

-- Criação do banco de dados para o cenário de E-commerce

```
CREATE DATABASE ecommerce;
```

```
USE ecommerce;
```

-- Tabela cliente

```
CREATE TABLE Clients (  
    idClient INT AUTO_INCREMENT PRIMARY KEY,  
    Fname VARCHAR(10),  
    Minit CHAR(3),  
    Lname VARCHAR(20),  
    CPF CHAR(11) NOT NULL,  
    Address VARCHAR(30),  
    Type ENUM('PF', 'PJ') NOT NULL,  
    CONSTRAINT unique_cpf_cliente UNIQUE (CPF)  
);
```

```
ALTER TABLE clients AUTO_INCREMENT = 1;
```

-- Tabela produto

```
CREATE TABLE Product (  
    idProduct INT AUTO_INCREMENT PRIMARY KEY,  
    Pname VARCHAR(10) NOT NULL,  
    Classification_kids BOOL,  
    Category ENUM('Eletronico', 'Vestimenta', 'Brinquedos', 'Alimentos', 'Moveis'),  
    Avaliacao FLOAT DEFAULT 0,  
    Size VARCHAR(10)  
);
```

-- Tabela pagamento

```
CREATE TABLE Payments (  
    idPayment INT AUTO_INCREMENT PRIMARY KEY,  
    TypePayment ENUM('Boleto', 'Cartão', 'Dois cartões'),  
    LimitAvailable FLOAT,  
    PRIMARY KEY (idPayment)  
);
```

-- Tabela pedido

```
CREATE TABLE Orders (  
    idOrder INT AUTO_INCREMENT PRIMARY KEY,  
    idOrderClient INT,  
    OrderStatus ENUM('Cancelado', 'Confirmado', 'Em processamento') DEFAULT 'Em processamento',  
    OrderDescription VARCHAR(255),  
    SendValue FLOAT DEFAULT 10,  
    PaymentCash BOOLEAN DEFAULT FALSE,  
    idPayment INT,  
    CONSTRAINT fk_order_client FOREIGN KEY (idOrderClient) REFERENCES Clients(idClient) ON  
    UPDATE CASCADE  
);
```

-- Tabela entrega

```
CREATE TABLE Delivery (  
    idDelivery INT AUTO_INCREMENT PRIMARY KEY,  
    idOrder INT,  
    Status ENUM('Em trânsito', 'Entregue', 'Pendente'),  
    TrackingCode VARCHAR(20),  
    FOREIGN KEY (idOrder) REFERENCES Orders(idOrder) ON UPDATE CASCADE  
);
```

-- Tabela estoque

```
CREATE TABLE ProductStorage (  
    idProdStorage INT AUTO_INCREMENT PRIMARY KEY,  
    StorageLocation VARCHAR(255),  
    Quantity INT DEFAULT 0  
);
```

-- Tabela fornecedor

```
CREATE TABLE Supplier (  
    idSupplier INT AUTO_INCREMENT PRIMARY KEY,  
    SocialName VARCHAR(255) NOT NULL,  
    CNPJ CHAR(15) NOT NULL,  
    Contact CHAR(11) NOT NULL,
```

```
CONSTRAINT unique_supplier UNIQUE (CNPJ)
);

-- Tabela vendedor
CREATE TABLE Seller (
    idSeller INT AUTO_INCREMENT PRIMARY KEY,
    SocialName VARCHAR(255) NOT NULL,
    AbstName VARCHAR(255),
    CNPJ CHAR(15),
    CPF CHAR(9),
    Location VARCHAR(255),
    Contact CHAR(11) NOT NULL,
    CONSTRAINT unique_cnpj_seller UNIQUE (CNPJ),
    CONSTRAINT unique_cpf_seller UNIQUE (CPF)
);
```

```
-- Tabela produtos_vendedor
CREATE TABLE ProductSeller (
    idPseller INT AUTO_INCREMENT PRIMARY KEY,
    idProduct INT,
    ProdQuantity INT DEFAULT 1,
    PRIMARY KEY (idPseller, idProduct),
    FOREIGN KEY (idPseller) REFERENCES Seller(idSeller),
    FOREIGN KEY (idProduct) REFERENCES Product(idProduct)
);
```

```
-- Tabela produto_pedido
CREATE TABLE ProductOrder (
    idPOproduct INT AUTO_INCREMENT PRIMARY KEY,
    idPOorder INT,
    POQuantity INT DEFAULT 1,
    POStatus ENUM('Disponivel', 'Sem estoque') DEFAULT 'Disponivel',
    PRIMARY KEY (idPOproduct, idPOorder),
    FOREIGN KEY (idPOproduct) REFERENCES Product(idProduct),
    FOREIGN KEY (idPOorder) REFERENCES Orders(idOrder)
```

);

-- Tabela local_loja

```
CREATE TABLE StorageLocation (  
    idLproduct INT,  
    idLstorage INT,  
    Location VARCHAR(255) NOT NULL,  
    PRIMARY KEY (idLproduct, idLstorage),  
    FOREIGN KEY (idLproduct) REFERENCES Product(idProduct),  
    FOREIGN KEY (idLstorage) REFERENCES ProductStorage(idProdStorage)  
);
```

-- Tabela produto_fornecedor

```
CREATE TABLE ProductSupplier (  
    idPsSupplier INT AUTO_INCREMENT PRIMARY KEY,  
    idPsProduct INT,  
    Quantity INT NOT NULL,  
    PRIMARY KEY (idPsSupplier, idPsProduct),  
    FOREIGN KEY (idPsSupplier) REFERENCES Supplier(idSupplier),  
    FOREIGN KEY (idPsProduct) REFERENCES Product(idProduct)  
);
```

-- Inserir dados de clientes

```
INSERT INTO Clients (Fname, Minit, Lname, CPF, Address, Type)  
VALUES  
    ('Maria', 'M', 'Silva', '123456789', 'Rua Silva da Prata 29, Carangola - Cidade das Flores', 'PF'),  
    ('Matheus', 'O', 'Pimentel', '987654321', 'Rua Alameda 289, Centro - Cidade das Flores', 'PF'),  
    ('Ricardo', 'F', 'Silva', '45678913', 'Avenida Alameda Vinha 1009, Centro - Cidade das Flores', 'PF'),  
    ('Julia', 'S', 'França', '789123456', 'Rua Laranjeiras 861, Centro - Cidade das Flores', 'PF'),  
    ('Roberta', 'G', 'Assis', '98745631', 'Avenida Koller 19, Centro - Cidade das Flores', 'PF'),  
    ('Isabela', 'M', 'Cruz', '654789123', 'Rua Alameda das Flores 28, Centro - Cidade das Flores', 'PF');
```

-- Inserir dados de produtos

```
INSERT INTO Product (Pname, Classification_kids, Category, Avaliacao, Size)  
VALUES
```

```
('Fone de ouvido', FALSE, 'Eletronico', 4, NULL),
('Barbie Elsa', TRUE, 'Brinquedos', 3, NULL),
('Body Carters', TRUE, 'Vestimenta', 5, NULL),
('Microfone Vedo - Youtuber', FALSE, 'Eletronico', 4, NULL),
('Sofá retrátil', FALSE, 'Moveis', 3, '3X57X80'),
('Farinha de arroz', FALSE, 'Alimentos', 2, NULL),
('Fire Stick Amazon', FALSE, 'Eletronico', 3, NULL);
```

-- Inserir dados de pedidos

```
INSERT INTO Orders (idOrderClient, OrderStatus, OrderDescription, SendValue, PaymentCash)
VALUES
(1, DEFAULT, 'Compra via aplicativo', NULL, 1),
(2, DEFAULT, 'Compra via aplicativo', 50, 0),
(3, 'Confirmado', NULL, NULL, 1),
(4, DEFAULT, 'Compra via website', 150, 0);
```

-- Inserir dados de produtos-pedidos

```
INSERT INTO ProductOrder (idPOproduct, idPOorder, POQuantity, POStatus)
VALUES
(1, 5, 2, NULL),
(2, 5, 1, NULL),
(3, 6, 1, NULL);
```

-- Inserir dados de produtos-loja

```
INSERT INTO ProductStorage (StorageLocation, Quantity)
VALUES
('Rio de Janeiro', 1000),
('Rio de Janeiro', 500),
('São Paulo', 10),
('São Paulo', 100),
('São Paulo', 10),
('Brasília', 60);
```

-- Inserir dados de local-loja

```
INSERT INTO StorageLocation (idLproduct, idLstorage, Location)
```

VALUES

(1, 2, 'RJ'),
(2, 6, 'GO');

-- Inserir dados de fornecedores

INSERT INTO Supplier (SocialName, CNPJ, Contact)

VALUES

('Almeida e Filhos', '123456789123456', '21985474'),
('Eletrônicos Silva', '854519649143457', '21985484'),
('Eletrônicos Valma', '934567893934695', '21975474');

-- Inserir dados de produtos-fornecedores

INSERT INTO ProductSupplier (idPsSupplier, idPsProduct, Quantity)

VALUES

(1, 1, 500),
(1, 2, 400),
(2, 2, 633),
(3, 3, 5),
(2, 5, 10);

-- Inserir dados de vendedores

INSERT INTO Seller (SocialName, AbstName, CNPJ, CPF, Location, Contact)

VALUES

('Tech Eletrônicos', NULL, '123456789456321', NULL, 'Rio de Janeiro', '219946287'),
('Boutique Durgas', NULL, NULL, '123456783', 'Rio de Janeiro', '219567895'),
('Kids World', NULL, '456789123654485', NULL, 'São Paulo', '1198657484');

-- Inserir dados de produtos-vendedores

INSERT INTO ProductSeller (idPseller, idProduct, ProdQuantity)

VALUES

(1, 6, 80),
(2, 7, 10);

-- Consultas:

-- Quantos pedidos foram feitos por cada cliente?

```
SELECT C.idClient, CONCAT(C.Fname, ' ', C.Lname) AS ClientName, COUNT(O.idOrder) AS TotalOrders
FROM Clients C
LEFT JOIN orders O ON C.idClient = O.idOrderClient
GROUP BY C.idClient;
```

-- Algum vendedor também é fornecedor?

```
SELECT S.socialName AS SellerName, S.CNPJ AS SellerCNPJ, F.socialName AS SupplierName, F.CNPJ
AS SupplierCNPJ
FROM seller S
LEFT JOIN productSeller PS ON S.idSeller = PS.idPseller
LEFT JOIN productSupplier FS ON PS.idProduct = FS.idPsProduct
LEFT JOIN supplier F ON FS.idPsSupplier = F.idSupplier
WHERE F.socialName IS NOT NULL;
```

-- Relação de produtos fornecedores e estoques:

```
SELECT P.Pname AS ProductName, F.socialName AS SupplierName, PS.quantity AS SupplierQuantity,
PS.location AS StorageLocation, PSL.quantity AS StorageQuantity
FROM product P
LEFT JOIN productSupplier PS ON P.idProduct = PS.idPsProduct
LEFT JOIN supplier F ON PS.idPsSupplier = F.idSupplier
LEFT JOIN storageLocation PSL ON P.idProduct = PSL.idLproduct;
```

-- Relação de nomes dos fornecedores e nomes dos produtos:

```
SELECT F.socialName AS SupplierName, P.Pname AS ProductName
FROM productSupplier PS
LEFT JOIN supplier F ON PS.idPsSupplier = F.idSupplier
LEFT JOIN product P ON PS.idPsProduct = P.idProduct;
```