SUBTRACT

This chapter covers the following topics:

- Function
- Syntax Description
- Example

Related Statements: ADD | COMPRESS | COMPUTE | DIVIDE | EXAMINE | MOVE | MOVE ALL | MULTIPLY | RESET | SEPARATE

Belongs to Function Group: *Arithmetic and Data Movement Operations*

Function

The SUBTRACT statement is used to subtract the values of two or more operands.

If a database field is used as the result field, the SUBTRACT operation only results in an update to the internal value that is used within the program. The value for the field in the database remains unchanged.

Syntax Description

Syntax 1 - SUBTRACT

SUBTRACT [**ROUNDED**] *operand1* **...** **FROM** *operand2*

Operand Definition Table:

Operand	Possible Structure					Possible Formats											Referencing Permitted	Dynamic Definition
operand1	C	S	A		N		N	P	I	F		D	T					yes	no
operand2		S	A		M		N	P	I	F		D	T					yes	no

Syntax Element Description:

operand1* FROM *operand2	**Operands:** *operand1* is the minuend, *operand2* is the subtrahend, hence the statement is equivalent to: <*oper2*> := <*oper2*> - <*oper1*> As for the formats of the operands, see also *Rules for Arithmetic Assignments*, *Performance Considerations for Mixed Formats* (in the *Programming Guide*).
ROUNDED	**Rounding:** If you specify the keyword `ROUNDED`, the result will be rounded. For information on rounding, see *Rules for Arithmetic Assignment, Field Truncation and Field Rounding* (in the *Programming Guide*).

Syntax 2

`SUBTRACT` [**`ROUNDED`**] *operand1* **`...`** **`FROM`** *operand2* **`GIVING`** *operand3*

Operand Definition Table:

Operand	Possible Structure					Possible Formats												Referencing Permitted	Dynamic Definition
operand1	C	S	A		N			N	P	I	F		D	T					yes	no
operand2	C	S	A		N			N	P	I	F		D	T					yes	no
operand3		S	A		M	A	U	N	P	I	F	B*	D	T					yes	yes

* Format B of *operand3* may be used only with a length of less than or equal to 4.

Syntax Element Description:

GIVING	**Result Field:** If the GIVING clause is used, *operand2* will not be modified and the result will be stored in *operand3*.
operand1* FROM *operand2* GIVING *operand3	**Operands:** *operand2* is the minuend, *operand1* is the subtrahend, *operand3* is the result field, hence the statement is equivalent to: <*oper3*> := <*oper2*>- <*oper1*> As for the formats of the operands, see also the section *Performance Considerations for Mixed Formats* (in the *Programming Guide*).
ROUNDED	**Rounding:** If you specify the keyword ROUNDED, the result will be rounded. For information on rounding, see *Rules for Arithmetic Assignment*, *Field Truncation and Field Rounding* (in the *Programming Guide*).

Example

```
** Example 'SUBEX1': SUBTRACT
************************************************************************
DEFINE DATA LOCAL
1 #A (P2) INIT <50>
1 #B (P2)
1 #C (P1.1) INIT <2.4>
END-DEFINE
*
SUBTRACT 6 FROM #A
WRITE NOTITLE 'SUBTRACT 6 FROM #A              '  10X '=' #A
*
SUBTRACT 6 FROM 11 GIVING #A
WRITE         'SUBTRACT 6 FROM 11 GIVING #A    '  10X '=' #A
*
SUBTRACT 3 4 FROM #A GIVING #B
WRITE         'SUBTRACT 3 4 FROM #A GIVING #B  '  10X '=' #A '=' #B
*
SUBTRACT -3 -4 FROM #A GIVING #B
WRITE         'SUBTRACT -3 -4 FROM #A GIVING #B'  10X '=' #A '=' #B
*
SUBTRACT ROUNDED 2.06 FROM #C
WRITE         'SUBTRACT ROUNDED 2.06 FROM #C   '  10X '=' #C
*
END
```

Output of Program SUBEX1:

```
SUBTRACT 6 FROM #A                      #A:  44
SUBTRACT 6 FROM 11 GIVING #A            #A:   5
SUBTRACT 3 4 FROM #A GIVING #B          #A:   5 #B:  -2
SUBTRACT -3 -4 FROM #A GIVING #B        #A:   5 #B:  12
SUBTRACT ROUNDED 2.06 FROM #C           #C:  0.3
```