**RETR statement (Retrieve)**

RETR (Retrieve) obtains data from a table based on a search argument.

1. VisualAge Generator: Programmer's Reference

**RETR**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ÊÊ |  | RETR |  | *dataitem1* |  | *table* |  |  |  |  |  |  |  | *dataitem2* |  |  |  |  |  | ; |  | ÊÍ |
|  |  |  |  |  | . |  | *search column* |  |  |  |  | *return column* |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Attribute** | **Description** |
|  |  |
| dataitem1 | Data item name or literal. The data item can be subscripted, qualified, |
|  | or both. |
|  |  |
| table | Name of a table. |
|  |  |
| search column | Name of a column in the table. The default is the first column in the |
|  | table. |
|  |  |
| dataitem2 | Data item name (cannot be literal). The data item can be subscripted, |
|  | qualified, or both. This data item receives the value of the return |
|  | column. |
|  |  |
| return column | Name of a column in the table. The default is the second column in |
|  | the table. |
|  |  |

**Definition considerations for RETR**

If the value in dataitem1 is found in the search column of a row in the table, the data in the return column of the same row is moved to dataitem2. If the

data in the search column is not unique, the first occurrence in the table is the one used.

The special function word EZETST is loaded with a value depending on the results of the RETR statement. The contents of EZETST will be either:

1. Zero if the data item is not found
2. The row number where the data item is located if the data item is found.

When EZETST contains a row number, it can be used as a subscript for other statements that reference other columns in that same row of the table.

**Target environments for RETR**

Supported in all environments without compatibility considerations.

**Examples for RETR**

In the following example, a table (INFO) has 50 rows and 3 columns called STATE, POPULATION, and AREA. The first column has an entry for each of the 50 states, the second column contains the population for each state, and the third column contains the area in square miles for each state.

INFO:

|  |  |  |
| --- | --- | --- |
| STATE | POPULATION | AREA |
|  |  |  |
| Alabama | 3,500,000 | 51,600 |
|  |  |  |

Chapter 10. Program processing statements **419**

**RETR**

|  |  |  |
| --- | --- | --- |
| Alaska | 302,000 | 586,000 |
|  |  |  |
| . | . | . |
|  |  |  |
| . | . | . |
|  |  |  |

A RETR statement could be used to pick up the area information from the above table, based on a matching state.

MOVE ©ALASKA© TO ITEM;

RETR ITEM INFO.STATE AMOUNT AREA;

AMOUNT now has 586,000 in it. EZETST contains 2, the row number of the matching state. If no match is made, EZETST is set to 0 and the contents of AMOUNT are not changed. If the match is found, you can now obtain the population for ALASKA by the following statement:

MOVE INFO.POPULATION[EZETST] TO PEOPLE;

You could also code:

PEOPLE = INFO.POPULATION[EZETST];