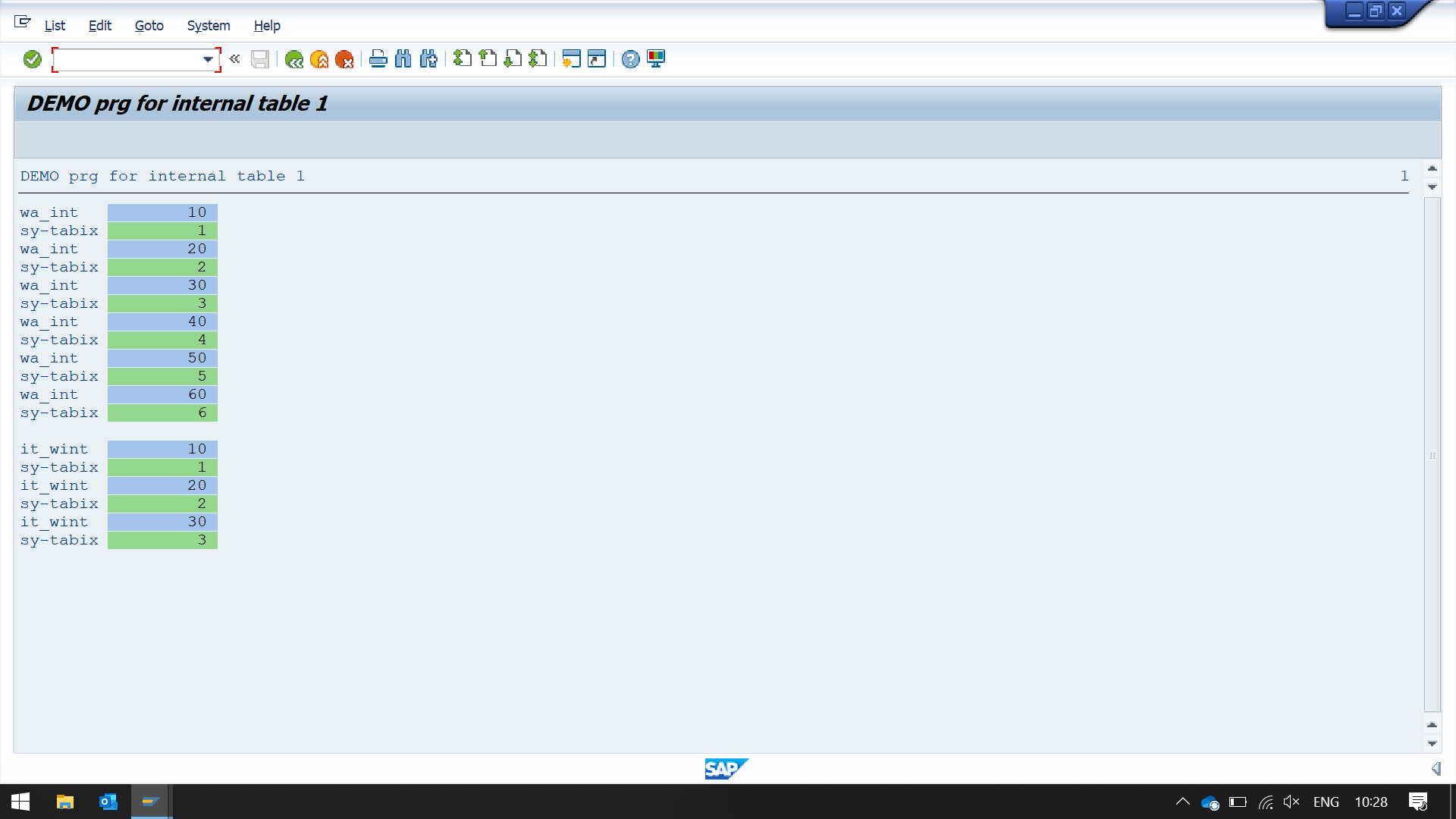
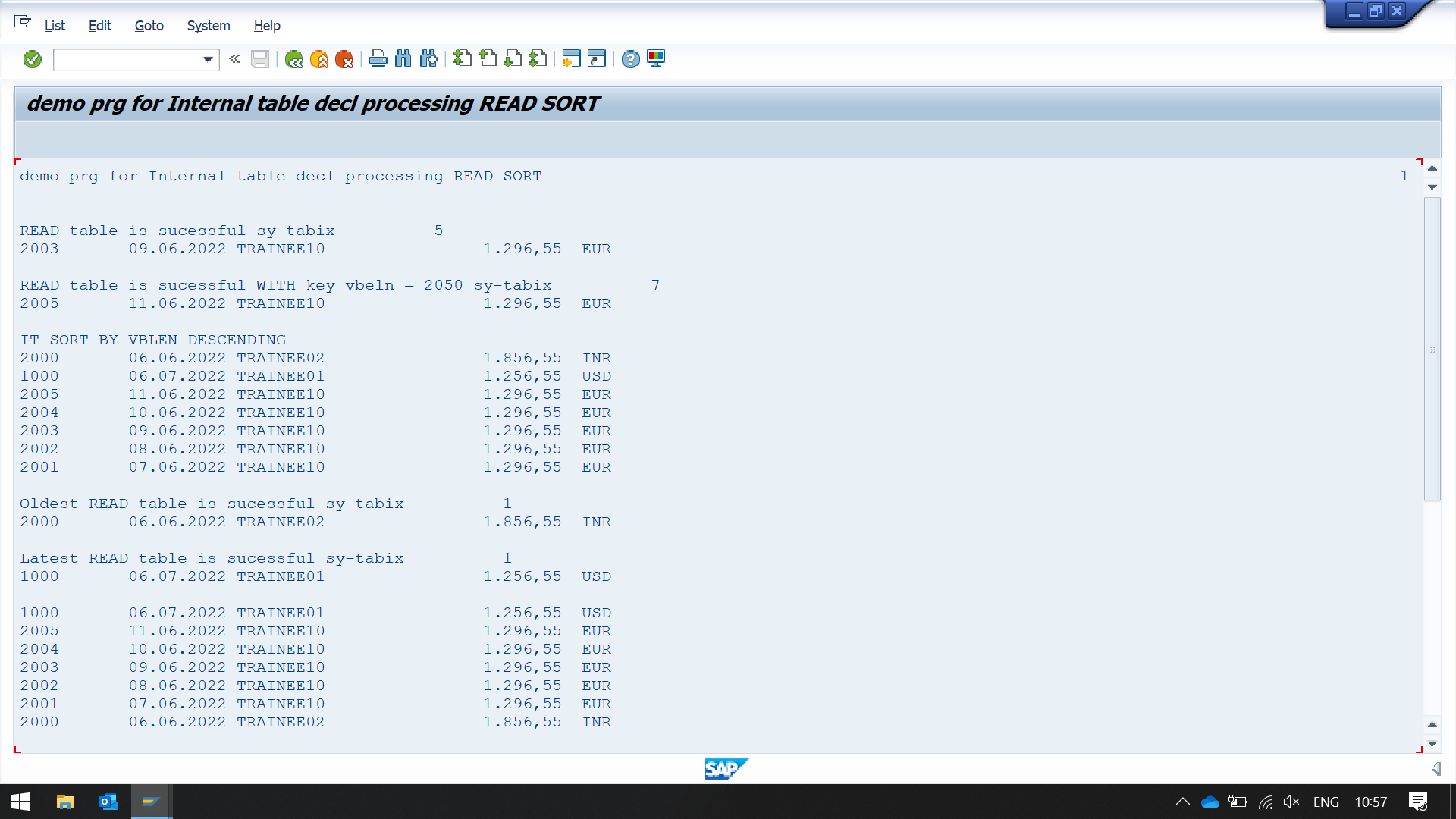
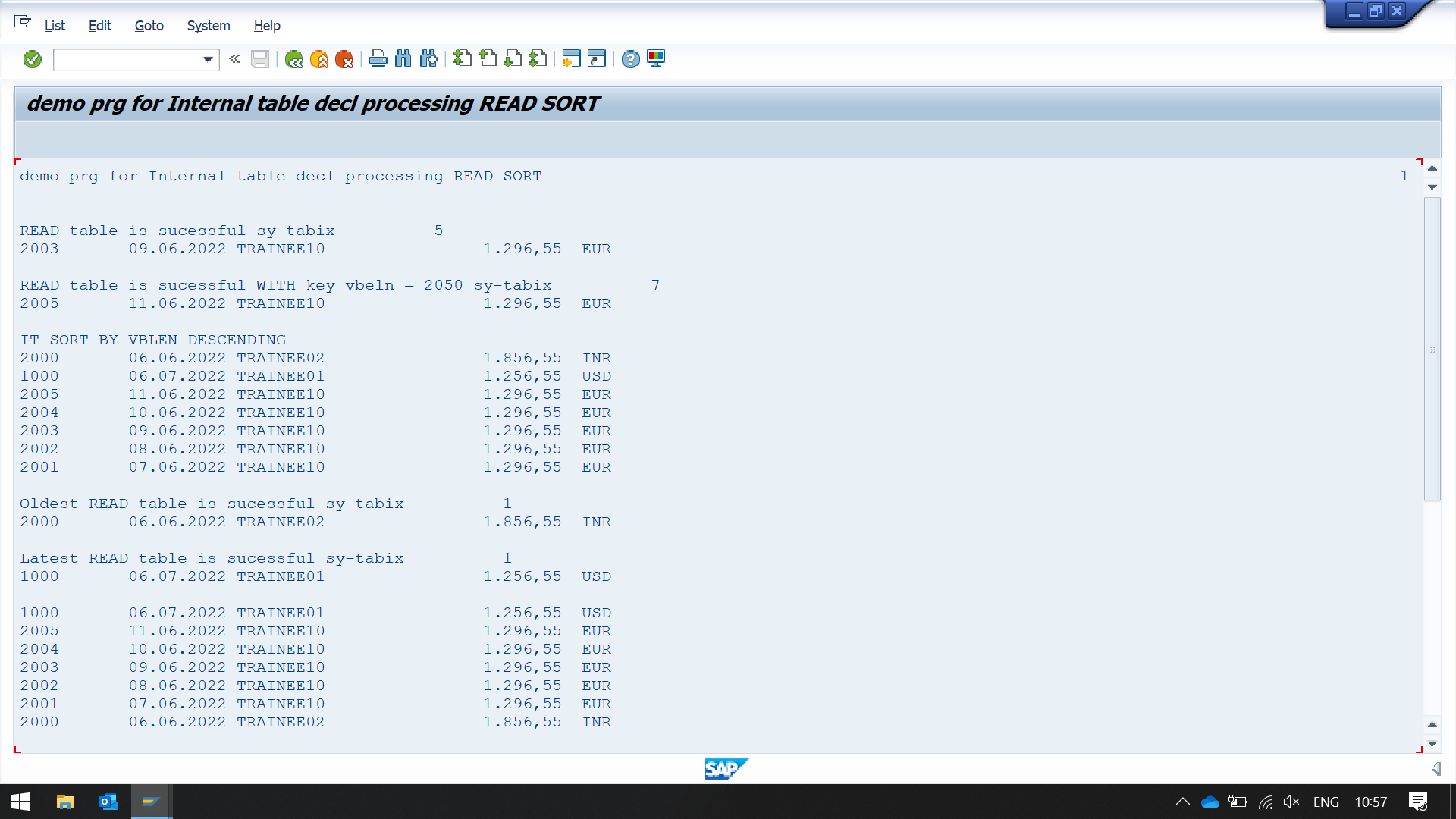
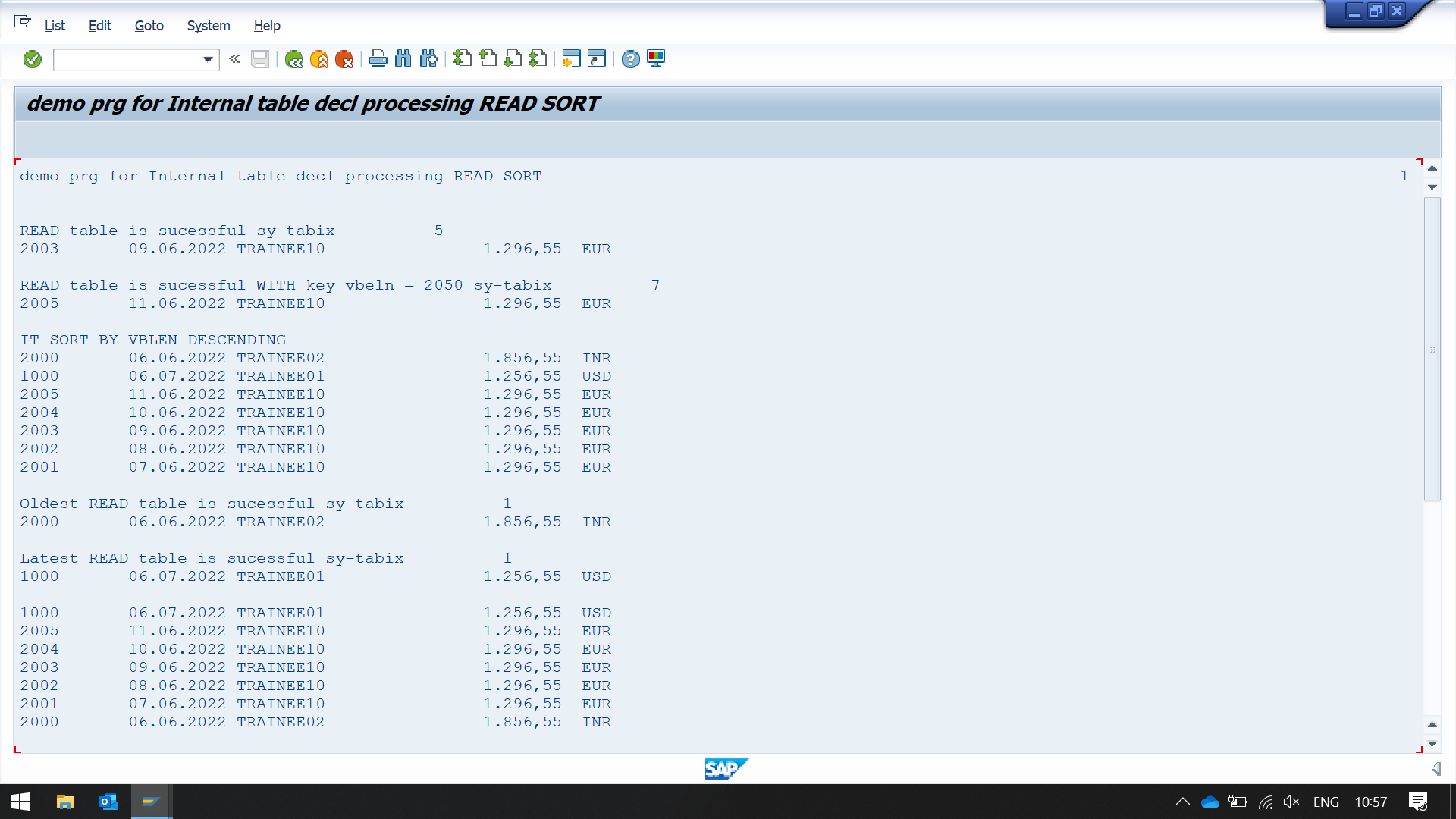
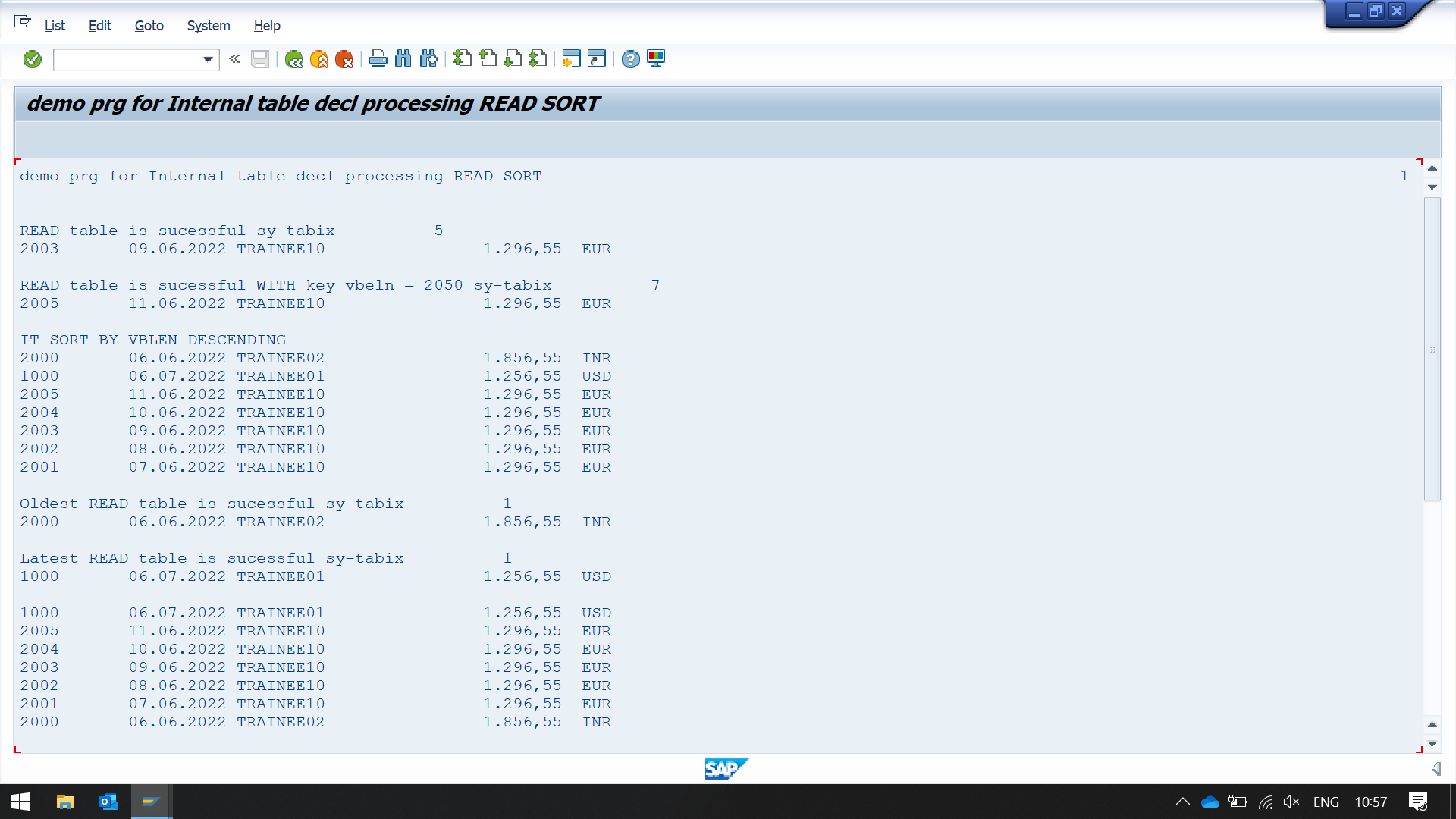
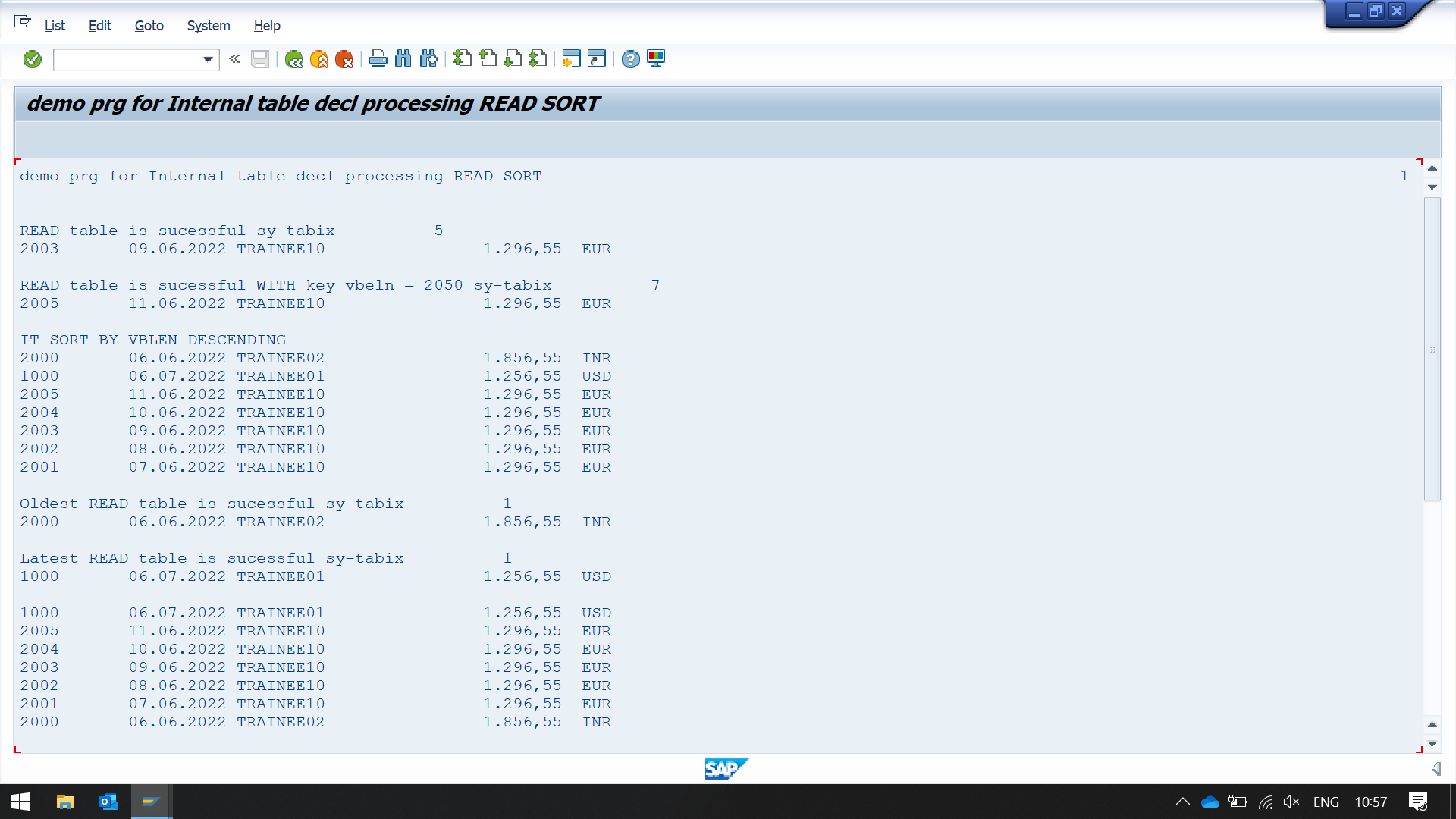
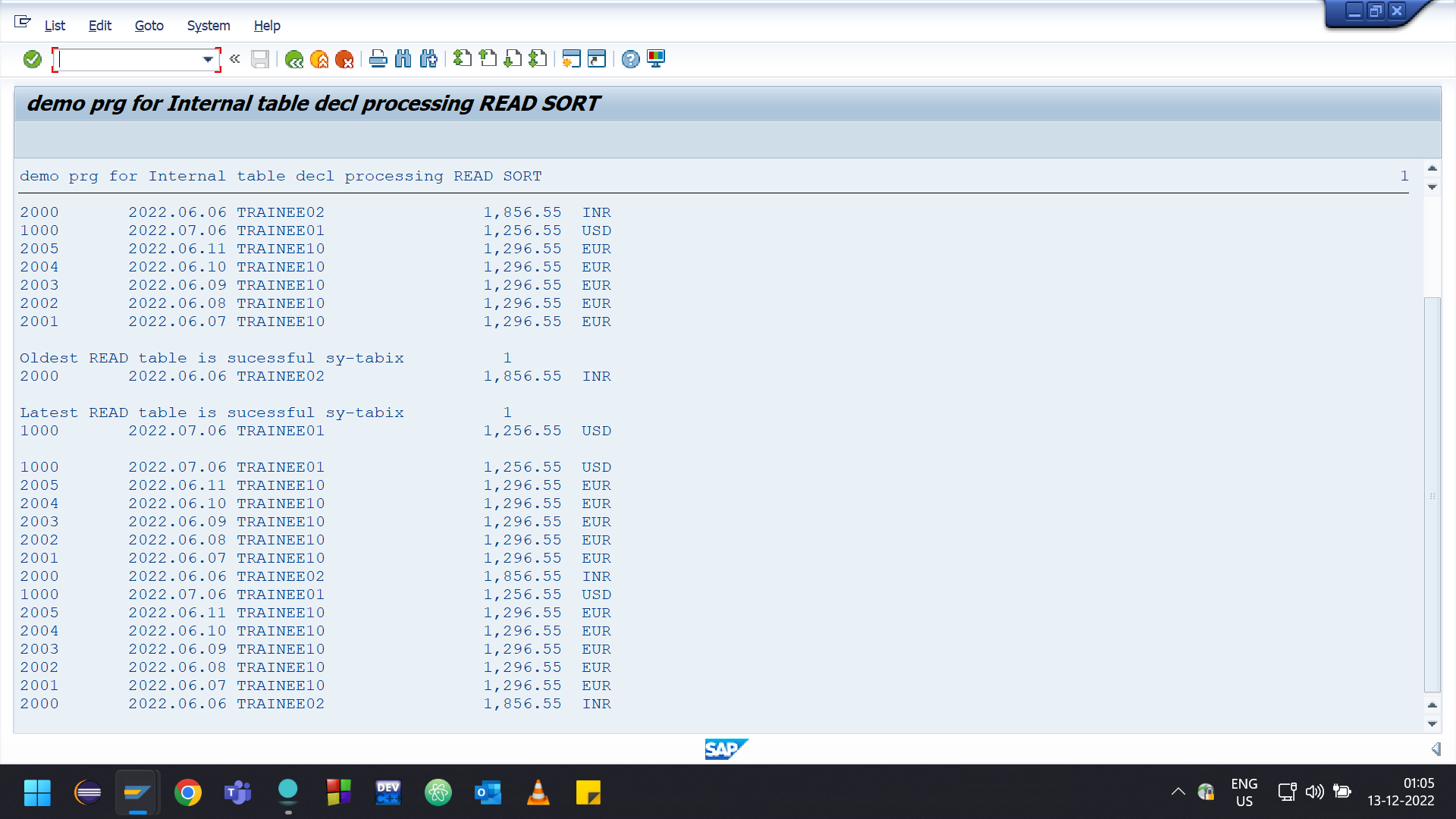
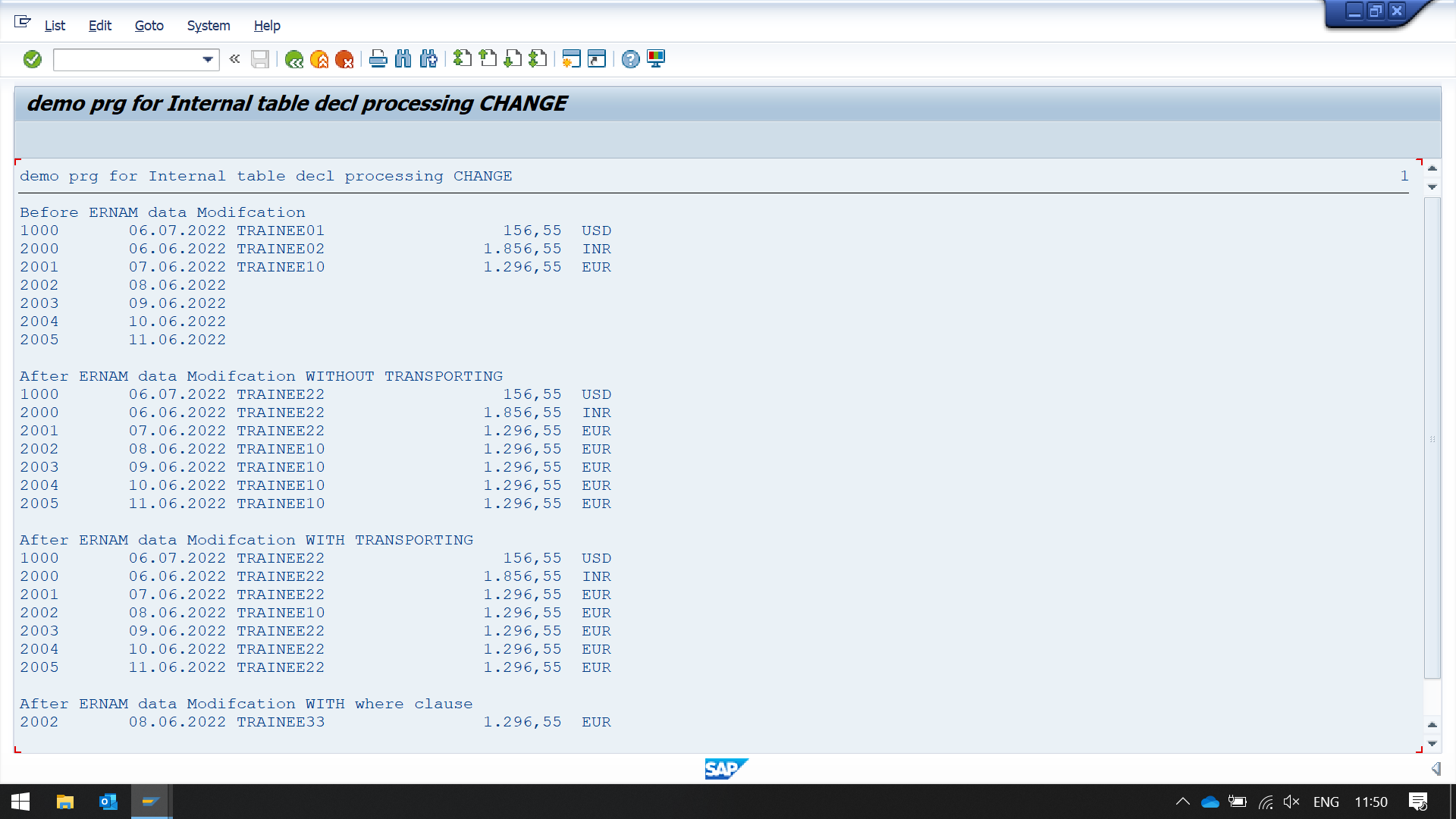
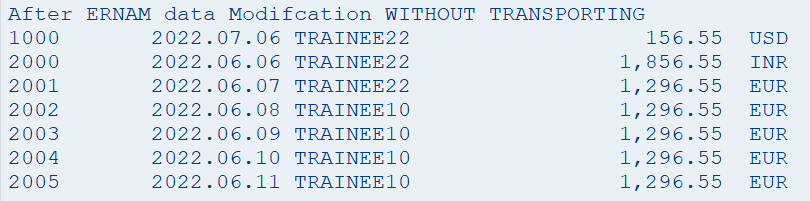
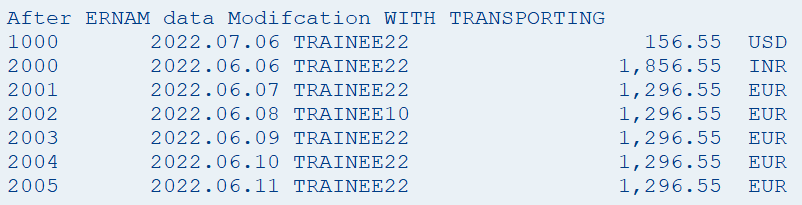
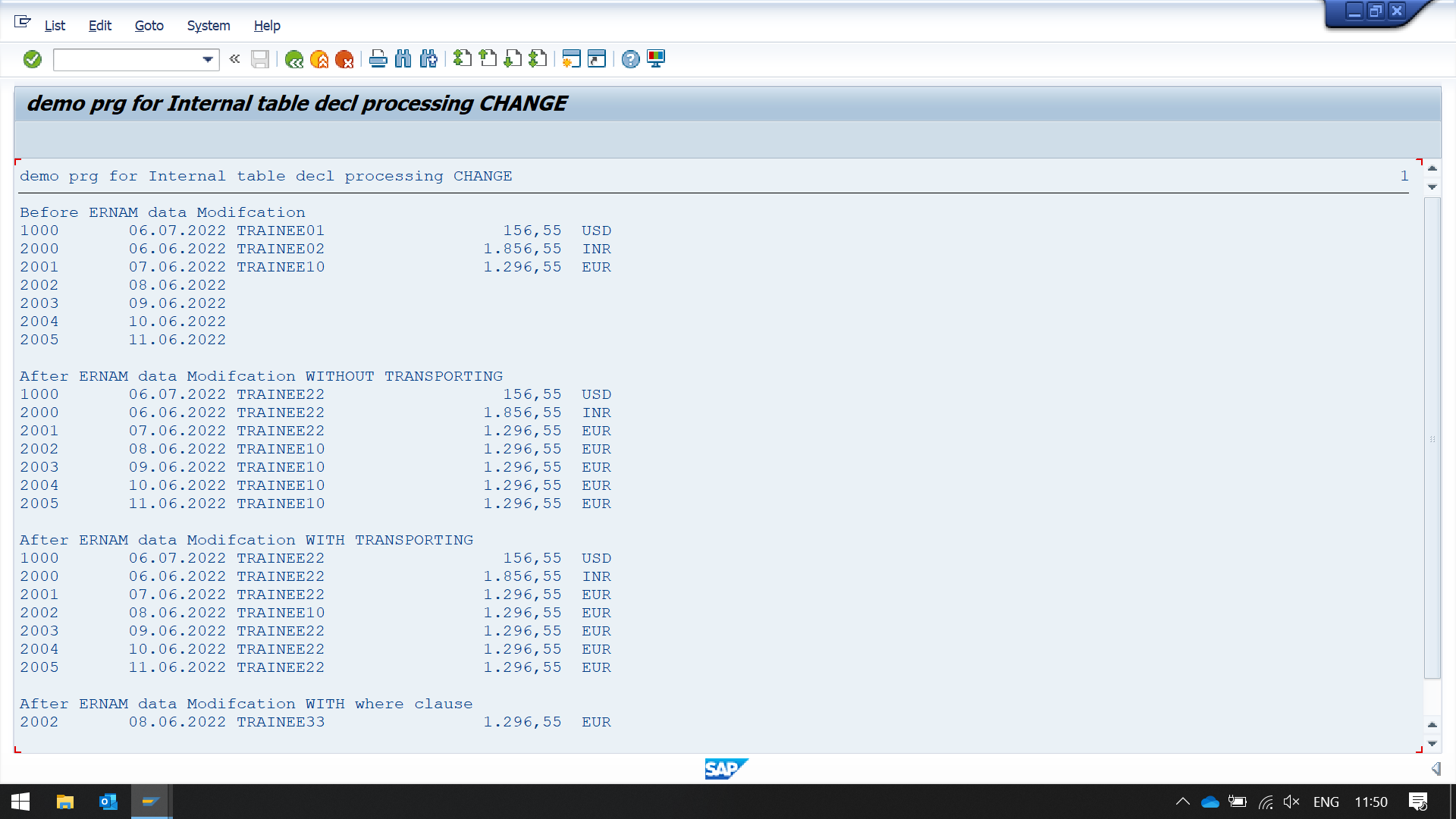
INTERNAL TABLE WITH HEADER, WITHOUT HEADER, APPEND LINES OF .. TO .., TRANSPORTING, MODIFY, DELETE, DELETE ADJACENT, sorted by, initial, standard table, collect, at first .. endat, at last .. endat, format color .., format off, sort by, at new .. endat, at end of .. endat, sum, on change of .. endon

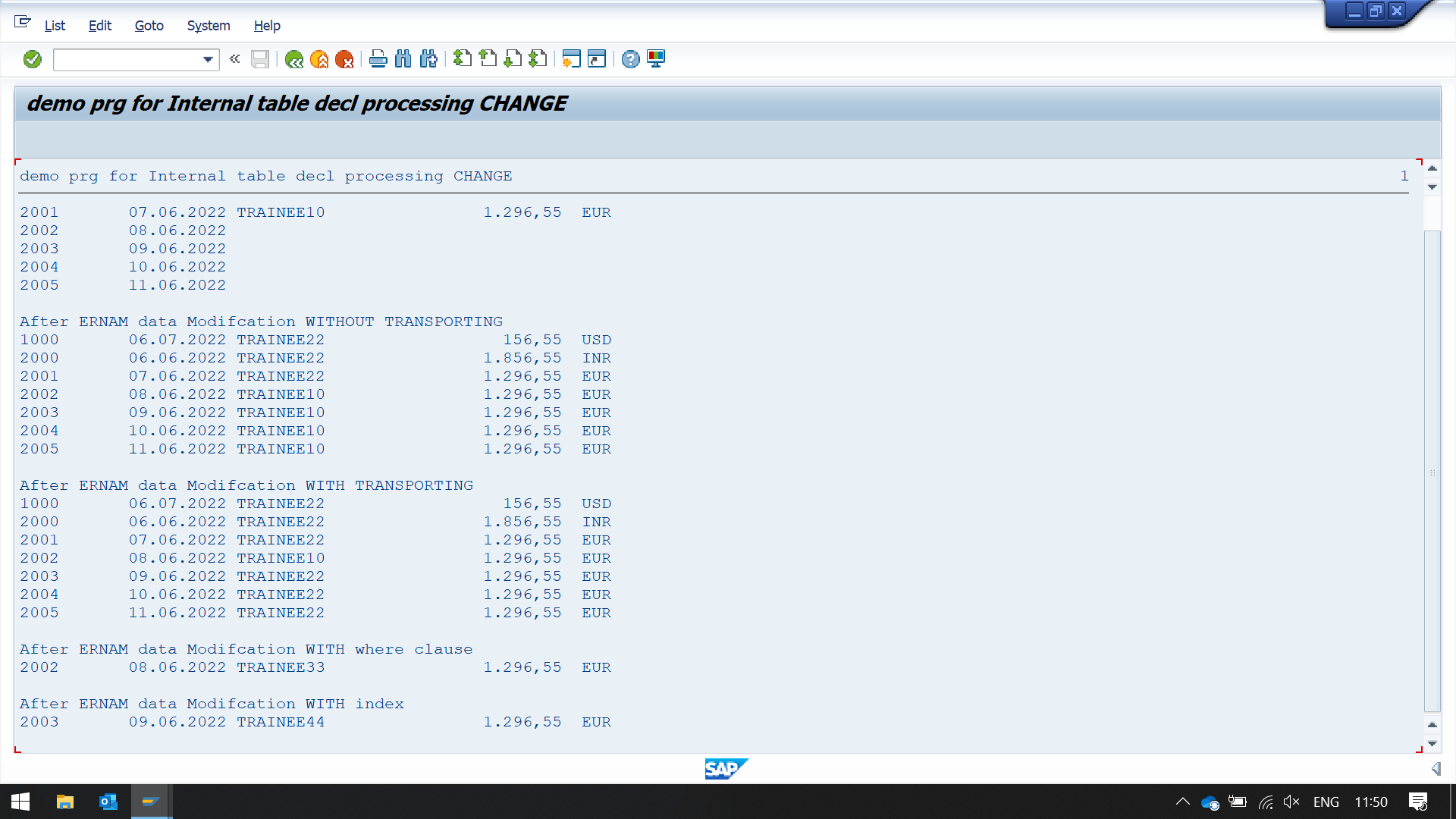
REPORT zyny\_demo\_int\_table\_01.  
  
\*\*\* ITAB declaration  
\*data: it\_int TYPE TABLE OF i.  
DATA: it\_int TYPE STANDARD TABLE OF i, " ITAB declaration without header line  
      wa\_int TYPE i. " work area  
  
DATA: it\_wint TYPE TABLE OF i WITH HEADER LINE.  
  
\*\*\* DATA population into IT\_INT using work area WA\_INT  
CLEAR : wa\_int.  
wa\_int = 10.  
APPEND wa\_int TO it\_int.  
CLEAR : wa\_int.  
wa\_int = 20.  
APPEND wa\_int TO it\_int.  
CLEAR : wa\_int.  
wa\_int = 30.  
APPEND wa\_int TO it\_int.  
CLEAR : wa\_int.  
wa\_int = 40.  
APPEND wa\_int TO it\_int.  
CLEAR : wa\_int.  
wa\_int = 50.  
APPEND wa\_int TO it\_int.  
CLEAR : wa\_int.  
wa\_int = 60.  
APPEND wa\_int TO it\_int.  
  
\*\*\*\* DATA processing of it\_int using work area wa\_int  
  
LOOP AT it\_int INTO wa\_int.  
  WRITE:/ 'wa\_int  ', wa\_int COLOR 4. " Printing the data line by line  
  WRITE:/ 'sy-tabix', sy-tabix COLOR 5." Current loop pass of IT\_INT  
ENDLOOP.  
  
SKIP.  
  
it\_wint = 10.  
APPEND it\_wint.  
it\_wint = 20.  
APPEND it\_wint.  
it\_wint = 30.  
APPEND it\_wint.  
  
LOOP AT it\_wint .  
  WRITE:/ 'it\_wint ', it\_wint COLOR 4. " Printing the data line by line  
  WRITE:/ 'sy-tabix', sy-tabix COLOR 5." Current loop pass of IT\_INT  
ENDLOOP.

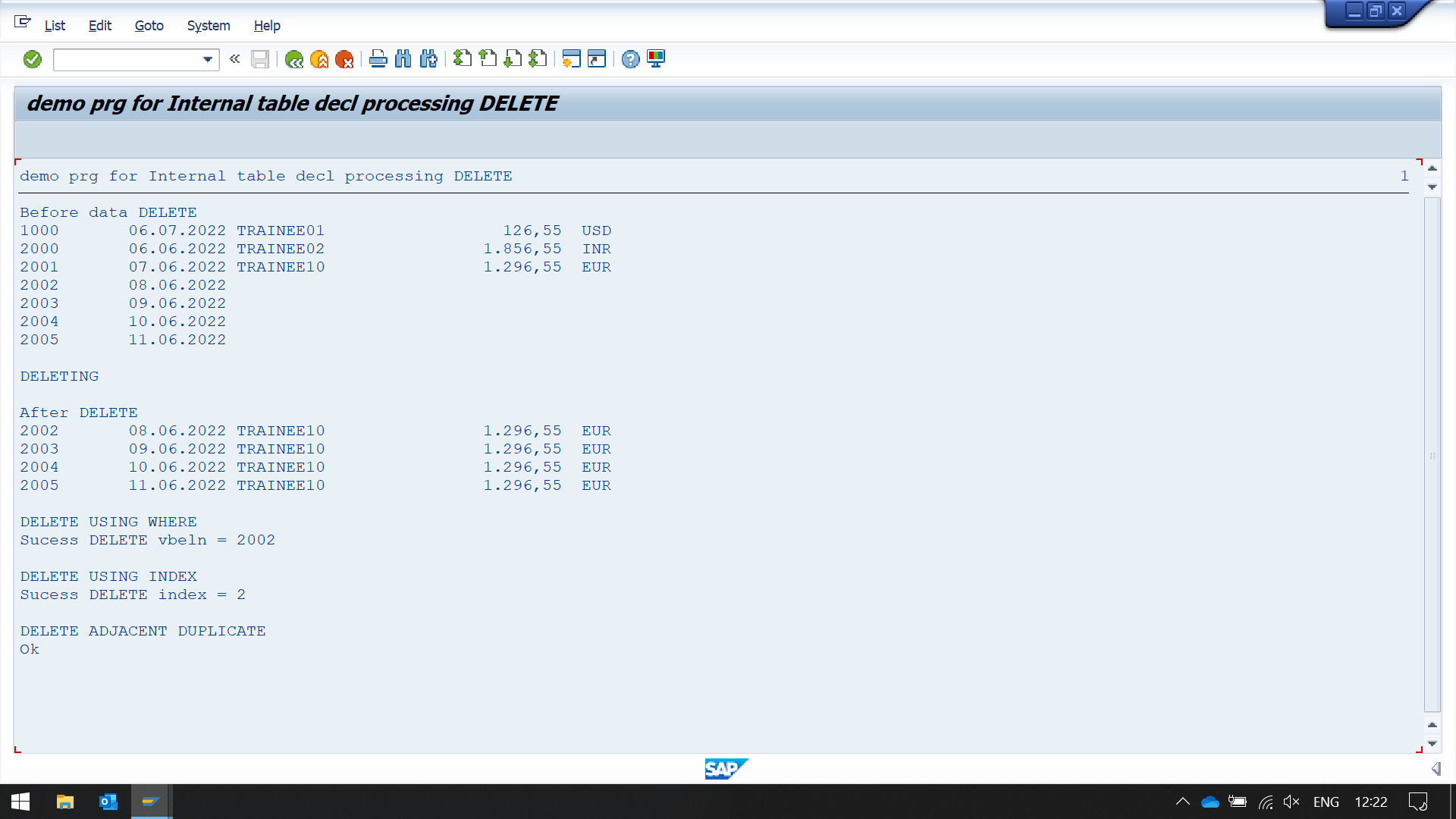
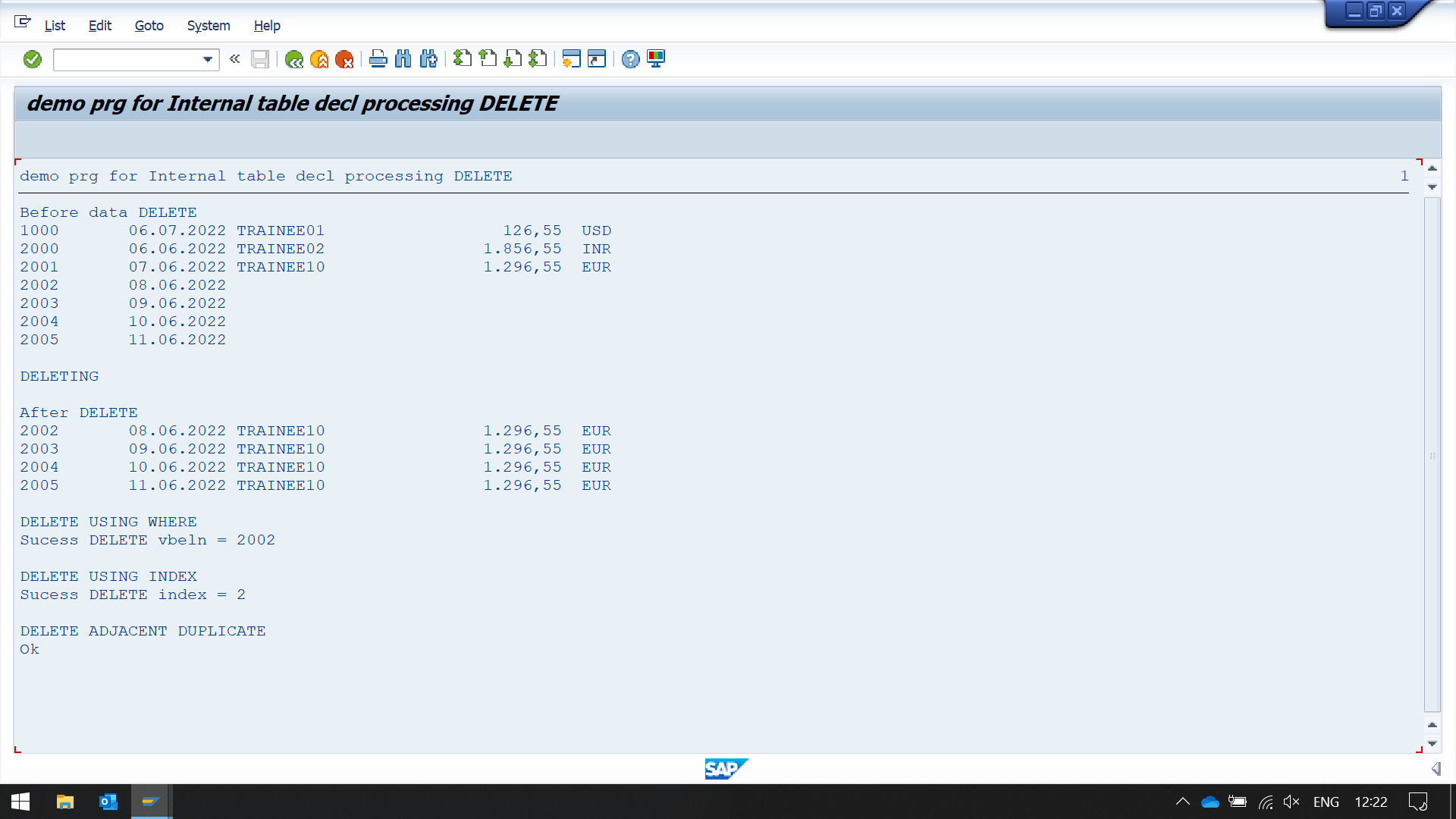
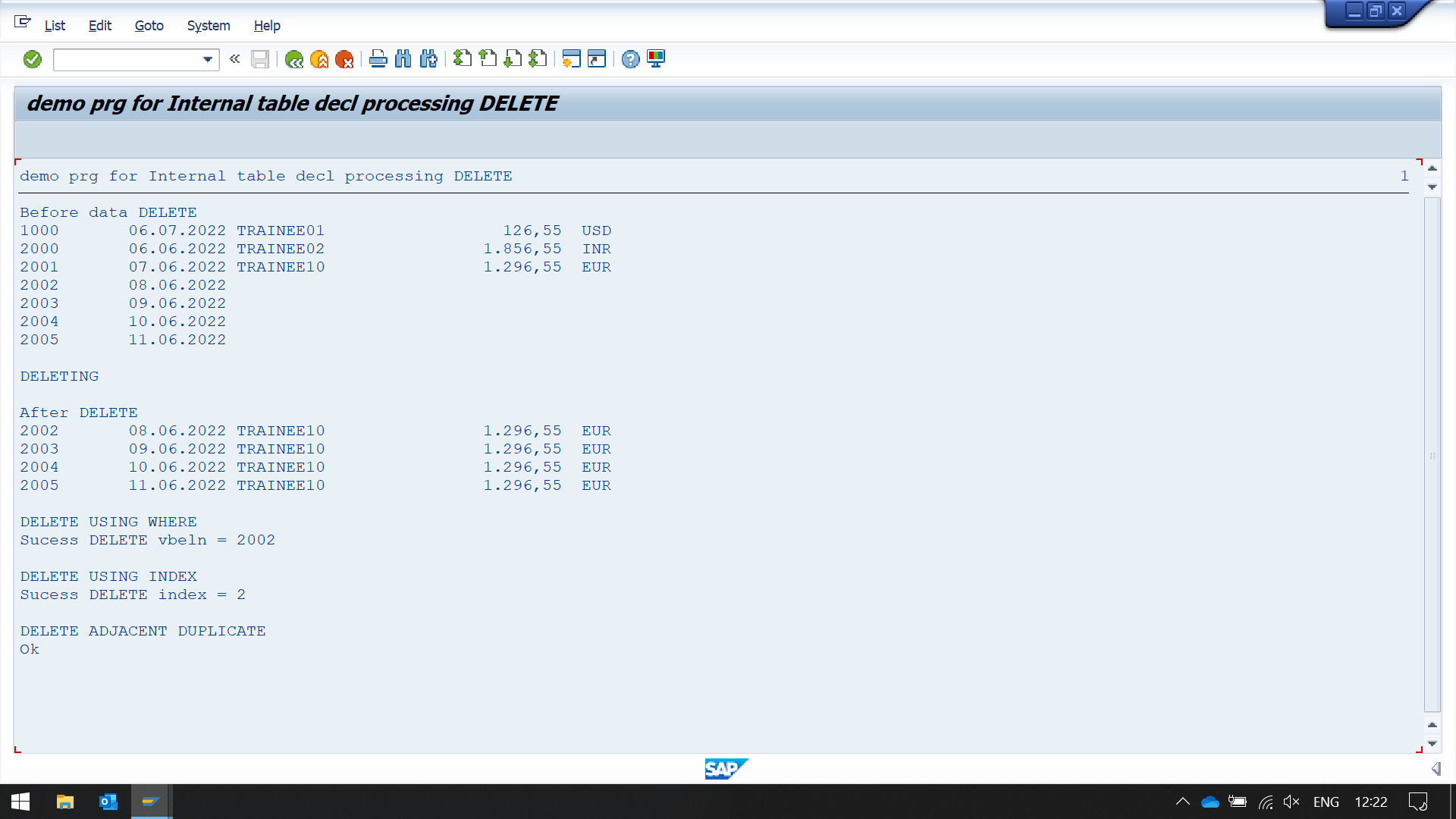


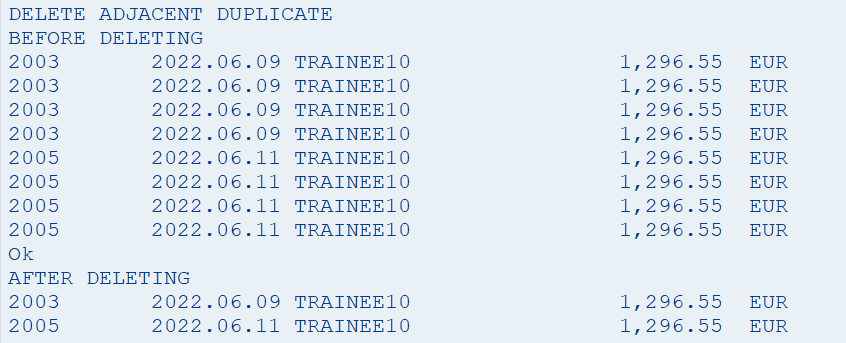
REPORT zyny\_demo\_int\_table\_04.  
  
\*\* Local user defined type  
TYPES: BEGIN OF gty\_tab,  
         vbeln      TYPE vbak-vbeln,  
         date\_crted TYPE erdat,  
         ernam      TYPE vbak-ernam,  
         auart      TYPE vbak-auart,  
         netwr      TYPE vbak-netwr,  
         waerk      TYPE vbak-waerk,  
       END OF gty\_tab.  
  
\*\*\*\* Internal table declaration  
DATA: it\_so TYPE STANDARD TABLE OF  gty\_tab INITIAL SIZE 0, " ITAB  
      wa\_so TYPE gty\_tab. " Header line or work area  
  
\*\*\*\* Internal table declaration  
DATA: it\_so\_new TYPE STANDARD TABLE OF  gty\_tab INITIAL SIZE 0, " ITAB  
      wa\_so\_new TYPE gty\_tab. " Header line or work area  
  
\*START-OF-SELECTION.  
\*\*\*\* DATA population  
CLEAR : wa\_so.  
wa\_so-vbeln = '1000'.  
wa\_so-date\_crted = '20220706'.  
wa\_so-ernam      = 'TRAINEE01'.  
wa\_so-auart    = 'RO'.  
wa\_so-netwr    = '1256.55'.  
wa\_so-waerk    = 'USD'.  
APPEND wa\_so TO it\_so.  
  
CLEAR : wa\_so.  
wa\_so-vbeln = '2000'.  
wa\_so-date\_crted = '20220606'.  
wa\_so-ernam      = 'TRAINEE02'.  
wa\_so-auart    = 'SO'.  
wa\_so-netwr    = '1856.55'.  
wa\_so-waerk    = 'INR'.  
APPEND wa\_so TO it\_so.  
  
DO 5 TIMES.  
\*  CLEAR : wa\_so.  
  wa\_so-vbeln = wa\_so-vbeln + 1 .  
  wa\_so-date\_crted = wa\_so-date\_crted + 1.  
  wa\_so-ernam      = 'TRAINEE10'.  
  wa\_so-auart    = 'SO'.  
  wa\_so-netwr    = '1296.55'.  
  wa\_so-waerk    = 'EUR'.  
  APPEND wa\_so TO it\_so.  
ENDDO.  
  
SKIP.  
  
\*\*\* data processing  
\*\*\*\*\* READ INTERNAL TABLE  
CLEAR wa\_so.  
READ TABLE it\_so INTO wa\_so INDEX 5. " reading table with index  
IF sy-subrc = 0.  
  WRITE:/ 'READ table is sucessful sy-tabix', sy-tabix.  
  WRITE:/ wa\_so-vbeln, wa\_so-date\_crted, wa\_so-ernam, wa\_so-netwr,wa\_so-waerk.  
ELSE.  
  WRITE:/ 'READ table is failed '.  
ENDIF.  
  
SKIP.  
  
\*\*\*\*\* READ INTERNAL TABLE  
CLEAR wa\_so.  
\*read TABLE it\_so INTO wa\_so WITH key vbeln = '     2050' .  
READ TABLE it\_so INTO wa\_so WITH KEY vbeln = 2005 . "reading table with keys  
IF sy-subrc = 0.  
  WRITE:/ 'READ table is sucessful WITH key vbeln = 2050 sy-tabix', sy-tabix.  
  WRITE:/ wa\_so-vbeln, wa\_so-date\_crted, wa\_so-ernam, wa\_so-netwr,wa\_so-waerk.  
ELSE.  
  WRITE:/ 'READ table is failed '.  
ENDIF.  
  
SKIP.  
  
\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\* SORT  
  
SORT it\_so BY vbeln DESCENDING. " sorting table in descending order, ascending order by default  
WRITE: / 'IT SORT BY VBLEN DESCENDING'.  
LOOP AT it\_so INTO wa\_so.  
  IF wa\_so-vbeln < 2050.  
    WRITE:/ wa\_so-vbeln, wa\_so-date\_crted, wa\_so-ernam, wa\_so-netwr,wa\_so-waerk.  
  ELSE.  
    WRITE:/ wa\_so-vbeln, wa\_so-date\_crted.  
  ENDIF.  
ENDLOOP.  
  
SKIP.  
  
\*\*\*\*\*\*  Oldest date  
SORT it\_so BY date\_crted.  
  
CLEAR wa\_so.  
READ TABLE it\_so INTO wa\_so INDEX 1.  
IF sy-subrc = 0.  
  WRITE:/ 'Oldest READ table is sucessful sy-tabix', sy-tabix.  
  WRITE:/ wa\_so-vbeln, wa\_so-date\_crted, wa\_so-ernam, wa\_so-netwr,wa\_so-waerk.  
ELSE.  
  WRITE:/ 'READ table is failed '.  
ENDIF.  
  
SKIP.  
  
\*\*\*\*\*\*  Latest date  
SORT it\_so BY date\_crted DESCENDING.  
  
CLEAR wa\_so.  
READ TABLE it\_so INTO wa\_so INDEX 1.  
IF sy-subrc = 0.  
  WRITE:/ 'Latest READ table is sucessful sy-tabix', sy-tabix.  
  WRITE:/ wa\_so-vbeln, wa\_so-date\_crted, wa\_so-ernam, wa\_so-netwr,wa\_so-waerk.  
ELSE.  
  WRITE:/ 'READ table is failed '.  
ENDIF.  
  
SKIP.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\* APPEND LINE OF  
  
APPEND LINES OF it\_so TO it\_so\_new. " table1 is appended on table2, it is also not sorted by default  
  
APPEND LINES OF it\_so TO it\_so\_new.  
APPEND LINES OF it\_so TO it\_so\_new.  
  
LOOP AT it\_so\_new INTO wa\_so\_new.  
  IF wa\_so\_new-vbeln < 2050.  
    WRITE:/ wa\_so\_new-vbeln, wa\_so\_new-date\_crted, wa\_so\_new-ernam, wa\_so\_new-netwr,wa\_so\_new-waerk.  
  ELSE.  
    WRITE:/ wa\_so\_new-vbeln, wa\_so\_new-date\_crted.  
  ENDIF.  
ENDLOOP.  
  
SKIP.



REPORT zyny\_demo\_int\_table\_05.  
  
\*\* Local user defined type  
TYPES: BEGIN OF gty\_tab,  
         vbeln      TYPE vbak-vbeln,  
         date\_crted TYPE erdat,  
         ernam      TYPE vbak-ernam,  
         auart      TYPE vbak-auart,  
         netwr      TYPE vbak-netwr,  
         waerk      TYPE vbak-waerk,  
       END OF gty\_tab.  
  
\*\*\*\* Internal table declaration  
DATA: it\_so TYPE STANDARD TABLE OF  gty\_tab INITIAL SIZE 0, " ITAB  
      wa\_so TYPE gty\_tab. " Header line or work area  
  
\*\*\*\* Internal table declaration  
DATA: it\_so\_new TYPE STANDARD TABLE OF  gty\_tab INITIAL SIZE 0, " ITAB  
      wa\_so\_new TYPE gty\_tab. " Header line or work area  
  
\*START-OF-SELECTION.  
\*\*\*\* DATA population  
CLEAR : wa\_so.  
wa\_so-vbeln = '1000'.  
wa\_so-date\_crted = '20220706'.  
wa\_so-ernam      = 'TRAINEE01'.  
wa\_so-auart    = 'RO'.  
wa\_so-netwr    = '156.55'.  
wa\_so-waerk    = 'USD'.  
APPEND wa\_so TO it\_so.  
  
CLEAR : wa\_so.  
wa\_so-vbeln = '2000'.  
wa\_so-date\_crted = '20220606'.  
wa\_so-ernam      = 'TRAINEE02'.  
wa\_so-auart    = 'SO'.  
wa\_so-netwr    = '1856.55'.  
wa\_so-waerk    = 'INR'.  
APPEND wa\_so TO it\_so.  
  
DO 5 TIMES.  
  wa\_so-vbeln = wa\_so-vbeln + 1 .  
  wa\_so-date\_crted = wa\_so-date\_crted + 1.  
  wa\_so-ernam      = 'TRAINEE10'.  
  wa\_so-auart    = 'SO'.  
  wa\_so-netwr    = '1296.55'.  
  wa\_so-waerk    = 'EUR'.  
  APPEND wa\_so TO it\_so.  
ENDDO.  
  
\*\*\* data processing  
WRITE:/ 'Before ERNAM data Modifcation'.  
LOOP AT it\_so INTO wa\_so.  
  IF wa\_so-vbeln < 2002.  
    WRITE:/ wa\_so-vbeln, wa\_so-date\_crted, wa\_so-ernam, wa\_so-netwr,wa\_so-waerk.  
  ELSE.  
    WRITE:/ wa\_so-vbeln, wa\_so-date\_crted.  
  ENDIF.  
ENDLOOP.  
  
SKIP.  
  
\*\*\* WITHOUT TRANSPORTING  
LOOP AT it\_so INTO wa\_so.  
  IF wa\_so-vbeln < 2002.  
    wa\_so-ernam      = 'TRAINEE22'.  
    MODIFY it\_so FROM wa\_so. " modify internal table from work area  
    clear wa\_so.  
  ELSE.  
  ENDIF.  
ENDLOOP.  
  
WRITE:/ 'After ERNAM data Modifcation WITHOUT TRANSPORTING'.  
LOOP AT it\_so INTO wa\_so.  
  WRITE:/ wa\_so-vbeln, wa\_so-date\_crted, wa\_so-ernam, wa\_so-netwr,wa\_so-waerk.  
ENDLOOP.  
  
SKIP.  
  
\*\* DATA CHANGING  
LOOP AT it\_so INTO wa\_so.  
  IF wa\_so-vbeln > 2002.  
    wa\_so-ernam      = 'TRAINEE22'.  
    MODIFY it\_so FROM wa\_so TRANSPORTING ernam. " due to transporting it will only take ernam field from work area The addition TRANSPORTING has the effect that only the specified components [comp1](sapevent:ABENITAB_COMPONENTS), [comp2](sapevent:ABENITAB_COMPONENTS), ... of the work area are assigned to the corresponding components of the row(s) to be changed. For [sorted tables](sapevent:ABENSORTED_TABLE_GLOSRY) and [hashed tables](sapevent:ABENHASHED_TABLE_GLOSRY), no primary table key components may be declared after TRANSPORTING.  
    clear wa\_so.  
  ELSE.  
  ENDIF.  
ENDLOOP.  
  
WRITE:/ 'After ERNAM data Modifcation WITH TRANSPORTING'.  
LOOP AT it\_so INTO wa\_so.  
  WRITE:/ wa\_so-vbeln, wa\_so-date\_crted, wa\_so-ernam, wa\_so-netwr,wa\_so-waerk.  
ENDLOOP.  
  
SKIP.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\* MODIFICATION USING WHERE  
wa\_so-ernam      = 'TRAINEE33'.  
MODIFY it\_so FROM wa\_so TRANSPORTING ernam WHERE vbeln = 2002. " transporting statement condition with where condition  
IF sy-subrc = 0.  
  CLEAR wa\_so.  
  READ TABLE it\_so INTO wa\_so WITH KEY vbeln = 2002.  
  IF sy-subrc = 0.  
    WRITE:/ 'After ERNAM data Modifcation WITH where clause'.  
    WRITE:/ wa\_so-vbeln, wa\_so-date\_crted, wa\_so-ernam, wa\_so-netwr,wa\_so-waerk.  
  ENDIF.  
ELSE.  
  WRITE:/'FAILED', 'MODIFY it\_so FROM wa\_so TRANSPORTING ernam WHERE vbeln = ''2002"'.  
ENDIF.  
  
SKIP.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\* MODIFICATION USING index  
wa\_so-ernam      = 'TRAINEE44'.  
MODIFY it\_so FROM wa\_so INDEX 5 TRANSPORTING ernam. " modify statement with index and transporting  
IF sy-subrc = 0.  
  CLEAR wa\_so.  
  READ TABLE it\_so INTO wa\_so INDEX 5.  
  IF sy-subrc = 0.  
    WRITE:/ 'After ERNAM data Modifcation WITH index'.  
    WRITE:/ wa\_so-vbeln, wa\_so-date\_crted, wa\_so-ernam, wa\_so-netwr,wa\_so-waerk.  
  ENDIF.  
ELSE.  
  WRITE:/'FAILED', 'MODIFY it\_so FROM wa\_so TRANSPORTING ernam index = ''102"'.  
ENDIF.  
  
SKIP.



REPORT ZYNY\_DEMO\_INT\_TABLE\_06.  
  
\*\* Local user defined type  
TYPES: BEGIN OF GTY\_TAB,  
         VBELN      TYPE VBAK-VBELN,  
         DATE\_CRTED TYPE ERDAT,  
         ERNAM      TYPE VBAK-ERNAM,  
         AUART      TYPE VBAK-AUART,  
         NETWR      TYPE VBAK-NETWR,  
         WAERK      TYPE VBAK-WAERK,  
       END OF GTY\_TAB.  
  
\*\*\*\* Internal table declaration  
DATA: IT\_SO     TYPE STANDARD TABLE OF  GTY\_TAB INITIAL SIZE 0, " ITAB  
      WA\_SO     TYPE GTY\_TAB, " Header line or work area  
      IT\_SO\_NEW TYPE STANDARD TABLE OF  GTY\_TAB INITIAL SIZE 0, " ITAB  
      WA\_SO\_NEW TYPE GTY\_TAB. " Header line or work area  
  
\*START-OF-SELECTION.  
\*\*\*\* DATA population  
CLEAR : WA\_SO.  
WA\_SO-VBELN = '1000'.  
WA\_SO-DATE\_CRTED = '20220706'.  
WA\_SO-ERNAM      = 'TRAINEE01'.  
WA\_SO-AUART    = 'RO'.  
WA\_SO-NETWR    = '126.55'.  
WA\_SO-WAERK    = 'USD'.  
APPEND WA\_SO TO IT\_SO.  
  
CLEAR : WA\_SO.  
WA\_SO-VBELN = '2000'.  
WA\_SO-DATE\_CRTED = '20220606'.  
WA\_SO-ERNAM      = 'TRAINEE02'.  
WA\_SO-AUART    = 'SO'.  
WA\_SO-NETWR    = '1856.55'.  
WA\_SO-WAERK    = 'INR'.  
APPEND WA\_SO TO IT\_SO.  
  
DO 5 TIMES.  
  WA\_SO-VBELN = WA\_SO-VBELN + 1 .  
  WA\_SO-DATE\_CRTED = WA\_SO-DATE\_CRTED + 1.  
  WA\_SO-ERNAM      = 'TRAINEE10'.  
  WA\_SO-AUART    = 'SO'.  
  WA\_SO-NETWR    = '1296.55'.  
  WA\_SO-WAERK    = 'EUR'.  
  APPEND WA\_SO TO IT\_SO.  
ENDDO.  
  
\*\*\* data processing  
WRITE:/ 'Before data DELETE'.  
LOOP AT IT\_SO INTO WA\_SO.  
  IF WA\_SO-VBELN < 2002.  
    WRITE:/ WA\_SO-VBELN, WA\_SO-DATE\_CRTED, WA\_SO-ERNAM, WA\_SO-NETWR,WA\_SO-WAERK.  
  ELSE.  
    WRITE:/ WA\_SO-VBELN, WA\_SO-DATE\_CRTED.  
  ENDIF.  
ENDLOOP.  
SKIP.  
  
WRITE : / 'DELETING'.  
  
\*\*\* DATA DELETE  
LOOP AT IT\_SO INTO WA\_SO.  
  IF WA\_SO-VBELN < 2002.  
    DELETE IT\_SO . " delete the first record from the table  
  ELSE.  
  ENDIF.  
ENDLOOP.  
  
SKIP.  
WRITE:/ 'After DELETE'.  
LOOP AT IT\_SO INTO WA\_SO.  
  WRITE:/ WA\_SO-VBELN, WA\_SO-DATE\_CRTED, WA\_SO-ERNAM, WA\_SO-NETWR,WA\_SO-WAERK.  
ENDLOOP.  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\*\* DELETE USING WHERE  
SKIP.  
WRITE : / 'DELETE USING WHERE'.  
DELETE IT\_SO WHERE VBELN = 2002. " delete command with condition  
IF SY-SUBRC <> 0.  
  CLEAR WA\_SO.  
  READ TABLE IT\_SO INTO WA\_SO WITH KEY VBELN = 2002.  
  IF SY-SUBRC = 0.  
    WRITE:/ 'FAILED, After DELETE'.  
    WRITE:/ WA\_SO-VBELN, WA\_SO-DATE\_CRTED, WA\_SO-ERNAM, WA\_SO-NETWR,WA\_SO-WAERK.  
  ENDIF.  
ELSE.  
  WRITE:/'Sucess', 'DELETE vbeln = 2002'.  
ENDIF.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\*\* DELETE USING index  
SKIP.  
WRITE : / 'DELETE USING INDEX'.  
DELETE IT\_SO INDEX 2 ." use 12 as index to check SY-SUBRC = 4 " delete with index  
IF SY-SUBRC <> 0.  
  CLEAR WA\_SO.  
  READ TABLE IT\_SO INTO WA\_SO INDEX 2.  
  IF SY-SUBRC = 0.  
    WRITE:/ 'FAILED DELETE WITH index'.  
    WRITE:/ WA\_SO-VBELN, WA\_SO-DATE\_CRTED, WA\_SO-ERNAM, WA\_SO-NETWR,WA\_SO-WAERK.  
  ENDIF.  
ELSE.  
  WRITE:/'Sucess', 'DELETE index = 2'.  
ENDIF.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\* deleTE ADJACENT DUPLICATE  
SKIP.  
WRITE : / 'DELETE ADJACENT DUPLICATE'.  
  
APPEND LINES OF IT\_SO TO IT\_SO\_NEW .  
APPEND LINES OF IT\_SO TO IT\_SO\_NEW .  
APPEND LINES OF IT\_SO TO IT\_SO\_NEW .  
APPEND LINES OF IT\_SO TO IT\_SO\_NEW .  
  
SORT  IT\_SO\_NEW BY VBELN.  
  
WRITE : / 'BEFORE DELETING'.  
LOOP AT IT\_SO\_NEW INTO WA\_SO\_NEW.  
    WRITE:/ WA\_SO\_NEW-VBELN, WA\_SO\_NEW-DATE\_CRTED, WA\_SO\_NEW-ERNAM, WA\_SO\_NEW-NETWR,WA\_SO\_NEW-WAERK.  
ENDLOOP.  
  
DELETE ADJACENT DUPLICATES FROM IT\_SO\_NEW COMPARING VBELN. " DELETING ADJACENT DUPLICATES  
IF SY-SUBRC = 0.  
  WRITE:/ 'Ok'.  
  WRITE :/ 'AFTER DELETING'.  
  LOOP AT IT\_SO\_NEW INTO WA\_SO\_NEW.  
    WRITE:/ WA\_SO\_NEW-VBELN, WA\_SO\_NEW-DATE\_CRTED, WA\_SO\_NEW-ERNAM, WA\_SO\_NEW-NETWR,WA\_SO\_NEW-WAERK.  
  ENDLOOP.  
ELSE.  
  WRITE:/ 'NOT ok'.  
ENDIF.



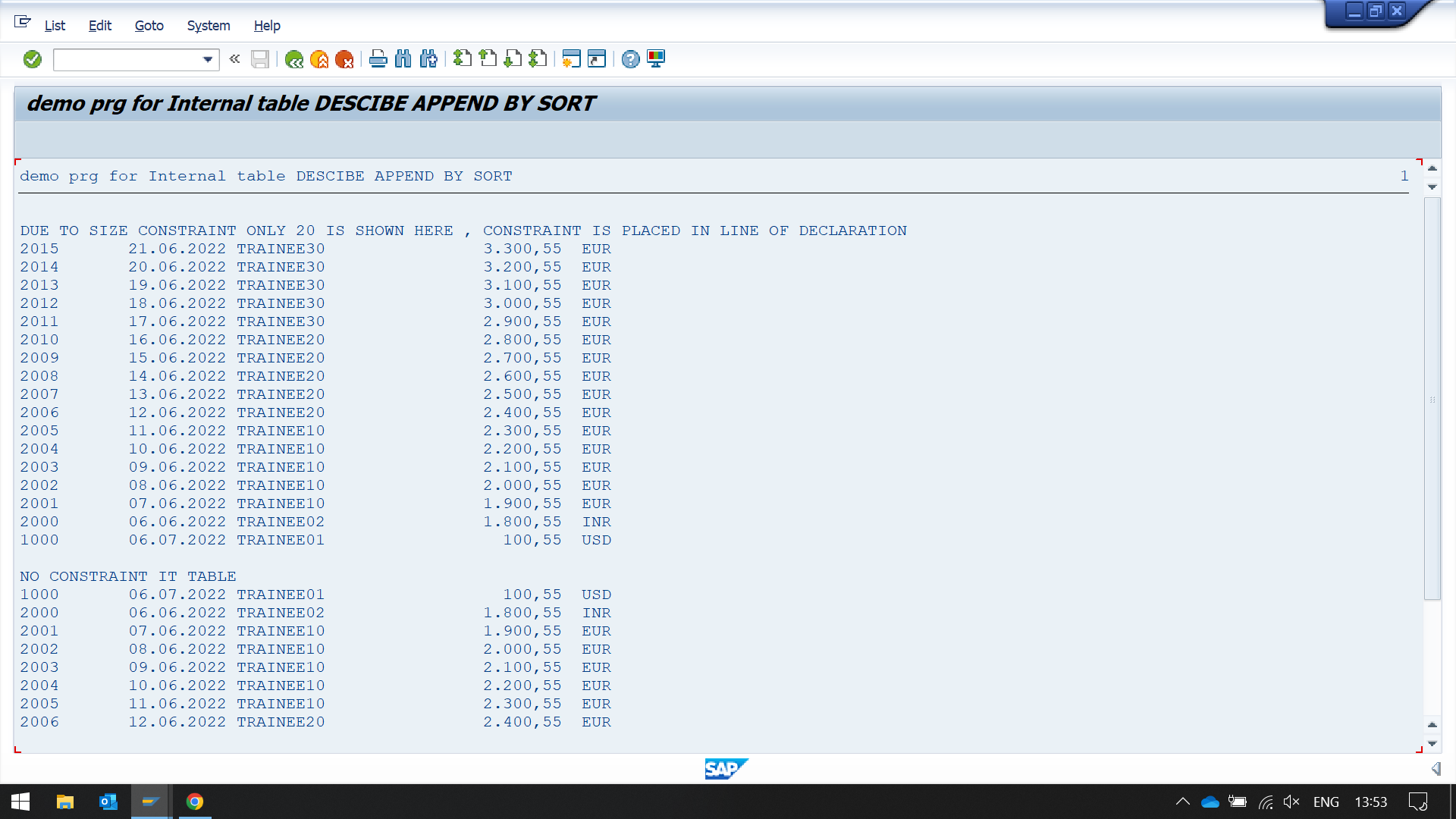
REPORT ZYNY\_DEMO\_INT\_TABLE\_07.  
  
\*\* Local user defined type  
TYPES: BEGIN OF GTY\_TAB,  
         VBELN      TYPE VBAK-VBELN,  
         DATE\_CRTED TYPE ERDAT,  
         ERNAM      TYPE VBAK-ERNAM,  
         AUART      TYPE VBAK-AUART,  
         NETWR      TYPE VBAK-NETWR,  
         WAERK      TYPE VBAK-WAERK,  
       END OF GTY\_TAB.  
  
\*\*\*\* Internal table declaration  
\*SORTED BY IS NOT WORKING ON IT WITHOUT INTIAL SIZE

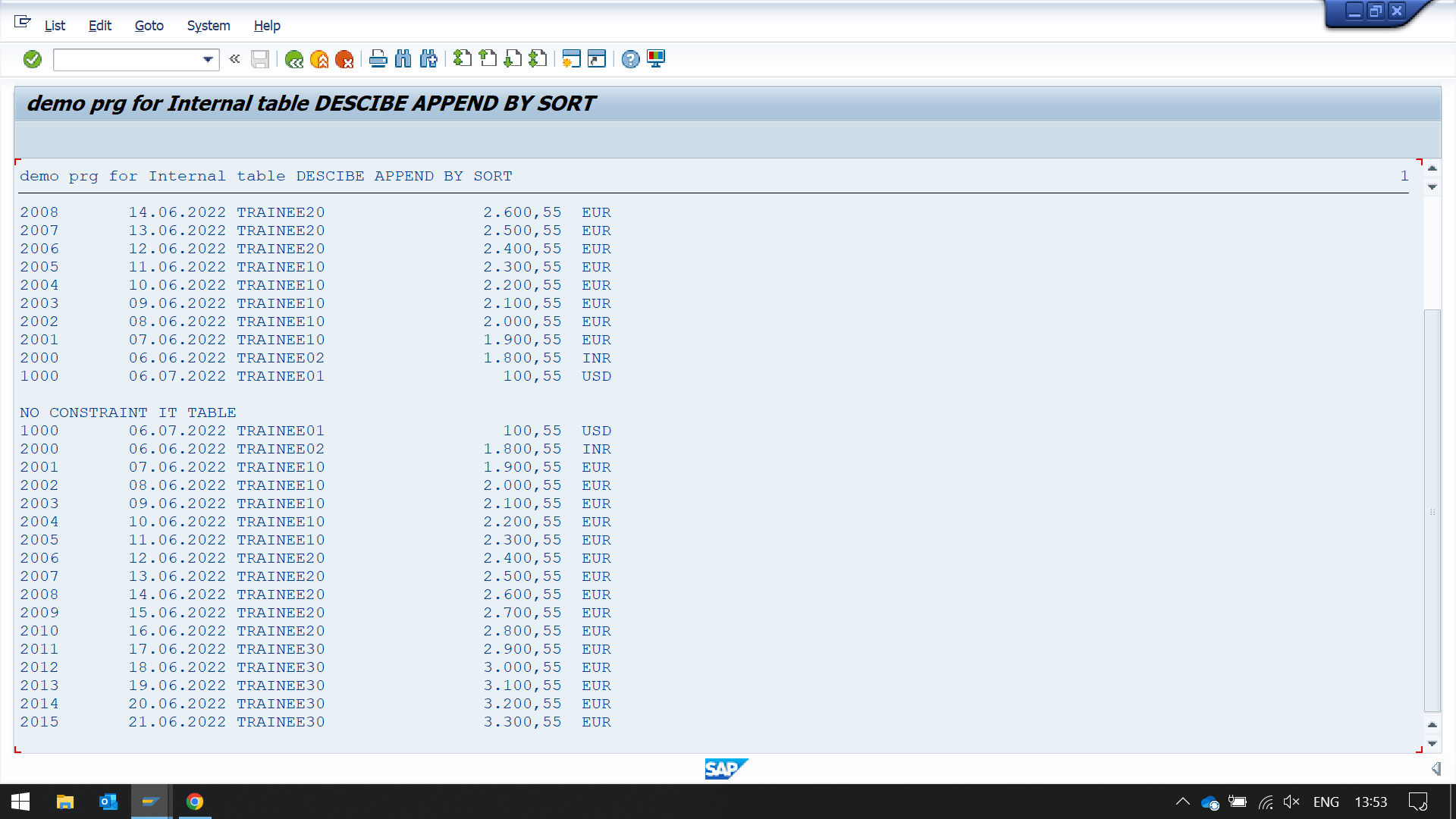
Used correctly, this addition can produce ranking lists in descending order. This only works if a value greater than 0 is specified in the declaration of the internal table in the addition [INITIAL SIZE](sapevent:ABAPTYPES_ITAB). If the value 0 is specified for INITIAL SIZE, the statement APPEND has no effect when used with the addition SORTED BY.

read more in documentation of sorted by.

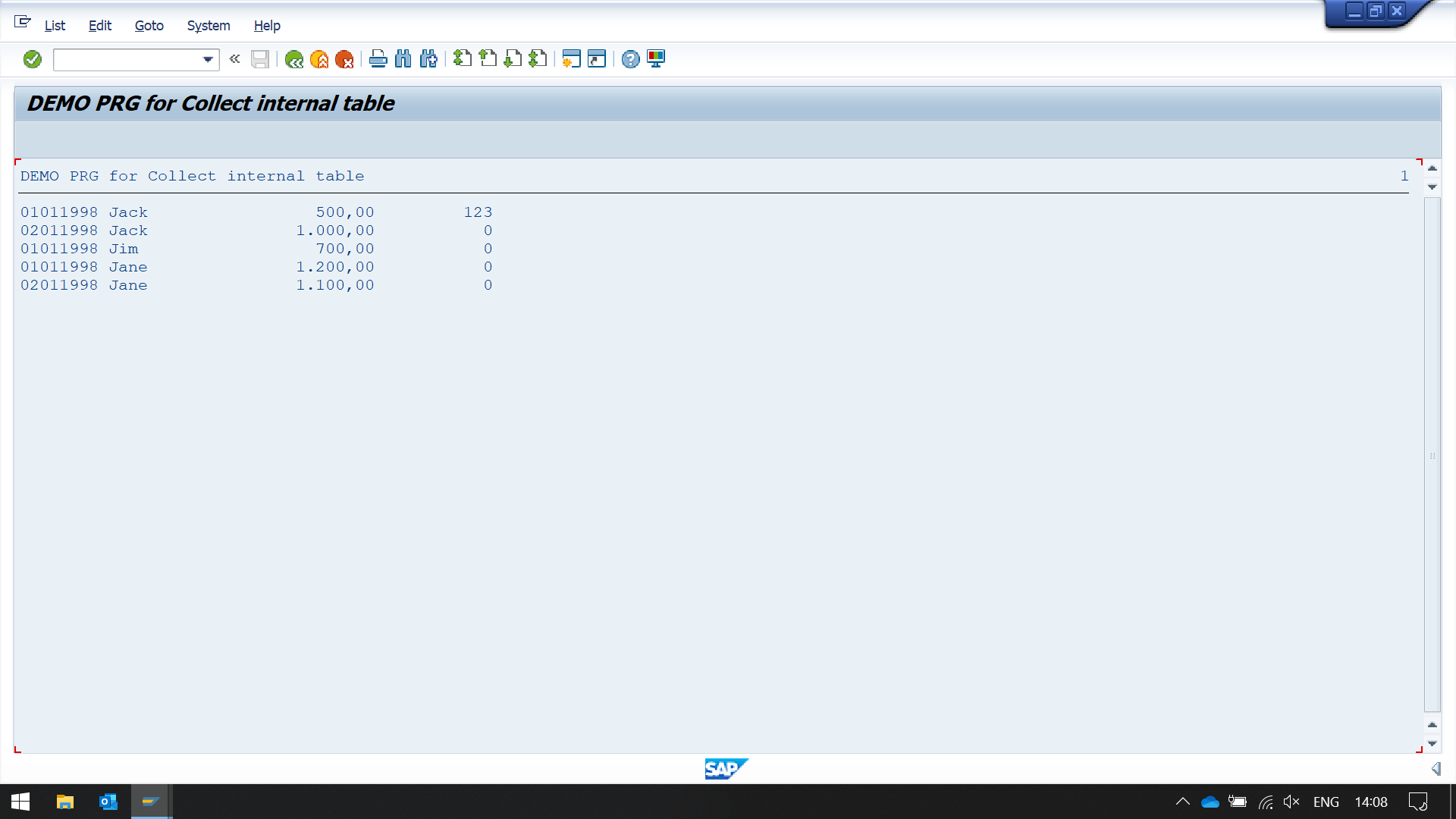
DATA: IT\_SO\_NEW TYPE STANDARD TABLE OF GTY\_TAB, " ITAB  
      IT\_SO     TYPE STANDARD TABLE OF GTY\_TAB INITIAL SIZE 32, " ITAB  
      WA\_SO     TYPE GTY\_TAB, " Header line or work area  
      WA\_SO\_NEW TYPE GTY\_TAB, " Header line or work area  
      IT\_SO\_NEW2 TYPE STANDARD TABLE OF  GTY\_TAB INITIAL SIZE 0. " ITAB  
  
\*START-OF-SELECTION.  
\*\*\*\* DATA population  
CLEAR : WA\_SO.  
CLEAR : WA\_SO\_NEW.  
WA\_SO-VBELN = '1000'.  
WA\_SO-DATE\_CRTED = '20220706'.  
WA\_SO-ERNAM      = 'TRAINEE01'.  
WA\_SO-AUART    = 'RO'.  
WA\_SO-NETWR    = '100.55'.  
WA\_SO-WAERK    = 'USD'.  
MOVE-CORRESPONDING WA\_SO TO WA\_SO\_NEW. " IT WILL MOVE ONLY CORRESPONDING FIELDS (MEANS THE NAME OF FIELDS OF DESTINATION SHOULD BE SAME AS SOURCE)  
APPEND WA\_SO TO IT\_SO SORTED BY NETWR.  
\*WRITE:/ WA\_SO-VBELN, WA\_SO-DATE\_CRTED, WA\_SO-ERNAM, WA\_SO-NETWR,WA\_SO-WAERK.  
\*APPEND WA\_SO\_NEW TO IT\_SO\_NEW SORTED BY NETWR.  
APPEND WA\_SO\_NEW TO IT\_SO\_NEW.  
\*LOOP AT IT\_SO\_NEW INTO WA\_SO\_NEW.  
\*  WRITE:/ WA\_SO\_NEW-VBELN, WA\_SO\_NEW-DATE\_CRTED, WA\_SO\_NEW-ERNAM, WA\_SO\_NEW-NETWR,WA\_SO\_NEW-WAERK.  
\*ENDLOOP.  
\*SKIP.  
  
CLEAR : WA\_SO.  
CLEAR : WA\_SO\_NEW.  
WA\_SO-VBELN = '2000'.  
WA\_SO-DATE\_CRTED = '20220606'.  
WA\_SO-ERNAM      = 'TRAINEE02'.  
WA\_SO-AUART    = 'SO'.  
WA\_SO-NETWR    = '1800.55'.  
WA\_SO-WAERK    = 'INR'.  
MOVE-CORRESPONDING WA\_SO TO WA\_SO\_NEW.  
\*WRITE:/ WA\_SO\_NEW-VBELN, WA\_SO\_NEW-DATE\_CRTED, WA\_SO\_NEW-ERNAM, WA\_SO\_NEW-NETWR,WA\_SO\_NEW-WAERK.  
APPEND WA\_SO TO IT\_SO SORTED BY NETWR.  
\*WRITE:/ WA\_SO-VBELN, WA\_SO-DATE\_CRTED, WA\_SO-ERNAM, WA\_SO-NETWR,WA\_SO-WAERK.  
APPEND WA\_SO\_NEW TO IT\_SO\_NEW.  
\*LOOP AT IT\_SO\_NEW INTO WA\_SO\_NEW.  
\*  WRITE:/ WA\_SO\_NEW-VBELN, WA\_SO\_NEW-DATE\_CRTED, WA\_SO\_NEW-ERNAM, WA\_SO\_NEW-NETWR,WA\_SO\_NEW-WAERK.  
\*ENDLOOP.  
\*SKIP.  
  
DO 5 TIMES.  
  WA\_SO-VBELN = WA\_SO-VBELN + 1 .  
  WA\_SO-DATE\_CRTED = WA\_SO-DATE\_CRTED + 1.  
  WA\_SO-ERNAM      = 'TRAINEE10'.  
  WA\_SO-AUART    = 'SO'.  
  WA\_SO-NETWR    = WA\_SO-NETWR + 100.  
  WA\_SO-WAERK    = 'EUR'.  
  APPEND WA\_SO TO IT\_SO SORTED BY NETWR.  
  MOVE-CORRESPONDING WA\_SO TO WA\_SO\_NEW.  
\*   WRITE:/ WA\_SO-VBELN, WA\_SO-DATE\_CRTED, WA\_SO-ERNAM, WA\_SO-NETWR,WA\_SO-WAERK.  
   APPEND WA\_SO\_NEW TO IT\_SO\_NEW.  
ENDDO.  
\*LOOP AT IT\_SO\_NEW INTO WA\_SO\_NEW.  
\*  WRITE:/ WA\_SO\_NEW-VBELN, WA\_SO\_NEW-DATE\_CRTED, WA\_SO\_NEW-ERNAM, WA\_SO\_NEW-NETWR,WA\_SO\_NEW-WAERK.  
\*ENDLOOP.  
\*SKIP.  
  
DO 5 TIMES.  
  WA\_SO-VBELN = WA\_SO-VBELN + 1 .  
  WA\_SO-DATE\_CRTED = WA\_SO-DATE\_CRTED + 1.  
  WA\_SO-ERNAM      = 'TRAINEE20'.  
  WA\_SO-AUART    = 'SO'.  
  WA\_SO-NETWR    = WA\_SO-NETWR + 100.  
  WA\_SO-WAERK    = 'EUR'.  
  MOVE-CORRESPONDING WA\_SO TO WA\_SO\_NEW.  
APPEND WA\_SO TO IT\_SO SORTED BY NETWR.  
\*WRITE:/ WA\_SO-VBELN, WA\_SO-DATE\_CRTED, WA\_SO-ERNAM, WA\_SO-NETWR,WA\_SO-WAERK.  
APPEND WA\_SO\_NEW TO IT\_SO\_NEW.  
ENDDO.  
\*LOOP AT IT\_SO\_NEW INTO WA\_SO\_NEW.  
\*  WRITE:/ WA\_SO\_NEW-VBELN, WA\_SO\_NEW-DATE\_CRTED, WA\_SO\_NEW-ERNAM, WA\_SO\_NEW-NETWR,WA\_SO\_NEW-WAERK.  
\*ENDLOOP.  
\*SKIP.  
  
DO 5 TIMES.  
  WA\_SO-VBELN = WA\_SO-VBELN + 1 .  
  WA\_SO-DATE\_CRTED = WA\_SO-DATE\_CRTED + 1.  
  WA\_SO-ERNAM      = 'TRAINEE30'.  
  WA\_SO-AUART    = 'SO'.  
  WA\_SO-NETWR    = WA\_SO-NETWR + 100.  
  WA\_SO-WAERK    = 'EUR'.  
  MOVE-CORRESPONDING WA\_SO TO WA\_SO\_NEW.  
APPEND WA\_SO TO IT\_SO SORTED BY NETWR.  
\*WRITE:/ WA\_SO-VBELN, WA\_SO-DATE\_CRTED, WA\_SO-ERNAM, WA\_SO-NETWR,WA\_SO-WAERK.  
APPEND WA\_SO\_NEW TO IT\_SO\_NEW.  
ENDDO.  
\*LOOP AT IT\_SO\_NEW INTO WA\_SO\_NEW.  
\*  WRITE:/ WA\_SO\_NEW-VBELN, WA\_SO\_NEW-DATE\_CRTED, WA\_SO\_NEW-ERNAM, WA\_SO\_NEW-NETWR,WA\_SO\_NEW-WAERK.  
\*ENDLOOP.  
\*SKIP.  
  
\*\*\* data processing  
SKIP.

if table is initialized with size of 20 then it will only contain 20 records  
WRITE : / 'DUE TO SIZE CONSTRAINT ONLY 20 IS SHOWN HERE , CONSTRAINT IS PLACED IN LINE OF DECLARATION'.  
LOOP AT IT\_SO INTO WA\_SO.  
  WRITE:/ WA\_SO-VBELN, WA\_SO-DATE\_CRTED, WA\_SO-ERNAM, WA\_SO-NETWR,WA\_SO-WAERK.  
ENDLOOP.  
  
SKIP.  
WRITE: / 'NO CONSTRAINT IT TABLE'.  
LOOP AT IT\_SO\_NEW INTO WA\_SO\_NEW.  
  WRITE:/ WA\_SO\_NEW-VBELN, WA\_SO\_NEW-DATE\_CRTED, WA\_SO\_NEW-ERNAM, WA\_SO\_NEW-NETWR,WA\_SO\_NEW-WAERK.  
ENDLOOP.

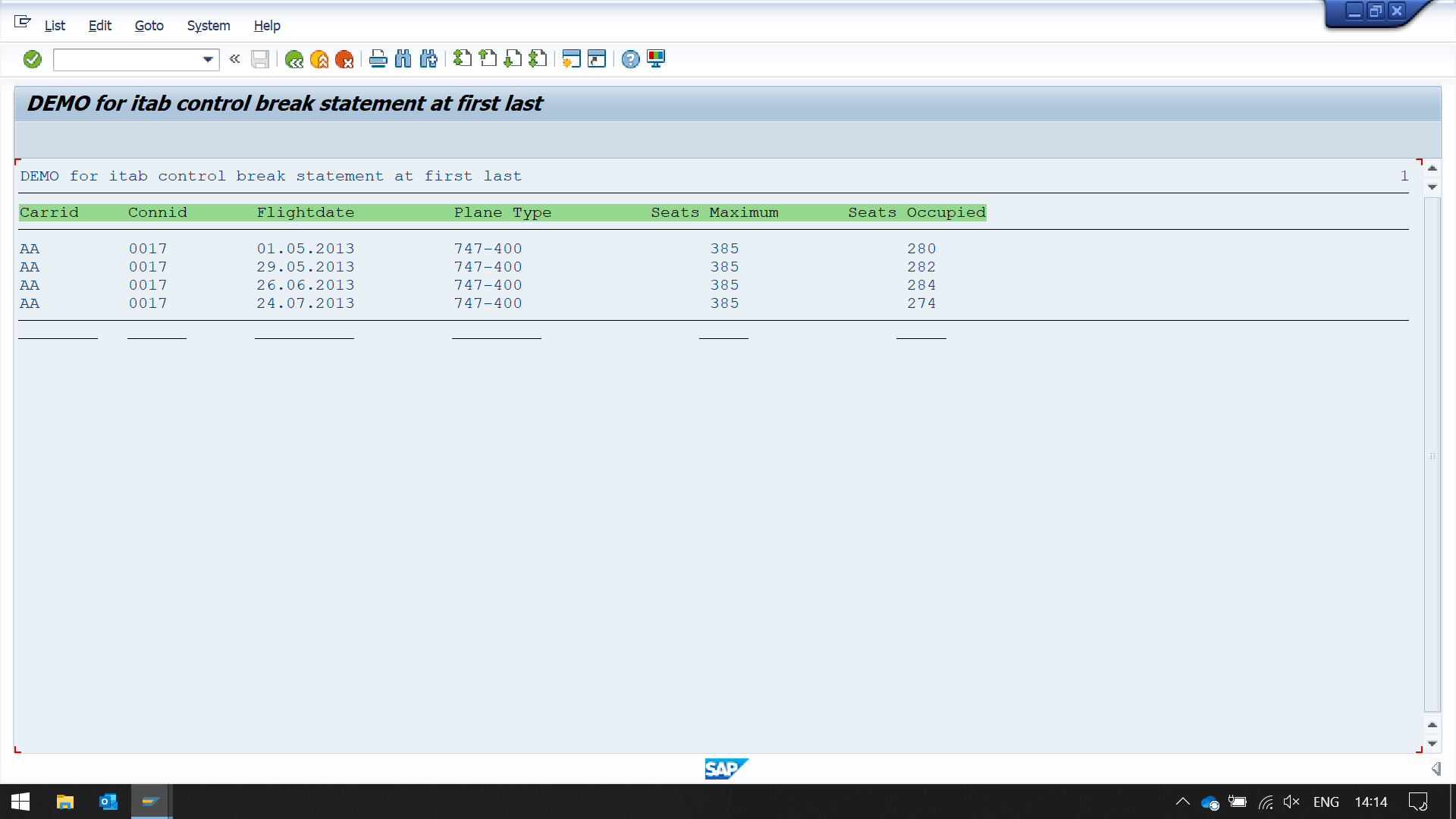


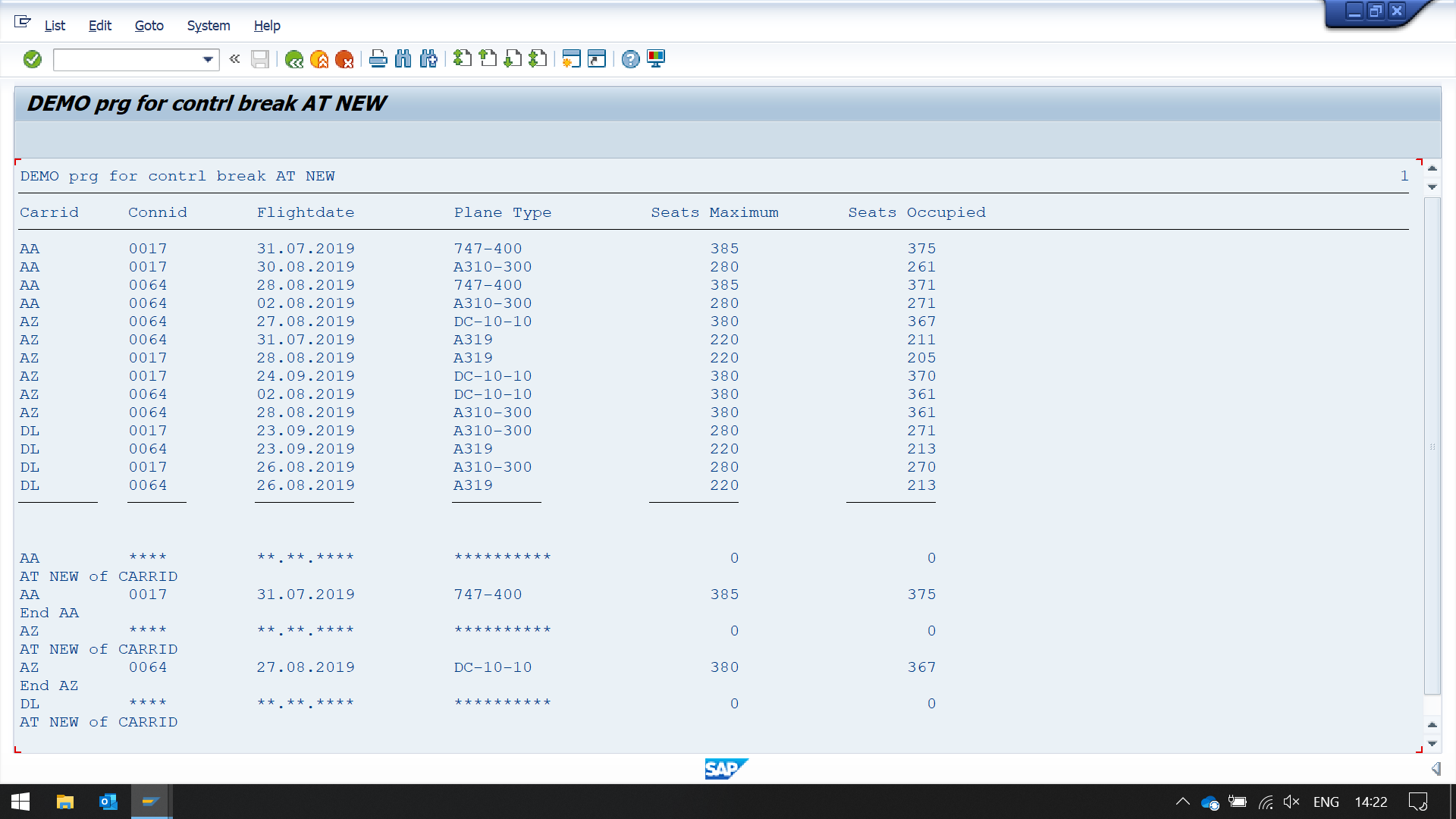


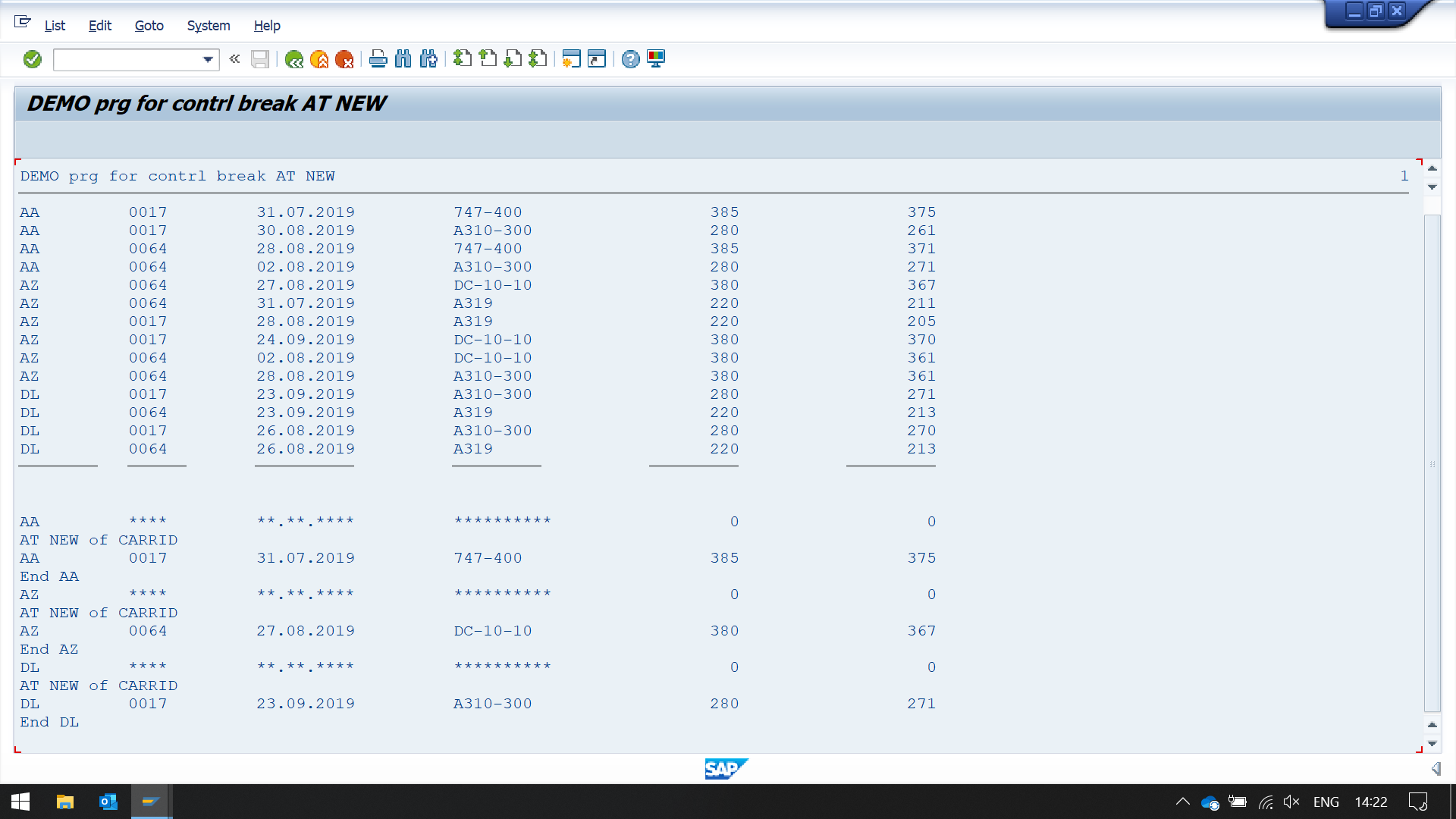
REPORT ZYNY\_DEMO\_INT\_TABLE\_08.  
  
TYPES: BEGIN OF ITTYPE ,  
         SALES    TYPE P DECIMALS 2,  
         NAME(10) TYPE C,  
         DATE     TYPE D,  
         num      type i,  
       END OF ITTYPE.  
DATA: IT TYPE STANDARD TABLE OF ITTYPE INITIAL SIZE 0,  
       WA TYPE ITTYPE.  
\*This statement inserts the content of a work area wa either as a single row in an internal table itab  
\*adds the values of its numeric components to the corresponding values of existing rows with the same  
\*primary table key. wa is a functional operand position.  
WA-DATE = '19980101'. WA-SALES = 100. WA-NAME  = 'Jack'. wa-num = 100.  
COLLECT WA INTO IT.  
WA-DATE = '19980101'. WA-SALES = 400. WA-NAME  = 'Jack'. wa-num = 23.  
COLLECT WA INTO IT.  
clear wa.  
 " here sales adds up 100 + 400 = 500 and same with num  
WA-DATE = '19980102'. WA-SALES = 300. WA-NAME  = 'Jack'.  
COLLECT WA INTO IT.  
WA-DATE = '19980102'. WA-SALES = 700. WA-NAME  = 'Jack'.  
COLLECT WA INTO IT.  
 " here we are clearing the work area so 0 + 0 = 0, but sales are still adding 300 + 700 = 1000  
WA-DATE = '19980101'. WA-SALES = 200. WA-NAME  = 'Jim'.  
COLLECT WA INTO IT.  
WA-DATE = '19980101'. WA-SALES = 500. WA-NAME  = 'Jim'.  
COLLECT WA INTO IT.  
  
WA-DATE = '19980101'. WA-SALES = 600. WA-NAME  = 'Jane'.  
COLLECT WA INTO IT.  
WA-DATE = '19980101'. WA-SALES = 600. WA-NAME  = 'Jane'.  
COLLECT WA INTO IT.  
  
WA-DATE = '19980102'. WA-SALES = 600. WA-NAME  = 'Jane'.  
COLLECT WA INTO IT.  
WA-DATE = '19980102'. WA-SALES = 500. WA-NAME  = 'Jane'.  
COLLECT WA INTO IT.  
  
  
LOOP AT IT INTO WA.  
  WRITE: /  WA-DATE, WA-NAME, WA-SALES , wa-num.  
ENDLOOP.

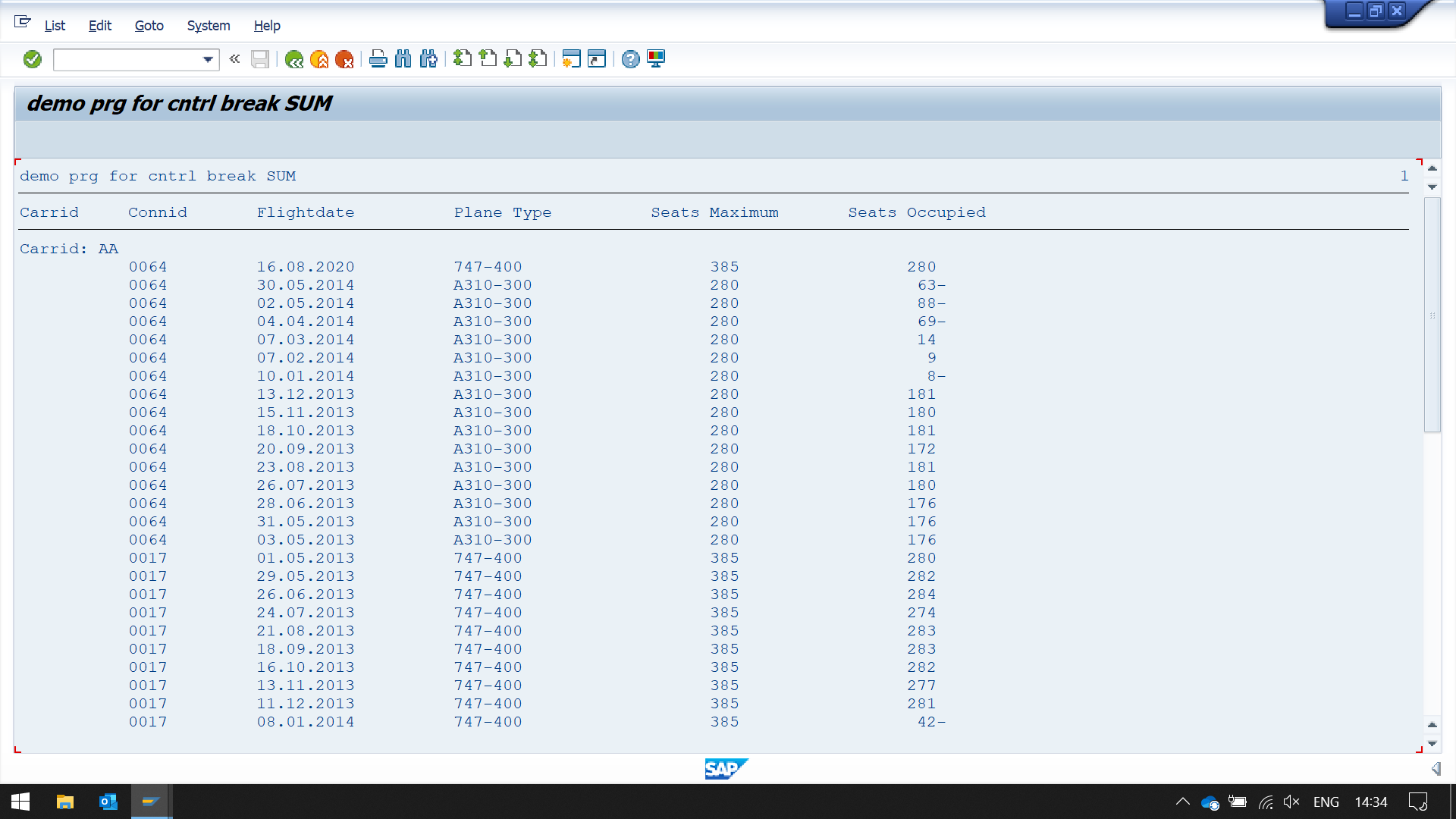


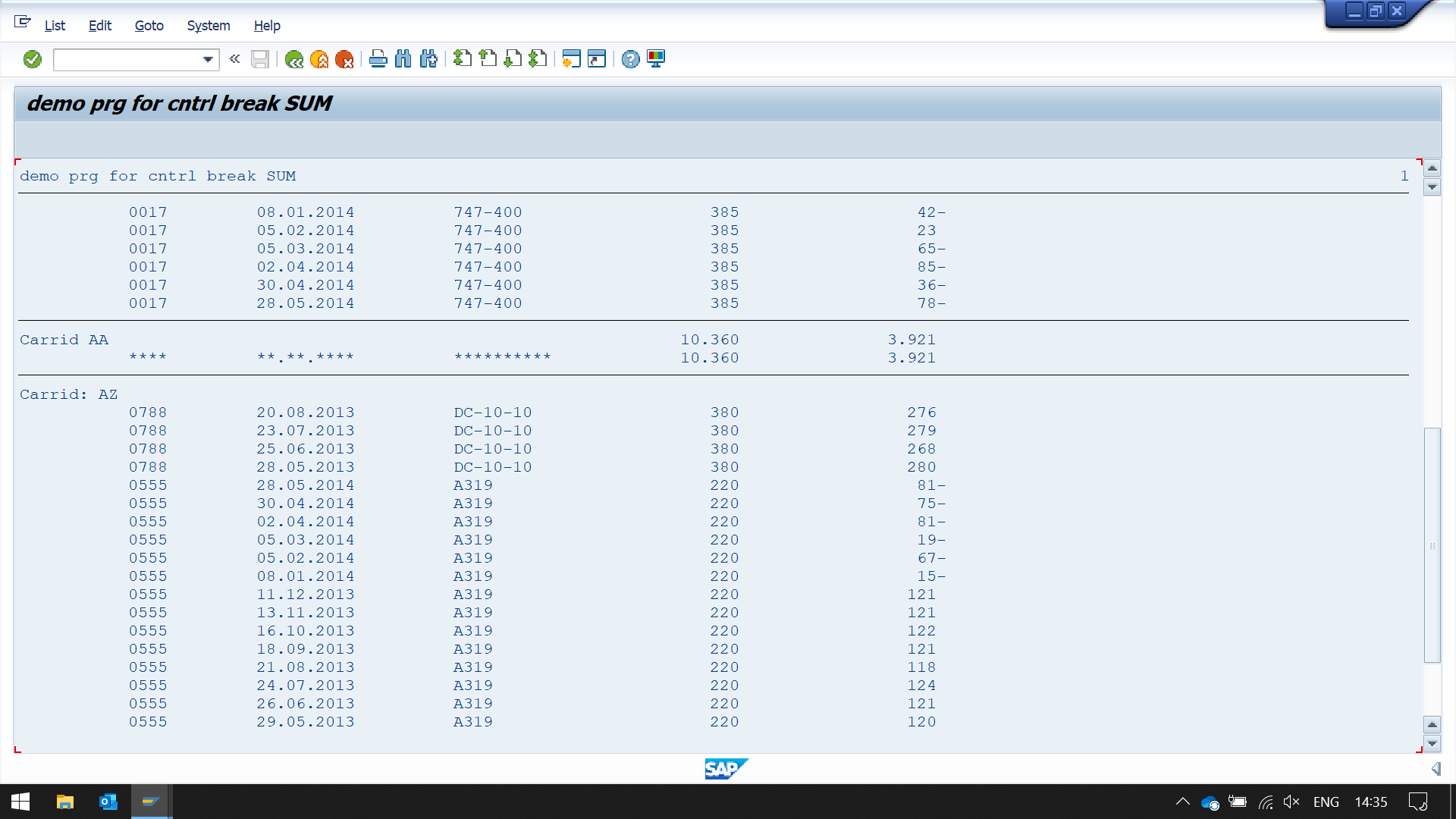
REPORT ZYNY\_DEMO\_INT\_TABLE\_09.  
  
TYPES: BEGIN OF FLIGHTTYPE,  
         CARRID    TYPE SFLIGHT-CARRID,  
         CONNID    TYPE SFLIGHT-CONNID,  
         FLDATE    TYPE SFLIGHT-FLDATE,  
         PLANETYPE TYPE SFLIGHT-PLANETYPE,  
         SEATSMAX  TYPE SFLIGHT-SEATSMAX,  
         SEATSOCC  TYPE SFLIGHT-SEATSOCC,  
       END OF   FLIGHTTYPE.  
DATA: ITSFLIGHT TYPE STANDARD TABLE OF FLIGHTTYPE INITIAL SIZE 0,  
      WA        TYPE  FLIGHTTYPE.  
  
SELECT CARRID CONNID FLDATE PLANETYPE SEATSMAX SEATSOCC UP TO 4 ROWS  
    FROM SFLIGHT  
  INTO TABLE ITSFLIGHT.  
  
LOOP AT ITSFLIGHT INTO WA.  
  AT FIRST.  
    FORMAT COLOR 5. " coloring the result  
    WRITE: /  'Carrid',  
            12 'Connid',  
            25 'Flightdate',  
            45 'Plane Type',  
            65 'Seats Maximum',  
            85 'Seats Occupied'.  
    FORMAT COLOR OFF.  
    ULINE.  
  ENDAT.  
  WRITE : / WA-CARRID ,  
           12 WA-CONNID,  
           25 WA-FLDATE,  
           45 WA-PLANETYPE,  
           65 WA-SEATSMAX,  
           85 WA-SEATSOCC.  
  AT LAST.  
    write : / sy-uline.  
    WRITE: /  '--------',  
           12 '------',  
           25 '----------',  
           45 '---------',  
           70 '-----',  
           90 '-----'.  
  ENDAT.  
ENDLOOP.

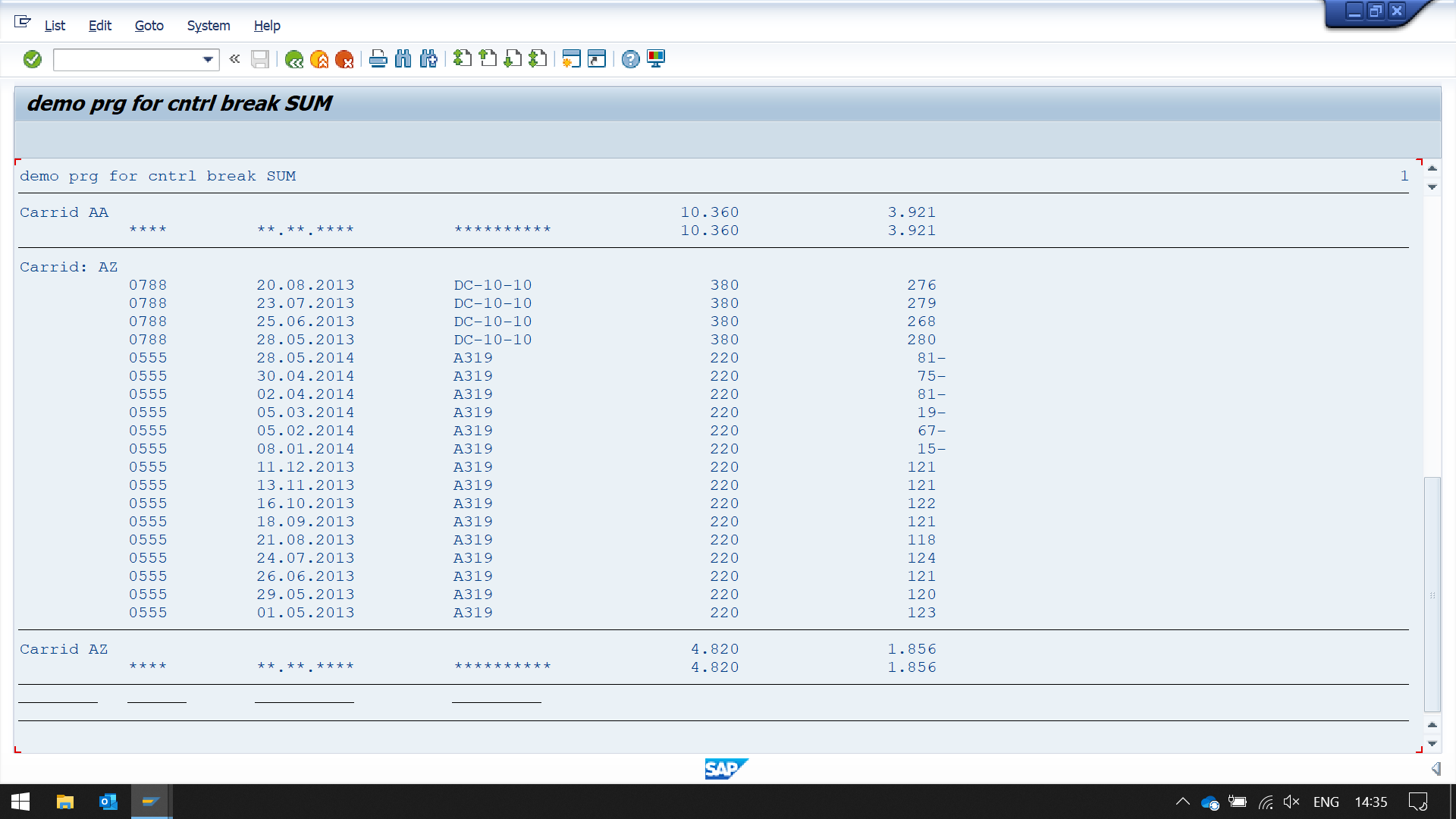
REPORT zyny\_demo\_int\_table\_10.  
  
TYPES: BEGIN OF flighttype,  
         carrid    TYPE sflight-carrid,  
         connid    TYPE sflight-connid,  
         fldate    TYPE sflight-fldate,  
         planetype TYPE sflight-planetype,  
         seatsmax  TYPE sflight-seatsmax,  
         seatsocc  TYPE sflight-seatsocc,  
       END OF   flighttype.  
DATA: itsflight TYPE STANDARD TABLE OF flighttype       INITIAL SIZE 0,  
      wa        TYPE  flighttype.  
  
data: wa\_new        TYPE  flighttype.  
data: gv\_flag TYPE c.  
  
wa-carrid = 'AA'. wa-connid = '17'. wa-fldate = '20190731'. wa-planetype = '747-400'.  wa-seatsmax = 385. wa-seatsocc = 375.  
APPEND wa TO itsflight.  
  
wa-carrid = 'AZ'. wa-connid = '64'. wa-fldate = '20190827'. wa-planetype = 'DC-10-10'. wa-seatsmax = 380. wa-seatsocc = 367.  
APPEND wa TO itsflight.  
  
wa-carrid = 'DL'. wa-connid = '17'. wa-fldate = '20190923'. wa-planetype = 'A310-300'. wa-seatsmax = 280. wa-seatsocc = 271.  
APPEND wa TO itsflight.  
  
wa-carrid = 'AZ'. wa-connid = '64'. wa-fldate = '20190731'. wa-planetype = 'A319'.     wa-seatsmax = 220. wa-seatsocc = 211.  
APPEND wa TO itsflight.  
  
wa-carrid = 'AA'. wa-connid = '17'. wa-fldate = '20190830'. wa-planetype = 'A310-300'. wa-seatsmax = 280. wa-seatsocc = 261.  
APPEND wa TO itsflight.  
  
wa-carrid = 'DL'. wa-connid = '64'. wa-fldate = '20190923'. wa-planetype = 'A319'.     wa-seatsmax = 220. wa-seatsocc = 213.  
APPEND wa TO itsflight.  
  
wa-carrid = 'AA'. wa-connid = '64'. wa-fldate = '20190828'. wa-planetype = '747-400'.  wa-seatsmax = 385. wa-seatsocc = 371.  
APPEND wa TO itsflight.  
  
wa-carrid = 'AZ'. wa-connid = '17'. wa-fldate = '20190828'. wa-planetype = 'A319'.     wa-seatsmax = 220. wa-seatsocc = 205.  
APPEND wa TO itsflight.  
  
wa-carrid = 'AA'. wa-connid = '64'. wa-fldate = '20190802'. wa-planetype = 'A310-300'. wa-seatsmax = 280. wa-seatsocc = 271.  
APPEND wa TO itsflight.  
  
wa-carrid = 'AZ'. wa-connid = '17'. wa-fldate = '20190924'. wa-planetype = 'DC-10-10'. wa-seatsmax = 380. wa-seatsocc = 370.  
APPEND wa TO itsflight.  
  
wa-carrid = 'DL'. wa-connid = '17'. wa-fldate = '20190826'. wa-planetype = 'A310-300'. wa-seatsmax = 280. wa-seatsocc = 270.  
APPEND wa TO itsflight.  
  
wa-carrid = 'AZ'. wa-connid = '64'. wa-fldate = '20190802'. wa-planetype = 'DC-10-10'. wa-seatsmax = 380. wa-seatsocc = 361.  
APPEND wa TO itsflight.  
  
wa-carrid = 'DL'. wa-connid = '64'. wa-fldate = '20190826'. wa-planetype = 'A319'.     wa-seatsmax = 220. wa-seatsocc = 213.  
APPEND wa TO itsflight.  
  
wa-carrid = 'AZ'. wa-connid = '64'. wa-fldate = '20190828'. wa-planetype = 'A310-300'. wa-seatsmax = 380. wa-seatsocc = 361.  
APPEND wa TO itsflight.  
  
  
SORT itsflight BY carrid.  
LOOP AT  itsflight  INTO wa.  
  AT FIRST. " at first iteration of loop  
    PERFORM dispfirst.  
  ENDAT.  
  PERFORM dispallfields.  
  AT LAST. " at last iteration of loop  
    PERFORM displast.  
  ENDAT.  
ENDLOOP.  
  
SKIP 2.  
LOOP AT  itsflight  INTO wa.  
  wa\_new = wa.              " Collect the data in new work area  
  clear gv\_flag.  
  AT NEW carrid. " at every new value of carrid  
    gv\_flag = 'X'.  
    PERFORM dispallfields.  
  ENDAT.  
  AT END OF carrid. " at last value of wa first field (means every ending of a continuous value)  
    WRITE : / 'End' ,  wa-carrid.  
  ENDAT.  
ENDLOOP.  
SKIP.  
  
FORM dispallfields.  
  WRITE : / wa-carrid ,  
          12 wa-connid,  
          25 wa-fldate,  
          45 wa-planetype,  
          65 wa-seatsmax,  
          85 wa-seatsocc.  
  
if gv\_flag = 'X'.  
  WRITE : / 'AT NEW of CARRID'.  
  WRITE : / wa\_new-carrid ,  
          12 wa\_new-connid,  
          25 wa\_new-fldate,  
          45 wa\_new-planetype,  
          65 wa\_new-seatsmax,  
          85 wa\_new-seatsocc.  
 ENDIF.  
ENDFORM.  
  
FORM dispfirst.  
  WRITE: /  'Carrid',  
          12 'Connid',  
          25 'Flightdate',  
          45 'Plane Type',  
          65 'Seats Maximum',  
          85 'Seats Occupied'.  
  ULINE.  
ENDFORM.  
  
  
FORM displast.  
  WRITE: /  '--------',  
            12 '------',  
            25 '----------',  
            45 '---------',  
            65 '---------',  
            85 '---------'.  
ENDFORM.



REPORT zyny\_demo\_int\_table\_11.  
  
TYPES: BEGIN OF flighttype,  
         carrid    TYPE sflight-carrid,  
         connid    TYPE sflight-connid,  
         fldate    TYPE sflight-fldate,  
         planetype TYPE sflight-planetype,  
         seatsmax  TYPE sflight-seatsmax,  
         seatsocc  TYPE sflight-seatsocc,  
       END OF   flighttype.  
  
DATA: itsflight TYPE STANDARD TABLE OF flighttype INITIAL SIZE 0,  
      wa        TYPE  flighttype.  
  
PERFORM selectfromdb.  
  
SORT itsflight BY CARRID.  
LOOP AT  itsflight  INTO wa.  
  AT FIRST.  
    PERFORM dispfirst.  
  ENDAT.  
  AT NEW carrid.  
    WRITE : / 'Carrid:', wa-carrid.  
  ENDAT.  
  
  PERFORM disprestfields.  
  
  AT END OF carrid.  
    SUM. " HERE SUM OF ALL THE FIELDS ARE STORED IN THE WORK AREA  
    ULINE.  
    WRITE : / 'Carrid' ,  wa-carrid , 65 wa-seatsmax, 85 wa-seatsocc.  
    WRITE: /12 wa-connid,  
          25 wa-fldate,  
          45 wa-planetype,  
          65 wa-seatsmax,  
          85 wa-seatsocc.  
    ULINE.  
  ENDAT.  
  AT LAST.  
    PERFORM displast.  
  ENDAT.  
ENDLOOP.  
SKIP.  
  
FORM selectfromdb.  
  SELECT carrid connid fldate planetype seatsmax seatsocc UP TO 50 ROWS  
      FROM sflight  
    INTO TABLE itsflight.  
ENDFORM.  
  
FORM disprestfields.  
  WRITE: /12 wa-connid,  
          25 wa-fldate,  
          45 wa-planetype,  
          65 wa-seatsmax,  
          85 wa-seatsocc.  
ENDFORM.  
  
FORM dispfirst.  
  WRITE: /'Carrid',  
          12 'Connid',  
          25 'Flightdate',  
          45 'Plane Type',  
          65 'Seats Maximum',  
          85 'Seats Occupied'.  
  ULINE.  
ENDFORM.  
  
FORM displast.  
  WRITE: /'--------',  
          12 '------',  
          25 '----------',  
          45 '---------'.  
  uline.  
ENDFORM.







REPORT ZYNY\_DEMO\_INT\_TABLE\_12.  
  
TYPES: BEGIN OF FLIGHTTYPE,  
         FLDATE    TYPE SFLIGHT-FLDATE,  
         PLANETYPE TYPE SFLIGHT-PLANETYPE,  
         CARRID    TYPE SFLIGHT-CARRID,  
         CONNID    TYPE SFLIGHT-CONNID,  
         SEATSMAX  TYPE SFLIGHT-SEATSMAX,  
         SEATSOCC  TYPE SFLIGHT-SEATSOCC,  
       END OF   FLIGHTTYPE.  
DATA: ITSFLIGHT TYPE STANDARD TABLE OF FLIGHTTYPE  
      INITIAL SIZE 0,  
      WA        TYPE  FLIGHTTYPE.  
  
WA-CARRID = 'AA'. WA-CONNID = '17'. WA-FLDATE = '20190731'. WA-PLANETYPE = '747-400'. WA-SEATSMAX = 385. WA-SEATSOCC = 375.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'AZ'. WA-CONNID = '64'. WA-FLDATE = '20190827'. WA-PLANETYPE = 'DC-10-10'. WA-SEATSMAX = 380. WA-SEATSOCC = 367.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'DL'. WA-CONNID = '17'. WA-FLDATE = '20190923'. WA-PLANETYPE = 'A310-300'. WA-SEATSMAX = 280. WA-SEATSOCC = 271.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'AZ'. WA-CONNID = '64'. WA-FLDATE = '20190731'. WA-PLANETYPE = 'A319'. WA-SEATSMAX = 220. WA-SEATSOCC = 211.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'AA'. WA-CONNID = '17'. WA-FLDATE = '20190830'. WA-PLANETYPE = 'A310-300'. WA-SEATSMAX = 280. WA-SEATSOCC = 261.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'DL'. WA-CONNID = '64'. WA-FLDATE = '20190923'. WA-PLANETYPE = 'A319'. WA-SEATSMAX = 220. WA-SEATSOCC = 213.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'AA'. WA-CONNID = '64'. WA-FLDATE = '20190828'. WA-PLANETYPE = '747-400'.  WA-SEATSMAX = 385. WA-SEATSOCC = 371.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'AZ'. WA-CONNID = '17'. WA-FLDATE = '20190828'. WA-PLANETYPE = 'A319'. WA-SEATSMAX = 220. WA-SEATSOCC = 205.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'AA'. WA-CONNID = '64'. WA-FLDATE = '20190802'. WA-PLANETYPE = 'A310-300'. WA-SEATSMAX = 280. WA-SEATSOCC = 271.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'AZ'. WA-CONNID = '17'. WA-FLDATE = '20190924'. WA-PLANETYPE = 'DC-10-10'. WA-SEATSMAX = 380. WA-SEATSOCC = 370.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'DL'. WA-CONNID = '17'. WA-FLDATE = '20190826'. WA-PLANETYPE = 'A310-300'. WA-SEATSMAX = 280. WA-SEATSOCC = 270.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'AZ'. WA-CONNID = '64'. WA-FLDATE = '20190802'. WA-PLANETYPE = 'DC-10-10'. WA-SEATSMAX = 380. WA-SEATSOCC = 361.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'DL'. WA-CONNID = '64'. WA-FLDATE = '20190826'. WA-PLANETYPE = 'A319'. WA-SEATSMAX = 220. WA-SEATSOCC = 213.  
APPEND WA TO ITSFLIGHT.  
  
WA-CARRID = 'AZ'. WA-CONNID = '64'. WA-FLDATE = '20190828'. WA-PLANETYPE = 'A310-300'. WA-SEATSMAX = 380. WA-SEATSOCC = 361.  
APPEND WA TO ITSFLIGHT.  
  
  
SORT ITSFLIGHT BY CARRID.  
LOOP AT  ITSFLIGHT  INTO WA.  
  AT FIRST.  
    PERFORM DISPFIRST.  
  ENDAT.  
  PERFORM DISPALLFIELDS.  
  AT LAST.  
    PERFORM DISPLAST.  
  ENDAT.  
ENDLOOP.  
  
SKIP 2.  
LOOP AT  ITSFLIGHT  INTO WA.  
\*  here we are displaying content of only that carrid, when it change  
  ON CHANGE OF WA-CARRID.  
    PERFORM DISPALLFIELDS.  
  ENDON.  
ENDLOOP.  
SKIP.  
  
FORM DISPALLFIELDS.  
  WRITE : / WA-CARRID ,  
          12 WA-CONNID,  
          25 WA-FLDATE,  
          45 WA-PLANETYPE,  
          65 WA-SEATSMAX,  
          85 WA-SEATSOCC.  
ENDFORM.  
  
FORM DISPFIRST.  
  WRITE: /  'Carrid',  
          12 'Connid',  
          25 'Flightdate',  
          45 'Plane Type',  
          65 'Seats Maximum',  
          85 'Seats Occupied'.  
  ULINE.  
ENDFORM.  
  
FORM DISPLAST.  
  WRITE: /  '--------',  
            12 '------',  
            25 '----------',  
            45 '---------',  
            65 '---------',  
            85 '---------'.  
ENDFORM.

