

READ RESULT SET - SQL

Common Set Syntax:

```

READ [(limit)] RESULT SET result-set INTO { VIEW view-name } FROM ddm-name
                                     {
                                     parameter
                                     }

  [GIVING [:] sql-code]

END-RESULT

```

Extended Set Syntax:

```

READ [(limit)] RESULT SET result-set INTO { VIEW view-name } FROM ddm-name
                                     {
                                     parameter
                                     }

  [WITH INSENSITIVE SCROLL [:] scroll-hv]

  [GIVING [:] sql-code]

  [WITH ROWSET POSITIONING FOR { [:] row_hv } ROWS]
                               {
                               integer
                               }

END-RESULT

```

This chapter covers the following topics:

- Function
- Restriction
- Syntax Description
- Example

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

Belongs to Function Group: *Database Access and Update*

See also *NDB - READ RESULT SET* in the *Natural for DB2* part of the *Database Management System Interfaces* documentation.

Function

The SQL statement `READ RESULT SET` can only be used in conjunction with a `CALLDBPROC` statement. It is used to read a result set which was created by a stored procedure that was invoked by a previous `CALLDBPROC` statement.

Restriction

This statement is not available with Natural for SQL and Natural SQL Gateway.

Syntax Description

<i>limit</i>	You can limit the number of rows to be read. You can specify the limit either as a numeric constant (0 to 4294967295) or as a variable of format N, P or I.
<i>result-set</i>	As result-set you specify a result-set locator variable filled by a preceding CALLDBPROC statement. Result-set has to be a variable of format/length I4. Note: If a syncpoint operation takes place between the CALLDBPROC statement and the READ RESULT SET statement, the result sets can no longer be accessed by the READ RESULT SET statement.
FROM <i>ddm-name</i>	As <i>ddm-name</i> you specify the name of the DDM which is used to "address" the database executing the stored procedure. For more information, see <i>ddm-name</i> .

<p>WITH INSENSITIVE SCROLL [:] <i>scroll_hv</i></p>	<p>This clause belongs to the SQL Extended Set.</p> <p>Using this clause causes the application to use an insensitive scrollable cursor to access the result set created by the previously invoked stored procedure. In order to use this clause, the stored procedure must have created the result set with a scrollable cursor. The <i>scroll_hv</i> has to be an alphanumeric Natural variable which contains the scrolling direction. The <i>scroll_hv</i> will be evaluated each time the <code>READ RESULT SET</code> processing loop is executed.</p> <p>If the <code>GIVING sqlcode</code> option is specified as well, the processing loop will stay open, even if an <i>sqlcode</i> +100 (row not found) is returned from the RDBMS.</p> <p>The processing will be terminated, if the application issues an <code>ESCAPE</code> statement or if the <i>sqlcode</i> +100 (row not found) is encountered five times successively without a terminal I/O.</p> <p>If the <code>GIVING sqlcode</code> option is not specified, the processing loop will be closed, if any <i>sqlcode</i> other than 0 (success) is returned from the RDBMS.</p>
<p>GIVING <i>sqlcode</i></p>	<p>This option may be used to obtain the SQL code of the SQL "fetch" operation used to process the result set.</p> <p>If this option is specified and the SQL code of the SQL operation is not 0, no Natural error message will be issued. In this case, the action to be taken in reaction to the SQL code value has to be coded in the invoking Natural object.</p> <p>The <i>sqlcode</i> field has to be a variable of format/length I4.</p> <p>If the <code>GIVING sqlcode</code> option is omitted, a Natural error message will be issued if the SQL code is not 0.</p>

<p>WITH ROWSET POSITIONING FOR ... ROWS</p>	<p>WITH ROWSET POSITIONING FOR ... ROWS Clause:</p> <p>This clause belongs to the SQL Extended Set.</p> <p>Using this clause causes the application to use an insensitive scrollable cursor to access the result set created by the previously invoked stored procedure. In order to use this clause, the stored procedure must have created the result set with a scrollable cursor. The <i>scroll_hv</i> has to be an alphanumeric Natural variable which contains the scrolling direction. The <i>scroll_hv</i> will be evaluated each time the READ RESULT SET processing loop is executed.</p> <p>If the GIVING <i>sqlcode</i> option is specified as well, the processing loop will stay open, even if an sqlcode +100 (row not found) is returned from the RDBMS.</p> <p>The processing will be terminated, if the application issues an ESCAPE statement or if the sqlcode +100 (row not found) is encountered five times successively without a terminal I/O.</p> <p>If the GIVING <i>sqlcode</i> option is not specified, the processing loop will be closed, if any sqlcode other than 0 (success) is returned from the RDBMS.</p>
<p>END-RESULT</p>	<p>The Natural reserved keyword END-RESULT must be used to end the READ RESULT SET statement.</p>

Example

See the example in the CALLDBPROC statement.

In addition, see the corresponding Natural database interface documentation.