

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

ALISSON AUGUSTO DE ALMEIDA

DIONES ROCHA GOMES

Cajazeiras-PB

2018

ALISSON AUGUSTO DE ALMEIDA

DIONES ROCHA GOMES

**SISTEMA DE CONTROLE DE UMA MERCEARIA**

Projeto de Banco de Dados apresentado ao professor, Dr. Fabio Gomes de Andrade do curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Ciência, Educação e Tecnologia da Paraíba (IFPB) como requisito parcial para a aprovação da cadeira.

Cajazeiras-PB

2018

Sumário

[1. Introdução 4](#_Toc527060610)

[2. Modelo Conceitual 5](#_Toc527060611)

[3. Dicionário Conceitual de Dados 8](#_Toc527060612)

[4. Modelo Lógico 14](#_Toc527060613)

[4.1 Mapeamento Entidade Relacionamento 14](#_Toc527060614)

[4.2 Dicionário Lógico de Dados 15](#_Toc527060615)

[5 Modelo Físico 26](#_Toc527060616)

[5.1 Scripts SQL 26](#_Toc527060617)

[5.2 Criando as Visões. 43](#_Toc527060618)

[5.3 Scripts SQL para consultas. 44](#_Toc527060619)

[5.4 Criando Procedimentos 48](#_Toc527060620)

[5.5 Criando Gatilhos. 49](#_Toc527060621)

# Introdução

Hoje em dia, com os avanços tecnológicos e o surgimento de vários meios de aprimorar o meio comercial em geral, cresce também a procura de tecnologias que possam proporcionar mais praticidade, agilidade e desempenho, no contexto que a mesma seja inserida. O assunto em questão aqui é o ambiente comercial, um mercado de pequeno porte que deseja automatizar o seu ambiente de vendas, e com o aumento significativo dos clientes e das vendas, tornou-se cada vez mais inviável manter o controle dos negócios.

Portanto, afim de promover um melhor gerenciamento do estabelecimento, consultamos o senhor Francisco da Bodega, dono do estabelecimento em questão, afim de compreendermos os requisitos, que por sua vez, devem ser atendidos nesse tipo de sistema. Vale ressaltar que não é feito nenhum tipo de controle nas informações do estabelecimento, podendo ocasionar um prejuízo significativo nas suas finanças, assim também, como a não persistência dos dados relevantes ao estabelecimento é também um problema em questão.

# Modelo Conceitual

**2.1** Levantamento de Requisitos

A bodega do senhor Francisco possui alguns fornecedores (Pessoa Jurídica), que devemos armazenar alguns dados são eles; telefone, nome, razão social, CNPJ, inscrição estadual e endereço (rua, cidade, bairro, número da residência, UF). O fornecedor por sua vez emite nota fiscal para qualquer produto e armazena a data de quando essa nota foi emitida, nota fiscal deverá ser transportada por uma pessoa, a nota fiscal será destinada para uma pessoa física ou jurídica, a nota fiscal e gerada por uma venda.

Por sua vez devemos armazenar as informações pertinentes aos produtos são elas validade, valor, descrição. Toda venda deverá possui uma lista de produtos que a mesma deverá conter a quantidade daquele referido produto naquela venda.

Devemos armazenar as informações da venda que será a data, valor e o tipo da venda e a mesma após concluída deverá gera um cupom fiscal se a venda for do tipo consumidor final (pessoa física) aquela venda gera um cupom fiscal se a venda for do tipo empresa (pessoa jurídica) ela deverá gera uma nota fiscal e um cupom fiscal deverá conter as informações número do cupom, chave de acesso, valor total, data e hora e por sua vez ele será emitido por um caixa e o cupom fiscal mostrara a forma de pagamento que foi usada para aquela venda e a pessoa(física) que possui aquele cupom fiscal.

No Pagamento armazenamos data e o valor, mas temos três formas de pagamento são elas, cheque, cartão e avista para cheque devemos armazenar número do cheque, para cartão devemos armazenar número de parcelas e para pagamento a vista a data da efetivação do pagamento, e todo o pagamento será realizado por uma pessoa.

Caixa deverá armazenar os dados que são eles valor, data e hora da abertura e ainda o valor, data e hora do fechamento e durante o dia de trabalho senhor Francisco da bodega poderá solicita o sangramento do caixa que o mesmo terá o valor vendido, hora e data no momento do sangramento do caixa e o caixa quem deverá emitir o cupom fiscal, o operador do caixa será funcionário do senhor Francisco.

Senhor Francisco deseja armazena as informações de seus funcionários são elas CPF, salário, CEP, data admissão, terá ao menos um telefone, nome e número da CTPS e o mesmo ainda pode realizar venda de produto e o funcionário deverá opera o caixa.

Como sabemos senhor Francisco está modernizando toda a sua empresa ele não trabalhara mais com produtos sem nota fiscal e será um cidadão correto, toda nota tem destinatário de que poderá ser o senhor Francisco na pessoa de empresa mas também poderá ser cliente pessoa física, que devemos armazenar as informações de telefone, nome, CPF, RG e endereço(rua, cidade, bairro, número da residência, UF) e tanto um pessoa física quanto uma pessoa jurídica realiza pagamento, senhor Francisco está também querendo vender para outras empresas assim o mesmo também emitira nota fiscais.

**2.3** Digrama de Entidade e Relacionamento

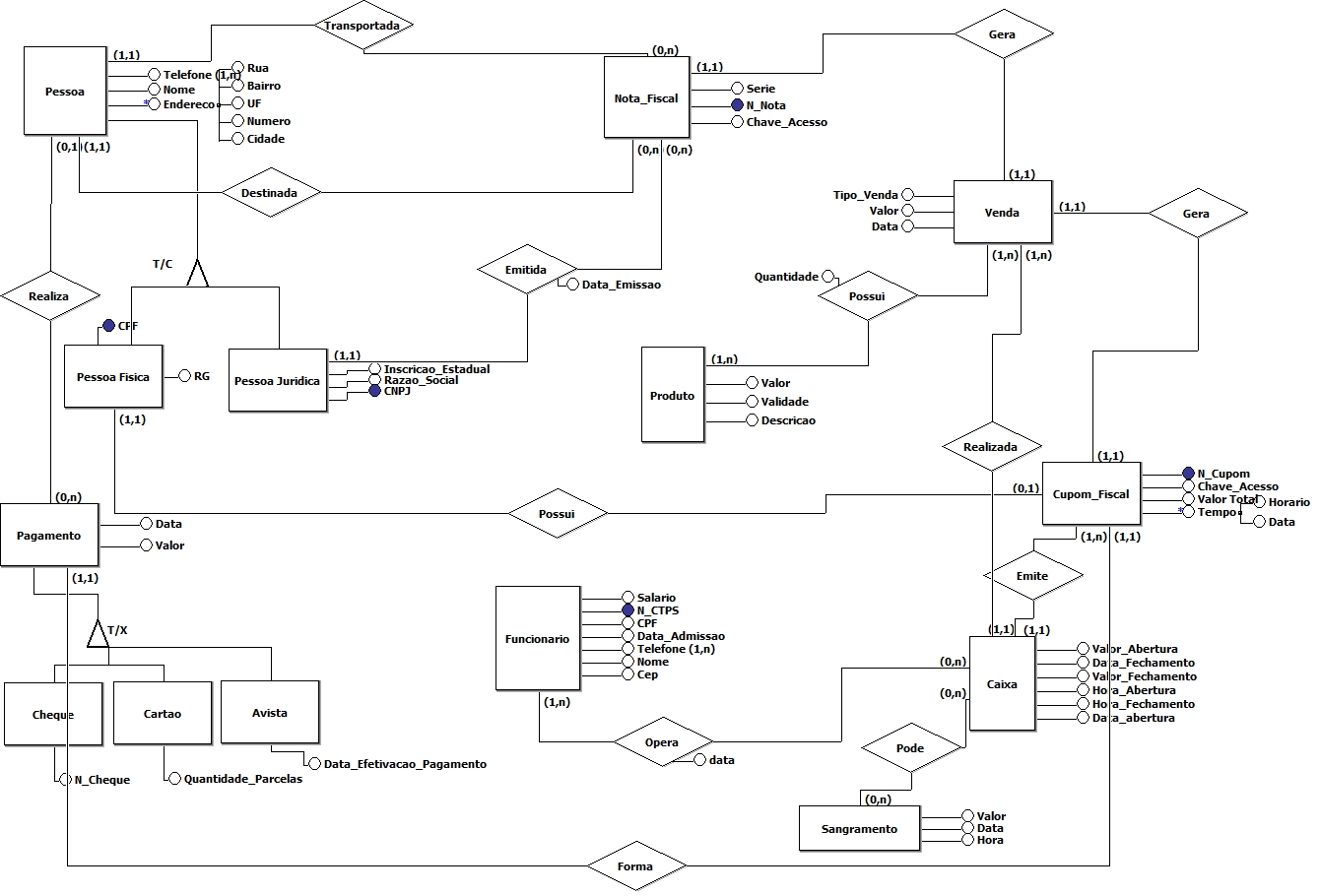


Figura 1: Diagrama de Entidade e Relacionamento

# Dicionário Conceitual de Dados

* **Entidade Pessoa:**
* Generalização das entidades Pessoa Física e Pessoa Jurídica, criada para armazenar as informações referentes a uma pessoa.

**Atributo (s):**

* Telefone: armazena o telefone da pessoa.
* Endereço: armazena o endereço da pessoa que é composto por: rua, bairro, uf, número, cidade.
* Nome: armazena o nome da pessoa.

**Relacionamento (s):**

* Realiza: Relaciona Pessoa e Pagamento. Um pagamento e realizado por no mínimo zero e máximo uma pessoa e uma pessoa realiza no mínimo zero pagamentos e no máximo muitos.
* Transportada: Relaciona Pessoa e Nota Fiscal. Uma pessoa transporta no mínimo zero e no máximo muitas notas fiscais e uma nota fiscal e transportada por no mínimo uma pessoa e no máximo uma pessoa.
* Destinada: Relaciona Pessoa e Nota Fiscal. Uma pessoa será destinatário de no mínimo zero nota e no máximo muitas notas e uma nota fiscal e destinada para no mínimo uma pessoa e no máximo uma pessoa
* **Entidade Pessoa Física:**
* Especialização da entidade Pessoa, criada para armazenar informações referentes ao Cliente.

**Atributo (s):**

* CPF: armazena o Cpf do cliente.
* RG: armazena o Rg do cliente.

**Relacionamento (s):**

* Possui: Relaciona Pessoa Física e Cupom Fiscal. Uma pessoa física possui no mínimo zero cupom e no máximo muitos cupons e um cupom possui no mínimo uma pessoa física e no máximo uma pessoa física.
* **Entidade Pessoa Jurídica:**
* Especialização da entidade Pessoa, criada para armazenar informações referentes a uma Pessoa Jurídica.

**Atributo (s):**

* Inscrição\_Estadual: armazena a inscrição estadual do fornecedor.
* Razão\_Social: armazena a razão social do fornecedor.
* CNPJ: armazena o Cnpj do fornecedor.

**Relacionamento (s):**

* Emitida: Relaciona Pessoa Jurídica e Nota Fiscal. Armazenando a data de emissão da nota fiscal. Uma pessoa jurídica emite no mínimo zero nota e no máximo muitas notas e uma nota emitida por no mínimo uma pessoa jurídica e no máximo uma pessoa jurídica.

* **Entidade Pagamento:**
* Generalização das entidades Cheque, Cartão, avista, criada para armazenar informações referentes ao pagamento.

**Atributo (s):**

* Data: armazena a data do pagamento.
* Valor: armazena o valor do pagamento.

**Relacionamento (s):**

* Realiza: Já descrito anteriormente.
* Forma: Relaciona Cupom fiscal e Pagamento. Cupom fiscal possui no mínimo uma forma de pagamento o no máximo um e pagamento possui no mínimo um cupom fiscal e no máximo um.
* **Entidade Cheque:**
* Especialização da entidade pagamento, criada para armazenar informações referentes ao pagamento com o cheque.

**Atributo (s):**

* NumeroCheque: armazena o número do cheque.
* **Entidade Cartão:**
* Especialização da entidade pagamento, criada para armazenar informações referentes ao pagamento com o cartão.

**Atributo (s):**

* QuantidadeParcelas: armazena a quantidade parcelada no cartão.
* **Entidade Avista:**
* Especialização da entidade pagamento, criada para armazenar as informações referente ao pagamento avista.

**Atributo (s):**

* DataEfetivacaoPagamento: armazena as informações referentes a data do pagamento a vista.
* **Entidade Nota\_Fiscal:**
* Criada para armazenar as informações referentes a emissão de nota fiscal.

**Atributo (s):**

* Serie: armazena as informações de serie da nota fiscal.
* NNota: armazena informações referentes ao número da nota fiscal.
* ChaveDeAcesso: armazena as informações referentes a chave de acesso na nota fiscal.

**Relacionamento (s):**

* Transportada: Já descrito anteriormente.
* Destinada: Já descrito anteriormente.
* Emitida: Já descrito anteriormente.
* Gera: Relaciona Nota Fiscal e Venda. Uma nota e gerada por no mínimo uma venda e no máximo muitas vendas e uma venda gera no mínimo uma nota e no máximo uma nota.
* **Entidade Produto:**
* Criada para armazenar informações sobre o produto.

**Atributo (s):**

* Validade: armazena informações referentes a validade do produto.
* Valor: armazena as informações referentes ao valor do produto.
* Descrição: armazena as informações referentes a descrição do produto.

**Relacionamento (s):**

* Possui: Relaciona Produto e Venda. Uma venda possui no mínimo um produto e no máximo muitos produtos e um produto está em no mínimo uma venda e no máximo muitas e o mesmo devera armazenar a quantidade do produto na venda.
* **Entidade Funcionário:**
* Criada para armazenar informações do funcionário.

**Atributo (s):**

* Salario: armazena informações do salário do funcionário.
* CPF: armazena informações do CPF do funcionário.
* CEP: armazena as informações do CEP do funcionário.
* DataAdmissao: armazena as informações da data de admissão do funcionário.
* Telefone: armazena as informações de telefone do funcionário.
* Nome: armazena as informações do nome do funcionário.
* NCTPS: armazena as informações do número da carteira de trabalho.

**Relacionamento (s):**

* Opera: Relaciona Funcionário e Caixa. Funcionário opera no mínimo zero caixa e no máximo um caixa e caixa e operado por no mínimo um funcionário e no máximo um funcionário.
* **Entidade Vendas:**
* Criada para armazenar informações das vendas.

**Atributo (s):**

* Data: armazena informações da data da venda.
* Valor: armazena as informações referentes ao valor da venda.
* Tipo de Venda: Armazena as informações pertinentes ao tipo de venda.

**Relacionamento (s):**

* Gera: Relaciona Venda e Cupom Fiscal. Uma venda gera no mínimo um cupom e no máximo um cupom e cupom e gerado por no mínimo uma venda e no máximo muitas.
* Gera: Relaciona Nota Fiscal e Venda: Já descrita anteriormente.
* Possui: Já descrita anteriormente.
* Realiza: Relaciona Venda e Caixa. Uma venda realizada no mínimo um e no máximo por um Caixa e o Caixa pode realiza no mínimo uma venda e no máximo muitas.
* **Entidade Cupom Fiscal:**
* Criada para armazenar informações referentes ao cupom fiscal.

**Atributo (s):**

* NCupom: armazena informações do número do cupom fiscal.
* ChavaAcesso: armazena as informações referentes a chave de acesso do cupom fiscal.
* ValorTotal: armazena informações do valor total do cupom fiscal.
* ProtAutorizacao: armazena as informações do protocolo de autorização.
* Tempo: armazena o tempo é composto por: data e horário.

**Relacionamento (s):**

* Emite: Relaciona Cupom Fiscal e Caixa. Cupom e emitido por no mínimo um caixa e no máximo um caixa e um caixa emite no mínimo um cupom fiscal e no máximo muitos cupons.
* Possui: Já descrito anteriormente.
* Gera: Já descrito anteriormente.
* Forma: Já descrito anteriormente.
* **Entidade Caixa:**
* Criada para armazenar informações referentes ao caixa.

**Atributo (s):**

* ValorAbertura: armazena informações de abertura do caixa.
* DataFechamento: armazena as informações da data de fechamento do caixa.
* ValorFechamento: armazena informações do valor do fechamento do caixa.
* HoraAbertura: armazena as informações referentes a hora de abertura do caixa.
* DataAbertura: armazena as informações referentes a data de abertura do caixa.
* HoraFuncionamento: armazena as informações referentes a hora de funcionamento do caixa.

**Relacionamento (s):**

* Pode: Relaciona Caixa e Sangramento. Caixa pode ser sangrado no mínimo zero e no máximo muitas e sangramento do caixa pode ser feito no mínimo zero e no máximo muitas.
* Emite: Já descrito anteriormente.
* Opera: Já descrito anteriormente.
* Realizada: Já descrito anteriormente.
* **Entidade Sangramento:**
* Criada para armazenar informações referentes ao sangramento do caixa.

**Atributo (s):**

* Hora: armazena informações de hora do sangramento do caixa.
* Data: armazena as informações da data do sangramento do caixa.
* Valor: armazena as informações referentes ao valor sangrado do caixa.

**Relacionamento (s):**

* Pode: Já descrito anteriormente.

# Modelo Lógico

# Mapeamento Entidade Relacionamento

**Pessoa** (ID, Nome, Rua, Bairro, UF, Numero, Cidade).

**PessoaTelefone** (IDPessoa, Telefone).

**PessoaFisica** (IDPessoa, CPF, RG).

**PessoaJuridica** (IDPessoa, InscriçãoEstadual, RazaoSocial, Cnpj).

**Pagamento** (ID, Data, Valor, IDPessoa, Ncupom).

**PagamentoCheque** (IDPagamento, NCheque).

**PagamentoCartao** (IDPagamento, numeroParcelas).

**PagamentoAvista** (IDPagamento, dataEfetivacao).

**NotaFiscal** (NNota, Serie, ChaveAcesso, IDPessoa, IDPessoaJuridica, DataEmissao).

**Produto** (ID, Validade, Valor, Descricao).

**Funcionario** (NCTPS, Salario, CPF, Nome, DataAdmissao, CEP).

**FuncionarioTelefone** (NCTPS, Telefone).

**Venda** (ID, Data, TipoVenda, Valor, NNota, NCupom).

**CupomFiscal** (NCupom, ChaveAcesso, ValorTotal, Data, Hora, IDPessoaFisica, IDCaixa).

**Caixa** (ID, ValorAbertura, ValorFechamento, DataAbertura, DataFechamento, HoraAbertura, HoraFechamento, NCTPS).

**Sangramento** (Valor, Data, Hora).

**VendaProduto** (IDProduto, IDVenda, Quantidade).

**SangramentoCaixa**(IDCaixa,Valor,Data).

# Dicionário Lógico de Dados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pessoa: Relação que armazena os dados gerais dos funcionários e clientes | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| ID | Representa o número de identificação do cliente ou do funcionário | SERIAL | Números inteiros positivos. | Chave Substituta.  Chave Primária. |
| Nome | Representa o nome de um cliente ou funcionário. | String(100) | String(100) | Não nulo. |
| Rua | Representa o nome da rua onde o  funcionário ou cliente mora. | String(100) | String(100) | Não nulo. |
| Bairro | Representa o nome do bairro onde o funcionário ou o cliente mora. | String(100) | String(100) | Não nulo. |
| UF | Representa o nome do estado onde o funcionário ou o cliente está situado. | String(50) | String(50) | Não nulo. |
| Numero | Representa o número da residência do funcionário ou cliente. | Int | Números inteiros positivos. | Não nulo. |
| Cidade | Representa o nome da cidade onde o funcionário ou o cliente mora. | String(100) | String(100) | Não nulo. |

Tabela 1 – Relação PESSOA.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PessoaTelefone: Relação que armazena os telefones dos clientes e dos funcionários | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| IDPessoa | Representa o número de identificação da pessoa cujo o telefone pertence. | SERIAL | Números inteiros positivos. | Chave primária.  Chave estrangeira. |
| Telefone | Representa os números de telefone dos funcionários ou dos clientes. | String(20) | String(20) | Chave primária. |

Tabela 2 – Relação PESSOATELEFONE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PessoaFísica: Relação que armazena os dados referentes a pessoa física | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| IDPessoa | Representa o número de identificação da pessoa...  (Alisson) | SERIAL | Números inteiros positivos. | Não nulo.  Chave estrangeira. |
| CPF | Representa o CPF de um funcionário ou cliente. | String(20) | String(20) | Não nulo. |
| RG | Representa o RG de um cliente ou funcionário. | String(50) | String(50) | Não nulo. |

Tabela 3 - Relação PESSOAFÍSICA.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PessoaJurídica: Relação que armazena os dados referentes a uma pessoa jurídica | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| IDPessoa | Representa o número de identificação da pessoa...  (Alisson) | SERIAL | Números inteiros positivos. | Não nulo.  Chave estrangeira. |
| InscriçãoEstadual | Representa a inscrição Estadual da pessoa jurídica. | String(50) | String(50) | Não nulo. |
| RazãoSocial | Nome do dono | String(50) | String(50) | Não nulo. |
| CNPJ | Representa o CNPJ da pessoa física | String(50) | String(50) | Não nulo. |
|  |  |  |  |  |

Tabela 4 - Relação PESSOAJURÍDICA.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pagamento: Relação que armazena os dados referentes aos pagamentos | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| ID | Representa o número de identificação do pagamento. | SERIAL | Números inteiros positivos. | Chave estrangeira. |
| Data | Representa a data do pagamento. | Data | Date | Não nulo. |
| Valor | Representa o valor do pagamento. | Real | Números reais positivos. | Não nulo. |
| IDPessoa | Representa o número de identificação da pessoa que realiza o pagamento. | SERIAL | Números inteiros positivos. | Não nulo  Chave estrangeira. |
| Ncupom | Representa o número do cupom | SERIAL | Números inteiros positivos. | Não nulo. |

Tabela 5 - Relação PAGAMENTO.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PagamentoCheque: Relação que armazena os dados referentes ao pagamento usando cheque | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| IDPagamento | Representa o número de identificação do pagamento. | SERIAL | Números inteiros positivos. | Chave primária. |
| NCheque | Representa o número do cheque para o pagamento. | INT | Números inteiros positivos. | Não nulo. |

Tabela 6 Relação - PAGAMENTOCHEQUE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PagamentoCartão: Relação que armazena os dados referentes ao pagamento usando cartão | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| IDPagamento | Representa o número de identificação do pagamento. | SERIAL | Números inteiros positivos. | Chave primária. |
| NumeroParcelas | Representa o número de parcelas do pagamento usando cartão. | Int | Números inteiros positivos. | Não nulo. |

Tabela 7 - Relação PAGAMENTOCARTÃO.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PagamentoAvista: Relação que armazena os dados referentes ao pagamento a vista | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| IDPagamento | Representa o número de identificação do pagamento. | SERIAL | Números inteiros positivos. | Chave primária. |
| DataEfetivação | Representa a data da efetivação do pagamento a vista. | Date | Data | Não nulo. |

Tabela 8 - Relação PAGAMENTOAVISTA.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NotaFiscal: Relação que armazena os dados referentes a emissão da nota fiscal. | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| NNota | Representa a número da nota fiscal. | SERIAL | Números inteiros positivos. | Chave primária. |
| Serie | Representa o número de série da nota fiscal. | INT | Números inteiros positivos. | Não nulo. |
| ChaveAcesso | Representa a chave de acesso da nota fiscal. | String(44) | String(44) | Não nulo. |
| CodigoBarras | Representa o número do código de barras da nota fiscal. | ? | ? | Não nulo. |
| IDPessoa | Representa o número de identificação da pessoa que emitira a nota fiscal. | SERIAL | Números inteiros positivos. | Chave estrangeira. |
| IDPessoaJuridica | Representa o número de identificação referente a uma pessoa jurídica. | SERIAL | Números inteiros positivos. | Chave estrangeira. |
| DataEmissão | Representa a data da emissão da nota fiscal. | Date | Date | Não nulo. |

Tabela 9 - Relação NOTAFISCAL.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Produto: Relação que armazena os dados referentes os produtos | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| ID | Representa o número de identificação do produto. | SERIAL | Números inteiros positivos. | Chave primária. |
| Validade | Representa a data de validade do produto. | Date | Date | Não nulo. |
| Valor | Representa o valor do produto | Real | Números reais positivos. | Não nulo. |
| Descrição | Representa a descrição do produto. | String(100) | String(100) | Não nulo. |

Tabela 10 - Relação PRODUTO.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Funcionário: Relação que armazena dos dados referentes aos funcionários | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| NCTPS | Representa o número da carteira de trabalho. | SERIAL | Números inteiros positivos. | Chave primária. |
| Salário | Representa o valor do salário do funcionário. | Real | Números reais positivos. | Não nulo. |
| CPF | Representa o número do CPF do funcionário. | String(20) | String(20) | Não nulo. |
| Nome | Representa o nome do funcionário. | String(100) | String(100) | Não nulo. |
| DataAdmissão | Representa a data que o funcionário foi contratado. | Data | Data | Não nulo. |
| CEP | Representa o CEP do funcionário. | String(20) | String(20) | Não nulo. |

Tabela 11- Relação FUNCIONÁRIO.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FuncionárioTelefone: Relação que armazena os números dos telefones dos funcionários. | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| NCPTS | Representa o número da carteira de trabalho. | SERIAL | Números inteiros positivos. | Chave primária.  Chave estrangeira. |
| Telefone | Representa os números dos telefones dos funcionários. | String(50) | String(50) | Chave primária. |

Tabela 12 - Relação FUNCIONÁRIOTELEFONE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vendas: Relação que armazena os dados referentes das vendas | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| ID | Representa o número de identificação da venda. | SERIAL | Números inteiros positivos. | Chave primeira. |
| Data | Representa a data das vendas realizadas. | Data | Data | Não nulo. |
| Valor | Presenta o valor da venda realizada. | Real | Números reais positivos. | Não nulo. |
| TipoVenda | Representa o tipo da venda. | String(100) | String(100) | Não nulo. |
| NNota | Representa o número da nota fiscal emitida na venda. | SERIAL | Números inteiros positivos. | Não nulo.  Chave estrangeira. |
| Ncupom | Representa o número do cupom fiscal emitido na venda. | SERIAL | Números inteiros positivos. | Não nulo.  Chave estrangeira. |

Tabela 13 - Relação VENDA.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CupomFiscal: Relação que armazena os dados referentes aos cupons fiscais. | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| Ncupom | Representa o número do cupom fiscal. | SERIAL | Números inteiros positivos. | Chave primária. |
| ChaveAcesso | Representa a chave de acesso do cupom fiscal. | String(44) | String(44) | Não nulo |
| ValorTotal | Representa o valor total no cupom fiscal | Real | Números reais positivos. | Não nulo. |
| Data | Representa a data do cupom fiscal. | Data | Data | Não nulo. |
| Hora | Representa a hora que foi emita o cupom fiscal. | Time | Time | Não nulo. |
| IDPessoaFisica | Representa o número de identificação da pessoa física. | SERIAL | Números inteiros positivos. | Não nulo.  Chave estrangeira. |
| IDCaixa | Representa o número de identificação do caixa. | SERIAL | Números inteiros positivos. | Não nulo.  Chave estrangeira. |

Tabela 14 - Relação CUPOMFISCAL.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caixa: Relação que armazena os dados referentes ao caixa | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| ID | Representa o número de identificação do caixa. | SERIAL | Números inteiros positivos. | Chave primária. |
| ValorAbertura | Representa o valor inicial quando o caixa foi aberto. | Real | Números reais positivos. | Não nulo. |
| ValorFechamento | Representa o valor após o caixa ser fechado. | Real | Números reais positivos. | Não nulo. |
| DataAbertura | Representa a data da abertura do caixa. | Data | Data | Não nulo. |
| DataFechamento | Representa a data do fechamento do caixa. | Data | Data | Não nulo. |
| HoraAbertura | Representa a hora da abertura do caixa. | Time | Time | Não nulo. |
| HoraFechamento | Representa a hora do fechamento do caixa. | Time | Time | Não nulo. |
| NCPTS | Representa o número da carteira de trabalho. | SERIAL | Números inteiros positivos. | Não nulo.  Chave estrangeira. |

Tabela 15 - Relação CAIXA.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sangramento: Relação que armazena os dados referentes ao sangramento | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| Valor | Representa o valor do sangramento no caixa. | Real | Números reais positivos. | Não nulo.  UNIQUE. |
| Data | Representa a data em que o caixa foi sangrado | Date | Data | Não nulo. |
| Hora | Representa a hora em que o caixa foi sangrado. | Time | Time | Não nulo. |

Tabela 16 - Relação SANGRAMENTO.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VendaProduto: Relação que armazena os dados referentes as vendas dos produtos | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| IDProduto | Representa o número de identificação dos produtos vendidos. | SERIAL | Números inteiros positivos. | Chave primária. |
| IDVenda | Representa o número de identificação das vendas realizadas | SERIAL | Números inteiros positivos. | Chave primaria  Chave Estrangeira. |
| Quantidade | Representa a quantidade de produtos vendidos. | Int | Números inteiros positivos. | Não nulo. |

Tabela 17 - Relação VENDAPRODUTO.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SangramentoCaixa: Relação que armazena dos dados sobre o sangramento do caixa | | | | |
| Atributo | **Descrição** | **Tipo** | **Domínio** | **Restrição** |
| IDCaixa | Presenta o número de identificação do caixa | SERIAL | Números inteiros positivos. | Chave primária |
| Valor | Representa o valor sangrado no caixa. | Real | Números reais positivos. | Chave Estrangeira.  Não nulo. |
| Data | Representa a data onde o caixa foi sangrado. | Date | Date | Não nulo.  Chave estrangeira. |

Tabela 18- Relação SANGRAMENTOCAIXA.

# Modelo Físico

# Scripts SQL

**Criando a relação Pessoa.**

CREATE TABLE Pessoa (

ID SERIAL,

Nome VARCHAR(100) NOT NULL,

Rua VARCHAR(100) NOT NULL,

Bairro VARCHAR(100) NOT NULL,

UF VARCHAR(50) NOT NULL,

Numero INT NOT NULL,

Cidade VARCHAR(100) NOT NULL,

PRIMARY KEY(ID)

)

**;**

**Criando a relação PessoaTelefone.**

CREATE TABLE PessoaTelefone(

IDPessoa SERIAL,

Telefone VARCHAR(20),

PRIMARY KEY(IDPessoa,Telefone),

FOREIGN KEY(IDPessoa) REFERENCES Pessoa(ID)

)

;

**Criando a relação PessoaFísica.**

CREATE TABLE PessoaFisica(

IDPessoa SERIAL,

CPF VARCHAR(20) NOT NULL,

RG VARCHAR(50) NOT NULL,

PRIMARY KEY(IDPessoa),

FOREIGN KEY(IDPessoa) REFERENCES Pessoa(ID)

)

;

**Criando a relação PessoaJuridica.**

CREATE TABLE PessoaJuridica(

IDPessoa SERIAL,

InscricaoEsdatual VARCHAR(50) NOT NULL,

RazaoSocial VARCHAR(50) NOT NULL,

CNPJ VARCHAR(50) NOT NULL,

PRIMARY KEY(IDPessoa),

FOREIGN KEY(IDPessoa) REFERENCES Pessoa(ID)

)

;

**Criando a relação Funcionário.**

NCTPS SERIAL,

Salario REAL NOT NULL,

CPF VARCHAR(50) NOT NULL,

Nome VARCHAR(100) NOT NULL,

DataAdmissao Date NOT NULL,

CEP VARCHAR(20) NOT NULL,

PRIMARY KEY(NCTPS)

)

;

**Criando a relação FuncionarioTelefone**

NCTPS SERIAL,

Telefone VARCHAR(50) NOT NULL,

PRIMARY KEY(NCTPS, Telefone),

FOREIGN KEY(NCTPS) REFERENCES Funcionario(NCTPS)

)

;

**Criando a relação Caixa.**

ID SERIAL,

ValorAbertura REAL NOT NULL,

ValorFechamento REAL NOT NULL,

DataAbertura DATE NOT NULL,

DataFechamento DATE NOT NULL,

HoraAbertura TIME NOT NULL,

HoraFechamento TIME NOT NULL,

NCTPS SERIAL NOT NULL,

PRIMARY KEY(ID),

FOREIGN KEY(NCTPS) REFERENCES Funcionario(NCTPS)

)

;

**Criando a relação CupomFiscal.**

NCupom SERIAL NOT NULL,

ChaveAcesso VARCHAR(44) NOT NULL,

ValorTotal REAL NOT NULL,

Data DATE NOT NULL,

Hora TIME NOT NULL,

IDPessoaFisica SERIAL NOT NULL,

IDCaixa SERIAL NOT NULL,

PRIMARY KEY(NCupom),

FOREIGN KEY(IDPessoaFisica) REFERENCES PessoaFisica(IDPessoa),

FOREIGN KEY(IDCaixa) REFERENCES Caixa(ID)

)

;

**Criando a relação Pagamento.**

ID SERIAL,

Data DATE NOT NULL,

Valor REAL NOT NULL,

IDPessoa SERIAL NOT NULL,

NCupom SERIAL,

PRIMARY KEY(ID),

FOREIGN KEY(IDPessoa) REFERENCES Pessoa(ID),

FOREIGN KEY(NCupom) REFERENCES CupomFiscal(NCupom)

)

;

**Criando a relação PagamentoCheque.**

IDPagamento SERIAL,

NCheque INT NOT NULL,

PRIMARY KEY(IDPagamento),

FOREIGN KEY(IDPagamento) REFERENCES Pagamento(ID)

);

**Criando a relação PagamentoCartão.**

IDPagamento SERIAL,

NumeroParcelas INT NOT NULL,

PRIMARY KEY(IDPagamento),

FOREIGN KEY(IDPagamento) REFERENCES Pagamento(ID)

)

;

**Criando a relação PagamentoAvista.**

IDPagamento SERIAL,

DataEfetivacao DATE NOT NULL,

PRIMARY KEY(IDPagamento),

FOREIGN KEY(IDPagamento) REFERENCES Pagamento(ID)

);

**Criando a relação NotaFiscal.**

NNota SERIAL,

Serie INT NOT NULL,

ChaveAcesso VARCHAR(44) NOT NULL,

IDPessoa SERIAL,

IDPessoaJuridica SERIAL,

DataEmissao DATE NOT NULL,

PRIMARY KEY(NNota),

FOREIGN KEY(IDPessoa) REFERENCES Pessoa(ID),

FOREIGN KEY(IDPessoaJuridica) REFERENCES PessoaJuridica(IDPessoa)

)

;

**Criando a relação Produto.**

ID SERIAL,

Validade Date NOT NULL,

Valor REAL NOT NULL,

Descricao VARCHAR(100) NOT NULL,

PRIMARY KEY(ID)

)

;

**Criando a relação Venda.**

ID SERIAL,

Data DATE NOT NULL,

TipoVenda VARCHAR(100) NOT NULL,

Valor REAL NOT NULL,

NNota SERIAL,

NCupom SERIAL,

PRIMARY KEY(ID),

FOREIGN KEY(NNota) REFERENCES NotaFiscal(NNota),

FOREIGN KEY(NCupom) REFERENCES CupomFiscal(NCupom)

)

;

**Criando a relação Sangramento.**

Valor REAL UNIQUE NOT NULL,

Data DATE UNIQUE NOT NULL,

Hora TIME NOT NULL,

PRIMARY KEY(Valor,Data)

)

;

**Criando relação VendaProduto.**

IDProduto SERIAL,

IDVenda SERIAL,

Quantidade INT NOT NULL,

PRIMARY KEY(IDProduto, IDVenda),

FOREIGN KEY(IDProduto) REFERENCES Produto(ID),

FOREIGN KEY(IDVenda) REFERENCES Venda(ID)

)

;

**Criando a relação SangramentoCaixa.**

IDCaixa SERIAL,

Valor REAL,

Data DATE NOT NULL,

PRIMARY KEY(IDCaixa, Valor, Data),

FOREIGN KEY(IDCaixa) REFERENCES Caixa(ID),

FOREIGN KEY(Valor) REFERENCES Sangramento(Valor),

FOREIGN KEY(Data) REFERENCES Sangramento(Data)

)

;

**Povoando a relação Pessoa.**

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(1, 'Jesica', 'José Marrocos', 'Salesianos', 'Ceará', 267, 'Juazeiro do Norte');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(2, 'David', 'Marechal Floriano', 'Triangulo', 'Ceará', 45, 'Juazeiro do Norte');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(3, 'Thiago', 'Leão Sampaio', 'Triangulo', 'Ceará', 70, 'Juazeiro do Norte');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(4, 'Ana', 'Macedo Cabral', 'Franciscanos', 'Ceará', 69, 'Juazeiro do Norte');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(5, 'Marcos', 'Senhor Martins', 'João Cabral', 'Ceará', 12, 'Juazeiro do Norte');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(6, 'Joana', 'Marechal Floriano', 'Triangulo', 'Ceará', 456, 'Juazeiro do Norte');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(7, 'Shopping da Segurança', 'Marechal Floriano', 'Triangulo', 'Ceará', 1023, 'Juazeiro do Norte');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(8, 'DTS Distribuidaora', 'Leão Sampaio', 'Triangulo', 'Ceará', 1052, 'Juazeiro do Norte');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(9, 'DSA Distribuidora', 'Gastão de Medeiros Forte', 'Bela Vista', 'Paraiba', 1243, 'Sousa');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(10, 'Rio Piranhas Distribuidora', 'Gastão de Medeiros Forte', 'Sorrilandia II', 'Paraiba', 1098, 'Sousa');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(11, 'Distribuidora Pau Brasil', 'Gastão de Medeiros Forte', 'Bela Vista', 'Paraiba', 1245, 'Sousa');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(12, 'DSG Distribuidora', 'Rua Projetada', 'Andre Gadelha', 'Paraiba', 12, 'Sousa');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(13, 'ISIS', 'Rua Projetada', 'Varzea da Cruz', 'Paraiba', 2, 'Sousa');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(14, 'JOAO DA BUDEGA', 'Rua Projetada', 'Varzea da Cruz', 'Paraiba', 90, 'Sousa');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(15, 'Paulo', 'José Marrocos', 'Salesianos', 'Ceará', 360, 'Juazeiro do Norte');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(16, 'Maria', 'Senhora Santana', 'Salesianos', 'Ceará', 80, 'Juazeiro do Norte');

INSERT INTO Pessoa(ID, Nome, Rua, Bairro, UF, Numero, Cidade) VALUES(17, 'Matheus', 'São Paulo', 'Franciscanos', 'Ceará', 54, 'Juazeiro do Norte');

**Povoando a relação PessoaTelefone.**

INSERT INTO PessoaTelefone VALUES(1, '(88)842-978319');

INSERT INTO PessoaTelefone VALUES(1, '(88)714-257705');

INSERT INTO PessoaTelefone VALUES(2, '(88)243-769549');

INSERT INTO PessoaTelefone VALUES(3, '(88)785-972804');

INSERT INTO PessoaTelefone VALUES(3, '(88)572-102230');

INSERT INTO PessoaTelefone VALUES(4, '(88)533-809854');

INSERT INTO PessoaTelefone VALUES(5, '(88)616-113394');

INSERT INTO PessoaTelefone VALUES(6, '(88)513-829454');

INSERT INTO PessoaTelefone VALUES(7, '(88)617-113454');

INSERT INTO PessoaTelefone VALUES(8, '(88)133-846851');

INSERT INTO PessoaTelefone VALUES(9, '(88)636-127299');

INSERT INTO PessoaTelefone VALUES(10, '(83)896-127299');

INSERT INTO PessoaTelefone VALUES(11, '(83)832-127229');

INSERT INTO PessoaTelefone VALUES(12, '(83)537-127299');

INSERT INTO PessoaTelefone VALUES(13, '(83)432-127299');

INSERT INTO PessoaTelefone VALUES(14, '(83)132-127299');

**Povoando a relação Pessoa Física.**

INSERT INTO PessoaFisica VALUES(1, '149.595.603-20', '184205657');

INSERT INTO PessoaFisica VALUES(2, '039.299.573-53', '183391937');

INSERT INTO PessoaFisica VALUES(3, '768.685.843-00', '177363447');

INSERT INTO PessoaFisica VALUES(4, '909.896.243-23', '422366596');

INSERT INTO PessoaFisica VALUES(5, '190.779.353-40', '186110303');

INSERT INTO PessoaFisica VALUES(6, '339.299.573-51', '183491731');

INSERT INTO PessoaFisica VALUES(7, '668.685.843-12', '147353442');

INSERT INTO PessoaFisica VALUES(8, '809.896.243-25', '132365596');

INSERT INTO PessoaFisica VALUES(9, '290.779.353-48', '586316303');

**Povoando a relação Pessoa Jurídica.**

INSERT INTO PessoaJuridica (idpessoa, inscricaoesdatual, razaosocial, cnpj)VALUES(10,'66888556-4', 'Mega Import', '87.027.742/0001-99');

INSERT INTO PessoaJuridica (idpessoa, inscricaoesdatual, razaosocial, cnpj)VALUES(11,'31686725-0', 'Intelbras', '75.889.884/0001-86');

INSERT INTO PessoaJuridica (idpessoa, inscricaoesdatual, razaosocial, cnpj)VALUES(12,'77408403-0', 'DSA Alimentos', '67.516.156/0001-09');

INSERT INTO PessoaJuridica (idpessoa, inscricaoesdatual, razaosocial, cnpj)VALUES(13,'25686071-8', 'Rio Piranhas', '39.430.149/0001-46');

INSERT INTO PessoaJuridica (idpessoa, inscricaoesdatual, razaosocial, cnpj)VALUES(14,'02334539-0', 'Pau Brasil Bebidas', '69.020.061/0001-17');

INSERT INTO PessoaJuridica (idpessoa, inscricaoesdatual, razaosocial, cnpj)VALUES(15,'63429428-8', 'Vo Ita Frios', '76.210.487/0001-07');

INSERT INTO PessoaJuridica (idpessoa, inscricaoesdatual, razaosocial, cnpj)VALUES(16,'71986138-1', 'Laticionio Belo Vale', '94.541.261/0001-62');

INSERT INTO PessoaJuridica (idpessoa, inscricaoesdatual, razaosocial, cnpj)VALUES(17,'09485138-1', 'A Budega', '24.341.431/0001-23');

**Povoando a relação funcionário.**

INSERT INTO Funcionario(NCTPS, Salario, CPF, Nome, DataAdmissao, CEP)VALUES(1, 1000, '925.376.433-30', 'Santhiago', '2018-09-01', '63043-200');

INSERT INTO Funcionario(NCTPS, Salario, CPF, Nome, DataAdmissao, CEP)VALUES(2, 1000, '926.343.432-10', 'Paulo César', '2018-02-12', '63043-200');

INSERT INTO Funcionario(NCTPS, Salario, CPF, Nome, DataAdmissao, CEP)VALUES(3, 1500, '245.276.533-50', 'Diego', '2018-02-04', '63043-200');

INSERT INTO Funcionario(NCTPS, Salario, CPF, Nome, DataAdmissao, CEP)VALUES(4, 1300, '925.376.423-20', 'Antônio Marcos', '2018-04-08', '63043-200');

INSERT INTO Funcionario(NCTPS, Salario, CPF, Nome, DataAdmissao, CEP)VALUES(5, 1100, '825.636.153-40', 'Efraim', '2018-09-01', '63043-200');

**Povoando a relação Funcionário Telefone.**

INSERT INTO FuncionarioTelefone VALUES(1, '(88)927-72-8768');

INSERT INTO FuncionarioTelefone VALUES(1, '(88)803-41-6639');

INSERT INTO FuncionarioTelefone VALUES(2, '(88)252-78-5944');

INSERT INTO FuncionarioTelefone VALUES(2, '(88)941-71-3680');

INSERT INTO FuncionarioTelefone VALUES(3, '(88)994-23-5908');

INSERT INTO FuncionarioTelefone VALUES(4, '(88)613-98-2486');

INSERT INTO FuncionarioTelefone VALUES(5, '(88)662-74-3102');

INSERT INTO FuncionarioTelefone VALUES(5, '(88)412-94-6447');

INSERT INTO FuncionarioTelefone VALUES(3, '(88)234-62-3861');

**Povoando a relação Caixa.**

INSERT INTO Caixa (ID, valorabertura, valorfechamento, dataabertura, datafechamento, horaabertura, horafechamento, nctps)VALUES(1, 45, 1900, '2018-08-12', '2018-08-12', '06:30:00.00', '19:00:00.00', 1);

INSERT INTO Caixa (ID, valorabertura, valorfechamento, dataabertura, datafechamento, horaabertura, horafechamento, nctps)VALUES(2, 42, 1800, '2018-08-06', '2018-08-06', '06:30:00.00', '19:00:00.00', 2);

INSERT INTO Caixa (ID, valorabertura, valorfechamento, dataabertura, datafechamento, horaabertura, horafechamento, nctps)VALUES(3, 90, 6000, '2018-06-03', '2018-06-03', '06:30:00.00', '19:00:00.00', 3);

INSERT INTO Caixa (ID, valorabertura, valorfechamento, dataabertura, datafechamento, horaabertura, horafechamento, nctps)VALUES(4, 34, 1300, '2018-04-16', '2018-04-16', '06:30:00.00', '19:00:00.00', 4);

INSERT INTO Caixa (ID, valorabertura, valorfechamento, dataabertura, datafechamento, horaabertura, horafechamento, nctps)VALUES(5, 54, 3421, '2016-05-19', '2016-05-19', '06:30:00.00', '19:00:00.00', 5);

**Povoando a relação Cupomfiscal.**

INSERT INTO CupomFiscal (NCupom, chaveacesso, valortotal, data, hora, idpessoafisica,idcaixa)VALUES(1, '9485 2248 2496 7242 3358 8898 2639', 280, '2018-08-06', '14:12:00.00', 2, 1);

INSERT INTO CupomFiscal (NCupom, chaveacesso, valortotal, data, hora, idpessoafisica,idcaixa)VALUES(2, '7710 7740 7452 4463 7112 2966 7774', 432, '2018-08-07', '08:23:00.00', 5, 2);

INSERT INTO CupomFiscal (NCupom, chaveacesso, valortotal, data, hora, idpessoafisica,idcaixa)VALUES(3, '3781 7297 1131 4814 1213 4874 8820', 342, '2018-08-08', '09:59:08.00', 7, 3);

INSERT INTO CupomFiscal (NCupom, chaveacesso, valortotal, data, hora, idpessoafisica,idcaixa)VALUES(4, '2445 2924 4997 5340 9423 8957 4149', 34, '2018-08-09', '07:43:09.00', 9, 5);

INSERT INTO CupomFiscal (NCupom, chaveacesso, valortotal, data, hora, idpessoafisica,idcaixa)VALUES(5, '7512 9751 5648 7928 2918 4057 5889', 24, '2018-08-10', '13:32:00.00', 3, 4);

INSERT INTO CupomFiscal (NCupom, chaveacesso, valortotal, data, hora, idpessoafisica,idcaixa)VALUES(6, '9524 2037 9698 1817 1940 1569 3588', 120, '2018-08-11', '16:24:34.00', 8, 4);

INSERT INTO CupomFiscal (NCupom, chaveacesso, valortotal, data, hora, idpessoafisica,idcaixa)VALUES(7, '4182 7683 4529 8230 2149 5736 7853', 405, '2018-08-12', '18:23:12.00', 6, 5);

INSERT INTO CupomFiscal (NCupom, chaveacesso, valortotal, data, hora, idpessoafisica,idcaixa)VALUES(8, '1790 8048 3187 1882 9859 7877 4484', 298, '2018-08-13', '18:59:45.00', 2, 2);

INSERT INTO CupomFiscal (NCupom, chaveacesso, valortotal, data, hora, idpessoafisica,idcaixa)VALUES(9, '1790 8048 3187 1882 9859 7877 4484', 298, '2018-08-13', '18:59:45.00', 2, 2);

**Povoando a relação Pagamento.**

INSERT INTO Pagamento(ID, data, valor, idpessoa, ncupom)VALUES(1,'2018-09-17', 280, 2, 1);

INSERT INTO Pagamento(ID, data, valor, idpessoa, ncupom)VALUES(2, '2018-02-18', 432, 3, 2);

INSERT INTO Pagamento(ID, data, valor, idpessoa, ncupom)VALUES(3, '2018-01-12', 342, 6, 3);

INSERT INTO Pagamento(ID, data, valor, idpessoa, ncupom)VALUES(4, '2017-09-23', 34, 7, 5);

INSERT INTO Pagamento(ID, data, valor, idpessoa, ncupom)VALUES(5, '2016-03-17', 24, 1, 4);

INSERT INTO Pagamento(ID, data, valor, idpessoa, ncupom)VALUES(6, '2017-09-19', 120, 4, 7);

INSERT INTO Pagamento(ID, data, valor, idpessoa, ncupom)VALUES(7, '2018-05-04', 405, 8, 6);

INSERT INTO Pagamento(ID, data, valor, idpessoa, ncupom)VALUES(8, '2018-07-10', 298, 5, 8);

INSERT INTO Pagamento(ID, data, valor, idpessoa, ncupom)VALUES(9,'2018-09-17', 324, 2, 9);

**Povoando a relação Pagamento Cheque**.

INSERT INTO PagamentoCheque(idpagamento, ncheque)Values(2, 002);

INSERT INTO PagamentoCheque(idpagamento, ncheque)Values(1, 003);

INSERT INTO PagamentoCheque(idpagamento, ncheque)Values(7, 004);

**Povoando a relação Pagamento Cartão.**

INSERT INTO PagamentoCartao(idpagamento, numeroparcelas)Values(3, 4);

INSERT INTO PagamentoCartao(idpagamento, numeroparcelas)Values(4, 3);

INSERT INTO PagamentoCartao(idpagamento, numeroparcelas)Values(8, 4);

**Povoando a relação Pagamento Avista.**

INSERT INTO PagamentoAvista(idpagamento, dataefetivacao)VALUES(5,'2017-09-24');

INSERT INTO PagamentoAvista(idpagamento, dataefetivacao)VALUES(6,'2016-03-19');

INSERT INTO PagamentoAvista(idpagamento, dataefetivacao)VALUES(8,'2016-03-19');

**Povoando a relação NotaFiscal.**

INSERT INTO NotaFiscal(NNota, serie, chaveacesso, idpessoa, idpessoajuridica, dataemissao)VALUES(1, 1,'9485 2248 2496 7242 3358 8898 2639', 1, 10,'2018-09-17');

INSERT INTO NotaFiscal(NNota, serie, chaveacesso, idpessoa, idpessoajuridica, dataemissao)VALUES(2, 2,'7710 7740 7452 4463 7112 2966 7774', 2, 11,'2018-09-02');

INSERT INTO NotaFiscal(NNota, serie, chaveacesso, idpessoa, idpessoajuridica, dataemissao)VALUES(3, 1,'3781 7297 1131 4814 1213 4874 8820', 3, 12,'2018-09-05');

INSERT INTO NotaFiscal(NNota, serie, chaveacesso, idpessoa, idpessoajuridica, dataemissao)VALUES(4, 1,'2445 2924 4997 5340 9423 8957 4149', 1, 13,'2018-09-09');

INSERT INTO NotaFiscal(NNota, serie, chaveacesso, idpessoa, idpessoajuridica, dataemissao)VALUES(5, 1,'7512 9751 5648 7928 2918 4057 5889', 3, 14,'2018-09-08');

INSERT INTO NotaFiscal(NNota, serie, chaveacesso, idpessoa, idpessoajuridica, dataemissao)VALUES(6, 1,'9524 2037 9698 1817 1940 1569 3588', 2, 15,'2018-09-06');

INSERT INTO NotaFiscal(NNota, serie, chaveacesso, idpessoa, idpessoajuridica, dataemissao)VALUES(7, 1,'4182 7683 4529 8230 2149 5736 7853', 1, 16,'2018-09-01');

INSERT INTO NotaFiscal(NNota, serie, chaveacesso, idpessoa, idpessoajuridica, dataemissao)VALUES(8, 1,'1790 8048 3187 1882 9859 7877 4484', 3, 14,'2018-09-11');

**Povoando a relação Produto.**

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(1, '2018-09-13', 4.89, 'Arroz Namorado');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(2, '2018-09-11', 2.20, 'Macarrão Gostoso');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(3, '2018-09-15', 4.50, 'Oléo de Soja');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(4, '2018-09-17', 4.35, 'Arror Parborizado');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(5, '2018-09-15', 1.80, 'Vitamilho Dona Benta');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(6, '2018-09-18', 2.00, 'Vitamilho flocão');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(7, '2018-09-19', 2.20, 'Papel Higienico');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(8, '2018-09-17', 3.89, 'Feijão de Corda');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(9, '2018-09-16', 4.00, 'Feijão Mulatinho');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(10, '2018-09-13', 3.90, 'Açucar refinado');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(11, '2023-09-13', 120.00, 'Camera Intelbras HD');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(12, '2024-09-17', 300.00, 'DVR Hikvision');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(13, '2021-09-15', 180.00, 'Camera IP 4k HIK');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(14, '2024-09-18', 189.00, 'Central De Alarme ecp');

INSERT INTO Produto(ID, Validade, Valor, Descricao)VALUES(15, '2027-09-16', 165.00, 'Central de Choque Monitorada');

**Removendo a restrição NOT NULL da tabela venda.**

ALTER TABLE Venda ALTER COLUMN NCupom DROP NOT NULL;

ALTER TABLE Venda ALTER COLUMN NNota DROP NOT NULL;

**Povoando a relação Venda.**

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(1, '2018-08-06','Consumidor', 232, null, 1);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(2, '2017-07-07','Consumidor', 22, null, 2);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(3, '2018-06-05','Consumidor', 21, null, 3);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(4, '2017-03-26','Consumidor', 32, null, 4);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(5, '2018-11-16','Consumidor', 432, null, 5);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(6, '2018-12-21','Consumidor', 453, null, 6);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(7, '2018-08-04','Consumidor', 567, null, 7);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(8, '2017-08-06','Consumidor', 278, null, 8);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(9, '2018-08-06','Empresa', 567, 3, null);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(10, '2017-07-07','Empresa', 656, 4, null);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(11, '2018-06-05','Empresa', 678, 5, null);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(12, '2017-03-26','Empresa', 142, 6, null);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(13, '2018-11-16','Empresa', 432, 7, null);

INSERT INTO Venda (ID, Data, TipoVenda, Valor, NNota, NCupom)VALUES(14, '2018-12-21','Empresa', 890, 8, null);

**Povoando a relação Sangramento.**

INSERT INTO Sangramento (Valor, Data, Hora)VALUES(1800, '2018-06-05','18:32:31.00');

INSERT INTO Sangramento (Valor, Data, Hora)VALUES(1400, '2018-06-06','17:12:24.00');

INSERT INTO Sangramento (Valor, Data, Hora)VALUES(1000, '2018-06-07','14:23:34.00');

INSERT INTO Sangramento (Valor, Data, Hora)VALUES(1100, '2018-06-08','15:25:12.00');

INSERT INTO Sangramento (Valor, Data, Hora)VALUES(1120, '2018-06-09','15:30:57.00');

INSERT INTO Sangramento (Valor, Data, Hora)VALUES(1300, '2018-06-10','17:58:59.00');

**Povoando a relação Venda Produto.**

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(13, 2, 8);

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(1, 3, 15);

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(3, 4, 5);

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(14, 5, 4);

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(2, 6, 20);

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(4, 7, 23);

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(6, 8, 4);

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(5, 9, 5);

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(8, 10, 80);

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(7, 11, 190);

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(10, 12, 8);

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(9, 13, 3);

INSERT INTO VendaProduto (IDProduto, IDVenda, Quantidade)VALUES(10, 14, 16);

**Povoando a relação Sangramento Caixa.**

INSERT INTO SangramentoCaixa (IDCaixa, Valor, Data)VALUES(1, 1000, '2018-06-07');

INSERT INTO SangramentoCaixa (IDCaixa, Valor, Data)VALUES(2, 1400, '2018-06-06');

INSERT INTO SangramentoCaixa (IDCaixa, Valor, Data)VALUES(3, 1300, '2018-06-10');

INSERT INTO SangramentoCaixa (IDCaixa, Valor, Data)VALUES(4, 1800, '2018-06-05');

INSERT INTO SangramentoCaixa (IDCaixa, Valor, Data)VALUES(5, 1100, '2018-06-08');

# Criando as Visões.

1. **Esta visão foi criada com o proposito de ter a fixa com os dados dos clientes que efetuarem um pagamento com cheque, para o caso de o cheque não ter fundo. Com isso o dono terá sempre os dados do cliente que passou o cheque.**

Create view fichaCliente as

select pf.cpf, p.nome,p.uf as Estado, p.cidade,p.bairro,p.rua, p.numero as NumeroCasa, pg.data as DataPagamento,pg.valor as ValorCompra, pc.ncheque

from Pessoa p, PessoaFisica pf, Pagamento pg, PagamentoCheque pc

where p.id=pf.idPessoa and pg.idPessoa=p.id and pg.id=pc.idPagamento;

1. **Essa visão foi criada com o intuito de obter os dados dos clientes que fizerem uma compra a vista. Ela auxilia na execução de um gatilho que por sua vez, é responsável por conceder um desconto para quem paga a compra avista.**

create view pagamentosAvista as

select p.nome, pg.data as dataCompra, pg.valor as ValorCompra, pa.dataEfetivacao

from Pessoa p, Pagamento pg, pagamentoAvista pa

where p.id=pg.idPessoa and pg.id=pa.idPagamento;

1. **Essa visão foi criada com o intuído de obter os dados dos clientes que efetuaram um pagamento via cartão de crédito. Essa visão auxilia na execução de um gatilho, que por sua vez concede um desconto para quem paga usando cartão de crédito.**

create view pagamentoscartao as

select p.nome, pg.data as datapagamento, pg.valor as valorcompra

from pessoa p, pagamento pg, pagamentocartao pc

where p.id = pg.idpessoa and pg.id = pc.idpagamento;

1. **Essa visão foi criada com intuito de retornar todos os dados do caixa e dos funcionários que opera um caixa**

create view public.caixafuncionario as

select distinct \*

from caixa c natural join funcionario f;

# Scripts SQL para consultas.

1. Recupere todos os números das notas fiscais, juntamente com a data de emissão onde o CNPJ é igual a 87.027.742/0001-99.

**select nf.nnota, nf.dataemissao**

**from pessoajuridica pf join notafiscal nf on(pf.idpessoa=nf.idpessoajuridica)**

**where pf.cnpj= '87.027.742/0001-99'**

1. Recupere todos os Produtos vendidos, a descrição e o valor dos produtos que foram vendidos mais de 3 unidades.

**select p.descricao, p.valor**

**from produto p join vendaProduto vp on(p.id=vp.idProduto)**

**where vp.quantidade >= 3**

1. Recupere o nome e o valor da compra de todas as pessoas que efetuaram um pagamento via cartão de crédito, cujo o valor da compra foi dividido em 4 parcelas. O resultado final deve mostrar o valor de cada compra ordenado de forma descendente.

**select p.nome as NomePessoa, pg.valor as ValorCompra**

**from Pessoa p, Pagamento pg , PagamentoCartao pc**

**where p.id = pg.idPessoa and pg.id=pc.idPagamento and pc.numeroParcelas=4**

**order by pg.valor desc**

1. Recupere o ID, o Nome das pessoas que efetuaram um pagamento com cartão de crédito e também um pagamento a vista, a exibição deve ordenar o ID da pessoa de forma ascendente.

**select nome, id**

**from Pessoa**

**where id in(**

**(select pg.idPessoa**

**from Pagamento pg, PagamentoCartao pc**

**where pg.id=pc.idPagamento)**

**Intersect**

**(select pg.idPessoa**

**from pagamento pg, PagamentoAvista pc**

**where pg.id=pc.idPagamento)**

**)**

**order by id**

1. Recupere o Nome e o Telefone de todas as pessoas cuja a primeira letra do nome começa com ‘M’.

**select p.nome as NomePessoa, pf.telefone as TelefonePessoa**

**from Pessoa p join PessoaTelefone pf on(p.id=pf.idPessoa )**

**where p.nome like 'M%'**

1. Recupere o ID, Nome e a quantidade de telefones de todas as pessoas que tem mais de 1 telefone cadastrado. Deve-se exibir o resultado ordenado pelo ID da pessoa de forma ascendente.

**select p.id, p.nome, count(p.id) as QuantidadeTelefones**

**from Pessoa p join PessoaTelefone pf on(p.id=pf.idPessoa)**

**group by p.id, p.nome**

**having count(\*)>1**

**order by p.id,p.nome**

1. Crie uma visão que mostre o nome do funcionário e valor do caixa após o fechamento do mesmo. O resultado deve exibir o valor do caixa após fechado em ordem descendente.

**create view FechamentoCaixa as**

**select f.nome, c.valorFechamento**

**from funcionario f join caixa c on (f.nctps=c.nctps)**

**group by c.valorFechamento, f.nome**

**order by c.valorFechamento desc**

1. Recupere o id, a data e a descrição do produto onde o número da nota seja NULO e o valor da venda seja maior que 250 reais.

**select v.id, p.descricao, v.data**

**from venda v, vendaProduto vp, produto p**

**where v.id= vp.idVenda and vp.idProduto = p.id and v.nnota is null and v.valor >250**

1. Recupere o número da nota fiscal, o nome da pessoa Jurídica e a razão social do mesmo onde na venda o número da nota fiscal seja diferente de NULO e o valor da venda seja menor que 999 reais. Deve-se exibir o nome da pessoa jurídica em ordem descendente.

**select p.nome as NomePessoaJuridica, pj.razaosocial, nf.nnota as NúmeroNotaFiscal**

**from Pessoa p , PessoaJuridica pj, NotaFiscal nf, Venda v**

**where p.id=pj.idPessoa and v.nnota=nf.nnota and nf.idPessoajuridica=pj.idPessoa and v.nnota is not null and v.valor < 999**

**order by p.nome desc**

1. Recupere a descrição do produto onde a venda seja do tipo ‘consumidor’ e a descrição do produto tenha a letra ‘c’ em sua descrição.

**select p.descricao**

**from Produto p, vendaProduto vp , venda v**

**where p.id=vp.idProduto and vp.idVenda=v.id and v.tipoVenda = 'Consumidor' and p.descricao like 'A%'**

1. Recupere o menor salário, o maior salário e a média salarial e valor total do salário de todos os empregados onde a data de admissão e igual a ‘2018-09-01’.

**select**

**Max(salario) as MaiorSalario,**

**Min(Salario) as MenorSalario,**

**AVG(Salario) as MediaSalarial,**

**SUM(Salario) as Total**

**from Funcionario**

**where dataAdmissao = '2018-09-01'**

1. Recupere o nome e o salário dos funcionários que ganham acima de 1000 reais. O resultado deve ser ordenado pelo salário de forma descendente.

**select nome, salario**

**from Funcionario**

**group by nome, salario**

**having (MIN(salario)>1000)**

**order by salario desc**

1. Recupere todos os ID e valor dos pagamentos feitos a prazo aonde o valor do mesmo seja maior que 280.

**(select idpagamento, p.valor from pagamentocheque pc, pagamento p where p.id = pc.idpagamento and valor > 280)**

**union**

**(select idpagamento, p.valor from pagamentocartao pca, pagamento p where p.id = pca.idpagamento and valor > 280)**

1. Recupere todos os nomes dos funcionários que não possuem telefone.

**select nome from funcionario f**

**where not exists**

**(select \* from funcionariotelefone ft**

**where f.nctps = ft.nctps)**

1. Recupere o ID e o número de Parcelas dos Pagamentos feitos no cartão.

**select pc.idpagamento, pc.numeroparcelas from pagamentocartao pc**

**where exists**

**(select \* from pagamento p)**

1. Recupere o ID e o número do cheque dos Pagamentos feitos no cheque.

**select pc.idpagamento, pc.ncheque from pagamentocheque pc**

**where exists**

**(select \* from pagamento p)**

1. Recupere o nome e o valor do pagamento de todas as pessoas, cujo valor do mesmo foi maior do que o pagamento de 'David' na data “2018-08-17”. O resultado deve ser ordenado pelo valor do pagamento em ordem descendente.

**select p.nome, pg.valor**

**from pessoa p join pagamento pg on(p.id=pg.idpessoa)**

**where pg.valor > all**

**(select pg.valor**

**from pessoa p join pagamento pg on(p.id=pg.idpessoa)**

**where p.nome= 'david' and pg.data='2018-08-17');**

# Criando Procedimentos

1. Crie um procedimento armazenado para retornar o total do valor do caixa operado por um determinado funcionario na data ‘X’ onde o funcionario e a data será passado por parâmetro.

**create or replace function totalcaixafuncionariodata (date, integer) returns caixa.valorfechamento%type**

**as $$**

**declare**

**dia alias for $1;**

**ncarteira alias for $2;**

**soma caixa.valorfechamento%type := 0;**

**valor caixa.valorfechamento%type;**

**curs refcursor;**

**begin**

**open curs for select valorfechamento from caixa where datafechamento = dia and nctps = ncarteira;**

**fetch curs into valor;**

**while found loop**

**soma := soma + valor;**

**fetch curs into valor;**

**end loop;**

**close curs;**

**return soma;**

**end**

**$$**

**language plpgsql;**

1. Crie um Procedimento armazenado para retorna o somatório de todas as vendas realizadas em uma data passado por parâmetro, no mercado independente do tipo da venda se a mesma e consumidor ou empresa.

**create or replace function totaldasvendas (date) returns venda.valor%type**

**as $$**

**declare**

**dia alias for $1;**

**totalvendas venda.valor%type := 0;**

**valoresvenda venda.valor%type;**

**ponteiro refcursor;**

**begin**

**open ponteiro for**

**select valor from venda where data=dia;**

**fetch ponteiro into valoresvenda;**

**while found loop**

**totalvendas := totalvendas + valoresvenda;**

**fetch ponteiro into valoresvenda;**

**end loop;**

**close ponteiro;**

**return totalvendas;**

**end**

**$$**

**language plpgsql;**

# Criando Gatilhos.

1. Crie um gatilho para conceder um desconto de 3% para todas as vendas acima de 300 reais.

**create or replace function descontoconcedido()**

**returns trigger as $$**

**declare**

**desconto real = 300;**

**begin**

**select into desconto from venda**

**where desconto > new.id**

**group by id;**

**update venda set valor = valor \* 0.97 where id = new.id;**

**return null;**

**end**

**$$**

**language plpgsql;**

**create trigger desconta**

**after insert on venda**

**for each row**

**execute procedure descontoconcedido();**

1. Crie um gatilho para conceder um desconto de 1% para todos os pagamentos feitos a vista, cujo o valor esteja acima de 100 reais.

**create or replace function descontopagamentocartao()**

**returns trigger as $$**

**declare**

**desconto real = 100;**

**begin**

**select into desconto from pagamentoscartao**

**where desconto > new.id;**

**update pagamento set valor = valor \* 0.99 where id = new.id;**

**return null;**

**end**

**$$**

**language plpgsql;**

**create trigger descontocartao**

**after insert on pagamento**

**for each row**

**execute procedure descontopagamentocartao();**

1. Crie um gatilho para conceder um desconto de 3% para todos os pagamentos feitos avista cujo o valor esteja acima de 100 reais.

**create or replace function descontoavistapagamento()**

**returns trigger as $$**

**declare**

**desconto real = 100;**

**begin**

**select into desconto from pagamentosavista**

**where desconto > new.id;**

**update pagamento set valor = valor \* 0.97 where id = new.id;**

**return null;**

**end**

**$$**

**language plpgsql;**

**create trigger descontoavista**

**after insert on pagamento**

**for each row**

**execute procedure descontoavistapagamento();**