# Projeto Banco de Dados: Parte 2

## Responsáveis:

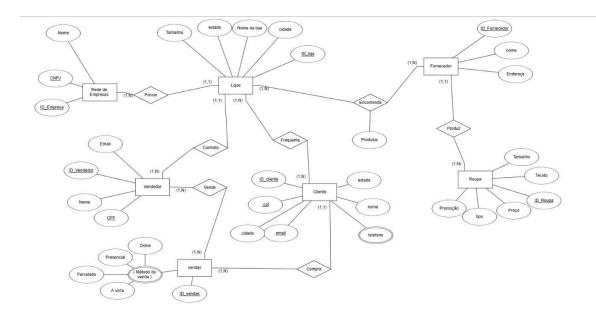
Gustavo Matias Félix; RA: 22.124.007-0 Lucas Galvano de Paula; RA: 22.124.015-3 Matheus Sarmento; RA: 22.124.062-5

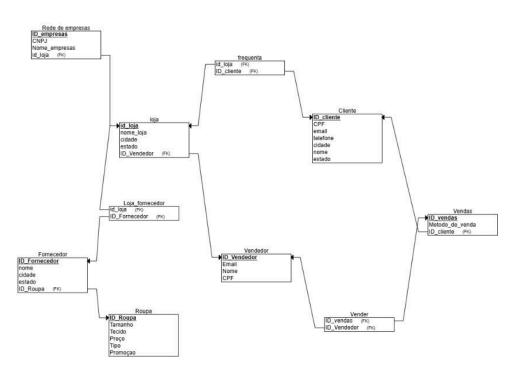
### 1. Descrição do Modelo Relacional

O sistema modelado representa uma rede de empresas que trabalham com vendas de roupas, envolvendo fornecedores, lojas, vendedores e clientes. A rede gerencia informações sobre:

- Empresas e Lojas: Cada empresa é identificada por um CNPJ e possui várias lojas associadas, que possuem informações como nome e localização.
- Fornecedores: São responsáveis por produzir e fornecer roupas para as lojas. Cada fornecedor está vinculado a uma ou mais roupas que produz.
- Roupas: Incluem informações como tamanho, tecido, preço, tipo e promoções.
- **Clientes**: Podem realizar compras nas lojas. As informações do cliente incluem CPF, nome, endereço, telefone e e-mail.
- **Vendedores**: Associados a uma loja, realizam vendas de diferentes métodos: online ou presencial.
- **Vendas**: Armazenam dados das transações realizadas, incluindo método de venda, cliente, vendedor, tipo de pagamento (à vista ou parcelado) e loja associada.

Imagens do Esquema MER e MR





## 2. Modelagem Conceitual

A modelagem conceitual foi realizada utilizando o diagrama de Entidade-Relacionamento (MER), que contém os seguintes elementos principais:

#### **Entidades**

- Rede de Empresas: Representada pela entidade "Rede de Empresas", identificada por ID Empresa.
- Loja: Identificada por ID\_Loja, possui atributos como Nome e Localização.
- Cliente: Identificado por ID\_Cliente, com atributos como CPF, Nome, E-mail, entre outros.
- Vendedor: Identificado por ID\_Vendedor, com atributos como Nome, CPF e E-mail.
- Fornecedor: Identificado por ID\_Fornecedor, com atributos como Nome, Cidade, Estado e Endereço.
- Roupa: Identificada por ID\_Roupa, com atributos como Tamanho, Tecido, Preço, Tipo e Promoção.
- Venda: Identificada por ID\_Venda, com atributos como Método de Venda e Tipo de Pagamento.

#### Relacionamentos

- 1. **Possui**: Relacionamento entre Rede de Empresas e Lojas (1:N).
- 2. **Produz**: Relacionamento entre Fornecedor e Roupa (1:N).
- 3. **Encomenda**: Relacionamento entre Lojas e Fornecedor (N:N).
- 4. Compra: Relacionamento entre Cliente e Venda (1:N).
- 5. **Vende**: Relacionamento entre Vendedor e Venda (1:N).
- 6. Frequenta: Relacionamento entre Cliente e Loja (N:N).
- 7. Contrata: Relacionamento entre Loja e Vendedor (1:N).
- 8. **Método de Venda**: Relacionamento de especialização que define se a venda foi online ou presencial.

# 3. Regras de Transformação Aplicadas

#### **Entidades e Relacionamentos**

## 1. Transformação de Entidades Simples:

- Cada entidade foi transformada em uma tabela no modelo relacional, onde o identificador da entidade se tornou a chave primária (PK).
- Exemplo:
  - Entidade Cliente → Tabela Cliente com PK ID\_Cliente.

#### 2. Relacionamentos 1:N.

o Foram resolvidos adicionando a chave primária da entidade do lado "1" como chave estrangeira (FK) na entidade do lado "N".

# • Exemplo:

■ O relacionamento entre Rede de Empresas e Loja foi implementado adicionando ID Empresa como FK na tabela Loja.

#### 3. Relacionamentos N:N:

 Foram resolvidos criando tabelas intermediárias, que possuem as chaves primárias das duas entidades relacionadas como FKs.

#### • Exemplo:

 O relacionamento Frequenta entre Cliente e Loja foi transformado em uma tabela intermediária chamada Frequenta, contendo ID\_Cliente e ID\_Loja.

### 4. Atributos Compostos:

 Atributos compostos foram desmembrados em atributos simples para adequação ao modelo relacional.

# o Exemplo:

 O atributo de Localização na entidade Loja foi dividido em Cidade e Estado.

#### 5. Atributos Derivados:

• Atributos derivados (calculados) foram removidos do modelo relacional e devem ser calculados dinamicamente durante consultas SQL.

#### • Exemplo:

■ O total de vendas por cliente não foi armazenado como um atributo, mas pode ser obtido por meio de consultas SQL.

#### Especialização/Generalização

 A especialização Método de Venda foi implementada utilizando uma coluna discriminadora (Metodo\_de\_venda) na tabela Vendas, armazenando os valores Online ou Presencial.

# 4. Normalização

# Primeira Forma Normal (1FN)

• Todas as tabelas estão em 1FN, garantindo que cada célula contenha valores atômicos e que não existam grupos repetitivos.

# • Exemplo:

 A tabela Cliente contém atributos como Nome, CPF, Telefone, sem atributos compostos ou multivalorados.

## Segunda Forma Normal (2FN)

• Todas as tabelas estão em 2FN, garantindo que não existam dependências parciais entre atributos e a chave primária.

## • Exemplo:

 Na tabela Vendas, atributos como Metodo\_de\_venda dependem totalmente da PK ID\_Venda.

# Justificativas para Tabelas Não-Normalizadas (Se Aplicável)

No contexto atual, todas as tabelas estão devidamente normalizadas.

# 5. Implementação no SGBD PostgreSQL

# Criação das Tabelas

O script de criação de tabelas foi implementado utilizando comandos DDL. Segue um exemplo da criação da tabela Cliente:

```
CREATE TABLE Cliente (
ID_Cliente SERIAL PRIMARY KEY,
Nome VARCHAR(100) NOT NULL,
CPF VARCHAR(11) NOT NULL UNIQUE,
Email VARCHAR(100),
Telefone VARCHAR(15),
Cidade VARCHAR(50),
Estado VARCHAR(50)
);
```

# Inserção de Dados

Instâncias significativas foram adicionadas a cada tabela para garantir que as consultas retornem resultados adequados. Exemplo:

```
INSERT INTO Cliente (ID_Cliente, Nome_Cliente, CPF, Email, Telefone, Cidade, Estado)
VALUES
('4', 'Larissa', '01100000000', 'larissafernandesmoya7@gmail.com', '131313131', 'Alta Floresta', 'Mato Grosso'),
('5', 'Gabriel', '10101010101', 'gabrielrosinelli@gmail.com', '999199942', 'Bangu', 'Rio de Janeiro');
```

# **Consultas SQL**

Exemplo de consulta para listar as vendas online:

```
SELECT *
FROM Vendas
WHERE Metodo_de_venda = 'Online';
```