1001487: Ciclo 1 – Interpretador de comandos e estruturação dos metadados para a base

Jander Moreira*

Entrevista: 20 de agosto de 2019†

1 Apresentação

O primeiro ciclo do projeto prevê a implementação de um interpretador simples de comandos e a definição de como os dados internos à base serão mantidos.

O interpretador de comandos define a linguagem básica simplificada que deve ser reconhecida.

Os dados internos são os metadados que definem como o sistema em desenvolvimento guarda dados sobre si mesmo.

2 Descrição

Esta seção descreve as duas partes envolvidas.

2.1 Interpretador de comandos

O interpretador deve ser capaz de reconhecer comandos simples em uma linha única, não diferenciando maiúsculas de minúsculas e ignorando espaços consecutivos. Note que para os valores dos dados, maiúsculas e minúsculas devem ser diferenciadas.

Exemplos de comandos são apresentados no final desta seção.

O termo *tabela* é usado para especificar um arquivo de dados e os metadados e estruturas de indexação associados a ele. Os comandos de manipulação de tabelas incluem:

Comando	Descrição
`CT` *tabela campos*	Cria um arquivo vazio associado ao nome *tabela* e o registra como ativo na base, sendo *campos* uma lista de especificações dos campos que formarão o arquivo.
`RT` *tabela*	Apaga o arquivo relativo a *tabela* e remove seus metadados da base, bem como estruturas associadas.
`AT` *tabela*	Apresenta um resumo dos metadados da *tabela* indicada: arquivos, campos e índices existentes.
`LT`	Lista o nome de todas as tabelas existentes na base.

A especificação de campos é uma lista separada por pontos-e-vírgulas, sem espaços intermediários, com cada campo especificado pelo par *tipo*`:`*nome*.

Os tipos que devem ser considerados incluem:

Tipo	Descrição
`INT`	Um valor inteiro com sinal.
`FLT`	Um valor real.
`STR`	Uma cadeia de caracteres qualquer.
`BIN`	Uma sequência de bytes (formato binário genérico).

A manipulação de registros nas tabelas permitem inserção, remoção e busca. Os comandos sempre especificam uma tabela a ser usada:

Comando	Descrição
`IR` *tabela registro*	Insere o *registro* no arquivo de *tabela*, usando a política de inserção adequada.
`BR N` *tabela busca*	Busca em *tabela* todos os registros que satisfaçam o critério *busca*.
`BR U` *tabela busca*	Busca em *tabela* pelo primeiro registro que satisfaça o critério *busca*.
`AR` *tabela*	Apresenta na tela os valores dos registros retornados pela última busca.
`RR` *tabela*	Remove, segundo a política de remoção da tabela, todos os registros da última busca realizada.

A operação de inserção deve acrescentar um novo registro à tabela, o qual deve ser especificado por uma

*Jander Moreira – Universidade Federal de São Carlos – Departamento de Computação – *Rodovia Washington Luis, km 235 – 13565-905 - São Carlos/SP – Brasil* – jander@dc.ufscar.br

†Dia dos Maçons.

sequência de valores separados por pontos-e-vírgulas, na mesma ordem de campos usada para na criação. Os valores `INT`, `FLT` e `STR` são indicados diretamente; valores `BIN` especificam o nome de um arquivo que contém os bytes.

Na busca, o argumento de busca é sempre simples e responde apenas pela igualdade. O termo *busca* é especificado, assim, pelo nome do campo seguido pelo valor procurado, no formato *campo*`:`*valor*. Campos `BIN` não permitem busca, embora a implementação possa opcionalmente ser incluída no desenvolvimento.

As instruções sobre a manutenção de estruturas de indexação envolvem tanto árvores de múltiplos caminhos quanto uso de *hashing*. Os comandos são:

Comando	Descrição
`CI A` *tabela chave*	Cria um índice estruturado como árvore de múltiplos caminhos para a *tabela*, usando *chave* como chave de busca, atualizando os metadados.
`CI H` *tabela chave*	Cria um índice usando *hashing* para a *tabela*, usando *chave* como chave de busca, atualizando os metadados.
`RI` *tabela chave*	Remove o índice relativo à *chave*, atualizando os metadados e apagando as estruturas envolvidas.
`GI` *tabela chave*	Gera novamente o índice de *tabela* referente à *chave*, partindo do zero.

Os identificadores usados para *tabela*, *campo*, *chave* etc. consistem em um nome simples formado apenas por letras. Números ou sublinhas podem, opcionalmente, ser aceitos, mas não serão testados nas avaliações.

Um comando de encerramento deve ser adicionado, embora se espere que seu uso não seja obrigatório nas sessões interativas ou no arquivo. O término de uma sessão de comandos (terminal ou arquivo) pode ser indicada tanto pelo comando `EB` quanto pelo fim dos dados (i.e., encerramento do dados no terminal ou fim do arquivo).

Comando	Descrição
`EB`	Encerra a interpretação e termina a execução do programa.

Os espaços iniciais (antes do comando) devem ser ignorados, assim como espaços repetidos entre as partes do comando. Porém, espaços no final nos valores de campo devem ser preservados.

Linhas em branco (comprimento zero ou somente com espaços) devem ser ignoradas. Linhas com erros de sintaxe devem encerrar a interpretação, apresentando uma mensagem de erro e terminando a execução do programa.

É importante salientar que, nesta parte do projeto, apenas o interpretador de comandos deve ser implementado, e não as operações descritas por ele.

Exemplos

Seguem alguns exemplos comentados de sequências de comandos que podem ser usados para criar arquivos.

Esta sequência cria uma nova tabela chamada `CLIENTES` a adiciona a ela cinco registros.

```
CT CLIENTES INT:CODIGO;STR:NOME;BIN:CERTIF
IR CLIENTES 10;JOSE DA SILVA;jose_cert.crt
IR CLIENTES 20;JOSE DE SOUZA;souza.crt
IR CLIENTES 30;JOAO PASCOAL;jp.crt
IR CLIENTES 40;LUIS BERTOLO;lb_novo.crt
IR CLIENTES 50;JOAO PASCOAL;jp2.crt
EB
```

A criação de um índice (*hash*) para `CLIENTES` usando o campo `CODIGO` é feita pelo comando seguinte.

```
CI H CLIENTES CODIGO
EB
```

Os comandos seguintes realizam a busca e remoção de registros.

```
BR U CLIENTES NOME:JOAO PASCOAL
RR CLIENTES
EB
```

2.2 Metadados

Além do interpretador de comandos, nesta primeira etapa de desenvolvimento deve ser projetado o conjunto de metadados da base de dados. Os metadados devem manter as informações sobre quais tabelas existem e seus campos, quais arquivos a tabela usa e quaisquer outras informações que permitam o gerenciamento dos dados.

Espera-se que, neste ciclo, esteja claro para o grupo quais são os dados que serão mantidos, considerando-se que a base pode ser formadas por várias tabelas, cada uma com sua estrutura de campos específica.

Neste momento, a inclusão dos índices nos metadados não é requerida, pois será retomada no ciclo 4.

3 Metas

A avaliação do projeto é sempre feita com base em metas coletivas e individuais.

3.1 Metas coletivas

Segue a lista de metas coletivas.

#	Descrição
1	Programa que responde ao modo interativo e por arquivo.
2	Resposta adequada ao término dos dados e comando `EB`.
3	Interpretação dos comandos `CT` e `RT`.
4	Interpretação do comando `IR`
5	Interpretação dos comandos `BR N` e `BR U`.
6	Interpretação do comando `RR`.
7	Interpretação dos comandos `CI A` e `CI H`.
8	Interpretação dos comandos `RI` e `GI`.
9	Interpretação dos comandos `AT`, `LT` e `AR`.
10	Resposta adequada a linhas em branco e com erros de sintaxe.
11	Lista dos arquivos para os metadados.
12	Estrutura e lógica dos arquivos de metadados.

Valores limiares para o número de metas a serem atingidas para obtenção da nota coletiva do ciclo:

Limiar	Quantidade de metas
${}^{\text{c}}n_1$	9
${}^{\text{c}}n_2$	6

3.2 Metas individuais

Todas as metas individuais deste ciclo serão avaliadas na entrevista. Segue a lista.

#	Descrição
1	Domínio sobre o código que estrutura o interpretador.
2	Conhecimento sobre a possibilidade de aumentar a gama de comandos que o interpretador reconhece e suas implicações no código.
3	Clareza no código sobre onde serão tratadas cada uma das operações reconhecidas pelo interpretador.
4	Domínio sobre o uso dos arquivos de metadados propostos: o que mantém e porquê.

Valores limiares para o número de metas a serem atingidas para obtenção da nota individual do ciclo:

Limiar	Quantidade de metas
${}^{\text{i}}n_1$	4
${}^{\text{i}}n_2$	2

4 Fechamento

Em caso de dúvidas, pergunte ao professor no fórum de dúvidas do AVA. Lembre-se que questionamentos e soluções de nível mais alto podem ser compartilhadas com os colegas e que é vedada qualquer apresentação do código em desenvolvimento.

Note que as metas coletivas não cumpridas podem ser feitas para um ciclo posterior e serão contabilizadas normalmente. As metas individuais, porém, serão avaliadas somente neste momento da entrevista.

◇

Revisão: 5 de agosto de 2019
Dia Nacional da Saúde.