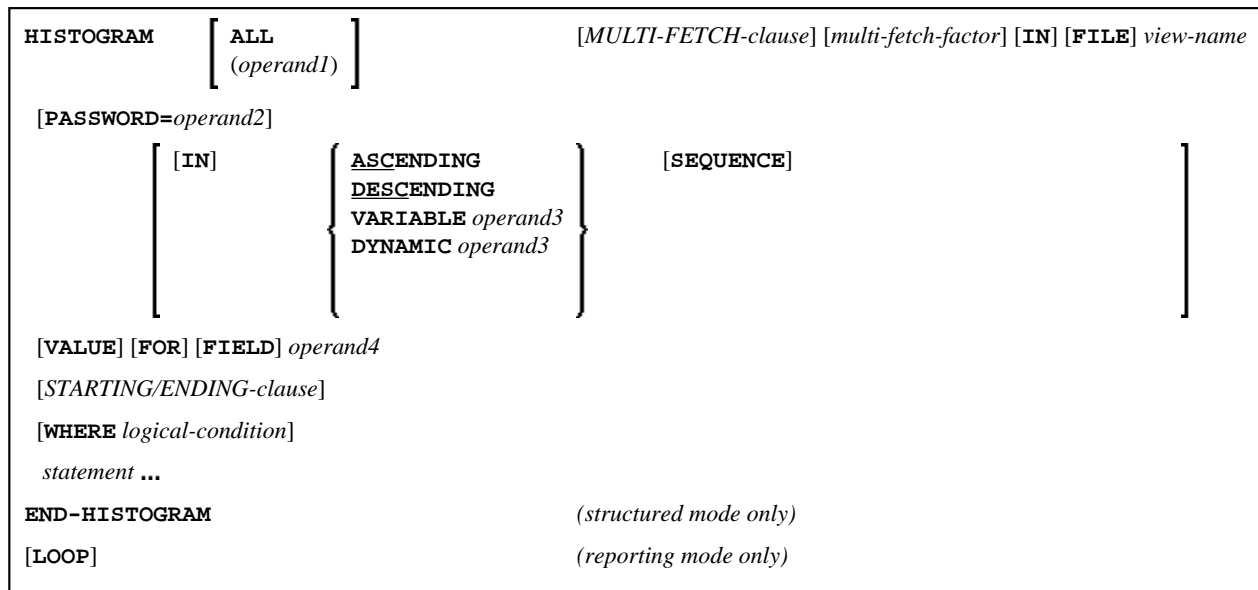


# HISTOGRAM



This chapter covers the following topics:

- Function
- Restrictions
- Syntax Description
- Examples

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Related Statements: ACCEPT/REJECT | AT BREAK | AT START OF DATA | AT END OF DATA | BACKOUT TRANSACTION | BEFORE BREAK PROCESSING | DELETE | END TRANSACTION | FIND | GET | GET SAME | GET TRANSACTION DATA | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | STORE | UPDATE

Belongs to Function Group: *Database Access and Update*

## Function

The HISTOGRAM statement is used to read the values of a database field which is defined as a descriptor, subdescriptor, or a superdescriptor. The values are read directly from the Adabas inverted lists.

The HISTOGRAM statement causes a processing loop to be initiated but does not provide access to any database fields other than the field specified in the HISTOGRAM statement.

See also *HISTOGRAM Statement* (in the *Programming Guide*).

**Note:**

For SQL databases: HISTOGRAM returns the number of rows which have the same value in a specific column.

## Restrictions

- This statement cannot be used with XML databases.
- This statement cannot be used with Entire System Server.

## Syntax Description

Operand Definition Table:

Operand	Possible Structure		Possible Formats													Referencing Permitted	Dynamic Definition					
<i>operand1</i>	C	S							N	P	I	B	*								yes	no
<i>operand2</i>	C	S				A															yes	no
<i>operand3</i>		S				A															yes	no
<i>operand4</i>		S				A	N	P	I	F	B	D	T	L							no	no

\* Format B of *operand1* may be used only with a length of less than or equal to 4.

Syntax Element Description:

Syntax Element	Description
<i>operand1</i> / ALL	<p><b>Processing Loop Limit:</b></p> <p>You can limit the number of descriptor values to be processed with the HISTOGRAM statement by specifying <i>operand1</i> - either as a numeric constant (0 - 4294967295) or as a user-defined variable (containing an integer value).</p> <p>ALL may optionally be specified to emphasize that all descriptor values are to be processed.</p> <p>For this statement, the specified limit has priority over a limit set with a LIMIT statement.</p> <p>If a smaller limit is set with the LT parameter (Limit for Processing Loops), the LT limit applies.</p> <p><b>Note:</b> If you wish to process a 4-digit number of descriptor values, specify it with a leading zero (0nnnn); because Natural interprets every 4-digit number enclosed in parentheses as a line-number reference to a statement. <i>operand1</i> is evaluated when the HISTOGRAM loop is entered. If the value of <i>operand1</i> is modified within the HISTOGRAM loop, this does not affect the number of values read.</p>
<i>MULTI-FETCH-clause</i>	<p><b>MULTI-FETCH Clause:</b></p> <p>See <i>MULTI-FETCH Clause</i> below.</p>
<i>view-name</i>	<p><b>View Name:</b></p> <p>As <i>view-name</i>, you specify the name of a view, which is defined either within a DEFINE DATA statement or in a separate global or local data area.</p> <p>The view must not contain any other fields apart from the field used in the HISTOGRAM statement (<i>operand4</i>).</p> <p>If the field in the view is a periodic-group field or multiple-value field that is defined with an index range, only the first occurrence of that range is filled by the HISTOGRAM statement; all other occurrences are not affected by the execution of the HISTOGRAM statement.</p> <p>In reporting mode, <i>view-name</i> is the name of a DDM if no DEFINE DATA LOCAL statement is used.</p>

Syntax Element	Description
PASSWORD= <i>operand2</i>	<p><b>PASSWORD Clause:</b></p> <p>The PASSWORD clause is used to provide a password (<i>operand2</i>) when retrieving data from an Adabas file which is password-protected. See the statements FIND and PASSW for further information.</p>
SEQUENCE	<p><b>SEQUENCE Clause:</b></p> <p>This clause can only be used for Adabas and SQL databases.</p> <p>With this clause, you can determine whether the records are to be read in ascending sequence or in descending sequence.</p> <ul style="list-style-type: none"> <li>● The default sequence is ascending (which may, but need not, be explicitly specified by using the keyword ASCENDING).</li> <li>● If the records are to be read in descending sequence, you specify the keyword DESCENDING.</li> <li>● If, instead of determining it in advance, you want to have the option of determining at runtime whether the records are to be read in ascending or descending sequence, you either specify the keyword VARIABLE or DYNAMIC, followed by a variable (<i>operand3</i>). <i>operand3</i> has to be of format/length A1 and can contain the value A (for "ascending") or D (for "descending"). <ul style="list-style-type: none"> <li>○ If keyword VARIABLE is used, the reading direction (value of <i>operand3</i>) is evaluated at start of the HISTOGRAM processing loop and remains same until the loop is terminated, regardless if the <i>operand3</i> field is altered in the HISTOGRAM loop or not.</li> <li>○ If keyword DYNAMIC is used, the reading direction (value of <i>operand3</i>) is evaluated before every record fetch in the HISTOGRAM processing loop and may be changed from record to record. This allows to change the scroll sequence from ascending to descending (and vice versa) at any place in the HISTOGRAM loop.</li> </ul> </li> </ul> <p>Examples of SEQUENCE clause:</p> <ul style="list-style-type: none"> <li>● <i>Example 2 - HISTOGRAM Statement with Records Read in Descending Sequence</i></li> <li>● <i>Example 3 - HISTOGRAM Statement Using Variable Sequence</i></li> </ul>

Syntax Element	Description
<i>operand4</i>	<p><b>Descriptor:</b></p> <p>As <i>operand4</i>, a descriptor, subdescriptor, superdescriptor or hyperdescriptor may be specified.</p> <p>A descriptor contained within a periodic group may be specified with or without an index. If no index is specified, the descriptor will be selected if the value specified is located in any occurrence. If an index is specified, the descriptor will be selected only if the value is located in the occurrence specified by the index. The index specified must be a constant. An index range must not be used.</p> <p>For a descriptor which is a multiple-value field an index must not be specified; the descriptor will be selected if the value is located in the record regardless of the position of the value.</p>
<i>STARTING-ENDING-clause</i>	<p><b>STARTING/ENDING Clause:</b></p> <p>Starting and ending values may be specified using the keywords <i>STARTING</i> and <i>ENDING</i> (or <i>THRU</i>) followed by a constant or a user-defined variable representing the value with which processing is to begin/end.</p> <p>For further information, see <i>Specifying Starting/Ending Values</i> below.</p>

Syntax Element	Description
<p>WHERE <i>logical-condition</i></p>	<p><b>WHERE Clause:</b></p> <p>The WHERE clause may be used to specify an additional selection criterion (<i>logical-condition</i>) which is evaluated <i>after</i> a value has been read and <i>before</i> any processing is performed on the value (including the AT BREAK evaluation).</p> <p>The descriptor specified in the WHERE clause must be the same descriptor referenced in the HISTOGRAM statement. No other fields from the selected file are available for processing with a HISTOGRAM statement.</p> <p>The syntax for a <i>logical-condition</i> is described in the section <i>Logical Condition Criteria</i> (in the <i>Programming Guide</i>).</p> <p><b>System Variables</b></p> <p>The Natural system variables *ISN, *NUMBER, and *COUNTER are available with the HISTOGRAM statement.</p> <p>*NUMBER and *ISN are only set after the evaluation of the WHERE clause. They must not be used in the logical condition of the WHERE clause.</p> <p>*NUMBER      The system variable *NUMBER contains the number of database records that contain the last value read.</p> <p>For SQL databases, see <i>*NUMBER for SQL Databases</i> in the <i>System Variables</i> documentation.</p> <p>*ISN            The system variable *ISN contains the number of the occurrence in which the descriptor value last read is contained. *ISN will contain 0 if the descriptor is not contained within a periodic group.</p> <p>*ISN is not available for SQL databases.</p> <p>*COUNTER      The system variable *COUNTER contains a count of the total number of values which have been read (after evaluation of the WHERE clause).</p>
<p>END-HISTOGRAM</p>	<p><b>End of HISTOGRAM Statement:</b></p> <p>The Natural reserved word END-HISTOGRAM must be used to end the HISTOGRAM statement.</p>

## MULTI-FETCH Clause

### Note:

This clause can only be used for Adabas databases.

<b>MULTI-FETCH</b> { <b>ON</b> <b>OFF</b> <b>OF</b> <i>multi-fetch-factor</i> }
---

### Note:

[**MULTI-FETCH** **OF** *multi-fetch-factor*] is not evaluated for database types ADA and ADA2. The default processing mode is applied; see profile parameter MFSET. When used in conjunction with database type ADA2, the **MULTI-FETCH** clause is ignored completely; see *Database Management System Assignments* in the *Configuration Utility* documentation.

For more information, see the section *Multi-Fetch Clause (Adabas)* in the *Programming Guide*.

## Specifying Starting/Ending Values

Starting and ending values may be specified using the keywords **STARTING** and **ENDING** (or **THRU**) followed by a constant or a user-defined variable representing the value with which processing is to begin/end.

If a starting value is specified and the value is not present, the next higher value is used as the starting value. If no higher value is present, the **HISTOGRAM** loop will not be entered.

If an ending value is specified, values will be read up to and including the ending value.

Hexadecimal constants may be specified as a starting or ending value for descriptors of format A or B.

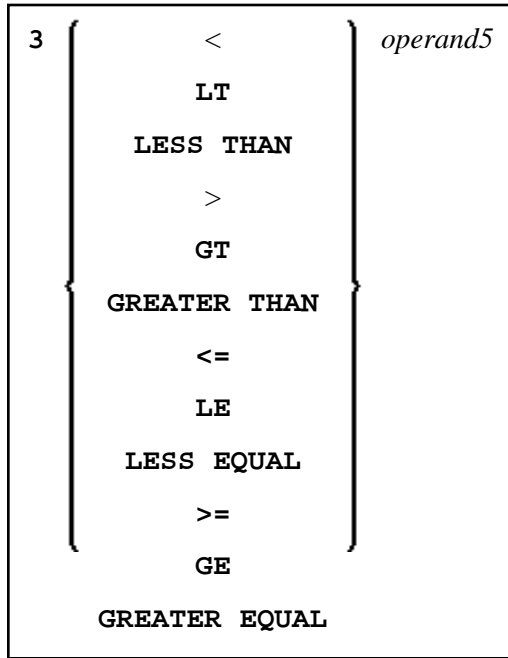
Syntax Option 1:

<b>1</b> [ <b>STARTING</b> ] [ <b>WITH</b> <b>FROM</b> ] [ <u>VALUES</u> ] <i>operand5</i> ] [ [ <b>THRU</b> ] <i>operand6</i> ] [ <b>ENDING AT</b> ]
---

Syntax Option 2:

<b>2</b> [ <b>STARTING</b> ] [ <b>WITH</b> <b>FROM</b> ] [ <u>VALUES</u> ] <i>operand5</i> <b>TO</b> <i>operand6</i>
--

Syntax Option 3:



**Note:**

If the comparators of Diagram 3 are used, the options ENDING AT, THRU and TO may not be used. These comparators are also valid for the READ statement.

Operand Definition Table:

Operand	Possible Structure				Possible Formats												Referencing Permitted	Dynamic Definition
<i>operand5</i>	C	S			A	U	N	P	I	F	B	D	T	L			yes	no
<i>operand6</i>	C	S			A	U	N	P	I	F	B	D	T	L			yes	no

Syntax Element Description:



Syntax Element	Description
STARTING FROM ... ENDING AT   TO	<p><b>STARTING FROM / ENDING AT Clauses:</b></p> <p>The STARTING FROM and ENDING AT clauses are used to limit reading to a user-specified range of values.</p> <p>The STARTING FROM clause (= or EQ or EQUAL TO or [ STARTING ] FROM) determines the starting value for the read operation. If a starting value is specified, reading will begin with the value specified. If the starting value does not exist, the next higher (or lower for a DESCENDING read) value will be returned. If no higher (or lower for DESCENDING) value exists, the HISTOGRAM loop will not be entered.</p> <p>In order to limit the values to an end-value, you may specify an ENDING AT clause with the terms THRU, ENDING AT or TO, that imply an inclusive range. Whenever the descriptor field exceeds the end-value specified, an automatic loop termination is performed. Although the basic functionality of the TO, THRU and ENDING AT keywords looks quite similar, internally they differ in how they work.</p>
THRU   ENDING AT	<p><b>THRU / ENDING AT Option:</b></p> <p>If THRU or ENDING AT is used, only the start-value is supplied to the database, but the end-value check is performed by the Natural runtime system, after the value is returned by the database.</p> <p>The THRU and ENDING AT options can be used for all databases which support the HISTOGRAM statements.</p>
TO	<p><b>Range:</b></p> <p>If the keyword TO is used, both the start-value and the end-value are sent to the database and Natural does not perform checks for value ranges. If the end-value is exceeded, the database reacts in the same way as when "end-of-file" is reached and the database loop is exited. Since the complete range checking is done by the database, the lower-value (of the range) is always supplied in the start-value and the higher-value filled into the end-value, regardless whether you are browsing in ASCENDING or in DESCENDING order.</p>

## Examples

- Example 1 - HISTOGRAM Statement
- Example 2 - HISTOGRAM Statement with Records Read in Descending Sequence
- Example 3 - HISTOGRAM Statement Using Variable Sequence

### Example 1 - HISTOGRAM Statement

```

** Example 'HSTEX1S': HISTOGRAM (structured mode)
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 CITY
END-DEFINE
*
LIMIT 8
HISTOGRAM EMPLOY-VIEW CITY STARTING FROM 'M'
  DISPLAY NOTITLE
    CITY 'NUMBER OF/PERSONS' *NUMBER *COUNTER
END-HISTOGRAM
*
END
    
```

#### Output of Program HSTEX1S:

CITY	NUMBER OF PERSONS	CNT
MADISON	3	1
MADRID	41	2
MAILLY LE CAMP	1	3
MAMERS	1	4
MANSFIELD	4	5
MARSEILLE	2	6
MATLOCK	1	7
MELBOURNE	2	8

Equivalent reporting-mode example: HSTEX1R.

### Example 2 - HISTOGRAM Statement with Records Read in Descending Sequence

```

** Example 'HSTDSCND': HISTOGRAM (with DESCENDING)
*****
DEFINE DATA LOCAL
1 EMPL VIEW OF EMPLOYEES
  2 NAME
END-DEFINE
*
HISTOGRAM (10) EMPL IN DESCENDING SEQUENCE FOR NAME FROM 'ZZZ'
  DISPLAY NAME *NUMBER
END-HISTOGRAM
END
    
```

#### Output of Program HSTDSCND:

Page 1 05-01-13 13:41:03

NAME	NMBR
ZINN	1
YOT	1
YNCLAN	1
YATES	1

```

YALCIN                1
YACKX-COLTEAU        1
XOLIN                 1
WYLLIS                2
WULFRING              1
WRIGHT                1

```

### Example 3 - HISTOGRAM Statement Using Variable Sequence

```

** Example 'HSTVSEQ': HISTOGRAM (with VARIABLE SEQUENCE)
*****
DEFINE DATA LOCAL
1 EMPL VIEW OF EMPLOYEES
  2 NAME
*
1 #DIR      (A1)
1 #STARTVAL (A20)
END-DEFINE
*
SET KEY PF3 PF7 PF8
*
MOVE 'ADKINSON' TO #STARTVAL
*
HISTOGRAM (9) EMPL FOR NAME FROM #STARTVAL
  WRITE NAME *NUMBER
  IF *COUNTER = 5
    MOVE NAME TO #STARTVAL
  END-IF
END-HISTOGRAM
*
#DIR := 'A'
*
REPEAT
  HISTOGRAM EMPL IN VARIABLE #DIR SEQUENCE
    FOR NAME FROM #STARTVAL
    MOVE NAME TO #STARTVAL
    INPUT NO ERASE (IP=OFF AD=0)
      15/01 NAME *NUMBER
      // 'Direction:' #DIR
      // 'Press PF3 to stop'
      / '      PF7 to go step back'
      / '      PF8 to go step forward'
      / '      ENTER to continue in that direction'
    /*
    IF *PF-KEY = 'PF7' AND #DIR = 'A'
      MOVE 'D' TO #DIR
      ESCAPE BOTTOM
    END-IF
    IF *PF-KEY = 'PF8' AND #DIR = 'D'
      MOVE 'A' TO #DIR
      ESCAPE BOTTOM
    END-IF
    IF *PF-KEY = 'PF3'
      STOP
    END-IF
  END-HISTOGRAM
  /*
  IF *COUNTER(0250) = 0
    STOP
  END-IF
END-REPEAT
END

```

**Output of Program HSTVSEQ:**

Page 1

05-01-13 13:50:31

ADKINSON	8
AECKERLE	1
AFANASSIEV	2
AHL	1
AKROYD	1
ALEMAN	1
ALESTIA	1
ALEXANDER	5
ALLEGRE	1

MORE

**After pressing ENTER:**

Page 1

05-01-13 13:50:31

ADKINSON	8
AECKERLE	1
AFANASSIEV	2
AHL	1
AKROYD	1
ALEMAN	1
ALESTIA	1
ALEXANDER	5
ALLEGRE	1

AKROYD 1

Direction: A

Press PF3 to stop  
PF7 to go step back  
PF8 to go step forward  
ENTER to continue in that direction