ETEC da Zona Leste

Pedro Hiroshi Godeny Takamura

**Trabalho de Banco de dados**

Prof. Marcelo Collado

3o ano do curso de Desenvolvimento de Sistemas - Novotec

**Sumário**

[ **1 - Introdução** 3](#_Toc121617157)

[ **2 - Desenvolvimento** 4](#_Toc121617158)

[ **2.1 - MER** 4](#_Toc121617159)

[ **2.2 – Código** 5](#_Toc121617160)

[ **2.2.1 – Criação do banco de dados** 5](#_Toc121617161)

[ **2.2.2 – Inserção de informações** 7](#_Toc121617162)

[ **2.2.3 – Seleção de informações** 7](#_Toc121617163)

[ **2.3 – Dicionário de dados** 9](#_Toc121617164)

[ **3 - Observações** 11](#_Toc121617165)

**Índice de imagens**

Figura 1: Primeira versão do MER 4

Figura 2: Versão correta do MER 4

Figura 3: Código de criação do banco de dados 6

Figura 4: Inserção de dados no banco 7

Figura 5: Select com inner join 7

Figura 6: Tabela resultante do select com inner join 8

Figura 7: Select simples 8

Figura 8: Tabela resultante do select simples 8

Figura 9: Select com subquery 8

Figura 10: Tabela resultante do select com subquery 8

* **1 - Introdução**

Esse trabalho consiste em criar um banco de dados dês de seu início até seu término de forma completamente formal, envolvendo também o modelo de entidade e relação para sua realização.

O código fonte desse banco de dados poderá ser encontrado no meu GitHub pessoal: <https://github.com/HiroshiGTakamura/final-database>.

As instruções para a construção do banco de dados foram as seguintes:

***PROJETO FINAL***

**ESTUDO DE CASO :**O Departamento de Vendas da Indústria Beleza Ltda, após estudos de mercado, verificou que para atingir seus objetivos seria necessário adquirir frota de veículos próprios para motorizar seus vendedores. O mercado consumidor foi dividido em regiões de venda; foram estabelecidos percursos de entrega abrangendo pontos estratégicos dessas regiões e vendedores foram designados para cobrir estes percursos. Um sistema deve ser construído para administração da nova sistemática de vendas adotada pela empresa. Após entrevistas com o gerente da área, foram obtidas as seguintes informações:

* + - - Cada região é identificada por um código;
    - - Uma região é composta de vários pontos estratégicos;
    - - As regiões não têm pontos estratégicos em comum;
    - - O vendedor tem a responsabilidade de cobrir uma região;
    - - Uma região pode ser coberta por vários vendedores;
    - - A cada dia, um veículo fica sob a responsabilidade de um vendedor;
    - - Um vendedor pode vender quaisquer itens ativos da tabela de produtos;
    - - O vendedor é responsável pela identificação de cada cliente consumidor na nota fiscal;
    - - A nota fiscal contendo identificação do vendedor, itens e quantidades vendidas é exigida para comprovação da venda.
* **2 - Desenvolvimento**
* **2.1 - MER**

Com apenas as informações que foram dadas, já é possível iniciar o desenvolvimento de um Modelo de Entidade e Relação (MER) para esse banco de dados. A primeira versão do MER desse banco de dados não corresponde ao banco de dados final.

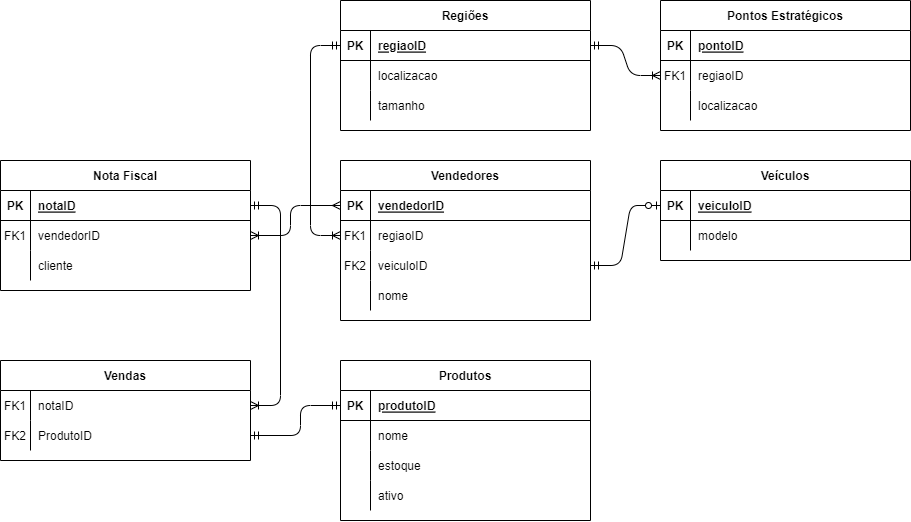
Figura 1: Primeira versão do MER

Diagrama

Descrição gerada automaticamente

A versão corrigida do MER do banco de dados é assim:

Figura 2: Versão correta do MER

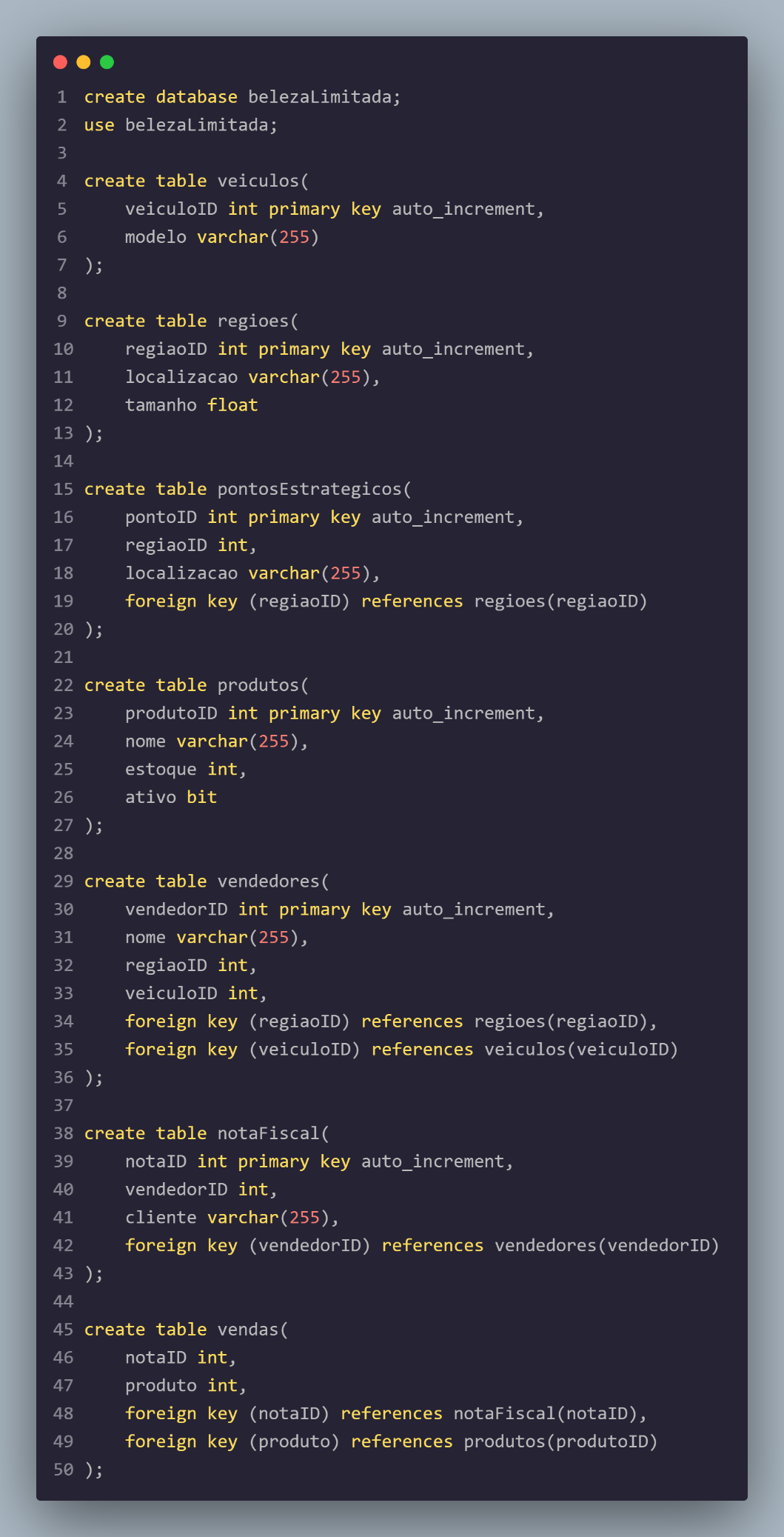


Sobretudo, a maior mudança notável é a adição de uma tabela a mais. Essa tabela serve para melhor descrever os itens vendidos. Assim, cada item vendido seria uma entrada na tabela de vendas, indicado pelo seu ID da tabela de produtos, e relacionando-se com a venda específica que foi feita por meio do ID da nota fiscal. No MER anterior seria impossível de vender mais que um item por nota fiscal.

* **2.2 – Código**
  + **2.2.1 – Criação do banco de dados**

O código a seguir foi o código usado para criar o banco de dados e suas tabelas, e segue a versão final do MER do projeto.

Figura 3: Código de criação do banco de dados



Os espaços de 255 para as variáveis de tipo varchar foram escolhidos por simplicidade e falta de necessidade de mudá-lo durante o funcionamento da base de dados, caso uma informação maior do que o valor escolhido fosse recebida.

* + **2.2.2 – Inserção de informações**

O código a seguir foi o código usado para inserir informações no banco de dados. Os nomes escolhidos não necessariamente se referem a nada, sabendo que a tarefa de criar essa base de dados é um exemplo. Em seus lugares ficariam as próprias informações de nomes de produtos, modelos de carros e nomes reais.

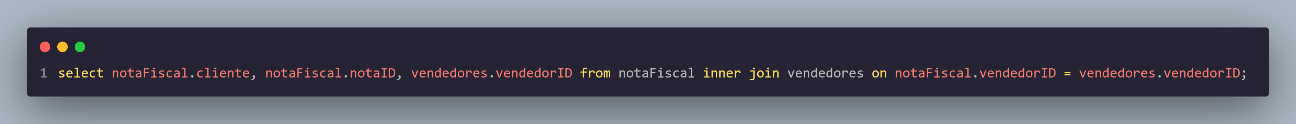
Figura 4: Inserção de dados no banco



* + **2.2.3 – Seleção de informações**

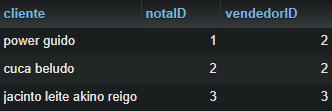
Os códigos a seguir foram utilizados para realizar a seleção das informações agora inseridas no banco de dados.

Figura 5: Select com inner join



Esse código faz um select com um inner join, e ele retorna a seguinte tabela:

Figura 6: Tabela resultante do select com inner join



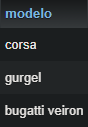
O seguinte código faz a seleção de dados com um select simples:

Figura 7: Select simples



Ele retorna a seguinte tabela:

Figura 8: Tabela resultante do select simples



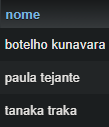
Por fim, o seguinte código faz a seleção de dados com subquery, onde, nesse caso, ele só retornará o primeiro select se o segundo retornar algo.

Figura 9: Select com subquery



Ele retorna esta tabela:

Figura 10: Tabela resultante do select com subquery



* **2.3 – Dicionário de dados**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tabela | veiculos | | | |
| Descrição | Armazena os dados dos veículos | | | |
| Observações |  | | | |
| Campos | | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de Domínio |
| veiculoID | ID do veículo | INT |  | PK |
| modelo | modelo do veículo | VARCHAR | 255 |  |
|  |  |  |  |  |
| Tabela | regioes | | | |
| Descrição | Armazena os dados das regiões | | | |
| Observações |  | | | |
| Campos | | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de Domínio |
| regiaoID | ID da região | INT |  | PK |
| localizacao | localização da região | VARCHAR | 255 |  |
| tamanho | tamanho da área demarcada no mapa | float |  |  |
|  |  |  |  |  |
| Tabela | pontosEstrategicos | | | |
| Descrição | Armazena os dados dos pontos estratégicos | | | |
| Observações | Possui uma FK da tabela regiao | | | |
| Campos | | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de Domínio |
| pontoID | ID do ponto | INT |  | PK |
| localizacao | localização do ponto | VARCHAR | 255 |  |
| regiaoID | FK da região | INT |  | FK |
|  |  |  |  |  |
| Tabela | produtos | | | |
| Descrição | Armazena a informação de cada produto diferente | | | |
| Observações | A variável bit é incomum, mas é comumente usada para informações booleanas | | | |
| Campos | | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de Domínio |
| produtoID | ID do produto | INT |  | PK |
| nome | nome do produto | VARCHAR | 255 |  |
| estoque | quantidade do produto disponível | INT |  |  |
| ativo | se o produto pode ser vendido | BIT |  |  |
|  |  |  |  |  |
| Tabela | vendedores | | | |
| Descrição | Armazena as informações de cada vendedor | | | |
| Observações | possui FKs das tabelas regiao e veiculo | | | |
| Campos | | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de Domínio |
| vendedorID | ID do vendedor | INT |  | PK |
| nome | nome do vendedor | VARCHAR | 255 |  |
| regiaoID | FK da região | INT |  | FK |
| veiculoID | FK do veículo | INT |  | FK |
|  |  |  |  |  |
| Tabela | notaFiscal | | | |
| Descrição | Armazena cada nota fiscal emitida | | | |
| Observações | possui um FK da tabela vendedores | | | |
| Campos | | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de Domínio |
| notaID | ID da nota fiscal | INT |  | PK |
| cliente | nome do cliente atendido | VARCHAR | 255 |  |
| vendedorID | FK do vendedor | INT |  | FK |
|  |  |  |  |  |
| Tabela | vendas | | | |
| Descrição | Armazena os itens vendidos e os conecta com a nota fiscal relacionada à sua venda | | | |
| Observações | possui FKs das tabelas notaFiscal e produtos | | | |
| Campos | | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de Domínio |
| notaID | nota fiscal que o vendeu o produto | INT |  | FK |
| produtoID | produto vendido | INT |  | FK |

* **3 – Observações finais**

Esse projeto foi inicialmente feito no SQL Manager da Microsoft, e terminado no phpMyAdmin. O código funciona por completo no phpMyAdmin, e precisa ser adaptado para ser usado no SQL Manager da Microsoft ou qualquer outra IDE que utilize de uma forma diferente de sintaxe. Por exemplo, a linha de código “use belezaLimitada” não é necessária para o SQL Manager, e toda instância de “auto\_increment” é transformada para “identity”.

Também gostaria de anotar que os comandos do SQL não precisam ser escritos em caixa alta. Os códigos que escrevi em caixa baixa funcionaram perfeitamente e salvaram muito tempo e muitos usos das minhas teclas [Shift] e [Caps Lock].

Por fim, essa tarefa foi interessante de se fazer. Passar pelo processo inteiro e realizar a escrita do documento foi divertido, e encerrou o curso com chave de ouro.