



Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus Cajazeiras
Análise e Desenvolvimento de Sistemas

Banco de Dados I

Ana Beatriz Queiroz de Almeida
Raquel Oliveira Ferreira
Darkiane de Lima Galdino
Gustavo Lopes Lemos

M&S Roletes e Quixabeiras Rolos
Banco de dados relacional

Cajazeiras-PB
2024

Sumário

1.	Introdução.....	03
2.	Modelo conceitual	03
2.1	Levantamento de requisitos	03
2.2	Diagrama de entidade-relacionamento	04
2.3	Consultas	04
2.4	Dicionário conceitual de dados	05
2.4.1	Entidades.....	05
2.4.2	Relacionamentos.....	07
3.	Modelo lógico	07
3.1	Mapeamento entidade-relacionamento	08
3.2	Dicionário lógico de dados.....	08
4.	Modelo físico	10
4.1	Scripts de criação das tabela.....	14
4.2	Scripts de criação dos índices.....	17
4.3	Scripts de criação das visões	18
4.4	Script de criação dos procedimentos armazenados.....	18
4.5	Script de criação dos gatilhos.....	19
4.6	Script de povoamento	21
4.7	Script de criação das consultas	24

1. Introdução

O sistema de banco de dados desenvolvido para as empresas M&S Roletes e Quixabeiras Rolos tem como objetivo controlar e gerenciar pagamentos de compras e vendas de produtos, controle de funcionários, entrada e saída de produtos no estoque, e detalhamento sobre as empresas envolvidas.

2. Modelo conceitual

2.1. Levantamento de requisitos

- A **empresa** pode ser uma **matriz** e **participante(fornecedora ou cliente)**. É necessário armazenar o nome e CNPJ de todas as empresas. Além disso, é necessário armazenar o endereço, telefones para contato e o tipo da empresa participante.
- Uma **empresa** pode realizar um **pagamento**, sendo necessário armazenar o valor de entrada, o ID, a data de pagamento e o tipo.
- Para cada uma das **empresas matriz** pode receber um **pagamento**, além de controlar informações sobre os **funcionários**;
- Para cada **funcionário** deve-se armazenar o nome, CPF, número da conta, Nome do banco, valor do salário, data do pagamento e função;
- Para cada **item** do estoque deve ser armazenado o código, categoria e nome;
- Cada **item** pode ser dividido em **produto** e **serviço**. O **produto** possui quantidade estoque;
- A **empresa participante** pode fazer ou fornecer um ou mais pedido;
- Cada **pedido** possui número do pedido, data do pedido, valor total, data prevista para a entrega, data da entrega, data da coleta, volume e peso;
- Deve-se armazenar os dados das **transportadoras**, o nome, CNPJ, destino, telefones, endereço e Inscrição estadual;
- Um **pedido** deve ser enviado por uma transportadora e uma transportadora deve transportar no mínimo um pedido. E deve estar relacionado a uma única **nota fiscal**.
- Cada **nota fiscal** deve conter a data da emissão, o valor total e o número.
- Uma **nota fiscal** deve estar relacionada a um único **pagamento**.
- Um **pagamento** possui id, valor entrada e data de pagamento, o tipo pode ser realizado à vista e PARCELADO.
- O pagamento tem **parcela** que deve receber número boleto, id, valor, data. Obrigatoriamente, o parcelamento é feito por boleto.

2.2. Diagrama entidade relacionamento

- Recuperar o nome dos funcionários e o nome da empresa matriz onde estão alocados;
- Recuperar o número dos pedidos junto com os nomes e CNPJs da transportadora e da empresa participante;
- Recuperar o nome das empresas participantes que não possuem um segundo telefone cadastrado;
- Recuperar o nome das empresas participantes que possuem os dois telefones cadastrados;
- Recuperar os dados dos funcionários cujo o nome comece com a letra “A”;
- Recuperar o nome das empresas com “Aço” no nome;
- Recuperar todos os pedidos, do mais recente para o mais antigo;
- Recuperar o nome dos funcionários ordenados pelo salário (maiores salários primeiro);
- Recuperar os itens com estoque abaixo da média;
- Recuperar os pedidos com valor total acima da média;
- Recuperar o nome das transportadoras que nunca realizaram pedidos;
- Recuperar os itens com a quantidade em estoque acima da média de sua categoria;
- Recuperar todos os CNPJs das empresas matriz e participante;
- Recuperar os funcionários com salário acima da média e abaixo da média;
- Recuperar o total de salários pagos aos funcionários de cada empresa matriz;
- Recuperar a média do valor das notas fiscais;
- Recuperar o número total de pedidos enviados por cada transportadora;

- Recuperar o total de itens vendidos por categoria exibindo apenas categorias com mais de 30 vendas;

2.4. Dicionário conceitual de dados

2.4.1. Entidades

Entidade Empresa:

- A entidade empresa é uma especialização generalizada de empresa matriz e participante.

Atributos:

- CNPJ: armazena o CNPJ (Cadastro Nacional da Pessoa Jurídica).
- Nome: armazena o nome da empresa, podendo ser a razão social.

Entidade Matriz:

- Entidade especializada que apenas herda os atributos da entidade Empresa, armazenando os dados da empresa matriz.

Entidade Participante:

- Entidade especializada que herda os atributos da entidade Empresa e armazena os dados relacionados às empresas que fazem e fornecem pedidos à matriz.

Atributos:

- Endereço: armazena o endereço da empresa fornecedora.
- Telefone1: armazena o primeiro telefone para contato da empresa fornecedora.
- Telefone2: armazena o segundo telefone para contato da empresa fornecedora.
- Tipo: armazena qual o tipo da empresa (cliente ou fornecedora).

Entidade Funcionário:

- Entidade utilizada para armazenar os dados dos funcionários que trabalham na empresa matriz.

Atributos:

- CPF: armazena o dado do CPF (Cadastro de Pessoa Física) do funcionário.
- Nome: armazena o nome do funcionário.
- Função: armazena a função exercida pelo funcionário na empresa.
- Data de Pagamento: armazena a data de pagamento do funcionário.
- Valor do salário: armazena o valor total do salário do funcionário.
- Número da conta: armazena o número da conta do banco do funcionário.
- Nome do Banco: armazena o nome do banco utilizado pelo funcionário.

Entidade Parcela:

- Entidade utilizada para armazenar dados referentes a uma parcela de um pagamento.

Atributos:

- Data: armazena a data do pagamento.

- Valor: armazena o valor a ser pago na parcela.
- ID: identificador do pagamento.
- Número do boleto: armazena o número do boleto da parcela a ser paga.

Entidade Transportadora:

- Entidade utilizada para armazenar os dados da transportadora que será utilizada.
Atributos:
- CNPJ: armazena o CNPJ da transportadora.
- Nome: armazena o nome ou razão social da transportadora.
- Endereço: armazena o endereço da empresa cliente.
- Telefone1: armazena o primeiro telefone para contato da transportadora.
- Telefone2: armazena o segundo telefone para contato da transportadora.
- Inscrição estadual: armazena a inscrição estadual da transportadora.

Entidade Nota Fiscal:

- Entidade utilizada para armazenar os dados das notas fiscais utilizadas nas compras e vendas.
Atributos:
- Número: armazena o número da nota fiscal.
- Valor total: armazena o valor total da nota fiscal.
- Data de emissão: armazena a data de emissão da nota fiscal.

Entidade Pagamento:

- Entidade utilizada para representar um pagamento a ser realizado.
Atributos:
- ID: armazena o identificador do pagamento realizado.
- Tipo: armazena se o pagamento é à vista ou a prazo.
- Valor de entrada: armazena o valor de entrada do pagamento, caso seja à vista, esse valor será correspondente ao total a ser pago.
- Data de pagamento: armazena a data de pagamento.

Entidade Pedido:

- A entidade pedido é utilizada para armazenar os valores referentes a um pedido feito em uma compra.

Atributos:

- Número do pedido: armazena o número identificador do pedido.
- Data: armazena a data em que o pedido foi realizado.
- Data prevista para entrega: armazena a possível data para a entrega.
- Data da entrega: armazena a data de entrega.
- Data da coleta: armazena a data da coleta realizada pela transportadora.
- Volume: armazena o volume total do pedido.
- Peso: armazena o peso total do pedido.

Entidade Item:

- A entidade item é uma especialização generalizada de Produto e Serviço.

Atributos:

- Código: armazena o código do item.
- Categoria: armazena a categoria do item.
- Nome: armazena o nome do item.

Entidade Produto:

- Entidade especializada de item que armazena os produtos no estoque.
- Atributos:
- Quantidade em estoque: armazena a quantidade do produto no estoque.

Entidade Serviço:

- Entidade especializada de item que representa os serviços que podem ser prestados pela empresa.

2.4.2. Relacionamentos

- Possui: relaciona “Empresa matriz” com “Funcionário” onde a “Empresa matriz” é responsável pelo “funcionário”.
- Controla: relaciona “Empresa matriz” com “Item” onde a “Empresa matriz” é responsável pelo controle dos itens em estoque.
- Recebe: relaciona “Empresa matriz” com “Pagamento” onde a “Empresa matriz” recebe pagamentos.
- Faz: relaciona “Empresa” com “Pagamento” onde a “empresa” é responsável por fazer um pagamento.
- Faz/fornece: relaciona “Empresa participante” com “Pedido” onde a “Empresa participante” faz ou fornece um pedido.
- Possui: relaciona “Pedido” com “Item” onde cada pedido possui item(produto ou serviço).
- Possui: relaciona “Pedido” com “Nota fiscal” onde cada pedido está relacionado a uma nota fiscal.
- Possui: relaciona “Pagamento” com “ Nota fiscal” onde o pagamento é direcionado a uma nota fiscal.
- Tem: relaciona “Pagamento” com “Parcela” onde um pagamento pode possuir nenhuma ou várias parcelas.
- Enviado por: relacionar “Pedido” com “Transportadora” onde um produto que foi vendido será transportado por uma transportadora.

3. Modelo lógico

3.1. Mapeamento entidade-relacionamento

Empresa_Matriz(CNPJ, Nome)

Empresa_Participante(CNPJ, Nome, Tipo, Telefone1, Telefone2, Endereço)

Funcionario(CPF, Nome, Função, Número_conta, Nome_banco, Valor_salário, Data_pagamento, CNPJ_empresa_matriz)

Nota_Fiscal(Número, Valor_total, Data_emissão)

Pagamento(ID, Tipo, Data_pagamento, Valor_entrada, CNPJ_empresa_matriz, CNPJ_empresa_participante, Número_nota_fiscal)

Parcela(ID, Valor, Data_vencimento, Número_boleto, Id_pagamento)

Transportadora(CNPJ, Nome, Inscrição_estadual, Telefone1, Telefone2, Endereço)

Pedido(Número_pedido, Data_pedido, Data_prevista_entrega, Data_coleta, Peso, Volume, Destino, Número_nota_fiscal, CNPJ_participante, CNPJ_transportadora)

Item(Código, Nome, Tipo, Categoria, Quantidade_estoque, CNPJ_empresa_matriz)

Pedido_possui_item(Número_pedido, Código_item, Preço_unitário, Quantidade)

3.2. Dicionário lógico de dados

Empresa_matriz

Atributo	Descrição	Tipo	Domínio	Restrição
CNPJ	CNPJ da empresa	CHAR(18)	CHAR(18)	- Chave primária
Nome	Nome da empresa	VARCHAR(100)	VARCHAR(100)	- Not Null

Empresa_participante

Atributo	Descrição	Tipo	Domínio	Restrição
CNPJ	CNPJ da empresa	CHAR(18)	CHAR(18)	- Chave primária
Nome	Nome da empresa	VARCHAR(100)	VARCHAR(100)	- Not Null

Empresa_participante

Tipo	Tipo da empresa	VARCHAR(20)	VARCHAR(20)	- Not Null
Telefone1	Telefone da empresa	VARCHAR(20)	VARCHAR(20)	- Not Null
Telefone2	Telefone secundário da empresa	VARCHAR(20)	VARCHAR(20)	
Endereço	Localização da empresa	VARCHAR(100)	VARCHAR(100)	- Not Null

Funcionário

Atributo	Descrição	Tipo	Domínio	Restrição
CNPJ	CNPJ da empresa	CHAR(18)	CHAR(18)	- Chave primária
Nome	Nome da empresa	VARCHAR(100)	VARCHAR(100)	- Not Null
Função	Função do funcionário na empresa	VARCHAR(50)	VARCHAR(50)	- Not Null
Numero_conta	Número da conta do funcionário	VARCHAR(20)	VARCHAR(20)	- Not Null
Nome_banco	Nome do banco do funcionário	VARCHAR(20)	VARCHAR(20)	- Not Null
Salário	Salário do funcionário	REAL	Números reais positivos	- Not Null
Data_pagamento	Data do pagamento do funcionário	DATE	DATE	- Not Null
CNPJ_empresa_matriz	CNPJ da empresa matriz	CHAR (18)	CHAR(18)	- Chave estrangeira

Nota_fiscal

Atributo	Descrição	Tipo	Domínio	Restrição
Numero	Número da nota fiscal	VARCHAR(50)	VARCHAR(50)	- Chave primária
Valor	Valor da nota fiscal	REAL	Números reais positivos	- Not Null
Data_emissao	Data da emissão da nota fiscal	DATE	DATE	- Not Null

Pagamento

Atributo	Descrição	Tipo	Domínio	Restrição
Id	Id do pagamento	VARCHAR(20)	VARCHAR(20)	- Chave primária
Tipo	Nome da empresa	VARCHAR(100)	VARCHAR(100)	- Not Null
Data_pagamento	Data do pagamento	DATE	DATE	
Valor_entrada	Valor de entrada do pagamento	REAL	Números reais positivos	
CNPJ_empresa_matriz	CNPJ da empresa matriz	CHAR(18)	CHAR(18)	- Chave estrangeira -Not Null
CNPJ_empresa_participante	CNPJ da empresa participante	CHAR(18)	CHAR(18)	- Chave estrangeira -Not Null
Numero_nota_fiscal	Número da nota fiscal	VARCHAR(50)	VARCHAR(50)	-Not Null

Parcela

Atributo	Descrição	Tipo	Domínio	Restrição
ID	Id da parcela	CHAR(18)	CHAR(18)	- Chave primária
Valor	Nome da empresa	VARCHAR(100)	VARCHAR(100)	- Not Null
Data_vencimento	Data de vencimento da parcela	DATE	DATE	- Not Null
Numero_boleto	Número do boleto da parcela	VARCHAR(25)	VARCHAR(25)	- Chave primária
ID_pagamento	Id do pagamento da parcela	VARCHAR(20)	VARCHAR(20)	-Chave estrangeira - Not Null

Transportadora

Atributo	Descrição	Tipo	Domínio	Restrição
CNPJ	CNPJ da empresa	CHAR(18)	CHAR(18)	- Chave primária
Nome	Nome da empresa	VARCHAR(100)	VARCHAR(100)	- Not Null
Inscricao_estadual	Inscrição estadual da transportadora	CHAR(12)	CHAR(12)	- Unique - Not Null
Telefone1	Telefone da transportadora	VARCHAR(20)	VARCHAR(20)	- Not Null
Telefone2	Telefone da transportadora	VARCHAR(20)	VARCHAR(20)	
Endereco	Endereço da transportadora	VARCHAR(100)	VARCHAR(100)	- Not Null

Pedido

Atributo	Descrição	Tipo	Domínio	Restrição
Numero	Número do pedido	VARCHAR(50)	VARCHAR(50)	- Chave primária
Data_pedido	Data do pedido	VARCHAR(100)	VARCHAR(100)	- Not Null
Data_prevista_entrega	Data prevista de entrega do pedido	DATE	DATE	-Not Null
Data_coleta	Data da coleta do pedido	DATE	DATE	- Not Null
Peso	Peso do pedido	VARCHAR(100)	VARCHAR(100)	-Not Null
Volume	Volume do pedido	VARCHAR(100)	VARCHAR(100)	- Not Null
Destino	Destino do pedido	VARCHAR(100)	VARCHAR(100)	- Not Null
Numero_nota_fiscal	Número da nota fiscal do pedido	VARCHAR(50)	VARCHAR(50)	- Not Null
CNPJ_empresa_participante	CNPJ da empresa participante do pedido	CHAR(18)	CHAR(18)	- Chave estrangeira - Not Null
CNPJ_transportadora	CNPJ transportadora do pedido	CHAR(18)	CHAR(18)	Chave estrangeira -Not Null

Item

Atributo	Descrição	Tipo	Domínio	Restrição
Código	Código do item	INTEGER	Números inteiros positivos	- Chave primária

Item				
Nome	Nome do item	VARCHAR(100)	VARCHAR(100)	- Not Null
Tipo	Tipo do item	VARCHAR(10)	VARCHAR(10)	- Not Null
Categoria	Categoria do item	VARCHAR(50)	VARCHAR(50)	- Not Null
Quantidade_e estoque	Quantidade do estoque do item	INTEGER	Números inteiros positivos	- Not Null
CNPJ_empresa_matriz	CNPJ da empresa matriz	CHAR(18)	CHAR(18)	- Chave estrangeira

Pedido_possui_item				
Atributo	Descrição	Tipo	Domínio	Restrição
Numero_pedido	Número do pedido	CHAR(18)	CHAR(18)	- Chave primária - Chave estrangeira
Codigo_item	Código do item	INTEGER	Números inteiros positivos	- Chave primária - Chave estrangeira - Not Null
Preco_unitario	Preço unitário do item no pedido	REAL	Números reais positivos	- Not Null
Quantidade	Quantidade de cada item no pedido	INTEGER	Números inteiros positivos	- Not Null

4. Modelo físico

4.1. Scripts de criação das tabelas

- **Tabela Empresa_matriz**

```
CREATE TABLE Empresa_matriz(
    CNPJ CHAR(18),
    Nome VARCHAR(100) NOT NULL,
    CONSTRAINT PK_Empresa_matriz PRIMARY KEY(CNPJ)
);
```

- **Tabela Empresa_participante**

```
CREATE TABLE Empresa_participante(
    CNPJ CHAR(18),
    Nome VARCHAR(100) NOT NULL,
    Tipo VARCHAR(20) NOT NULL,
    Telefone1 VARCHAR(20) NOT NULL,
    Telefone2 VARCHAR(20),
    Endereco VARCHAR(100) NOT NULL,
    CONSTRAINT PK_Empresa_participante PRIMARY KEY(CNPJ)
);
```

- **Tabela Funcionario**

```
CREATE TABLE Funcionario(
    CPF CHAR(14),
    Nome VARCHAR(100) NOT NULL,
    Funcao VARCHAR(50) NOT NULL,
    Numero_conta VARCHAR(20) NOT NULL,
    Nome_banco VARCHAR(20) NOT NULL,
    Salario REAL NOT NULL,
    Data_pagamento DATE NOT NULL,
    CNPJ_empresa_matriz CHAR(18) NOT NULL,
    CONSTRAINT PK_Funcionario PRIMARY KEY(CPF),
    CONSTRAINT FK_Funcionario FOREIGN KEY (CNPJ_empresa_matriz)
REFERENCES Empresa_matriz(CNPJ)
ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT CH_Funcionario CHECK(Salario>0)
);
```

- **Tabela Nota_fiscal**

```
CREATE TABLE Nota_fiscal(
    Numero VARCHAR(50),
    Valor REAL NOT NULL,
    Data_emisao DATE NOT NULL,
    CONSTRAINT PK_Nota_fiscal PRIMARY KEY(Numero)
);
```

- **Tabela Pagamento**

```

CREATE TABLE Pagamento(
    ID VARCHAR(20),
    Tipo VARCHAR(50) NOT NULL,
    Data_pagamento DATE,
    Valor_entrada REAL,
    CNPJ_empresa_matriz CHAR(18) NOT NULL,
    CNPJ_empresa_participante CHAR(18) NOT NULL,
    Numero_nota_fiscal VARCHAR(50) NOT NULL,
    CONSTRAINT PK_Pagamento PRIMARY KEY(ID),
    CONSTRAINT FK_Pagamento_empresa_matriz FOREIGN
KEY(CNPJ_empresa_matriz) REFERENCES Empresa_matriz(CNPJ)
ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT FK_Pagamento_empresa_participante FOREIGN
KEY(CNPJ_empresa_participante) REFERENCES Empresa_participante(CNPJ)
ON UPDATE CASCADE ON DELETE RESTRICT
);

```

- **Tabela Parcela**

```

CREATE TABLE Parcela(
    ID VARCHAR(20),
    Valor REAL NOT NULL,
    Data_vencimento DATE NOT NULL,
    Numero_boleto VARCHAR(25),
    ID_pagamento VARCHAR(20) NOT NULL,
    CONSTRAINT PK_Parcels PRIMARY KEY(ID, Numero_boleto),
    CONSTRAINT FK_Parcels FOREIGN KEY(ID_pagamento) REFERENCES
Pagamento(ID) ON UPDATE CASCADE ON DELETE RESTRICT
);

```

- **Tabela Transportadora**

```

CREATE TABLE Transportadora(
    CNPJ CHAR(18),
    Nome VARCHAR(100) NOT NULL,
    Inscricao_estadual CHAR(12) UNIQUE NOT NULL,
    Telefone1 VARCHAR(20) NOT NULL,
    Telefone2 VARCHAR(20),
    Endereco VARCHAR(100) NOT NULL,
    CONSTRAINT PK_Transportadora PRIMARY KEY(CNPJ)
);

```

- **Tabela Pedido**

```

CREATE TABLE Pedido(
    Numero VARCHAR(50),
    Data_pedido DATE NOT NULL,
    Data_prevista_entrega DATE NOT NULL,

```

```

Data_coleta DATE NOT NULL,
Peso VARCHAR(100) NOT NULL,
Volume VARCHAR(100) NOT NULL,
Destino VARCHAR(100) NOT NULL,
Numero_nota_fiscal VARCHAR(50) NOT NULL,
CNPJ_empresa_participante CHAR(18) NOT NULL,
CNPJ_transportadora CHAR(18) NOT NULL,
CONSTRAINT PK_Pedido PRIMARY KEY(Numero),
    CONSTRAINT FK_Pedido FOREIGN KEY(CNPJ_empresa_participante)
REFERENCES Empresa_participante(CNPJ)
    ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT FK_Pedido_transportadora FOREIGN KEY(CNPJ_transportadora)
REFERENCES Transportadora(CNPJ)
    ON UPDATE CASCADE ON DELETE RESTRICT
);

```

- **Tabela Item**

```

CREATE TABLE Item(
    Codigo INTEGER,
    Nome VARCHAR(100) NOT NULL,
    Tipo VARCHAR(10) NOT NULL,
    Categoria VARCHAR(50) NOT NULL,
    Quantidade_estoque INTEGER NOT NULL,
    CNPJ_empresa_matriz CHAR(18) NOT NULL,
    CONSTRAINT PK_Item PRIMARY KEY(Codigo),
    CONSTRAINT FK_Item FOREIGN KEY(CNPJ_empresa_matriz) REFERENCES
Empresa_matriz(CNPJ)
);

```

- **Tabela Pedido_possui_item**

```

CREATE TABLE Pedido_possui_item(
    Numero_pedido VARCHAR(50),
    Codigo_item INTEGER NOT NULL,
    Preco_unitario REAL NOT NULL,
    Quantidade INTEGER NOT NULL,
    CONSTRAINT PK_Item_pedido PRIMARY KEY(Numero_pedido, Codigo_item),
    CONSTRAINT FK_Item_pedido FOREIGN KEY(Numero_pedido) REFERENCES
Pedido(Numero)
    ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT FK_Item_codigo FOREIGN KEY(Codigo_item) REFERENCES
Item(Codigo)
    ON UPDATE CASCADE ON DELETE RESTRICT
)

```


4.2. Scripts de criação de index das tabelas

```
CREATE INDEX idx_empresa_matriz_cnpj ON empresa_matriz (cnpj);
CREATE INDEX idx_empresa_participante_cnpj ON empresa_participante (cnpj);
CREATE INDEX idx_funcionario_cpf ON funcionario (cpf);
CREATE INDEX idx_item_codigo ON item (codigo);
CREATE INDEX idx_nota_fiscal_numero ON nota_fiscal (numero);
CREATE INDEX idx_pagamento_id ON pagamento (id);
CREATE INDEX idx_parcela_id ON parcela (id);
CREATE INDEX idx_pedido_numero ON pedido (numero);
CREATE INDEX idx_pedido_possui_item_numero_pedido ON pedido_possui_item
(numero_pedido);
CREATE INDEX idx_transportadora_cnpj ON transportadora (cnpj);
```

4.3. Scripts de criação das visões

- **Visão que auxilia na exibição dos valores de pedidos, juntamente com a nota fiscal e o CNPJ e nome da transportadora referente.**

```
CREATE VIEW visao_pedido_valor_transportadora AS
SELECT P.numero AS numero_pedido, P.data_pedido, P.data_prevista_entrega,
NF.numero AS Numero_nf, NF.valor, P.cnpj_transportadora, T.nome AS
Nome_transportadora
FROM pedido P INNER JOIN nota_fiscal NF ON P.numero_nota_fiscal = NF.numero
INNER JOIN transportadora T ON P.cnpj_transportadora = T.cnpj
```

- **Visão que auxilia na exibição de itens presentes nos pedidos.**

```
CREATE VIEW visao_produtos_pedidos AS
SELECT PPI.numero_pedido, PPI.codigo_item, PPI.preco_unitario, PPI.quantidade,
I.nome AS nome_produto, I.categoria
FROM pedido_possui_item PPI JOIN item I ON PPI.codigo_item = I.codigo
```

- **Visão que auxilia na exibição das informações dos funcionários ocultando dados sensíveis e adicionando informações da empresa matriz.**

```
CREATE VIEW visao_funcionario_empresa_matriz AS
SELECT F.nome AS Nome_funcionario, F.funcao, F.salario, F.data_pagamento,
EM.nome AS Empresa_filiada, EM.cnpj AS CNPJ_empresa_filiada
FROM funcionario F JOIN empresa_matriz em ON F.cnpj_empresa_matriz = EM.cnpj
```

4.4. Scripts de criação dos procedimentos armazenados

- **Criar um procedimento armazenado para calcular o código do próximo item a ser adicionado.**

```
CREATE OR REPLACE FUNCTION ProximoItem()
RETURNS INTEGER AS $$
```

```

DECLARE
    Maior INTEGER;
BEGIN
    SELECT INTO Maior MAX(codigo)
    FROM Item;
    IF Maior IS NULL THEN
        RETURN 1;
    ELSE
        RETURN Maior + 1;
    END IF;
END
$$ LANGUAGE PLPGSQL;

```

- **Criar um procedimento armazenado que retorna a quantidade de pedidos realizados por uma empresa específica.**

```

CREATE OR REPLACE FUNCTION QuantidadeDePedidos(Cnpj CHAR) RETURNS
INTEGER
AS $$
DECLARE
    Total INTEGER;
BEGIN
    SELECT INTO Total COUNT(*)
    FROM Pedido
    WHERE Cnpj_empresa_participante = Cnpj;
    RETURN Total;
END
$$ LANGUAGE PLPGSQL

```

- **Criar um procedimento armazenado que reajuste os salários de todos os funcionários com base em um percentual de aumento informado.**

```

CREATE OR REPLACE PROCEDURE AumentoSalarial(Percentual REAL)
AS $$
BEGIN
    UPDATE Funcionario
    SET Salario = Salario + (Salario * Percentual / 100);
END
$$ LANGUAGE PLPGSQL

```

4.5 Scripts criação dos gatilhos

- **Gatilho para alterar o valor final da nota fiscal sempre que houver uma inserção ou atualização na tabela Pedido_possui_item.**

```

CREATE FUNCTION AlteraPrecoNotaFiscal()
RETURNS TRIGGER AS $$

```

```

BEGIN
  UPDATE nota_fiscal nf
  SET valor = (
    SELECT SUM(ppi.preco_unitario * ppi.quantidade)
    FROM pedido p
    INNER JOIN pedido_possui_item ppi ON p.numero = ppi.numero_pedido
    WHERE p.numero_nota_fiscal = nf.numero
  )
  WHERE EXISTS (
    SELECT *
    FROM pedido p
    INNER JOIN pedido_possui_item ppi ON p.numero = ppi.numero_pedido
    WHERE p.numero_nota_fiscal = nf.numero);

  RETURN NEW;
END
$$ LANGUAGE PLPGSQL;

```

```

CREATE TRIGGER AlteraPrecoNF
AFTER INSERT OR UPDATE OR DELETE ON pedido_possui_item
FOR EACH ROW
EXECUTE FUNCTION AlteraPrecoNotaFiscal();

```

- **Gatilho para atualizar a quantidade em estoque na tabela Item com base na quantidade selecionada na coluna quantidade da tabela Pedido_possui_item.**

```

CREATE OR REPLACE FUNCTION AtualizarEstoque()
RETURNS TRIGGER AS $$
BEGIN
  UPDATE Item
  SET Quantidade_estoque = Quantidade_estoque - NEW.Quantidade
  WHERE Codigo = NEW.Codigo_item;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER AtualizaEstoqueTrigger
AFTER INSERT OR UPDATE OR DELETE ON pedido_possui_item
FOR EACH ROW
EXECUTE FUNCTION AtualizarEstoque();

```

- **Gatilho para verificar a quantidade itens no estoque na tabela Item: caso o estoque esteja vazio ou a quantidade na coluna quantidade da tabela Pedido_possui_item seja maior que a disponível no estoque, o pedido não poderá ser realizado.**

```

CREATE OR REPLACE FUNCTION verificar_estoque()
RETURNS TRIGGER AS $$
DECLARE
    estoque_atual INT;
BEGIN
    SELECT Quantidade_estoque INTO estoque_atual
    FROM Item
    WHERE Codigo = NEW.Codigo_item;
    IF estoque_atual < NEW.Quantidade THEN
        RETURN NULL;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER verificar_estoque_trigger
BEFORE INSERT OR UPDATE ON Pedido_possui_item
FOR EACH ROW
EXECUTE FUNCTION verificar_estoque();

```

4.6. Scripts de Povoamento das Relações

- **Tabela Empresa_matriz**

```

INSERT INTO Empresa_matriz VALUES ('23.374.558/0001-32', 'M&S Roletes');
INSERT INTO Empresa_matriz VALUES ('15.474.344/0001-00', 'Quixabeira Rolos');

```

- **Tabela Empresa_participante**

```

INSERT INTO Empresa_participante VALUES ('24.316.385/0001-69', 'Industria
Palmovani', 'Cliente', '(15)2075-4782', NULL, 'Avenida Prestes Maia, 150, Centro, São
Paulo - SP');
INSERT INTO Empresa_participante VALUES ('43.913.879/0001-75', 'Aço Cabral',
'Fornecedora', '(18)3642-6247', '(19)2443-1883', 'Rua Lázaro Félix da Silva, 200, Vila
Três Marias, São Paulo - SP');
INSERT INTO Empresa_participante VALUES ('45.678.912/0001-56', 'MetalTech
Indústria', 'Fornecedora', '(11) 94567-8912', '(11) 93456-7891', 'Av. das Indústrias, 500,
São Paulo - SP');
INSERT INTO Empresa_participante VALUES ('78.901.234/0001-67', 'Mecânica de
Precisão LTDA', 'Fornecedora', '(31) 97890-1234', NULL, 'Rua dos Engenheiros, 250,
Belo Horizonte - MG');
INSERT INTO Empresa_participante VALUES ('23.456.789/0001-89', 'AutoParts
Solutions', 'Cliente', '(21) 92345-6789', '(21) 93456-7890', 'Rodovia BR-040, Km 30,
Bairro Automotivo, Rio de Janeiro - RJ');

```

- **Tabela Funcionario**

```
INSERT INTO Funcionario VALUES ('670.672.730-05', 'Jorge Da silva', 'Repositor',
'79270-1', 'Itaú', 1510.00, '2025-03-05', '15.474.344/0001-00');
INSERT INTO Funcionario VALUES ('161.296.700-01', 'Ana Beatriz Queiroz',
'Programadora', '12932-5', 'Banco do Brasil', 13800.00, '2025-03-05',
'23.374.558/0001-32');
INSERT INTO Funcionario VALUES ('123.456.789-00', 'Carlos Eduardo Silva',
'Engenheiro de Automação', '12345-6', 'Banco do Brasil', 7500.00, '2025-03-05',
'15.474.344/0001-00');
INSERT INTO Funcionario VALUES ('456.789.123-22', 'Rafael Mendes Oliveira',
'Técnico em Mecatrônica', '98765-4', 'Bradesco', 4000.00, '2025-03-05',
'15.474.344/0001-00');
INSERT INTO Funcionario VALUES ('987.654.321-11', 'Mariana Costa Souza',
'Operadora de Máquinas de Produção', '65432-1', 'Santander', 2500.00, '2025-03-05',
'23.374.558/0001-32');
```

- **Tabela Transportadora**

```
INSERT INTO Transportadora VALUES ('66.649.429/0001-12', 'Transporcar',
'277674522621', '(12) 3068-0413', '(14) 2165-6319', 'Rua Lacedemônia, Jardim Brasil
(Zona Sul, SP);
INSERT INTO Transportadora VALUES('08.7582.58/0001-63', 'LevaLeva',
'579802195815', '(12) 3303-8497', '(11) 2813-4344', 'Praça Dante Bin Filho, Campos
Elíseos, SP');
INSERT INTO Transportadora VALUES ('12.345.678/0001-90', 'Transportadora Viação
Norte', '111223344556', '(61) 4002-6000', '(61) 9888-2222', 'Rua do Contorno, Setor
Comercial, Brasília - DF');
INSERT INTO Transportadora VALUES ('98.765.432/0001-01', 'Logistica Transportes',
'223344556677', '(31) 3333-7777', NULL, 'Avenida do Sol, Bairro São José, Belo
Horizonte - MG');
INSERT INTO Transportadora VALUES ('56.789.123/0001-02', 'Transporte Rapidex',
'334455667788', '(21) 5555-3333', NULL, 'Rua dos Navegantes, Barra da Tijuca, Rio de
Janeiro - RJ');
```

- **Tabela Pedido**

```
INSERT INTO Pedido VALUES ('PED001', '2025-02-20', '2025-03-18', '2025-02-27',
'400 kg', '5 m³', 'Avenida Prestes Maia, Centro, SP', '00001', '24.316.385/0001-69',
'08.7582.58/0001-63');
INSERT INTO Pedido VALUES ('PED002', '2025-02-22', '2025-03-27', '2025-03-10',
'587 kg', '3 m³', 'Avenida Prestes Maia, Centro, SP', '00002', '24.316.385/0001-69',
'08.7582.58/0001-63');
```

```
INSERT INTO Pedido VALUES ('PED003', '2025-02-22', '2025-03-10', '2025-02-28',
'291 kg', '2.3 m³', 'Avenida Prestes Maia, Centro, SP', '00003', '24.316.385/0001-69',
'66.649.429/0001-12');
```

```
INSERT INTO Pedido VALUES ('PED004', '2025-02-22', '2025-03-27', '2025-03-10',
'587 kg', '3 m³', 'Avenida Brasil, Bairro Jardim, São Paulo - SP', '00004',
'23.456.789/0001-89', '12.345.678/0001-90');
```

```
INSERT INTO Pedido VALUES ('PED005', '2025-02-22', '2025-03-10', '2025-02-28',
'291 kg', '2.3 m³', 'Rua das Flores, Bairro Primavera, Belo Horizonte - MG', '00005',
'23.456.789/0001-89', '98.765.432/0001-01');
```

- **Tabela Item**

```
INSERT INTO Item VALUES (1001, 'Exaustor axial', 'Produto', 'Purificação', 5,
'23.374.558/0001-32');
```

```
INSERT INTO Item VALUES (1002, 'Fogão industrial', 'Produto', 'Alimentação', 18,
'15.474.344/0001-00');
```

```
INSERT INTO Item VALUES ('1003', 'Placa de aço', 'Produto', 'Revestimento', 158,
'15.474.344/0001-00');
```

```
INSERT INTO Item VALUES (1004, 'Rolete de impacto', 'Produto', 'Correia
transportadora', 29, '15.474.344/0001-00');
```

```
INSERT INTO Item VALUES (1005, 'Rolete de aço', 'Produto', 'Esteira transportadora',
44, '23.374.558/0001-32');
```

```
INSERT INTO Item VALUES (1006, 'Rolete livre', 'Produto', 'Esteira transportadora',
28, '23.374.558/0001-32');
```

- **Tabela Nota_fiscal**

```
INSERT INTO Nota_fiscal VALUES ('00002', 0, '2025-02-22');
```

```
INSERT INTO Nota_fiscal VALUES ('00003', 0, '2025-02-22');
```

```
INSERT INTO Nota_fiscal VALUES('00001', 0, '2025-02-08');
```

```
INSERT INTO Nota_fiscal VALUES('00005', 0, '2025-03-16');
```

```
INSERT INTO Nota_fiscal VALUES('00004', 0, '2025-05-01');
```

- **Tabela Pedido_possui_item**

```
INSERT INTO Pedido_possui_item VALUES ('PED001', 1003, 62.44, 25);
```

```
INSERT INTO Pedido_possui_item VALUES ('PED001', 1001, 489.22, 1);
```

```
INSERT INTO Pedido_possui_item VALUES ('PED001', 1005, 47.31, 6);
```

```
INSERT INTO Pedido_possui_item VALUES ('PED002', 1004, 79.15, 7);
```

```
INSERT INTO Pedido_possui_item VALUES ('PED003', 1003, 62.44, 10);
```

```
INSERT INTO Pedido_possui_item VALUES ('PED003', 1002, 254.49, 1);
```

```
INSERT INTO Pedido_possui_item VALUES ('PED004', 1001, 489.22, 3);
```

```
INSERT INTO Pedido_possui_item VALUES ('PED005', 1004, 79.15, 5);
```

- **Tabela Pagamento**

```

INSERT INTO Pagamento VALUES ('P001', 'Parcelado', '2025-02-18', NULL ,
'15.474.344/0001-00', '24.316.385/0001-69', '00001');
INSERT INTO Pagamento VALUES ('P002', 'À vista', '2025-02-05', '1178.45' ,
'15.474.344/0001-00', '24.316.385/0001-69', '00002');
INSERT INTO Pagamento VALUES ('P003', 'À vista', '2025-03-30', '254.49' ,
'15.474.344/0001-00', '24.316.385/0001-69', '00003');
INSERT INTO Pagamento VALUES ('P004', 'Parcelado', '2025-03-17', NULL ,
'23.374.558/0001-32', '23.456.789/0001-89', '00004');
INSERT INTO Pagamento VALUES ('P005', 'À vista', '2025-04-30', '395.75 ' ,
'23.374.558/0001-32', '23.456.789/0001-89', '00005');

```

- **Tabela Parcela**

```

INSERT INTO Parcela VALUES ('PARC001', 1167.04, '2025-02-18',
'5896925411946786453429545', 'P001');
INSERT INTO Parcela VALUES ('PARC002', 1167.04, '2025-03-18',
'2173417223451660615562067', 'P001');
INSERT INTO Parcela VALUES ('PARC003', 489.22, '2025-03-17',
'6135127230296412946175772', 'P004');
INSERT INTO Parcela VALUES ('PARC004', 489.22, '2025-05-17',
'5526720515547771075909299', 'P004');
INSERT INTO Parcela VALUES ('PARC005', 489.22, '2025-06-17',
'9157222183374078346319956', 'P004');

```

4.7. Scripts das Consultas

Recuperar o nome dos funcionários e o nome da empresa matriz onde estão alocados:

```

SELECT T.nome AS nome_funcionario, EM.nome AS empresa_matriz
FROM funcionario T INNER JOIN empresa_matriz EM ON
T.cnpj_empresa_matriz = EM.cnpj

```

Recuperar o número dos pedidos junto com os nomes e CNPJs da transportadora e da empresa participante:

```

SELECT P.numero AS numero_pedido, T.nome AS nome_transportadora, T.cnpj AS
cnpj_transportadora, EP.nome AS nome_empresa_participante, EP.cnpj AS
cnpj_empresa_participante
FROM (pedido P INNER JOIN transportadora T ON P.cnpj_transportadora = T.cnpj)
INNER JOIN empresa_participante EP ON P.cnpj_empresa_participante = EP.cnpj

```

Recuperar o nome das empresas participantes que não possuem um segundo telefone cadastrado:

```

SELECT nome
FROM empresa_participante
WHERE telefone2 IS NULL

```

Recuperar o nome das empresas participantes que possuem os dois telefone cadastrado:

```
SELECT nome
FROM empresa_participante
WHERE telefone1 IS NOT NULL AND telefone2 IS NOT NULL
```

Recuperar os dados dos funcionários cujo o nome comece com a letra “A”:

```
SELECT *
FROM funcionario
WHERE nome LIKE 'A%'
```

Recuperar o nome das empresas com “Aço” no nome:

```
SELECT *
FROM empresa_participante
WHERE nome LIKE '%Aço%'
```

Recuperar todos os pedidos, do mais recente para o mais antigo:

```
SELECT *
FROM Pedido
ORDER BY Data_pedido DESC
```

Recuperar o nome dos funcionários ordenados pelo salário (maiores salários primeiro):

```
SELECT Nome
FROM funcionario
ORDER BY salario DESC
```

Recuperar os itens com estoque abaixo da média:

```
SELECT *
FROM Item
WHERE quantidade_estoque < ALL (SELECT AVG(quantidade_estoque) FROM Item)
```

Recuperar os pedidos com valor total acima da média:

```
SELECT *
FROM pedido
WHERE numero_nota_fiscal IN ((SELECT numero
FROM nota_fiscal
WHERE valor > (SELECT AVG(valor)
FROM nota_fiscal))))
```

Recuperar o nome das transportadoras que nunca realizaram pedidos:

```
SELECT T.nome
FROM transportadora T
```



```

WHERE NOT EXISTS
(SELECT *
FROM pedido P
WHERE T.cnpj = P.cnpj_transportadora)

```

Recuperar os itens com a quantidade em estoque acima da média de sua categoria:

```

SELECT *
FROM item I
WHERE EXISTS (
SELECT I2.categoria
FROM item I2
WHERE I2.categoria = I.categoria
GROUP BY I2.categoria
HAVING AVG(I2.quantidade_estoque) < I.quantidade_estoque
)

```

Recuperar todos os CNPJs das empresas matriz e participante:

```

SELECT cnpj
FROM empresa_matriz
UNION
SELECT cnpj
FROM empresa_participante

```

Recuperar os funcionários com salário acima da média e abaixo da média:

```

SELECT CPF, nome, salario, 'Acima da Média' AS Status_Salario
FROM funcionario
WHERE salario > (SELECT AVG(salario) FROM Funcionario)
UNION
SELECT CPF, nome, salario, 'Abaixo da Média' AS Status_Salario
FROM funcionario
WHERE salario < (SELECT AVG(salario) FROM Funcionario)
ORDER by salario DESC;

```

Recuperar o total de salários pagos aos funcionários de cada empresa matriz:

```

SELECT EM.cnpj, EM.nome, SUM(F.salario) AS salarios_total
FROM funcionario F INNER JOIN empresa_matriz EM ON F.cnpj_empresa_matriz =
EM.cnpj
GROUP BY EM.cnpj, EM.nome

```

Recuperar a média do valor das notas fiscais:

```

SELECT AVG(valor) AS media_valor_nota_fiscal
FROM nota_fiscal

```

Recuperar o número total de pedidos enviados por cada transportadora:

```
SELECT T.cnpj AS cnpj_transportadora , T.nome, COUNT(P.numero) AS total_pedidos
FROM pedido p RIGHT OUTER JOIN transportadora T ON P.cnpj_transportadora =
T.cnpj
GROUP BY T.cnpj, T.nome
ORDER BY COUNT(P.numero) DESC
```

Recuperar o total de itens vendidos por categoria exibindo apenas categorias com mais de 30 vendas:

```
SELECT I.categoria, SUM(pItem.quantidade) AS total_vendido
FROM Pedido_possui_item pItem
INNER JOIN Item I ON I.codigo = pItem.codigo_item
GROUP BY I.categoria
HAVING SUM(pItem.quantidade) > 30
```