

Database Processing Loops

This document discusses processing loops required to process data that have been selected from a database as a result of a FIND, READ or HISTOGRAM statement.

The following topics are covered:

- Creation of Database Processing Loops
 - Hierarchies of Processing Loops
 - Example of Nested FIND Loops Accessing the Same File
 - Further Examples of Nested READ and FIND Statements
-

Creation of Database Processing Loops

Natural automatically creates the necessary processing loops which are required to process data that have been selected from a database as a result of a FIND, READ or HISTOGRAM statement.

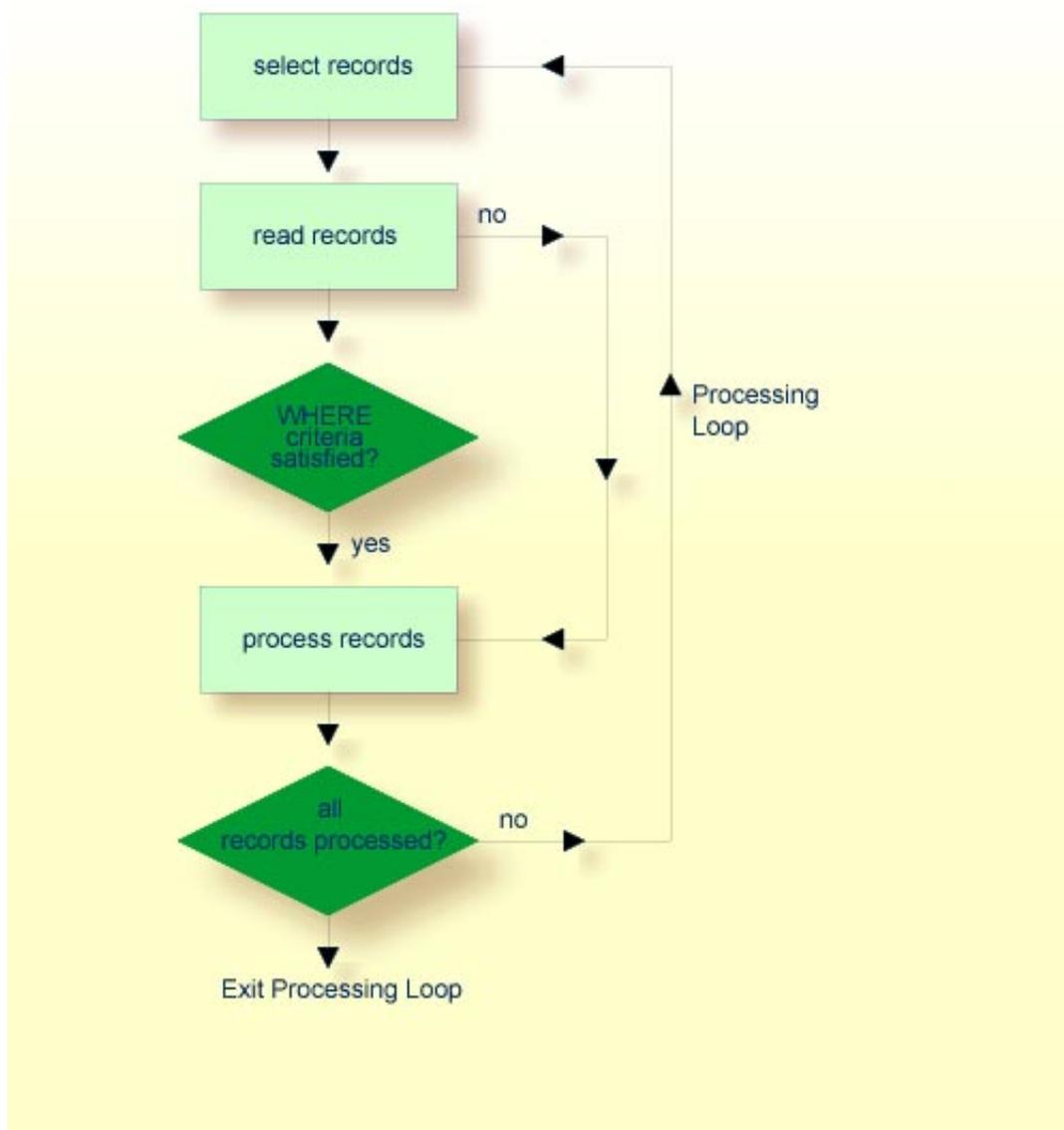
Example:

In the following example, the FIND loop selects all records from the EMPLOYEES file in which the field NAME contains the value "ADKINSON" and processes the selected records. In this example, the processing consists of displaying certain fields from each record selected.

```
** Example Program 'FINDX03'  
DEFINE DATA LOCAL  
1 MYVIEW VIEW OF EMPLOYEES  
  2 NAME  
  2 FIRST-NAME  
  2 CITY  
END-DEFINE  
*  
FIND MYVIEW WITH NAME = 'ADKINSON'  
  DISPLAY NAME FIRST-NAME CITY  
END-FIND  
END
```

If the FIND statement contained a WHERE clause in addition to the WITH clause, only those records that were selected as a result of the WITH clause **and** met the WHERE criteria would be processed.

The following diagram illustrates the flow logic of a database processing loop:



Hierarchies of Processing Loops

The use of multiple FIND and/or READ statements creates a hierarchy of processing loops, as shown in the following example:

Example of Processing Loop Hierarchy

```

** Example Program 'FINDX04'
DEFINE DATA LOCAL
1 PERSONVIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
1 AUTOVIEW VIEW OF VEHICLES
  2 PERSONNEL-ID
  2 MAKE
  2 MODEL
END-DEFINE
*

```

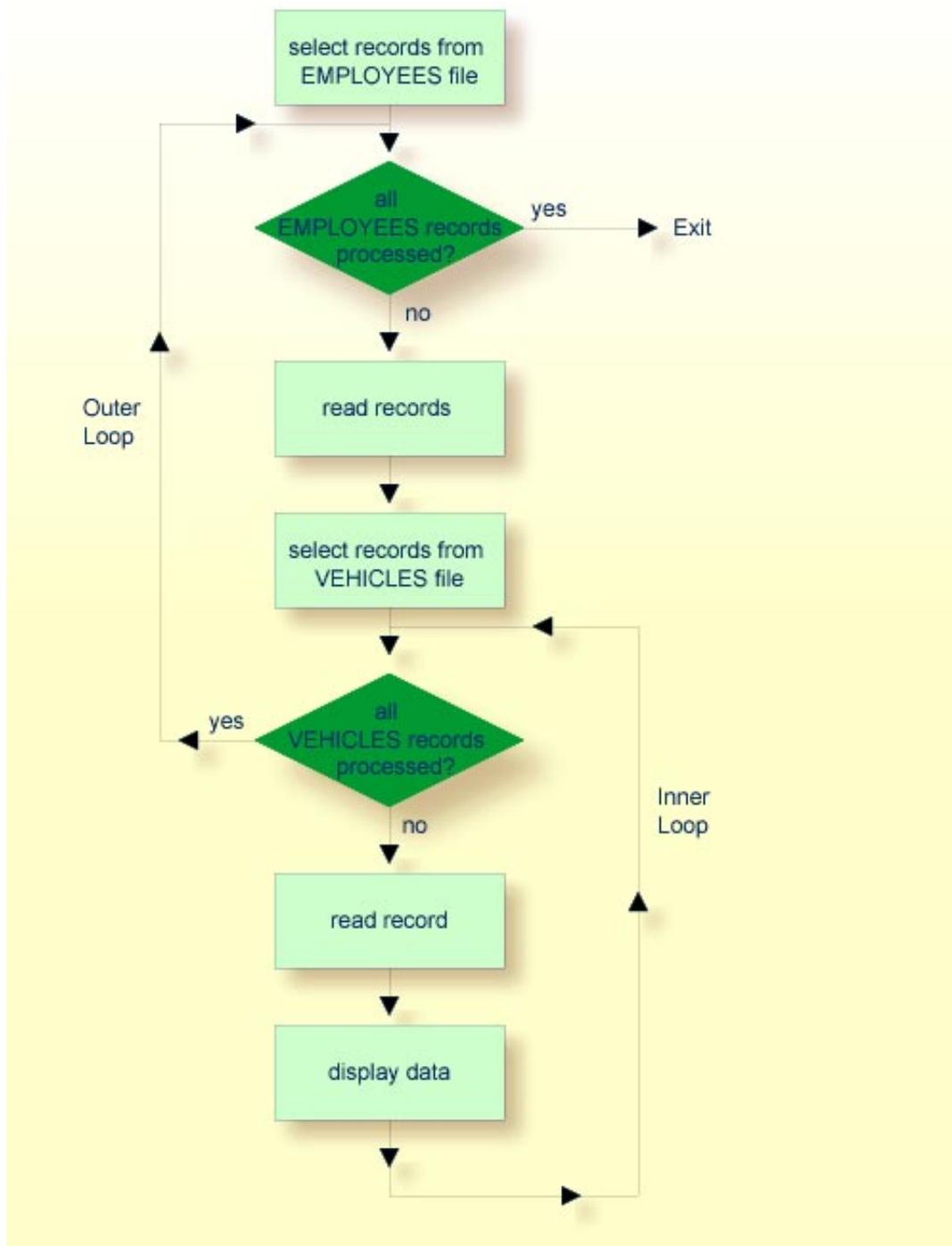
```
EMP. FIND PERSONVIEW WITH NAME = 'ADKINSON'  
VEH.  FIND AUTOVIEW WITH PERSONNEL-ID = PERSONNEL-ID (EMP.)  
      DISPLAY NAME MAKE MODEL  
      END-FIND  
      END-FIND  
END
```

The above program selects from the EMPLOYEES file all people with the name "ADKINSON". Each record (person) selected is then processed as follows:

1. The second FIND statement is executed to select the automobiles from the VEHICLES file, using as selection criterion the PERSONNEL-IDs from the records selected from the EMPLOYEES file with the first FIND statement.
2. The NAME of each person selected is displayed; this information is obtained from the EMPLOYEES file. The MAKE and MODEL of each automobile owned by that person is also displayed; this information is obtained from the VEHICLES file.

The second FIND statement creates an inner processing loop within the outer processing loop of the first FIND statement, as shown in the following diagram.

The diagram illustrates the flow logic of the hierarchy of processing loops in the previous example program:



Example of Nested FIND Loops Accessing the Same File

It is also possible to construct a processing loop hierarchy in which the same file is used at both levels of the hierarchy:

```

** Example Program 'FINDX05'
  DEFINE DATA LOCAL
  1 PERSONVIEW VIEW OF EMPLOYEES
    2 NAME
    2 FIRST-NAME
  
```

```

      2 CITY
1 #NAME (A40)
END-DEFINE
*
WRITE TITLE LEFT JUSTIFIED
  'PEOPLE IN SAME CITY AS:' #NAME / 'CITY:' CITY SKIP 1
FIND PERSONVIEW WITH NAME = 'JONES'
      WHERE FIRST-NAME = 'LAUREL'
      compress NAME FIRST-NAME INTO #NAME
      FIND PERSONVIEW WITH CITY = CITY
      DISPLAY NAME FIRST-NAME CITY
END-FIND
END-FIND
END

```

The above program first selects all people with name "JONES" and first name "LAUREL" from the EMPLOYEES file. Then all who live in the same city are selected from the EMPLOYEES file and a list of these people is created. All field values displayed by the DISPLAY statement are taken from the second FIND statement.

PEOPLE IN SAME CITY AS: JONES LAUREL		
CITY: BALTIMORE		
NAME	FIRST-NAME	CITY

JENSEN	MARTHA	BALTIMORE
LAWLER	EDDIE	BALTIMORE
FORREST	CLARA	BALTIMORE
ALEXANDER	GIL	BALTIMORE
NEEDHAM	SUNNY	BALTIMORE
ZINN	CARLOS	BALTIMORE
JONES	LAUREL	BALTIMORE

Further Examples of Nested READ and FIND Statements

See the following example programs in library SYSEXP:

- READX04
- LIMITX01