

Using the multiple-row FETCH statement

Last Updated: 2021-03-08

The multiple-row FETCH statement can be used to retrieve multiple rows from a table or view with a single FETCH statement. The program controls the blocking of rows by the number of rows requested on the FETCH statement (The Override Database File (OVRDBF) command has no effect.).

The maximum number of rows that can be requested on a single fetch call is 32 767. After the data is retrieved, the cursor is positioned on the last row retrieved.

There are two ways to define the storage where fetched rows are placed: a host structure array or a row storage area with an associated descriptor. Both methods can be coded in all of the languages supported by the SQL precompilers, with the exception of the host structure array in REXX. Both forms of the multiple-row FETCH statement allow the application to code a separate indicator array. The indicator array should contain one indicator for each host variable that is null capable.

The multiple-row FETCH statement can be used with both serial and scrollable cursors. The operations used to define, open, and close a cursor for a multiple-row FETCH remain the same. Only the FETCH statement changes to specify the number of rows to retrieve and the storage where the rows are placed.

After each multiple-row FETCH, information is returned to the program through the SQLCA. In addition to the SQLCODE and SQLSTATE fields, the SQLERRD provides the following information:

SQLERRD3 contains the number of rows retrieved on the multiple-row FETCH statement. If SQLERRD3 is less than the number of rows requested, then an error or end-of-data condition occurred.

SQLERRD4 contains the length of each row retrieved.

SQLERRD5 contains an indication that the last row in the table was fetched. It can be used to detect the end-of-data condition in the table being fetched when the cursor does not have immediate sensitivity to updates. Cursors which do have immediate sensitivity to updates should continue fetching until an SQLCODE +100 is received to detect an end-of-data condition.