

INSTITUTO FEDERAL DO AMAZONAS



Alunos: Jamily de Melo Pinheiro, Mariane da Silva Bindá, Adrielly Souza,
Ágata Júlia, Wylham Vitor

BANCO DE DADOS

**MANAUS
2023**

Alunos: Jamily de Melo Pinheiro, Mariane da Silva Bindá, Adrielly Souza, Ágata Júlia, Wylham Vitor

BANCO DE DADOS

Trabalho solicitado pelo professor Juan Marcelo Dell Oso, da disciplina de Banco de Dados, para obtenção de nota parcial da quarta etapa.

MANAUS

2023

• DESCRIÇÃO

O 'Inventory Management' é um sistema de gerenciamento de estoque em desenvolvimento, projetado para administrar o abastecimento de roupas em lojas. Este sistema permite um controle minucioso das entradas e saídas de mercadorias, possibilitando aos usuários monitorar os níveis de estoque, estabelecer alertas de reabastecimento e analisar as tendências de vendas para otimizar pedidos. Além disso, facilita a inclusão de informações abrangentes sobre produtos, como fotos, descrições, preços e disponibilidade de tamanhos e cores, mantendo um registro preciso do estoque. Também oferece a capacidade de criar perfis de clientes, registrar informações de contato e histórico de compras, contribuindo para um atendimento personalizado e promoções direcionadas.

O sistema disponibiliza ferramentas para gerar relatórios detalhados sobre vendas, estoque, desempenho de produtos e tendências de mercado, fornecendo aos gerentes informações embasadas nas estratégias de negócios para tomar decisões informadas. Em um mercado competitivo de moda, esse sistema é uma ferramenta valiosa que aprimora a eficiência operacional, a experiência do cliente e impulsiona os lucros, ao fornecer controle completo sobre todos os aspectos do negócio e informações críticas para tomadas de decisões bem fundamentadas.

Esse sistema terá três perfis para usuários : administrador, fornecedor e funcionário para o gerenciamento de roupas que dá à loja o controle detalhado do estoque de roupas, acompanhando a entrada e saída de mercadorias. Os usuários podem monitorar níveis de estoque, definir alertas de reabastecimento e acompanhar as tendências de vendas para fazer pedidos com eficiência.

Cadastro de produto: Facilita a inclusão de informações sobre os produtos, como fotos, descrições, preços, tamanhos e cores disponíveis. Isso ajuda a manter um registro preciso de todos os itens em estoque.

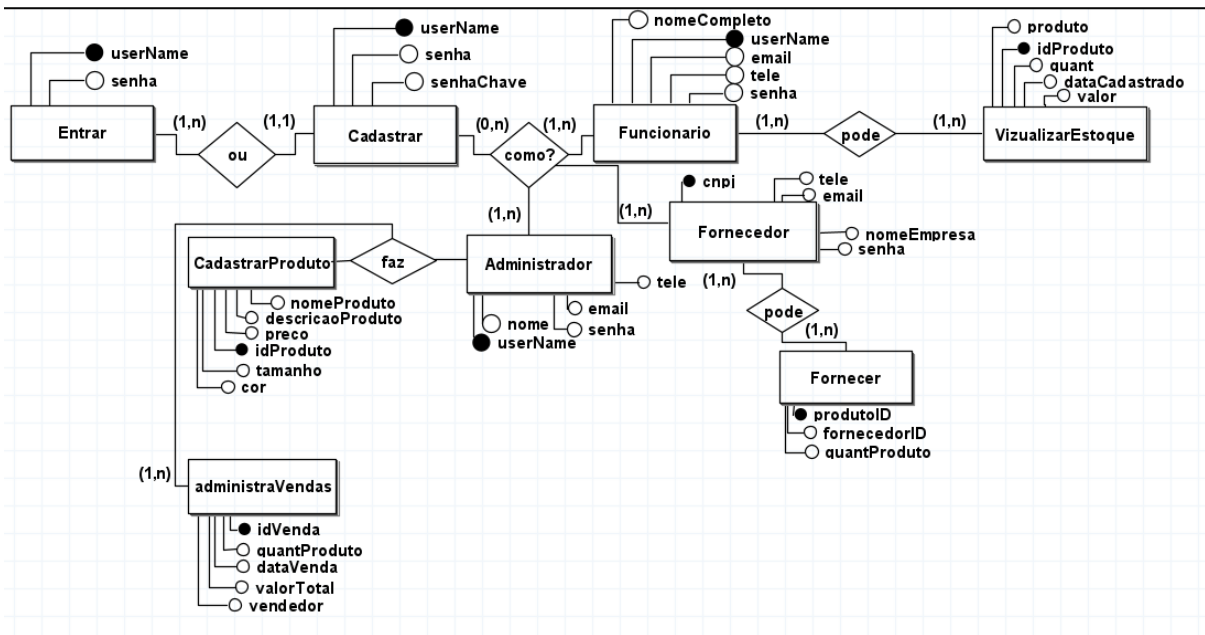
Gestão de Funcionários : Permite a criação de perfis de funcionários , registro de informações de contato e histórico de compras. Isso ajuda a personalizar o atendimento ao cliente e oferecer promoções direcionadas.

Relatórios e análises: Oferece ferramentas para gerar relatórios sobre vendas, estoque, desempenho de produtos e tendências de mercado. Essas informações ajudam os gerentes a tomar decisões informadas sobre estratégias de negócios.

Gestão de fornecedores: Facilita o acompanhamento de fornecedores, pedidos de reposição e histórico de compras. Isso ajuda a manter um relacionamento eficiente com os fornecedores e a garantir o estoque adequado de produtos.

Controle de caixa e finanças acompanha todas as transações financeiras, incluindo entradas e saídas de dinheiro, e auxilia na reconciliação de caixa.

- **MER**



● MAPEAMENTO

Entrar (userName, senha, confirmar senha, tipo usuario)

Cadastrar (userName, senha, senhaChave)

Fornecedor (cnpi, nomeEmpresa, tele, email, nomeEmpresa, senha)

Funcionário (nomeCompleto, userName, tele, email, senha)

Administrador (nomeCompleto, userName, tele, email, senha)

VisualizarEstoque (idProduto, quant, dataCadastro, valor)

Pagamento (id, metodo pagamento, tipo pagamento)

CadastrarProduto (idProduto, nomeProduto, descricaoProduto, preco, tamanho, cor)

Produtos (id, descricao, preco, quant estoque, nome fornecedor)

Estoque (id, nome produto, quant, preco unitario, data cadastro)

ProdutoFornecedor (id, *produto id*, *fornecedor id*)

EntradaMercadoria (id, **produto id**, quant, data entrada)

SaidaMercadoria (id, **produto** id, quant, data saida)

ControleCustos (id, descricao, valor, data registro)

GerenciamentoLotes (id, produto, lote, quant, data validade)

ControlePrecos (id, produto, preco, data_modificacao)

RelatorioVendaEstoque (id, data relatorio, total vendas, total estoque)

EstoqueTamanhoCor (id, produto, tamanho, cor, quant)

Vendas (id, cliente, produto, quant, valor total, data venda)

- **DICIONÁRIO:**

Entrar

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
userName🔑	VARCHAR (200)	Não	Nome de usuário da pessoa para o acesso ao sistema
senha	VARCHAR (30)	Não	Senha de acesso ao sistema
confimar_senha	VARCHAR (30)	Não	Confirmação da senha para o acesso ao sistema
tipo_usuario	VARCHAR (30)	Não	A qual grupo o usuário pertence

Cadastrar


ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id🔑	INT	Não	Identificação do usuário
email	VARCHAR (255)	Não	Email para a mudança de senha
nova_senha	VARCHAR (30)	Não	Nova senha de acesso ao sistema
confimar_nova_senha	VARCHAR (30)	Não	Confirmação da nova senha de acesso
tipo_usuario	VARCHAR (30)	Não	A qual grupo o usuário pertence

Fornecedor

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
nome_empresa🔑	VARCHAR (50)	Não	Nome da empresa do fornecedor
cnjp	VARCHAR (15)	Não	
tele_comercial	VARCHAR (12)	Não	Telefone para contato, comercial
email_comercial	VARCHAR(30)	Não	Email para contato,

			comercial
nome_repre	VARCHAR (200)	Não	Nome do representante da empresa
email_repre	VARCHAR(50)	Não	Email para contato do representante
tele_repre	VARCHAR(12)	Não	Telefone para contato do representante
senha	VARCHAR(30)	Não	Senha do fornecedor para acesso ao sistema
confirmar_senha	VARCHAR(30)	Não	Confirmação da senha para acesso

Funcionario

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id_funcionario 	INT	Não	Identificação do funcionário
nome_completo	VARCHAR(200)	Não	Nome completo do funcionário
userName	VARCHAR(100)	Não	Nome de usuário do funcionário para acesso ao sistema
data_nasc	DATE	Não	Data de nascimento do funcionário
tele	VARCHAR(12)	Não	Telefone para contato do funcionário
email	VARCHAR(50)	Não	Email para contato do funcionário
senha	VARCHAR(30)	Não	Senha para acessar o sistema
confirmar_senha	VARCHAR(30)	Não	Confirmação da senha para acesso ao sistema

Administrador

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
----------	------	------	-----------

id_adm🔑	INT	Não	Identificação do Administrador
nome_completo	VARCHAR(200)	Não	Nome completo do administrador
userName	VARCHAR(100)	Não	Nome de usuário do administrador para acessar o sistema
data_nasc	DATE	Não	Data de nascimento do administrador
tele	VARCHAR(12)	Não	Telefone para contato do Administrador
senha	VARCHAR(30)	Não	Senha para acessar o sistema
confirmar_senha	VARCHAR(30)	Não	Confirmação de senha para acesso ao sistema
email	VARCHAR(50)	Não	Email para contato

MenuFuncionario

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id🔑	INT	Não	Identificação para acessar o menu
opcao_menu	VARCHAR(50)	Não	

VisualizacaoEstoque


ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id🔑	INT	Não	Identificação para visualizar o estoque
item_estoque	VARCHAR(100)	Não	Itens que tem no estoque
esta_no_estoque	BOOLEAN	Não	Itens que estão disponíveis no estoque

Pagamento

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id🔑	INT	Não	Id do pagamento
metodo_pagamento	VARCHAR(50)	Não	Maneira que vai ser o pagamento

tipo_pagamento	VARCHAR(50)	Não	Como vai ser pago
----------------	-------------	-----	-------------------

CadastrarProduto

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id 	INT	Não	Id do produto
nome_produto	VARCHAR(100)	Não	Nome do produto
descricao_produto	TEXT	Não	Descrição do produto
preco	DECIMAL(10,2)	Não	Preço do produto


Produtos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id 	INT	Não	Id do produto
descricao	VARCHAR(255)	Não	Descrição do produto
preco	DECIMAL(10,2)	Não	Preço do produto
quant_estoque	INT	Não	Quantidade de produto que tem no estoque
nome_fornecedor	VARCHAR(200)	Não	Nome do fornecedor do produto

Estoque

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id 	INT	Não	Identificação
nome_produto	VARCHAR(260)	Não	Nome do produto
quant	INT	Não	Quantidade de produto
preco_unitario	DECIMAL(10,2)	Não	Preço unitário do produto
data_cadastro	DATE	Não	Data que foi cadastrado o produto

ProdutoFornecedor

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id 	INT	Não	Identificação

produto_id 🔑	INT	Não	Id do Produto
fornecedor_id 🔑	INT	Não	Id do fornecedor

EntradaMercadoria

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id 🔑	INT	Não	Identificação
produto_id 🔑	INT	Não	Id do produto
quant	INT	Não	Quantidade de produto
data_entrada	DATE	Não	Data que o produto saiu

SaidaMercadoria


ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id 🔑	INT	Não	Identificação
produto_id 🔑	INT	Não	Id do produto
quant	INT	Não	Quantidade de produto
data_saida	DATE	Não	Data em que o produto saiu

ControleCustos


ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id 🔑	INT	Não	Identificação
descricao	VARCHAR(255)	Não	Descrição
valor	DECIMAL(10,2)	Não	Valores
data_registro	DATE	Não	Data do registro

GerenciamentoLotes

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
----------	------	------	-----------

id 	INT	Não	Identificação
produto	VARCHAR(250)	Não	Produto
lote	VARCHAR(50)	Não	Lote do produto
quant	INT	Não	Quantidade de produto
data_validade	DATE	Não	Data de validade

ControlePrecos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id 	INT	Não	Identificação
produto	VARCHAR(250)	Não	Produto
preco	DECIMAL(10,2)	Não	Preço do produto
data_modificacao	DATE	Não	Data da modificação


RelatorioVendaEstoque

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id 	INT	Não	Id do relatório
data_relatorio	DATE	Não	Data que o relatório está sendo feito
total_vendas	DECIMAL(10,2)	Não	Total de vendas que teve
total_estoque	INT	Não	Total que restou no estoque

EstoqueTamanhoCor

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id 	INT	Não	Identificação
produto	VARCHAR(255)	Não	Produto
tamanho	VARCHAR(20)	Não	Tamanho do produto
cor	VARCHAR(20)	Não	Cor do produto
quant	INT	Não	Quantidade de produto

Vendas

ATRIBUTO	TIPO	NULO	DESCRIÇÃO
id 	INT	Não	Identificação
cliente	VARCHAR(255)	Não	Cliente que fez a venda
produto	VARCHAR(255)	Não	Produto
quant	INT	Não	Quantidade de produto
valor_total	DECIMAL(10,2)	Não	Valor total
data_venda	DATE	Não	Data da venda

- **SQL**

```
CREATE DATABASE Estoque;
```

```
USE Estoque;
```

```
CREATE TABLE Entrar (
```

```
  userName VARCHAR (200) UNIQUE NOT NULL PRIMARY KEY,  
  senha VARCHAR (30) NOT NULL,  
  confirmar_senha VARCHAR (30) NOT NULL,  
  tipo_usuario ENUM('Fornecedor', 'Funcionário', 'Administrador') NOT NULL  
);
```

```
CREATE TABLE Esqueceu_sua_senha (
```

```
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nome_completo VARCHAR(255) NOT NULL,  
  email VARCHAR(255) UNIQUE NOT NULL,  
  nova_senha VARCHAR(30) NOT NULL,  
  confirmar_nova_senha VARCHAR(30) NOT NULL,  
  tipo_usuario ENUM ('Fornecedor', 'Funcionário', 'Administrador') NOT NULL
```

);

CREATE TABLE Fornecedor (

nome_empresa VARCHAR (50) NOT NULL,
cnpj VARCHAR (15) NOT NULL,
tele_comercial VARCHAR (12) NOT NULL,
email_comercial VARCHAR (30) UNIQUE NOT NULL,
nome_repre VARCHAR (200) NOT NULL,
email_repre VARCHAR (50) UNIQUE NOT NULL,
tele_repre VARCHAR (12) NOT NULL,
senha VARCHAR (30) NOT NULL,
confirmar_senha VARCHAR (30) NOT NULL,
);

CREATE TABLE Funcionario (

id_funcionario INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
nome_completo VARCHAR (200) NOT NULL,
userName VARCHAR (100) UNIQUE NOT NULL,
data_nasc DATE NOT NULL,
tele VARCHAR (12) NOT NULL,
email VARCHAR (50) UNIQUE NOT NULL,
senha VARCHAR (30) NOT NULL,
confirmar_senha VARCHAR (30) NOT NULL,
);

CREATE TABLE Administrador (

id_adm INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
nome_completo VARCHAR(200) NOT NULL,
userName VARCHAR(100) UNIQUE NOT NULL,
data_nasc DATE NOT NULL,
tele VARCHAR(12) NOT NULL,
senha VARCHAR (30) NOT NULL,
confirmar_senha VARCHAR (30) NOT NULL,
email VARCHAR(50) UNIQUE NOT NULL,
);

CREATE TABLE MenuFuncionario (

id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
opcao_menu VARCHAR (50) NOT NULL
);

```
CREATE TABLE VisualizacaoEstoque (
```

```
id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
item_estoque VARCHAR (100) NOT NULL,  
esta_no_estoque BOOLEAN NOT NULL  
);
```

```
CREATE TABLE Pagamento (
```

```
id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
metodo_pagamento VARCHAR (50) NOT NULL,  
tipo_pagamento VARCHAR (50) NOT NULL  
);
```

```
CREATE TABLE CadastrarProduto (
```

```
id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
nome_produto VARCHAR (100) NOT NULL,  
descricao_produto TEXT,  
preco DECIMAL (10, 2) NOT NULL );  
CREATE TABLE Produtos (
```

```
id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
descricao VARCHAR (255),  
preco DECIMAL (10, 2) NOT NULL,  
quant_estoque INT NOT NULL,  
nome_fornecedor VARCHAR (200) NOT NULL  
);
```

```
CREATE TABLE Estoque (
```

```
id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
nome_produto VARCHAR (260) NOT NULL,  
quant INT NOT NULL,  
preco_unitario DECIMAL (10, 2) NOT NULL,  
data_cadastro DATE NOT NULL  
);
```

```
CREATE TABLE ProdutoFornecedor (
```

```
id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
produto_id INT NOT NULL,  
fornecedor_id INT NOT NULL,
```

```
FOREIGN KEY (Produto_id) REFERENCES Estoque (id),  
FOREIGN KEY (Fornecedor_id) REFERENCES Fornecedor (id)  
);
```

```
CREATE TABLE EntradaMercadoria (
```

```
id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
produto_id INT NOT NULL,  
quant INT NOT NULL,  
data_entrada DATE NOT NULL,  
FOREIGN KEY (produto_id) REFERENCES Estoque (id)  
);
```

```
CREATE TABLE SaidaMercadoria (
```

```
id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
produto_id INT NOT NULL,  
quant INT NOT NULL,  
data_saida DATE NOT NULL,  
FOREIGN KEY (produto_id) REFERENCES Estoque (id)  
);
```

```
CREATE TABLE ControleCustos (
```

```
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
descricao VARCHAR(255) NOT NULL,  
valor DECIMAL(10, 2) NOT NULL,  
data_registro DATE NOT NULL  
);
```

```
CREATE TABLE GerenciamentoLotes (
```

```
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
produto VARCHAR(250) NOT NULL,  
lote VARCHAR(50) NOT NULL,  
quant INT NOT NULL,  
data_validade DATE NOT NULL  
);
```

```
CREATE TABLE ControlePrecos (
```

```
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
produto VARCHAR(250) NOT NULL,  
preco DECIMAL(10, 2) NOT NULL,  
data_modificacao DATE NOT NULL
```

);

CREATE TABLE RelatorioVendaEstoque (

id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
data_relatorio DATE NOT NULL,
total_vendas DECIMAL(10, 2) NOT NULL,
total_estoque INT NOT NULL

);

CREATE TABLE EstoqueTamanhoCor (

id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
produto VARCHAR(255) NOT NULL,
tamanho VARCHAR(20) NOT NULL,
cor VARCHAR(20) NOT NULL,
quant INT NOT NULL

);

CREATE TABLE Vendas (

id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
cliente VARCHAR(255) NOT NULL,
produto VARCHAR(255) NOT NULL,
quant INT NOT NULL,
valor_total DECIMAL(10, 2) NOT NULL,
data_venda DATE NOT NULL

);

- **CONSULTAS**

Tabela Entrar :

```
SELECT * FROM Entrar;  
SELECT * FROM Entrar WHERE userName = 'nome_usuario';  
SELECT * FROM Entrar WHERE tipo_usuario = 'tipo_desejado';  
SELECT * FROM Entrar WHERE senha = 'senha_desejada' AND confirmar_senha =  
'senha_desejada';
```

Tabela Esqueceu_sua_senha :

```
SELECT * FROM Esqueceu_sua_senha;  
SELECT nome_completo, email FROM Esqueceu_sua_senha;
```

```
SELECT * FROM Esqueceu_sua_senha WHERE tipo_usuario = 'admin';
SELECT * FROM Esqueceu_sua_senha ORDER BY nome_completo;
SELECT tipo_usuario, COUNT(*) as total FROM Esqueceu_sua_senha GROUP BY
tipo_usuario;
```

Tabela Fornecedor :

```
SELECT * FROM Fornecedor;
SELECT nome_empresa, cnpj FROM Fornecedor;
SELECT * FROM Fornecedor WHERE nome_repre LIKE 'A%';
SELECT * FROM Fornecedor ORDER BY nome_empresa;
SELECT * FROM Fornecedor WHERE senha != confirmar_senha;
```

Tabela Funcionario :

```
SELECT * FROM Funcionario;
SELECT * FROM Funcionario WHERE nome_completo = 'NomeDesejado';
SELECT * FROM Funcionario WHERE userName = 'UserNameDesejado';
SELECT * FROM Funcionario WHERE data_nasc = 'DataNascimentoDesejada';
SELECT * FROM Funcionario WHERE email = 'EmailDesejado';
SELECT * FROM Funcionario WHERE tele = 'TelefoneDesejado';
```

Tabela Administrador :

```
SELECT * FROM Administrador;
SELECT * FROM Administrador WHERE nome_completo = 'Nome Completo';
SELECT * FROM Administrador WHERE userName = 'NomeDeUsuario';
SELECT * FROM Administrador WHERE data_nasc = 'YYYY-MM-DD';
SELECT * FROM Administrador WHERE email = 'exemplo@email.com';
SELECT * FROM Administrador WHERE tele = '123456789';
```

Tabela MenuFuncionario :

```
SELECT * FROM MenuFuncionario;
SELECT * FROM MenuFuncionario WHERE id = 1;
SELECT * FROM MenuFuncionario WHERE opcao_menu = 'NomeDoMenu';
SELECT * FROM MenuFuncionario ORDER BY id DESC;
SELECT COUNT(*) FROM MenuFuncionario;
```

Tabela VizualizacaoEstoque :

```
SELECT * FROM VisualizacaoEstoque;
SELECT * FROM VisualizacaoEstoque WHERE esta_no_estoque = true;
SELECT COUNT(*) AS total_registros FROM VisualizacaoEstoque;
```



```
SELECT DISTINCT item_estoque FROM VisualizacaoEstoque WHERE esta_no_estoque = true;
```

Tabela Pagamento :

```
SELECT * FROM Pagamento;  
SELECT * FROM Pagamento WHERE metodo_pagamento = 'Cartão de Crédito';  
SELECT * FROM Pagamento ORDER BY id DESC;  
SELECT metodo_pagamento, COUNT(*) as quantidade FROM Pagamento GROUP BY metodo_pagamento;
```

Tabela CadastrarProduto :

```
SELECT * FROM CadastrarProduto;  
SELECT * FROM CadastrarProduto ORDER BY id ASC;  
SELECT * FROM CadastrarProduto WHERE preço > X;  
SELECT * FROM CadastrarProduto WHERE nome_produto LIKE '%palavra%';  
SELECT AVG(preço) AS media_preco FROM CadastrarProduto;
```

Tabela Produtos :

```
SELECT * FROM Produtos;  
SELECT * FROM Produtos WHERE preco > 50;  
SELECT * FROM Produtos WHERE quant_estoque <= 10;  
SELECT * FROM Produtos WHERE nome_fornecedor = 'Nome do Fornecedor';  
SELECT descricao, preco FROM Produtos WHERE quant_estoque > 20;
```

Tabela Estoque :

```
SELECT * FROM Estoque;  
SELECT nome_produto, quant FROM Estoque WHERE quant > 10;  
SELECT * FROM Estoque WHERE data_cadastro > '2023-01-01';  
SELECT nome_produto, SUM(quant) AS quantidade_total, AVG(preco_unitario) AS preco_medio FROM Estoque GROUP BY nome_produto;  
SELECT * FROM Estoque ORDER BY preco_unitario DESC;
```

Tabela ProdutoFornecedor :

```
SELECT * FROM ProdutoFornecedor;  
SELECT * FROM ProdutoFornecedor WHERE fornecedor_id = 'id_fornecedor';  
SELECT COUNT(*) AS total_registros FROM ProdutoFornecedor;
```

```
SELECT produto_id, fornecedor_id FROM ProdutoFornecedor;
SELECT produto_id, COUNT(DISTINCT fornecedor_id) AS num_fornecedores
FROM ProdutoFornecedor GROUP BY produto_id HAVING num_fornecedores > 1;
```

Tabela EntradaMercadoria :

```
SELECT * FROM EntradaMercadoria;
SELECT * FROM EntradaMercadoria WHERE produto_id = 'id_do_produto';
SELECT * FROM EntradaMercadoria WHERE quant > 10;
SELECT * FROM EntradaMercadoria ORDER BY data_entrada DESC;
SELECT produto_id, SUM(quant) AS total_quantidade FROM EntradaMercadoria GROUP
BY produto_id;
```

Tabela SaidaMercadoria :

```
SELECT * FROM SaidaMercadoria;
SELECT * FROM SaidaMercadoria WHERE data_saida > '2023-01-01';
SELECT produto_id, SUM(quant) AS total_quantidade FROM SaidaMercadoria GROUP
BY produto_id;
SELECT * FROM SaidaMercadoria ORDER BY data_saida DESC LIMIT 10;
SELECT YEAR(data_saida) as ano, SUM(quant) AS total_quantidade FROM
SaidaMercadoria GROUP BY YEAR(data_saida);
```

Tabela ControleCustos :

```
SELECT * FROM ControleCustos;
SELECT * FROM ControleCustos WHERE valor = 100.00;
SELECT * FROM ControleCustos ORDER BY valor DESC;
SELECT * FROM ControleCustos WHERE data_registro > '2023-01-01';
SELECT COUNT(*) FROM ControleCustos;
```

Tabela GerenciamentoLotes :

```
SELECT * FROM GerenciamentoLotes;
SELECT * FROM GerenciamentoLotes WHERE produto = 'nome_do_produto';
SELECT * FROM GerenciamentoLotes WHERE quant > 100;
SELECT * FROM GerenciamentoLotes WHERE data_validade < CURDATE();
SELECT * FROM GerenciamentoLotes ORDER BY data_validade;
```

Tabela ControlePrecos :

```
SELECT * FROM ControlePrecos;
SELECT * FROM ControlePrecos WHERE id = 1;
SELECT * FROM ControlePrecos ORDER BY data_modificacao DESC;
SELECT * FROM ControlePrecos WHERE preco > 50.00;
```

```
SELECT COUNT(*) FROM ControlePrecos;
```

Tabela RelatorioVendaEstoque :

```
SELECT * FROM RelatorioVendaEstoque;  
SELECT * FROM RelatorioVendaEstoque WHERE total_vendas > 1000;  
SELECT * FROM RelatorioVendaEstoque ORDER BY data_relatorio DESC;  
SELECT AVG(total_vendas) AS media_vendas FROM RelatorioVendaEstoque;  
SELECT * FROM RelatorioVendaEstoque WHERE total_estoque < 50;
```

Tabela EstoqueTamanhoCor :

```
SELECT * FROM EstoqueTamanhoCor;  
SELECT tamanho, SUM(quant) AS total_quantidade FROM EstoqueTamanhoCor  
GROUP BY tamanho;  
SELECT cor, SUM(quant) AS quantidade_disponivel FROM EstoqueTamanhoCor  
GROUP BY cor;  
SELECT produto, tamanho, cor, quant FROM EstoqueTamanhoCor WHERE quant < 10;  
SELECT * FROM EstoqueTamanhoCor ORDER BY quant DESC;
```

Tabela Venda :

```
SELECT * FROM Vendas;  
SELECT * FROM Vendas WHERE cliente = 'NomeDoCliente';  
SELECT * FROM Vendas WHERE produto = 'NomeDoProduto' ORDER BY data_venda;  
SELECT produto, AVG(quant) AS media_quantidade FROM Vendas GROUP BY produto;  
SELECT * FROM Vendas WHERE data_venda = '2023-11-03';
```

- **INSERT**

```
INSERT INTO Entrar (nome_completo, data_nasc, tele, email, senha, confirmar_senha,  
tipo_usuario) VALUES  
( 'João Silva', '1990-05-15', '1234567890', 'joao@example.com', 'senha123', 'senha123',  
'Funcionario'),  
( 'Kauan Oliveira', '2000-01-01', '13355779076', 'kauan@example.com', 'olikauan', 'olikauan',  
'Fornecedor'),  
( 'Ryane Pereira', '1999-09-10', '51577208234', 'Ryane@example.com', 'nany678', 'nany678',  
'Funcionario'),
```

('Maria José', '1975-07-20', '90076587790', JoséMaria@example.com, 'jmaria9', 'jmaria9', 'Administrador'),
('Caio Bruno', '1998-06-28', '26367896540', brunocaio@example.com, 'bruninhocaio', 'bruninhocaio', 'Fornecedor'),
('Conrad Fisher', '2000-04-10', '56789076570', teamConrad@example.com, 'conrad#43', 'conrad#43', 'Administrador');

INSERIR EM Esqueceu_sua_senha (nome, email, nova_senha, confirmar_nova_senha, tipo_usuario) VALUES

('Joao Silva', joao@example.com, 'ahnes321', 'ahnes321', 'Funcionario'),
('Kauan Oliveira', kauan@example.com, 'kauan_457#', 'kauan_457#', 'Fornecedor'),
('Ryane Pereira', Ryane@example.com, 'nanyzinha90#', 'nanyzinha90#', 'Funcionario'),
('Maria Jose', JoseMaria@example.com, 'novasenha456', 'novasenha456', 'Administrador'),
('Caio Bruno', brunocaio@example.com, '46790!', '46790!', 'Fornecedor'),
('Conrad Fisher', teamConrad@example.com, '#34Fisher', '#34Fisher', 'Administrador');

INSERT INT Fornecedor (nome_empresa, cnpj, tele_comercial, email_comercial, nome_repre, email_repre, tele_repre, senha, confirmar_senha, botoes) VALUES

('Fornecedor ABC', '12345678901234', '9876543210', 'fornecedor@example.com', ' Mario Representante ', ' representanteMario@example.com', '9876543210', 'fornecedor123', 'fornecedor123', 'ativo'),
('adivinha', '34576509854323', '9785634103', karinafor@example.com, '9765456780', 'anirak', 'anirak', 'inativo');

INSERT INTO Funcionario (nome_completo, userName, data_nasc, tele, email, senha, confirmar_senha, botoes) VALUES

('Pedro Da Silva Santos', 'pedro123', '1985-08-20', '5678901234', ' pedro@example.com ', 'senha456', 'senha456', 'ativo'),
('Ryane Da Silva Pereira', 'Ryane Pereira', '1999-09-10', '9291541585', Ryane@example.com, 'nany678', 'nany678', 'ativo'),
('João Souza Da Silva', 'Joao Silva', '1990-05-15', '9234567890', joao@example.com, 'senha123', ' senha123', 'inativo');

```
INSERT INTO Administrador (nome_completo, userName, data_nasc, tele, senha,
confirmar_senha, email, botoes) VALUES
('Ana Costa Souza', 'ana_admin', '1970-03-10', '9998887777', 'admin123', 'admin123' ,
'ana@exemplo.com', 'ativo'),
('Conrad Fisher', 'Conrad Fisher', '2000-04-10', '9298907657', 'conrad#43', 'conrad#43',
'teamConrad@example.com', 'inativo'),
('Maria José Pereira Da Silva', 'Maria Jose', '1975-07-20', '9297658779', 'jmaria9', 'jmaria9',
'JoséMaria@example.com', 'ativo');
```

```
INSERT INTO VisualizacaoEstoque (item_estoque, esta_no_estoque) VALUES
('Produto A', true),
('Produto B', true);
```

```
INSERT INTO Pagamento (metodo_pagamento, tipo_pagamento) VALUES
('Cartão de Crédito', 'Débito'),
('Dinheiro', 'Boleto');
```

```
INSERT INTO CadastrarProduto (nome_produto, descricao_produto, preco) VALUES
('Produto X', 'Descrição do Produto X', 29.99),
('Produto Y', 'Descrição do Produto Y', 19.99),
('Produto U', 'Descrição do Produto U', 59.99),
('Produto G', 'Descrição do Produto G', 89.99),
('Produto W', 'Descrição do Produto W', 75.99);
```

```
INSERT INTO Produtos (descricao, preco, quant_estoque, nome_fornecedor) VALUES
('Produto A', '49.99', '100', 'Fornecedor 1'),
('Produto B', '39.99', '75', 'Fornecedor 2'),
('Produto R', '89.99', '60', 'Fornecedor 28'),
('Produto T', '69.99', '200', 'Fornecedor 9');
('Produto E', '59.99', '86', 'Fornecedor 7');
```

```
INSERT INTO Estoque (nome_produto, quant, preco_unitario, data_cadastro) VALUES
('Produto A', '100', '49.99', '2023-09-17'),
('Produto T', '200', '69.99', '2023-09-17'),
```

```
('Produto B', '75', '39.99', '2023-09-17'),  
('Produto G', '150', '64.99', '2023-09-17'),  
('Produto H', '80', '99.99', '2023-09-17');
```

```
INSERT INTO ProdutoFornecedor (produto_id, fornecedor_id) VALUES  
('60', '761'),  
('80', '77'),  
('8', '71'),  
('90', '70'),  
('89', '76');
```

```
INSERT INTO EntradaMercadoria (produto_id, quant, data_entrada) VALUES  
('60', '50', '2023-09-18'),  
('80', '90', '2023-08-10'),  
('8', '100', '2023-04-08'),  
('90', '80', '2023-01-18'),  
('89', '200', '2023-05-17');
```

```
INSERT INTO SaidaMercadoria (produto_id, quant, data_saida) VALUES  
('60', '10', '2023-09-29'),  
('80', '30', '2023-09-19'),  
('8', '70', '2023-05-29'),  
('90', '50', '2023-01-30'),  
('89', '100', '2023-06-01');
```

```
INSERT INTO ControleCustos (descricao, valor, data_registro) VALUES  
('Custo de transporte', '500.00', '2023-09-17'),  
('Custo de transporte', '200.00', '2023-08-19'),  
('Custo de transporte', '90.00', '2023-06-27'),  
('Custo de transporte', '400.00', '2023-05-08'),  
('Custo de transporte', '100.00', '2023-01-30');
```

```
INSERT INTO GerenciamentoLotes (produto, lote, quant, data_validade) VALUES  
('Produto A', 'Lote123', '100', '2023-09-10'),
```

```
('Produto G', 'Lote803', '200', '2023-02-01'),  
( 'Produto J', 'Lote923', '90', '2023-07-18'),  
( 'Produto U', 'Lote453', '150', '2023-05-30');
```

```
INSERT INTO ControlePrecos (produto, preco, data_modificacao) VALUES  
( 'Produto A', '27.99', '2023-09-07'),  
( 'Produto K', '87.99', '2023-08-17'),  
( 'Produto L', '90.99', '2023-04-19'),  
( 'Produto U', '70.99', '2023-02-09');
```

```
INSERT INTO RelatorioVendaEstoque (data_relatorio, total_vendas, total_estoque)  
VALUES  
( '2023-09-17', '1000.00', '70'),  
( '2023-09-27', '980.00', '80'),  
( '2023-06-07', '2000.00', '20'),  
( '2023-03-26', '300.00', '50'),
```

```
INSERT INTO EstoqueTamanhoCor (produto, tamanho, cor, quant) VALUES  
( 'Produto A', 'M', 'Azul', '20'),  
( 'Produto S', 'PP', 'Roxo', '80'),  
( 'Produto G', 'P', 'Vermelho', '100'),  
( 'Produto Y', 'M', 'Preto', '200'),  
( 'Produto L', 'G', 'Rosa', '70');
```

```
INSERT INTO Vendas (cliente, produto, quant, valor_total, data_venda) VALUES  
( 'Cliente A', 'Produto A', '5', '200.95', '2023-09-20'),  
( 'Cliente Y', 'Produto V', '8', '309.95', '2023-09-20'),  
( 'Cliente N', 'Produto D', '10', '399.95', '2023-09-20'),  
( 'Cliente W', 'Produto J', '3', '169.95', '2023-09-20'),  
( 'Cliente P', 'Produto U', '6', '280.95', '2023-09-20');
```

- **UNION**

```
SELECT 'Entrada' AS tipo, data_entrada AS data, produto_id AS id_produto, quant
FROM EntradaMercadoria
UNION
SELECT 'Saída' AS tipo, data_saida AS data, produto_id AS id_produto, quant
FROM SaidaMercadoria;
```

```
SELECT 'Venda' AS tipo, data_venda AS data, produto AS produto_nome, quant, valor_total
FROM Vendas
UNION
SELECT 'Estoque' AS tipo, data_cadastro AS data, nome_produto AS produto_nome, quant,
preco_unitario * quant AS valor_total
FROM Estoque;
```

- **VIEW**

```
CREATE VIEW view_ProdutosEmEstoque AS
SELECT E.id AS IDProduto, E.nome_produto AS NomeProduto,
       E.quant AS QuantidadeEmEstoque, E.preco_unitario AS PrecoUnitario,
       EF.nome_empresa AS NomeFornecedor FROM Estoque E
JOIN ProdutoFornecedor PF ON E.id = PF.produto_id
JOIN Fornecedor EF ON PF.fornecedor_id = EF.id;
```

```
CREATE VIEW view_InformacoesVendas AS
SELECT V.id AS IDVenda, V.cliente AS NomeCliente, V.produto AS NomeProduto,
       V.quant AS QuantidadeVendida, V.valor_total AS ValorTotal, V.data_venda AS DataVenda
FROM Vendas V;
```

- **INNER JOIN**

```
SELECT Produtos.id, Produtos.descricao, Produtos.preco, Produtos.quant_estoque,
Fornecedor.nome_empresa FROM Produtos
INNER JOIN ProdutoFornecedor ON Produtos.id = ProdutoFornecedor.produto_id
INNER JOIN Fornecedor ON ProdutoFornecedor.fornecedor_id = Fornecedor.id;
```

```
SELECT Vendas.id, Vendas.produto, Vendas.quant, Vendas.valor_total, Vendas.data_venda,
Fornecedor.nome_empresa, Vendas.cliente FROM Vendas
INNER JOIN Produtos ON Vendas.produto = Produtos.descricao
INNER JOIN ProdutoFornecedor ON Produtos.id = ProdutoFornecedor.produto_id
INNER JOIN Fornecedor ON ProdutoFornecedor.fornecedor_id = Fornecedor.id;
```

- **LEFT JOIN**


```
SELECT ProdutoFornecedor.id, ProdutoFornecedor.produto_id,  
ProdutoFornecedor.fornecedor_id, Fornecedor.nome_empresa  
FROM ProdutoFornecedor  
LEFT JOIN Fornecedor ON ProdutoFornecedor.fornecedor_id = Fornecedor.id;
```

```
SELECT Estoque.id, Estoque.nome_produto, Estoque.quant, Estoque.preco_unitario,  
Estoque.data_cadastro, EntradaMercadoria.quant AS quant_entrada  
FROM Estoque  
LEFT JOIN EntradaMercadoria ON Estoque.id = EntradaMercadoria.produto_id;
```

- **RIGHT JOIN**

```
SELECT * FROM ProdutoFornecedor  
RIGHT JOIN Fornecedor ON ProdutoFornecedor.fornecedor_id = Fornecedor.id;
```

```
SELECT * FROM Estoque  
RIGHT JOIN EntradaMercadoria ON Estoque.id = EntradaMercadoria.produto_id;
```

- **FULL OUTER JOIN**

```
SELECT * FROM Estoque  
LEFT JOIN EntradaMercadoria ON Estoque.id = EntradaMercadoria.produto_id  
UNION SELECT * FROM Estoque  
RIGHT JOIN EntradaMercadoria ON Estoque.id = EntradaMercadoria.produto_id;
```

```
SELECT * FROM ProdutoFornecedor  
LEFT JOIN Fornecedor ON ProdutoFornecedor.fornecedor_id = Fornecedor.id  
UNION SELECT * FROM ProdutoFornecedor  
RIGHT JOIN Fornecedor ON ProdutoFornecedor.fornecedor_id = Fornecedor.id;
```

- **PROCEDURE**

```
DELIMITER //
```

```
CREATE PROCEDURE InserirDadosEntrar(  
    userName VARCHAR(200),  
    senha VARCHAR(30),  
    confirmar_senha VARCHAR(30),
```

```

    tipo_usuario ENUM('Fornecedor', 'Funcionário', 'Administrador')
)
BEGIN
    INSERT INTO Entrar (userName, senha, confirmar_senha, tipo_usuario)
    VALUES ( userName, senha, confirmar_senha, tipo_usuario);
END //

DELIMITER ;

CALL InserirDadosEntrar('usuarioExemplo', 'senha123', 'senha123', 'Funcionário');

```

```

DELIMITER //

```

```

CREATE PROCEDURE ObterVendasPorCliente(
    nome_cliente VARCHAR(255)
)
BEGIN
    SELECT * FROM Vendas WHERE cliente = nome_cliente;
END //

```

```

DELIMITER ;

```

```

CALL ObterVendasPorCliente('Cliente Exemplo');

```

- **TRIGGERS**

```

DELIMITER //

```

```

CREATE TRIGGER atualizar_data_cadastro
BEFORE INSERT ON Estoque
FOR EACH ROW
SET NEW.data_cadastro = NOW();

```

```

//

```

```

DELIMITER ;

```

```

DELIMITER //

```

```

CREATE TRIGGER atualizar_quant_estoque
AFTER INSERT ON EntradaMercadoria
FOR EACH ROW
UPDATE Produtos
SET quant_estoque = quant_estoque + NEW.quant
WHERE id = NEW.produto_id;

```

```
//  
DELIMITER ;
```