Usando VARCHAR em COBOL

Salve jovem padawan, quem diria estamos aqui escrevendo sobre VARCHAR no ambiente mainframe, seu uso é similar ao de outras linguagens, mas a parte divertida e interessante é que podemos ver o seu funcionamento em natura.

Ajudando enormemente no processamento de grandes áreas de memória e ao mesmo tempo sendo prático e econômico, ao codificarmos temos que ter em mente, o custo de CPU, a alocação de espaço em disco e memória, afinal em tempos de processamento distribuído, cloude computer e uso indiscriminado de PDAs.

Programas mau-escritos podem gerar prejuízos e consumir toda a receita extra de um produto recém lançado no mercado, imagine 1 mega a mais trafegando em cada requisição, multiplicando por milhares de acessos por cada usuário.

Recentemente tivemos o caso da HBO MAX que deixo muitos clientes irritados, há alguns anos tivemos o naufrágio do TSB e basta olhar com atenção aos jornais econômicos, encontraremos notícias alarmantes sobre falhas em softwares.

Ok, entendi, mas o que tem haver o VARCHAR com tudo isso?

Elementar jovem padawan, mas no afã de terminarmos nossos programas, podemos incorrer em erros e um simples if ou movimento de valor numa variável e cabum, surge inesperadamente uma anomalia, que com azar evolui para um abend.

Por isso conhecer o bom uso e saber usar as ferramentas ajuda muito, e saber como ela funciona internamente muito mais, por isso entenda o VARCHAR em COBOL e entendera seu uso em qualquer linguagem.

Sintaxe do VARCHAR

           03  VARIAVEL-PAI.

               49 TAMANHO-VARIAVEL     **PIC** S9(004) **USAGE** **BINARY**.

               49 TEXTO-VARIAVEL       **PIC**  X(050).

A variável varchar é constituída em um variável pai, sendo sua estrutura uma redefinição dos campos, dividido em duas partes, uma constando o tamanho em bytes compactados e outra com a variável propriamente dita.

Em tempo de execução, o programa verifica a largura (length) a partir da variável TAMANHO-VARIAVEL e percorre a memória X bytes de acordo com a leitura anterior.

Exemplo:

MOVE “TOMATE” TO TEXTO-VARIAVEL

MOVE 6 TO TAMANHO-VARIAVEL

Se não informamos o tamanho da variável, ocorrera uma anomalia, pois o programa não saberá onde termina os valores atribuídos na variável, podendo assumir que o tamanho é 50, movendo espaços em branco ou pior espaços nulos.

Grandes confusões já aconteceram devido aos valores nulos dentro de variáveis texto, principalmente ao chamarmos o SGBD e guardarmos esses valores em uma tabela e usa-los em outros processos.

Um programador deve sempre atentar que ao programar, nunca temos a certeza absoluta de onde nossos dados e outputs serão utilizados. Lembre-se o Sistema é dinâmico e podem incluir seu programa em outras rotinas, suas tabelas podem ser utilizadas em outros processamentos.

O DB2 no programa COBOL

Num artigo futuro entrarei em maiores detalhes, mas por agora, fixe na mente, os comandos SQL são estrutura embebed, que na altura da pré-compilação são trocadas por funções CALL e passagem de parâmetros para os programas externos do DB2.

EXEC SQL

Comando SQL

END-EXEC.

A Arte do Impossível

Quando trabalha com VARCHAR atente-se que outros programas podem ter manipulado este campo na base de dados, podendo conter sujeira e campos nulos. Por isso nunca confiei plenamente, valida se possível ou apenas use-a com fé torcendo para não acontecer um abend em sua gestão.

Todo bom programador deve validar as variáveis que recebe, caso vá utiliza-las para algum tratamento, pois ao processar nem sempre os valores que acreditamos estar numa variável estarão, ao longo da minha carreira muitas vezes me deparei com sujeiras e inconsistência em variáveis recebidas de outro processo.

Formulários WEB e seus risco

Lembramos que os dados hoje são independentes de plataforma, muitas vezes formulários WEB, recebem dados que são inseridos em tabelas em SGBDs, que podem ser extraídos em processos batch de extração gerando arquivos sequenciais e usados em processos batch de transformação e novamente passarem por processos de LOAD em outras tabelas de outros SGBD.

Se ocorrerem problemas em algum processo de validação dos dados, este problema ira se perpetuar em todos os processos, dando a maior dor de cabeça, para descobrir onde ocorreu o erro inicial, uma tarefa hercúlea digna de um Sherlock Holmes dos sistemas centrais.

Aplicativos prontos

Sempre que possível use os aplicativos do próprio SGBD ou de fornecedores confiáveis, não queira inventar a roda, lembra-se do teorema da seringa?

No ambiente mainframe existem diversos aplicativos prontos para trabalho, cito alguns QMF, Spufi e SORT, sendo os dois primeiros destinados a trabalharem em SGBD, principalmente o DB2 (base de dados SQL) e o ADABAS (base de dados mista NoSQL e SQL) e o SORT usando em classificação e transformação através de somatórios e concatenações.

Segredos do Mainframe

O que poucos DEVs iniciantes sabem é que a maior parte do processamento de dados é em Mainframe, órgãos governamentais, forças armadas e empresas do setor financeiro, necessitam de integridade, segurança, velocidade e grande poder de processamento.

Por isso atividades principais são deixadas a processos em Mainframe, sem contar que mesmo querendo sair os custos da migração e o risco de negocio são assustadores, vejam o caso do TSB e o seguro desemprego em alguns estados dos EUA, que desesperadamente não tinham profissionais para codificarem algumas alterações nos sistemas centrais.

Divagações sobre o VARCHAR

O VARCHAR foi um clandestino que entrou no COBOL graças ao DB2 e seu campo VARCHAR no SQL, obrigando o lançamento de uma nova compilação do COBOL e gerando uma carga enorme de trabalho para recompilar velhos programas, que usavam DB2 para adapta-los a nova versão.

Acabei entrando em outros temas e deixando o pobre padawan confuso, mas é assim mesmo, em informática tudo é interligado como numa teia de aranha, puxamos um fio e acabamos indo parar em outra ponta.

Conclusão

O objetivo primário deste artigo era apresentar o varchar no COBOL e os pequenos perengues devido a declaração do tamanho da variavel