

Instituto Federal Sudeste de Minas Gerais – Campus Barbacena

Curso de Técnico em Informática

Disciplina: Lógica de Programação II

Prof.: Wender Magno Cota

Primeiro trabalho Prático

Descrição do Problema

Em Alvinópolis City, o senhor Louco das Pedras, é proprietário da única loja de produtos farmacológicos. Atualmente, como ele é um lunático, faz o controle dos dados de sua loja usando anotações em papel, ou seja, a antiga e conhecida cadernetinha. Um belo dia, o alucinado, louco e munheca de samambaia resolveu conhecer a praia de Iraparaug. Para chegar lá, ele resolveu usar o meio rodoviário, utilizando um método adotado por vários estudantes pra visitar parentes morando em outras cidades: carona. Vale ressaltar que Alvinópolis City possui um moderno aeroporto, com vôos regulares para todos os aeroportos do Brasil.

Chegando na cidade praiana, o sortudo conheceu uma mocinha com o codinome **Virgininha Só Que Só**. Como ela é uma pessoa extremamente bem informada, principalmente a respeito de dinheiro e como evitar utilizá-lo, após uma rápida conversa (afinal, tempo é dinheiro), Louco das Pedras finalmente decidiu acatar os conselhos recebidos: informatizar sua loja.

Após seu retorno à cidade natal, comprou um único computador e pelas informações passadas pela amiga, que no segundo período de Tecnologia em Sistemas para Internet - Campus Barbacena há excelentes programadores, resolveu contratar os serviços de tais alunos para desenvolver uma software para gerenciamento da sua loja.

Tabelas

Cliente

| Nome do Campo | Tipo | Tamanho |
|---------------|---------------|---------|
| id | unsigned long | |
| nome | char | 100 |
| cpf | char | 12 |
| email | char | 50 |
| telefone | char | 15 |

Chave Primária: id

Obs.: O campo cpf é único.

NotaFiscal

| Nome do Campo | Tipo | Tamanho |
|---------------|---------------|---------|
| id | unsigned long | |
| idCliente | unsigned long | |

| | | |
|------------|---------------|----|
| idVendedor | unsigned long | |
| dataCompra | char | 11 |
| ValorTotal | float | |

Chave Primária: id

Chaves Estrangeiras:

Campos Tabela de Origem

idCliente Cliente

idVendedor Vendedor

Obs.: O campo numeroNota é único.

ItemNotaFiscal

| Nome do Campo | Tipo | Tamanho |
|---------------|---------------|---------|
| id | unsigned long | |
| idNotaFiscal | unsigned long | |
| idProduto | unsigned long | |
| valorVenda | float | |
| quantidade | unsigened int | |

Chave Primária: id

Chaves Estrangeiras:

Campos Tabela de Origem

idNota notaFiscal

idProduto produto

Produto

| Nome do Campo | Tipo | Tamanho |
|-------------------|---------------|---------|
| id | unsigned long | |
| nome | char | 100 |
| quantidadeEstoque | unsigned int | |
| precoUnitario | float | |

Chave Primária: id

Vendedor

| Nome do Campo | Tipo | Tamenho |
|---------------|---------------|---------|
| id | unsigned long | |
| Nome | char | 100 |
| cpf | char | 12 |
| email | char | 50 |

| | | |
|----------|------|----|
| telefone | char | 15 |
| password | char | 20 |

Chave Primária: id

Obs.: Os campos CPF e email são únicos.

Fornecedor

| Nome do Campo | Tipo | Tamanho |
|---------------|---------------|---------|
| id | unsigned long | |
| CNPJ | char | 15 |
| nome | char | 100 |
| email | char | 50 |
| telefone | char | 15 |

Chave Primária: id

Obs.:

a) Validar o CNPJ;

b) Os campos nome e CNPJ são únicos.

NotaCompra

| Nome do Campo | Tipo | Tamanho |
|---------------|---------------|---------|
| id | unsigned long | |
| idFornecedor | unsigned long | |
| dataCompra | char | 11 |
| valorTotal | float | |

Chave Primária: id

Chave Estrangeira:

| Campos | Tabela de Origem |
|--------|------------------|
| idNota | idFornecedor |

Obs.:

a) O campo numeroNota é único.

b) Validar o campo dataCompra.

ItemNotaCompra

| Nome do Campo | Tipo | Tamanho |
|---------------|---------------|---------|
| id | unsigned long | |
| idProduto | unsigned long | |
| idNotaCompra | unsigned long | |

| | | |
|---------------|--------------|--|
| quantidade | unsigned int | |
| valorUnitario | float | |

Chave Primária: id

Chave Estrangeira:

| Campos | Tabela de Origem |
|------------------------|------------------|
| idProduto | produto |
| idNotaCompranotaCompra | |

Funcionalidades

1) Fazer a manutenção das tabelas Cliente (cadastro, alteração, consulta e Listagem). A consulta deve ser feita pelo id, CPF e o prefixo do nome.

2) Fazer a manutenção da tabela Vendedor (cadastro, alteração, consulta e Listagem). A consulta deve ser feita pelo id, CPF e o prefixo do nome.

3) Fazer a manutenção da tabela Fornecedor (cadastro, alteração, consulta e Listagem). A consulta deve ser feita pelo id, CNPJ e o prefixo do nome.

4) Cadastro de um produto. Atribuir o valor zero ao campo quantidadeEstoque.

5) Efetuar uma venda

Inicialmente fazer a autenticação do Vendedor(cpf e senha) e em seguida realizar a validação do cliente(campo id). Após a realização das operações anteriores, criar um Carrinho de Vendas que deve ser um vetor dinâmico, cujo tipo base, será uma struct formada pelos campos idProduto, quantidadeVendida e valorVenda. Após a confirmação da venda, deve-se atualizar a tabela de produtos e criar uma nota fiscal referente a venda e um registro de ItemNotaFiscal para cada produto vendido.

Obs.:

- a) Para o campo dataCompra da tabela NotaFiscal apresentar a data atual do Sistema Operacional, mas possibilitado ao usuário alterar tal campo com outra data.
- b) O Vendedor após realizar a autenticação pode efetuar várias vendas, bastando selecionar um novo Cliente.
- c) Durante a venda o Vendedor pode cancelar um produto do Carrinho, bem como alterar a quantidade do mesmo, respeitando a quantidade existente em estoque.
- d) O carrinho não pode possuir dois produtos iguais.
- e) O campo valorVenda do carrinho de Compras deve receber o valor do campo precoUnitario(obtido da tabela Produto).

6) Efetuar uma compra

Criar um carrinho de compras. Observe que cada compra deve ser efetuada de um fornecedor. Tal carrinho deve ser um vetor dinâmico, cujo tipo base, será uma struct formada pelos campos idProduto, quantidade e valorUnitario. Após confirmar a efetivação da compra, deve-se atualizar a tabela de produtos, ou seja, a quantidade em estoque e criar uma nota de compra referente a mesma e um registro de itemNotaCompra para cada produto comprado.

Obs.:

- a) Para o campo dataCompra da tabela NotaCompra, apresentar, inicialmente, a data atual do Sistema Operacional, mas possibilitado ao usuário alterar tal campo com outra data.
- b) Permitir a exclusão de um produto do Carrinho de Compra, bem como alteração da quantidade, antes da confirmação da operação de compras(Fechamento do Carrinho de Compras).
- c) O carrinho não pode possuir dois produtos iguais.

7) Atualizar preço

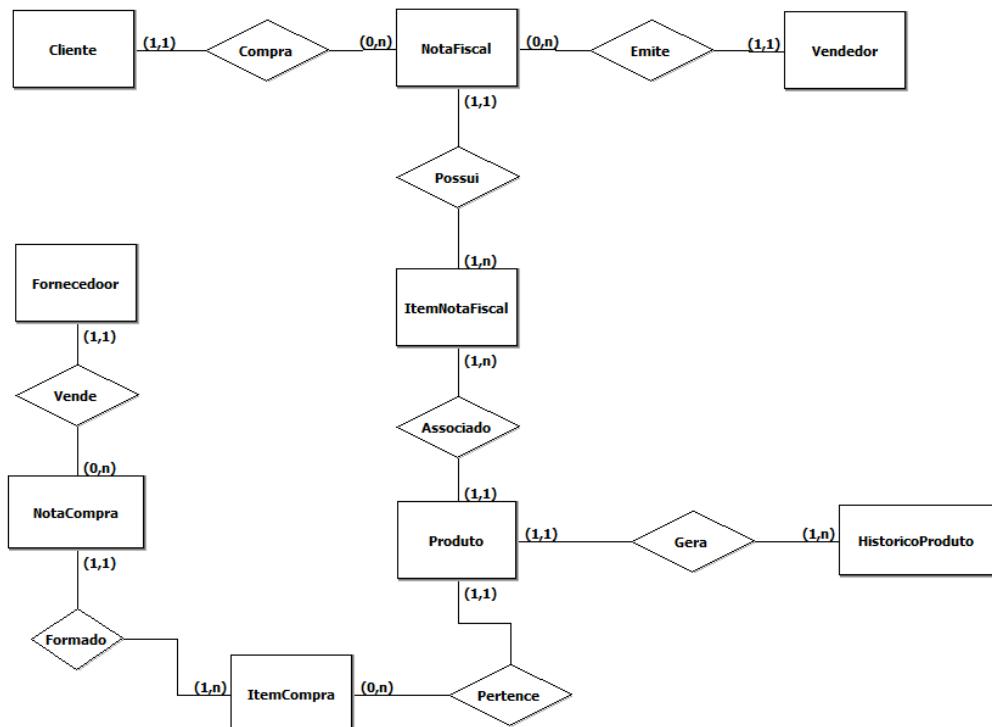
Permitir que se altere o preço de um único Produto ou de todos, em um determinado %. Criar um registro no arquivo de HistoricoPreco para cada produto que teve seu preço alterado. Tal arquivo deve possuir os campos: idProduto, dataAlteracao, Valor). A chave primária é formada pelos campos(idProduto+dataAlteracao).

7) Relatórios

- a) Dados de uma nota Fiscal e os itens associados a ela.
- b) Dados de uma compra e os itens associados a ela.
- c) Compras efetuadas por um cliente em uma data específica, bem como em um intervalo de datas. Mostrar os dados da nota junto com os produtos pertencentes a ela.
- d) Vendas efetuadas por um vendedor em um dia específico.
- e) Vendas realizadas por um vendedor em um determinado mês.
- f) O vendedor(es) com maior valor de venda em um determinado mês.
- g) O vendedor(es) com maior valor de venda em um intervalo de datas.
- h) Histórico de valor de venda de um determinado produto. Estes dados são obtidos no arquivo Histórico Produto.

Nota: Permitir que o usuário escolha se deseja que o relatório seja apresentado na tela do dispositivo ou em arquivo.

DER – Sistema da Loja de Produtos Farmacológicos



Faltou um atributo identificador no relacionamento **Gera (dataAlteração)**.

Observações Finais:

1. O código deve ser desenvolvido em linguagem C.
2. Utilize apenas recursos compatíveis com C padrão (ANSI C). Evite funções específicas de sistemas operacionais (Windows, Linux, etc.).
3. O programa deve compilar sem nenhum warning. Para verificar isso no Linux (opcional), utilize:
 gcc -Werror arquivos.c
4. Os campos **id** devem utilizar auto-incremento, gerando um número único sempre que um novo registro for inserido. Cada sequência de **id** deve começar em 1.
5. Não é necessário se preocupar com concorrência no acesso à base de dados, pois o sistema não será multiusuário nem executado em rede; toda a aplicação e os arquivos residirão em um único computador.
6. Trabalhos com cópias totais ou parciais terão nota zero.
7. Utilizem comentários na medida certa.
8. Os nomes dos campos, tipos e tamanhos devem seguir exatamente o que foi especificado.
9. Os nomes dos arquivos devem ser os mesmos definidos no enunciado, usando a extensão **.dat**.
10. Todo campo textual deve ser convertido para letras maiúsculas sem acentuação.
11. Iniciem o desenvolvimento o quanto antes - a data de entrega chega mais rápido do que parece.
12. Data de entrega: 08/01/2026.