

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
PARAÍBA
Campus Cajazeiras

Projeto da disciplina Banco de Dados I

**Banco de Dados Relacional voltado para uma
distribuidora de bebidas**

IFPB Cajazeiras

Análise e Desenvolvimento de Sistemas

Flavio Henrique de Sousa Silva

2017.1

SUMÁRIO

1. 1ª Etapa	3
1.1 Análise do Sistema	4
1.1.1 Descrição do problema	4
1.1.2 Descrição do domínio de aplicação	5
1.1.3 Descrição das consultas oferecidas	6
1.2 Projeto Conceitual	7
1.2.1 Figura 1: Diagrama entidade-relacionamento	7
1.2.2 Dicionário de dados conceitual	8
2. 2ª Etapa	15
2.1 Projeto Lógico	16
2.1.1 Esquema lógico	16
2.1.2 Dicionário de dados lógico	17
2.2 Início da implementação	31
2.2.1. Scripts SQL (criação de tabelas)	31
2.2.2. Scripts SQL (povoamento)	39
3. 3ª Etapa	50
3.1 Conclusão da implementação	51
3.1.1. Scripts SQL (índices)	51
3.1.2. Scripts SQL (consultas)	54
3.1.3. Scripts SQL (visões)	58
3.1.4. Scripts SQL (procedimentos armazenados)	60
3.1.5. Scripts SQL (gatilhos)	63

1ª Etapa

Projeto da disciplina Banco de Dados I

1.1 Análise do sistema

1.1.1. Descrição do problema

Vivemos em uma época em que a tecnologia vem crescendo de forma muito rápida, e tem feito parte da vida das pessoas em todas as áreas, como a educação, lazer, relacionamentos e também a área profissional. Hoje em dia, qualquer empresa necessita de um sistema que gerencia as informações produzidas e dados registrados, pois os mesmos crescem a todo momento, e o armazenamento de tantas informações de forma física, como pastas, amontoados de papeis e documentos é uma tarefa muito difícil, devido à velocidade em que surgem novos dados para serem registrados. É fundamental a utilização de softwares para facilitar o controle de tudo que diz respeito à empresa e a solução de problemas de forma mais rápida.

O domínio de aplicação utilizado neste projeto será uma distribuidora de bebidas localizada em Sousa-PB (Graal Distribuidora), onde é necessário o registro de todos os funcionários que trabalham na empresa, de todas as vendas feitas, todos os estabelecimentos que são clientes dessa distribuidora, o controle de todas as mercadorias que são distribuídas, de todos os veículos disponíveis na empresa, e tudo isso gera a necessidade do desenvolvimento de um banco de dados para gerenciar essas informações, para organizar todos esses dados e torná-los de fácil acesso, organizando as ações da empresa e guardando tudo que acontece no ambiente de trabalho, possibilitando um maior controle e uma melhor administração.

1.1.2. Descrição do domínio de aplicação

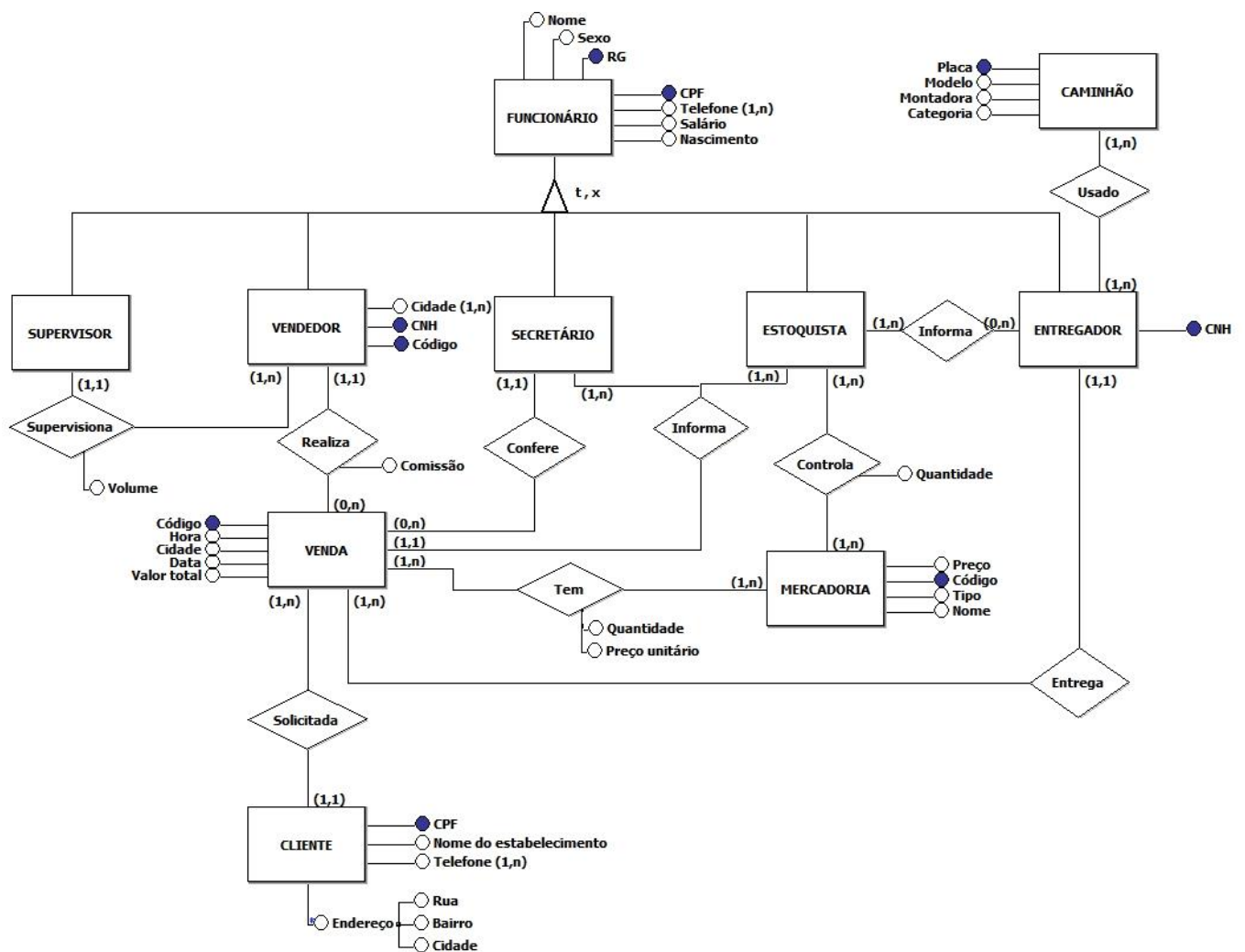
Na empresa atuam funcionários, que tem suas informações guardadas como o seu nome, sexo, CPF, RG, valor do salário, números de telefone e data de nascimento. Os funcionários se dividem em 5 tipos, que são: supervisor, vendedor, secretário, estoquista e entregador, de modo que todos os funcionários devem pertencer a algum desses 5 tipos, e nenhum funcionário pode pertencer a mais de um tipo ao mesmo tempo. Vendedores tem número da CNH, cidades em que trabalha e o seu código. Entregadores também tem o número da sua CNH registrado. Também são registradas as vendas realizadas pelos vendedores, guardando informações sobre a hora e a cidade em que foi realizada, a data, o valor total e seu código, além disso, os vendedores recebem sua comissão de acordo com o valor da venda realizada. As mercadorias existentes nas vendas têm seu preço, código, o tipo e seu nome, além de serem registradas a quantidade de cada mercadoria e seu preço unitário em cada venda. Também são registradas as informações dos clientes que solicitam as vendas, que tem o nome do estabelecimento, o CPF, os números de telefone e seu endereço, composto por rua, bairro e cidade. Para serem realizadas as entregas da empresa, os entregadores usam caminhões, que tem sua placa, seu modelo, montadora e categoria. Vendedores são supervisionados pelos supervisores, que os informam o volume que deve ser vendido em cada venda, já os secretários conferem as vendas realizadas pelos vendedores e informam sobre elas para os estoquistas, que controlam a quantidade de mercadorias, que por sua vez informam os entregadores, que entregam as vendas realizando a sua conclusão.

1.1.3. Descrição das consultas oferecidas

- 1 - O nome, CPF e salário de todos os vendedores que trabalham na cidade Sousa.
- 2- Todos os dados dos estoquistas que controlam a mercadoria com código '001', ordenados pelo salário em ordem decrescente.
- 3- A quantidade de vendas com valor total acima de R\$ 300,00.
- 4- O nome, código e preço de todas as mercadorias do tipo "Cerveja", ordenados em ordem alfabética.
- 5- O telefone dos clientes que estão na cidade Cajazeiras.
- 6- O código dos vendedores supervisionados pelo supervisor de RG '4132793'.
- 7- O modelo, a montadora e a placa de todos os caminhões de categoria "leve".
- 8- O código e valor total de todas as vendas conferidas pelo secretário com CPF '02547274418'.
- 9- O valor acumulado em todas as vendas registradas.
- 10- O CPF de todos os clientes com o nome do estabelecimento iniciado pela letra 'F'.
- 11- Para cada cidade em que existem clientes, recuperar o nome da cidade e a quantidade de clientes.
- 12- Recuperar o nome da cidade, a quantidade de vendas realizadas e a soma dos valores totais das vendas de cada cidade que tenha mais de uma venda registrada.
- 13 -O nome e o CPF dos entregadores que utilizaram caminhões.
- 14- O nome e o CPF dos secretários que ainda não informaram nenhum estoquista sobre vendas.

1.2 Projeto conceitual

1.2.1. Figura 1: Diagrama entidade-relacionamento



1.2.2. Dicionário de dados conceitual

Entidade FUNCIONÁRIO:

Guarda as informações referentes a algum funcionário da empresa, e apresenta 5 especializações: Supervisor, Vendedor, Secretário, Estoquista e Entregador.

Atributos da entidade FUNCIONÁRIO:

Nome: Atributo que armazena o nome do funcionário.

Sexo: Atributo que armazena o sexo do funcionário.

Salário: Atributo que armazena o valor do salário dos funcionários.

CPF: Atributo que armazena o número do CPF dos funcionários.

RG: Atributo que armazena o número da carteira de identidade dos funcionários.

Nascimento: Atributo que armazena a data de nascimento dos funcionários.

Telefone: Atributo multivalorado que armazena os números dos telefones dos funcionários.

Entidade SUPERVISOR:

Guarda as informações dos supervisores que trabalham na empresa.

Relacionamentos da entidade SUPERVISOR:

Supervisiona: Indica o relacionamento entre SUPERVISOR e VENDEDOR, em que cada supervisor pode supervisionar um ou mais vendedores, e cada vendedor pode ser supervisionado apenas por um supervisor.

Atributos do relacionamento supervisiona:

Volume: Atributo que armazena a quantidade de mercadorias que devem ser vendidas pelo vendedor em cada venda, que é determinada pelo supervisor.

Entidade VENDEDOR:

Armazena informações sobre os vendedores que trabalham na empresa.

Atributos da entidade VENDEDOR:

Cidade: Atributo multivalorado que armazena os nomes das cidades que serão visitadas pelo vendedor.

CNH: Atributo que armazena o número da Carteira Nacional de Habilitação do vendedor.

Código: Atributo que armazena o código do vendedor.

Relacionamentos da entidade VENDEDOR:

Supervisiona: indica o relacionamento entre Vendedor e Supervisor, onde um vendedor pode ser supervisionado apenas por um Supervisor, e um Supervisor pode supervisionar um ou mais vendedores.

Realiza: Indica o relacionamento entre Vendedor e Venda, onde um vendedor pode realizar nenhuma ou muitas vendas, e uma venda pode ser realizada por apenas um vendedor.

Atributos do relacionamento Realiza:

Comissão: Atributo que indica o valor da comissão que o vendedor recebe em cada venda.

Entidade SECRETÁRIO:

Armazena as informações dos secretários que trabalham na empresa.

Relacionamentos da entidade SECRETÁRIO:

Confere: Indica o relacionamento entre secretário e venda, onde cada secretário confere nenhuma ou muitas vendas, e cada venda pode ser conferida por apenas um secretário.

Informa: Indica o relacionamento ternário entre Secretário, Estoquista e Venda, onde cada Secretário pode informar um ou muitos Estoquistas sobre uma determinada Venda.

Entidade ESTOQUISTA:

Armazena as informações dos Estoquistas que trabalham na empresa.

Relacionamentos da entidade ESTOQUISTA:

Informa: Indica o relacionamento ternário entre Estoquista, Secretário e Venda, onde cada Estoquista pode ser informado por um ou mais secretários sobre uma determinada Venda.

Informa: Indica o relacionamento entre Estoquista e Entregador, onde cada Estoquista informa a nenhum ou muitos Entregadores, e cada Entregador é informado por um ou mais Estoquistas.

Controla: Indica o relacionamento entre Estoquista e Mercadoria, onde cada Estoquista controla uma ou mais Mercadorias, e cada Mercadoria é controlada por um ou mais Estoquistas.

Atributos do relacionamento Controla:

Quantidade: Atributo que armazena a quantidade em estoque de cada mercadoria.

Entidade ENTREGADOR:

Armazena as informações sobre os entregadores que trabalham na empresa.

Atributos da entidade ENTREGADOR:

CNH: Atributo que armazena o número da Carteira Nacional de Habilitação do Entregador.

Relacionamentos da entidade ENTREGADOR:

Informa: Indica o relacionamento entre Entregador e Estoquista, onde cada Entregador pode ser informado por um ou mais Estoquistas, e cada Estoquista pode informar a nenhum ou muitos Entregadores.

Entrega: Indica o relacionamento entre Entregador e Venda, onde cada Entregador entrega uma ou mais Vendas, e cada Venda pode ser entregue por apenas um Entregador.

Usado: Indica a relação entre Entregador e Caminhão, onde cada Entregador usa um ou mais Caminhões, e cada Caminhão é usado por um ou mais Entregadores.

Entidade CAMINHÃO:

Armazena as informações dos caminhões que são utilizados na empresa.

Atributos da entidade CAMINHÃO:

Placa: Atributo que armazena a placa do caminhão.

Modelo: Atributo que armazena o modelo do caminhão.

Montadora: Atributo que armazena o nome da montadora do caminhão.

Categoria: Atributo que armazena a categoria do caminhão.

Relacionamentos da entidade CAMINHÃO:

Usado: Indica o relacionamento entre Caminhão e Entregador, onde cada Caminhão é usado por um ou mais Entregadores, e cada Entregador usa um ou muitos caminhões.

Entidade VENDA:

Armazena as informações das vendas realizadas pelos vendedores da empresa.

Atributos da entidade VENDA:

Hora: Atributo que armazena a hora em que a venda foi realizada.

Cidade: Atributo que armazena o nome da cidade em que a venda foi realizada.

Data: Atributo que armazena a data da realização da venda.

Valor total: Atributo que armazena o valor total de todas as mercadorias vendidas.

Código: Atributo que armazena o código de identificação de cada venda realizada.

Relacionamentos da entidade VENDA:

Realiza: Indica o relacionamento entre Venda e Vendedor, onde cada Venda é realizada por apenas um Vendedor, e cada Vendedor realiza nenhuma ou muitas vendas.

Confere: Indica o relacionamento entre Venda e Secretário, onde cada Venda é conferida por um Secretário, e um Secretário confere nenhuma ou várias Vendas.

Solicitada: Indica o relacionamento entre Venda e Cliente, onde uma Venda pode ser solicitada por apenas um Cliente, e cada Cliente solicita uma ou mais Vendas.

Tem: Indica o relacionamento entre Venda e Mercadoria, onde em uma Venda podem ter uma ou mais Mercadorias, e uma Mercadoria pode pertencer a uma ou mais Vendas.

Entrega: Indica o relacionamento entre Venda e Entregador, onde cada Venda é entregue por apenas um Entregador, e cada Entregador entrega uma ou mais vendas.

Informa: Indica o relacionamento ternário entre Venda, Secretário e Estoquista, onde cada Secretário informa um ou mais Estoquistas sobre uma determinada Venda.

Atributos do relacionamento Tem:

Quantidade: Atributo que armazena a quantidade de cada mercadoria presente na venda.

Preço unitário: Atributo que armazena o preço de cada mercadoria presente na venda.

Entidade MERCADORIA:

Armazena informações sobre as mercadorias que são vendidas na empresa.

Atributos da entidade MERCADORIA:

Preço: Atributo que armazena os preços das mercadorias fornecidas.

Código: Atributo que armazena o código identificador de cada mercadoria.

Tipo: Atributo que armazena o tipo de cada mercadoria.

Nome: Atributo que armazena o nome de cada mercadoria.

Relacionamentos da entidade MERCADORIA:

Controla: Indica o relacionamento entre Mercadoria e Estoquista, onde cada Mercadoria é controlada por um ou mais Estoquistas, e cada Estoquista controla uma ou mais Mercadorias.

Tem: Indica o relacionamento entre Mercadoria e Venda, onde cada Mercadoria pertence a uma ou mais Vendas, e cada Venda tem uma ou mais Mercadorias.

Entidade CLIENTE:

Armazena as informações dos clientes que são registrados na empresa.

Atributos da entidade CLIENTE:

Telefone: Atributo que armazena os números dos telefones dos clientes.

Nome do estabelecimento: Atributo que armazena o nome do estabelecimento do cliente.

CPF: Atributo que armazena o número do CPF do cliente.

Endereço: Atributo que armazena o endereço do cliente, composto pela rua, bairro e cidade.

Relacionamentos da entidade CLIENTE:

Solicitada: Indica o relacionamento entre Cliente e Venda, onde um Cliente solicita uma ou muitas Vendas, e uma Venda é solicitada por apenas um Cliente.

2ª Etapa

Projeto da disciplina Banco de Dados I

2.1 Projeto lógico

2.1.1. Esquema lógico

SUPERVISOR (Nome, Sexo, RG, CPF, Salário, Nascimento);

VENDEDOR (Nome, Sexo, RG, CPF, Salário, Nascimento, CNH, Código, CPFSupervisor, Volume);

SECRETÁRIO (Nome, Sexo, RG, CPF, Salário, Nascimento);

ESTOQUISTA (Nome, Sexo, RG, CPF, Salário, Nascimento);

ENTREGADOR (Nome, Sexo, RG, CPF, Salário, Nascimento, CNH);

CAMINHÃO (Placa, Modelo, Montadora, Categoria);

VENDA (Código, Hora, Cidade, Data, ValorTotal, CPFVendedor, Comissão, CPFSecretário, CPFCliente, CPFEntregador);

MERCADORIA (Preço, Código, Tipo, Nome);

CLIENTE (NomeEstabelecimento, CPF, Rua, Bairro, Cidade);

TEM_MERCADORIA (CodVenda, CodMercadoria, Quantidade, PreçoUnitário);

CONTROLA_MERCADORIA (CPFEstoquista, CodMercadoria, Quantidade);

USA_CAMINHÃO (PlacaCaminhão, CPFEntregador);

SECRETARIO_ESTOQUISTA _VENDA (CPFSecretário, CPFEstoquista, CodVenda);

INFORMA_ENTREGADOR (CPFEstoquista, CPFEntregador);

TELEFONE_SUPERVISOR (CPFSupervisor, Telefone);

TELEFONE_VENDEDOR (CPFVendedor, Telefone);

TELEFONE_ESTOQUISTA (CPFEstoquista, Telefone);

TELEFONE_ENTREGADOR (CPFEntregador, Telefone);

TELEFONE_SECRETARIO (CPFSecretario, Telefone);

CIDADES_VENDEDOR (CPFVendedor, Cidade);

TELEFONE_CLIENTE (CPFCliente, Telefone);

2.1.2. Dicionário de dados lógico

SUPERVISOR: Relação que armazena os dados dos supervisores da empresa.				
Atributo	Descrição	Tipo	Domínio	Restrições
Nome	Atributo que representa o nome do supervisor	String(30)	String(30)	<ul style="list-style-type: none">• Não-nulo
Sexo	Atributo que representa o sexo do supervisor	Char	'M' ou 'F'	<ul style="list-style-type: none">• Não-nulo
RG	Atributo que representa o número da carteira da identidade do supervisor	String(7)	Números inteiros positivos	<ul style="list-style-type: none">• Não-nulo• Único
CPF	Atributo que representa o número do CPF do supervisor	String(11)	Números inteiros positivos	<ul style="list-style-type: none">• Chave primária
Salário	Atributo que representa o valor do salário do supervisor	Real	Números reais positivos	<ul style="list-style-type: none">• Não-nulo
Nascimento	Atributo que representa a data de nascimento do supervisor	Date	"dd/mm/yyyy"	<ul style="list-style-type: none">• Não - nulo

VENDEDOR: Relação que armazena os dados de cada vendedor da empresa.				
Atributo	Descrição	Tipo	Domínio	Descrições
Nome	Atributo que representa o nome do vendedor	String(30)	String(30)	<ul style="list-style-type: none">• Não-nulo
Sexo	Atributo que representa o	Char	'M' ou 'F'	<ul style="list-style-type: none">• Não-nulo

	sexo do vendedor			
RG	Atributo que representa o número da carteira da identidade do vendedor	String(7)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Único
CPF	Atributo que representa o número do CPF do vendedor	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave primária
Salário	Atributo que representa o valor do salário do vendedor	Real	Números reais positivos	<ul style="list-style-type: none"> • Não-nulo
Nascimento	Atributo que representa a data de nascimento do vendedor	Date	dd/mm/yyyy	<ul style="list-style-type: none"> • Não - nulo
CNH	Atributo que representa o número da carteira de habilitação do vendedor	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Único
Código	Atributo que representa o código identificador do vendedor	String(4)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Único
CPFSupervisor	Atributo que representa o CPF do supervisor que supervisiona o vendedor	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave estrangeira que referencia o atributo "CPF" da relação "SUPERVISOR" • Não-nulo
Volume	Atributo que representa a quantidade de mercadorias que devem ser vendidas	Int	Int	<ul style="list-style-type: none"> • Não-nulo

SECRETÁRIO: Relação que armazena os dados de cada secretário da empresa.				
Atributo	Descrição	Tipo	Domínio	Descrições
Nome	Atributo que representa o nome do secretário	String(30)	String(30)	<ul style="list-style-type: none"> • Não-nulo
Sexo	Atributo que representa o sexo do secretário	Char	'M' ou 'F'	<ul style="list-style-type: none"> • Não-nulo
RG	Atributo que representa o número da carteira da identidade do secretário	String(7)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Único
CPF	Atributo que representa o número do CPF do secretário	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave primária
Salário	Atributo que representa o valor do salário do secretário	Real	Números reais positivos	<ul style="list-style-type: none"> • Não-nulo
Nascimento	Atributo que representa a data de nascimento do secretário	Date	dd/mm/yyyy	<ul style="list-style-type: none"> • Não - nulo

ESTOQUISTA: Relação que armazena os dados de cada estoquista da empresa.				
Atributo	Descrição	Tipo	Domínio	Descrições
Nome	Atributo que representa o nome do estoquista	String(30)	String(30)	<ul style="list-style-type: none"> • Não-nulo

Sexo	Atributo que representa o sexo do estoquista	Char	'M' ou 'F'	<ul style="list-style-type: none"> • Não-nulo
RG	Atributo que representa o número da carteira da identidade do estoquista	String(7)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Único
CPF	Atributo que representa o número do CPF do estoquista	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave primária
Salário	Atributo que representa o valor do salário do estoquista	Real	Números reais positivos	<ul style="list-style-type: none"> • Não-nulo
Nascimento	Atributo que representa a data de nascimento do estoquista	Date	dd/mm/yyyy	<ul style="list-style-type: none"> • Não - nulo

ENTREGADOR: Relação que armazena os dados de cada entregador da empresa.

Atributo	Descrição	Tipo	Domínio	Descrições
Nome	Atributo que representa o nome do entregador	String(30)	String(30)	<ul style="list-style-type: none"> • Não-nulo
Sexo	Atributo que representa o sexo do entregador	Char	'M' ou 'F'	<ul style="list-style-type: none"> • Não-nulo
RG	Atributo que representa o número da carteira da identidade do entregador	String(7)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Único

CPF	Atributo que representa o número do CPF do entregador	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave primária
Salário	Atributo que representa o valor do salário do entregador	Real	Números reais positivos	<ul style="list-style-type: none"> • Não-nulo
Nascimento	Atributo que representa a data de nascimento do entregador	Date	dd/mm/yyyy	<ul style="list-style-type: none"> • Não - nulo
CNH	Atributo que representa o número da habilitação do entregador	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo

CAMINHÃO: Relação que armazena os dados de cada caminhão da empresa.

Atributo	Descrição	Tipo	Domínio	Descrições
Placa	Atributo que representa a placa do caminhão	String(7)	String(7)	<ul style="list-style-type: none"> • Chave primária
Modelo	Atributo que representa o modelo do caminhão	String(7)	String(7)	<ul style="list-style-type: none"> • Não-nulo
Montadora	Atributo que representa o nome da montadora do caminhão	String(10)	String(10)	<ul style="list-style-type: none"> • Não-nulo
Categoria	Atributo que representa a categoria do caminhão	String(10)	String(10)	<ul style="list-style-type: none"> • Não-nulo

VENDA: Relação que armazena os dados de cada venda registrada.				
Atributo	Descrição	Tipo	Domínio	Descrições
Código	Atributo que representa o código da venda	Int	Int	<ul style="list-style-type: none"> Chave primária
Hora	Atributo que representa a hora em que a venda foi realizada	Time	Time	<ul style="list-style-type: none"> Não-nulo
Cidade	Atributo que representa a cidade em que a venda foi realizada	String(20)	String(20)	<ul style="list-style-type: none"> Não-nulo
Data	Atributo que representa a data de realização da venda	Date	dd/mm/yyyy	<ul style="list-style-type: none"> Não-nulo
ValorTotal	Atributo que representa o valor total de todas as mercadorias da venda	Real	Números reais positivos	<ul style="list-style-type: none"> Não-nulo
CPFVendedor	Atributo que representa o CPF do vendedor que realizou a venda	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> Chave estrangeira que referencia o atributo "CPF" da relação "VENDEDOR" Não-nulo
Comissão	Atributo que representa o valor da comissão que o vendedor recebe em cada venda	Real	Números reais positivos	<ul style="list-style-type: none"> Não-nulo
CPFSecretário	Atributo que representa o CPF do secretário	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> Chave estrangeira que referencia o atributo "CPF" da

	que conferiu a venda			relação "SECRETÁRIO" <ul style="list-style-type: none"> • Não-nulo
CPFCliente	Atributo que representa o CPF do cliente que solicitou a venda	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave estrangeira que referencia o atributo "CPF" da relação "CLIENTE" • Não-nulo
CPFEntregador	Atributo que representa o CPF do Entregador que entregou a venda	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave estrangeira que referencia o atributo "CPF" da relação "ENTREGADOR" • Não-nulo

MERCADORIA: Relação que armazena os dados de cada mercadoria distribuída pela empresa.				
Atributo	Descrição	Tipo	Domínio	Descrições
Preço	Atributo que representa o preço da mercadoria	Real	Números reais positivos	<ul style="list-style-type: none"> • Não-nulo
Código	Atributo que representa o código da mercadoria	String(3)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave primária
Tipo	Atributo que representa o tipo de mercadoria	String(20)	String(20)	<ul style="list-style-type: none"> • Não-nulo
Nome	Atributo que representa o nome da mercadoria	String(20)	String(20)	<ul style="list-style-type: none"> • Não-nulo

CLIENTE: Relação que armazena os dados de cada cliente da empresa.				
Atributo	Descrição	Tipo	Domínio	Descrições
Nome-Estabelecimento	Atributo que representa o preço da mercadoria	String(30)	String(30)	<ul style="list-style-type: none"> • Não-nulo
CPF	Atributo que representa o CPF do cliente	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave primária
Rua	Atributo que representa a rua do estabelecimento do cliente	String(30)	String(30)	<ul style="list-style-type: none"> • Não-nulo
Bairro	Atributo que representa o bairro do cliente	String(30)	String(30)	<ul style="list-style-type: none"> • Não-nulo
Cidade	Atributo que representa o nome da cidade do cliente	String(30)	String(30)	<ul style="list-style-type: none"> • Não-nulo

TEM_MERCADORIA: Relação que armazena os dados do relacionamento entre “VENDA” e “MERCADORIA”.				
Atributo	Descrição	Tipo	Domínio	Descrições
CodVenda	Atributo que representa o código da venda	Int	Int	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo “CÓDIGO” da relação “VENDA” • Chave primária

CodMercadoria	Atributo que representa o Código da mercadoria	Int	Int	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo "CÓDIGO" da relação "MERCADORIA" • Chave primária
Quantidade	Atributo que representa a quantidade da cada mercadoria presente na venda	Int	Int	<ul style="list-style-type: none"> • Não-nulo
Preço unitário	Atributo que representa o preço unitário de cada produto na venda	Real	Real	<ul style="list-style-type: none"> • Não-nulo

CONTROLA_MERCADORIA: Relação que armazena os dados do relacionamento entre "ESTOQUISTA" e "MERCADORIA".				
Atributo	Descrição	Tipo	Domínio	Descrições
CPFEstoquista	Atributo que representa o CPF do estoquista	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo "CPF" da relação "ESTOQUISTA" • Chave primária
CodMercadoria	Atributo que representa o Código da mercadoria	String(3)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo "CÓDIGO" da relação "MERCADORIA" • Chave primária

Quantidade	Atributo que representa a quantidade da cada mercadoria em estoque	Int	Int	<ul style="list-style-type: none"> • Não-nulo
------------	--	-----	-----	--

USA_CAMINHÃO: Relação que armazena os dados do relacionamento entre “ENTREGADOR” e “CAMINHÃO”.

Atributo	Descrição	Tipo	Domínio	Descrições
CPFEntregador	Atributo que representa o CPF do entregador	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo “CPF” da relação “ENTREGADOR” • Chave primária
PlacaCaminhao	Atributo que representa a placa do caminhão	String(7)	String(7)	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo “PLACA” da relação “CAMINHÃO” • Chave primária

SECRETARIO_ESTOQUISTA_VENDA: Relação que armazena os dados do relacionamento ternário entre “SECRETÁRIO” , “ESTOQUISTA” e “VENDA”.

Atributo	Descrição	Tipo	Domínio	Descrições
CPFSecretário	Atributo que representa o CPF do secretário	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo “CPF” da relação “SECRETÁRIO” • Chave primária

CPFEstoquista	Atributo que representa o CPF do estoquista	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo "CPF" da relação "ESTOQUISTA" • Chave primária
CodVenda	Atributo que representa o código da venda	Int	Int	<ul style="list-style-type: none"> • Chave estrangeira que referencia o atributo "CODIGO" da relação "VENDA"

INFORMA_ENTREGADOR: Relação que armazena os dados do relacionamento entre "ESTOQUISTA" e "ENTREGADOR".

Atributo	Descrição	Tipo	Domínio	Descrições
CPFEstoquista	Atributo que representa o CPF do estoquista	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo "CPF" da relação "ESTOQUISTA" • Chave primária
CPFEntregador	Atributo que representa o CPF do entregador	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo "CPF" da relação "ENTREGADOR" • Chave primária

TELEFONE_SUPERVISOR: Relação que armazena os números dos telefones dos supervisores da empresa.

Atributo	Descrição	Tipo	Domínio	Descrições
CPFSupervisor	Atributo que representa o CPF do supervisor	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo “CPF” da relação “SUPERVISOR” • Chave primária
Telefone	Atributo que representa o telefone do supervisor	String(9)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave primária

TELEFONE_VENDEDOR: Relação que armazena os números dos telefones dos vendedores da empresa.

Atributo	Descrição	Tipo	Domínio	Descrições
CPFVendedor	Atributo que representa o CPF do vendedor	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo “CPF” da relação “VENDEDOR” • Chave primária
Telefone	Atributo que representa o telefone do vendedor	String(9)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave primária

TELEFONE_SECRETARIO: Relação que armazena os números dos telefones dos secretários da empresa.

Atributo	Descrição	Tipo	Domínio	Descrições
CPFSecretario	Atributo que representa	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que

	o CPF do secretário			referencia o atributo “CPF” da relação “SECRETARIO” <ul style="list-style-type: none"> • Chave primária
Telefone	Atributo que representa o telefone do secretário	String(9)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave primária

TELEFONE_ESTOQUISTA: Relação que armazena os números dos telefones dos estoquistas da empresa.

Atributo	Descrição	Tipo	Domínio	Descrições
CPFEstoquista	Atributo que representa o CPF do estoquista	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo “CPF” da relação “ESTOQUISTA” • Chave primária
Telefone	Atributo que representa o telefone do estoquista	String(9)	Números inteiros positivos	<ul style="list-style-type: none"> • Chave primária

TELEFONE_ENTREGADOR: Relação que armazena os números dos telefones dos entregadores da empresa.

Atributo	Descrição	Tipo	Domínio	Descrições
CPFEntregador	Atributo que representa o CPF do entregador	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> • Não-nulo • Chave estrangeira que referencia o atributo “CPF” da relação “ENTREGADOR” • Chave primária

Telefone	Atributo que representa o telefone do entregador	String(9)	Números inteiros positivos	<ul style="list-style-type: none"> Chave primária
----------	--	-----------	----------------------------	--

CIDADES_VENDEDOR: Relação que armazena os nomes das cidades em que cada vendedor trabalha.

Atributo	Descrição	Tipo	Domínio	Descrições
CPFVendedor	Atributo que representa o CPF do vendedor	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> Não-nulo Chave estrangeira que referencia o atributo "CPF" da relação "VENDEDOR" Chave primária
Cidade	Atributo que representa a cidade que o vendedor visita	String(30)	String(30)	<ul style="list-style-type: none"> Não-nulo Chave primária

TELEFONE_CLIENTE: Relação que armazena os números dos telefones dos clientes.

Atributo	Descrição	Tipo	Domínio	Descrições
CPFCliente	Atributo que representa o CPF do cliente	String(11)	Números inteiros positivos	<ul style="list-style-type: none"> Não-nulo Chave estrangeira que referencia o atributo "CPF" da relação "CLIENTE" Chave primária
Telefone	Atributo que representa o telefone do cliente	String(9)	Números inteiros positivos	<ul style="list-style-type: none"> Não-nulo Chave primária

2.2 Início da implementação

2.2.1. Scripts SQL (criação de tabelas)

```
CREATE TABLE SUPERVISOR(  
    Nome VARCHAR(30),  
    Sexo CHAR NOT NULL,  
    RG VARCHAR(7) UNIQUE NOT NULL,  
    CPF VARCHAR(11),  
    Salario REAL NOT NULL,  
    Nascimento Date NOT NULL,  
    PRIMARY KEY (CPF),  
    CHECK (Sexo='M' OR Sexo='F')  
);
```

```
CREATE TABLE VENDEDOR(  
    Nome VARCHAR(30) NOT NULL,  
    Sexo CHAR NOT NULL,  
    RG VARCHAR(7) UNIQUE NOT NULL,  
    CPF VARCHAR(11),  
    Salario REAL NOT NULL,  
    Nascimento Date NOT NULL,  
    CNH VARCHAR(11) UNIQUE NOT NULL,  
    Codigo VARCHAR(4) UNIQUE NOT NULL,  
    CPFSupervisor VARCHAR(11),  
    Volume int NOT NULL,  
    PRIMARY KEY (CPF),  
    CHECK (Sexo='M' OR Sexo='F'),  
    FOREIGN KEY (CPFSupervisor) REFERENCES
```

```
SUPERVISOR(CPF) ON DELETE RESTRICT ON  
UPDATE CASCADE  
);
```

```
CREATE TABLE SECRETARIO(  
    Nome VARCHAR(30),  
    Sexo CHAR NOT NULL,  
    RG VARCHAR(7) UNIQUE NOT NULL,  
    CPF VARCHAR(11),  
    Salario REAL NOT NULL,  
    Nascimento Date NOT NULL,  
    PRIMARY KEY (CPF),  
    CHECK (Sexo='M' OR Sexo='F')  
);
```

```
CREATE TABLE ESTOQUISTA(  
    Nome VARCHAR(30),  
    Sexo CHAR NOT NULL,  
    RG VARCHAR(7) UNIQUE NOT NULL,  
    CPF VARCHAR(11),  
    Salario REAL NOT NULL,  
    Nascimento Date NOT NULL,  
    PRIMARY KEY (CPF),  
    CHECK (Sexo='M' OR Sexo='F')  
);
```

```
CREATE TABLE ENTREGADOR(  
    Nome VARCHAR(30),  
    Sexo CHAR NOT NULL,
```



```
RG VARCHAR(7) UNIQUE NOT NULL,  
CPF VARCHAR(11),  
Salario REAL NOT NULL,  
Nascimento Date NOT NULL,  
CNH VARCHAR(11) UNIQUE NOT NULL,  
PRIMARY KEY (CPF),  
CHECK (Sexo='M' OR Sexo='F')  
);
```

```
CREATE TABLE CAMINHAO(  
Placa VARCHAR(7),  
Modelo VARCHAR(7) NOT NULL,  
Montadora VARCHAR(10) NOT NULL,  
Categoria VARCHAR(10) NOT NULL,  
PRIMARY KEY (Placa)  
);
```

```
CREATE TABLE MERCADORIA (  
Preço REAL NOT NULL,  
Codigo VARCHAR(3),  
Tipo VARCHAR(20) NOT NULL,  
Nome VARCHAR(20) NOT NULL,  
PRIMARY KEY (Codigo)  
);
```

```
CREATE TABLE CLIENTE(  
NomeEstabelecimento VARCHAR(30) NOT NULL,  
CPF VARCHAR(11),  
Rua VARCHAR(30) NOT NULL,
```

```
Bairro VARCHAR(30) NOT NULL,  
Cidade VARCHAR(30) NOT NULL,  
PRIMARY KEY (CPF)  
);
```

```
CREATE TABLE VENDA (  
Codigo int,  
Hora TIME NOT NULL,  
Cidade VARCHAR(20) NOT NULL,  
Data DATE NOT NULL,  
ValorTotal REAL NOT NULL,  
CPFVendedor VARCHAR(11) NOT NULL,  
Comissao REAL NOT NULL,  
CPFSecretario VARCHAR(11) NOT NULL,  
CPFCliente VARCHAR(11) NOT NULL,  
CPFEntregador VARCHAR(11) NOT NULL,  
PRIMARY KEY (Codigo),  
FOREIGN KEY (CPFVendedor) REFERENCES VENDEDOR(CPF)  
ON DELETE RESTRICT ON UPDATE CASCADE,  
FOREIGN KEY (CPFSecretario)REFERENCES SECRETARIO(CPF)  
ON DELETE RESTRICT ON UPDATE CASCADE,  
FOREIGN KEY (CPFCliente) REFERENCES CLIENTE(CPF)  
ON DELETE RESTRICT ON UPDATE CASCADE,  
FOREIGN KEY (CPFEntregador) REFERENCES ENTREGADOR(CPF)  
ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

```

CREATE TABLE TEM_MERCADORIA(
    CodVenda int,
    CodMercadoria VARCHAR(3),
    Quantidade int NOT NULL,
    PreçoUnitario REAL NOT NULL,
    PRIMARY KEY (CodVenda, CodMercadoria),
    FOREIGN KEY (CodVenda) REFERENCES VENDA(Codigo)
    ON DELETE RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY (CodMercadoria) REFERENCES MERCADORIA(Codigo)
    ON DELETE RESTRICT ON UPDATE CASCADE
);

```

```

CREATE TABLE CONTROLA_MERCADORIA(
    CPFEstoquista VARCHAR(11) NOT NULL,
    CodMercadoria VARCHAR(3) NOT NULL,
    Quantidade int NOT NULL,
    PRIMARY KEY (CPFEstoquista, CodMercadoria),
    FOREIGN KEY (CPFEstoquista) REFERENCES ESTOQUISTA(CPF)
    ON DELETE RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY (CodMercadoria) REFERENCES MERCADORIA(Codigo)
    ON DELETE RESTRICT ON UPDATE CASCADE
);

```

```

CREATE TABLE USA_CAMINHAO(
    PlacaCaminhao VARCHAR(7),
    CPFEntregador VARCHAR(11),
    PRIMARY KEY (PlacaCaminhao, CPFEntregador),
    FOREIGN KEY (PlacaCaminhao) REFERENCES CAMINHAO(Placa)
    ON DELETE RESTRICT ON UPDATE CASCADE,

```

```
FOREIGN KEY (CPFEntregador) REFERENCES ENTREGADOR(CPF)
ON DELETE RESTRICT ON UPDATE CASCADE
);
```

```
CREATE TABLE SECRETARIO_ESTOQUISTA_VENDA(
    CPFSecretario VARCHAR(11),
    CPFEstoquista VARCHAR(11),
    CodVenda Int,
    PRIMARY KEY (CPFSecretario, CPFEstoquista),
    FOREIGN KEY (CPFSecretario) REFERENCES SECRETARIO(CPF)
    ON DELETE RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY(CPFEstoquista) REFERENCES ESTOQUISTA(CPF)
    ON DELETE RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY (CodVenda) REFERENCES VENDA(Codigo)
    ON DELETE RESTRICT ON UPDATE CASCADE
);
```

```
CREATE TABLE INFORMA_ENTREGADOR(
    CPFEstoquista VARCHAR(11),
    CPFEntregador VARCHAR(11),
    PRIMARY KEY (CPFEstoquista, CPFEntregador),
    FOREIGN KEY (CPFEstoquista) REFERENCES ESTOQUISTA(CPF)
    ON DELETE RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY (CPFEntregador) REFERENCES ENTREGADOR(CPF)
    ON DELETE RESTRICT ON UPDATE CASCADE );
```

```
CREATE TABLE TELEFONE_SUPERVISOR(
    CPFSupervisor VARCHAR(11),
    Telefone VARCHAR(9),
```

```
PRIMARY KEY (CPFSupervisor, Telefone),  
FOREIGN KEY (CPFSupervisor) REFERENCES  
SUPERVISOR(CPF) ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

```
CREATE TABLE TELEFONE_VENDEDOR(  
CPFVendedor VARCHAR(11),  
Telefone VARCHAR(9),  
PRIMARY KEY (CPFVendedor, Telefone),  
FOREIGN KEY (CPFVendedor) REFERENCES  
VENDEDOR(CPF) ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

```
CREATE TABLE TELEFONE_SECRETARIO(  
CPFSecretario VARCHAR(11),  
Telefone VARCHAR(9),  
PRIMARY KEY (CPFSecretario, Telefone),  
FOREIGN KEY (CPFSecretario) REFERENCES  
SECRETARIO(CPF) ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

```
CREATE TABLE TELEFONE_ESTOQUISTA(  
CPFEstoquista VARCHAR(11),  
Telefone VARCHAR(9),  
PRIMARY KEY (CPFEstoquista, Telefone),  
FOREIGN KEY (CPFEstoquista) REFERENCES  
ESTOQUISTA(CPF) ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

```
CREATE TABLE TELEFONE_ENTREGADOR(  
    CPFEntregador VARCHAR(11),  
    Telefone VARCHAR(9),  
    PRIMARY KEY (CPFEntregador, Telefone),  
    FOREIGN KEY (CPFEntregador) REFERENCES  
    ENTREGADOR(CPF) ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

```
CREATE TABLE CIDADES_VENDEDOR(  
    CPFVendedor VARCHAR(11),  
    Cidade VARCHAR(30),  
    PRIMARY KEY (CPFVendedor, Cidade),  
    FOREIGN KEY (CPFVendedor) REFERENCES VENDEDOR(CPF)  
    ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

```
CREATE TABLE TELEFONE_CLIENTE(  
    CPFCliente VARCHAR(11),  
    Telefone VARCHAR(9),  
    PRIMARY KEY (CPFCliente, Telefone),  
    FOREIGN KEY (CPFCliente) REFERENCES CLIENTE(CPF)  
    ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

2.2.2. Scripts SQL (povoamento)

Tabela SUPERVISOR:

```
INSERT INTO SUPERVISOR(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Flavio Henrique', 'M','4132793','10902340409',2500,'28/09/1998');
INSERT INTO SUPERVISOR(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Diego Gonçalves', 'M','1234567','12804144408',3000,'11/11/1995');
INSERT INTO SUPERVISOR(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Mickael Pereira', 'M','7654321','11122233344',2900,'07/07/1995');
INSERT INTO SUPERVISOR(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Maiary Pereira','F','1081243','77788899910',2800,'12/07/1992');
INSERT INTO SUPERVISOR(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Arlene Santana','F','4734737','22134114302',3100,'12/11/1973');
```

Tabela VENDEDOR:

```
INSERT INTO VENDEDOR (Nome, Sexo, RG, CPF, Salario, Nascimento,
CNH, Codigo, CPFSupervisor, Volume) VALUES ('Flavio Flor', 'M',
'1241465','71384758553',1800,'08/05/1970','04307558988','1111','1090234040
9',35);
INSERT INTO VENDEDOR (Nome, Sexo, RG, CPF, Salario, Nascimento,
CNH, Codigo, CPFSupervisor, Volume) VALUES ('Anderson Sá', 'M',
'9988776','50540430320',2000,'19/06/1998','17642132301','1112','1280414440
8',50);
INSERT INTO VENDEDOR (Nome, Sexo, RG, CPF, Salario, Nascimento,
CNH, Codigo, CPFSupervisor, Volume) VALUES ('Ricardo Lucas', 'M',
'6677889','20230340455',1750,'23/09/1985','11123478925','1113','1112223334
4',130);
INSERT INTO VENDEDOR (Nome, Sexo, RG, CPF, Salario, Nascimento,
CNH, Codigo, CPFSupervisor, Volume) VALUES ('Saulo Victor', 'M',
'5641421','35485748317',2200,'01/02/1981','88985570111','1114','7778889991
0',120);
```

```
INSERT INTO VENDEDOR (Nome, Sexo, RG, CPF, Salario, Nascimento,
CNH, Codigo, CPFSupervisor, Volume) VALUES ('Jonathas Dias', 'M',
'5546678', '66177255409', 1750, '23/08/1991', '70102437555', '1115', '2213411430
2', 120);
```

Tabela SECRETARIO:

```
INSERT INTO SECRETARIO(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Acidilene Santana', 'F', '9815481', '02547274418', 2350, '03/05/1973');
INSERT INTO SECRETARIO(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Jaqueline Lira', 'F', '4051692', '70232244406', 4000, '12/07/1996');
INSERT INTO SECRETARIO(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Juliana Lira', 'F', '4062783', '70132144401', 3500, '08/03/1998');
INSERT INTO SECRETARIO(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Beatriz Sarmento', 'F', '5435411', '80155112167', 2000, '12/09/1998');
INSERT INTO SECRETARIO(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Kaiky Sarmento', 'M', '6546522', '80255533450', 1800, '24/06/1998');
```

Tabela ESTOQUISTA:

```
INSERT INTO ESTOQUISTA(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES (Fábio Andrade', 'M', '7787576', '10802840808', 2700, '01/01/1981');
INSERT INTO ESTOQUISTA(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Francinete Flor', 'F', '3131313', '10902930909', 2510, '02/02/1982');
INSERT INTO ESTOQUISTA(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Jefferson Pereira', 'M', '4141415', '10540502545', 1950, '03/03/1983');
INSERT INTO ESTOQUISTA(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Fernanda Flor', 'F', '2021279', '20120244402', 2440, '04/04/1984');
INSERT INTO ESTOQUISTA(Nome, Sexo, RG, CPF, Salario, Nascimento)
VALUES ('Davi Gomes', 'M', '9101299', '30130240403', 3000, '05/05/1985');
```


Tabela ENTREGADOR:

```
INSERT INTO ENTREGADOR(Nome, Sexo, RG, CPF, Salario, Nascimento, CNH) VALUES ('Lucas Moura', 'M', '1122334', '60232266606', 4500, '21/07/1981', '01020304051');
```

```
INSERT INTO ENTREGADOR(Nome, Sexo, RG, CPF, Salario, Nascimento, CNH) VALUES ('Anderson Silva', 'M', '4455667', '70242144405', 3800, '22/07/1982', '01234567891');
```

```
INSERT INTO ENTREGADOR(Nome, Sexo, RG, CPF, Salario, Nascimento, CNH) VALUES ('Jessica Martins', 'F', '7788990', '80190110505', 3050, '23/07/1983', '08543217891');
```

```
INSERT INTO ENTREGADOR(Nome, Sexo, RG, CPF, Salario, Nascimento, CNH) VALUES ('Jheiziany Ruth', 'F', '0102030', '80244261601', 2990, '24/07/1984', '04140515936');
```

```
INSERT INTO ENTREGADOR(Nome, Sexo, RG, CPF, Salario, Nascimento, CNH) VALUES ('Leonardo Deca', 'M', '1021036', '50182171702', 2810, '25/07/1985', '01213141867');
```

Tabela CAMINHAO:

```
INSERT INTO CAMINHAO (Placa, Modelo, Montadora, Categoria) VALUES ('NQD1745', '24250', 'Volks', 'Semipesado');
```

```
INSERT INTO CAMINHAO (Placa, Modelo, Montadora, Categoria) VALUES ('NPT1213', '8150', 'Volks', 'Leve');
```

```
INSERT INTO CAMINHAO (Placa, Modelo, Montadora, Categoria) VALUES ('GRU4568', '710', 'Mercedes', 'Leve');
```

```
INSERT INTO CAMINHAO (Placa, Modelo, Montadora, Categoria) VALUES ('ARE2356', '9150', 'Volks', 'Leve');
```

```
INSERT INTO CAMINHAO (Placa, Modelo, Montadora, Categoria) VALUES ('YOU6789', '13180', 'Volks', 'Médio');
```

```
INSERT INTO CAMINHAO (Placa, Modelo, Montadora, Categoria) VALUES ('OLR5500', 'FH460', 'Volvo', 'Pesado');
```

Tabela MERCADORIA:

```
INSERT INTO MERCADORIA(Preço, Codigo, Tipo, Nome) VALUES  
(4.1,'001','Cerveja','Devassa');
```

```
INSERT INTO MERCADORIA(Preço, Codigo, Tipo, Nome) VALUES  
(16.65,'002','Cerveja','BadenBaden');
```

```
INSERT INTO MERCADORIA(Preço, Codigo, Tipo, Nome) VALUES  
(5.41,'003','Cerveja','Eisenbahn');
```

```
INSERT INTO MERCADORIA(Preço, Codigo, Tipo, Nome) VALUES  
(1.70,'004','Suco','Skinka');
```

```
INSERT INTO MERCADORIA(Preço, Codigo, Tipo, Nome) VALUES  
(1.43,'005','Água','Schin');
```

```
INSERT INTO MERCADORIA(Preço, Codigo, Tipo, Nome) VALUES  
(4.41,'006','Energético','K');
```

Tabela CLIENTE:

```
INSERT INTO CLIENTE(NomeEstabelecimento,CPF,Rua,Bairro,Cidade)  
VALUES('Super Félix','12356798341','Francisco Cartaxo Corrêa  
Sá','Estação','Sousa-PB');
```

```
INSERT INTO CLIENTE(NomeEstabelecimento,CPF,Rua,Bairro,Cidade)  
VALUES('Franco Mercantil','12343255507','Capitão Israel','Centro','Uiraúna-  
PB');
```

```
INSERT INTO CLIENTE(NomeEstabelecimento,CPF,Rua,Bairro,Cidade)  
VALUES('Churrascaria do Motor','23356688808','Júlio Marquês  
Nascimento','Lot. dos Médicos','Cajazeiras-PB');
```

```
INSERT INTO CLIENTE(NomeEstabelecimento,CPF,Rua,Bairro,Cidade)  
VALUES('Ponto 8','56787644405','Júlio Marquês Nascimento','Lot. Parque  
Rosina Parente','Cajazeiras-PB');
```

```
INSERT INTO CLIENTE(NomeEstabelecimento,CPF,Rua,Bairro,Cidade)  
VALUES('Bom Preço','10876544406','Firmino Tomaz de Aquino','Centro','Bom  
Jesus-PB');
```

Tabela VENDA:

```
INSERT INTO VENDA (Codigo, Hora, Cidade, Data, ValorTotal, CPFVendedor,
Comissao, CPFSecretario, CPFCliente, CPFEntregador)
VALUES(111,'12:30','Sousa-
PB','22/12/2016',2000,'71384758553',100,'02547274418','12356798341','70242
144405');
```

```
INSERT INTO VENDA (Codigo, Hora, Cidade, Data, ValorTotal, CPFVendedor,
Comissao, CPFSecretario, CPFCliente, CPFEntregador)
VALUES(112,'16:20','Bom Jesus-
PB','05/08/2017',250,'50540430320',12.50,'02547274418','10876544406','7024
2144405');
```

```
INSERT INTO VENDA (Codigo, Hora, Cidade, Data, ValorTotal, CPFVendedor,
Comissao, CPFSecretario, CPFCliente, CPFEntregador)
VALUES(113,'09:40','Cajazeiras-
PB','12/06/2017',290,'20230340455',5.8,'70232244406','23356688808','802442
61601');
```

```
INSERT INTO VENDA (Codigo, Hora, Cidade, Data, ValorTotal, CPFVendedor,
Comissao, CPFSecretario, CPFCliente, CPFEntregador)
VALUES(114,'11:13','Cajazeiras-
PB','03/07/2017',450,'20230340455',90,'70232244406','56787644405','5018217
1702');
```

```
INSERT INTO VENDA (Codigo, Hora, Cidade, Data, ValorTotal, CPFVendedor,
Comissao, CPFSecretario, CPFCliente, CPFEntregador)
VALUES(115,'13:45','Uiraúna-
PB','19/08/2017',1500,'66177255409',30,'70132144401','12343255507','801901
10505');
```

Tabela TEM_MERCADORIA:

```
INSERT INTO
TEM_MERCADORIA(CodVenda,CodMercadoria,Quantidade,PreçoUnitario)
VALUES(111,'001',20,4.1);
```

```
INSERT INTO
TEM_MERCADORIA(CodVenda,CodMercadoria,Quantidade,PreçoUnitario)
VALUES(111,'002',50,16.60);
```

```
INSERT INTO  
TEM_MERCADORIA(CodVenda,CodMercadoria,Quantidade,PreçoUnitario)  
VALUES(112,'005',50,1.43);
```

```
INSERT INTO  
TEM_MERCADORIA(CodVenda,CodMercadoria,Quantidade,PreçoUnitario)  
VALUES(113,'004',100,1.70);
```

```
INSERT INTO  
TEM_MERCADORIA(CodVenda,CodMercadoria,Quantidade,PreçoUnitario)  
VALUES(114,'003',40,5.41);
```

Tabela CONTROLA_MERCADORIA:

```
INSERT INTO  
CONTROLA_MERCADORIA(CPFEstoquista,CodMercadoria,Quantidade)  
VALUES('10802840808','002',2000);
```

```
INSERT INTO  
CONTROLA_MERCADORIA(CPFEstoquista,CodMercadoria,Quantidade)  
VALUES('10902930909','001',3500);
```

```
INSERT INTO  
CONTROLA_MERCADORIA(CPFEstoquista,CodMercadoria,Quantidade)  
VALUES('10540502545','001',1400);
```

```
INSERT INTO  
CONTROLA_MERCADORIA(CPFEstoquista,CodMercadoria,Quantidade)  
VALUES('20120244402','003',4300);
```

```
INSERT INTO  
CONTROLA_MERCADORIA(CPFEstoquista,CodMercadoria,Quantidade)  
VALUES('30130240403','005',550);
```

Tabela USA_CAMINHAO:

```
INSERT INTO USA_CAMINHAO(PlacaCaminhao,CPFEntregador)
VALUES('NQD1745','50182171702');
```

```
INSERT INTO USA_CAMINHAO(PlacaCaminhao,CPFEntregador)
VALUES('NPT1213','80244261601');
```

```
INSERT INTO USA_CAMINHAO(PlacaCaminhao,CPFEntregador)
VALUES('GRU4568','80244261601');
```

```
INSERT INTO USA_CAMINHAO(PlacaCaminhao,CPFEntregador)
VALUES('ARE2356','60232266606');
```

```
INSERT INTO USA_CAMINHAO(PlacaCaminhao,CPFEntregador)
VALUES('YOU6789','70242144405');
```

Tabela SECRETARIO_ESTOQUISTA_VENDA:

```
INSERT INTO
SECRETARIO_ESTOQUISTA_VENDA(CPFSecretario,CPFEstoquista,CodVen
da) VALUES('80255533450','10540502545',111);
```

```
INSERT INTO
SECRETARIO_ESTOQUISTA_VENDA(CPFSecretario,CPFEstoquista,CodVen
da) VALUES('80255533450','20120244402',112);
```

```
INSERT INTO
SECRETARIO_ESTOQUISTA_VENDA(CPFSecretario,CPFEstoquista,CodVen
da) VALUES('80155112167','10802840808',113);
```

```
INSERT INTO
SECRETARIO_ESTOQUISTA_VENDA(CPFSecretario,CPFEstoquista,CodVen
da)VALUES('70132144401','30130240403',114);
```

```
INSERT INTO
SECRETARIO_ESTOQUISTA_VENDA(CPFSecretario,CPFEstoquista,CodVen
da)VALUES('70132144401','10902930909',115);
```

Tabela INFORMA_ENTREGADOR:

```
INSERT INTO INFORMA_ENTREGADOR (CPFEstoquista, CPFEntregador)
VALUES('30130240403','80190110505');
INSERT INTO INFORMA_ENTREGADOR (CPFEstoquista, CPFEntregador)
VALUES('20120244402','60232266606');
INSERT INTO INFORMA_ENTREGADOR (CPFEstoquista, CPFEntregador)
VALUES('20120244402','80244261601');
INSERT INTO INFORMA_ENTREGADOR (CPFEstoquista, CPFEntregador)
VALUES('10540502545','50182171702');
INSERT INTO INFORMA_ENTREGADOR (CPFEstoquista, CPFEntregador)
VALUES('10902930909','60232266606');
```

Tabela CIDADES_VENDEDOR:

```
INSERT INTO CIDADES_VENDEDOR(CPFVendedor,Cidade)
VALUES('71384758553','Sousa-PB');
INSERT INTO CIDADES_VENDEDOR(CPFVendedor,Cidade)
VALUES('71384758553','Bom Jesus-PB');
INSERT INTO CIDADES_VENDEDOR(CPFVendedor,Cidade)
VALUES('35485748317','Uiraúna-PB');
INSERT INTO CIDADES_VENDEDOR(CPFVendedor,Cidade)
VALUES('20230340455','Cajazeiras-PB');
INSERT INTO CIDADES_VENDEDOR(CPFVendedor,Cidade)
VALUES('50540430320','Marizópolis-PB');
```

Tabela TELEFONE_CLIENTE:

```
INSERT INTO TELEFONE_CLIENTE(CPFCliente,Telefone)
VALUES('12356798341','991023212');
```

```
INSERT INTO TELEFONE_CLIENTE(CPFCliente,Telefone)
VALUES('12356798341','982456678');
```

```
INSERT INTO TELEFONE_CLIENTE(CPFCliente,Telefone)
VALUES('12343255507','981209890');
```

```
INSERT INTO TELEFONE_CLIENTE(CPFCliente,Telefone)
VALUES('23356688808','991030405');
```

```
INSERT INTO TELEFONE_CLIENTE(CPFCliente,Telefone)
VALUES('56787644405','991212425');
```

Tabela TELEFONE_SUPERVISOR:

```
INSERT INTO TELEFONE_SUPERVISOR(CPFSupervisor,Telefone)
VALUES('10902340409','991641456');
```

```
INSERT INTO TELEFONE_SUPERVISOR(CPFSupervisor,Telefone)
VALUES('10902340409','991823454');
```

```
INSERT INTO TELEFONE_SUPERVISOR(CPFSupervisor,Telefone)
VALUES('12804144408','991878543');
```

```
INSERT INTO TELEFONE_SUPERVISOR(CPFSupervisor,Telefone)
VALUES('11122233344','981232321');
```

```
INSERT INTO TELEFONE_SUPERVISOR(CPFSupervisor,Telefone)
VALUES('11122233344','993456578');
```

Tabela TELEFONE_VENDEDOR:

```
INSERT INTO TELEFONE_VENDEDOR(CPFVendedor, Telefone)
VALUES('71384758553','991878667');
```

```
INSERT INTO TELEFONE_VENDEDOR(CPFVendedor, Telefone)
VALUES('35485748317','982343009');
INSERT INTO TELEFONE_VENDEDOR(CPFVendedor, Telefone)
VALUES('71384758553','982767879');
INSERT INTO TELEFONE_VENDEDOR(CPFVendedor, Telefone)
VALUES('71384758553','993435215');
INSERT INTO TELEFONE_VENDEDOR(CPFVendedor, Telefone)
VALUES('66177255409','993515216');
```

Tabela TELEFONE_SECRETARIO:

```
INSERT INTO TELEFONE_SECRETARIO(CPFSecretario, Telefone)
VALUES('02547274418','991080706');
INSERT INTO TELEFONE_SECRETARIO(CPFSecretario, Telefone)
VALUES('70232244406','982343789');
INSERT INTO TELEFONE_SECRETARIO(CPFSecretario, Telefone)
VALUES('70132144401','991654567');
INSERT INTO TELEFONE_SECRETARIO(CPFSecretario, Telefone)
VALUES('80155112167','991232324');
INSERT INTO TELEFONE_SECRETARIO(CPFSecretario, Telefone)
VALUES('80255533450','991042321');
```

Tabela TELEFONE_ESTOQUISTA:

```
INSERT INTO TELEFONE_ESTOQUISTA(CPFEstoquista, Telefone)
VALUES('30130240403','991121314');
INSERT INTO TELEFONE_ESTOQUISTA(CPFEstoquista, Telefone)
VALUES('20120244402','991151617');
INSERT INTO TELEFONE_ESTOQUISTA(CPFEstoquista, Telefone)
```



```
VALUES('10540502545','981232425');  
INSERT INTO TELEFONE_ESTOQUISTA(CPFEstoquista,Telefone)  
VALUES('10902930909','982343536');  
INSERT INTO TELEFONE_ESTOQUISTA(CPFEstoquista,Telefone)  
VALUES('10802840808','982767532');
```

Tabela TELEFONE_ENTREGADOR:

```
INSERT INTO TELEFONE_ENTREGADOR(CPFEntregador,Telefone)  
VALUES('80244261601','991565744');  
INSERT INTO TELEFONE_ENTREGADOR(CPFEntregador,Telefone)  
VALUES('80244261601','981233009');  
INSERT INTO TELEFONE_ENTREGADOR(CPFEntregador,Telefone)  
VALUES('70242144405','991232115');  
INSERT INTO TELEFONE_ENTREGADOR(CPFEntregador,Telefone)  
VALUES('60232266606','981237655');  
INSERT INTO TELEFONE_ENTREGADOR(CPFEntregador,Telefone)  
VALUES('50182171702','982343007');
```

3ª Etapa

Projeto da disciplina Banco de Dados I

3.1 Conclusão da implementação

3.1.1. Scripts SQL (índices)

1-índice para o atributo CPF da relação Supervisor.

```
CREATE INDEX ON SUPERVISOR(CPF);
```

2-índice para o atributo Codigo da relação Vendedor.

```
CREATE INDEX ON VENDEDOR(Codigo);
```

3-índice para o atributo CPF da relação Secretario.

```
CREATE INDEX ON SECRETARIO(CPF);
```

4-índice para o atributo CPF da relação Estoquista.

```
CREATE INDEX ON ESTOQUISTA(CPF);
```

5-índice para o atributo CPF da relação Entregador.

```
CREATE INDEX ON ENTREGADOR(CPF);
```

6-índice para o atributo Placa da relação Caminhao.

```
CREATE INDEX ON CAMINHAO(Placa);
```

7-índice para o atributo CPF da relação Cliente.

```
CREATE INDEX ON CLIENTE(CPF);
```

8-índice para o atributo Codigo da relação Mercadoria.

```
CREATE INDEX ON MERCADORIA(Codigo);
```

9-índice para o atributo Codigo da relação Venda.

```
CREATE INDEX ON VENDA(Codigo);
```

10-índice para o atributo CPFVendedor da relação Cidades_Vendedor.

```
CREATE INDEX ON CIDADES_VENDEDOR(CPFVendedor);
```

11-índice para o atributo CodMercadoria da relação Controla_Mercadoria.

```
CREATE INDEX ON CONTROLA_MERCADORIA(CodMercadoria);
```

12-índice para o atributo CPFEstoquista da relação Informa_Entregador.

```
CREATE INDEX ON INFORMA_ENTREGADOR(CPFEstoquista);
```

13-índice para o atributo CPFEstoquista da relação
Secretario_Estoquista_Venda.

```
CREATE INDEX ON SECRETARIO_ESTOQUISTA_VENDA(CPFEstoquista);
```

14-índice para o atributo Telefone da relação Telefone_Cliente.

```
CREATE INDEX ON TELEFONE_CLIENTE(Telefone);
```

15-índice para o atributo Telefone da relação Telefone_Entregador.

```
CREATE INDEX ON TELEFONE_ENTREGADOR(Telefone);
```

16-índice para o atributo Telefone da relação Telefone_Estoquista.

```
CREATE INDEX ON TELEFONE_ESTOQUISTA(Telefone);
```

17-índice para o atributo Telefone da relação Telefone_Secretario.

```
CREATE INDEX ON TELEFONE_SECRETARIO(Telefone);
```

18-índice para o atributo Telefone da relação Telefone_Supervisor.

```
CREATE INDEX ON TELEFONE_SUPERVISOR(Telefone);
```

19-índice para o atributo Telefone da relação Telefone_Vendedor.

```
CREATE INDEX ON TELEFONE_VENDEDOR(Telefone);
```

20-índice para o atributo CodVenda da relação Tem_Mercadoria.

```
CREATE INDEX ON TEM_MERCADORIA(CodVenda);
```

21-índice para o atributo PlacaCaminhao da relação Usa_Caminhao.

```
CREATE INDEX ON USA_CAMINHAO(PlacaCaminhao);
```

3.1.2. Scripts SQL (consultas)

1 - O nome, CPF e salário de todos os vendedores que trabalham na cidade Sousa.

```
SELECT V.Nome, V.Cpf, V.Salario  
  
FROM CIDADES_VENDEDOR CV, VENDEDOR V  
  
WHERE V.CPF=CV.CPFVendedor AND CV.Cidade LIKE 'Sousa%'
```

2- Todos os dados dos estoquistas que controlam a mercadoria com código '001', ordenados pelo salário em ordem decrescente.

```
SELECT E.*  
  
FROM ESTOQUISTA E, CONTROLA_MERCADORIA CM  
  
WHERE E.CPF=CM.CPFEstoquista AND CM.CodMercadoria='001'  
  
ORDER BY E.Salario DESC
```

3- A quantidade de vendas com valor total acima de R\$ 300,00.

```
SELECT COUNT(*) AS Quantidade  
  
FROM VENDA  
  
WHERE ValorTotal>300
```

4- O nome, código e preço de todas as mercadorias do tipo "Cerveja", ordenados em ordem alfabética.

```
SELECT Nome, Codigo, Preço  
  
FROM MERCADORIA  
  
WHERE Tipo='Cerveja' ORDER BY Nome
```

5- O telefone dos clientes que estão na cidade Cajazeiras.

```
SELECT Telefone  
FROM Telefone_Cliente  
WHERE CPFCliente IN  
(SELECT cpf  
FROM CLIENTE  
where cidade like 'Cajazeiras%')
```

**6- O código dos vendedores supervisionados pelo supervisor de RG
'4132793'.**

```
SELECT Codigo  
FROM Vendedor  
WHERE CpfSupervisor IN  
(SELECT Cpf  
FROM Supervisor  
WHERE rg='4132793')
```

**7- O modelo, a montadora e a placa de todos os caminhões de categoria
"Leve".**

```
SELECT Placa,Modelo,Montadora  
FROM CAMINHAO  
WHERE Categoria = 'Leve'
```

8- O código e valor total de todas as vendas conferidas pelo secretário com CPF '02547274418'.

```
SELECT Codigo,ValorTotal  
FROM Venda  
WHERE CPFSecretario='02547274418'
```

9- O valor acumulado em todas as vendas registradas.

```
SELECT SUM(ValorTotal) AS ValorAcumulado  
FROM VENDA  
WHERE ValorTotal IS NOT NULL
```

10- O nome do estabelecimento e CPF de todos os clientes com o nome do estabelecimento iniciado pela letra 'F'.

```
SELECT NomeEstabelecimento, CPF  
FROM CLIENTE  
WHERE NomeEstabelecimento LIKE 'F%'  
AND NomeEstabelecimento IS NOT NULL
```

11- Para cada cidade em que existem clientes, recuperar o nome da cidade e a quantidade de clientes.

```
SELECT Cidade,COUNT(*) as numDeClientes  
FROM Cliente  
GROUP BY Cidade
```


12- Recuperar o nome da cidade, a quantidade de vendas realizadas e a soma dos valores totais das vendas de cada cidade que tenha mais de uma venda registrada.

```
SELECT Cidade, COUNT(*) AS Quantidade, SUM(valortotal) AS Valortotal  
FROM Venda  
GROUP BY Cidade  
HAVING COUNT(*)>1
```

13- O nome e o CPF dos entregadores que utilizaram caminhões.

```
SELECT E.Nome, E.CPF  
FROM ENTREGADOR E  
WHERE EXISTS(  
SELECT *  
FROM USA_CAMINHAO UC  
WHERE UC.CPFEntregador=E.CPF  
)
```

14- O nome e o CPF dos secretários que ainda não informaram nenhum estoquista sobre vendas.

```
SELECT S.Nome,S.CPF  
FROM SECRETARIO S  
WHERE NOT EXISTS  
(SELECT *  
FROM SECRETARIO_ESTOQUISTA_VENDA SEV  
WHERE SEV.CPFSecretario=S.CPF)
```

3.1.3. Scripts SQL (visões)

1-O nome do estabelecimento do cliente e o valor de todas as vendas realizadas na cidade Cajazeiras, ordenados pelo valor total da venda.

```
CREATE VIEW VendaClientes AS  
  
SELECT V.Cidade AS Cidade, C.NomeEstabelecimento AS  
Estabelecimento,V.ValorTotal AS Valor  
  
FROM CLIENTE C, VENDA V  
  
WHERE C.Cpf=V.CpfCliente AND V.Cidade LIKE 'Cajazeiras%'  
  
ORDER BY V.ValorTotal DESC
```

2-O nome, o código e a quantidade de cada mercadoria em estoque na empresa.

```
CREATE VIEW Estoque AS  
  
SELECT M.Nome, M.Codigo, SUM(CM.Quantidade) AS Quantidade  
  
FROM MERCADORIA M, CONTROLA_MERCADORIA CM  
  
WHERE M.Codigo=CM.CodMercadoria  
  
GROUP BY M.Codigo  
  
ORDER BY SUM(CM.Quantidade) DESC
```

3-O nome do entregador e a quantidade de vezes em que cada caminhão foi usado por ele.

```
CREATE VIEW ENTREGAS AS  
  
SELECT E.Nome, COUNT(*) AS Quantidade  
  
FROM USA_CAMINHAO UC, ENTREGADOR E  
  
WHERE E.CPF=UC.CPFEntregador  
  
GROUP BY E.CPF
```

3.1.4. Scripts SQL (procedimentos armazenados)

1- Retornar a quantidade de funcionários que tem salário maior do que um valor determinado.

```
CREATE FUNCTION SalarioFuncionarios(Real)
    RETURNS int AS $$
    DECLARE Sal Alias FOR $1;
    cont INTEGER:=0;
    Resultado Supervisor%RowType;
    BEGIN
        FOR Resultado IN SELECT * FROM SUPERVISOR
                        WHERE Salario>Sal
        LOOP
            Cont:=cont+1;
        END LOOP;
        FOR Resultado IN SELECT * FROM VENDEDOR
                        WHERE Salario>Sal
        LOOP
            Cont:=cont+1;
        END LOOP;
        FOR Resultado IN SELECT * FROM SECRETARIO
                        WHERE Salario>Sal
        LOOP
            Cont:=cont+1;
        END LOOP;
        FOR Resultado IN SELECT * FROM ESTOQUISTA
                        WHERE Salario>Sal
        LOOP
```

```

        Cont:=cont+1;

    END LOOP;

    FOR Resultado IN SELECT * FROM ENTREGADOR
                        WHERE Salario>Sal

    LOOP

        Cont:=cont+1;

    END LOOP;

    RETURN cont;

END;$$

LANGUAGE PLPGSQL

```

2- Recuperar o código da mercadoria mais requisitada em uma venda determinada.

```

CREATE FUNCTION MercadoriaAdquirida(INTEGER)
RETURNS VARCHAR(3) AS $$

DECLARE

    Codigo ALIAS FOR $1;

    quant integer :=0;

    resultado TEM_MERCADORIA%rowtype;

    merc varchar(3);

BEGIN

    FOR Resultado IN SELECT * FROM TEM_MERCADORIA WHERE
    CodVenda=Codigo

    LOOP

        IF resultado.quantidade>quant then

            begin

                quant:=resultado.quantidade;

                merc:=resultado.codmercadoria;

            end

```

```

                                end;

        END IF;

        END LOOP;

        RETURN merc;

END;$$

LANGUAGE PLPGSQL

```

3- Retornar o valor adquirido por uma mercadoria em uma determinada venda.

```

CREATE FUNCTION LUCRO(VARCHAR,INTEGER)
RETURNS NUMERIC AS $$
DECLARE
codigomerc alias for $1;
codigovenda alias for $2;
Valor Numeric;
BEGIN
    SELECT INTO Valor cast(quantidade*preçounitario as numeric(12,2))
    FROM TEM_MERCADORIA
    WHERE codigomerc=CodMercadoria AND codigovenda=codvenda;
    RETURN Valor;
END;$$

LANGUAGE PLPGSQL

```

3.1.5. Scripts SQL (gatilhos)

1- Designar o controle de 100 unidades da mercadoria de código '004' para cada novo estoquista na empresa.

```
CREATE FUNCTION ControlarMercadoria()  
RETURNS TRIGGER AS $$  
BEGIN  
    INSERT INTO CONTROLA_MERCADORIA  
        (CPFEstoquista,CodMercadoria,Quantidade)  
        VALUES(NEW.CPF,'004',100);  
    RETURN NULL;  
END;$$  
LANGUAGE PLPGSQL;
```

```
CREATE TRIGGER NovoEstoquista  
AFTER INSERT ON ESTOQUISTA  
FOR EACH ROW  
EXECUTE PROCEDURE ControlarMercadoria();
```

2- Ao remover um caminhão, devem ser apagados os registros de uso do mesmo.

```
CREATE FUNCTION RemoveCaminhao()  
RETURNS TRIGGER AS $$  
BEGIN  
    DELETE FROM USA_CAMINHAO WHERE PlacaCaminhao=OLD.Placa;  
    RETURN OLD;  
END;$$ LANGUAGE PLPGSQL;
```

```
CREATE TRIGGER ExcluirCaminhao  
BEFORE DELETE ON CAMINHAO  
FOR EACH ROW  
EXECUTE PROCEDURE RemoveCaminhao();
```

3- Quando um novo cliente for adicionado, o vendedor com CPF '66177255409' deve passar a trabalhar também na cidade do cliente novo, caso ele não esteja registrado nela.

```
CREATE FUNCTION CidadeCliente()  
RETURNS TRIGGER AS $$  
DECLARE  
    cont INTEGER:=0;  
BEGIN  
    SELECT INTO cont COUNT(*)  
    FROM CIDADES_VENDEDOR  
    WHERE CPFVendedor='66177255409' AND NEW.Cidade=Cidade;  
    IF cont<=0 THEN  
        INSERT INTO CIDADES_VENDEDOR(CPFVendedor,Cidade)  
        VALUES('66177255409',NEW.Cidade);  
    END IF;  
    RETURN NULL;  
END;$$  
LANGUAGE PLPGSQL;
```

```
CREATE TRIGGER NovoCliente  
AFTER INSERT ON CLIENTE  
FOR EACH ROW  
EXECUTE PROCEDURE CidadeCliente();
```