NEW ABAP SYNTAX TIPS & EXPRESSIONS

INSTEAD OF CATCH CX_SY_ITAB_LINE_NOT_FOUND:

```
data(gs_booking) = value #( gt_booking•
                   [ 1 ] optional ).
```

QUICK DATA DISPLAY:

```
cl_demo_output=>write( gt_booking ).•
cl_demo_output=>write( gs_booking ).
cl_demo_output=>display( ).
```

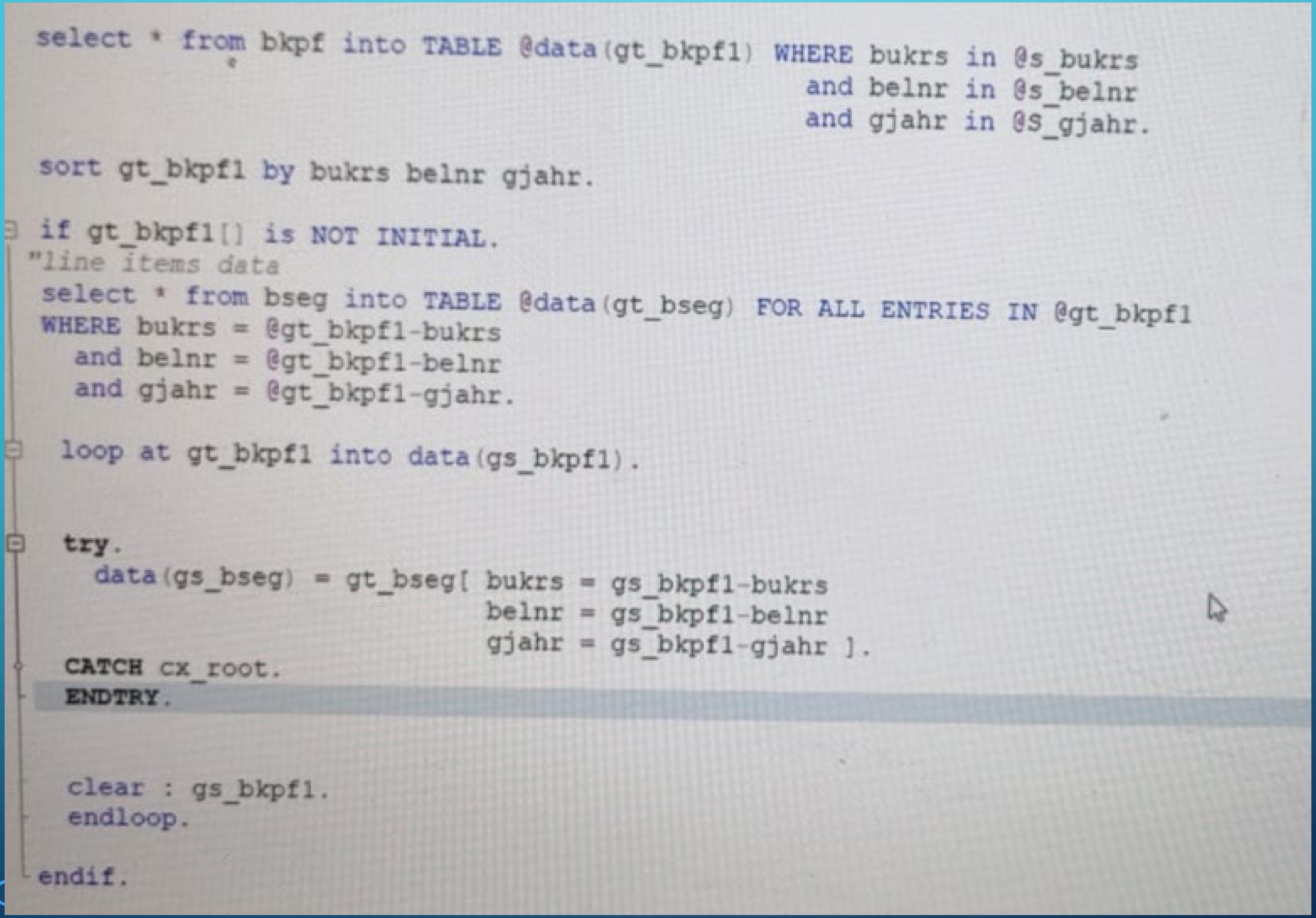
INSTEAD OF DESCRIBE TABLE…LINES:

```
data(gv_lines) = lines( gt_booking ).
```

Read last row:

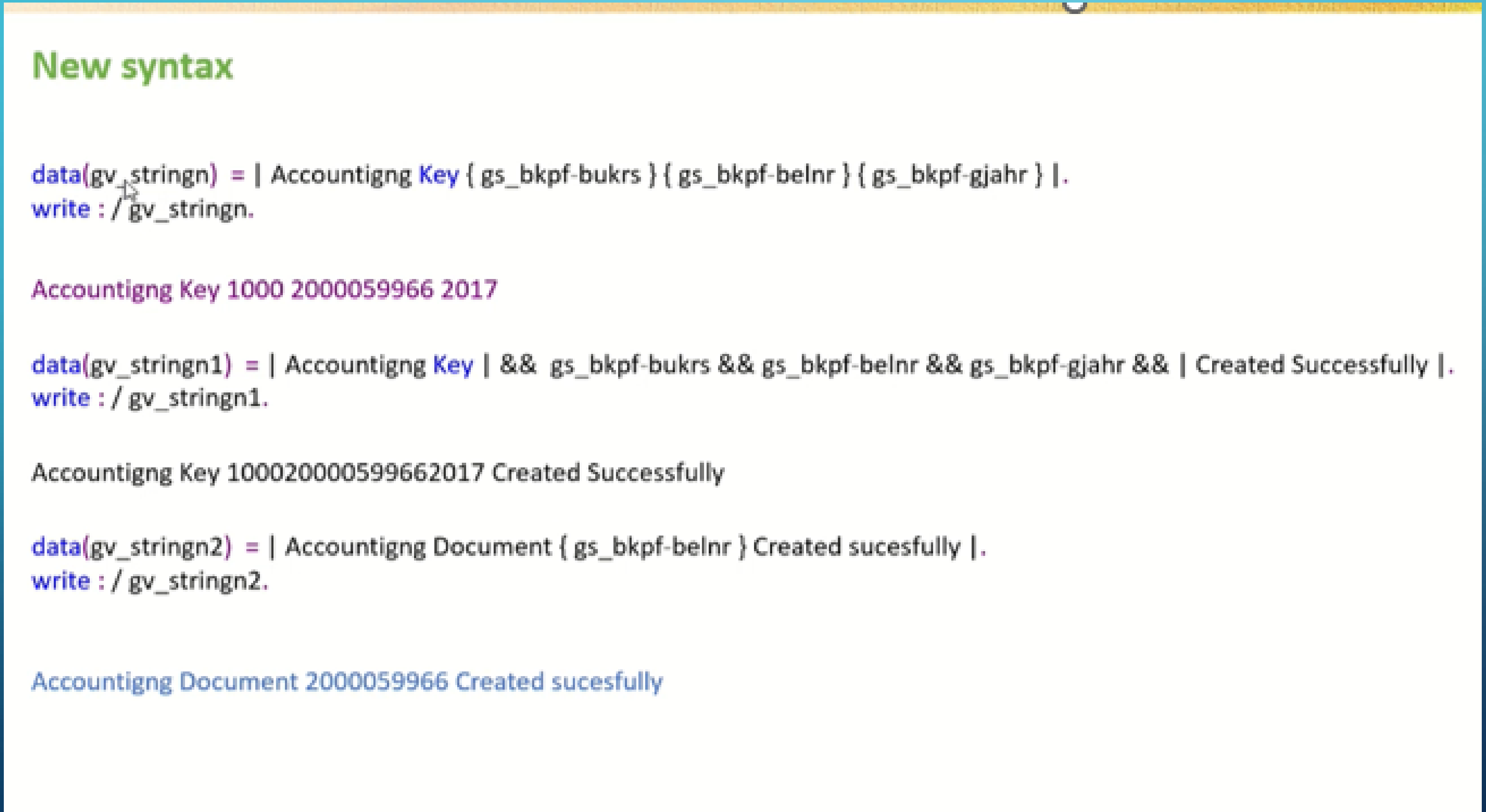
```
data(gs_booking1) = value #( gt_booking[ gv_lines ] optional ).
```

USING READ TABLES:



```
select * from bkpf into TABLE @data(gt_bkpf1) WHERE bukrs in @s_bukrs
                                                and belnr in @s_belnr
                                                and gjahr in @s_gjahr.

sort gt_bkpf1 by bukrs belnr gjahr.

if gt_bkpf1[] is NOT INITIAL.
"line items data
 select * from bseg into TABLE @data(gt_bseg) FOR ALL ENTRIES IN @gt_bkpf1
 WHERE bukrs = @gt_bkpf1-bukrs
   and belnr = @gt_bkpf1-belnr
   and gjahr = @gt_bkpf1-gjahr.

  loop at gt_bkpf1 into data(gs_bkpf1).


  try.
    data(gs_bseg) = gt_bseg[ bukrs = gs_bkpf1-bukrs
                             belnr = gs_bkpf1-belnr
                             gjahr = gs_bkpf1-gjahr ].
  CATCH cx_root.
  ENDTRY.


  clear : gs_bkpf1.
  endloop.

endif.
```

NEW CONCATENATE SYNTAX:



New syntax

```
data(gv_stringn)  = | Accountigng Key { gs_bkpf-bukrs } { gs_bkpf-belnr } { gs_bkpf-gjahr } |.
write : / gv_stringn.
```

```
Accountigng Key 1000 2000059966 2017
```

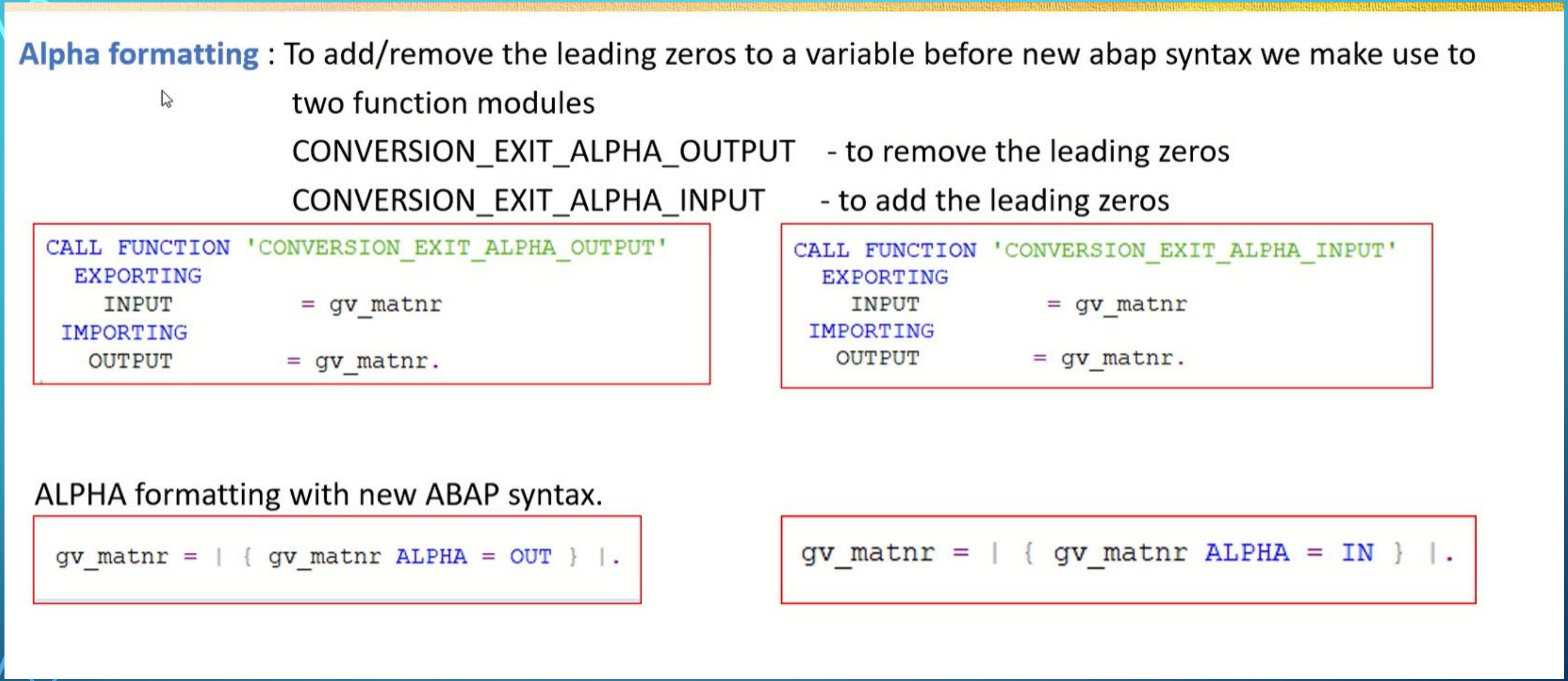
```
data(gv_stringn1)  = | Accountigng Key | &&  gs_bkpf-bukrs && gs_bkpf-belnr && gs_bkpf-gjahr && | Created Successfully |.
write : / gv_stringn1.
```

```
Accountigng Key 100020000599662017 Created Successfully
```

```
data(gv_stringn2)  = | Accountigng Document { gs_bkpf-belnr } Created sucesfully |.
write : / gv_stringn2.
```

```
Accountigng Document 2000059966 Created sucesfully
```

FORMATTING



Alpha formatting : To add/remove the leading zeros to a variable before new abap syntax we make use to two function modules

CONVERSION_EXIT_ALPHA_OUTPUT - to remove the leading zeros

CONVERSION_EXIT_ALPHA_INPUT - to add the leading zeros

```
CALL FUNCTION 'CONVERSION_EXIT_ALPHA_OUTPUT'
  EXPORTING
    INPUT            = gv_matnr
 IMPORTING
   OUTPUT           = gv_matnr.
```

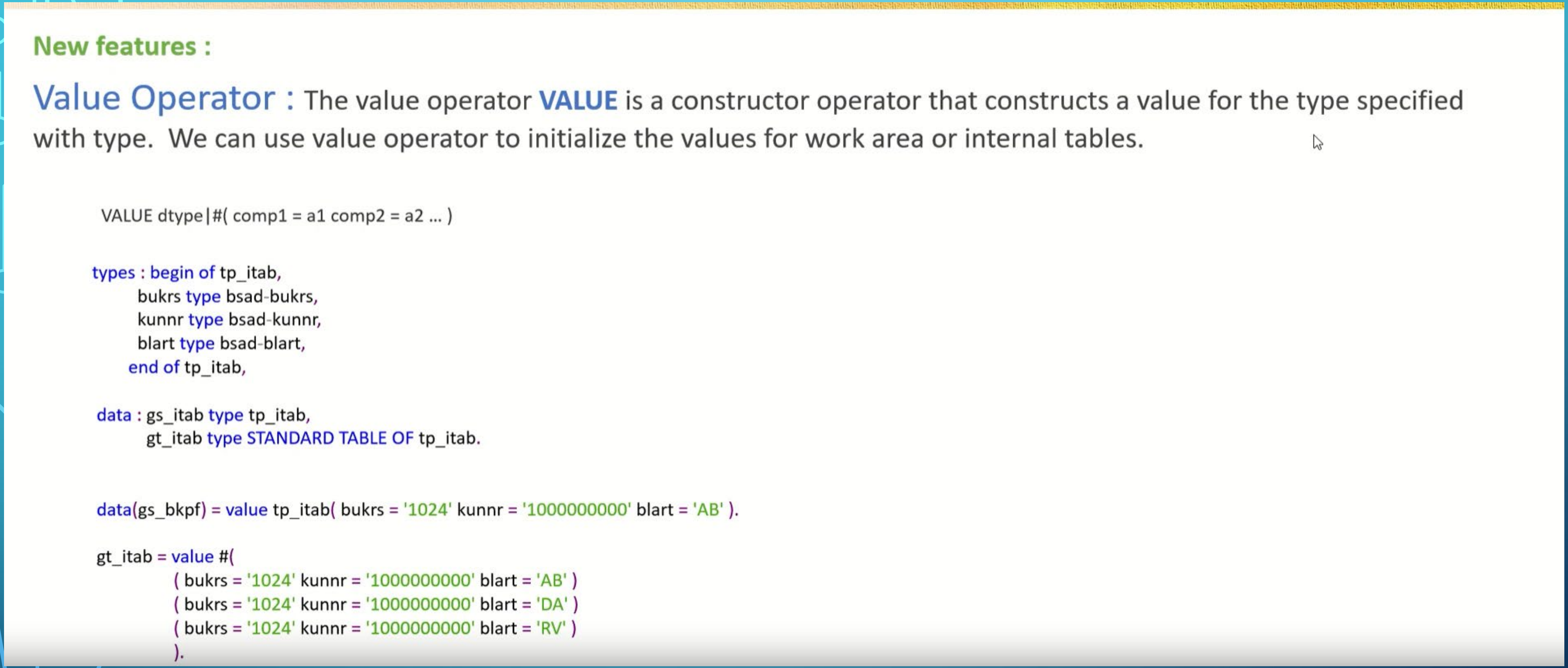
```
CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
  EXPORTING
    INPUT            = gv_matnr
 IMPORTING
   OUTPUT           = gv_matnr.
```

ALPHA formatting with new ABAP syntax.

```
gv_matnr = | { gv_matnr ALPHA = OUT } |.
```

```
gv_matnr = | { gv_matnr ALPHA = IN } |.
```

VALUE OPERATOR

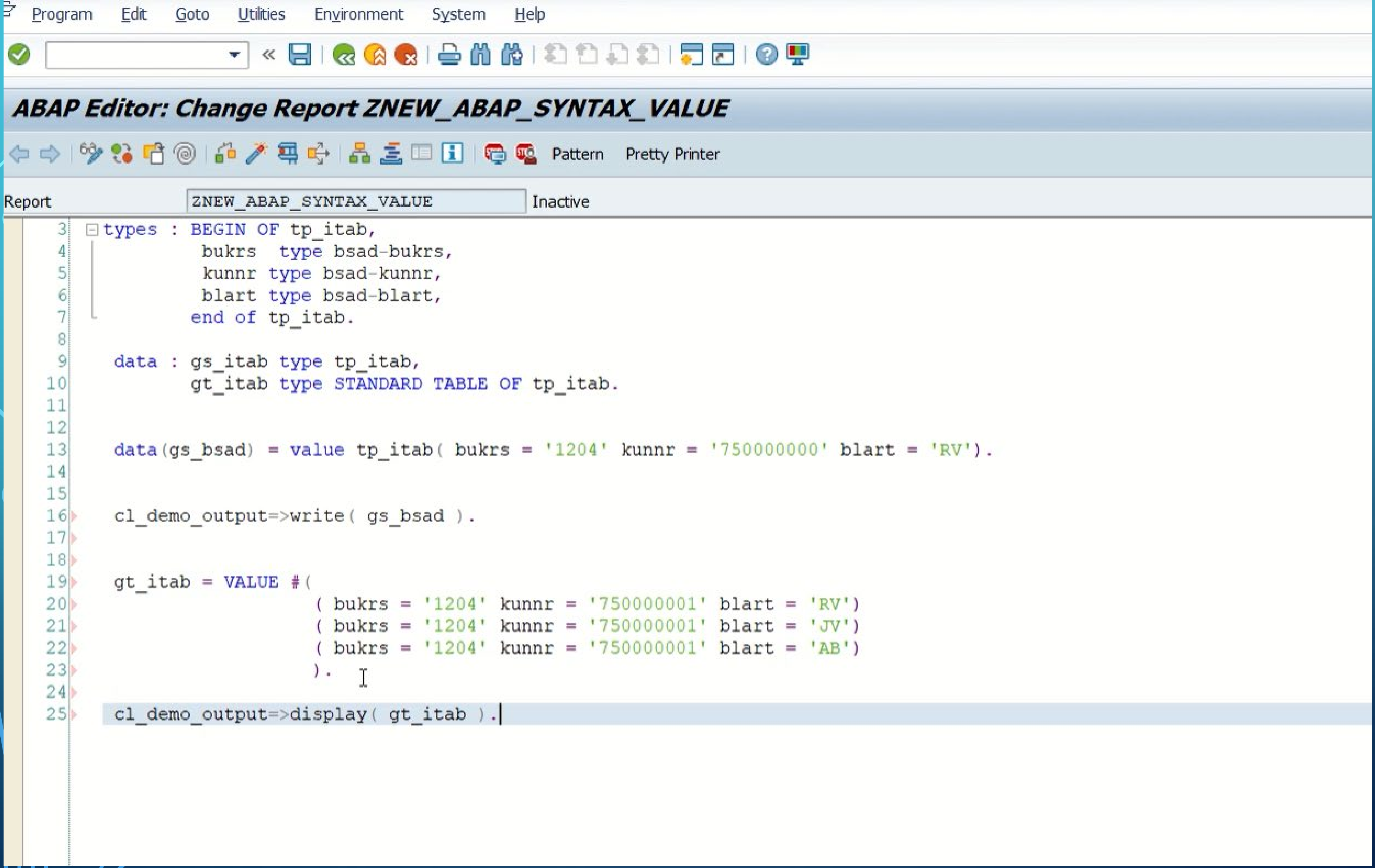


New features :

Value Operator : The value operator **VALUE** is a constructor operator that constructs a value for the type specified with type. We can use value operator to initialize the values for work area or internal tables.

```
VALUE dtype|#( comp1 = a1 comp2 = a2 ... )


types : begin of tp_itab,
      bukrs type bsad-bukrs,
      kunnr type bsad-kunnr,
      blart type bsad-blart,
    end of tp_itab,

data : gs_itab type tp_itab,
       gt_itab type STANDARD TABLE OF tp_itab.


data(gs_bkpf) = value tp_itab( bukrs = '1024' kunnr = '1000000000' blart = 'AB' ).

gt_itab = value #(
            ( bukrs = '1024' kunnr = '1000000000' blart = 'AB' )
            ( bukrs = '1024' kunnr = '1000000000' blart = 'DA' )
            ( bukrs = '1024' kunnr = '1000000000' blart = 'RV' )
            ).
```

VALUE OPERATOR

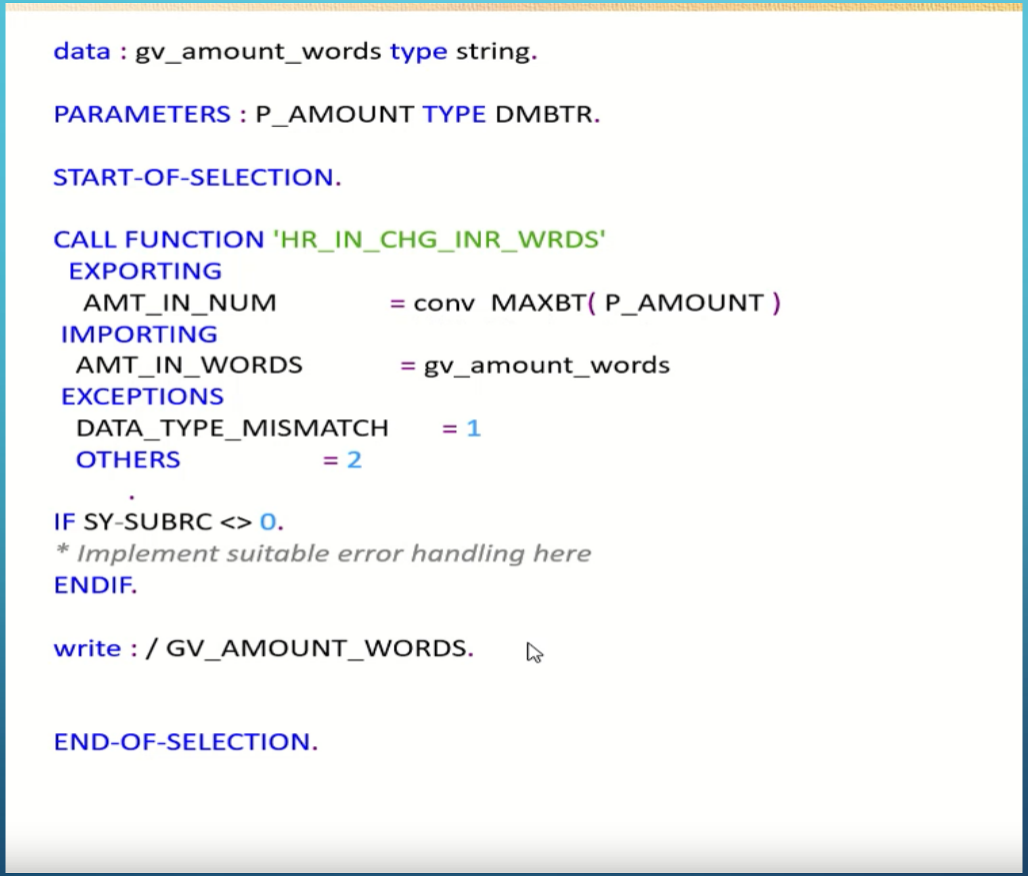


ABAP Editor: Change Report ZNEW_ABAP_SYNTAX_VALUE

Report ZNEW_ABAP_SYNTAX_VALUE Inactive

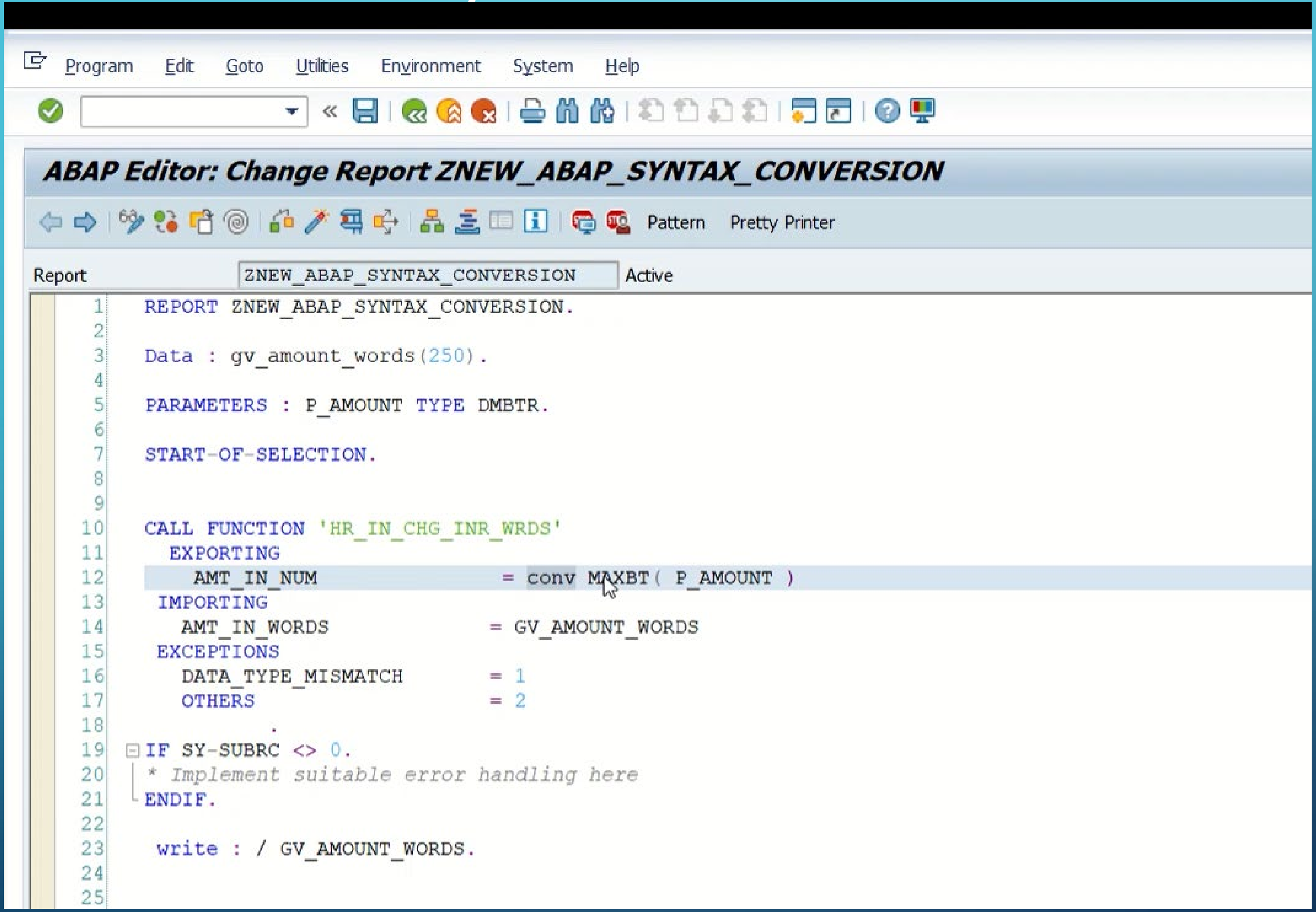
```
types : BEGIN OF tp_itab,
         bukrs  type bsad-bukrs,
         kunnr type bsad-kunnr,
         blart type bsad-blart,
        end of tp_itab.

 data : gs_itab type tp_itab,
        gt_itab type STANDARD TABLE OF tp_itab.


 data(gs_bsad) = value tp_itab( bukrs = '1204' kunnr = '750000000' blart = 'RV').


 cl_demo_output=>write( gs_bsad ).


 gt_itab = VALUE #(
                   ( bukrs = '1204' kunnr = '750000001' blart = 'RV')
                   ( bukrs = '1204' kunnr = '750000001' blart = 'JV')
                   ( bukrs = '1204' kunnr = '750000001' blart = 'AB')
                   ).

 cl_demo_output=>display( gt_itab ).
```

CONV OPERATOR



```
data : gv_amount_words type string.

PARAMETERS : P_AMOUNT TYPE DMBTR.

START-OF-SELECTION.

CALL FUNCTION 'HR_IN_CHG_INR_WRDS'
  EXPORTING
    AMT_IN_NUM             = conv  MAXBT( P_AMOUNT )
 IMPORTING
   AMT_IN_WORDS            = gv_amount_words
 EXCEPTIONS
   DATA_TYPE_MISMATCH      = 1
   OTHERS              = 2
          .
IF SY-SUBRC <> 0.
* Implement suitable error handling here
ENDIF.

write : / GV_AMOUNT_WORDS.


END-OF-SELECTION.
```

CONV OPERATOR)

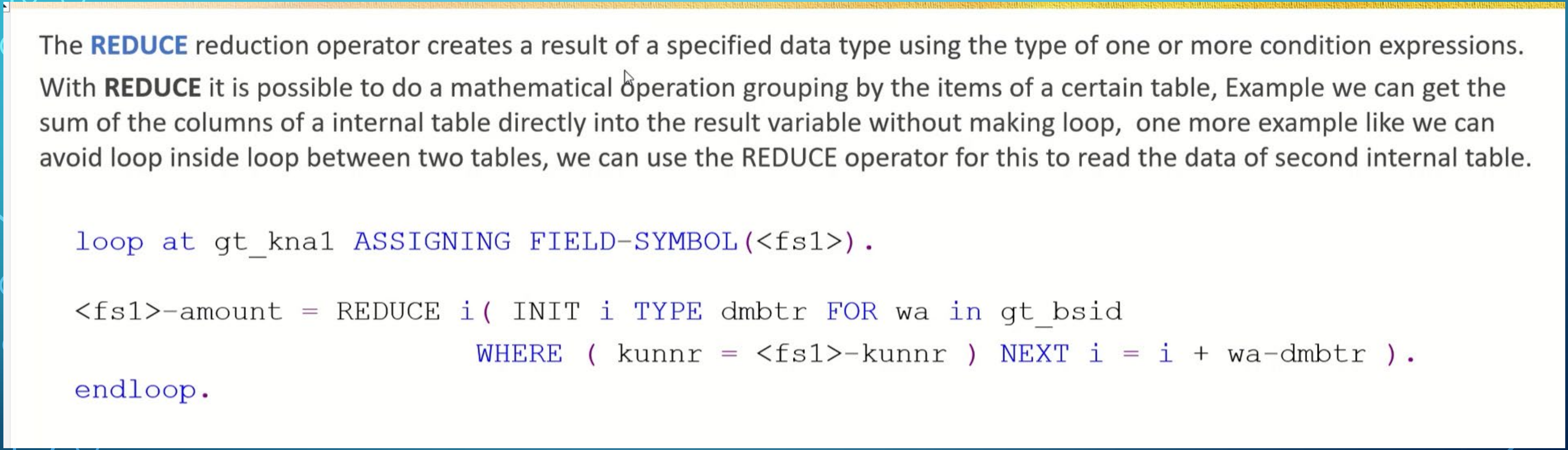


ABAP Editor: Change Report ZNEW_ABAP_SYNTAX_CONVERSION

Report ZNEW_ABAP_SYNTAX_CONVERSION Active

```
REPORT ZNEW_ABAP_SYNTAX_CONVERSION.

Data : gv_amount_words(250).

PARAMETERS : P_AMOUNT TYPE DMBTR.

START-OF-SELECTION.


CALL FUNCTION 'HR_IN_CHG_INR_WRDS'
  EXPORTING
    AMT_IN_NUM                   = conv MAXBT( P_AMOUNT )
 IMPORTING
   AMT_IN_WORDS                = GV_AMOUNT_WORDS
 EXCEPTIONS
   DATA_TYPE_MISMATCH          = 1
   OTHERS                      = 2
          .
IF SY-SUBRC <> 0.
* Implement suitable error handling here
ENDIF.

 write : / GV_AMOUNT_WORDS.
```

REDUCE OPERATOR

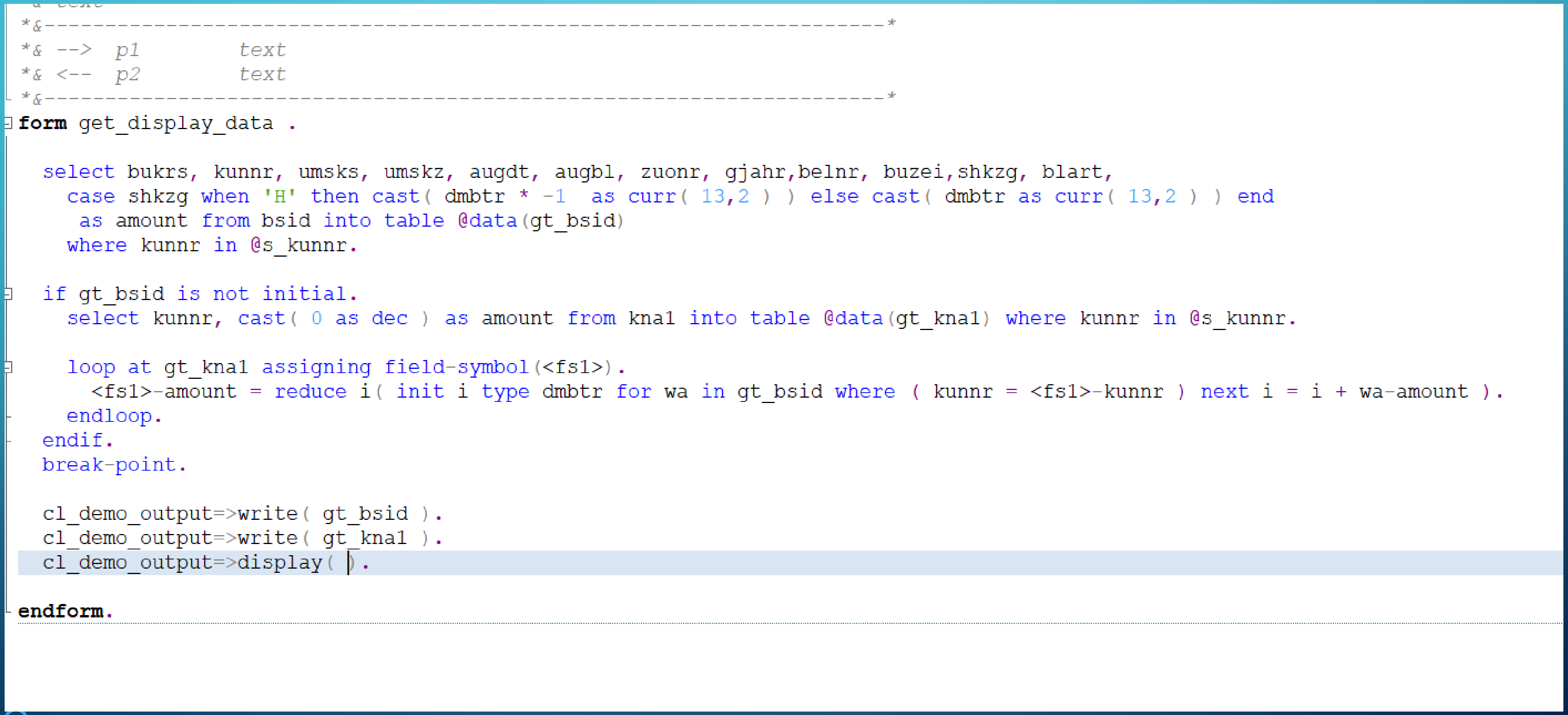


The **REDUCE** reduction operator creates a result of a specified data type using the type of one or more condition expressions.

With **REDUCE** it is possible to do a mathematical operation grouping by the items of a certain table, Example we can get the sum of the columns of a internal table directly into the result variable without making loop, one more example like we can avoid loop inside loop between two tables, we can use the REDUCE operator for this to read the data of second internal table.

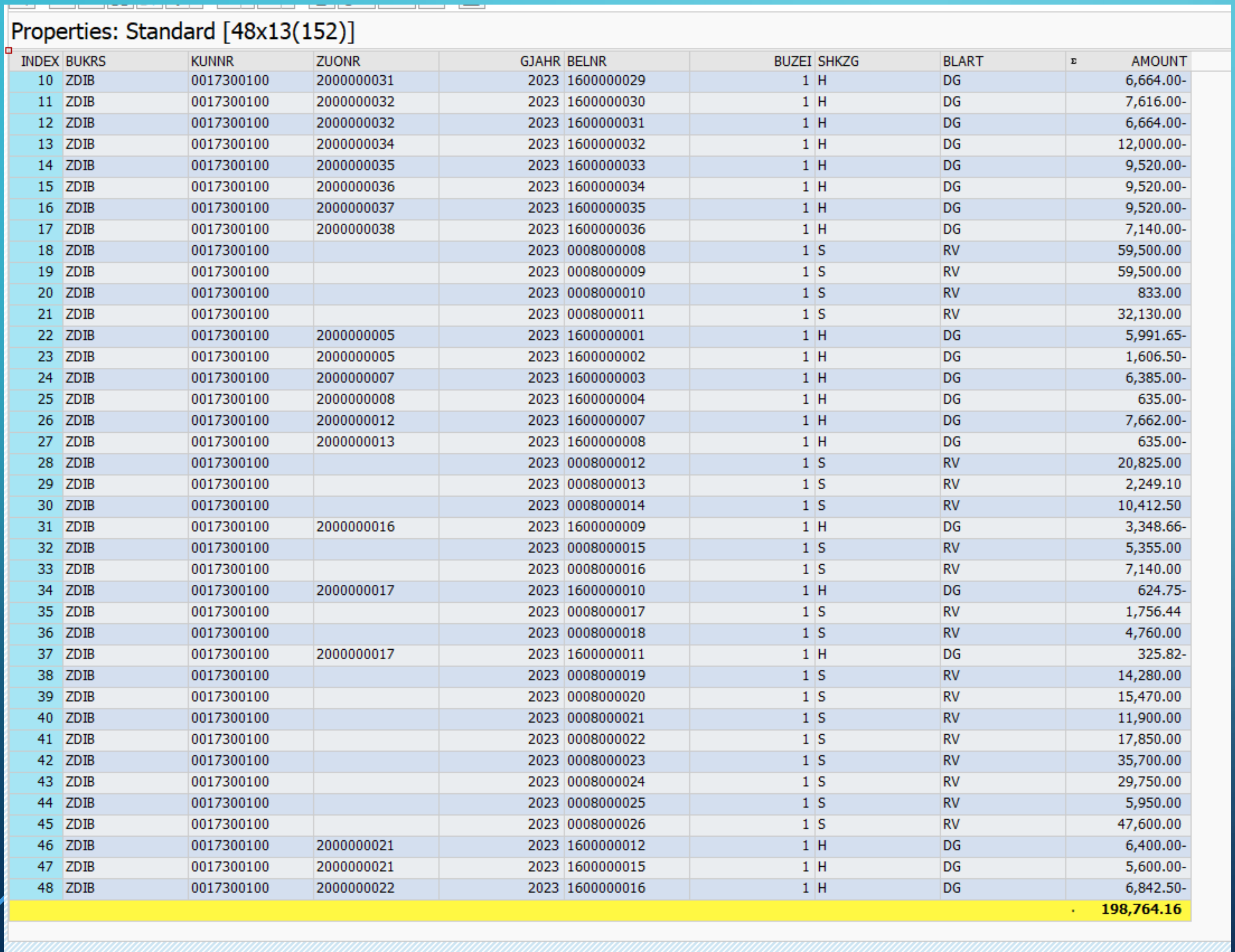
```
loop at gt_kna1 ASSIGNING FIELD-SYMBOL(<fs1>).

<fs1>-amount = REDUCE i( INIT i TYPE dmbtr FOR wa in gt_bsid
                       WHERE ( kunnr = <fs1>-kunnr ) NEXT i = i + wa-dmbtr ).
endloop.
```

REDUCE OPERATOR (USING ALSO CAST OPERATOR)



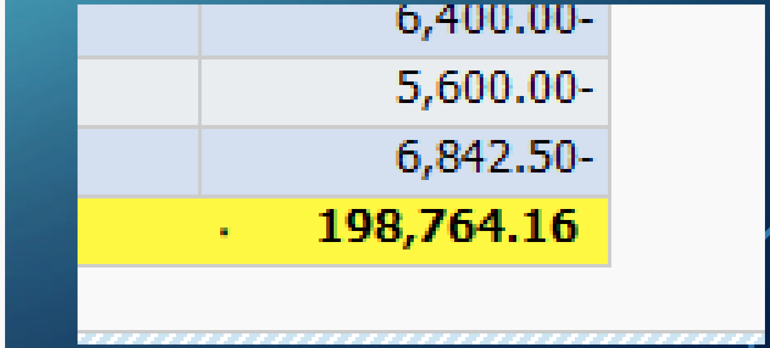
```
*& text
*&---------------------------------------------------------------------*
*& --> p1          text
*& <-- p2          text
*&---------------------------------------------------------------------*
form get_display_data .

  select bukrs, kunnr, umsks, umskz, augdt, augbl, zuonr, gjahr,belnr, buzei,shkzg, blart,
    case shkzg when 'H' then cast( dmbtr * -1  as curr( 13,2 ) ) else cast( dmbtr as curr( 13,2 ) ) end
     as amount from bsid into table @data(gt_bsid)
    where kunnr in @s_kunnr.

  if gt_bsid is not initial.
    select kunnr, cast( 0 as dec ) as amount from kna1 into table @data(gt_kna1) where kunnr in @s_kunnr.

    loop at gt_kna1 assigning field-symbol(<fs1>).
      <fs1>-amount = reduce i( init i type dmbtr for wa in gt_bsid where ( kunnr = <fs1>-kunnr ) next i = i + wa-amount ).
    endloop.
  endif.
  break-point.

  cl_demo_output=>write( gt_bsid ).
  cl_demo_output=>write( gt_kna1 ).
  cl_demo_output=>display( ).

endform.
```

REDUCE OPERATOR (USING ALSO CAST OPERATOR) – FROM DEBUGGER BEFORE REDUCE



Properties: Standard [48x13(152)]

INDEX	BUKRS	KUNNR	ZUONR	GJAHR	BELNR	BUZEI	SHKZG	BLART	Σ	AMOUNT
10	ZDIB	0017300100	2000000031	2023	1600000029	1	H	DG		6,664.00-
11	ZDIB	0017300100	2000000032	2023	1600000030	1	H	DG		7,616.00-
12	ZDIB	0017300100	2000000032	2023	1600000031	1	H	DG		6,664.00-
13	ZDIB	0017300100	2000000034	2023	1600000032	1	H	DG		12,000.00-
14	ZDIB	0017300100	2000000035	2023	1600000033	1	H	DG		9,520.00-
15	ZDIB	0017300100	2000000036	2023	1600000034	1	H	DG		9,520.00-
16	ZDIB	0017300100	2000000037	2023	1600000035	1	H	DG		9,520.00-
17	ZDIB	0017300100	2000000038	2023	1600000036	1	H	DG		7,140.00-
18	ZDIB	0017300100		2023	0008000008	1	S	RV		59,500.00
19	ZDIB	0017300100		2023	0008000009	1	S	RV		59,500.00
20	ZDIB	0017300100		2023	0008000010	1	S	RV		833.00
21	ZDIB	0017300100		2023	0008000011	1	S	RV		32,130.00
22	ZDIB	0017300100	2000000005	2023	1600000001	1	H	DG		5,991.65-
23	ZDIB	0017300100	2000000005	2023	1600000002	1	H	DG		1,606.50-
24	ZDIB	0017300100	2000000007	2023	1600000003	1	H	DG		6,385.00-
25	ZDIB	0017300100	2000000008	2023	1600000004	1	H	DG		635.00-
26	ZDIB	0017300100	2000000012	2023	1600000007	1	H	DG		7,662.00-
27	ZDIB	0017300100	2000000013	2023	1600000008	1	H	DG		635.00-
28	ZDIB	0017300100		2023	0008000012	1	S	RV		20,825.00
29	ZDIB	0017300100		2023	0008000013	1	S	RV		2,249.10
30	ZDIB	0017300100		2023	0008000014	1	S	RV		10,412.50
31	ZDIB	0017300100	2000000016	2023	1600000009	1	H	DG		3,348.66-
32	ZDIB	0017300100		2023	0008000015	1	S	RV		5,355.00
33	ZDIB	0017300100		2023	0008000016	1	S	RV		7,140.00
34	ZDIB	0017300100	2000000017	2023	1600000010	1	H	DG		624.75-
35	ZDIB	0017300100		2023	0008000017	1	S	RV		1,756.44
36	ZDIB	0017300100		2023	0008000018	1	S	RV		4,760.00
37	ZDIB	0017300100	2000000017	2023	1600000011	1	H	DG		325.82-
38	ZDIB	0017300100		2023	0008000019	1	S	RV		14,280.00
39	ZDIB	0017300100		2023	0008000020	1	S	RV		15,470.00
40	ZDIB	0017300100		2023	0008000021	1	S	RV		11,900.00
41	ZDIB	0017300100		2023	0008000022	1	S	RV		17,850.00
42	ZDIB	0017300100		2023	0008000023	1	S	RV		35,700.00
43	ZDIB	0017300100		2023	0008000024	1	S	RV		29,750.00
44	ZDIB	0017300100		2023	0008000025	1	S	RV		5,950.00
45	ZDIB	0017300100		2023	0008000026	1	S	RV		47,600.00
46	ZDIB	0017300100	2000000021	2023	1600000012	1	H	DG		6,400.00-
47	ZDIB	0017300100	2000000021	2023	1600000015	1	H	DG		5,600.00-
48	ZDIB	0017300100	2000000022	2023	1600000016	1	H	DG		6,842.50-
									·	198,764.16



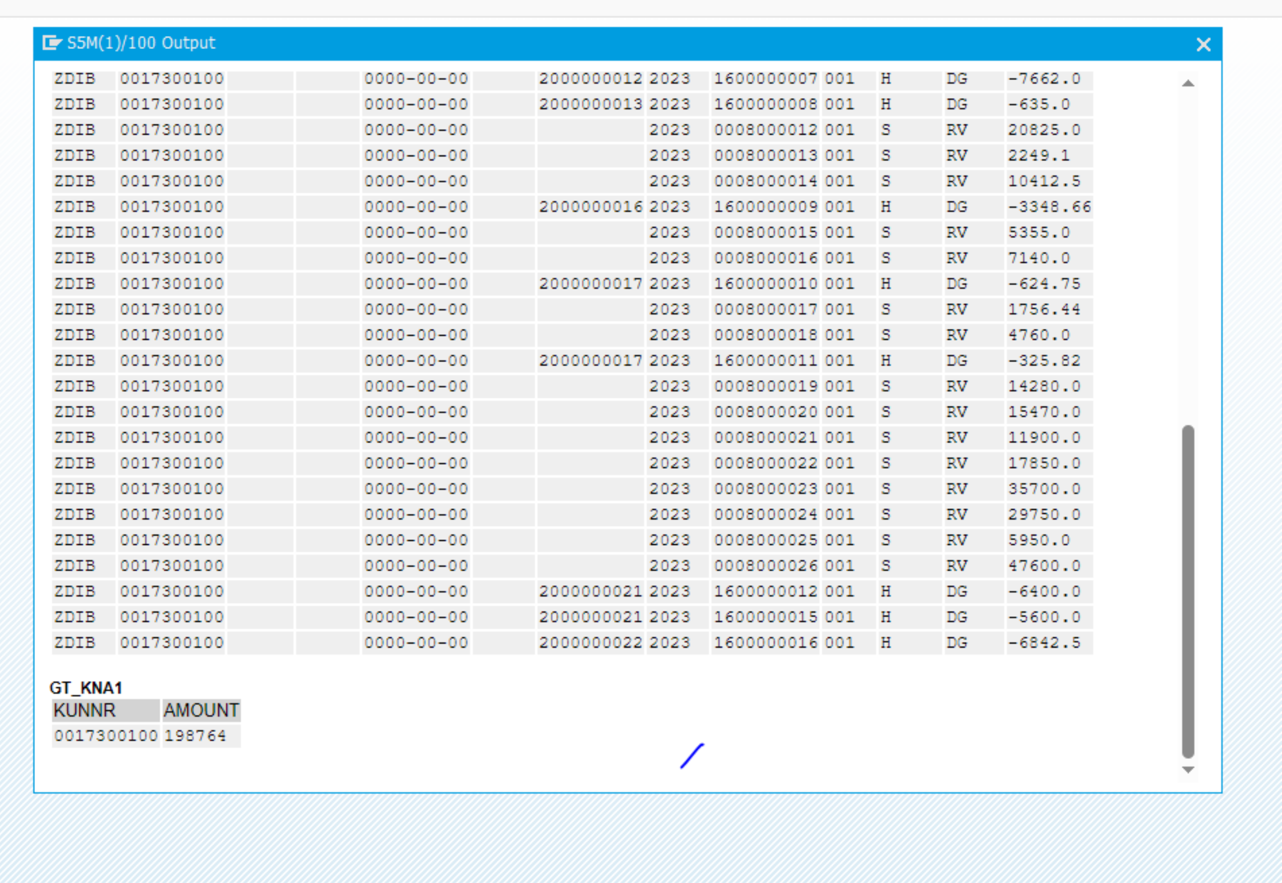
6,400.00-

5,600.00-

6,842.50-

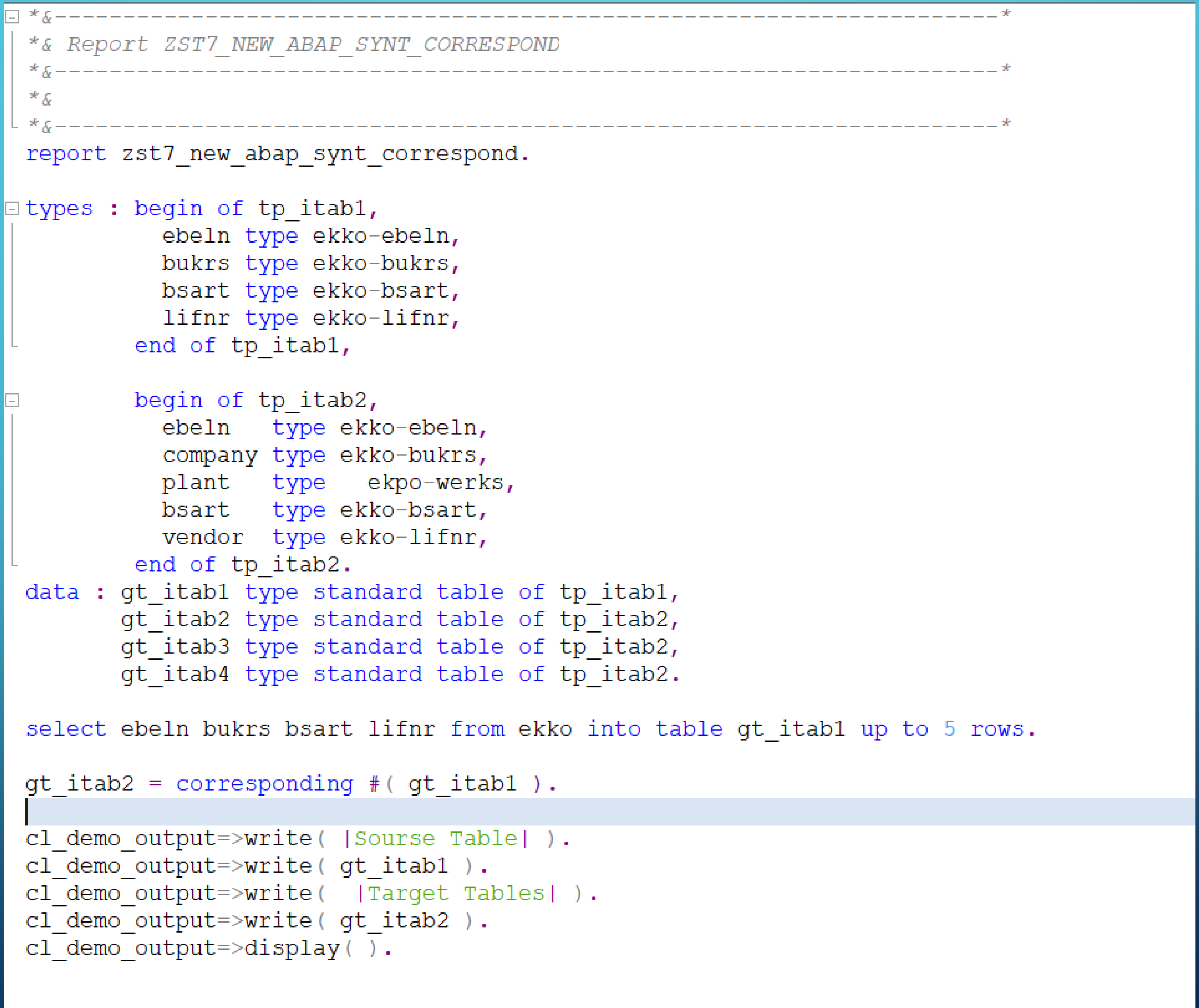
· 198,764.16

REDUCE OPERATOR (USING ALSO CAST OPERATOR) – RESULT



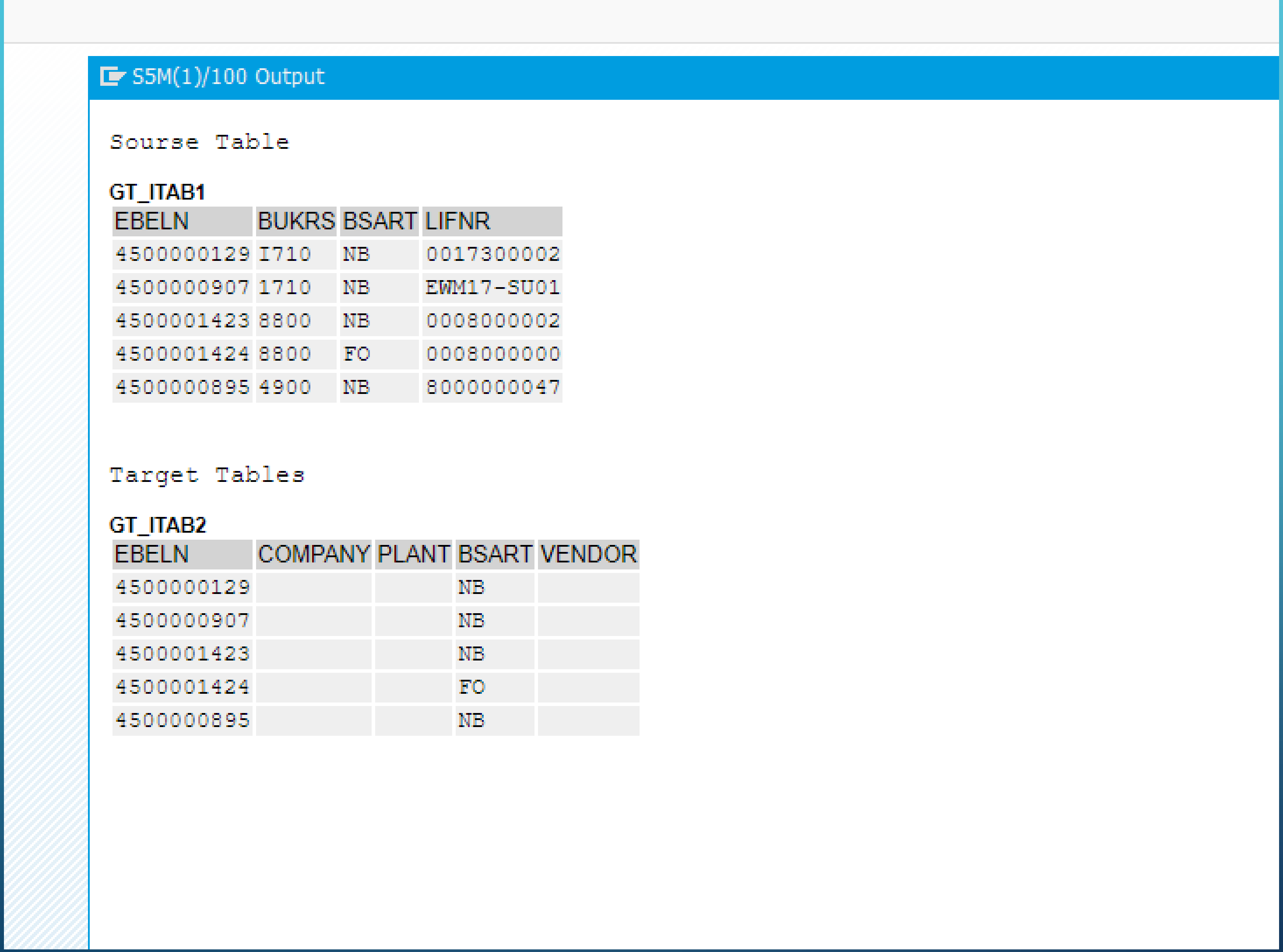
```
ZDIB  0017300100      0000-00-00  2000000012 2023  1600000007 001  H  DG  -7662.0
ZDIB  0017300100      0000-00-00  2000000013 2023  1600000008 001  H  DG  -635.0
ZDIB  0017300100      0000-00-00             2023  0008000012 001  S  RV  20825.0
ZDIB  0017300100      0000-00-00             2023  0008000013 001  S  RV  2249.1
ZDIB  0017300100      0000-00-00             2023  0008000014 001  S  RV  10412.5
ZDIB  0017300100      0000-00-00  2000000016 2023  1600000009 001  H  DG  -3348.66
ZDIB  0017300100      0000-00-00             2023  0008000015 001  S  RV  5355.0
ZDIB  0017300100      0000-00-00             2023  0008000016 001  S  RV  7140.0
ZDIB  0017300100      0000-00-00  2000000017 2023  1600000010 001  H  DG  -624.75
ZDIB  0017300100      0000-00-00             2023  0008000017 001  S  RV  1756.44
ZDIB  0017300100      0000-00-00             2023  0008000018 001  S  RV  4760.0
ZDIB  0017300100      0000-00-00  2000000017 2023  1600000011 001  H  DG  -325.82
ZDIB  0017300100      0000-00-00             2023  0008000019 001  S  RV  14280.0
ZDIB  0017300100      0000-00-00             2023  0008000020 001  S  RV  15470.0
ZDIB  0017300100      0000-00-00             2023  0008000021 001  S  RV  11900.0
ZDIB  0017300100      0000-00-00             2023  0008000022 001  S  RV  17850.0
ZDIB  0017300100      0000-00-00             2023  0008000023 001  S  RV  35700.0
ZDIB  0017300100      0000-00-00             2023  0008000024 001  S  RV  29750.0
ZDIB  0017300100      0000-00-00             2023  0008000025 001  S  RV  5950.0
ZDIB  0017300100      0000-00-00             2023  0008000026 001  S  RV  47600.0
ZDIB  0017300100      0000-00-00  2000000021 2023  1600000012 001  H  DG  -6400.0
ZDIB  0017300100      0000-00-00  2000000021 2023  1600000015 001  H  DG  -5600.0
ZDIB  0017300100      0000-00-00  2000000022 2023  1600000016 001  H  DG  -6842.5

GT_KNA1
KUNNR      AMOUNT
0017300100 198764
```

CORRESPONDING OPERATOR



```
*&---------------------------------------------------------------------*
*& Report ZST7_NEW_ABAP_SYNT_CORRESPOND
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
report zst7_new_abap_synt_correspond.

types : begin of tp_itab1,
          ebeln type ekko-ebeln,
          bukrs type ekko-bukrs,
          bsart type ekko-bsart,
          lifnr type ekko-lifnr,
        end of tp_itab1,

        begin of tp_itab2,
          ebeln   type ekko-ebeln,
          company type ekko-bukrs,
          plant   type   ekpo-werks,
          bsart   type ekko-bsart,
          vendor  type ekko-lifnr,
        end of tp_itab2.
data : gt_itab1 type standard table of tp_itab1,
       gt_itab2 type standard table of tp_itab2,
       gt_itab3 type standard table of tp_itab2,
       gt_itab4 type standard table of tp_itab2.

select ebeln bukrs bsart lifnr from ekko into table gt_itab1 up to 5 rows.

gt_itab2 = corresponding #( gt_itab1 ).

cl_demo_output=>write( |Sourse Table| ).
cl_demo_output=>write( gt_itab1 ).
cl_demo_output=>write(  |Target Tables| ).
cl_demo_output=>write( gt_itab2 ).
cl_demo_output=>display( ).
```

CORRESPONDING OPERATOR – RESULT:



S5M(1)/100 Output

```
Sourse Table
```

GT_ITAB1

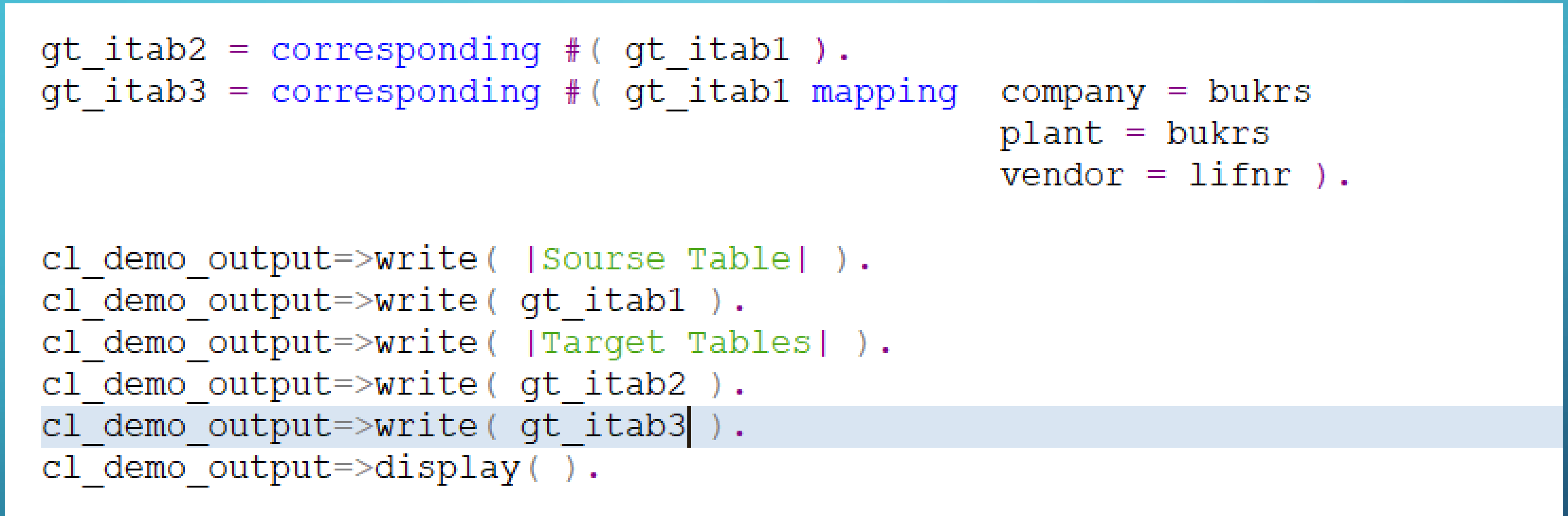
EBELN	BUKRS	BSART	LIFNR
4500000129	I710	NB	0017300002
4500000907	1710	NB	EWM17-SU01
4500001423	8800	NB	0008000002
4500001424	8800	FO	0008000000
4500000895	4900	NB	8000000047

```
Target Tables
```

GT_ITAB2

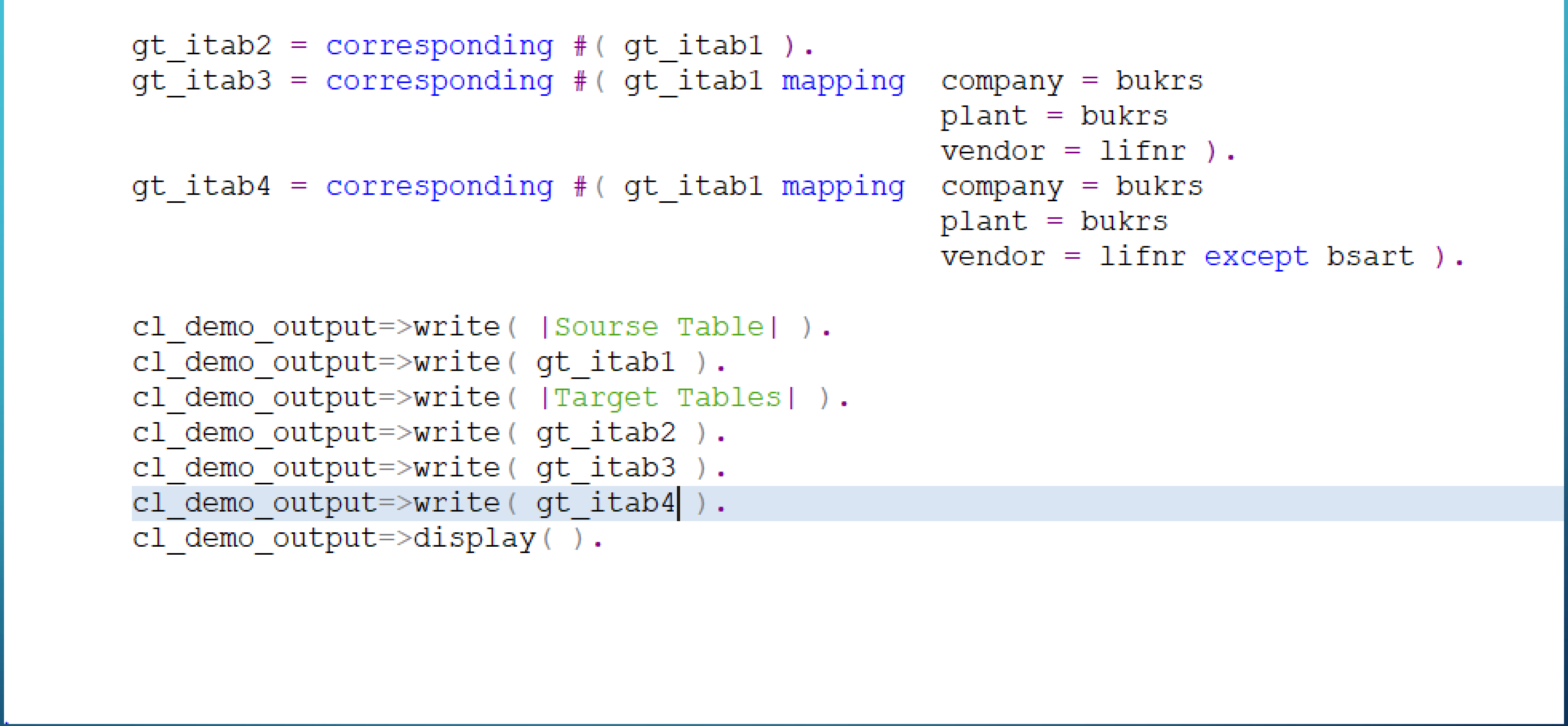
EBELN	COMPANY	PLANT	BSART	VENDOR
4500000129			NB	
4500000907			NB	
4500001423			NB	
4500001424			FO	
4500000895			NB	

CORRESPONDING OPERATOR



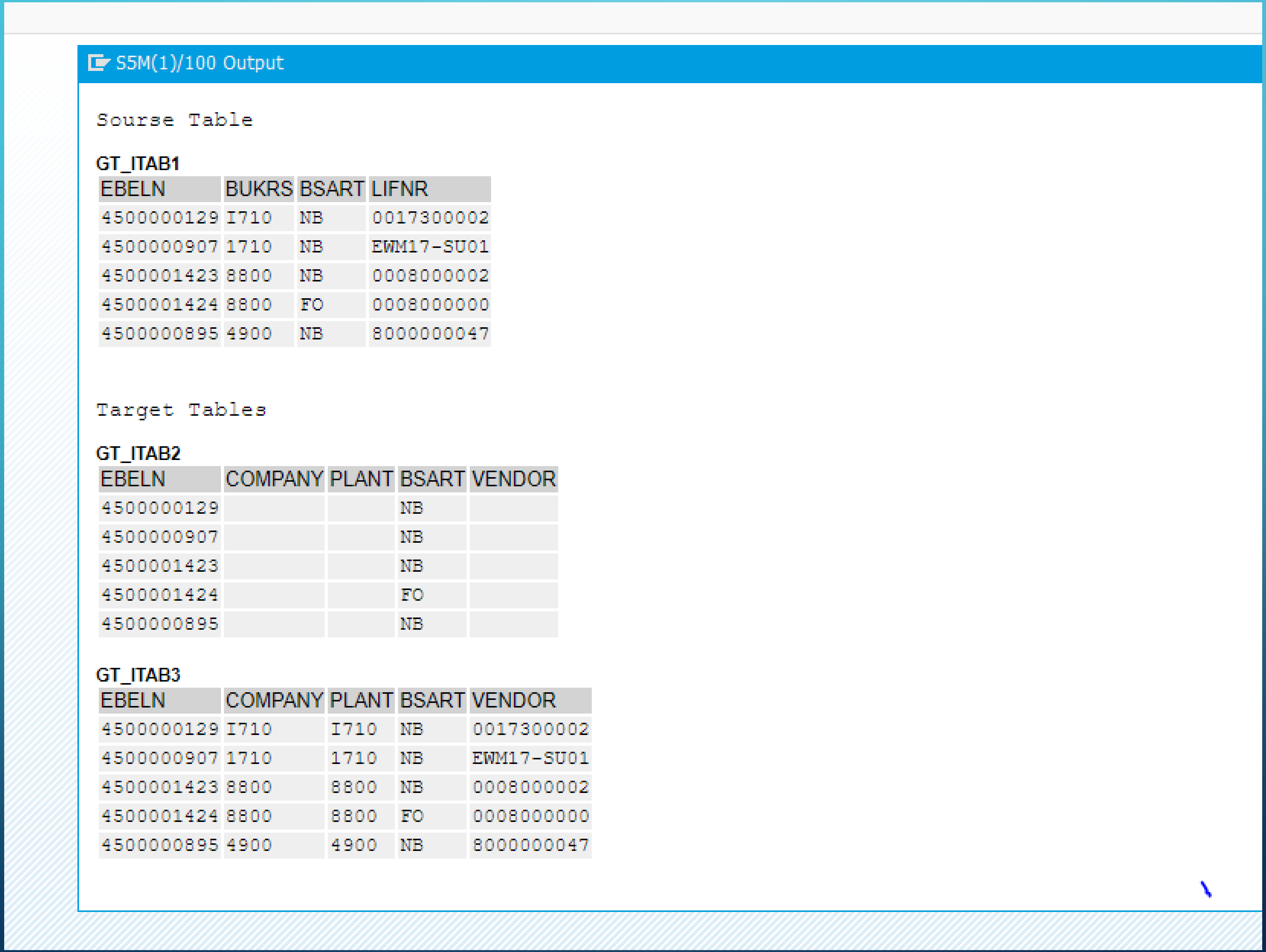
```
gt_itab2 = corresponding #( gt_itab1 ).
gt_itab3 = corresponding #( gt_itab1 mapping  company = bukrs
                                              plant = bukrs
                                              vendor = lifnr ).

cl_demo_output=>write( |Sourse Table| ).
cl_demo_output=>write( gt_itab1 ).
cl_demo_output=>write( |Target Tables| ).
cl_demo_output=>write( gt_itab2 ).
cl_demo_output=>write( gt_itab3 ).
cl_demo_output=>display( ).
```

CORRESPONDING OPERATOR



```
gt_itab2 = corresponding #( gt_itab1 ).
gt_itab3 = corresponding #( gt_itab1 mapping  company = bukrs
                                              plant = bukrs
                                              vendor = lifnr ).
gt_itab4 = corresponding #( gt_itab1 mapping  company = bukrs
                                              plant = bukrs
                                              vendor = lifnr except bsart ).

cl_demo_output=>write( |Sourse Table| ).
cl_demo_output=>write( gt_itab1 ).
cl_demo_output=>write( |Target Tables| ).
cl_demo_output=>write( gt_itab2 ).
cl_demo_output=>write( gt_itab3 ).
cl_demo_output=>write( gt_itab4 ).
cl_demo_output=>display( ).
```

CORRESPONDING OPERATOR – RESULT:



S5M(1)/100 Output

Sourse Table

GT_ITAB1

EBELN	BUKRS	BSART	LIFNR
4500000129	I710	NB	0017300002
4500000907	1710	NB	EWM17-SU01
4500001423	8800	NB	0008000002
4500001424	8800	FO	0008000000
4500000895	4900	NB	8000000047

Target Tables

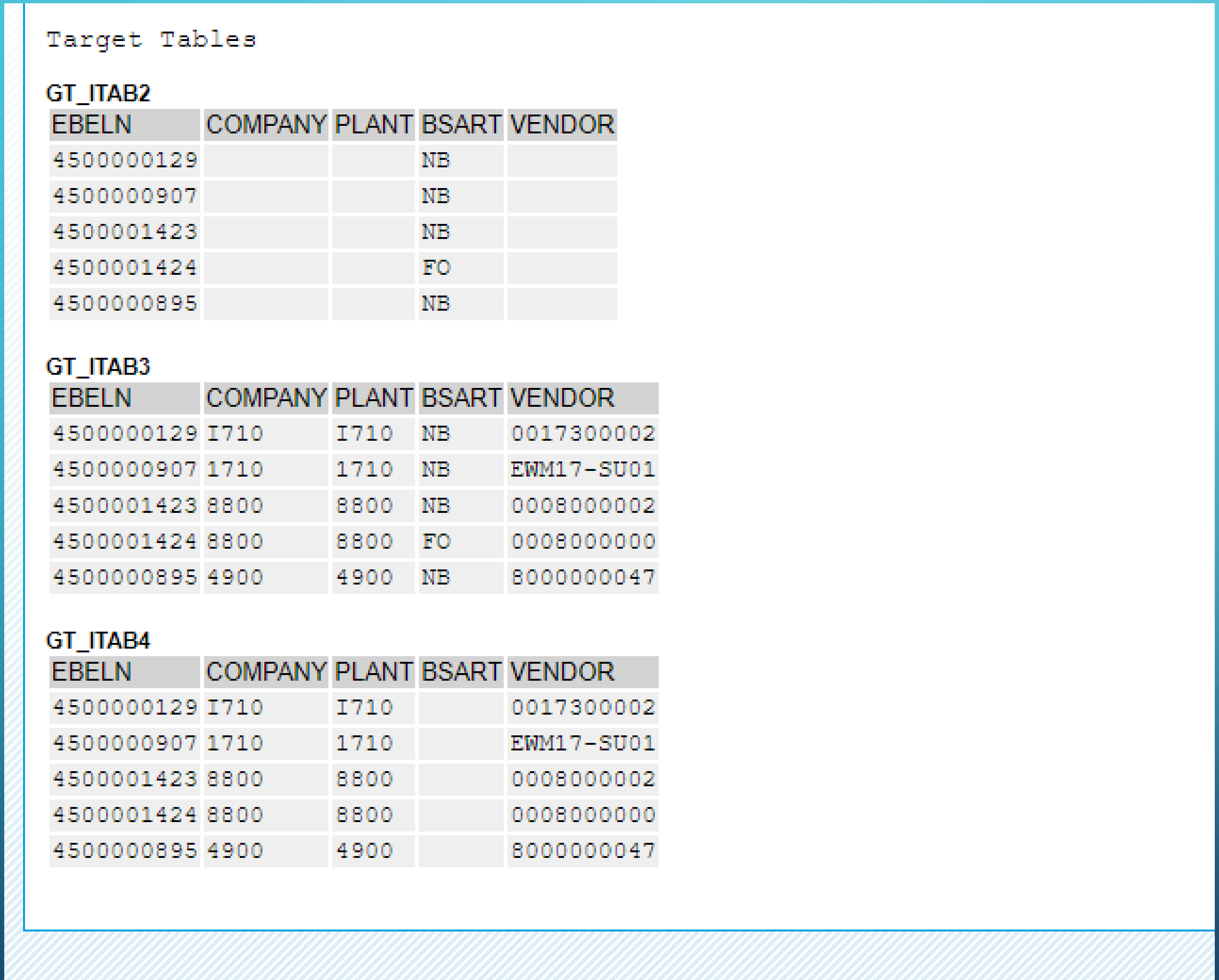
GT_ITAB2

EBELN	COMPANY	PLANT	BSART	VENDOR
4500000129			NB	
4500000907			NB	
4500001423			NB	
4500001424			FO	
4500000895			NB	

GT_ITAB3

EBELN	COMPANY	PLANT	BSART	VENDOR
4500000129	I710	I710	NB	0017300002
4500000907	1710	1710	NB	EWM17-SU01
4500001423	8800	8800	NB	0008000002
4500001424	8800	8800	FO	0008000000
4500000895	4900	4900	NB	8000000047

CORRESPONDING OPERATOR – RESULT:



Target Tables

GT_ITAB2

EBELN	COMPANY	PLANT	BSART	VENDOR
4500000129			NB	
4500000907			NB	
4500001423			NB	
4500001424			FO	
4500000895			NB	

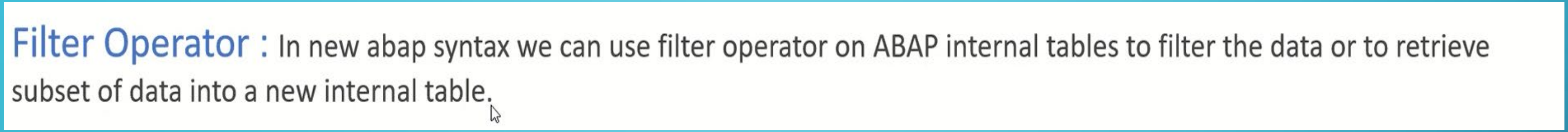
GT_ITAB3

EBELN	COMPANY	PLANT	BSART	VENDOR
4500000129	I710	I710	NB	0017300002
4500000907	1710	1710	NB	EWM17-SU01
4500001423	8800	8800	NB	0008000002
4500001424	8800	8800	FO	0008000000
4500000895	4900	4900	NB	8000000047

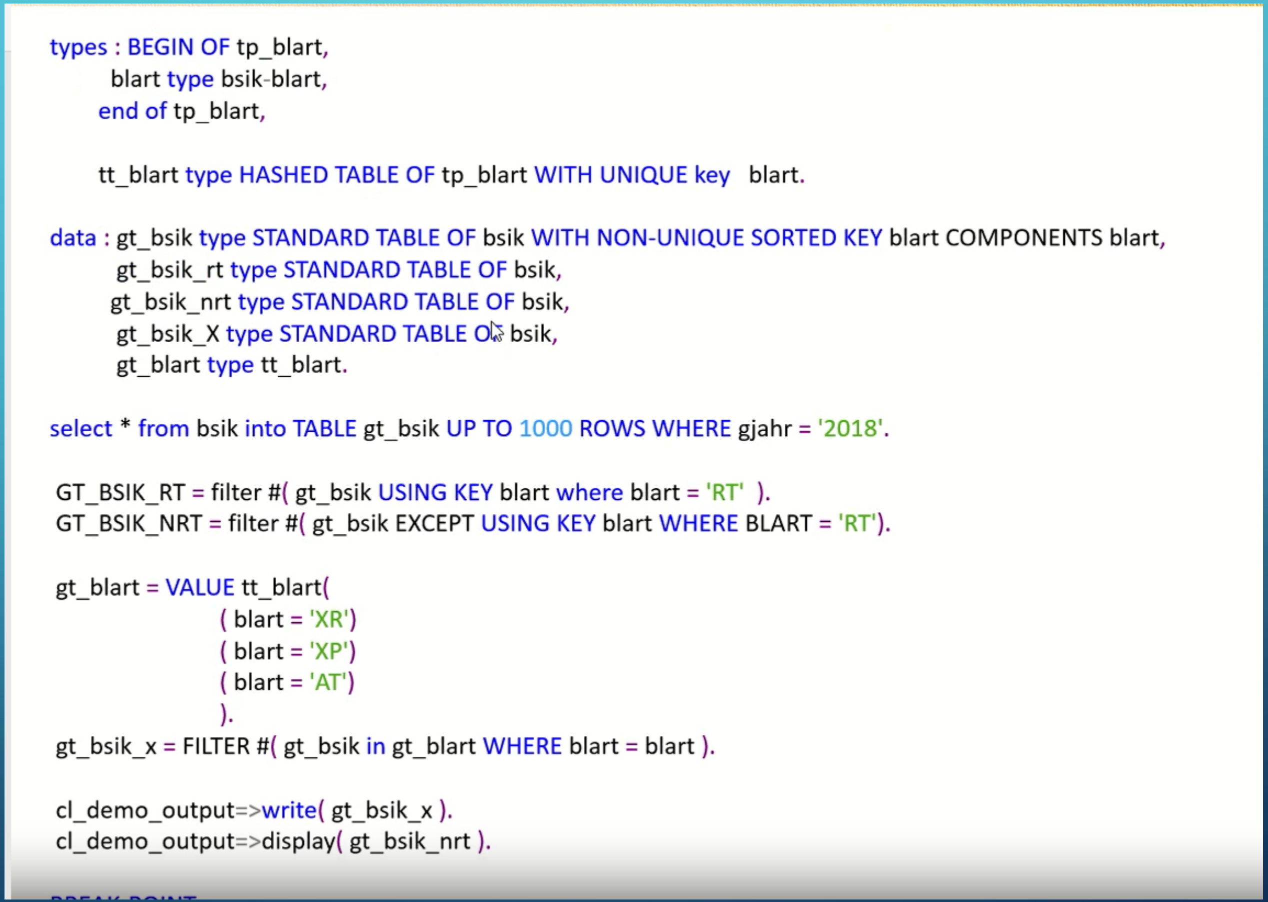
GT_ITAB4

EBELN	COMPANY	PLANT	BSART	VENDOR
4500000129	I710	I710		0017300002
4500000907	1710	1710		EWM17-SU01
4500001423	8800	8800		0008000002
4500001424	8800	8800		0008000000
4500000895	4900	4900		8000000047

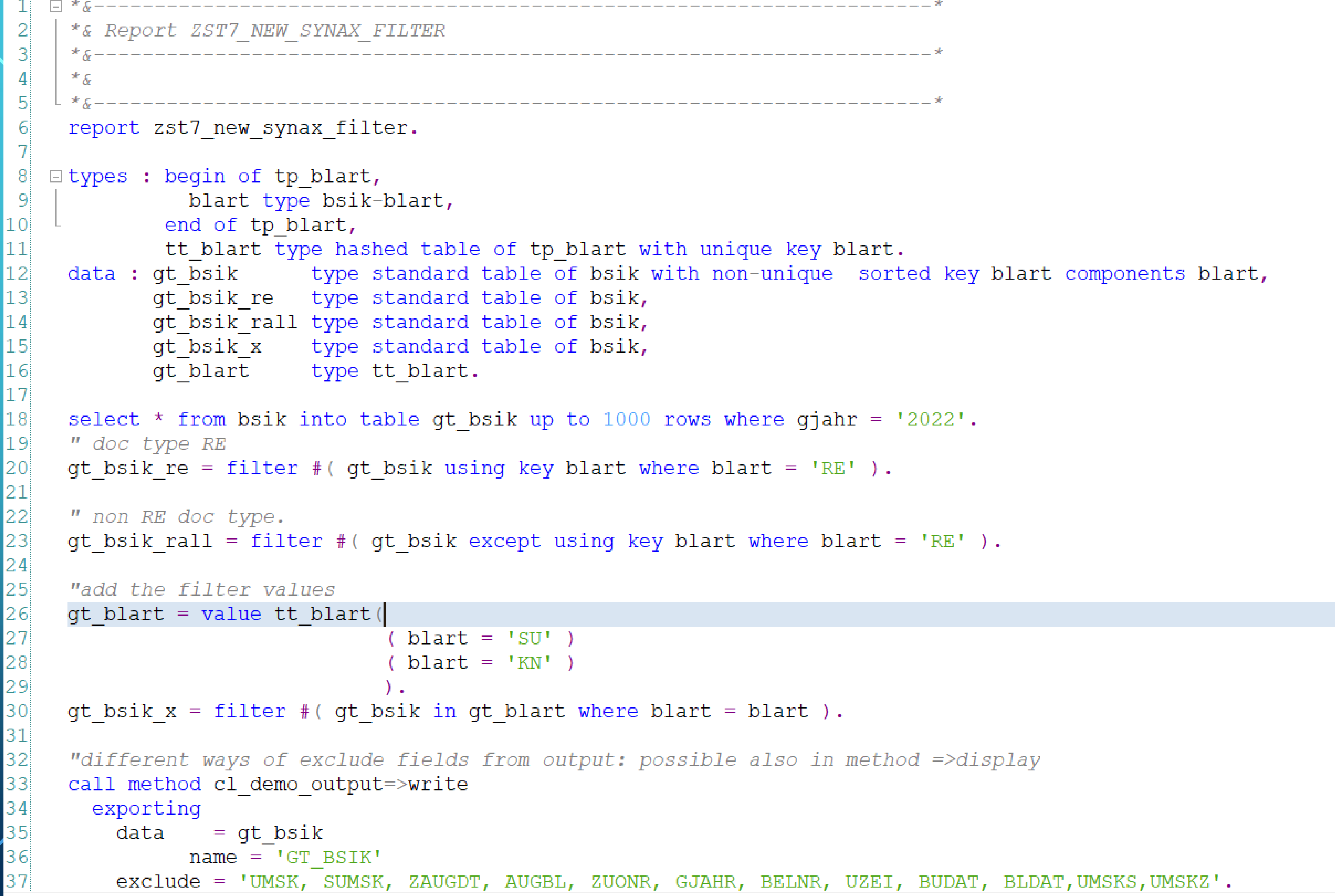
FILTER OPERATOR



Filter Operator : In new abap syntax we can use filter operator on ABAP internal tables to filter the data or to retrieve subset of data into a new internal table.

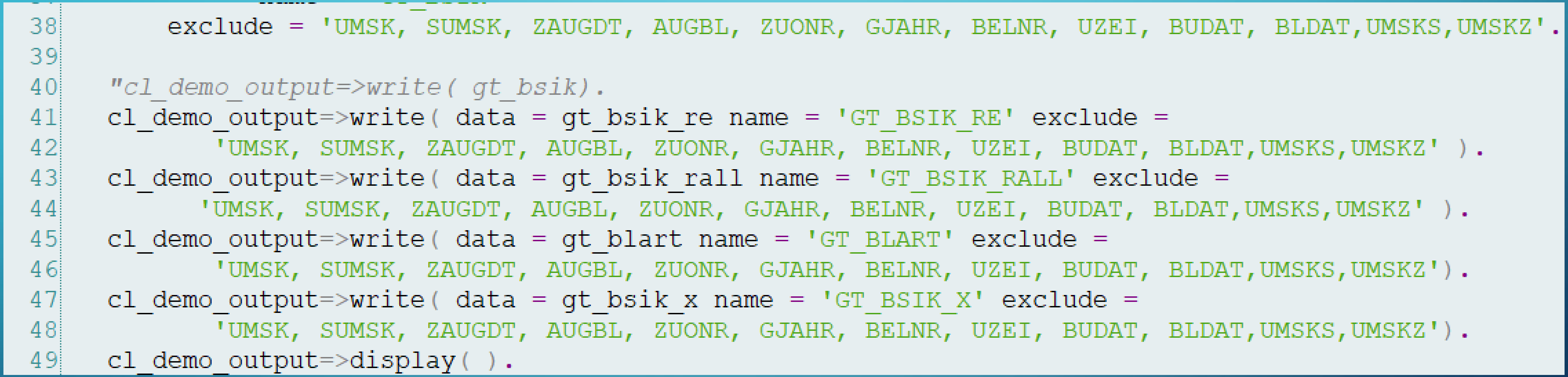


```
types : BEGIN OF tp_blart,
         blart type bsik-blart,
       end of tp_blart,

       tt_blart type HASHED TABLE OF tp_blart WITH UNIQUE key   blart.

data : gt_bsik type STANDARD TABLE OF bsik WITH NON-UNIQUE SORTED KEY blart COMPONENTS blart,
        gt_bsik_rt type STANDARD TABLE OF bsik,
       gt_bsik_nrt type STANDARD TABLE OF bsik,
        gt_bsik_X type STANDARD TABLE OF bsik,
        gt_blart type tt_blart.

select * from bsik into TABLE gt_bsik UP TO 1000 ROWS WHERE gjahr = '2018'.

 GT_BSIK_RT = filter #( gt_bsik USING KEY blart where blart = 'RT'  ).
 GT_BSIK_NRT = filter #( gt_bsik EXCEPT USING KEY blart WHERE BLART = 'RT').

 gt_blart = VALUE tt_blart(
                         ( blart = 'XR')
                         ( blart = 'XP')
                         ( blart = 'AT')
                         ).
 gt_bsik_x = FILTER #( gt_bsik in gt_blart WHERE blart = blart ).

 cl_demo_output=>write( gt_bsik_x ).
 cl_demo_output=>display( gt_bsik_nrt ).

BREAK-POINT
```



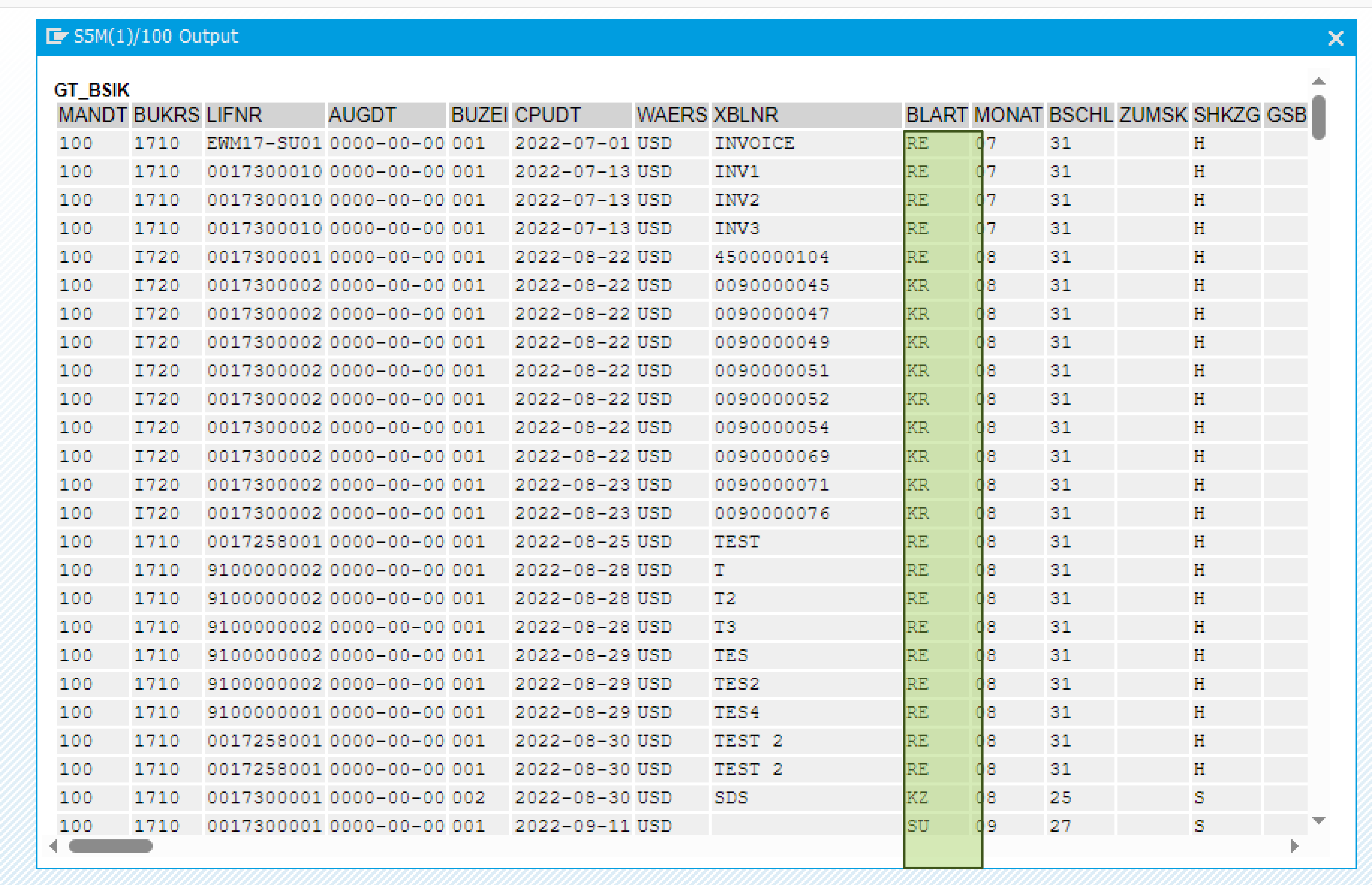
```
*&---------------------------------------------------------------------*
*& Report ZST7_NEW_SYNAX_FILTER
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
report zst7_new_synax_filter.

types : begin of tp_blart,
          blart type bsik-blart,
        end of tp_blart,
        tt_blart type hashed table of tp_blart with unique key blart.
data : gt_bsik      type standard table of bsik with non-unique  sorted key blart components blart,
       gt_bsik_re   type standard table of bsik,
       gt_bsik_rall type standard table of bsik,
       gt_bsik_x    type standard table of bsik,
       gt_blart     type tt_blart.

select * from bsik into table gt_bsik up to 1000 rows where gjahr = '2022'.
" doc type RE
gt_bsik_re = filter #( gt_bsik using key blart where blart = 'RE' ).

" non RE doc type.
gt_bsik_rall = filter #( gt_bsik except using key blart where blart = 'RE' ).

"add the filter values
gt_blart = value tt_blart(
                          ( blart = 'SU' )
                          ( blart = 'KN' )
                          ).
gt_bsik_x = filter #( gt_bsik in gt_blart where blart = blart ).

"different ways of exclude fields from output: possible also in method =>display
call method cl_demo_output=>write
  exporting
    data    = gt_bsik
          name = 'GT_BSIK'
    exclude = 'UMSK, SUMSK, ZAUGDT, AUGBL, ZUONR, GJAHR, BELNR, UZEI, BUDAT, BLDAT,UMSKS,UMSKZ'.
```

FILTER OPERATOR – SAMPLE PROGRAM



```
      exclude = 'UMSK, SUMSK, ZAUGDT, AUGBL, ZUONR, GJAHR, BELNR, UZEI, BUDAT, BLDAT,UMSKS,UMSKZ'.

  "cl_demo_output=>write( gt_bsik).
  cl_demo_output=>write( data = gt_bsik_re name = 'GT_BSIK_RE' exclude =
         'UMSK, SUMSK, ZAUGDT, AUGBL, ZUONR, GJAHR, BELNR, UZEI, BUDAT, BLDAT,UMSKS,UMSKZ' ).
  cl_demo_output=>write( data = gt_bsik_rall name = 'GT_BSIK_RALL' exclude =
        'UMSK, SUMSK, ZAUGDT, AUGBL, ZUONR, GJAHR, BELNR, UZEI, BUDAT, BLDAT,UMSKS,UMSKZ' ).
  cl_demo_output=>write( data = gt_blart name = 'GT_BLART' exclude =
         'UMSK, SUMSK, ZAUGDT, AUGBL, ZUONR, GJAHR, BELNR, UZEI, BUDAT, BLDAT,UMSKS,UMSKZ').
  cl_demo_output=>write( data = gt_bsik_x name = 'GT_BSIK_X' exclude =
         'UMSK, SUMSK, ZAUGDT, AUGBL, ZUONR, GJAHR, BELNR, UZEI, BUDAT, BLDAT,UMSKS,UMSKZ').
  cl_demo_output=>display( ).
```

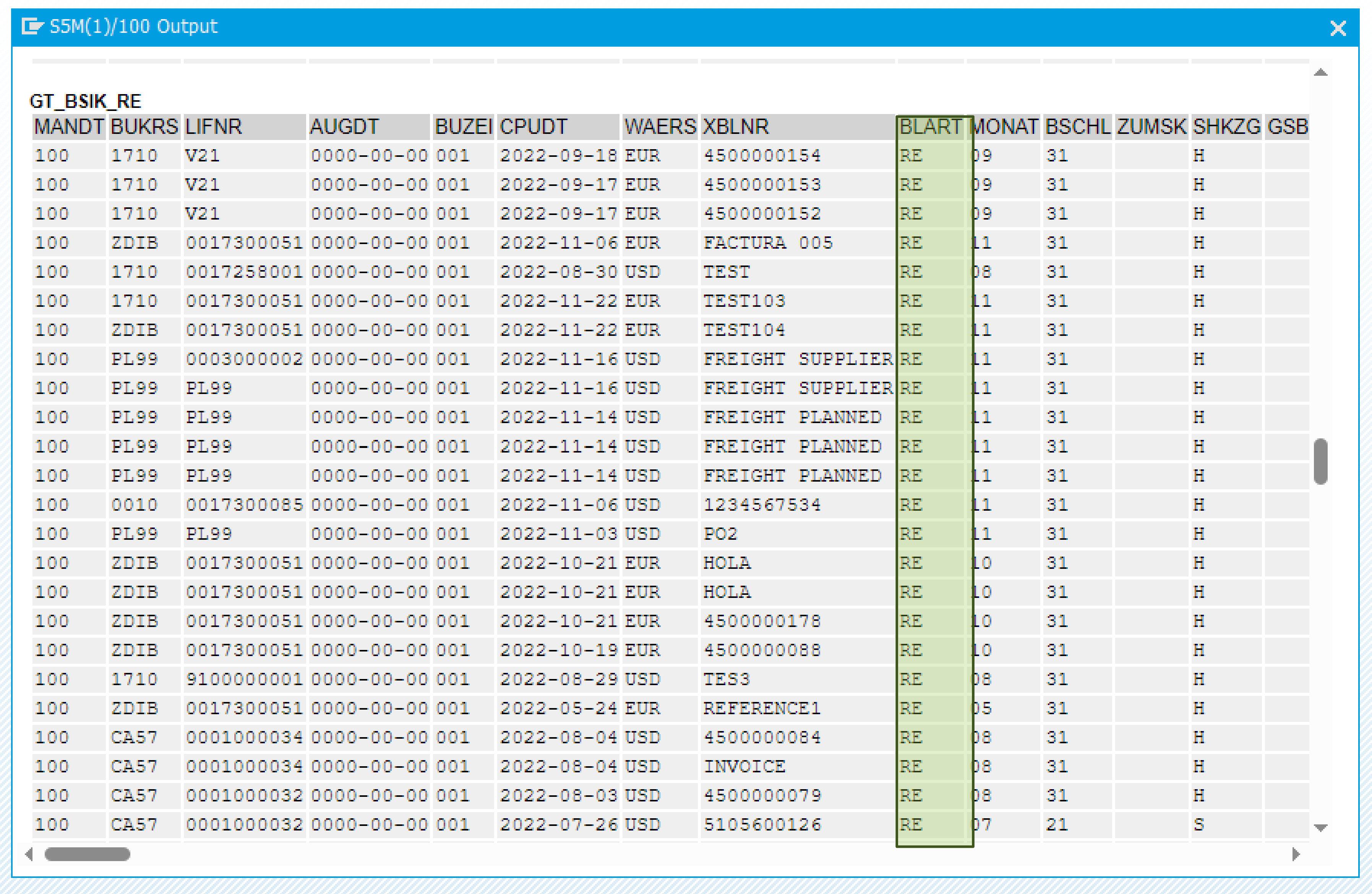
OUTPUT GT_BSIK:



GT_BSIK

MANDT	BUKRS	LIFNR	AUGDT	BUZEI	CPUDT	WAERS	XBLNR	BLART	MONAT	BSCHL	ZUMSK	SHKZG	GSB
100	1710	EWM17-SU01	0000-00-00	001	2022-07-01	USD	INVOICE	RE	07	31		H	
100	1710	0017300010	0000-00-00	001	2022-07-13	USD	INV1	RE	07	31		H	
100	1710	0017300010	0000-00-00	001	2022-07-13	USD	INV2	RE	07	31		H	
100	1710	0017300010	0000-00-00	001	2022-07-13	USD	INV3	RE	07	31		H	
100	I720	0017300001	0000-00-00	001	2022-08-22	USD	4500000104	RE	08	31		H	
100	I720	0017300002	0000-00-00	001	2022-08-22	USD	0090000045	KR	08	31		H	
100	I720	0017300002	0000-00-00	001	2022-08-22	USD	0090000047	KR	08	31		H	
100	I720	0017300002	0000-00-00	001	2022-08-22	USD	0090000049	KR	08	31		H	
100	I720	0017300002	0000-00-00	001	2022-08-22	USD	0090000051	KR	08	31		H	
100	I720	0017300002	0000-00-00	001	2022-08-22	USD	0090000052	KR	08	31		H	
100	I720	0017300002	0000-00-00	001	2022-08-22	USD	0090000054	KR	08	31		H	
100	I720	0017300002	0000-00-00	001	2022-08-22	USD	0090000069	KR	08	31		H	
100	I720	0017300002	0000-00-00	001	2022-08-23	USD	0090000071	KR	08	31		H	
100	I720	0017300002	0000-00-00	001	2022-08-23	USD	0090000076	KR	08	31		H	
100	1710	0017258001	0000-00-00	001	2022-08-25	USD	TEST	RE	08	31		H	
100	1710	9100000002	0000-00-00	001	2022-08-28	USD	T	RE	08	31		H	
100	1710	9100000002	0000-00-00	001	2022-08-28	USD	T2	RE	08	31		H	
100	1710	9100000002	0000-00-00	001	2022-08-28	USD	T3	RE	08	31		H	
100	1710	9100000002	0000-00-00	001	2022-08-29	USD	TES	RE	08	31		H	
100	1710	9100000002	0000-00-00	001	2022-08-29	USD	TES2	RE	08	31		H	
100	1710	9100000001	0000-00-00	001	2022-08-29	USD	TES4	RE	08	31		H	
100	1710	0017258001	0000-00-00	001	2022-08-30	USD	TEST 2	RE	08	31		H	
100	1710	0017258001	0000-00-00	001	2022-08-30	USD	TEST 2	RE	08	31		H	
100	1710	0017300001	0000-00-00	002	2022-08-30	USD	SDS	KZ	08	25		S	
100	1710	0017300001	0000-00-00	001	2022-09-11	USD		SU	09	27		S	

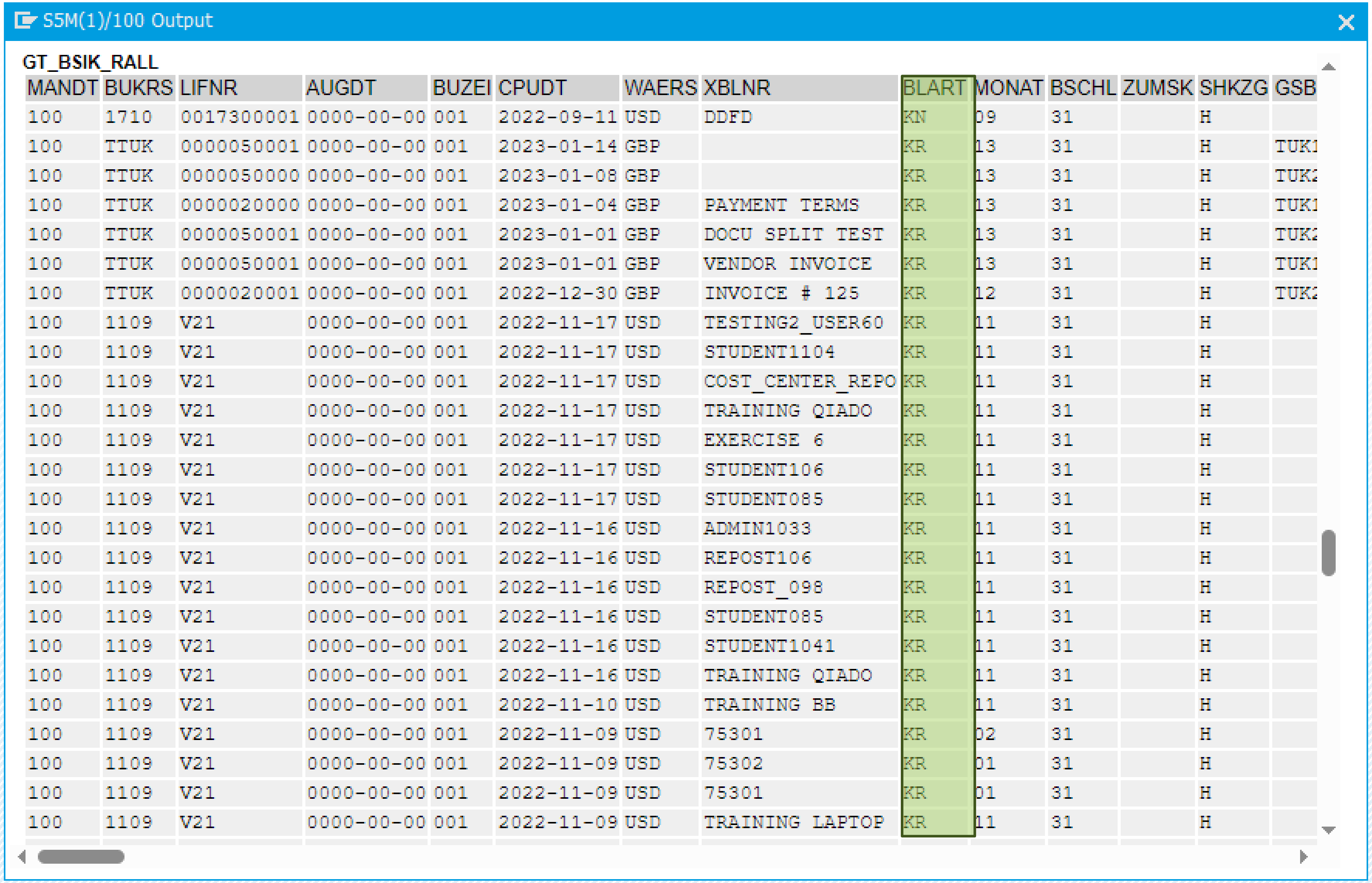
OUTPUT GT_BSIK_RE:



GT_BSIK_RE

MANDT	BUKRS	LIFNR	AUGDT	BUZEI	CPUDT	WAERS	XBLNR	BLART	MONAT	BSCHL	ZUMSK	SHKZG	GSB
100	1710	V21	0000-00-00	001	2022-09-18	EUR	4500000154	RE	09	31		H	
100	1710	V21	0000-00-00	001	2022-09-17	EUR	4500000153	RE	09	31		H	
100	1710	V21	0000-00-00	001	2022-09-17	EUR	4500000152	RE	09	31		H	
100	ZDIB	0017300051	0000-00-00	001	2022-11-06	EUR	FACTURA 005	RE	11	31		H	
100	1710	0017258001	0000-00-00	001	2022-08-30	USD	TEST	RE	08	31		H	
100	1710	0017300051	0000-00-00	001	2022-11-22	EUR	TEST103	RE	11	31		H	
100	ZDIB	0017300051	0000-00-00	001	2022-11-22	EUR	TEST104	RE	11	31		H	
100	PL99	0003000002	0000-00-00	001	2022-11-16	USD	FREIGHT SUPPLIER	RE	11	31		H	
100	PL99	PL99	0000-00-00	001	2022-11-16	USD	FREIGHT SUPPLIER	RE	11	31		H	
100	PL99	PL99	0000-00-00	001	2022-11-14	USD	FREIGHT PLANNED	RE	11	31		H	
100	PL99	PL99	0000-00-00	001	2022-11-14	USD	FREIGHT PLANNED	RE	11	31		H	
100	PL99	PL99	0000-00-00	001	2022-11-14	USD	FREIGHT PLANNED	RE	11	31		H	
100	0010	0017300085	0000-00-00	001	2022-11-06	USD	1234567534	RE	11	31		H	
100	PL99	PL99	0000-00-00	001	2022-11-03	USD	PO2	RE	11	31		H	
100	ZDIB	0017300051	0000-00-00	001	2022-10-21	EUR	HOLA	RE	10	31		H	
100	ZDIB	0017300051	0000-00-00	001	2022-10-21	EUR	HOLA	RE	10	31		H	
100	ZDIB	0017300051	0000-00-00	001	2022-10-21	EUR	4500000178	RE	10	31		H	
100	ZDIB	0017300051	0000-00-00	001	2022-10-19	EUR	4500000088	RE	10	31		H	
100	1710	9100000001	0000-00-00	001	2022-08-29	USD	TES3	RE	08	31		H	
100	ZDIB	0017300051	0000-00-00	001	2022-05-24	EUR	REFERENCE1	RE	05	31		H	
100	CA57	0001000034	0000-00-00	001	2022-08-04	USD	4500000084	RE	08	31		H	
100	CA57	0001000034	0000-00-00	001	2022-08-04	USD	INVOICE	RE	08	31		H	
100	CA57	0001000032	0000-00-00	001	2022-08-03	USD	4500000079	RE	08	31		H	
100	CA57	0001000032	0000-00-00	001	2022-07-26	USD	5105600126	RE	07	21		S	

OUTPUT GT_BSIK_RALL:

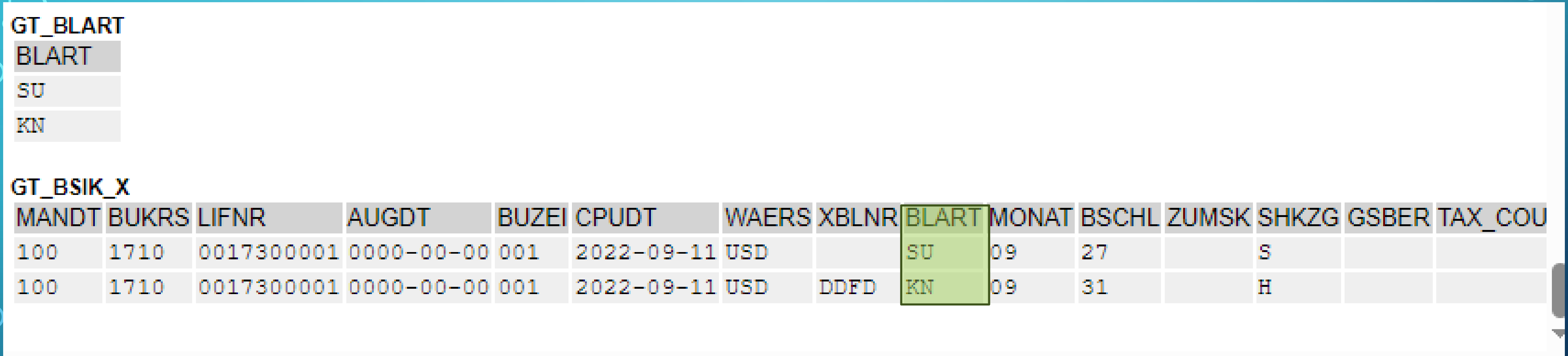


S5M(1)/100 Output

GT_BSIK_RALL

MANDT	BUKRS	LIFNR	AUGDT	BUZEI	CPUDT	WAERS	XBLNR	BLART	MONAT	BSCHL	ZUMSK	SHKZG	GSB
100	1710	0017300001	0000-00-00	001	2022-09-11	USD	DDFD	KN	09	31		H	
100	TTUK	0000050001	0000-00-00	001	2023-01-14	GBP		KR	13	31		H	TUK1
100	TTUK	0000050000	0000-00-00	001	2023-01-08	GBP		KR	13	31		H	TUK2
100	TTUK	0000020000	0000-00-00	001	2023-01-04	GBP	PAYMENT TERMS	KR	13	31		H	TUK1
100	TTUK	0000050001	0000-00-00	001	2023-01-01	GBP	DOCU SPLIT TEST	KR	13	31		H	TUK2
100	TTUK	0000050001	0000-00-00	001	2023-01-01	GBP	VENDOR INVOICE	KR	13	31		H	TUK1
100	TTUK	0000020001	0000-00-00	001	2022-12-30	GBP	INVOICE # 125	KR	12	31		H	TUK2
100	1109	V21	0000-00-00	001	2022-11-17	USD	TESTING2_USER60	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-17	USD	STUDENT1104	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-17	USD	COST_CENTER_REPO	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-17	USD	TRAINING QIADO	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-17	USD	EXERCISE 6	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-17	USD	STUDENT106	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-17	USD	STUDENT085	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-16	USD	ADMIN1033	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-16	USD	REPOST106	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-16	USD	REPOST_098	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-16	USD	STUDENT085	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-16	USD	STUDENT1041	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-16	USD	TRAINING QIADO	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-10	USD	TRAINING BB	KR	11	31		H	
100	1109	V21	0000-00-00	001	2022-11-09	USD	75301	KR	02	31		H	
100	1109	V21	0000-00-00	001	2022-11-09	USD	75302	KR	01	31		H	
100	1109	V21	0000-00-00	001	2022-11-09	USD	75301	KR	01	31		H	
100	1109	V21	0000-00-00	001	2022-11-09	USD	TRAINING LAPTOP	KR	11	31		H	

OUTPUT GT_BLART , GT_BSIK_X:



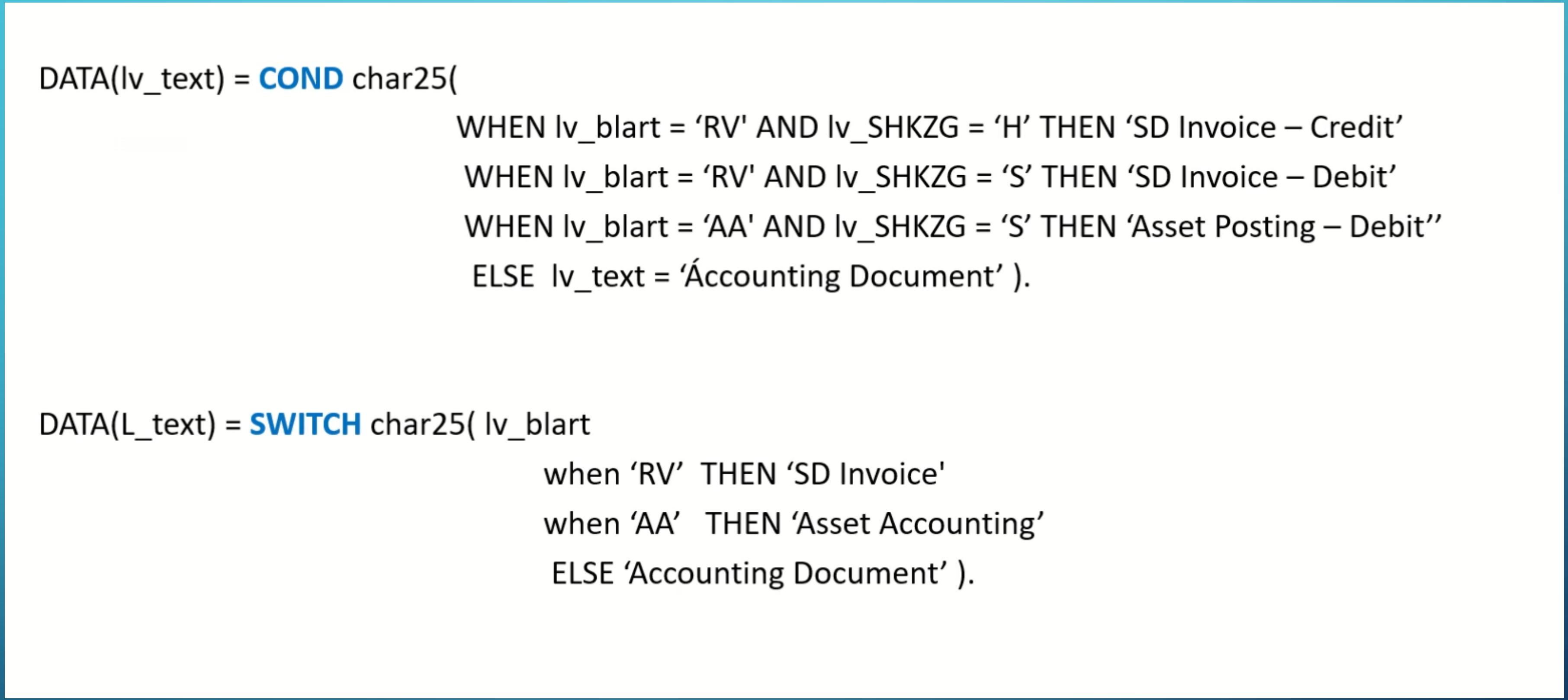
GT_BLART

BLART
SU
KN

GT_BSIK_X

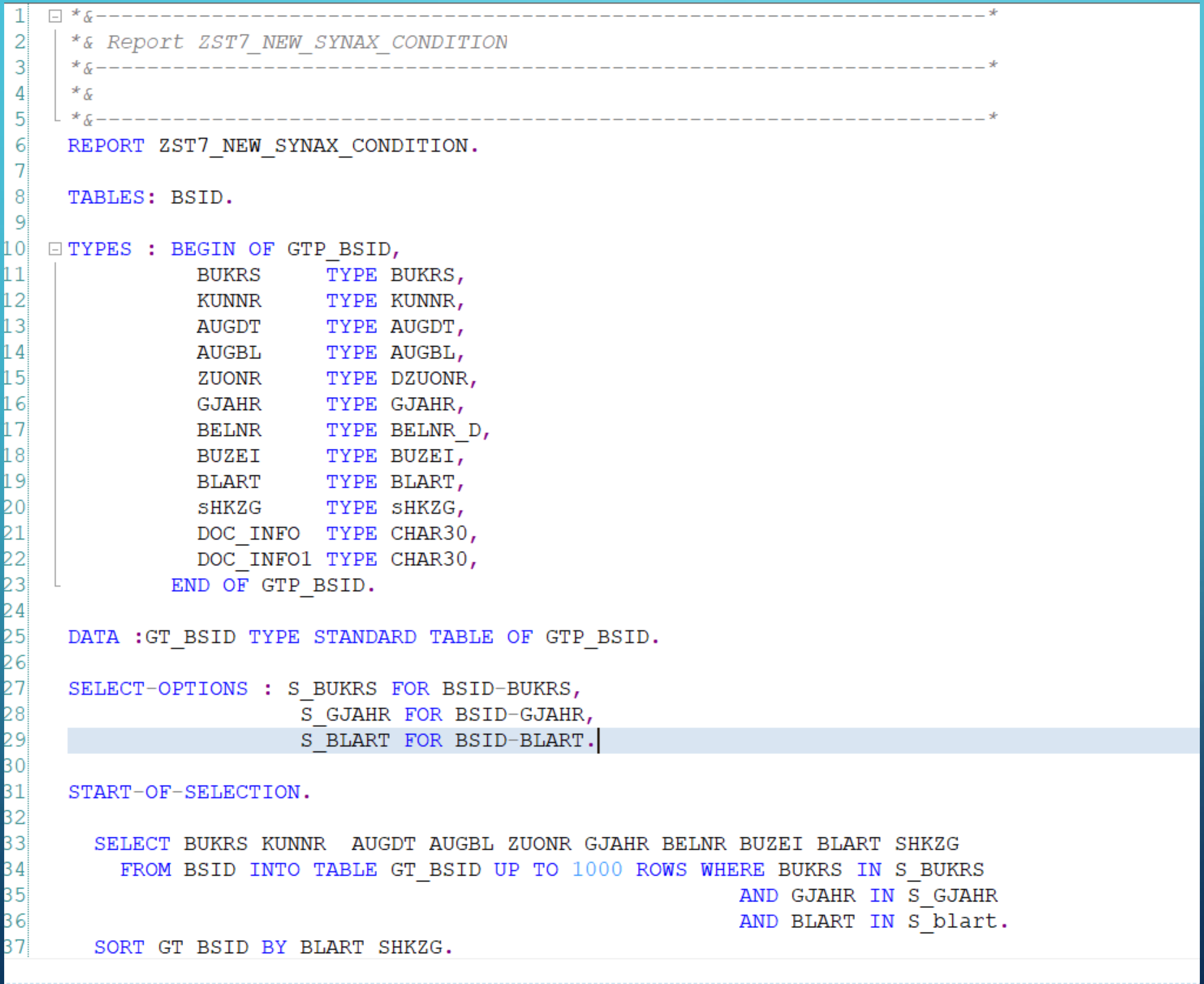
MANDT	BUKRS	LIFNR	AUGDT	BUZEI	CPUDT	WAERS	XBLNR	BLART	MONAT	BSCHL	ZUMSK	SHKZG	GSBER	TAX_COU
100	1710	0017300001	0000-00-00	001	2022-09-11	USD		SU	09	27		S		
100	1710	0017300001	0000-00-00	001	2022-09-11	USD	DDFD	KN	09	31		H		

NEW SYNTAX CONDITIONAL & SWITCH OPERATORS:



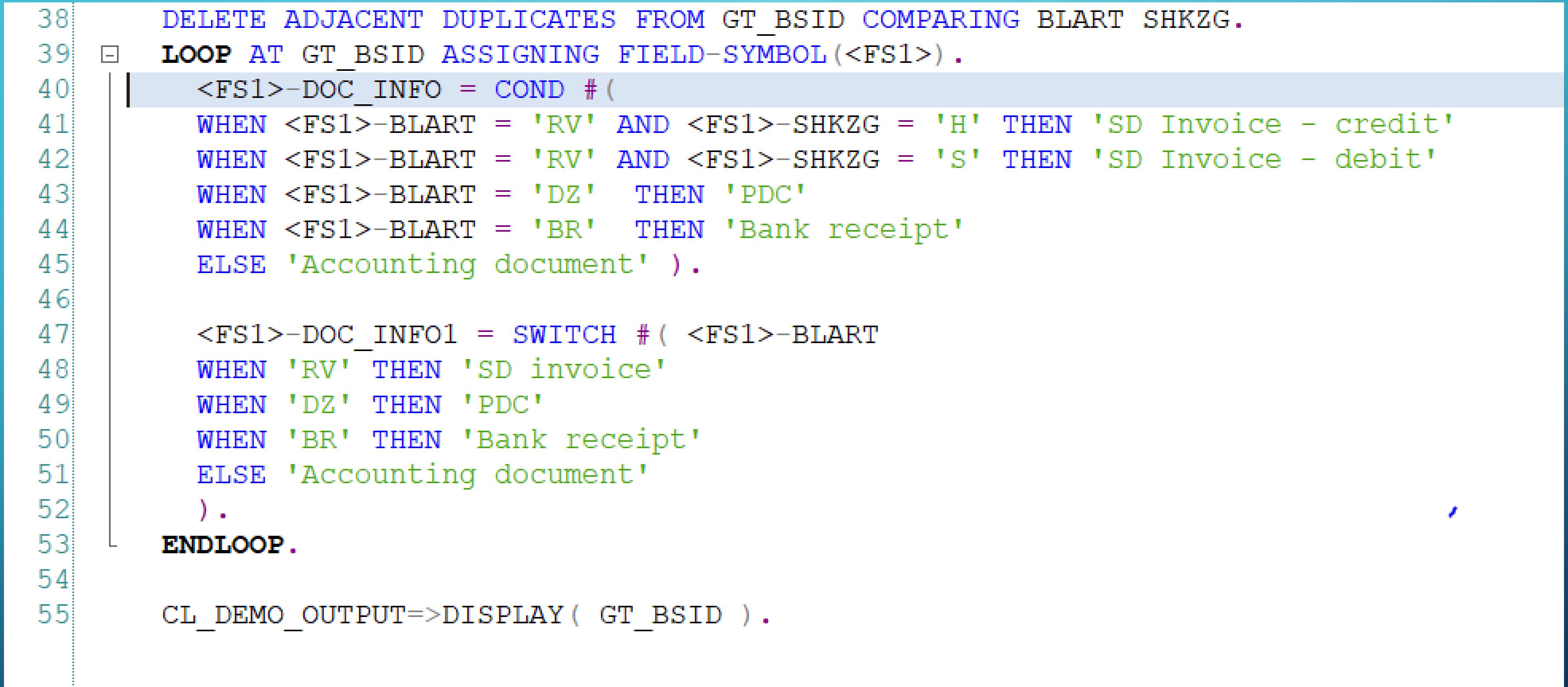
```
DATA(lv_text) = COND char25(
                                   WHEN lv_blart = ‘RV' AND lv_SHKZG = ‘H’ THEN ‘SD Invoice – Credit’
                                    WHEN lv_blart = ‘RV' AND lv_SHKZG = ‘S’ THEN ‘SD Invoice – Debit’
                                    WHEN lv_blart = ‘AA' AND lv_SHKZG = ‘S’ THEN ‘Asset Posting – Debit’’
                                     ELSE  lv_text = ‘Áccounting Document’ ).


DATA(L_text) = SWITCH char25( lv_blart
                                          when ‘RV’  THEN ‘SD Invoice'
                                          when ‘AA’   THEN ‘Asset Accounting’
                                           ELSE ‘Accounting Document’ ).
```

NEW SYNTAX CONDITIONAL & SWITCH OPERATORS-PROGRAM :



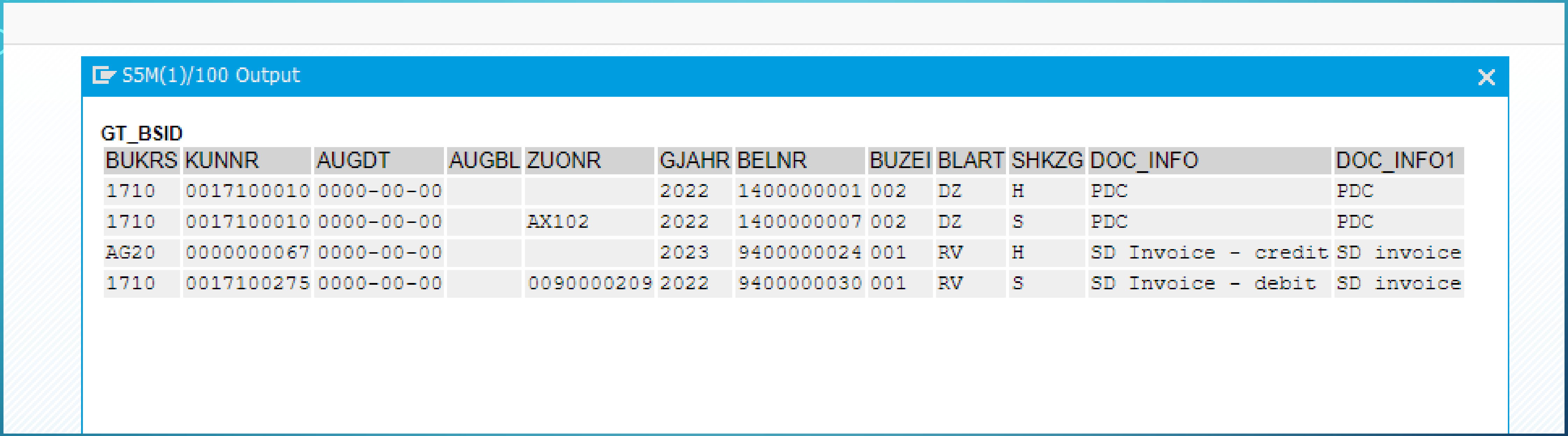
```
*&---------------------------------------------------------------------*
*& Report ZST7_NEW_SYNAX_CONDITION
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
REPORT ZST7_NEW_SYNAX_CONDITION.

TABLES: BSID.

TYPES : BEGIN OF GTP_BSID,
          BUKRS     TYPE BUKRS,
          KUNNR     TYPE KUNNR,
          AUGDT     TYPE AUGDT,
          AUGBL     TYPE AUGBL,
          ZUONR     TYPE DZUONR,
          GJAHR     TYPE GJAHR,
          BELNR     TYPE BELNR_D,
          BUZEI     TYPE BUZEI,
          BLART     TYPE BLART,
          sHKZG     TYPE sHKZG,
          DOC_INFO  TYPE CHAR30,
          DOC_INFO1 TYPE CHAR30,
        END OF GTP_BSID.

DATA :GT_BSID TYPE STANDARD TABLE OF GTP_BSID.

SELECT-OPTIONS : S_BUKRS FOR BSID-BUKRS,
                  S_GJAHR FOR BSID-GJAHR,
                  S_BLART FOR BSID-BLART.

START-OF-SELECTION.

  SELECT BUKRS KUNNR  AUGDT AUGBL ZUONR GJAHR BELNR BUZEI BLART SHKZG
    FROM BSID INTO TABLE GT_BSID UP TO 1000 ROWS WHERE BUKRS IN S_BUKRS
                                                    AND GJAHR IN S_GJAHR
                                                    AND BLART IN S_blart.

  SORT GT_BSID BY BLART SHKZG.
```

NEW SYNTAX CONDITIONAL & SWITCH OPERATORS- PROGRAM :



```
DELETE ADJACENT DUPLICATES FROM GT_BSID COMPARING BLART SHKZG.
LOOP AT GT_BSID ASSIGNING FIELD-SYMBOL(<FS1>).
  <FS1>-DOC_INFO = COND #(
  WHEN <FS1>-BLART = 'RV' AND <FS1>-SHKZG = 'H' THEN 'SD Invoice - credit'
  WHEN <FS1>-BLART = 'RV' AND <FS1>-SHKZG = 'S' THEN 'SD Invoice - debit'
  WHEN <FS1>-BLART = 'DZ'  THEN 'PDC'
  WHEN <FS1>-BLART = 'BR'  THEN 'Bank receipt'
  ELSE 'Accounting document' ).

  <FS1>-DOC_INFO1 = SWITCH #( <FS1>-BLART
  WHEN 'RV' THEN 'SD invoice'
  WHEN 'DZ' THEN 'PDC'
  WHEN 'BR' THEN 'Bank receipt'
  ELSE 'Accounting document'
  ).
ENDLOOP.

CL_DEMO_OUTPUT=>DISPLAY( GT_BSID ).
```

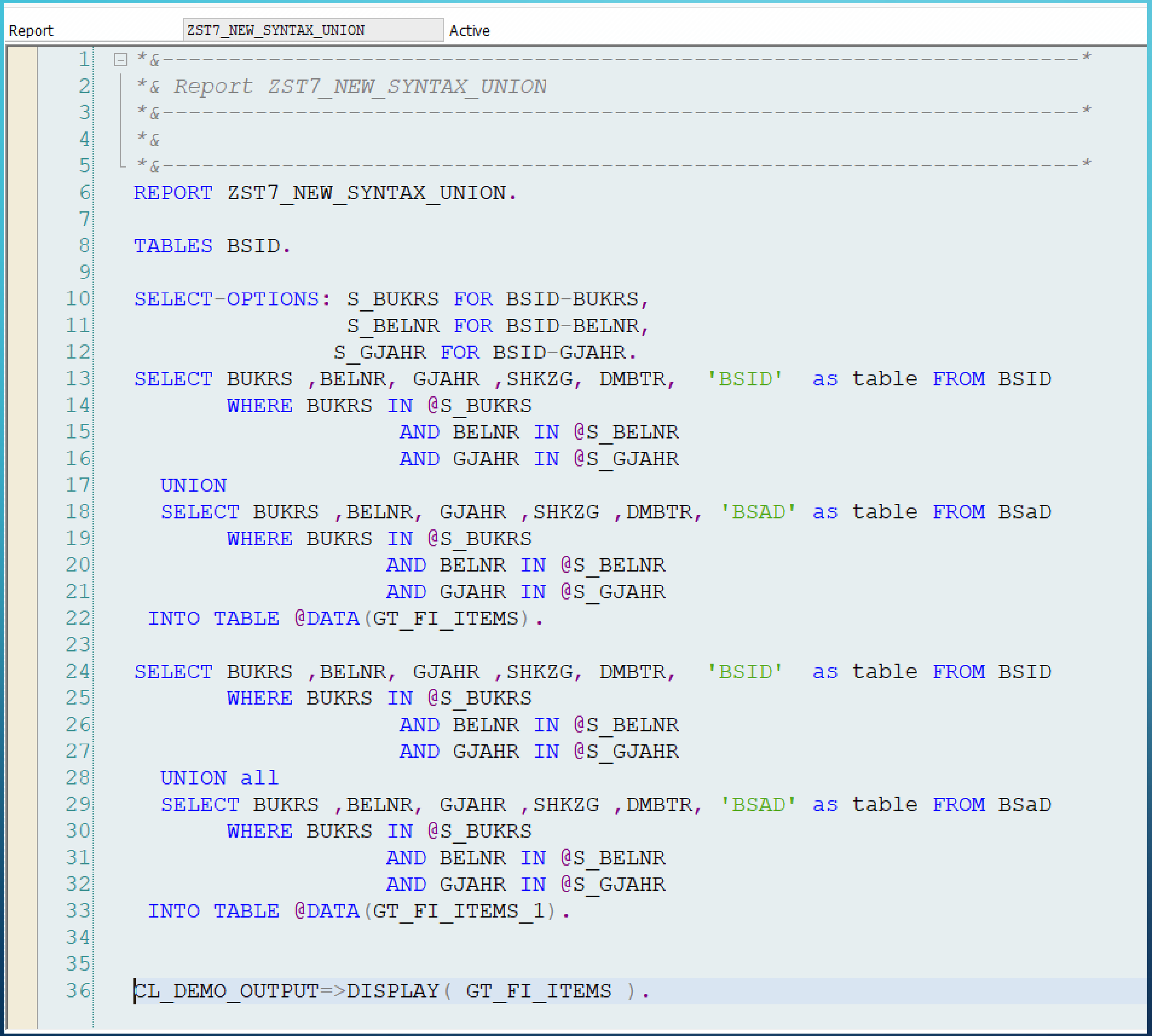
NEW SYNTAX CONDITIONAL & SWITCH OPERATORS- PROGRAM RESULT:



GT_BSID

BUKRS	KUNNR	AUGDT	AUGBL	ZUONR	GJAHR	BELNR	BUZEI	BLART	SHKZG	DOC_INFO	DOC_INFO1
1710	0017100010	0000-00-00			2022	1400000001	002	DZ	H	PDC	PDC
1710	0017100010	0000-00-00		AX102	2022	1400000007	002	DZ	S	PDC	PDC
AG20	0000000067	0000-00-00			2023	9400000024	001	RV	H	SD Invoice - credit	SD invoice
1710	0017100275	0000-00-00		0090000209	2022	9400000030	001	RV	S	SD Invoice - debit	SD invoice

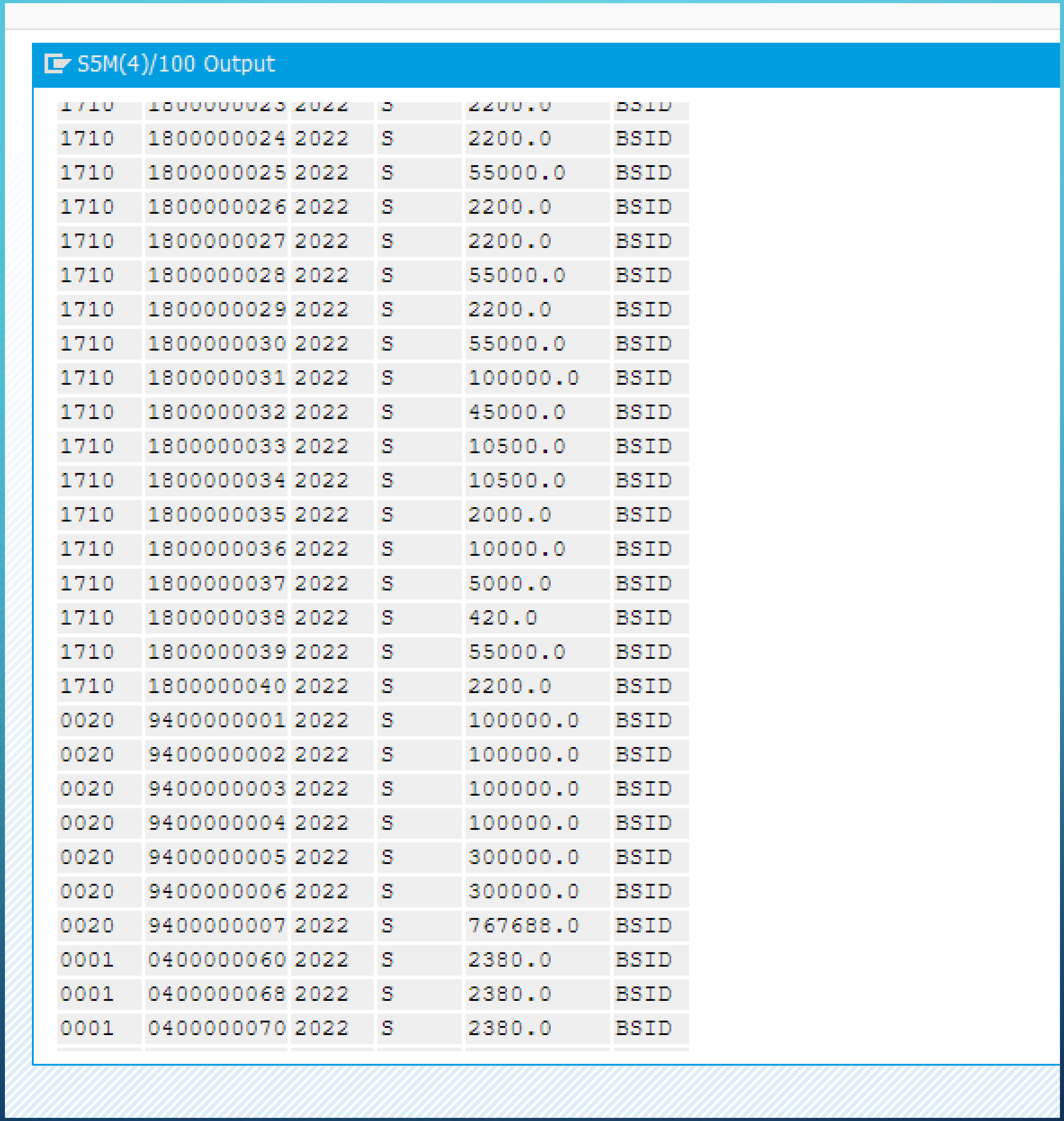
NEW SYNTAX UNION OPERATOR



Report ZST7_NEW_SYNTAX_UNION Active

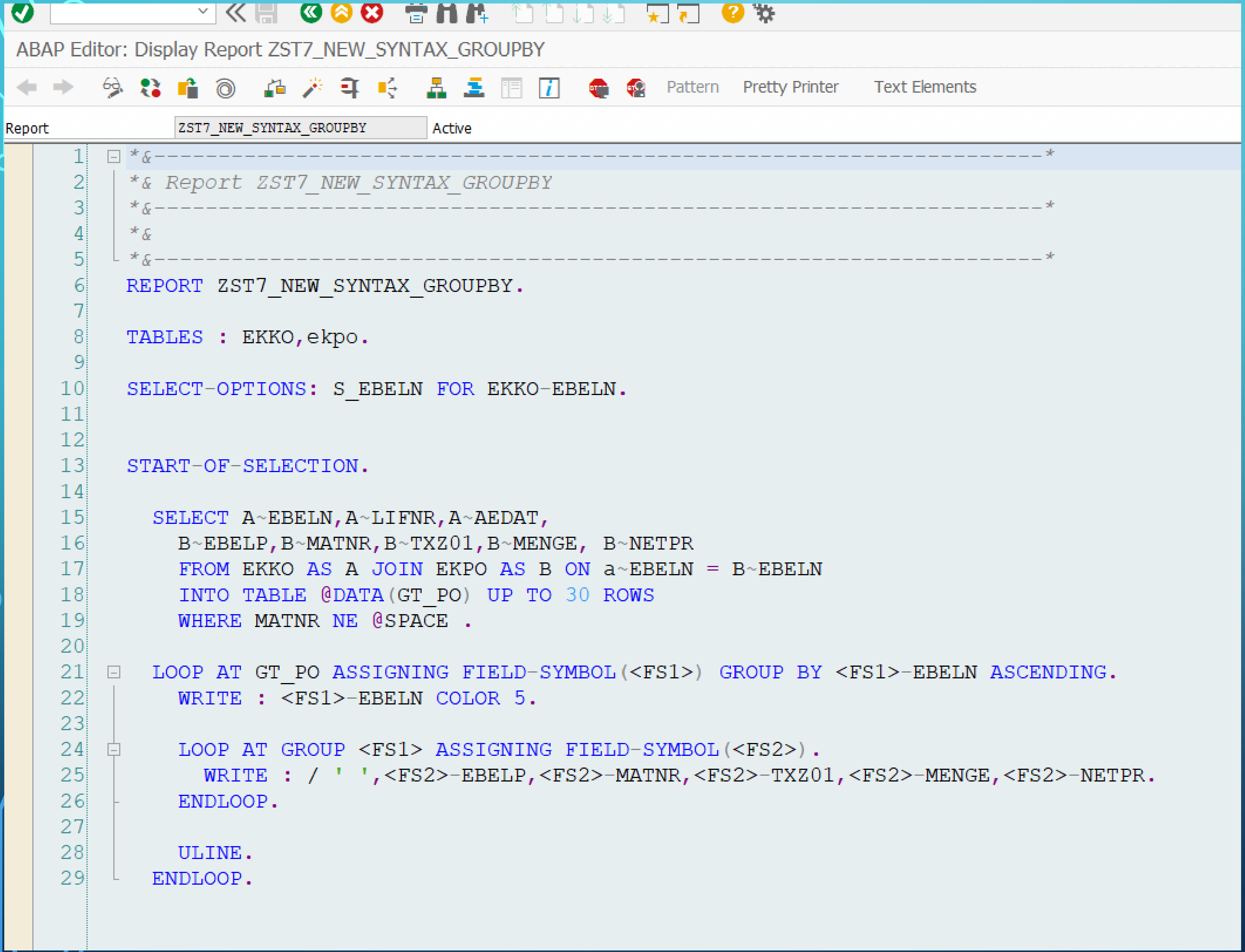
```
*&---------------------------------------------------------------------*
*& Report ZST7_NEW_SYNTAX_UNION
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
REPORT ZST7_NEW_SYNTAX_UNION.

TABLES BSID.

SELECT-OPTIONS: S_BUKRS FOR BSID-BUKRS,
                S_BELNR FOR BSID-BELNR,
               S_GJAHR FOR BSID-GJAHR.
SELECT BUKRS ,BELNR, GJAHR ,SHKZG, DMBTR,  'BSID'  as table FROM BSID
       WHERE BUKRS IN @S_BUKRS
                    AND BELNR IN @S_BELNR
                    AND GJAHR IN @S_GJAHR
  UNION
  SELECT BUKRS ,BELNR, GJAHR ,SHKZG ,DMBTR, 'BSAD' as table FROM BSaD
       WHERE BUKRS IN @S_BUKRS
                   AND BELNR IN @S_BELNR
                   AND GJAHR IN @S_GJAHR
 INTO TABLE @DATA(GT_FI_ITEMS).

SELECT BUKRS ,BELNR, GJAHR ,SHKZG, DMBTR,  'BSID'  as table FROM BSID
       WHERE BUKRS IN @S_BUKRS
                    AND BELNR IN @S_BELNR
                    AND GJAHR IN @S_GJAHR
  UNION all
  SELECT BUKRS ,BELNR, GJAHR ,SHKZG ,DMBTR, 'BSAD' as table FROM BSaD
       WHERE BUKRS IN @S_BUKRS
                   AND BELNR IN @S_BELNR
                   AND GJAHR IN @S_GJAHR
 INTO TABLE @DATA(GT_FI_ITEMS_1).


CL_DEMO_OUTPUT=>DISPLAY( GT_FI_ITEMS ).
```

NEW SYNTAX UNION OPERATOR -RESULT



```
1710  1800000023 2022  S    2200.0     BSID
1710  1800000024 2022  S    2200.0     BSID
1710  1800000025 2022  S    55000.0    BSID
1710  1800000026 2022  S    2200.0     BSID
1710  1800000027 2022  S    2200.0     BSID
1710  1800000028 2022  S    55000.0    BSID
1710  1800000029 2022  S    2200.0     BSID
1710  1800000030 2022  S    55000.0    BSID
1710  1800000031 2022  S    100000.0   BSID
1710  1800000032 2022  S    45000.0    BSID
1710  1800000033 2022  S    10500.0    BSID
1710  1800000034 2022  S    10500.0    BSID
1710  1800000035 2022  S    2000.0     BSID
1710  1800000036 2022  S    10000.0    BSID
1710  1800000037 2022  S    5000.0     BSID
1710  1800000038 2022  S    420.0      BSID
1710  1800000039 2022  S    55000.0    BSID
1710  1800000040 2022  S    2200.0     BSID
0020  9400000001 2022  S    100000.0   BSID
0020  9400000002 2022  S    100000.0   BSID
0020  9400000003 2022  S    100000.0   BSID
0020  9400000004 2022  S    100000.0   BSID
0020  9400000005 2022  S    300000.0   BSID
0020  9400000006 2022  S    300000.0   BSID
0020  9400000007 2022  S    767688.0   BSID
0001  0400000060 2022  S    2380.0     BSID
0001  0400000068 2022  S    2380.0     BSID
0001  0400000070 2022  S    2380.0     BSID
```

NEW ABAP SYNTAX GROUP BY- SAMPLE PROGRAM

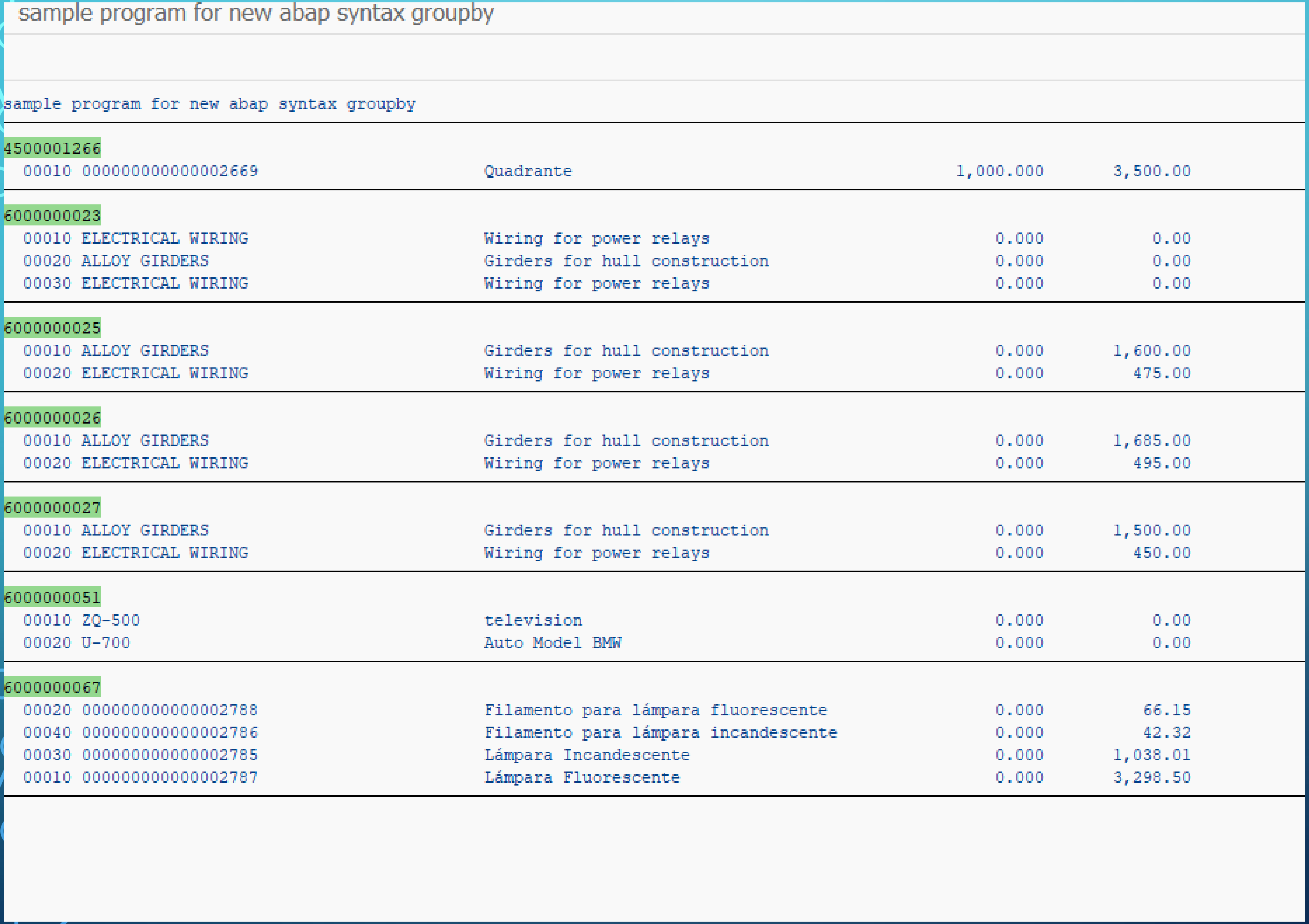


ABAP Editor: Display Report ZST7_NEW_SYNTAX_GROUPBY

Report ZST7_NEW_SYNTAX_GROUPBY Active

```
*&---------------------------------------------------------------------*
*& Report ZST7_NEW_SYNTAX_GROUPBY
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
REPORT ZST7_NEW_SYNTAX_GROUPBY.

TABLES : EKKO,ekpo.

SELECT-OPTIONS: S_EBELN FOR EKKO-EBELN.


START-OF-SELECTION.

  SELECT A~EBELN,A~LIFNR,A~AEDAT,
    B~EBELP,B~MATNR,B~TXZ01,B~MENGE, B~NETPR
    FROM EKKO AS A JOIN EKPO AS B ON a~EBELN = B~EBELN
    INTO TABLE @DATA(GT_PO) UP TO 30 ROWS
    WHERE MATNR NE @SPACE .

  LOOP AT GT_PO ASSIGNING FIELD-SYMBOL(<FS1>) GROUP BY <FS1>-EBELN ASCENDING.
    WRITE : <FS1>-EBELN COLOR 5.

    LOOP AT GROUP <FS1> ASSIGNING FIELD-SYMBOL(<FS2>).
      WRITE : / ' ',<FS2>-EBELP,<FS2>-MATNR,<FS2>-TXZ01,<FS2>-MENGE,<FS2>-NETPR.
    ENDLOOP.

    ULINE.
  ENDLOOP.
```

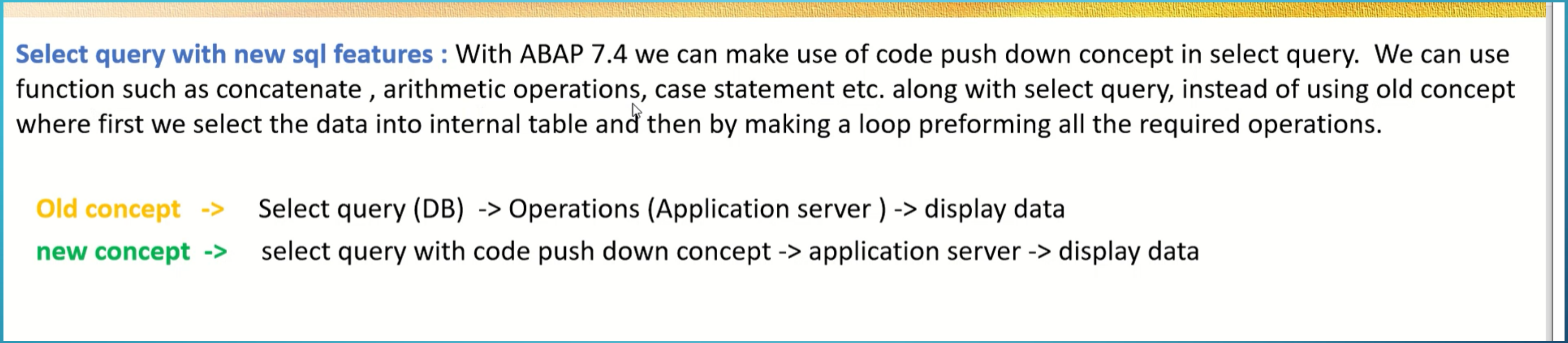
NEW ABAP SYNTAX GROUP BY- RESULT



sample program for new abap syntax groupby

```
sample program for new abap syntax groupby

4500001266
  00010 000000000000002669                        Quadrante                                    1,000.000       3,500.00

6000000023
  00010 ELECTRICAL WIRING                         Wiring for power relays                          0.000           0.00
  00020 ALLOY GIRDERS                             Girders for hull construction                    0.000           0.00
  00030 ELECTRICAL WIRING                         Wiring for power relays                          0.000           0.00

6000000025
  00010 ALLOY GIRDERS                             Girders for hull construction                    0.000       1,600.00
  00020 ELECTRICAL WIRING                         Wiring for power relays                          0.000         475.00

6000000026
  00010 ALLOY GIRDERS                             Girders for hull construction                    0.000       1,685.00
  00020 ELECTRICAL WIRING                         Wiring for power relays                          0.000         495.00

6000000027
  00010 ALLOY GIRDERS                             Girders for hull construction                    0.000       1,500.00
  00020 ELECTRICAL WIRING                         Wiring for power relays                          0.000         450.00

6000000051
  00010 ZQ-500                                    television                                       0.000           0.00
  00020 U-700                                     Auto Model BMW                                   0.000           0.00

6000000067
  00020 000000000000002788                        Filamento para lámpara fluorescente              0.000          66.15
  00040 000000000000002786                        Filamento para lámpara incandescente             0.000          42.32
  00030 000000000000002785                        Lámpara Incandescente                            0.000       1,038.01
  00010 000000000000002787                        Lámpara Fluorescente                             0.000       3,298.50
```

NEW ABAP SYNTAX – SQL FEATURES

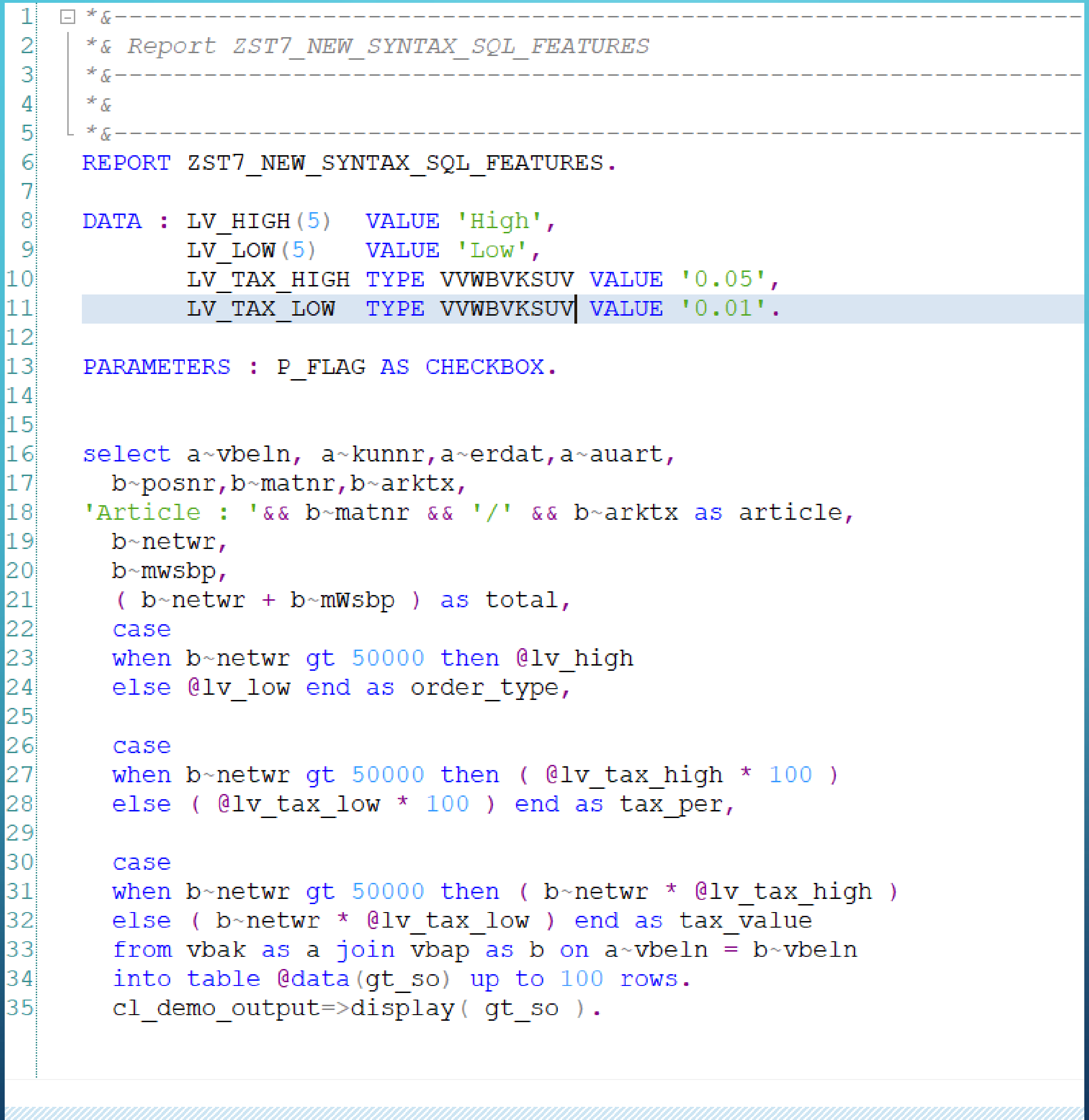


Select query with new sql features : With ABAP 7.4 we can make use of code push down concept in select query. We can use function such as concatenate , arithmetic operations, case statement etc. along with select query, instead of using old concept where first we select the data into internal table and then by making a loop preforming all the required operations.

Old concept -> Select query (DB) -> Operations (Application server) -> display data

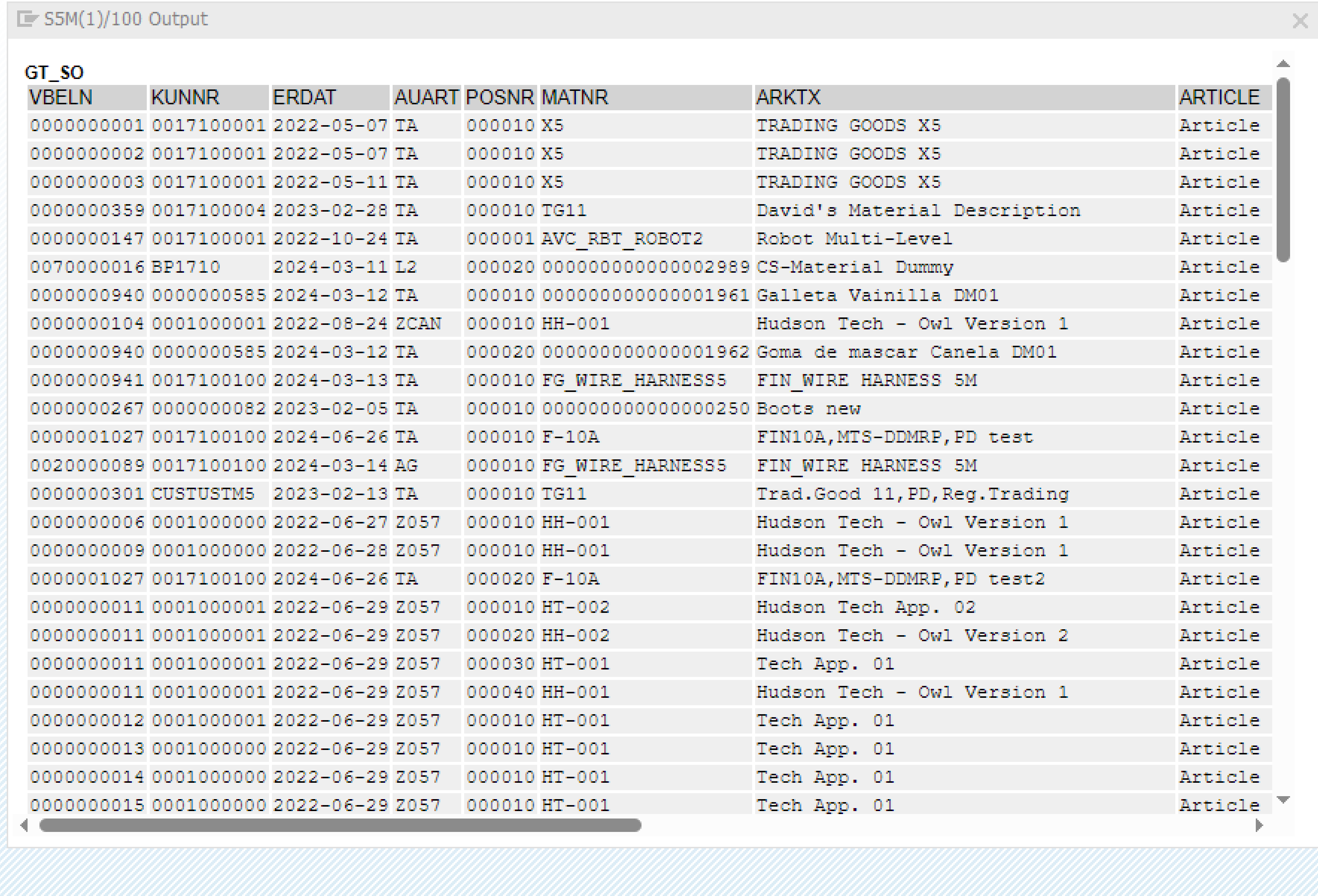
new concept -> select query with code push down concept -> application server -> display data

NEW ABAP SYNTAX – SQL FEATURES- SAMPLE PROGRAM



```
*&---------------------------------------------------------------------
*& Report ZST7_NEW_SYNTAX_SQL_FEATURES
*&---------------------------------------------------------------------
*&
*&---------------------------------------------------------------------
REPORT ZST7_NEW_SYNTAX_SQL_FEATURES.

DATA : LV_HIGH(5)  VALUE 'High',
       LV_LOW(5)   VALUE 'Low',
       LV_TAX_HIGH TYPE VVWBVKSUV VALUE '0.05',
       LV_TAX_LOW  TYPE VVWBVKSUV VALUE '0.01'.

PARAMETERS : P_FLAG AS CHECKBOX.


select a~vbeln, a~kunnr,a~erdat,a~auart,
  b~posnr,b~matnr,b~arktx,
'Article : '&& b~matnr && '/' && b~arktx as article,
  b~netwr,
  b~mwsbp,
  ( b~netwr + b~mWsbp ) as total,
  case
  when b~netwr gt 50000 then @lv_high
  else @lv_low end as order_type,

  case
  when b~netwr gt 50000 then ( @lv_tax_high * 100 )
  else ( @lv_tax_low * 100 ) end as tax_per,

  case
  when b~netwr gt 50000 then ( b~netwr * @lv_tax_high )
  else ( b~netwr * @lv_tax_low ) end as tax_value
  from vbak as a join vbap as b on a~vbeln = b~vbeln
  into table @data(gt_so) up to 100 rows.
  cl_demo_output=>display( gt_so ).
```

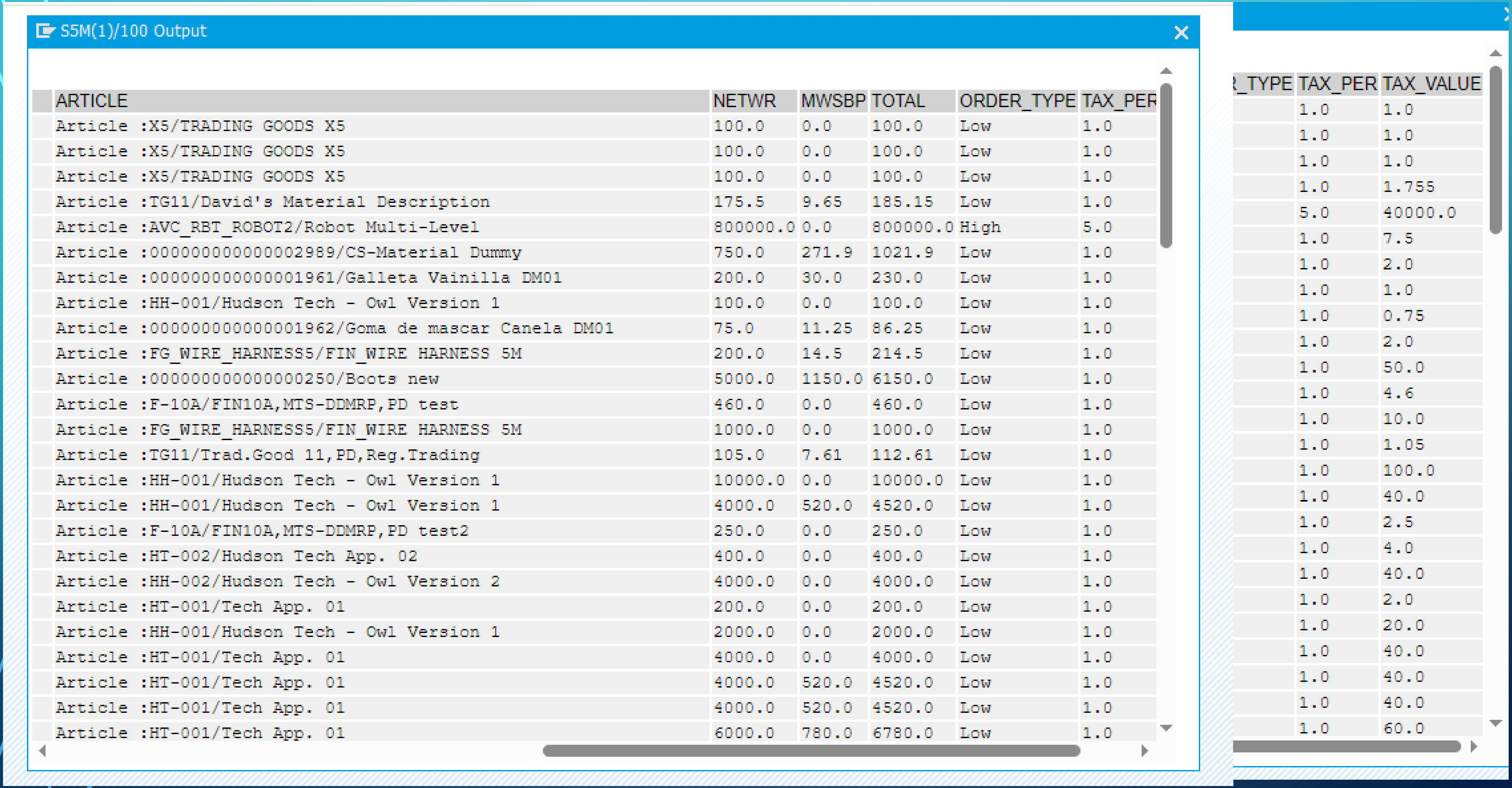
NEW ABAP SYNTAX – SQL FEATURES- RESULT



GT_SO

VBELN	KUNNR	ERDAT	AUART	POSNR	MATNR	ARKTX	ARTICLE
0000000001	0017100001	2022-05-07	TA	000010	X5	TRADING GOODS X5	Article
0000000002	0017100001	2022-05-07	TA	000010	X5	TRADING GOODS X5	Article
0000000003	0017100001	2022-05-11	TA	000010	X5	TRADING GOODS X5	Article
0000000359	0017100004	2023-02-28	TA	000010	TG11	David's Material Description	Article
0000000147	0017100001	2022-10-24	TA	000001	AVC_RBT_ROBOT2	Robot Multi-Level	Article
0070000016	BP1710	2024-03-11	L2	000020	000000000000002989	CS-Material Dummy	Article
0000000940	0000000585	2024-03-12	TA	000010	000000000000001961	Galleta Vainilla DM01	Article
0000000104	0001000001	2022-08-24	ZCAN	000010	HH-001	Hudson Tech - Owl Version 1	Article
0000000940	0000000585	2024-03-12	TA	000020	000000000000001962	Goma de mascar Canela DM01	Article
0000000941	0017100100	2024-03-13	TA	000010	FG_WIRE_HARNESS5	FIN_WIRE HARNESS 5M	Article
0000000267	0000000082	2023-02-05	TA	000010	000000000000000250	Boots new	Article
0000001027	0017100100	2024-06-26	TA	000010	F-10A	FIN10A,MTS-DDMRP,PD test	Article
0020000089	0017100100	2024-03-14	AG	000010	FG_WIRE_HARNESS5	FIN_WIRE HARNESS 5M	Article
0000000301	CUSTUSTM5	2023-02-13	TA	000010	TG11	Trad.Good 11,PD,Reg.Trading	Article
0000000006	0001000000	2022-06-27	Z057	000010	HH-001	Hudson Tech - Owl Version 1	Article
0000000009	0001000000	2022-06-28	Z057	000010	HH-001	Hudson Tech - Owl Version 1	Article
0000001027	0017100100	2024-06-26	TA	000020	F-10A	FIN10A,MTS-DDMRP,PD test2	Article
0000000011	0001000001	2022-06-29	Z057	000010	HT-002	Hudson Tech App. 02	Article
0000000011	0001000001	2022-06-29	Z057	000020	HH-002	Hudson Tech - Owl Version 2	Article
0000000011	0001000001	2022-06-29	Z057	000030	HT-001	Tech App. 01	Article
0000000011	0001000001	2022-06-29	Z057	000040	HH-001	Hudson Tech - Owl Version 1	Article
0000000012	0001000001	2022-06-29	Z057	000010	HT-001	Tech App. 01	Article
0000000013	0001000000	2022-06-29	Z057	000010	HT-001	Tech App. 01	Article
0000000014	0001000000	2022-06-29	Z057	000010	HT-001	Tech App. 01	Article
0000000015	0001000000	2022-06-29	Z057	000010	HT-001	Tech App. 01	Article

NEW ABAP SYNTAX – SQL FEATURES- RESULT

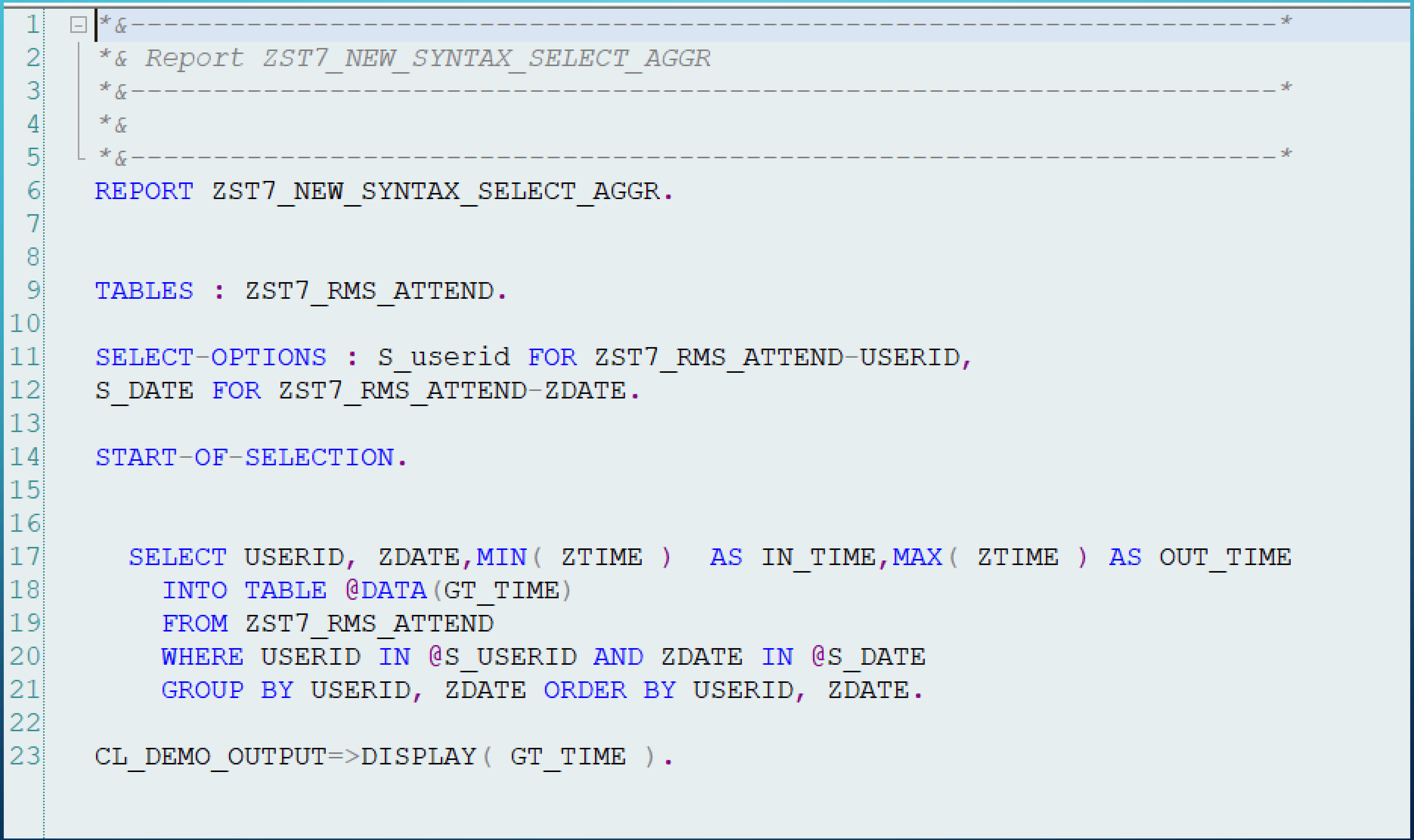


S5M(1)/100 Output

ARTICLE	NETWR	MWSBP	TOTAL	ORDER_TYPE	TAX_PER
Article :X5/TRADING GOODS X5	100.0	0.0	100.0	Low	1.0
Article :X5/TRADING GOODS X5	100.0	0.0	100.0	Low	1.0
Article :X5/TRADING GOODS X5	100.0	0.0	100.0	Low	1.0
Article :TG11/David's Material Description	175.5	9.65	185.15	Low	1.0
Article :AVC_RBT_ROBOT2/Robot Multi-Level	800000.0	0.0	800000.0	High	5.0
Article :000000000000002989/CS-Material Dummy	750.0	271.9	1021.9	Low	1.0
Article :000000000000001961/Galleta Vainilla DM01	200.0	30.0	230.0	Low	1.0
Article :HH-001/Hudson Tech - Owl Version 1	100.0	0.0	100.0	Low	1.0
Article :000000000000001962/Goma de mascar Canela DM01	75.0	11.25	86.25	Low	1.0
Article :FG_WIRE_HARNESS5/FIN_WIRE HARNESS 5M	200.0	14.5	214.5	Low	1.0
Article :000000000000000250/Boots new	5000.0	1150.0	6150.0	Low	1.0
Article :F-10A/FIN10A,MTS-DDMRP,PD test	460.0	0.0	460.0	Low	1.0
Article :FG_WIRE_HARNESS5/FIN_WIRE HARNESS 5M	1000.0	0.0	1000.0	Low	1.0
Article :TG11/Trad.Good 11,PD,Reg.Trading	105.0	7.61	112.61	Low	1.0
Article :HH-001/Hudson Tech - Owl Version 1	10000.0	0.0	10000.0	Low	1.0
Article :HH-001/Hudson Tech - Owl Version 1	4000.0	520.0	4520.0	Low	1.0
Article :F-10A/FIN10A,MTS-DDMRP,PD test2	250.0	0.0	250.0	Low	1.0
Article :HT-002/Hudson Tech App. 02	400.0	0.0	400.0	Low	1.0
Article :HH-002/Hudson Tech - Owl Version 2	4000.0	0.0	4000.0	Low	1.0
Article :HT-001/Tech App. 01	200.0	0.0	200.0	Low	1.0
Article :HH-001/Hudson Tech - Owl Version 1	2000.0	0.0	2000.0	Low	1.0
Article :HT-001/Tech App. 01	4000.0	0.0	4000.0	Low	1.0
Article :HT-001/Tech App. 01	4000.0	520.0	4520.0	Low	1.0
Article :HT-001/Tech App. 01	4000.0	520.0	4520.0	Low	1.0
Article :HT-001/Tech App. 01	6000.0	780.0	6780.0	Low	1.0

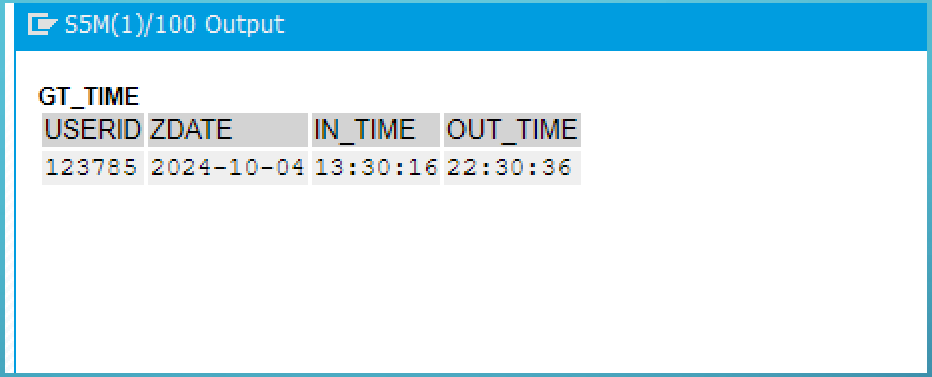
_TYPE	TAX_PER	TAX_VALUE
	1.0	1.0
	1.0	1.0
	1.0	1.0
	1.0	1.755
	5.0	40000.0
	1.0	7.5
	1.0	2.0
	1.0	1.0
	1.0	0.75
	1.0	2.0
	1.0	50.0
	1.0	4.6
	1.0	10.0
	1.0	1.05
	1.0	100.0
	1.0	40.0
	1.0	2.5
	1.0	4.0
	1.0	40.0
	1.0	2.0
	1.0	20.0
	1.0	40.0
	1.0	40.0
	1.0	40.0
	1.0	60.0

SELECT AGGREGATIONS SAMPLE PROGRAM



```
*&---------------------------------------------------------------------*
*& Report ZST7_NEW_SYNTAX_SELECT_AGGR
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
REPORT ZST7_NEW_SYNTAX_SELECT_AGGR.


TABLES : ZST7_RMS_ATTEND.

SELECT-OPTIONS : S_userid FOR ZST7_RMS_ATTEND-USERID,
S_DATE FOR ZST7_RMS_ATTEND-ZDATE.

START-OF-SELECTION.


  SELECT USERID, ZDATE,MIN( ZTIME )  AS IN_TIME,MAX( ZTIME ) AS OUT_TIME
    INTO TABLE @DATA(GT_TIME)
    FROM ZST7_RMS_ATTEND
    WHERE USERID IN @S_USERID AND ZDATE IN @S_DATE
    GROUP BY USERID, ZDATE ORDER BY USERID, ZDATE.

CL_DEMO_OUTPUT=>DISPLAY( GT_TIME ) .
```

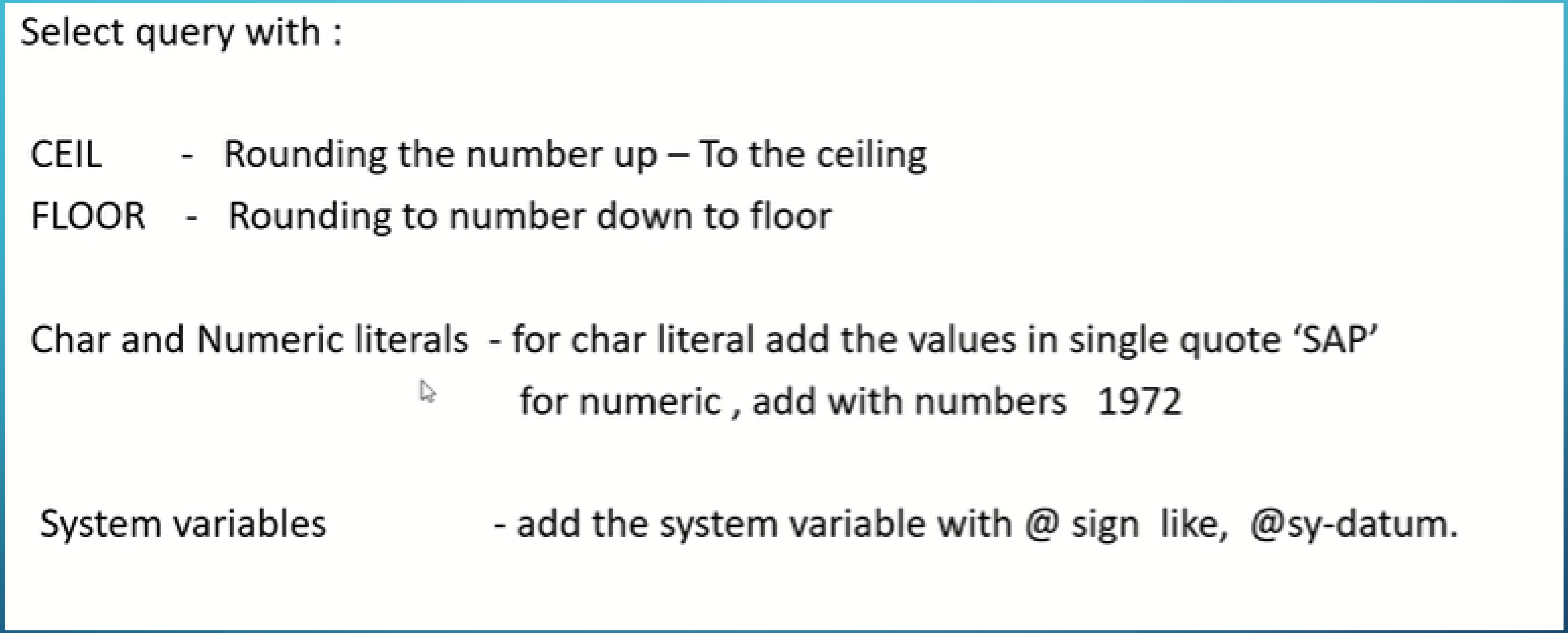
SELECT AGGREGATIONS SAMPLE PROGRAM-RESULT



GT_TIME

USERID	ZDATE	IN_TIME	OUT_TIME
123785	2024-10-04	13:30:16	22:30:36

ROUNDING THE NUMBERS



Select query with :

CEIL - Rounding the number up – To the ceiling

FLOOR - Rounding to number down to floor

Char and Numeric literals - for char literal add the values in single quote ‘SAP’

for numeric , add with numbers 1972

System variables - add the system variable with @ sign like, @sy-datum.