

Trilha Prática 1

Link para o github com todos os códigos para teste:

<https://github.com/daniele-lins/Trilha-Pratica-BD>

1. MER atualizado

Para complementar o modelo entidade-relacionamento para o banco de dados da CompraEsperta, foram adicionadas as seguintes entidades, atributos e relacionamentos:

Entidades:

- **tbl_cliente**: tabela que concentra informações sobre os clientes que realizam pedidos.
 - **cp_cod_cliente** [int, incremental]: chave primária que identifica o cliente. Único e incremental.
 - **nm_cliente** [str, 200 caracteres]: nome completo do cliente.
 - **cpf_cliente** [str, 11 caracteres]: CPF do cliente.
- **tbl_pedido**: tabela que concentra informações sobre os pedidos que os clientes realizam.
 - **cp_id_pedido** [int, incremental]: chave primária que identifica o pedido. Único e incremental.
 - **ce_cod_cliente** [int, 8 bytes]: chave estrangeira que define o cliente autor do pedido.
 - **data_pedido** [date, 4 bytes]: data na qual o pedido foi realizado.
 - **hora_pedido** [time, 8 bytes]: horário no qual o pedido foi realizado.
 - **valor_total** [float, 4 bytes]: valor total do pedido. Precisão de duas casas decimais.

Atributos:

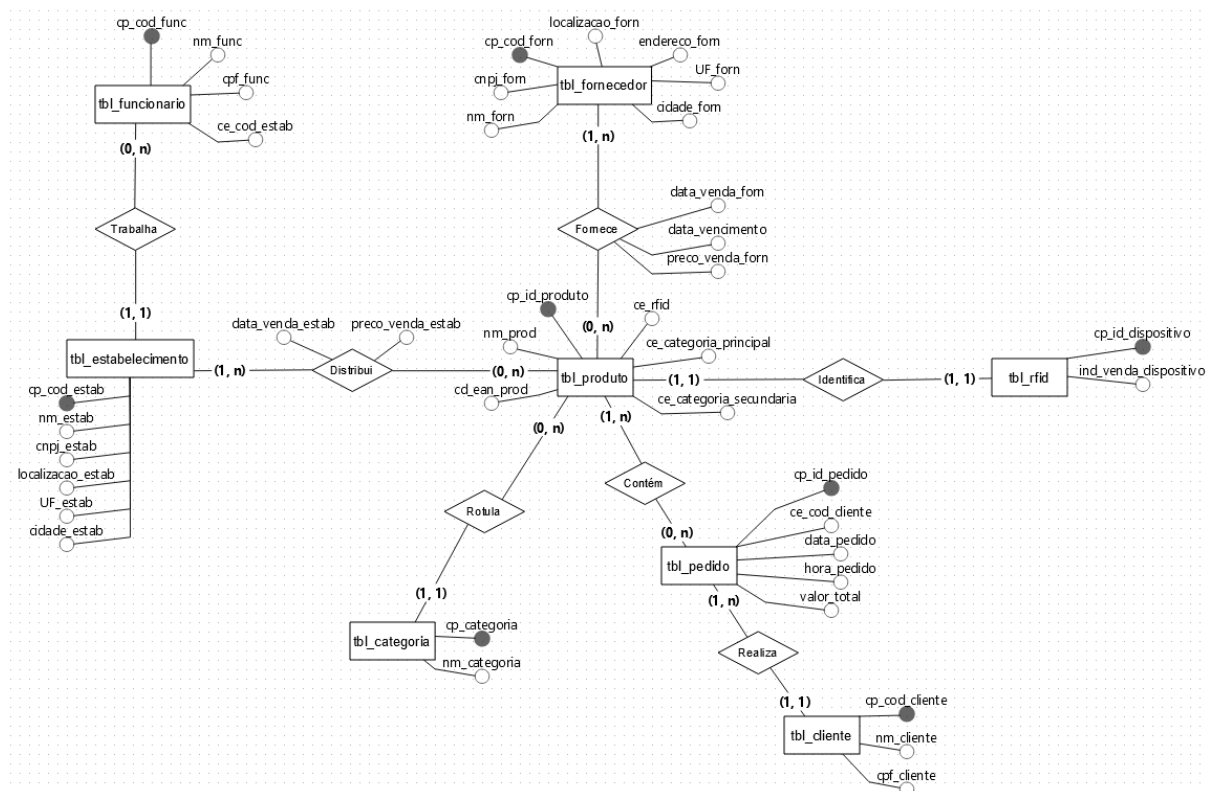
- **data_venda_forn** [date, 4 bytes]: data na qual o produto foi fornecido. Pertence ao relacionamento entre **tbl_fornecedor** e **tbl_produto**.
- **data_vencimento** [date, 4 bytes]: data na qual o produto fornecido irá vencer. Pertence ao relacionamento entre **tbl_fornecedor** e **tbl_produto**.
- **preco_venda_forn** [float, 4 bytes]: preço pelo qual um fornecedor vendeu um produto para um estabelecimento. Pertence ao relacionamento entre **tbl_fornecedor** e **tbl_produto**.
- **data_venda_estab** [date, 4 bytes]: data na qual o produto foi vendido em um estabelecimento. Pertence ao relacionamento entre **tbl_estabelecimento** e **tbl_produto**.
- **preco_venda_estab** [float, 4 bytes]: preço de venda de um produto em um estabelecimento. Pertence ao relacionamento entre **tbl_estabelecimento** e **tbl_produto**.
- **ce_cod_estab** [int, 8 bytes]: chave estrangeira que define o estabelecimento ao qual no qual o funcionário trabalha. Pertence a **tbl_funcionario**.
- **nm_forn** [str, 60 caracteres]: nome do fornecedor. Pertence a **tbl_fornecedor**.

Relacionamentos:

- Um cliente pode realizar um ou mais pedidos e um pedido só pode ser realizado por um cliente.
- Um pedido pode conter um ou vários produtos e um produto pode estar em nenhum ou vários pedidos.
- Em um estabelecimento podem trabalhar nenhum ou vários funcionários e um funcionário só pode estar registrado em um estabelecimento.

2. DER

Modelagem Conceitual:



O diagrama de banco de dados relacional para o sistema de vendas apresenta as seguintes tabelas e atributos:

- tbl_funcionario**: cp_cod_func: int (PK), nm_func: varchar(200), cpf_func: varchar(11) (U), funcao_func: varchar(40), ca_cod_estab: int (FK).
- tbl_estabelecimento**: cp_cod_estab: int (PK), nm_estab: varchar(60), cnpj_estab: varchar(14) (U), localizacao_estab: real(8), endereco_estab: varchar(100), uf_estab: varchar(2), cidade_estab: varchar(5).
- forn_produto**: fp_cod_forn: int (FK), fp_id_produto: int (FK), data_venda_forn: date, data_vencimento: date, preco_venda_forn: real.
- tbl_fornecedor**: cp_cod_forn: int (PK), cnpj_forn: varchar(14) (U), nm_forn: varchar(60), localizacao_forn: real(8), endereco_forn: varchar(100), uf_forn: varchar(2), cidade_forn: varchar(5).
- estab_produto**: ep_cod_estab: int (FK), ep_id_produto: int (FK), data_venda_estab: date, preco_venda_estab: real.
- tbl_produto**: cp_id_produto: serial (PK), ca_rfid: int (FK), cd_sen_prod: varchar(12), nm_produto: varchar(60), ca_categoria_principal: int (FK), ca_categoria_secundaria: int (FK).
- tbl_rfid**: cp_id_dispositivo: int (PK), ind_venda_dispositivo: boolean.
- tbl_categoria**: cp_cod_categoria: int (PK), nm_categoria: varchar(20).
- pedido_produto**: cp_id_pedido: int (FK), cp_id_produto: int (FK).
- tbl_pedido**: cp_id_pedido: serial (PK), ca_cod_cliente: int (FK), data_pedido: date, hora_pedido: time, valor_total: real.
- tbl_cliente**: cp_cod_cliente: serial (PK), nm_cliente: varchar(200), cpf_cliente: varchar(11) (U).

As relações entre as tabelas são as seguintes:

- tbl_funcionario** para **tbl_estabelecimento**: (0,n) para (1,1).
- tbl_estabelecimento** para **estab_produto**: (1,1) para (0,n).
- estab_produto** para **tbl_produto**: (1,n) para (1,1).
- forn_produto** para **tbl_fornecedor**: (0,n) para (1,1).
- forn_produto** para **tbl_produto**: (1,n) para (1,1).
- tbl_produto** para **tbl_rfid**: (1,1) para (1,1).
- tbl_categoria** para **tbl_produto**: (1,1) para (0,n).
- tbl_produto** para **pedido_produto**: (1,n) para (1,1).
- pedido_produto** para **tbl_pedido**: (1,1) para (1,n).
- tbl_pedido** para **tbl_cliente**: (1,n) para (1,1).

```
--CRIAR TABELAS--
CREATE TABLE tbl_rfid (
    cp_id_dispositivo int PRIMARY KEY,
    ind_venda_dispositivo boolean
);

CREATE TABLE tbl_categoria (
    cp_cod_categoria int PRIMARY KEY,
    nm_categoria varchar(20)
);
```

```
CREATE TABLE tbl_cliente (  
    cp_cod_cliente serial PRIMARY KEY,  
    nm_cliente varchar(200),  
    cpf_cliente varchar(11) UNIQUE  
);
```

```
CREATE TABLE tbl_estabelecimento (  
    cp_cod_estab int PRIMARY KEY,  
    nm_estab varchar(60),  
    cnpj_estab varchar(60) UNIQUE,  
    localizacao_estab real[8],  
    endereco_estab varchar(200),  
    uf_estab varchar(2),  
    cidade_estab varchar(5)  
);
```

```
CREATE TABLE tbl_fornecedor (  
    cp_cod_forn int PRIMARY KEY,  
    cnpj_forn varchar(14) UNIQUE,  
    nm_forn varchar(60),  
    localizacao_forn real[8],  
    endereco_forn int,  
    uf_forn varchar(2),  
    cidade_forn varchar(5)  
);
```

```
CREATE TABLE tbl_produto (  
    cp_id_produto serial PRIMARY KEY,  
    ce_rfid int UNIQUE REFERENCES tbl_rfid(cp_id_dispositivo),  
    cd_ean_prod varchar(12),  
    nm_produto varchar(60),  
    ce_categoria_principal int REFERENCES tbl_categoria(cp_cod_categoria),  
    ce_categoria_secundaria int REFERENCES tbl_categoria(cp_cod_categoria)  
);
```

```
CREATE TABLE tbl_funcionario (  
    cp_cod_func int PRIMARY KEY,  
    nm_func varchar(200),  
    cpf_func varchar(11) UNIQUE,  
    funcao_func varchar(40),  
    ce_cod_estab int REFERENCES tbl_estabelecimento(cp_cod_estab)  
);
```

```
CREATE TABLE tbl_pedido (  
    cp_id_pedido serial PRIMARY KEY,  
    ce_cod_cliente int REFERENCES tbl_cliente(cp_cod_cliente),  
    data_pedido date,
```

```

        hora_pedido time,
        valor_total real
    );

CREATE TABLE pedido_produto (
    cp_id_pedido int REFERENCES tbl_pedido(cp_id_pedido),
    cp_id_produto int REFERENCES tbl_produto(cp_id_produto),
    PRIMARY KEY (cp_id_pedido, cp_id_produto)
);

CREATE TABLE estab_produto (
    cp_cod_estab int REFERENCES tbl_estabelecimento(cp_cod_estab),
    cp_id_produto int REFERENCES tbl_produto(cp_id_produto),
    data_venda_estab date,
    preco_venda_estab real,
    PRIMARY KEY (cp_cod_estab, cp_id_produto)
);

CREATE TABLE forn_produto (
    cp_cod_forn int REFERENCES tbl_fornecedor(cp_cod_forn),
    cp_id_produto int REFERENCES tbl_produto(cp_id_produto),
    data_venda_forn date,
    data_vencimento date,
    preco_venda_forn real,
    PRIMARY KEY (cp_cod_forn, cp_id_produto)
);

-- Reiniciar todas as sequências para começarem em 1
ALTER SEQUENCE tbl_cliente_cp_cod_cliente_seq RESTART WITH 1;
ALTER SEQUENCE tbl_produto_cp_id_produto_seq RESTART WITH 1;
ALTER SEQUENCE tbl_pedido_cp_id_pedido_seq RESTART WITH 1;

```

4. População das tabelas

```

--tbl_rfid
DO $$
BEGIN
    FOR i IN 1..200 LOOP
        INSERT INTO tbl_rfid (cp_id_dispositivo, ind_venda_dispositivo)
        VALUES (i, 0::boolean);
    END LOOP;
END $$;

--tbl_categoria
DO $$
DECLARE
    v_nome_categoria varchar(20);
BEGIN

```

```

FOR i IN 1..200 LOOP
v_nome_categoria := 'Categoria ' || i;

INSERT INTO tbl_categoria (cp_cod_categoria, nm_categoria)
VALUES (i, v_nome_categoria);
END LOOP;
END $$;

--tbl_cliente
DO $$
DECLARE
    v_nome_cliente varchar(200);
    v_cpf_cliente varchar(11);
BEGIN
    FOR i IN 1..200 LOOP
        v_nome_cliente := 'Cliente ' || i;
        v_cpf_cliente := LPAD(i::text, 11, '0'); -- Gerando um CPF sequencial (00000000001,
00000000002, etc.)

        INSERT INTO tbl_cliente (nm_cliente, cpf_cliente)
        VALUES (v_nome_cliente, v_cpf_cliente);
    END LOOP;
END $$;

--tbl_estabelecimento
DO $$
DECLARE
    v_nome_estab varchar(60);
    v_cnpj_estab varchar(60);
    v_localizacao_estab real[8];
    v_endereco_estab varchar(200);
    v_uf_estab varchar(2);
    v_cidade_estab varchar(5);
BEGIN
    FOR i IN 1..200 LOOP
        v_nome_estab := 'Estabelecimento ' || i;
        v_cnpj_estab := LPAD(i::text, 14, '0') || '0001'; -- Gerando um CNPJ fictício com 14
dígitos
        v_localizacao_estab := ARRAY[
            random(), random(), random(), random(), random(), random(), random(), random()
        ]; -- Gerando 8 números reais aleatórios para a localização
        v_endereco_estab := 'Endereço ' || i;
        v_uf_estab := 'UF';
        v_cidade_estab := 'CDD';

        INSERT INTO tbl_estabelecimento (
            cp_cod_estab, nm_estab, cnpj_estab, localizacao_estab, endereco_estab, uf_estab,
            cidade_estab

```

```

    )
    VALUES (i, v_nome_estab, v_cnpj_estab, v_localizacao_estab, v_endereco_estab,
v_uf_estab, v_cidade_estab);
    END LOOP;
END $$;

```

```

--tbl_fornecedor

```

```

DO $$

```

```

DECLARE

```

```

    v_nome_forn varchar(60);

```

```

    v_cnpj_forn varchar(14);

```

```

    v_localizacao_forn real[8];

```

```

    v_endereco_forn int;

```

```

    v_uf_forn varchar(2);

```

```

    v_cidade_forn varchar(5);

```

```

BEGIN

```

```

    FOR i IN 1..200 LOOP

```

```

        v_nome_forn := 'Fornecedor ' || i;

```

```

        v_cnpj_forn := LPAD(i::text, 14, '0'); -- Gerando um CNPJ fictício com 14 dígitos

```

```

        v_localizacao_forn := ARRAY[

```

```

            random(), random(), random(), random(), random(), random(), random(), random()

```

```

        ]; -- Gerando 8 números reais aleatórios para a localização

```

```

        v_endereco_forn := i * 1000; -- Gerando um número inteiro para o endereço

```

```

        v_uf_forn := 'UF';

```

```

        v_cidade_forn := 'CDD';

```

```

        INSERT INTO tbl_fornecedor (

```

```

            cp_cod_forn, cnpj_forn, nm_forn, localizacao_forn, endereco_forn, uf_forn,

```

```

            cidade_forn

```

```

        )

```

```

        VALUES (i, v_cnpj_forn, v_nome_forn, v_localizacao_forn, v_endereco_forn,

```

```

v_uf_forn, v_cidade_forn);

```

```

    END LOOP;

```

```

END $$;

```

```

--tbl_produto

```

```

DO $$

```

```

DECLARE

```

```

    v_nome_produto varchar(60);

```

```

    v_cd_ean_prod varchar(12);

```

```

    v_ce_rfid int;

```

```

BEGIN

```

```

    FOR i IN 1..200 LOOP

```

```

        v_nome_produto := 'Produto ' || i;

```

```

        v_cd_ean_prod := LPAD(i::text, 12, '0'); -- Gerando um EAN fictício de 12 dígitos

```

```

        v_ce_rfid := i; -- Atribuindo o cp_id_dispositivo da tbl_rfid como ce_rfid

```

```

        INSERT INTO tbl_produto (

```

```

        ce_rfid, cd_ean_prod, nm_produto, ce_categoria_principal, ce_categoria_secundaria
    )
    VALUES (v_ce_rfid, v_cd_ean_prod, v_nome_produto, 1, 2); -- Categoria principal =
1, Secundária = 2
    END LOOP;
END $$;

```

```

--tbl_funcionario

```

```

DO $$

```

```

DECLARE

```

```

    v_nome_func varchar(200);

```

```

    v_cpf_func varchar(11);

```

```

    v_funcao_func varchar(40);

```

```

    v_ce_cod_estab int;

```

```

BEGIN

```

```

    FOR i IN 1..200 LOOP

```

```

        v_nome_func := 'Funcionário ' || i;

```

```

        v_cpf_func := LPAD(i::text, 11, '0'); -- Gerando um CPF fictício com 11 dígitos

```

```

        v_funcao_func := 'Função ' || i; -- Função genérica como "Função 1", "Função 2",

```

```

etc.

```

```

        v_ce_cod_estab := 1; -- Associando a um estabelecimento existente (usando o id de
1 a 200)

```

```

        INSERT INTO tbl_funcionario (

```

```

            cp_cod_func, nm_func, cpf_func, funcao_func, ce_cod_estab

```

```

        )

```

```

        VALUES (i, v_nome_func, v_cpf_func, v_funcao_func, v_ce_cod_estab);

```

```

    END LOOP;

```

```

END $$;

```

```

--tbl_pedido

```

```

DO $$

```

```

DECLARE

```

```

    v_data_pedido date;

```

```

    v_hora_pedido time;

```

```

    v_valor_total real;

```

```

    v_ce_cod_cliente int;

```

```

BEGIN

```

```

    v_data_pedido := '2025-01-18'; -- Data fixa para todos os pedidos

```

```

    v_hora_pedido := '12:00:00'; -- Hora fixa para todos os pedidos

```

```

    v_valor_total := 0; -- Valor total fixo igual a 0

```

```

    FOR i IN 1..200 LOOP

```

```

        v_ce_cod_cliente := i; -- Atribuindo o código do cliente sequencial

```

```

        INSERT INTO tbl_pedido (

```

```

            cp_id_pedido, ce_cod_cliente, data_pedido, hora_pedido, valor_total

```

```

        )

```



```

VALUES (i, v_ce_cod_cliente, v_data_pedido, v_hora_pedido, v_valor_total);
END LOOP;
END $$;

```

```

--pedido_produto
DO $$
DECLARE
    v_cp_id_pedido int;
    v_cp_id_produto int;
BEGIN
    FOR i IN 1..200 LOOP
        v_cp_id_pedido := i; -- Referenciando o pedido com o id igual à iteração
        v_cp_id_produto := i; -- Referenciando o produto com o id igual à iteração

        INSERT INTO pedido_produto (
            cp_id_pedido, cp_id_produto
        )
        VALUES (v_cp_id_pedido, v_cp_id_produto);
    END LOOP;
END $$;

```

```

--estab_produto
DO $$
DECLARE
    v_data_venda_estab date;
    v_preco_venda_estab real;
    v_cp_id_produto int;
BEGIN
    v_data_venda_estab := '2025-01-18'; -- Data fixa para todas as vendas
    v_preco_venda_estab := 2.50; -- Preço de venda fixo para todos os produtos

    FOR i IN 1..200 LOOP
        v_cp_id_produto := i; -- Atribui o código do produto sequencial

        INSERT INTO estab_produto (
            cp_cod_estab, cp_id_produto, data_venda_estab, preco_venda_estab
        )
        VALUES (1, v_cp_id_produto, v_data_venda_estab, v_preco_venda_estab);
    END LOOP;
END $$;

```

```

--forn_produto
DO $$
DECLARE
    v_data_venda_forn date;
    v_data_vencimento date;
    v_preco_venda_forn real;
    v_cp_id_produto int;

```

```
BEGIN
    v_data_venda_forn := '2025-01-18'; -- Data fixa para todas as vendas
    v_data_vencimento := '2025-12-31'; -- Data fixa para o vencimento de todos os
produtos
    v_preco_venda_forn := 2.00;          -- Preço fixo de venda para todos os produtos

    FOR i IN 1..200 LOOP
        v_cp_id_produto := i; -- Atribui o código do produto sequencial

        INSERT INTO forn_produto (
            cp_cod_forn, cp_id_produto, data_venda_forn, data_vencimento, preco_venda_forn
        )
        VALUES (1, v_cp_id_produto, v_data_venda_forn, v_data_vencimento,
v_preco_venda_forn);
    END LOOP;
END $$;
```