE FORM processing  
\* VARIABLES declaration  
  
  DATA: lv\_fmname       TYPE fpname,  
        lv\_sign         TYPE xstring,  
        ls\_outputparams TYPE sfpoutputparams,  
        ls\_docparams    TYPE sfpdocparams,  
        ls\_pdf          TYPE fpformoutput  ##NEEDED.  
  
  DATA:lt\_data            TYPE STANDARD TABLE OF tabl1024,  
       lo\_pdf\_merger      TYPE REF TO cl\_rspo\_pdf\_merge,  
       lt\_pdf\_table       TYPE tfpcontent,  
       lv\_file            TYPE string,  
       lv\_path            TYPE string,  
       lv\_file\_name       TYPE string,  
       lv\_merged\_document TYPE xstring,  
       lv\_len             TYPE i,  
       lv\_rc              TYPE i VALUE 0,  
       result             TYPE sfpjoboutput,  
       l\_spoolid          TYPE rspoid,  
       lv\_filename        TYPE string VALUE 'ADOBE\_SAMPLE'.  
  
  
\* COSTANTS DECLARATION  
  CONSTANTS : lc\_form\_name TYPE fpname VALUE 'ZTEST\_ADOBE\_SAMPLE'.  
  
\* OUTPUT PRINTER SETTINGS  
  ls\_outputparams-dest     = 'LP01'. "Depends on user settings  
  
  IF rad1 NE abap\_true.  
    ls\_outputparams-nodialog = abap\_true.  
    ls\_outputparams-getpdf   = 'M'.  
    ls\_outputparams-assemble = 'S'.  
    ls\_outputparams-bumode   = 'M'.  
    ls\_outputparams-reqnew   = abap\_true.  
  ENDIF.  
  
  DATA(lt\_fin) = lt\_pdf\_table.  
  CLEAR lt\_fin.  
  
  TRY.  
      CALL FUNCTION 'FP\_FUNCTION\_MODULE\_NAME'   " Form Processing Generation  
        EXPORTING  
          i\_name     = lc\_form\_name  
        IMPORTING  
          e\_funcname = lv\_fmname.  
    CATCH  cx\_fp\_api\_usage ##NO\_HANDLER .  
    CATCH  cx\_fp\_api\_repository ##NO\_HANDLER .  
    CATCH  cx\_fp\_api\_internal ##NO\_HANDLER .  
  ENDTRY.  
  
  
  ls\_outputparams-connection = gv\_conn.  
\* Form Processing: Call Form  
  CALL FUNCTION 'FP\_JOB\_OPEN'  
    CHANGING  
      ie\_outputparams = ls\_outputparams  
    EXCEPTIONS  
      cancel          = 1  
      usage\_error     = 2  
      system\_error    = 3  
      internal\_error  = 4  
      OTHERS          = 5.  
  IF sy-subrc <> 0.  
    CLEAR ls\_outputparams.  
  ENDIF.  
  
  ls\_docparams-langu   = 'D'.  
  ls\_docparams-country = 'DE'.  
  
  CALL FUNCTION lv\_fmname  
    EXPORTING  
      /1bcdwb/docparams  = ls\_docparams  
\*     iv\_sign            = lv\_sign  
      header\_data        = gt\_header  
    IMPORTING  
      /1bcdwb/formoutput = ls\_pdf  
    EXCEPTIONS  
      usage\_error        = 1  
      system\_error       = 2  
      internal\_error     = 3  
      OTHERS             = 4.  
  IF sy-subrc <> 0.  
\* Implement suitable error handling here  
  ENDIF.  
  
  
\*&---- Close the spool job  
  CALL FUNCTION 'FP\_JOB\_CLOSE'  
    IMPORTING  
      e\_result       = result  
    EXCEPTIONS  
      usage\_error    = 1  
      system\_error   = 2  
      internal\_error = 3  
      OTHERS         = 4.  
  IF sy-subrc <> 0.  
    CLEAR ls\_pdf.  
  ENDIF.  
  
  
  IF rad1 EQ abap\_true.  
\* Check the result.  
    IF result-userexit IS INITIAL.  
      IF result-spoolids IS INITIAL.  
        MESSAGE ID 'PO' TYPE 'E' NUMBER '475'.  
      ELSE.  
        READ TABLE result-spoolids INDEX 1 INTO l\_spoolid.  
        MESSAGE ID 'PO' TYPE 'S' NUMBER '622' WITH l\_spoolid.  
      ENDIF.  
    ENDIF.  
  
  ENDIF.  
  
  
  IF rad2 EQ abap\_true OR  
     rad3 EQ abap\_true .  
  
    CALL FUNCTION 'FP\_GET\_PDF\_TABLE'  
      IMPORTING  
        e\_pdf\_table = lt\_pdf\_table.  
  
\*&&-- Merging different pdf files into one  
    CREATE OBJECT lo\_pdf\_merger.  
  
    APPEND LINES OF lt\_pdf\_table TO lt\_fin.  
  
\* Add documents to attribute table of PDF merger  
    LOOP AT lt\_fin INTO DATA(lwa\_form).  
      lo\_pdf\_merger->add\_document( lwa\_form ).  
    ENDLOOP.  
  
\* Call kernel method to do the merge of the specified files.  
    lo\_pdf\_merger->merge\_documents( IMPORTING merged\_document = lv\_merged\_document  
                                                           rc = lv\_rc ).  
  
    CALL FUNCTION 'SCMS\_XSTRING\_TO\_BINARY'  
      EXPORTING  
        buffer        = lv\_merged\_document  
      IMPORTING  
        output\_length = lv\_len  
      TABLES  
        binary\_tab    = lt\_data.  
  
  ENDIF.  
  
  
  IF rad2 EQ abap\_true.  
  
    lv\_filename = lv\_filename && '\_' && sy-datum && '\_' && sy-uzeit.  
    CALL METHOD cl\_gui\_frontend\_services=>file\_save\_dialog  
      EXPORTING  
        window\_title              = 'Save Form'  
        default\_extension         = 'pdf'  
        default\_file\_name         = lv\_filename  
        prompt\_on\_overwrite       = 'X'  
      CHANGING  
        filename                  = lv\_file\_name  
        path                      = lv\_path  
        fullpath                  = lv\_file  
      EXCEPTIONS  
        cntl\_error                = 1  
        error\_no\_gui              = 2  
        not\_supported\_by\_gui      = 3  
        invalid\_default\_file\_name = 4  
        OTHERS                    = 5.  
    IF sy-subrc <> 0.  
\* Implement suitable error handling here  
      MESSAGE 'PDF Process Error' TYPE 'E'.  
    ENDIF.  
  
  ENDIF.  
  
  
  IF rad2 EQ abap\_true OR  
     rad3 EQ abap\_true .  
  
  
    IF rad3 EQ abap\_true.  
      lv\_filename = lv\_filename && '\_' && sy-datum && '\_' && sy-timlo.  
      CONCATENATE p\_path '\' lv\_filename '.pdf' INTO lv\_file.  
    ENDIF.  
  
    CALL METHOD cl\_gui\_frontend\_services=>gui\_download  
      EXPORTING  
        bin\_filesize            = lv\_len  
        filename                = lv\_file  
        filetype                = 'BIN'  
      CHANGING  
        data\_tab                = lt\_data  
      EXCEPTIONS  
        file\_write\_error        = 1  
        no\_batch                = 2  
        gui\_refuse\_filetransfer = 3  
        invalid\_type            = 4  
        no\_authority            = 5  
        unknown\_error           = 6  
        header\_not\_allowed      = 7  
        separator\_not\_allowed   = 8  
        filesize\_not\_allowed    = 9  
        header\_too\_long         = 10  
        dp\_error\_create         = 11  
        dp\_error\_send           = 12  
        dp\_error\_write          = 13  
        unknown\_dp\_error        = 14  
        access\_denied           = 15  
        dp\_out\_of\_memory        = 16  
        disk\_full               = 17  
        dp\_timeout              = 18  
        file\_not\_found          = 19  
        dataprovider\_exception  = 20  
        control\_flush\_error     = 21  
        not\_supported\_by\_gui    = 22  
        error\_no\_gui            = 23  
        OTHERS                  = 24.  
    IF sy-subrc IS NOT INITIAL.  
      MESSAGE 'PDF Process Error' TYPE 'E'.  
    ENDIF.  
  
    CLEAR:lv\_file,lv\_len,lv\_path,  
          lv\_file\_name,lt\_data, lt\_fin.  
  
  ENDIF.

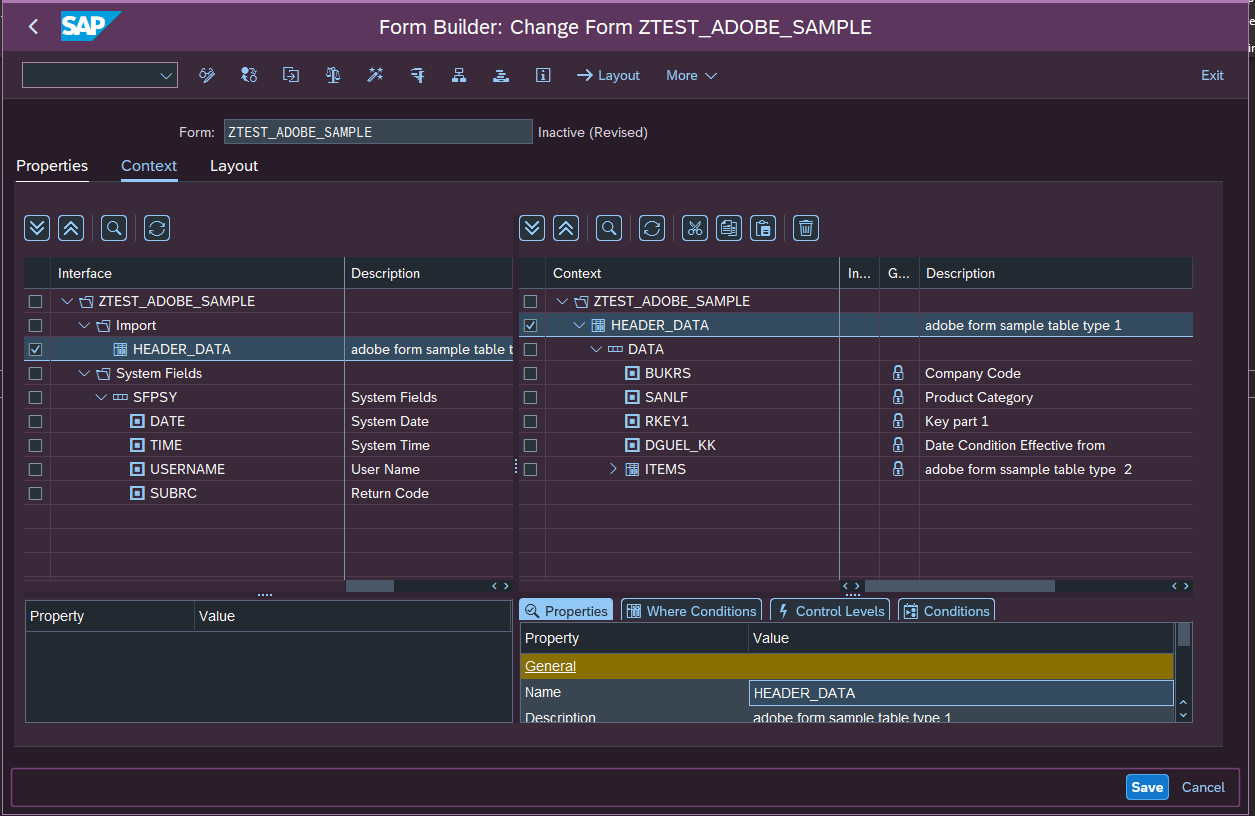
4. Create adobe form and assign earlier created adobe forms interface





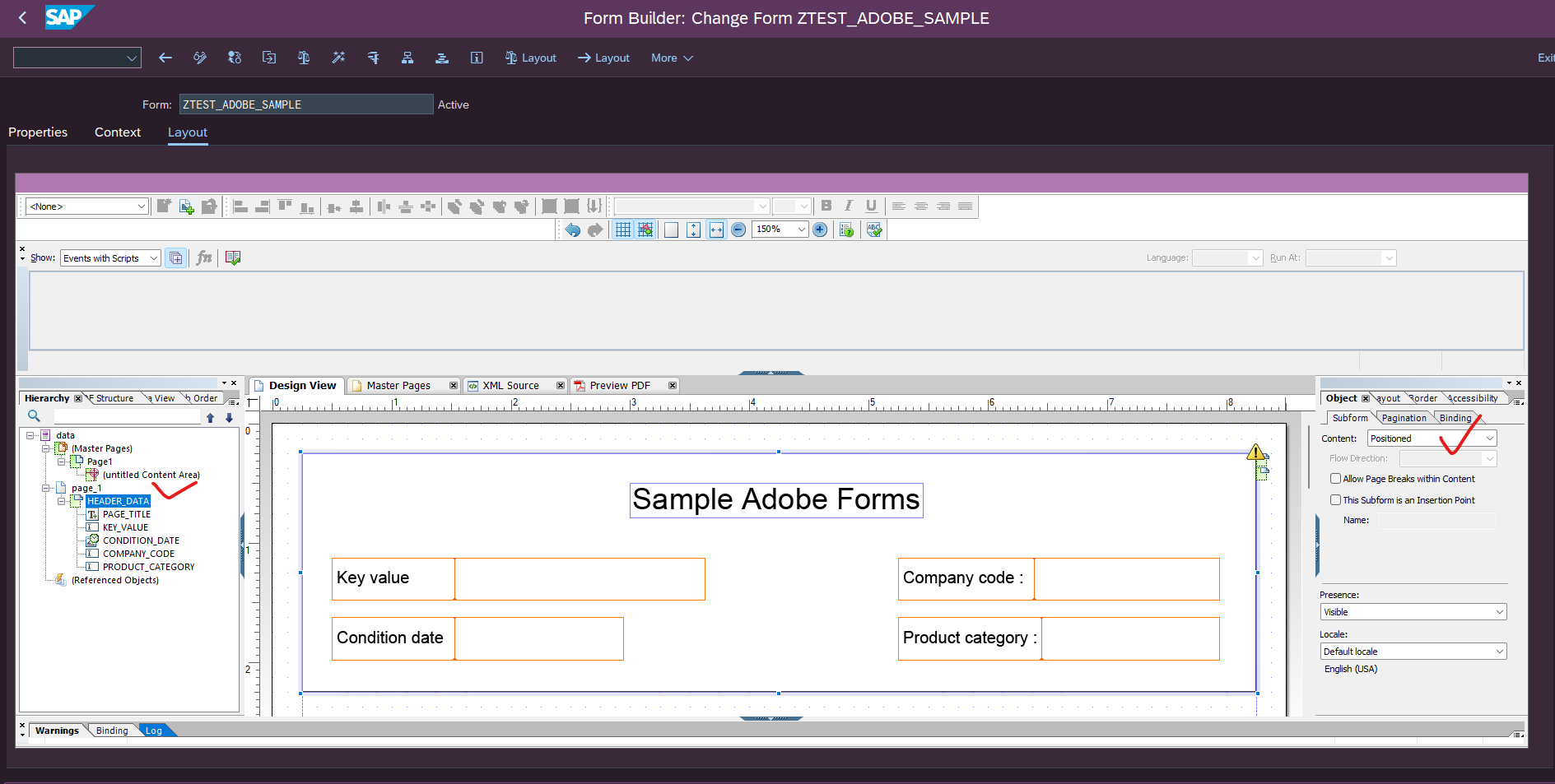
Drag and drop the HEADER\_DATA from INTERFACE side to CONTEXT side it may look as follows

Adobe forms configuring is ready next is layout design

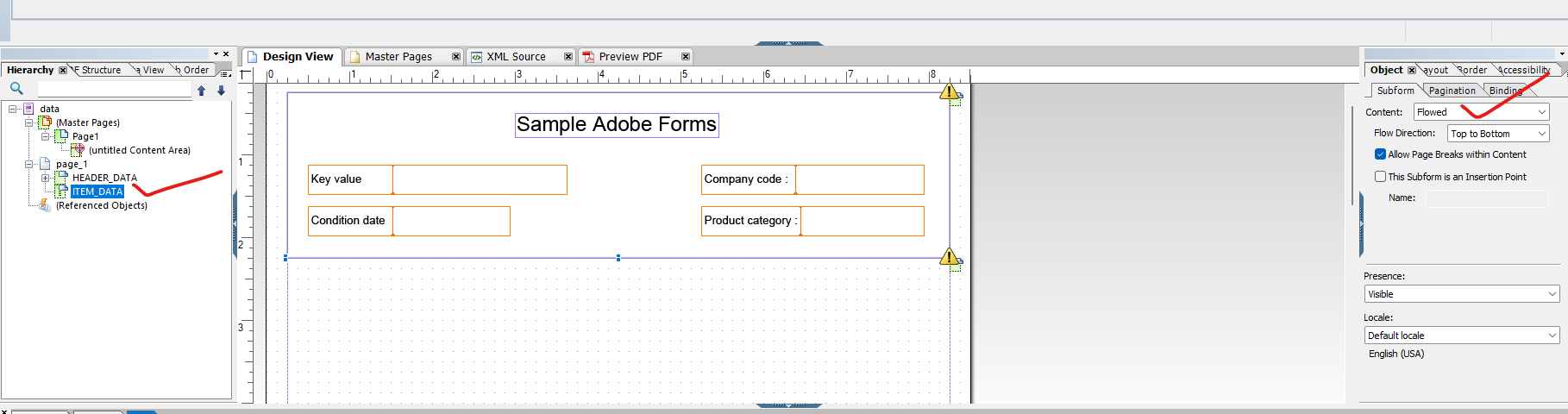


Select the layout tab and start designing

This layout contains subform with **SUBFORM** property – **POSITIONED** so all the text field can be freely placed any where within the layout as shown in this image.



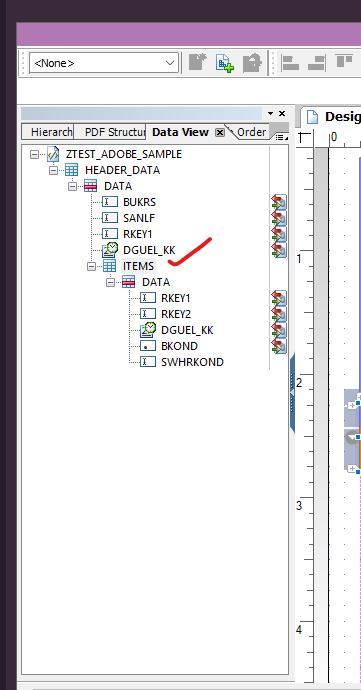
Create a subform with property set as FLOWED.



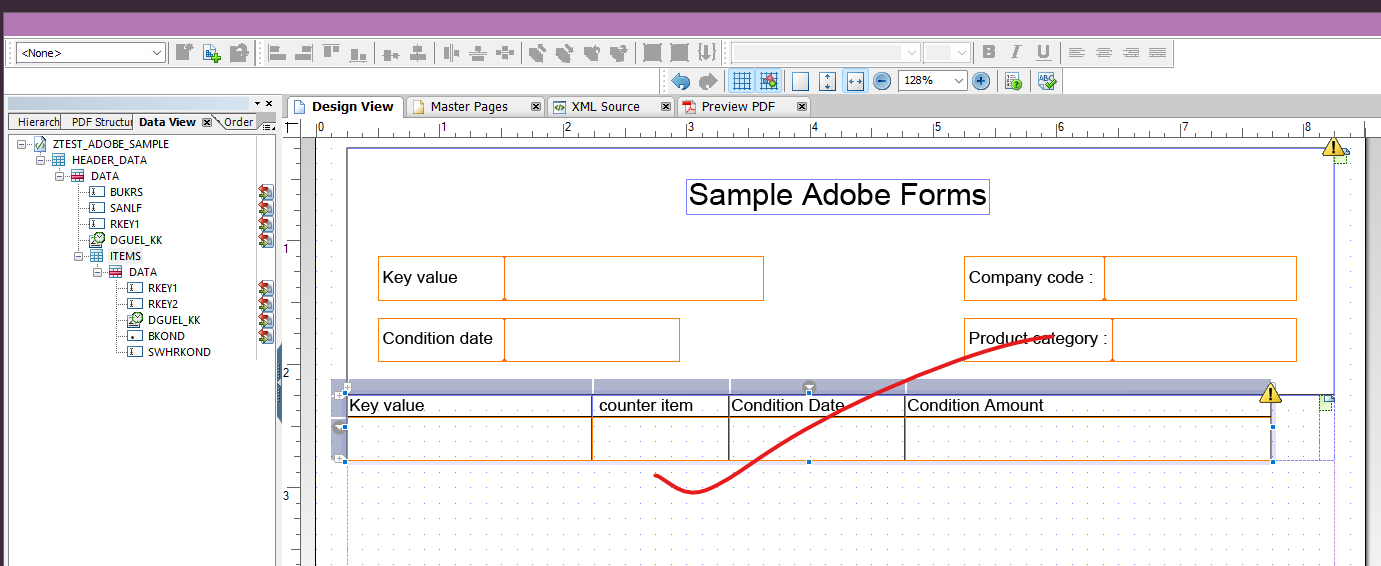
Objects created under **FLOWED** cannot be placed freely it acts like a **SEQUENCE OF STACK**

**This layout property helps to display line item data of a table to be displayed to next page.**

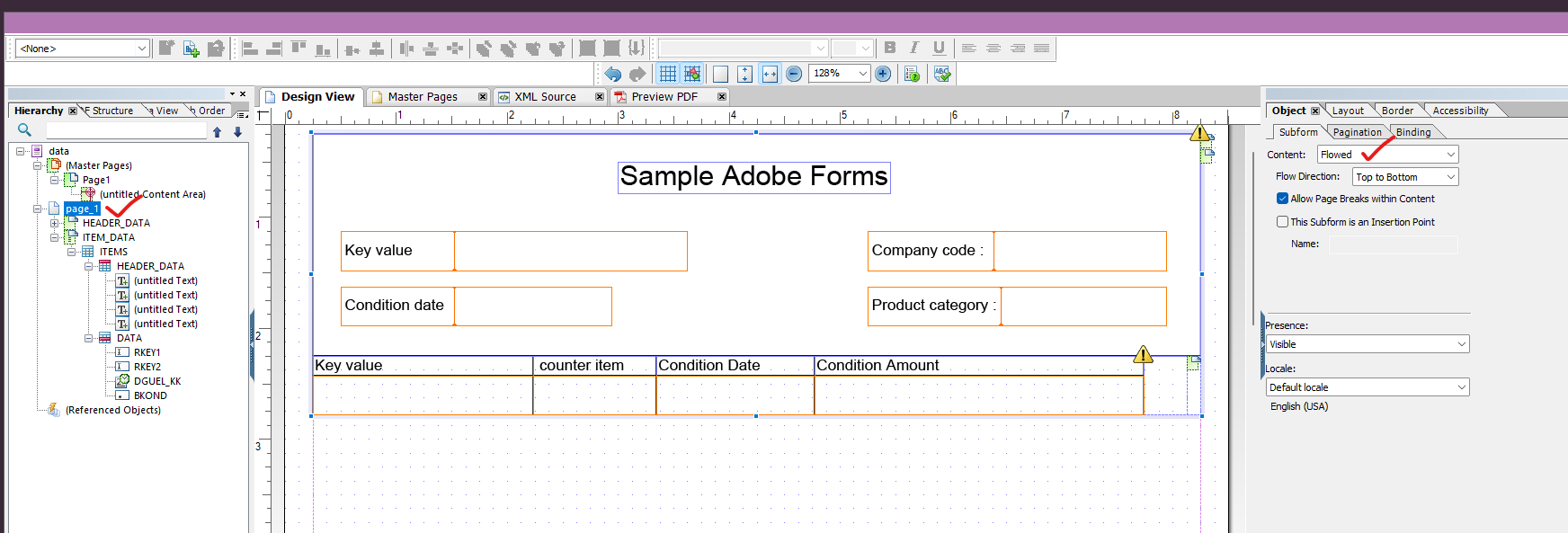
Next go to **DATA VIEW** tab and drag and drop the ITEM structure into the layout

****

After drag and drop it appears as follows

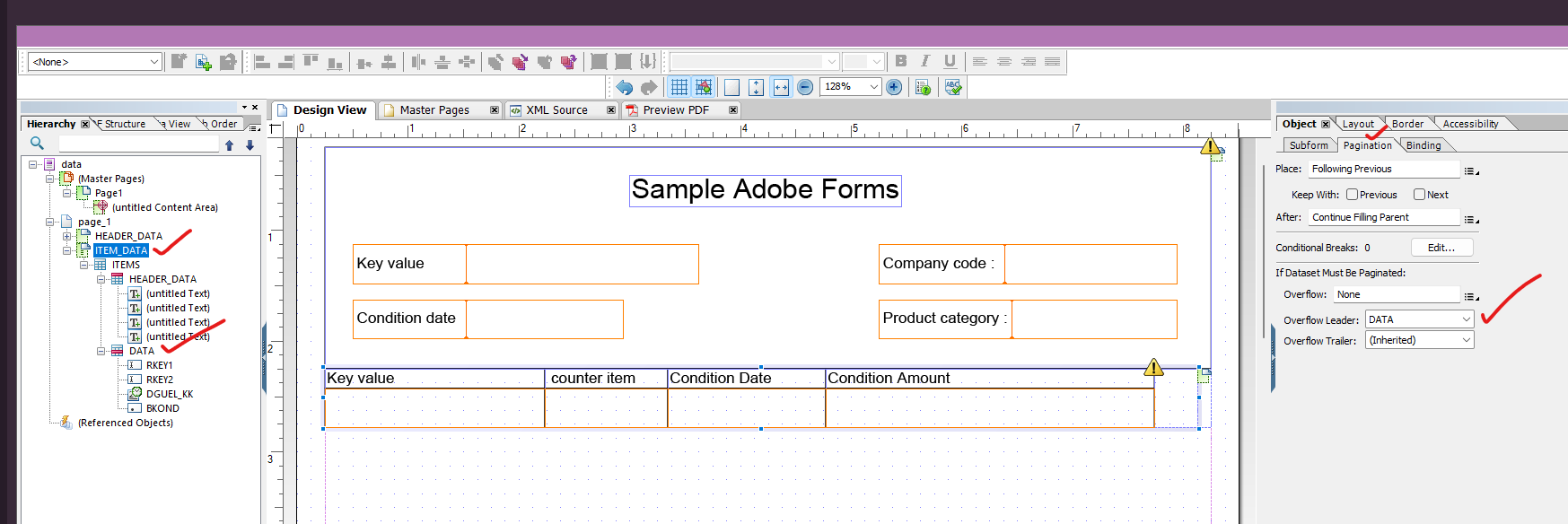


Now go to the **HIERARCHY** tab and select the **PAGE subform object** and set its subform property as **FLOWED**

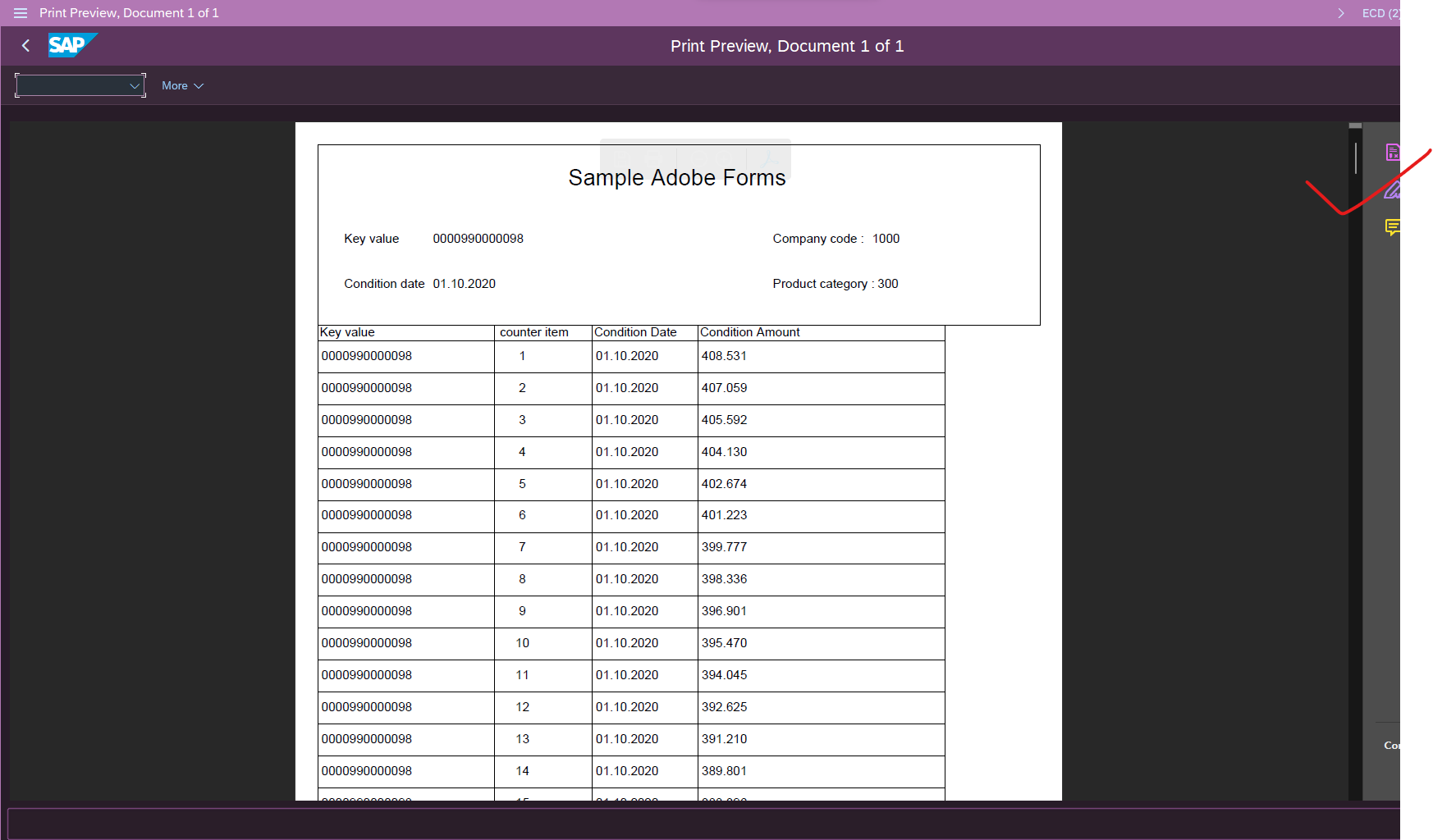


Now go to **ITEM\_DATA** sub form and select **PAGINATION** tab and select the Overflow leader property as

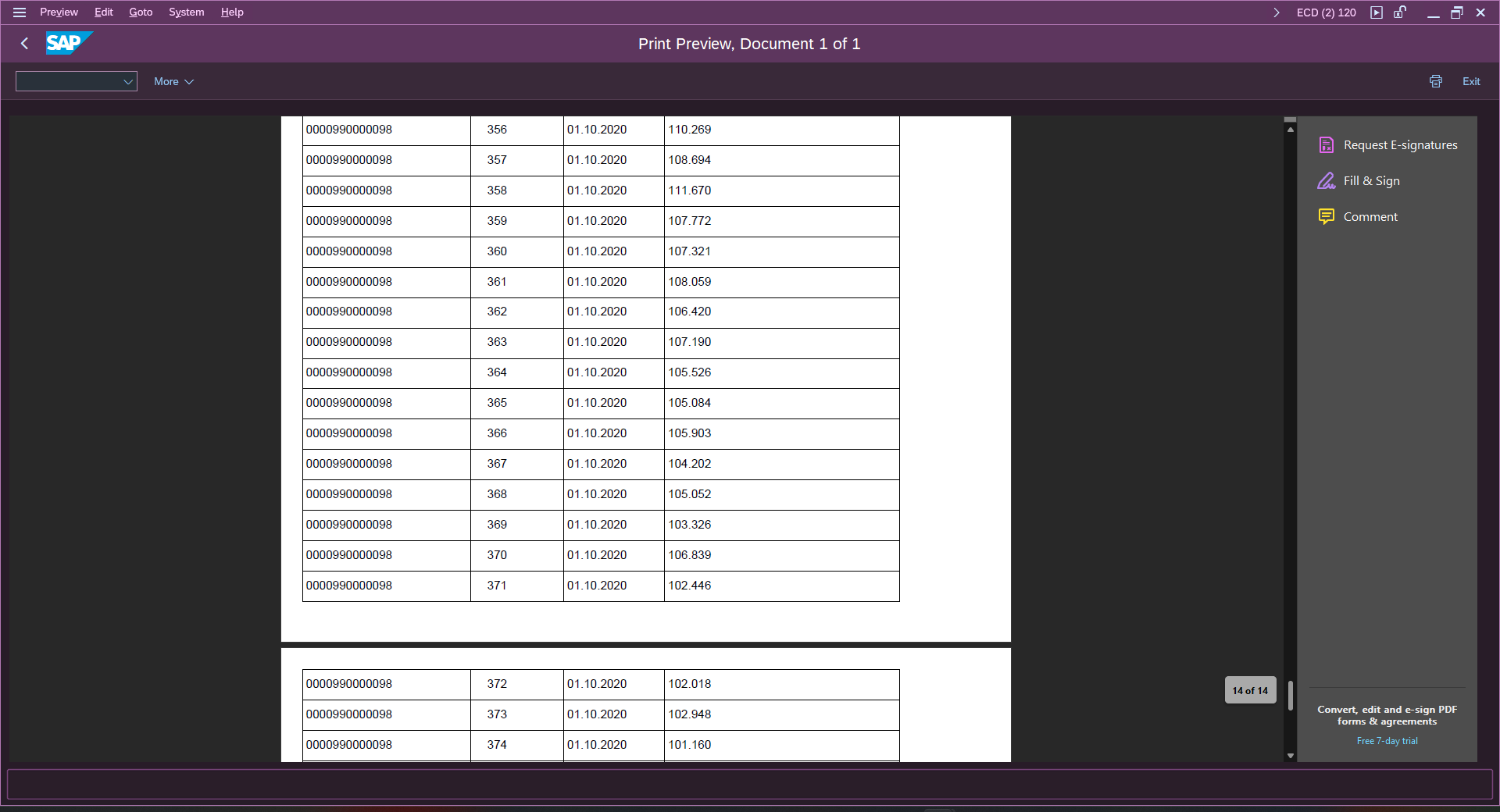
Table row item **(TABLE ROW ITEM will display only the line items in subsequent pages )**

**(TABLE ROW HEADER selection will display table header in every page)**

375 line items



This shows it has 14 pages displayed



SAMPLE PDF FILE ATTACHED



SAMPLE ADOBE FORM DESIGN FILE AND DATA SCHEMA FILES ATTACHED

