{arquivos sequenciais}

5.1 DE VOLTA AOS ANOS 60

Este site (DMU) é dedicado a pessoas/empresas que não têm acesso total a um equipamento servidor donde,geralmente, também não têm acesso a um programa servidor de bancos de dados que segue o padrão SQL.

Você vai ter então que usar arquivos sequenciais, conhecidos como arquivos ASCII ou TXT (que no Windows criamos com o NotePad). E vai ter que usar CGI, que tem algumas limitações.

Se você tem um provedor de "web hosting" que disponibiliza um banco de dados (a HIWAY tem o MiniSQL) então a coisa melhora. Temos um curso sobre o uso de MiniSQL aqui na DMU. Não deixe de fazê-lo!

Mas vamos considerar neste curso que você não tem acesso a nenhum banco de dados.

Trabalhando com arquivos sequenciais, se você for um coroa como eu (30 anos - de informática!) vai relembrar aqueles bons tempos do anos 60 em que curtíamos os Beatles,tomávamos Cuba Libre e não nos preocupávamos com a AIDS... Agora, um Stradivarius é velho, mas ainda toca melhor que muito violino eletrônico...

Se você não é desse tempo, vamos dar uma dicas iniciais de como trabalhar com esses arquivos. Depois (nas próximas aulas) vamos estudar um pouco mais de PERL e aprender a fazer as coisas direitinho.

5.2 UM ARQUIVO SEM CAMPOS

Um arquivo TXT (ou ASCII) não tem campos definidos, nem colunas, nem chaves. Tem apenas linhas, uma embaixo da outra.

Se você quer criar uma arquivo que contenha:

- CÓDIGO DE PRODUTO (4 dígitos)
- NOME DO PRODUTO (20 dígitos)
- PREÇO (6 dígitos, dois decimais)

você não vai definir nada.Um arquivo TXT só trabalha com caracteres (não com números) e se você usa uma linguagem como Cobol,você vai é gravar linhas assim:

```
0001BONECAbSHARONbbbbbbbbb2000
0002BOLAbANIMALbbbbbbbbbbb1000
0003TREMbDASbONZEbbbbbbbbb3000
```

Veja que o fato do preço ter duas decimais fica meio confuso (na realidade os preços são: 20,00 ou 10,00 ou 30,00.

O que colocamos como "b" são "brancos" (teclados com a tecla de espacejamento).

Repare que no primeiro registro temos 1 branco e mais 9 brancos porque a palavra "BONECA SHARON" tem 13 dígitos (com 1 que é branco) faltando 7 para o tamanho do campo que é 20. E depois o preço é "2000" (não numérico) e tem 4 algarismos faltando 2 algarismos/brancos para completar o campo.

Em PERL a coisa fica um pouquinho mais fácil (?) porque existem umas funções especiais (que vamos ver) para tratar strings. O arquivo ficaria assim:

```
0001:BONECA SHARON:20.00
 0002:BOLA ANIMAL:10.00
0003:TREM DAS ONZE:30.00
```

repare que o separador de decimal aparece com um ponto, no estilo americano. Repare também que entre cada campo colocamos um separador (:).

O que o PERL possibilita é que você leia cada uma das linhas de um arquivo assim. Então, você tem que jogar a linha em um array e, usando determinadas funções separar cada campo jogando-os em variáveis.

Depois de operar com esses campos, se você for regravar atualizando o registro tem que primeiro remontar a linha (em caracteres).

Depois surge o maior problema: Não existe UPDATE (do SQL) em arquivos ASCII. Uma maneira possível (existem outras) de atualizar um registro é ter jogado o arquivo TODO num array, atualizar a dita cuja linha (existem funções para isso) e REGRAVAR TODOS OS REGISTROS, como se fosse um novo arquivo (com o mesmo nome do antigo).

Daí você vê de onde vem os problemas de programar quando se usa arquivos que não são bancos de dados. Depois,se você usar esta técnica, eles não podem ser muito grandes porque o array com TODOS os dados vai para a memória.(Imagina o cadastro de contas correntes do BRADESCO na memória de um micro! - é mais ou menos como a piada do elefante que queria seduzir a formiguinha).

Se você quer acessar apenas um determinado registro, também não existe SELECT. Você tem que ler CADA linha, passar para campos e testar se um determinado campo bate com o string que você procura. Para facilitar essa procura costuma-se usar a chave como primeiro campo e testar apenas os primeiros dígitos do registro (existe uma função para isso).

Quem trabalhou com uma linguagem chamada APL na década de 70 vai matar saudade. Você pode não acreditar mais nós (na IBM, em setenta e poucos) desenvolvemos um sistema de Home Banking (teleprocessamento) para o Bradesco trabalhando dessa forma. O saudoso "Seu" Amador Aguiar (presidente do Bradesco - vivo na época) até foi na instalação do sistema (era um sistema importante). E a coisa funcionava...

Estou contando isso para você não desanimar completamente. Dá para fazer boas aplicações com PERL ou Java e arquivos sequenciais ASCII. Talvez não seja a maneira mais fácil e adequada. Mas é a mais barata, sem dúvida.

O melhor mesmo é usar os arquivos ASCII/TXT apenas como repositórios de entradas ou arquivos para consultas, tendo um banco de dados Access, dBASE,Oracle, DB2 etc. como retaguarda.

Alterações,cancelamentos etc. serão gravados em arquivos TXT auxiliares, on line, e a atualização dos dados no banco de dados e emissão de novo arquivo para consulta será feita stand alone.

É claro que você tem que carregar seu arquivo ASCII para seu provedor de espaço como ASCII e não como binário.

Arquivos de entrada de dados (caso de pedidos on line, por exemplo) podem ser apagados depois de passados para o banco de dados. Você pode fazer uma conexão entre esta entrada de dados pela Internet e seu sistema de pedidos já existente, por exemplo.

O Usuário deve ser alertado que a atualização de dados será feita posteriormente. Senão pode alterar um pedido e depois verificar na hora que ficou tudo na mesma e reclamar por email etc.

O banco de dados de retaguarda pode ser operado com programas em Visual Basic, Delphi, C++ ou Java 1.1 ou 1.2 com JDBC até usando compiladores que começam a aparecer no mercado.

Mas vamos ver então como gravamos e lemos esses arquivos ASCII.