

**GOVERNO FEDERAL**

**MINISTÉRIO DA EDUCAÇÃO**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA**

**CAMPUS CAJAZEIRAS**

**CURSO ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**REDE DE FARMÁCIAS**

**DANIEL ALVES DE LIMA**

**CAJAZEIRAS-PB**

**2019**

**DANIEL ALVES DE LIMA**

**REDE DE FARMÁCIAS**

Trabalho desenvolvido como requisito parcial para obtenção de aprovação na disciplina de Banco de Dados I, no curso de Análise e Desenvolvimento de Sistemas, no Instituto Federal de Educação, Ciência e Tecnologia, Campus Cajazeiras.

Prof. Dr. Fábio Gomes de Andrade

**CAJAZEIRAS-PB**

**2019**

Sumário

[1 Introdução 4](#_Toc4843499)

[2 Modelo Conceitual 5](#_Toc4843500)

[2.1 Levantamento dos requisitos 5](#_Toc4843501)

[2.2 Diagrama Entidade – Relacionamento 8](#_Toc4843502)

[2.3 Dicionário conceitual de dados 9](#_Toc4843503)

[3 Modelo Lógico 14](#_Toc4843504)

[3.1 Mapeamento Entidade Relacionamento 14](#_Toc4843505)

[3.2 Dicionário Lógico de Dados 15](#_Toc4843506)

[4 Modelo Físico 21](#_Toc4843507)

[4.1 Scripts SQL 21](#_Toc4843508)

[4.1.1 Criando Relações 21](#_Toc4843509)

[4.1.2 Criando Índices 26](#_Toc4843510)

[4.1.3 Visões 27](#_Toc4843511)

[4.1.4 Procedimentos Armazenados 28](#_Toc4843512)

[4.1.5 Gatilhos 29](#_Toc4843513)

[4.1.6 Povoando as relações 31](#_Toc4843514)

[4.1.7 Consultas 40](#_Toc4843515)

# Introdução

O trabalho desenvolvido na gestão de uma rede de farmácias pode ser muitas vezes desgastante, principalmente se não estiver sendo realizado em um ambiente automatizado. As centenas de entradas e saídas de clientes e de produtos do estoque podem acarretar uma perca de eficiência e de lucro se o controle dos dados não for feito de maneira automatizada, esse problema pode ser tornar ainda maior se tratando de um grande número de filiais para se administrar. Dessa maneira, visando uma melhor alocação das informações de modo que possam ser analisadas para promover uma melhora na gestão dessas unidades farmacêuticas, foram consultadas farmácias da região para auxiliar no desenvolvimento deste trabalho.

# Modelo Conceitual

# 2.1. Levantamento dos requisitos

Uma rede de farmácias conglomera várias unidades farmacêuticas, gerando assim a necessidade de se armazenar informações como: CNPJ, razão social, endereço e os telefones de contato. As Unidades farmacêuticas podem ser abastecidas por diversos fornecedores, demandando que as informações desses abastecedores sejam salvas como: CNPJ, nome, endereço e os telefones de contato. Cada fornecedor pode abastecer diversas farmácias.

Cada uma das unidades dessa rede pode empregar diversos funcionários, sendo assim necessário armazenar informações sobre eles, como: nome, CPF, matricula, salario, telefones de contato e endereço. Cada funcionário pode trabalhar em várias farmácias. De modo mais especifico, as farmácias necessitam de um farmacêutico que será responsável pela sua administração, sendo assim necessário armazenar informações especificas dele, como: Número no conselho regional de medicina e e-mail. É necessário salvar dados das sessões presentes na farmácia, como o código da sessão e o nome. Uma sessão aloca diversos funcionários e cada empregado deve trabalhar em apenas uma sessão. Além disso, cada sessão tem apenas um funcionário que a gerencia, não podendo gerenciar outras sessões. É necessário armazenar a data de início e data de término da gerência.

Sobre os produtos vendidos informações como: nome, descrição, validade, porcentagem de comissão, código de barras, preço por unidade e fabricante precisam ser salvas. Mais especificamente, notou-se a necessidade de se armazenar dados sobre os produtos de controle especial, onde é armazenada a tarja que ele apresenta. Para a compra desses produtos especiais é necessário apresentar uma receita médica, que apresenta os seguintes dados: classificação, paciente, prescrição, número e o médico que autorizou. Cada receita pode ser usada para a compra de vários controlados, e cada produto estar contido em várias receitas. Sobre o médico que autoriza as receitas é necessário armazenar dados como: número no conselho regional de medicina, CPF e nome. A receita pode ser autorizada por no máximo um médico, que por sua vez pode autorizar várias receitas.

Cada produto pode ser armazenado em diversas farmácias, sendo necessário armazenar informações como: estoque atual, status e estoque mínimo dos produtos armazenados. Cada farmácia estar relacionada a vários produtos. Os produtos são divididos e alocados em cada sessão. Um produto pode ser alocado em no máximo uma sessão, e uma sessão pode alocar diversos produtos.

Observando as numerosas vendas realizadas nas farmácias, foi notada a necessidade de se persistir dados como: data, hora e total da venda, funcionário que realizou a venda, os produtos adquiridos tais como a quantidade e o preço unitário de cada um. Cada venda é realizada por apenas um funcionário, que por sua vez pode realizar várias vendas. Um produto pode ser vendido em diversas vendas, assim como uma venda pode envolver vários produtos. As farmácias da franquia oferecem três formas de pagamento: dinheiro, cartão de credito e crediário, caso a compra seja paga com cartão de credito, devem ser armazenados: um número único, bandeira, número de parcelas e nome do titular. Cada cartão pode ser utilizado em diversas vendas à cartão e cada venda pode utilizar diversos cartões.

Caso opte pela forma de crediário, é necessário que seja feito um cadastro do cliente, onde devem ser salvos dados como: nome, CPF, sexo, telefones de contato, nascimento, endereço e debito. Cada venda crediário pode ser feita por apenas um cliente, esse por sua vez pode responder por várias compras nessa modalidade.

Contudo, o simples armazenamento dessas informações não traz benefícios tão relevantes além da clara organização dos dados. Dessa forma é necessário que sejam adicionadas algumas funcionalidades. Através de uma consulta, devem ser apresentados uma lista de produtos que estejam fora do período de validade, para que dessa forma não sejam postos à venda produtos prescritos. Além disso, para facilitar que o cliente adquira o produto com maior rapidez, é necessário que possa ser consultado em quais farmácias determinado produto está disponível.

Ainda na perspectiva dos produtos, é importante também que se possa consultar sempre que necessário, a quantidade de cada produto presente na farmacia, para que sejam repostos quando preciso. Em uma análise mais ampla, se faz necessário que o estado relacionado a quantidade dos produtos nas farmácias seja consultado, para que se saiba quais se encontram em estoque baixo e quais se encontram em estoque confortável.

Sobre as vendas, é importante que seja feita uma pesquisa que retorne o nome de cada cliente que não está em debito com a farmácia, em ordem alfabetica.

Para a localização mais eficiente dos produtos dentro da farmácia, é necessário que possa ser consultado os produtos disponíveis em cada sessão.

Cada produto vendido é composto por uma percentagem de comissão que é dada ao funcionário que realizou a venda, desse modo, é necessário que seja feita uma consulta que retorne o total de comissão adquirida para cada funcionário que deve ser somada ao seu salário e seja realizado seu pagamento.

Ainda no âmbito dos funcionários, é necessário que seja observado quais funcionários que são gerente de alguma sessão. É importante também que possam ser consultados quais funcionários realizaram mais vendas, afim de verificar a produtividade de cada um.

Por fim, é preciso saber quais funcionário nunca realizaram vendas na farmácia.

# Diagrama Entidade – Relacionamento

# Dicionário conceitual de dados

* **Entidade Farmácia:**

É uma entidade criada para armazenar informações referentes a uma farmácia.

* **Atributo (s):**
* CNPJ: Atributo chave que armazena o CNPJ da farmácia.
* Razão social: Atributo que armazena o nome de registro da farmácia.
* Endereço: Atributo que armazena o endereço da farmácia.
* Telefone: Atributo multivalorado que armazena os números de telefone da farmácia.
* **Entidade Fornecedor:**

É uma entidade criada para armazenar informações referentes a um fornecedor.

* **Atributo (s):**
* CNPJ: Atributo chave que armazena o CNPJ do fornecedor.
* Nome: Atributo que armazena o nome do fornecedor.
* Telefone: Atributo multivalorado que armazena os telefones do fornecedor.
* Endereço: Atributo que armazena o endereço do fornecedor.
* **Entidade Sessão:**

É uma entidade criada para armazenar informações referentes a uma sessão.

* **Atributo (s):**
* Código: Atributo chave que armazena um código identificador da sessão.
* Nome: Atributo que armazena o nome da sessão.
* **Entidade Funcionário:**

É uma generalização de farmacêutico criada para armazenar informações referentes a um funcionário.

* **Atributo (s):**
* Nome: Atributo que armazena o nome do funcionário.
* CPF: Atributo chave que armazena o CPF do funcionário.
* Salário: Atributo que armazena o valor do salário do funcionário
* Matricula: Atributo chave que armazena a matricula do funcionário.
* Telefone: Atributo multivalorado que armazena os telefones referentes ao funcionário.
* Endereço: Atributo que armazena o endereço referente ao funcionário.
* **Entidade Farmacêutico:**

É uma especificação de funcionário criada para armazenar informações referentes a um farmacêutico.

* **Atributo (s):**
* NúmeroCRF: Atributo chave que armazena o número do conselho regional de farmácia do farmacêutico.
* E-mail: Atributo chave que armazena o e-mail do farmacêutico
* **Entidade Venda:**

É uma generalização de venda cartão e venda prazo criada para armazenar informações de venda comuns a essas duas entidades.

* **Atributo (s):**
* Data: Atributo que armazena a data que foi realizada a venda.
* Hora: Atributo que armazena a hora que foi realizada a venda.
* Total: atributo que armazena o valor total da venda.
* **Entidade Venda Cartão:**

É uma especificação da entidade venda criada para armazenar informações sobre o número de parcelas da compra.

* **Atributo (s):**
* Número de Parcelas: Atributo que armazena o número de parcelas da compra.

* **Entidade Cartão de Crédito:**

É uma entidade criada para armazenar informações referentes a um cartão de credito.

* **Atributo (s):**
* Número: Atributo chave que armazena o número do cartão.
* Bandeira: Atributo que armazena a bandeira referente ao cartão.
* Número de parcelas: Atributo que armazena o número de parcelas da venda realizada.
* Nome: Atributo que armazena o nome do titular do cartão.
* **Entidade Crediário:**

É uma especificação da entidade vendas criadas para armazenar informações referente ao debito dos clientes.

* **Entidade Cliente:**

É uma entidade criada para armazenar informações referente a um cliente.

* **Atributo (s):**
* Nome: Atributo que armazena o nome do cliente.
* CPF: Atributo chave que armazena o CPF do cliente.
* Sexo: Atributo que armazena o sexo do cliente.
* Telefone: Atributo multivalorado que armazena os números de telefone dos clientes.
* Nascimento: Atributo que armazena a data de nascimento do cliente.
* Endereço: Atributo que armazena o endereço do cliente.
* Débito: Atributo que armazena o valor do débito do cliente.
* **Entidade Produto:**

É uma generalização de receituário medico criada para armazenar informações comuns a todos os produtos.

* **Atributo (s):**
* Nome: Atributo que armazena o nome do produto.
* Descrição: Atributo que armazena uma breve descrição do produto.
* Validade: Atributo que armazena a data de validade do produto.
* % comissão: Atributo que armazena o valor de comissão que um funcionário ganha na venda do produto.
* Código de barras: Atributo que armazena um código de barras que identifica o produto.
* Preço unitário: Atributo que armazena o valor de uma unidade do produto.
* Fabricante: Atributo que armazena o nome do fabricante do produto.
* **Entidade Receituário medico:**

É uma especificação de produtos criada para armazenar informações referentes a produtos que necessitem de receita.

* **Atributo (s):**
* Tarja: Atributo que armazena a tarja presente no produto.
* **Entidade Receita:**

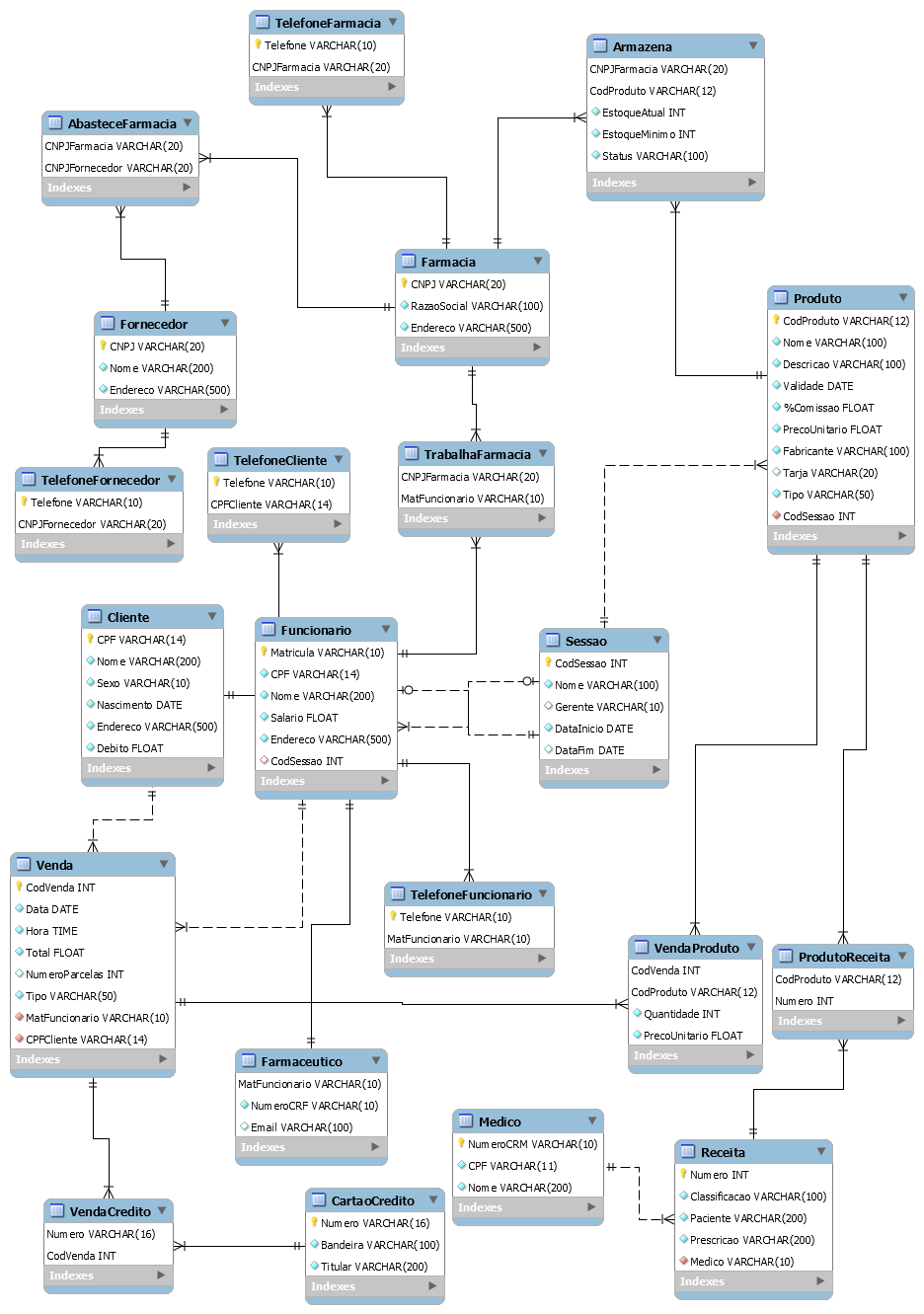
É uma entidade criada para armazenar informações referentes às receitas medicas.

* **Atributo (s):**
* Classificação: Atributo que armazena a classificação presente na receita.
* Paciente: Atributo que armazena o nome do paciente que necessita do produto.
* Prescrição: Atributo que armazena informação referente ao modo de administrar o produto.
* Número: Atributo chave que armazena o número da receita medica.
* **Entidade Médico:**

É uma entidade criada para armazenar informações referentes ao médico que autoriza uma receita.

* **Atributo (s):**
* NúmeroCRM: Atributo chave que armazena o número do conselho regional de medicina do médico.
* CPF: Atributo chave que armazena o CPF do médico.
* Nome: Atributo que armazena o nome do médico.
* **Relacionamentos**
* **Abastecida:** É um relacionamento entre as entidades *Farmácia* e *Fornecedor*. Uma farmácia pode ser abastecida, no mínimo, por um fornecedor e o máximo a vários. Um fornecedor pode abastecer, no mínimo uma farmácia e no máximo a várias farmácias.
* **Emprega:** É um relacionamento entre as entidades *Farmácia* e *Funcionário.* Uma farmácia pode empregar, no mínimo, um funcionário e no máximo vários. Um funcionário pode estar empregado, no mínimo, em uma farmácia, e no máximo e várias.
* **Trabalha:** É um relacionamento entre as entidades *Funcionário* e *Sessão.* Um funcionário pode trabalhar, no geral, em apenas uma sessão. Uma sessão pode estar relacionada, no mínimo, a um funcionário e no máximo vários
* **Gerencia:** É um relacionamento entre as entidades *Funcionário* e *Sessão.* Um funcionário pode gerenciar, no mínimo, nenhuma sessão e no máximo uma única sessão. Uma sessão pode ser gerenciada, no geral por um funcionário. Além disso contém dois atributos, Data de início e Data de término, que iram salvar respectivamente a data de início e de termino da gerencia.
* **Realiza:** É um relacionamento entre as entidades *Funcionário* e *Venda.* Um funcionário pode realizar, no mínimo, nenhuma venda e no máximo várias vendas. Uma venda por ser realizada no geral por apenas um funcionário.
* **Vendido:** É um relacionamento entre as entidades *Venda* e *Produto.* Em uma venda podem ser vendidos, no mínimo, um produto e no máximo vários. Um produto pode, no mínimo, não estar relacionado a nenhuma venda, e no máximo, estar relacionado a várias vendas. Além disso possui dois atributos denominados quantidade e preço unitário, que devem armazenar respectivamente a quantidade e o preço de cada produto contido na venda.
* **Utiliza:** É um relacionamento entre a especialização *Venda Cartão* e a entidade *Cartão de Credito.* Uma venda a cartão pode ser realizada com, no mínimo, um cartão de credito, e no máximo, vários cartões. Um cartão de credito pode estar relacionado a, no mínimo, uma venda a cartão, e no máximo, várias vendas.
* **Feita:** É um relacionamento entre a especialização *Crediário* e a entidade *Cliente.* Uma venda crediário pode realizada, no geral, por apenas um cliente. Um cliente pode estar associado a, no mínimo, uma venda crediário, e no máximo, a várias vendas.
* **Exige:** É um relacionamento entre a especialização *Receituário médico* e a entidade *Receita.* Um produto de receituário médico pode estar contido em, no mínimo, nenhuma receita, e no máximo, varias. Uma receita médica pode ser usada para se adquirir, no mínimo, um produto e no máximo vários.
* **Autorizada:** É um relacionamento entre as entidades *Receita* e *Médico.* Uma receita pode ser autorizada, no geral, por apenas um médico. Um médico por autorizar, no mínimo, nenhuma receita e no máximo várias receitas.
* **Armazena:** É um relacionamento entre as entidades *Farmacia* e *Produto.* Uma Farmacia pode armazenar, no mínimo, um produto e no máximo vários produtos. Um produto pode estar armazenado, no mínimo, em nenhuma farmacia, e no máximo em várias farmacias. Além disso contem três atributos: estoque atual, estoque mínimo e status, que armazenam respectivamente a quantidade atual do produto, a quantidade mínima necessária e o status de quantidade.
* **Aloca:** É um relacionamento entre as entidades *Sessão* e *Produto*. Uma sessão pode alocar, no mínimo, um produto, e no máximo vários produtos. Um produto pode estar disponível em no mínimo nenhuma sessão e no máximo uma sessão.

# Modelo Lógico

* 1. Mapeamento Entidade Relacionamento****

# Dicionário Lógico de Dados

|  |  |  |  |
| --- | --- | --- | --- |
| **FARMACIA: Relação que armazena os dados da farmácia** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| CNPJ | Representa o número do CNPJ da Farmácia | VarChar (20) | * Chave Primaria |
| RazaoSocial | Representa o nome da farmácia | VarChar (100) | * Não Nulo |
| Endereco | Representa o endereço da farmácia | VarChar (500) | * Não Nulo |

|  |  |  |  |
| --- | --- | --- | --- |
| **TELEFONEFARMACIA: Relação que armazena os telefones da farmácia** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| Telefone | Representa o telefone da farmácia | VarChar (10) | * Chave Primaria |
| CNPJFarmacia | Representa o CNPJ da farmácia | VarChar (20) | * Chave Primaria * Chave Estrangeira da tabela Farmacia |

|  |  |  |  |
| --- | --- | --- | --- |
| **FORNECEDOR: Relação que armazena os dados dos fornecedores** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| CNPJ | Representa o número do CNPJ do fornecedor | VarChar (20) | * Chave Primaria |
| Nome | Representa o nome do fornecedor | VarChar (200) | * Não nulo |
| Endereco | Representa o endereço do fornecedor | VarChar (500) | * Não nulo |

|  |  |  |  |
| --- | --- | --- | --- |
| **TELEFONEFORNECEDOR: Relação que armazena os telefones do fornecedor** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| Telefone | Representa o telefone da farmácia | VarChar (10) | * Chave Primaria |
| CNPJFornecedor | Representa o CNPJ do fornecedor | VarChar (20) | * Chave Primaria * Chave Estrangeira da tabela Fornecedor |

|  |  |  |  |
| --- | --- | --- | --- |
| **SESSÃO: Relação que armazena dados das sessões da farmácia** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| CodSessao | Representa o código único da sessão | Serial | * Chave Primaria |
| Nome | Representa o nome da sessão | VarChar (100) | * Não Nulo |
| Gerente | Representa o funcionário que gerencia a sessão | VarChar (10) | * Chave Estrangeira da tabela funcionário |
| DataInicio | Representa a data de início da gerência | Date | * Sem restrição |
| DataFim | Representa a data de fim da gerência | Date | * Sem restrição |

|  |  |  |  |
| --- | --- | --- | --- |
| **FUNCIONÁRIO: Relação que armazena dados gerais dos funcionários** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| Matricula | Representa a matricula do funcionário | VarChar (10) | * Chave Primaria |
| CPF | Representa o CPF do funcionário | VarChar (14) | * Não Nulo * Chave |
| Nome | Representa o nome do funcionário | VarChar (200) | * Não Nulo |
| Salario | Representa o valor do salário do funcionário | Float | * Não Nulo |
| Endereco | Representa o endereço do funcionário | VarChar (500) | * Não Nulo |
| CodSessao | Representa o código único da sessão | Serial | * Chave Estrangeira da tabela Sessão |

|  |  |  |  |
| --- | --- | --- | --- |
| **TELEFONEFUNCIONÁRIO: Relação que armazena os telefones dos funcionários** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| Telefone | Representa o telefone do funcionário | VarChar (10) | * Chave Primaria |
| MatFuncionario | Representa a matricula do funcionário | VarChar (10) | * Chave Primaria * Chave Estrangeira da tabela Funcionário |

|  |  |  |  |
| --- | --- | --- | --- |
| **FARMACÊUTICO: Relação que armazena dados específicos dos farmacêuticos** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| MatFuncionario | Representa a matricula do funcionário | VarChar (10) | * Chave Primaria * Chave Estrangeira da tabela Funcionário |
| NumeroCRF | Representa o registro no conselho regional de farmácia | VarChar (10) | * Chave * Não Nulo |
| Email | Representa o email do farmacêutico | VarChar (100) | * Chave * Não Nulo |

|  |  |  |  |
| --- | --- | --- | --- |
| **VENDA: Relação que armazena dados das vendas** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| CodVenda | Representa o código único da venda | Serial | * Chave Primaria |
| Data | Representa a data da venda | Date | * Não Nulo |
| Hora | Representa a hora da venda | Time | * Não Nulo |
| Total | Representa o total da venda | Float | * Não Nulo |
| NumeroParcelas | Representa o número de parcelas da venda | Integer | * Sem Restrição |
| Tipo | Representa o tipo da venda | VarChar (50) | * Não Nulo |
| MatFuncionario | Representa a matricula do funcionário | VarChar (10) | * Não Nulo * Chave Estrangeira da tabela Funcionário |
| CPFCliente | Representa o CPF do cliente | VarChar (11) | * Não Nulo * Chave Estrangeira da tabela Cliente |

|  |  |  |  |
| --- | --- | --- | --- |
| **CARTAOCREDITO: Relação que armazena dados do cartão de credito** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| Numero | Representa o número do cartão de credito | VarChar (16) | * Chave Primaria |
| Bandeira | Representa a bandeira do cartão de credito | VarChar (100) | * Não Nulo |
| Titular | Representa o titular do cartão de credito | VarChar (200) | * Não Nulo |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CLIENTE: Relação que armazena dados do cliente** | | | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | | **DOMÍNIO** | | **RESTRIÇÕES** |
| CPF | Representa o CPF do cliente | | VarChar (14) | | * Chave Primaria |
| Nome | Representa o nome do cliente | | VarChar (200) | | * Não Nulo |
| Sexo | Representa o sexo do cliente | | VarChar (10) | | * Não Nulo |
| Nascimento | Representa a data de nascimento do cliente | | Date | | * Não Nulo |
| Debito | | Representa o valor do debito da venda | Float | * Não Nulo | |
| Endereco | Representa o endereço do cliente | | VarChar(500) | | * Não Nulo |

|  |  |  |  |
| --- | --- | --- | --- |
| **TELEFONECLIENTE: Relação que armazena os telefones do cliente** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| Telefone | Representa o telefone do cliente | VarChar (10) | * Chave Primaria |
| CPFCliente | Representa o CPF do cliente | VarChar (11) | * Chave Primaria * Chave Estrangeira da tabela Cliente |

|  |  |  |  |
| --- | --- | --- | --- |
| **PRODUTO: Relação que armazena os dados do produto** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| CodProduto | Representa o código de barras do produto | VarChar (12) | * Chave Primaria |
| Nome | Representa o nome do produto | VarChar (100) | * Não Nulo |
| Descrição | Representa a descrição do produto | VarChar (100) | * Não Nulo |
| Validade | Representa a data de vencimento do produto | Date | * Não Nulo |
| Comissao | Representa a porcentagem de comissão adquirida na venda do produto | Float | * Não Nulo |
| PrecoUnitario | Representa o preço de venda por unidade | Float | * Não Nulo |
| Fabricante | Representa o nome do fabricante do produto | VarChar (100) | * Não Nulo |
| Tarja | Representa a tarja presente no produto | VarChar (20) | * Sem Restrição |
| Tipo | Representa o tipo de remédio | VarChar (50) | * Não Nulo |
| CodSessao | Representa o código único da sessão | Serial | * Chave Estrangeira da tabela Sessão * Não Nulo |

|  |  |  |  |
| --- | --- | --- | --- |
| **RECEITA: Relação que armazena os dados da receita medica** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| Numero | Representa o número da receita | Serial | * Chave Primaria |
| Classificação | Representa a classificação da receita | VarChar (100) | * Não Nulo |
| Paciente | Representa o paciente que necessita do produto | VarChar (200) | * Não Nulo |
| Prescrição | Representa como deve ser administrado o produto | VarChar (200) | * Não Nulo |
| Medico | Representa o médico que autorizou receita | VarChar (10) | * Chave Estrangeira da tabela medico * Não Nulo |

|  |  |  |  |
| --- | --- | --- | --- |
| **MEDICO: Relação que armazena os dados do médico** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| NumeroCRM | Representa o registro no conselho regional de medicina | VarChar (10) | * Chave Primaria |
| CPF | Representa o CPF do médico | VarChar (14) | * Não Nulo * Chave |
| Nome | Representa o nome do médico | VarChar (200) | * Não Nulo |

|  |  |  |  |
| --- | --- | --- | --- |
| **ABASTECEFARMACIA: Relação que armazena os dados dos fornecedores que abastecem a farmácia** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| CNPJFarmacia | Representa o CNPJ da farmácia | VarChar (20) | * Chave Primaria * Chave Estrangeira da tabela Farmacia |
| CNPJFornecedor | Representa o CNPJ do fornecedor | VarChar (20) | * Chave Primaria * Chave Estrangeira da tabela Fornecedor |

|  |  |  |  |
| --- | --- | --- | --- |
| **TRABALHAFARMACIA: Relação que armazena os dados dos funcionários que trabalham na farmácia** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| CNPJFarmacia | Representa o CNPJ da farmácia | VarChar (20) | * Chave Primaria * Chave Estrangeira da tabela Farmacia |
| MatFuncionario | Representa a matricula do funcionário | VarChar (10) | * Chave Primaria * Chave Estrangeira da tabela Funcionário |

|  |  |  |  |
| --- | --- | --- | --- |
| **ARMAZENAPRODUTO: Relação que armazena dados dos produtos e farmácias em que estão presentes** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| CNPJFarmacia | Representa o CNPJ da farmácia | VarChar (20) | * Chave Primaria * Chave Estrangeira da tabela Farmacia |
| CodProduto | Representa o código de barras do produto | VarChar (12) | * Chave Primaria * Chave Estrangeira da tabela Produto |
| EstoqueAtual | Representa o valor do estoque atual | Integer | * Não Nulo |
| Status | Representa o estado em que o estoque se encontra | VarChar (100) | * Não Nulo |
| EstoqueMinimo | Representa o menor valor do estoque aceitável | Integer | * Não Nulo |

|  |  |  |  |
| --- | --- | --- | --- |
| **VENDAPRODUTO: Relação que armazena os dados da venda e dos produtos associados a ela** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| CodVenda | Representa o código único da venda | Serial | * Chave Primaria * Chave Estrangeira da tabela Venda |
| CodProduto | Representa o código de barras do produto | VarChar (12) | * Chave Primaria * Chave Estrangeira da tabela Produto |
| Quantidade | Representa a quantidade do produto presente na venda | Int | * Não Nulo |
| PrecoUnitario | Representa a preço por unidade do produto presente na venda | Float | * Não Nulo |

|  |  |  |  |
| --- | --- | --- | --- |
| **VENDACREDITO: Relação que armazena os dados da venda e dos cartões de credito associados a ela** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| Numero | Representa o número do cartão de credito | VarChar (16) | * Chave Primaria * Chave Estrangeira da tabela CartaoCredito |
| CodVenda | Representa o código único da venda | Serial | * Chave Primaria * Chave Estrangeira da tabela Venda |

|  |  |  |  |
| --- | --- | --- | --- |
| **PRODUTORECEITA: Relação que armazena os dados do produto e das receitas associadas a ele** | | | |
| **ATRIBUTO** | **DESCRIÇÃO** | **DOMÍNIO** | **RESTRIÇÕES** |
| CodProduto | Representa o código de barras do produto | VarChar (12) | * Chave Primaria * Chave Estrangeira da tabela Produto |
| Numero | Representa o número da receita | Serial | * Chave Primaria * Chave Estrangeira da tabela Receita |

# Modelo Físico

# 4.1. Scripts SQL

# 4.1.1. Criando Relações

* **Criando a relação FARMACIA**

CREATE TABLE farmacia(

cnpj VARCHAR(20),

razaoSocial VARCHAR(100) NOT NULL,

endereco VARCHAR(500) NOT NULL,

CONSTRAINT cnpjFarmacia\_pk PRIMARY KEY(cnpj)

);

* **Criando a relação TELEFONEFARMACIA**

CREATE TABLE telefoneFarmacia(

telefone VARCHAR(10),

cnpjFarmacia VARCHAR(20),

CONSTRAINT telefoneFarmacia\_pk

PRIMARY KEY(telefone,cnpjFarmacia),

CONSTRAINT cnpjFarmacia\_fk FOREIGN KEY (cnpjFarmacia)

REFERENCES farmacia(cnpj)

);

* **Criando a relação FORNECEDOR**

CREATE TABLE fornecedor(

cnpj VARCHAR(20),

nome VARCHAR(200) NOT NULL,

endereco VARCHAR(500) NOT NULL,

CONSTRAINT cnpjFornecedor\_pk PRIMARY KEY(cnpj)

);

* **Criando a relação TELEFONEFORNECEDOR**

CREATE TABLE telefoneFornecedor(

telefone VARCHAR(10),

cnpjFornecedor VARCHAR(20),

CONSTRAINT telefoneFornecedor\_pk PRIMARY KEY(telefone, cnpjFornecedor),

CONSTRAINT cnpjFornecedor\_fk FOREIGN KEY(cnpjFornecedor)

REFERENCES fornecedor(cnpj)

);

* **Criando a relação SESSAO**

CREATE TABLE sessao(

codSessao serial,

nome VARCHAR(100) NOT NULL,

dataInicio DATE,

dataFim DATE,

CONSTRAINT codSessao\_pk PRIMARY KEY(codSessao)

);

* **Criando a relação FUNCIONARIO**

CREATE TABLE funcionario(

matricula VARCHAR(10),

cpf VARCHAR(14) UNIQUE NOT NULL,

nome VARCHAR(200) NOT NULL,

salario FLOAT NOT NULL,

endereco VARCHAR(500) NOT NULL,

codSessao SERIAL,

CONSTRAINT matricula\_pk PRIMARY KEY(matricula),

CONSTRAINT limitaSalario CHECK(salario>0),

CONSTRAINT codSessao\_fk FOREIGN KEY(codSessao)

REFERENCES sessao(codSessao)

);

* **Atualizando a relação sessão inserindo um atributo Gerente**

ALTER TABLE sessao ADD gerente VARCHAR(10),

ADD CONSTRAINT gerente\_fk FOREIGN KEY(gerente)

REFERENCES funcionario(matricula);

* **Criando a relação TELEFONEFUNCIONARIO**

CREATE TABLE telefoneFuncionario(

telefone VARCHAR(10),

matFuncionario VARCHAR(10),

CONSTRAINT telefoneFuncionario\_pk PRIMARY KEY(telefone, matFuncionario),

CONSTRAINT matFuncionario\_fk FOREIGN KEY(matFuncionario)

REFERENCES funcionario(matricula)

);

* **Criando a relação CARTAOCREDITO**

CREATE TABLE cartaoCredito(

numero VARCHAR(16),

bandeira VARCHAR(100) NOT NULL,

titular VARCHAR(200) NOT NULL,

CONSTRAINT numero\_pk PRIMARY KEY(numero)

);

* **Criando a relação FARMACEUTICO**

CREATE TABLE farmaceutico(

matFuncionario VARCHAR(10),

numeroCrf VARCHAR(10) UNIQUE NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

CONSTRAINT matFuncionario\_pk PRIMARY KEY(matFuncionario),

CONSTRAINT matFuncionario\_fk FOREIGN KEY(matFuncionario)

REFERENCES funcionario(matricula)

);

* **Criando a relação CLIENTE**

CREATE TABLE cliente(

cpf VARCHAR(14),

nome VARCHAR(200) NOT NULL,

sexo VARCHAR(10) NOT NULL,

nascimento DATE NOT NULL,

endereco VARCHAR(500) NOT NULL,

debito FLOAT NOT NULL DEFAULT 0,

CONSTRAINT cpf\_pk PRIMARY KEY(cpf)

);

* **Criando a relação TELEFONECLIENTE**

CREATE TABLE telefoneCliente(

telefone VARCHAR(10),

cpfCliente VARCHAR(14),

CONSTRAINT telefoneCliente\_pk PRIMARY KEY(telefone,cpfCliente),

CONSTRAINT cpfCliente\_fk FOREIGN KEY (cpfCliente)

REFERENCES cliente(cpf)

);

* **Criando a relação MEDICO**

CREATE TABLE medico(

numeroCrm VARCHAR(10),

cpf VARCHAR(14) UNIQUE NOT NULL,

nome VARCHAR(200) NOT NULL,

CONSTRAINT numeroCrm\_key PRIMARY KEY(numeroCrm)

);

* **Criando a relação RECEITA**

CREATE TABLE receita(

numeroReceita SERIAL,

classificacao VARCHAR(100) NOT NULL,

paciente VARCHAR(200) NOT NULL,

prescricao VARCHAR(200) NOT NULL,

medico VARCHAR(10) NOT NULL,

CONSTRAINT numeroReceita\_pk PRIMARY KEY(numeroReceita),

CONSTRAINT medico\_fk FOREIGN KEY(medico)

REFERENCES medico(numeroCrm)

);

* **Criando a relação PRODUTO**

CREATE TABLE produto(

codProduto VARCHAR(12),

nome VARCHAR(100) NOT NULL,

descricao VARCHAR(100) NOT NULL,

validade DATE NOT NULL,

comissao FLOAT NOT NULL,

precoUnitario FLOAT NOT NULL,

fabricante VARCHAR(100) NOT NULL,

tarja VARCHAR(20),

tipo VARCHAR(50) NOT NULL,

codSessao SERIAL NOT NULL,

CONSTRAINT codProduto\_pk PRIMARY KEY(codProduto),

CONSTRAINT codSessao\_fk FOREIGN KEY(codSessao)

REFERENCES sessao(codSessao)

);

* **Criando a relação PRODUTO**

CREATE TABLE venda(

codVenda SERIAL,

data DATE NOT NULL,

hora TIME NOT NULL,

total FLOAT NOT NULL,

numeroParcelas INTEGER,

tipo VARCHAR(50) NOT NULL,

matFuncionario VARCHAR(10) NOT NULL,

cpfCliente VARCHAR(14),

CONSTRAINT codVenda\_pk PRIMARY KEY(codVenda),

CONSTRAINT matFuncionario\_fk FOREIGN KEY(matFuncionario)

REFERENCES funcionario(matricula),

CONSTRAINT cpfCliente\_fk FOREIGN KEY(cpfCliente)

REFERENCES cliente(cpf)

);

* **Criando a relação ABASTECEFARMACIA**

CREATE TABLE abasteceFarmacia(

cnpjFarmacia VARCHAR(20),

cnpjFornecedor VARCHAR(20),

CONSTRAINT abasteceFarmacia\_pk PRIMARY KEY(cnpjFarmacia, cnpjFornecedor),

CONSTRAINT cnpjFarmacia\_fk FOREIGN KEY(cnpjFarmacia)

REFERENCES farmacia(cnpj),

CONSTRAINT cnpjFornecedor\_fk FOREIGN KEY(cnpjFornecedor)

REFERENCES fornecedor(cnpj)

);

* **Criando a relação ABASTECEFARMACIA**

CREATE TABLE armazenaProduto(

cnpjFarmacia VARCHAR(20),

codProduto VARCHAR(12),

estoqueAtual INTEGER NOT NULL,

status VARCHAR(100) NOT NULL,

estoqueMinimo INTEGER NOT NULL,

CONSTRAINT armazenaProduto\_pk PRIMARY KEY(cnpjFarmacia, codProduto),

CONSTRAINT cnpjFarmacia\_fk FOREIGN KEY(cnpjFarmacia)

REFERENCES farmacia(cnpj),

CONSTRAINT codProduto\_fk FOREIGN KEY(codProduto)

REFERENCES produto(codProduto)

);

* **Criando a relação PRODUTORECEITA**

CREATE TABLE produtoReceita(

codProduto VARCHAR(12),

numeroReceita SERIAL,

CONSTRAINT produtoReceita\_pk PRIMARY KEY(codProduto, numeroReceita),

CONSTRAINT codProduto\_fk FOREIGN KEY(codProduto)

REFERENCES produto(codProduto),

CONSTRAINT numeroReceita\_fk FOREIGN KEY(numeroReceita)

REFERENCES receita(numeroReceita)

);

* **Criando a relação VENDACREDITO**

CREATE TABLE vendaCredito(

numero VARCHAR(16),

codVenda SERIAL,

CONSTRAINT vendaCredito\_pk PRIMARY KEY(numero, codVenda),

CONSTRAINT numero\_fk FOREIGN KEY(numero)

REFERENCES cartaoCredito(numero),

CONSTRAINT codVenda\_fk FOREIGN KEY(codVenda)

REFERENCES venda(codVenda)

);

* **Criando a relação VENDAPRODUTO**

CREATE TABLE vendaProduto(

codVenda SERIAL,

codProduto VARCHAR(12),

quantidade INTEGER NOT NULL,

precoUnitario FLOAT NOT NULL,

CONSTRAINT vendaProduto\_pk PRIMARY KEY(codVenda, codProduto),

CONSTRAINT codVenda\_fk FOREIGN KEY(codVenda)

REFERENCES venda(codVenda),

CONSTRAINT codProduto\_fk FOREIGN KEY(codProduto)

REFERENCES produto(codProduto)

);

* **Criando a relação TRABALHAFARMACIA**

CREATE TABLE trabalhaFarmacia(

cnpjFarmacia VARCHAR(20),

matFuncionario VARCHAR(10),

CONSTRAINT trabalhaFarmacia\_pk PRIMARY KEY(cnpjFarmacia, matFuncionario),

CONSTRAINT cnpjFarmacia\_fk FOREIGN KEY(cnpjFarmacia)

REFERENCES farmacia(cnpj),

CONSTRAINT matFuncionarior\_fk FOREIGN KEY(matFuncionario)

REFERENCES funcionario(matricula)

);

# 4.1.2. Criando Índices

CREATE INDEX abasteceFarmaciaIndice ON abasteceFarmacia(cnpjFarmacia);

CREATE INDEX abasteceFornecedorIndice ON abasteceFarmacia(cnpjFornecedor);

CREATE INDEX armazenaFarmaciaIndice ON armazenaProduto(cnpjFarmacia);

CREATE INDEX armazenaProdutoIndice ON armazenaProduto(codProduto);

CREATE INDEX numeroIndice ON cartaoCredito(numero);

CREATE INDEX cpfIndice ON cliente(cpf);

CREATE INDEX matFuncionarioIndice ON farmaceutico(matfuncionario);

CREATE INDEX cnpjFarmaciaIndice ON farmacia(cnpj);

CREATE INDEX cnpjFornecedorIndice ON fornecedor(cnpj);

CREATE INDEX matriculaIndice ON funcionario(Matricula);

CREATE INDEX numeroCRMIndice ON medico(numeroCRM);

CREATE INDEX codProdutoIndice ON produto(codProduto);

CREATE INDEX produtoReceitaProIndice ON produtoReceita(codproduto);

CREATE INDEX produtoReceitaNumIndice ON produtoReceita(numeroReceita);

CREATE INDEX numeroReceitaIndice ON receita(numeroreceita);

CREATE INDEX codSessaoIndice ON sessao(codSessao);

CREATE INDEX telefoneClienteCpfIndice ON telefoneCliente(cpfCliente);

CREATE INDEX telefoneClienteTelIndice ON telefoneCliente(telefone);

CREATE INDEX telefoneFarmaciaCnpjIndice ON telefoneFarmacia(cnpjFarmacia);

CREATE INDEX telefoneFarmaciaTelIndice ON telefoneFarmacia(telefone);

CREATE INDEX telefoneFornecedorCnpjIndice ON telefoneFornecedor(cnpjFornecedor);

CREATE INDEX telefoneFornecedorTelIndice ON telefoneFornecedor(telefone);

CREATE INDEX telefoneFuncionarioMatIndice ON telefoneFuncionario(matFuncionario);

CREATE INDEX telefoneFuncionarioTelIndice ON telefoneFuncionario(telefone);

CREATE INDEX trabalhaFarmaciaCnpjIndice ON trabalhaFarmacia(cnpjFarmacia);

CREATE INDEX trabalhaFarmaciaMatIndice ON trabalhaFarmacia(matFuncionario);

CREATE INDEX codVendaIndice ON venda(codVenda);

CREATE INDEX vendaCreditoNumIndice ON vendaCredito(numero);

CREATE INDEX vendaCreditoCodIndice ON vendaCredito(codVenda);

CREATE INDEX vendaProdutoVenIndice ON vendaProduto(codVenda);

CREATE INDEX vendaProdutoProIndice ON vendaProduto(codProduto);

# 4.1.3. Visões

* Essa visão cria uma junção entre as relações ARMAZENAPRODUTO, PRODUTO e FARMACIA onde auxilia na recuperação dos dados do estoque.

CREATE VIEW produtosEmEstoque

AS select \*

from farmacia f, armazenaproduto ap NATURAL JOIN produto p

where f.cnpj = ap.cnpjfarmacia

* Essa visão ultiliza a tabela venda para auxiliar na recuperação de tuplas em que o tipo de venda e feito com cartão de credito.

CREATE VIEW tipoVenda

AS select \*

from venda v

where v.tipo ILIKE 'Car%'

* Essa visão cria uma junção entre as relações VENDA, VENDACREDITO e CARTAOCREDITO onde auxilia na recuperação dos dados mais detalhados nas vendas feitas com cartão.

CREATE VIEW vendasCartao

AS select \*

from venda v NATURAL JOIN vendaCredito vc NATURAL JOIN cartaocredito cc

# 4.1.4. Procedimentos Armazenados

* Procedimento armazenado que retorna a média salarial dos funcionários cadastrados

CREATE OR REPLACE FUNCTION mediaSalarial()

RETURNS INTEGER AS $$

DECLARE

media INTEGER;

BEGIN

SELECT INTO media AVG(f.salario)

FROM funcionario f;

RETURN media;

END

$$ LANGUAGE plpgsql

* Procedimento armazenado que retorna o total de vendas feitas por um funcionario

CREATE OR REPLACE FUNCTION quantidadeVendas(VARCHAR)

RETURNS INTEGER AS $$

DECLARE

matricula ALIAS FOR $1;

total INTEGER;

BEGIN

SELECT INTO total COUNT(\*)

FROM venda v

WHERE v.matfuncionario = matricula;

RETURN total;

END

$$ LANGUAGE plpgsql;

* Procedimento armazenado que retorna o total de debito de todos os clientes

CREATE OR REPLACE FUNCTION somaDebito()

RETURNS VARCHAR AS $$

DECLARE

somaDebito cliente.debito%TYPE;

BEGIN

SELECT INTO somaDebito SUM(c.debito)

FROM cliente c;

RETURN somaDebito;

END

$$ LANGUAGE plpgsql

# 4.1.5. Gatilhos

* Gatilho responsável por atualizar o status do estoque sempre que os valores de estoque atual e estoque mínimo forem inseridos ou atualizados.

CREATE OR REPLACE FUNCTION verificaEstoque()

RETURNS TRIGGER AS $$

BEGIN

IF (New.estoqueAtual < New.estoqueMinimo)

THEN NEW.status = 'Estoque baixo';

RETURN NEW;

ELSE IF (New.estoqueAtual / New.estoqueMinimo) < 10.00

THEN NEW.status = 'Estoque Moderado';

RETURN NEW;

ELSE IF (New.estoqueAtual / New.estoqueMinimo) > 9.99

THEN NEW.status = 'Estoque confortavel';

RETURN NEW;

END IF;

END IF;

END IF;

END

$$ LANGUAGE PLPGSQL;

CREATE TRIGGER atualizaStatus

BEFORE INSERT OR UPDATE ON armazenaProduto

FOR EACH ROW

EXECUTE PROCEDURE verificaEstoque();

* Gatilho responsável por atualizar o valor do debito do cliente sempre que uma nova compra em seu nome for realizada.

CREATE OR REPLACE FUNCTION calculaDebito()

RETURNS TRIGGER AS $$

DECLARE

debitoAtual cliente.debito%TYPE;

BEGIN

SELECT INTO debitoAtual c.debito

FROM Cliente c

WHERE c.cpf = NEW.cpfCliente;

debitoAtual := debitoAtual + NEW.total;

UPDATE cliente SET debito = debitoAtual

WHERE cpf = NEW.cpfCliente;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER atualizaDebito

AFTER INSERT OR UPDATE ON venda

FOR EACH ROW

WHEN (NEW.tipo ILIKE 'debito')

EXECUTE PROCEDURE calculadebito();

* Gatilho responsável por impedir que o gerente de uma sessão possa ser cadastrado como um farmacêutico.

CREATE OR REPLACE FUNCTION verificaFuncionario()

RETURNS TRIGGER AS $$

DECLARE

gerente sessao.gerente%TYPE;

BEGIN

SELECT INTO gerente s.gerente

FROM sessao s

WHERE s.gerente = NEW.matfuncionario;

IF gerente = NEW.matfuncionario THEN RETURN NULL;

ELSE RETURN NEW;

END IF;

END

$$ LANGUAGE plpgsql;

CREATE TRIGGER alocaFarmaceutico

BEFORE INSERT OR UPDATE ON farmaceutico

FOR EACH ROW

EXECUTE PROCEDURE verificafuncionario();

# 4.1.6. Povoando as relações

* **Povoando a relação FARMACIA**

INSERT INTO farmacia

VALUES ('22.682.420/0001-38', 'Raia Drogasil', 'Rua do Comércio, 771, Centro, Cajazeiras-PB');

INSERT INTO farmacia

VALUES ('54.407.897/0001-00', 'Brazil Pharma', 'Rua Raul de Alencar, 946, Centro, Antônio Martins-RN');

INSERT INTO farmacia

VALUES ('15.721.825/0001-72', 'Panvel Farmácia', 'Rua Sebastiana Santina de Araújo Ferrreira, 794, Planalto Boa Esperança, João Pessoa-PB');

INSERT INTO farmacia

VALUES ('33.668.192/0001-68', 'Drogaria Venancio', 'Rua José do Nascimento, 157, Cidade dos Funcionários, Fortaleza-CE');

INSERT INTO farmacia

VALUES ('34.665.298/0001-70', 'Drogaria Pague Menos.', 'Rua São João Batista, 187, Povoado Soledade, Apodi-RN');

* **Povoando a relação TELEFONEFARMACIA**

INSERT INTO telefoneFarmacia VALUES('3832822423', '22.682.420/0001-38');

INSERT INTO telefoneFarmacia VALUES('0171085212', '54.407.897/0001-00');

INSERT INTO telefoneFarmacia VALUES('0298056079', '15.721.825/0001-72');

INSERT INTO telefoneFarmacia VALUES('4190708799', '33.668.192/0001-68');

INSERT INTO telefoneFarmacia VALUES('4374244240', '34.665.298/0001-70');

* **Povoando a relação FORNECEDOR**

INSERT INTO fornecedor VALUES('63.218.319/0001-08', 'Body Generics', 'Rua Manoel de Barros, 188, Centro, Cubati-PB');

INSERT INTO fornecedor VALUES('41.124.638/0001-11', 'DFrance', 'Praça do Centenário, 334, Centro, Itaporanga-PB');

INSERT INTO fornecedor VALUES('20.061.234/0001-56', 'IHerb', 'Praça Osvaldo Ribeiro, 733, Centro, Macajuba-BA');

INSERT INTO fornecedor VALUES('03.023.774/0001-26', 'IOffer', 'Rua Marcos Rodrigues, 921, Caldeirões, Caxias-MA');

INSERT INTO fornecedor VALUES('17.138.707/0001-61', 'Perfumagi', 'Rua Principal, 551, Povoado Ponta do Mel, Areia Branca-RN');

* **Povoando a relação TELEFONEFORNECEDOR**

INSERT INTO telefoneFornecedor VALUES('5158533885', '63.218.319/0001-08');

INSERT INTO telefoneFornecedor VALUES('1924313824', '41.124.638/0001-11');

INSERT INTO telefoneFornecedor VALUES('2697764388', '20.061.234/0001-56');

INSERT INTO telefoneFornecedor VALUES('7016631360', '03.023.774/0001-26');

INSERT INTO telefoneFornecedor VALUES('2005101676', '17.138.707/0001-61');

* **Povoando a relação SESSAO**

INSERT INTO sessao(nome) VALUES('Medicamentos');

INSERT INTO sessao(nome) VALUES('Perfumaria');

INSERT INTO sessao(nome) VALUES('Suplementos');

INSERT INTO sessao(nome) VALUES('Higiene Pessoal');

INSERT INTO sessao(nome) VALUES('Ofertas');

* **Povoando a relação FUNCIONARIO**

INSERT INTO funcionario (matricula, cpf, nome, salario, endereco, codSessao) VALUES('3624044444', '530.764.680-14', 'Jorge Danilo Thiago da Mata', 1000, 'Rua Santa CecÃ­lia, 233, Jardim Oasis, Cajazeiras-PB', 2);

INSERT INTO funcionario (matricula, cpf, nome, salario, endereco, codSessao) VALUES('4281285561', '060.871.510-79', 'Julia Marli Novaes', 2500, 'Rua SÃ£o Geraldo, 764, VarjÃ£o, JoÃ£o Pessoa-PB', 3);

INSERT INTO funcionario (matricula, cpf, nome, salario, endereco, codSessao) VALUES('7567227710', '586.501.270-37', 'Geraldo Kaique CÃ©sar de Paula', 2500, 'Avenida Lauro Vieira Chaves, 527, Aeroporto, Fortaleza-PB', 1);

INSERT INTO funcionario (matricula, cpf, nome, salario, endereco, codSessao) VALUES('6047297385', '395.613.460-52', 'Brenda Nina da Costa', 3000, 'Rua Raul de Alencar, 412, Centro, AntÃ´nio Martins-RN', 5);

INSERT INTO funcionario (matricula, cpf, nome, salario, endereco, codSessao) VALUES('2460211116', '905.999.390-01', 'Henry Murilo JoÃ£o AssunÃ§Ã£o', 4500, 'Rua Principal, 352, Povoado Soledade, Apodi-RN', 4);

INSERT INTO funcionario (matricula, cpf, nome, salario, endereco, codSessao) VALUES('7295867836', '923.111.390-96', 'Diego Vinicius Ramos', 1000, 'Rua Professora Terezinha Carvalho, 280, Salobrinho, Ilhéus-BA', 2);

INSERT INTO funcionario (matricula, cpf, nome, salario, endereco, codSessao) VALUES('6767216111', '636.265.560-50', 'Vanessa Sandra Cardoso', 2500, 'Servidão Holegário José da Luz, 223, Ingleses do Rio Vermelho, Florianópolis-SC', 3);

INSERT INTO funcionario (matricula, cpf, nome, salario, endereco, codSessao) VALUES('6967593465', '022.340.750-09', 'Samuel André Anthony Castro', 2500, 'Rua Cupuaçú, 115, Conjunto Jardim São Francisco, Rio Branco-AC', 1);

INSERT INTO funcionario (matricula, cpf, nome, salario, endereco, codSessao) VALUES('4534009816', '750.972.420-17', 'Daniela Isabel Viana', 3000, 'Rua Pandiá Calógeras, 544, Centro, Castro-PR', 5);

INSERT INTO funcionario (matricula, cpf, nome, salario, endereco, codSessao) VALUES('7044396123', '522.712.910-03', 'Luan Renato Fernandes', 4500, 'Rua Moisés Cutrim, 302, Vila Conceição, São Luís-MA', 4);

* **Povoando a relação TELEFONEFUNCIONARIO**

INSERT INTO telefoneFuncionario VALUES('9889450408', '7567227710');

INSERT INTO telefoneFuncionario VALUES('7299746612', '6047297385');

INSERT INTO telefoneFuncionario VALUES('4860801659', '2460211116');

INSERT INTO telefoneFuncionario VALUES('3643809794', '3624044444');

INSERT INTO telefoneFuncionario VALUES('8709789245', '4281285561');

* **Atualizando a relação SESSAO**

UPDATE sessao

SET Gerente = '3624044444', dataInicio = '2015/05/18'

WHERE codSessao = 1;

UPDATE sessao

SET Gerente = '6047297385', dataInicio = '2016/10/28'

WHERE codSessao = 2;

UPDATE sessao

SET Gerente = '4281285561', dataInicio = '2012/06/01'

WHERE codSessao = 3;

UPDATE sessao

SET Gerente = '7567227710', dataInicio = '2017/10/15'

WHERE codSessao = 4;

UPDATE sessao

SET Gerente = '2460211116', dataInicio = '2014/11/20'

WHERE codSessao = 5;

* **Povoando a relação FARMACEUTICO**

INSERT INTO farmaceutico VALUES('7295867836', '6841-5345', 'franciscovitorfogaca@ipmmi.org.br');

INSERT INTO farmaceutico VALUES('6767216111', '8696-0944', 'filipemarcelocastro@saojose.biz');

INSERT INTO farmaceutico VALUES('6967593465', '5565-6651', 'simoneisabelnovaes@hotmmail.com');

INSERT INTO farmaceutico VALUES('4534009816', '7967-4438', 'alexandreantonio@tpltransportes.com.br');

INSERT INTO farmaceutico VALUES('7044396123', '1154-5775', 'gabrielafabianamoraes@taclog.com.br');

* **Povoando a relação ABASTECEFARMACIA**

INSERT INTO abasteceFarmacia VALUES('54.407.897/0001-00', '63.218.319/0001-08');

INSERT INTO abasteceFarmacia VALUES('33.668.192/0001-68', '03.023.774/0001-26');

INSERT INTO abasteceFarmacia VALUES('34.665.298/0001-70', '17.138.707/0001-61');

INSERT INTO abasteceFarmacia VALUES('22.682.420/0001-38', '20.061.234/0001-56');

INSERT INTO abasteceFarmacia VALUES('15.721.825/0001-72', '41.124.638/0001-11');

* **Povoando a relação CARTAOCREDITO**

INSERT INTO cartaoCredito VALUES('5181213573840682', 'MasterCard', 'Silvana Adriana Isabella');

INSERT INTO cartaoCredito VALUES('4532530518265413', 'Visa', 'Victor Enzo Brito');

INSERT INTO cartaoCredito VALUES('5067079390440249', 'Elo', 'Fabiana Ester Pietra');

INSERT INTO cartaoCredito VALUES('4485409083091156', 'Visa', 'Yuri Diogo Luiz Aparício');

INSERT INTO cartaoCredito VALUES('5404611081910767', 'MasterCard', 'Gabrielly Aline Fabiana');

* **Povoando a relação CLIENTE**

INSERT INTO cliente (cpf, nome, sexo, nascimento, endereco)

VALUES('090.460.320-27', 'Jorge Francisco Osvaldo Almada', 'Masculino', '1997/06/09' , 'Rua Esplendor, 904, Gabriela, Feira de Santana-BA');

INSERT INTO cliente (cpf, nome, sexo, nascimento, endereco)

VALUES('305.093.100-01', 'Aline Sophie Pereira', 'Feminino', '1999/01/08' , 'Rua T 4, 307, Centro (Taquaralto), Palmas-TO');

INSERT INTO cliente (cpf, nome, sexo, nascimento, endereco)

VALUES('242.783.240-57', 'Severino Renato Oliver Pereira', 'Masculino', '1995/04/14' , 'Largo Dois de Julho, 979, Centro, Juazeiro-BA');

INSERT INTO cliente (cpf, nome, sexo, nascimento, endereco)

VALUES('342.476.170-28', 'Benedita Clarice Priscila', 'Feminino', '1998/03/20' , 'Rua São José, 882, Santo Antônio, Cacoal-RO');

INSERT INTO cliente (cpf, nome, sexo, nascimento, endereco)

VALUES('082.730.420-06', 'Alexandre Fernando Assis', 'Masculino', '1994/12/12' , 'Avenida Paraná, 942, Navegantes, Porto Alegre-RS');

* **Povoando a relação TELEFONECLIENTE**

INSERT INTO telefoneCliente VALUES('7701381858', '090.460.320-27');

INSERT INTO telefoneCliente VALUES('8253822768', '305.093.100-01');

INSERT INTO telefoneCliente VALUES('8754279097', '242.783.240-57');

INSERT INTO telefoneCliente VALUES('0819095883', '342.476.170-28');

INSERT INTO telefoneCliente VALUES('0697607378', '082.730.420-06');

* **Povoando a relação MEDICO**

INSERT INTO medico VALUES('4582-8118','319.662.880-93','Heitor Mário Noah Pires');

INSERT INTO medico VALUES('7089-9724','477.733.260-88','Rosa Flávia Vieira');

INSERT INTO medico VALUES('6816-6126','111.895.420-30','Antonio Raul Márcio Pires');

INSERT INTO medico VALUES('9915-0077','613.307.370-59','Giovana Andreia Allana');

INSERT INTO medico VALUES('1600-3308','936.562.110-00','Geraldo Yago Vieira');

* **Povoando a relação PRODUTO**

INSERT INTO produto (codProduto, nome, descricao, validade, comissao, precoUnitario, fabricante, tarja, tipo, codSessao)

VALUES('789491651258', 'ALPRAZOLAM', '0,25 MG COM CT BL AL PLAS TRANS X 20', '2004/04/03', 0.5, 49.50, 'EMS S/A', 'Preta', 'Receituario médico', 1 );

INSERT INTO produto (codProduto, nome, descricao, validade, comissao, precoUnitario, fabricante, tarja, tipo, codSessao)

VALUES('894916502900', 'CITRATO DE FENTANILA', '50 MCG/ML SOL INJ CT 25 AMP VD INC X 2 ML', '2017/12/15', 0.2, 35.50,'BioCadBrazil', 'Vermelha','Receituario médico', 5);

INSERT INTO produto (codProduto, nome, descricao, validade, comissao, precoUnitario, fabricante, tarja, tipo, codSessao)

VALUES('960062065395', 'CLONAZEPAM', '2,5 MG/ML SOL OR CT FR GOT VD AMB X 20 ML', '2020/02/28', 0.10, 15.50, 'ThermoFisher', 'Amarela', 'Receituario médico', 1);

INSERT INTO produto (codProduto, nome, descricao, validade, comissao, precoUnitario, fabricante, tarja, tipo, codSessao)

VALUES('345418500012', 'DIAZEPAM', '10 MG COM CT BL AL PLAS TRANS X 10', '2022/10/13', 0.8, 23.99, 'Sincamesp', 'Azul', 'Receituario médico', 1 );

INSERT INTO produto (codProduto, nome, descricao, validade, comissao, precoUnitario, fabricante, tipo, codSessao)

VALUES('096004715308', 'DORMIUM', '1 MG/ML SOL INJ CT 5 AMP VD TRANS X 5 ML', '2019/12/30', 0.1, 200.00, 'FCE Pharma', 'Receituario médico', 3 );

* **Povoando a relação RECEITA**

INSERT INTO receita(classificacao, paciente, prescricao, medico)

VALUES('Simples', 'Tiago Victor Lopes', 'Tratamento por 30 dias/5 ampolas no caso de medicamento injetavél', '1600-3308');

INSERT INTO receita(classificacao, paciente, prescricao, medico)

VALUES('antimicrobianos', 'Camila Lavínia', 'Tomar 1 comprimido por via oral a cada 12Hrs por 7 dias', '9915-0077');

INSERT INTO receita(classificacao, paciente, prescricao, medico)

VALUES('controle especial', 'Henrique Elias Lopes', 'Uso Oral: Tomar 1 comprimido a cada 8Hrs', '6816-6126');

INSERT INTO receita(classificacao, paciente, prescricao, medico)

VALUES('simples', 'Kamilly Cláudia', 'Uso subcutaneo: Aplicar X unidades pela manhã. Uso continuo', '7089-9724');

INSERT INTO receita(classificacao, paciente, prescricao, medico)

VALUES('controle especial', 'Julio Oliver Melo', 'Tomar 1 capsula duas vezes ao dia', '4582-8118');

* **Povoando a relação PRODUTORECEITA**

INSERT INTO produtoReceita(codProduto, numeroReceita) VALUES('894916502900', 5);

INSERT INTO produtoReceita(codProduto, numeroReceita) VALUES('345418500012', 2);

INSERT INTO produtoReceita(codProduto, numeroReceita) VALUES('789491651258', 4);

INSERT INTO produtoReceita(codProduto, numeroReceita) VALUES('096004715308', 1);

INSERT INTO produtoReceita(codProduto, numeroReceita) VALUES('960062065395', 3);

* **Povoando a relação TRABALHAFARMACIA**

INSERT INTO trabalhaFarmacia VALUES('15.721.825/0001-72', '3624044444');

INSERT INTO trabalhaFarmacia VALUES('54.407.897/0001-00', '4281285561');

INSERT INTO trabalhaFarmacia VALUES('33.668.192/0001-68', '7567227710');

INSERT INTO trabalhaFarmacia VALUES('22.682.420/0001-38', '6047297385');

INSERT INTO trabalhaFarmacia VALUES('34.665.298/0001-70', '2460211116');

* **Povoando a relação VENDA**

INSERT INTO venda(data, hora, total, numeroParcelas, tipo, matFuncionario)

VALUES('2018/06/12', '18:44', 315.00 , 5 ,'Cartão de credito','4281285561');

INSERT INTO venda(data, hora, total, tipo, matFuncionario, cpfCliente)

VALUES('2019/03/15', '09:15', 39.99,'Debito','2460211116', '242.783.240-57');

INSERT INTO venda(data, hora, total, tipo, matFuncionario, cpfCliente)

VALUES('2012/04/20', '10:30', 40.00,'Debito','4281285561', '305.093.100-01');

INSERT INTO venda(data, hora, total, tipo, matFuncionario, cpfCliente)

VALUES('2013/10/25', '22:00', 38.50,'Debito','3624044444', '342.476.170-28');

INSERT INTO venda(data, hora, total, tipo, matFuncionario, cpfCliente)

VALUES('2014/06/17', '22:30', 20.00,'Debito','6047297385', '082.730.420-06');

INSERT INTO venda(data, hora, total, numeroParcelas, tipo, matFuncionario)

VALUES('2017/02/07', '15:30', 76.50, 2,'Cartão de credito','3624044444');

INSERT INTO venda(data, hora, total, tipo, matFuncionario, cpfCliente)

VALUES('2016/08/05', '08:20', 350,'Debito','7567227710', '090.460.320-27');

INSERT INTO venda(data, hora, total, tipo, matFuncionario, cpfCliente)

VALUES('2017/09/30', '20:20', 220,'Debito','7567227710', '090.460.320-27');

INSERT INTO venda(data, hora, total, numeroParcelas, tipo, matFuncionario)

VALUES('2015/06/21', '14:59', 243, 5,'Cartão de credito','6047297385');

INSERT INTO venda(data, hora, total, numeroParcelas, tipo, matFuncionario)

VALUES('2019/01/11', '19:44', 100, 1,'Cartão de credito','2460211116');

INSERT INTO venda(data, hora, total, numeroParcelas, tipo, matFuncionario)

VALUES('2018/03/12', '07:15', 685, 3,'Cartão de credito','7567227710');

* **Povoando a relação VENDACREDITO**

INSERT INTO vendaCredito(numero, codVenda) VALUES('5404611081910767', 10);

INSERT INTO vendaCredito(numero, codVenda) VALUES('4485409083091156', 9);

INSERT INTO vendaCredito(numero, codVenda) VALUES('4532530518265413', 6);

INSERT INTO vendaCredito(numero, codVenda) VALUES('5181213573840682', 11);

INSERT INTO vendaCredito(numero, codVenda) VALUES('5067079390440249', 1);

* **Povoando a relação VENDAPRODUTO**

INSERT INTO vendaProduto (codVenda, codProduto, quantidade, precoUnitario) VALUES(2 , '894916502900', 1, 39.99);

INSERT INTO vendaProduto (codVenda, codProduto, quantidade, precoUnitario) VALUES(5 , '345418500012', 6, 23.99 );

INSERT INTO vendaProduto (codVenda, codProduto, quantidade, precoUnitario) VALUES(3, '789491651258', 2, 49.50);

INSERT INTO vendaProduto (codVenda, codProduto, quantidade, precoUnitario) VALUES(4, '096004715308', 2, 200.00);

INSERT INTO vendaProduto (codVenda, codProduto, quantidade, precoUnitario) VALUES(1 , '960062065395', 20, 15.50);

* **Povoando a relação ARMAZENAPRODUTO**

INSERT INTO armazenaProduto VALUES('15.721.825/0001-72', '960062065395', 400, 'Estoque baixo', 1000);

INSERT INTO armazenaProduto VALUES('33.668.192/0001-68', '894916502900', 1550, 'Estoque moderado', 1000);

INSERT INTO armazenaProduto VALUES('22.682.420/0001-38', '096004715308', 210, 'Estoque confortável', 20);

INSERT INTO armazenaProduto VALUES('34.665.298/0001-70', '789491651258', 310, 'Estoque confortável', 50);

INSERT INTO armazenaProduto VALUES('54.407.897/0001-00', '345418500012', 10, 'Estoque baixo', 10);

# 4.1.7. Consultas

* **Listar produtos fora do prazo de validade**

select \*

from produto p

where p.validade < CURRENT\_DATE

* **Listar produtos que se encontram em estoque baixo ou confortável**

SELECT razaosocial, nome, status

FROM produtosemestoque

WHERE status ILIKE 'estoque b%' OR status ILIKE 'estoque c%'

* **Listar produtos que não tem tarja**

SELECT \*

FROM produto p

WHERE p.tarja IS NULL

* **Listar farmácias em que os produtos estão disponíveis**

SELECT razaosocial, endereco , nome

FROM produtosemestoque

* **Listar clientes que realizaram mais de uma compra**

SELECT c.cpf, c.nome, COUNT(\*) AS compras

FROM cliente c, venda v

WHERE c.cpf = v.cpfCliente

GROUP BY c.cpf, c.nome

HAVING COUNT(\*) > 1

* **Listar vendas em que o CPF do cliente não tem o valor nulo**

SELECT \*

FROM venda v

WHERE v.cpfCliente IS NOT NULL

* **Listar o nome dos clientes que começam com a letra J em ordem alfabética**

SELECT \*

FROM funcionario f

WHERE f.nome ILIKE 'j%'

ORDER BY f.nome

* **Listar a matricula e nome dos funcionários que não gerenciam nenhuma sessão**

SELECT f.matricula, f.nome

FROM funcionario f

WHERE f.matricula IN

(SELECT f.matricula

FROM funcionario f

EXCEPT

SELECT f.matricula

FROM funcionario f, sessao s

WHERE f.matricula = s.gerente)

* **Listar a matricula e nome dos funcionários que não realizaram nenhuma venda**

SELECT f.matricula, f.nome

FROM funcionario f

WHERE NOT EXISTS

(SELECT \*

FROM venda v

WHERE v.matfuncionario = f.matricula)

* **Listar a matricula e nome dos funcionários que são gerentes de alguma sessão**

SELECT f.matricula, f.nome

FROM funcionario f

WHERE EXISTS

(SELECT \*

FROM sessao s

WHERE f.matricula = s.gerente)

* **Listar o CPF e o nome dos clientes que não possuem debito com a farmácia**

SELECT c.cpf, c.nome

FROM cliente c

WHERE c.cpf in

((SELECT c.cpf

FROM Cliente c)

EXCEPT

(SELECT c.cpf

FROM Cliente c

WHERE c.debito > 0))

ORDER BY c.nome

* **Listar o status do estoque das farmácias cadastradas**

select p.razaosocial,p.nome, p.status

from produtosEmEstoque p

* **Listar o estoque atual de cada produto alocado na farmácia**

select p.razaosocial,p.nome, p.estoqueatual

from produtosEmEstoque p

* **Listar o salário dos funcionários acrescido do valor de comissão adquirido em cada venda realizada pelo mesmo**

SELECT f.matricula, f.nome , f.salario + ((vp.quantidade \* vp.precounitario) - ((vp.quantidade \* vp.precounitario) \* p.comissao)) AS salario

FROM funcionario f, venda v NATURAL JOIN vendaproduto vp, produto p

WHERE f.matricula = v.matfuncionario AND p.codproduto = vp.codproduto;

* **Listar os produtos e a sessão onde ele está alocado**

SELECT s.nome, p.nome

FROM sessao s, produto p

WHERE s.codSessao = p.codSessao

* **Listar a matricula, nome e a quantidade de vendas realizadas por cada funcionario**

SELECT f.matricula, f.nome, COUNT(\*) AS quantidadeVendas

FROM funcionario f, venda v

WHERE f.matricula = v.matfuncionario

GROUP BY f.matricula, f.nome

ORDER BY f.nome, COUNT(\*)