Cobol o fabuloso comando SEARCH

Salve jovem padawan, vamos falar sobre velharias do tiozão, vou apresentar um comando COBOL, único, grande amigo da performance e gerando grandes economias de ciclos de CPU, ao trazer uma tabela externa para o interior do programa.

Imagine uma tabela secundaria com alguns parâmetros, um programador preguiçoso codificaria um joiner, gerando consumo de SQL indevidamente, pois imagine se a tabela pai possuir 10.000 registros, teríamos que acessar esta tabela 10.000 vezes num SUBSELECT com opens e closes de cursor; ou sobrecarregando o SGBD com SELECT e Joiner, para exibir tuplas.

O maior inimigo da performance e otimização são as querys malfeitas, os programadores que especificaram a linguagem COBOL, criaram um comando que varre um array através de chaves.

Visando evitar problemas com consumo excessivo de Base de Dados, recomendo o uso de Tabelas Dinâmicas.

O que são Tabelas Dinâmicas?

Muito usada em programação, nada mais é que o uso de arrays, para criarem tabelas unidimensional ou multidimensional, carregadas em tempo de execução, visando economizar paradas e passagens de ponteiro para rotinas externas ao programa, possui alguns malefícios, devendo ser analisado caso a caso, para ver qual a melhor estratégia.

Conheça o comando SEARCH

É um comando que permite a navegação por tabelas de modo incremental, semelhante ao PERFORM UNTIL e pesquisa por chave. Otimizado para auxiliar o processamento dinâmico, criando tabelas em tempo de execução.

Sua sintaxe básica é

SEARCH indetificador1

AT END

MOVE valor TO contador

WHEN ocorrência-chave (índice) EQUAL chave-de-pesquisa

MOVE ocorrência-item TO variável

END-SEARCH

A seguir deixo um código exemplo, que permite pesquisar uma tabela através da Sigla da Unidade da Federação retornando o nome do Estado.

Sua sintaxe é simples e intuitiva, permitindo um fácil entendimento, depuração e inclusão de novos membros, podendo ser criada dinamicamente através de leitura a arquivo sequencial, acesso simples ao banco de dados, ou uma maneira menos performática em hard coded.

\*-----------------------------------------------------------------

WORKING-STORAGE SECTION.

\*-----------------------------------------------------------------

01 WS-TABELA-ESTADOS.

05 TAB-NOME-UF.

10 FILLER PIC X(021) VALUE "ACACRE ".

10 FILLER PIC X(021) VALUE "ALALAGOAS ".

10 FILLER PIC X(021) VALUE "AMAMAZONAS ".

10 FILLER PIC X(021) VALUE "APAMAPA ".

10 FILLER PIC X(021) VALUE "BABAHIA ".

10 FILLER PIC X(021) VALUE "CECEARA ".

10 FILLER PIC X(021) VALUE "DFDISTRITO FEDERAL ".

10 FILLER PIC X(021) VALUE "ESESPIRITO SANTO ".

10 FILLER PIC X(021) VALUE "GOGOIAS ".

10 FILLER PIC X(021) VALUE "MAMARANHAO ".

10 FILLER PIC X(021) VALUE "MGMINAS GERAIS ".

10 FILLER PIC X(021) VALUE "MSMATO GROSSO DO SUL ".

10 FILLER PIC X(021) VALUE "MTMATO GROSSO ".

10 FILLER PIC X(021) VALUE "PAPARA ".

10 FILLER PIC X(021) VALUE "PBPARAIBA ".

10 FILLER PIC X(021) VALUE "PEPERNAMBUCO ".

10 FILLER PIC X(021) VALUE "PIPIAUI ".

10 FILLER PIC X(021) VALUE "PRPARANA ".

10 FILLER PIC X(021) VALUE "RJRIO DE JANEIRO ".

10 FILLER PIC X(021) VALUE "RNRIO GRANDE DO NORTE".

10 FILLER PIC X(021) VALUE "RORONDONIA ".

10 FILLER PIC X(021) VALUE "RRRORAIAMA ".

10 FILLER PIC X(021) VALUE "RSRIO GRANDE DO SUL ".

10 FILLER PIC X(021) VALUE "SCSANTA CATARINA ".

10 FILLER PIC X(021) VALUE "SESERGIPE ".

10 FILLER PIC X(021) VALUE "SPSAO PAULO ".

10 FILLER PIC X(021) VALUE "TOTOCANTINS ".

05 TAB-NOME-UF-R REDEFINES TAB-NOME-UF.

07 WS-CAJUGUDU OCCURS 27 TIMES

DESCENDING KEY SIGLA-UF

INDEXED BY IX-TAB.

10 SIGLA-UF PIC X(002).

10 NOME-UF PIC X(019).

01 WS-AUXILIARES.

05 NAO-ACHOU-NADA PIC 9(003) VALUE ZEROES.

05 WS-SIGLA-UF PIC X(002) VALUE SPACES.

\*-----------------------------------------------------------------

PROCEDURE DIVISION.

\*-----------------------------------------------------------------

MOVE ZEROES TO NAO-ACHOU-NADA

MOVE 'SP' TO WS-SIGLA-UF

SET IX-TAB TO 1

SEARCH WS-CAJUGUDU

AT END

MOVE 1 TO NAO-ACHOU-NADA

WHEN SIGLA-UF (IX-TAB) EQUAL WS-SIGLA-UF

MOVE NOME-UF (IX-TAB) TO VARIAVEL

END-SEARCH

IF NAO-ACHOU-NADA EQUAL 1

codifique as instruções necessárias

END-IF.

|  |  |  |
| --- | --- | --- |
| SEARCH | Versus | SEARCH ALL |
| Pesquisa realizada de forma linear/sequencial |  | Pesquisa realizada de forma binaria |
| Leitura dos dados é lenta |  | Leitura realizada de forma binaria |
| Os itens do array/tabela dinâmica nao precisam estar em qualquer ordem |  | As entradas da tabela devem estar ordenadas |
| Uma ou mais condições WHEN podem ser codificadas |  | Somente uma condição WHEN pode ser codificada |
| O campo chave nao precisa estar ordenado |  | é necessário que os dados do campo chave estejam ordenados |
| inicialização do index é necessária |  | incrementarão do index é feita automaticamente pelo sistema |
| pode ser usado na expressão condicional os operadores = > < >= <= not= |  | Somente o operador de igualdade pode ser usado |
| funciona tanto para matrizes unidimensionais como para multidimensionais |  | é usado apenas para uma matriz unidimensional |

O que é código Hard Coded?

É uma técnica não recomendada, trata-se do mau costume de deixar parâmetros fixo dentro do código do programa, obrigando em caso de mudanças a edição, compilação e guarda em produção de um programa.

Técnica arriscada, pois pode gerar bugs ocultos e abends em programas saudáveis, desta forma, o mais aconselhável é criar um Função dentro de um sub-programa ou num arquivo sequencial.