



***FACULDADE DE TECNOLOGIA SENAI GASPAR RICARDO J UNIOR***

**Banco de Dados**

**Projeto Papelaria**

**André dos Santos Fernandes** (andrefs0199@hotmail.com);

**Hebert Santos Soares** (hebertsantossoaress@gmail.com);

**Rayan Luca Teixeira de Souza** (rayanlucateixeira10@gmail.com);

**Prof: André Souza;**

**Data de entrega:20/06/2025.**

# Sumário

## 1. Introdução

## 2. Modelagem de Dados

2.1. Entidades, Atributos e Relacionamentos

2.2. Diagrama Entidade-Relacionamento (DER)

## 3. Normalização

3.1. Primeira Forma Normal (1NF)

3.2. Segunda Forma Normal (2NF)

3.3. Terceira Forma Normal (3NF)

3.4. Justificativas da Normalização

## 4. Scripts SQL

4.1. Scripts de DDL (Data Definition Language)

4.1.1. Criação de Tabelas

4.1.2. Definição de Chaves Primárias e Estrangeiras

4.1.3. Restrições Adicionais

4.2. Scripts de DML (Data Manipulation Language)

4.2.1. Inserção de Dados (INSERT)

4.2.2. Atualização de Dados (UPDATE)

4.2.3. Exclusão de Dados (DELETE)

4.3. Scripts de DQL (Data Query Language)

4.3.1. Consultas SQL (SELECT)

## 5. Bonificação Extra

5.1. DCL (Data Control Language)

5.1.1. Controle de Acesso: GRANT e REVOKE

5.2. DTL (Data Transaction Language)

5.2.1. Controle de Transações: BEGIN, COMMIT, ROLLBACK e SAVEPOINT

## 6. Conclusão

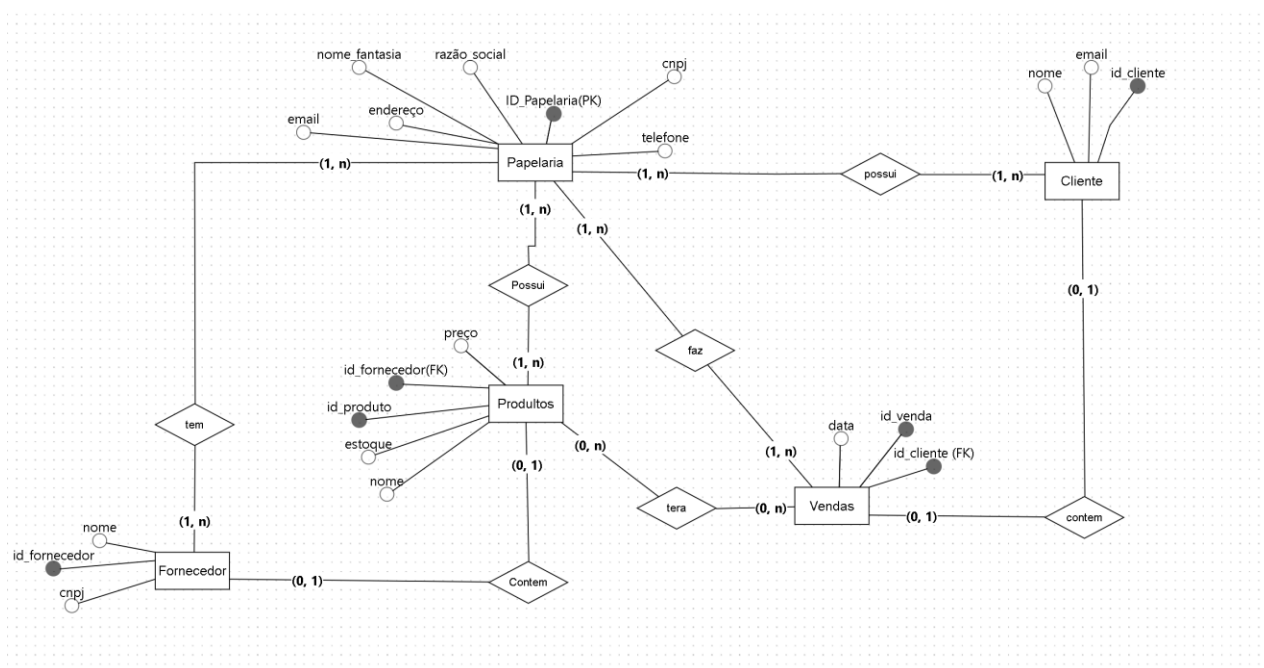
## 7. Bibliografia

# Introdução

Neste projeto, desenvolvemos um sistema completo de banco de dados, utilizando como tema uma papelaria. O objetivo principal foi aplicar os conceitos de modelagem de dados, normalização e manipulação de banco de dados por meio de comandos SQL. Ao longo do trabalho, realizamos a modelagem conceitual das informações da papelaria, representando as entidades, atributos e relacionamentos essenciais para o funcionamento do sistema. Além disso, aplicamos as regras de normalização até a Terceira Forma Normal (3NF), garantindo a integridade e a eficiência do banco de dados. Também elaboramos os scripts SQL necessários para a criação das tabelas, inserção, atualização, exclusão e consulta de dados, além de incluir recursos adicionais de controle de acesso (DCL) e transações (DTL) como forma de bonificação.

## Modelagem de dados

A modelagem de dados é uma representação visual de como os dados são estruturados e organizados em um sistema, sendo assim fundamental no desenvolvimento de uma aplicação, pois ajuda a definir como os dados serão coletados, armazenados, relacionados e acessados.



# Normalização

A normalização de um banco de dados é um processo usado para organizar as informações de forma mais eficiente. Ela ajuda a evitar dados repetidos e erros, deixando o banco de dados mais limpo, organizado e confiável.

## **TABELA Cliente**

**Forma Normal Aplicada:** 3NF

**Justificativa:**

- **1NF:** Valores atômicos, nome e e-mail únicos por cliente.
- **2NF:** Como a chave é simples (ID), não há dependências parciais.
- **3NF:** Todos os campos dependem apenas do ID, sem dependências transitivas.  
**Decisão de projeto:** Separação do telefone em outra tabela para permitir múltiplos números por cliente.

## **TABELA Telefone\_Cliente**

**Forma Normal Aplicada:** 3NF

**Justificativa:**

- **1NF:** Telefone armazenado de forma atômica, uma linha por número.
- **2NF:** Telefone depende totalmente da chave primária (ID).
- **3NF:** Sem dependências transitivas; relacionamento por chave estrangeira.  
**Decisão de projeto:** Permitir vários telefones por cliente sem redundância.

## **TABELA Fornecedor**

**Forma Normal Aplicada:** 3NF

**Justificativa:**

- **1NF:** Cada campo com um único valor.
- **2NF:** Todos os atributos dependem só do ID.
- **3NF:** Sem dependências transitivas.  
**Decisão de projeto:** CNPJ com restrição UNIQUE para evitar duplicidade.

## **TABELA Telefone\_Fornecedor**

**Forma Normal Aplicada:** 3NF

**Justificativa:**

- **1NF:** Telefones armazenados separadamente.
- **2NF:** Todos os campos dependem da chave primária (ID).
- **3NF:** Sem dependências entre atributos que não sejam chave.  
**Decisão de projeto:** Suporte a múltiplos telefones por fornecedor.

## **TABELA Produto**

**Forma Normal Aplicada:** 3NF

**Justificativa:**

- **1NF:** Informações atômicas por produto.
- **2NF:** Cada campo depende apenas do ID.
- **3NF:** Relacionamento com fornecedor por chave estrangeira, sem redundâncias.  
**Decisão de projeto:** Evitar repetição de dados do fornecedor nos produtos.

### TABELA Venda

Forma Normal Aplicada: 3NF

Justificativa:

- **1NF:** Dados atômicos (quantidade, data, IDs).
- **2NF:** Campos dependentes apenas do ID da venda.
- **3NF:** Cliente e produto relacionados por chaves estrangeiras, sem dependências transitivas.  
**Decisão de projeto:** Estrutura limpa para consultas históricas e integridade entre cliente, produto e venda.

## • Tabelas não normalizadas:

Clientes:			
ID	Nome	Email	Telefone
1	Carlos Souza	carlossouza21@gmail.com	(15) 9828-2177
2	Mario Augusto	quemario30@gmail.com	(15) 99730-1432
3	Margarete Katarina	margakate00@hotmail.com	(11) 99021-7732
4	Thomas Trem	thototrem@gmail.com	(15) 99879-2060
5	Solange Raimunda	solrai777@hotmail.com	(11) 3026-1900
Fornecedores:			
ID	Nome	CNPJ	Telefone
1	Fornecedor Park Papelaria	12.885.778/0001-90	(15) 99189-3021
2	Fornecedor Material Prima	12.773.221/0001-20	(15) 99736-1439
3	Fornecedor Lápís da Silva	15.860.339/0001-50	(15) 4002-8922

Produtos:				
ID	Nome	Preço Unitário	Estoque	Fornecedor ID
1	Caderno Espiral	40.90	100	1
2	Lápis Preto	1.20	300	3
3	Mochila	100.00	50	2
4	Caneta	2.00	1000	3
5	Borracha	1.00	1000	2
Vendas:				
ID	Cliente ID	Produto ID	Quantidade	Data da Venda
1	1	2	10	2025-06-04
2	2	1	2	2025-06-10
3	3	4	5	2025-06-12
4	4	4	4	2025-06-17
5	5	2	18	2025-06-23

- Tabelas normalizadas:

Tabela: Clientes			
id_cliente	nome	email	telefone
1	Carlos Souza	<a href="mailto:carlossouza21@gmail.com">carlossouza21@gmail.com</a>	(15) 9828-2177
2	Mario Augusto	<a href="mailto:quemario30@gmail.com">quemario30@gmail.com</a>	(15) 99730-1432
3	Margarete Katarina	<a href="mailto:margakate00@hotmail.com">margakate00@hotmail.com</a>	(11) 99021-7732
4	Thomas Trem	<a href="mailto:thototrem@gmail.com">thototrem@gmail.com</a>	(15) 99879-2060
5	Solange Raimunda	<a href="mailto:solrai777@hotmail.com">solrai777@hotmail.com</a>	(11) 3026-1900
Tabela: Fornecedores			
id_fornecedor	nome	cnpj	telefone
1	Fornecedor Park Papelaria	12.885.778/0001-90	(15) 99189-3021
2	Fornecedor Material Prima	12.773.221/0001-20	(15) 99736-1439
3	Fornecedor Lápis da Silva	15.860.339/0001-50	(15) 4002-8922

Tabela: Produtos				
id_produto	nome	preco_unitario	estoque	id_fornecedor
1	Caderno Espiral	40.90	100	1
2	Lápis Preto	1.20	300	3
3	Mochila	100.00	50	2
4	Caneta	2.00	1000	3
5	Borracha	1.00	1000	2

Tabela: Vendas				
id_venda	id_cliente	id_produto	quantidade	data_venda
1	1	2	10	2025-06-04
2	2	1	2	2025-06-10
3	3	4	5	2025-06-12
4	4	4	4	2025-06-17
5	5	2	18	2025-06-23

## Scripts de DDL, DML e DQL

Em SQL, existem três principais tipos de comandos:

- **DDL (Data Definition Language – Linguagem de Definição de Dados):**  
Responsável por criar, alterar ou excluir estruturas do banco de dados, como tabelas e índices.  
**Exemplos:** CREATE, ALTER, DROP.
- **DML (Data Manipulation Language – Linguagem de Manipulação de Dados):**  
Manipula os dados dentro das tabelas, permitindo inserir, atualizar ou excluir informações.  
**Exemplos:** INSERT, UPDATE, DELETE.
- **DQL (Data Query Language – Linguagem de Consulta de Dados):**  
Usada para consultar e recuperar dados armazenados.  
**Exemplo principal:** SELECT.

**Resumo:**

**DDL** define a estrutura, **DML** manipula os dados e **DQL** consulta as informações no banco de

dados.

- **Scripts Papelaria:**

## **DDL-criação de tabelas**

### **-- Tabela Clientes**

```
CREATE TABLE Clientes (  
    id_cliente INT PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL, email VARCHAR(100),  
    telefone VARCHAR(20) );
```

### **-- Tabela Fornecedores**

```
CREATE TABLE Fornecedores (  
    id_fornecedor INT PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    cnpj VARCHAR(20) UNIQUE NOT NULL,  
    telefone VARCHAR(20) );
```

### **-- Tabela Produtos**

```
CREATE TABLE Produtos (  
    id_produto INT PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    preco DECIMAL(10,2) NOT NULL,  
    estoque INT NOT NULL,  
    id_fornecedor INT,  
    FOREIGN KEY (id_fornecedor) REFERENCES Fornecedores(id_fornecedor)  
);
```



-- Tabela Vendas

```
CREATE TABLE Vendas (  
  id_venda INT PRIMARY KEY,  
  id_cliente INT, id_produto INT,  
  quantidade INT NOT NULL,  
  data_venda DATE NOT NULL,  
  FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente),  
  FOREIGN KEY (id_produto) REFERENCES Produtos(id_produto)  
);
```

## DML – Inserção de Dados

-- Inserir Clientes

```
INSERT INTO Clientes (id_cliente, nome, email, telefone)  
VALUES (1, 'Carlos Souza', 'carlossouza21@gmail.com', '(15) 9828-2177'),  
      (2, 'Mario Augusto', 'quemario30@gmail.com', '(15) 99730-1432'),  
      (3, 'Margarete Katarina', 'margakate00@hotmail.com', '(11) 99021-7732'),  
      (4, 'Thomas Trem', 'thototrem@gmail.com', '(15) 99879-2060'),  
      (5, 'Solange Raimunda', 'solrai777@hotmail.com', '(11) 3026-1900');
```

-- Inserir Fornecedores

```
INSERT INTO Fornecedores (id_fornecedor, nome, cnpj, telefone)  
VALUES (1, 'Fornecedor Park Papelaria', '12.885.778/0001-90', '(15) 99189-3021'),  
      (2, 'Fornecedor Material Prima', '12.773.221/0001-20', '(15) 99736-1439'),  
      (3, 'Fornecedor Lápis da Sivla', '15.860.339/0001-50', '(15) 4002-8922');
```

-- Inserir Produtos

```
INSERT INTO Produtos (id_produto, nome, preco, estoque, id_fornecedor)  
VALUES (1, 'Caderno Espiral', 40.90, 100, 1),  
      (2, 'Lápis Preto', 1.20, 300, 3),
```

```
(3, 'Mochila', 100.00, 50, 2),  
(4, 'Caneta', 2.00, 1000, 3),  
(5, 'Borracha', 1.00, 1000, 2);
```

*-- Inserir Vendas*

```
INSERT INTO Vendas (id_venda, id_cliente, id_produto, quantidade, data_venda)  
VALUES (1, 1, 2, 10, '2025-06-04'),  
(2, 2, 1, 2, '2025-06-10'),  
(3, 3, 4, 5, '2025-06-12'),  
(4, 4, 4, 4, '2025-06-17'),  
(5, 5, 2, 18, '2025-06-23');
```

## **DQL - Consultas SQL**

*-- Exemplos de consultas:*

*-- Listar todos os clientes*

```
SELECT * FROM Clientes;
```

*-- Listar todos os produtos com nome, preço e quantidade em estoque*

```
SELECT nome, preco, estoque FROM Produtos;
```

*-- Listar todas as vendas com nome do cliente, nome do produto e data da venda*

```
SELECT
```

```
Vendas.id_venda,
```

```
Clientes.nome AS cliente,
```

```
Produtos.nome AS produto,
```

```
Vendas.quantidade,
```

```
Vendas.data_venda
```

```
FROM Vendas
```

```
JOIN Clientes ON Vendas.id_cliente = Clientes.id_cliente
```

```
JOIN Produtos ON Vendas.id_produto = Produtos.id_produto;
```

-- Total de vendas por cliente

SELECT

Clientes.nome AS cliente,

SUM(Vendas.quantidade) AS total\_itens\_comprados

FROM Vendas

JOIN Clientes ON Vendas.id\_cliente = Clientes.id\_cliente

GROUP BY Clientes.nome;

-- Listar os produtos com estoque abaixo de 100 unidades

SELECT nome, estoque FROM Produtos WHERE estoque < 100;

## Bonificação Extra

### DCL – Controle de Acesso (GRANT e REVOKE)

-- Criar um usuário com permissões apenas de leitura

CREATE USER 'rayan\_luca1@email' IDENTIFIED BY '1234';

-- Conceder permissão de SELECT (somente leitura) ao usuário

GRANT SELECT ON papelaria.\* TO 'hebert\_lei32@email';

-- Criar um usuário com permissões de leitura e escrita

CREATE USER 'andre\_ger@email' IDENTIFIED BY '4321';

-- Conceder permissões de SELECT, INSERT, UPDATE e DELETE ao gerente

GRANT SELECT, INSERT, UPDATE, DELETE ON papelaria.\* TO 'andre\_ger@email';

-- Exemplo de revogar permissões (caso necessário)

---

```
REVOKE DELETE ON papelaria.* FROM 'hebert_lei32@email';
```

## **DTL – Controle de Transações (BEGIN, COMMIT, ROLLBACK, SAVEPOINT)**

```
-- Início da transação
```

```
START TRANSACTION;
```

```
-- Diminuir o estoque de um produto após uma venda
```

```
UPDATE Produtos SET estoque = estoque - 5 WHERE id_produto = 4;
```

```
-- Inserir a nova venda
```

```
INSERT INTO Vendas (id_venda, id_cliente, id_produto, quantidade, data_venda) VALUES (6, 3, 4, 5, '2025-06-24');
```

```
-- Criar um ponto de salvamento
```

```
SAVEPOINT depois_da_venda;
```

```
-- Simular um erro: tentar vender um produto que não existe
```

```
INSERT INTO Vendas (id_venda, id_cliente, id_produto, quantidade, data_venda) VALUES (7, 2, 999, 2, '2025-06-24'); -- Produto inexistente (erro)
```

```
-- Ao perceber o erro, fazer rollback até o ponto de salvamento
```

```
ROLLBACK TO depois_da_venda;
```

```
-- Finalizar a transação com sucesso
```

```
COMMIT;
```

## **Conclusão**

Durante o desenvolvimento deste projeto, tivemos a oportunidade de aprofundar nossos conhecimentos em Banco de Dados, aplicando na prática os conceitos aprendidos ao longo do primeiro semestre. Aprendemos a utilizar de forma eficiente as linguagens DDL, DML e DQL, essenciais para a criação, manipulação e consulta dos dados. Além disso, desenvolvemos habilidades importantes de modelagem de dados, incluindo a elaboração de diagramas entidade-relacionamento (DER) e a aplicação das formas normais no processo de normalização.

O projeto também nos proporcionou uma valiosa experiência em trabalho em equipe, permitindo a colaboração entre os membros do grupo para a resolução de problemas, divisão de tarefas e tomada de decisões. Essa vivência contribuiu não apenas para o nosso desenvolvimento técnico, mas também para o aprimoramento das nossas habilidades interpessoais e organizacionais.

Sem dúvida, os conhecimentos adquiridos serão fundamentais para os próximos desafios acadêmicos e profissionais na área de tecnologia da informação.

## Bibliografias

[https://www.google.com/search?q=scripts+dml+ddl+dql+oq+são&client=opera-gx&sca\\_esv=dcff718841928c00&sxsrf=AE3TifOj5A5YH\\_k-3Q8FxyQzbRzMIGWw8g%3A1750271955918&ei=0wdTaJTqN6W75OUPrvmigAQ&ved=0ahUKEwjUn8WRz\\_uNAxWIHbkGHa68CEAQ4dUDCBA&uact=5&oq=scripts+dml+ddl+dql+oq+são&gs\\_l=p=Egxnd3Mtd2l6LXNlcnAiG3NjcmlwdHMgZG1sIGRkbCBkcWwgb3Egc8OjbzIFECEYoAFI6idQ1ApYmx5wAHgCkAEAmAG9AaAB8QeqAQMwLje4AQPIAQD4AQGYAgigApglwglIEEAAYR8ICBxAhGKABGARCAgUQIRifBZgDAOIDBRIBMSBAiAYBkAYIkgcDMS43oAeuG7IHazAuN7gHkAjCBwUwLjUuM8gHGA&sclient=gws-wiz-serp](https://www.google.com/search?q=scripts+dml+ddl+dql+oq+são&client=opera-gx&sca_esv=dcff718841928c00&sxsrf=AE3TifOj5A5YH_k-3Q8FxyQzbRzMIGWw8g%3A1750271955918&ei=0wdTaJTqN6W75OUPrvmigAQ&ved=0ahUKEwjUn8WRz_uNAxWIHbkGHa68CEAQ4dUDCBA&uact=5&oq=scripts+dml+ddl+dql+oq+são&gs_l=p=Egxnd3Mtd2l6LXNlcnAiG3NjcmlwdHMgZG1sIGRkbCBkcWwgb3Egc8OjbzIFECEYoAFI6idQ1ApYmx5wAHgCkAEAmAG9AaAB8QeqAQMwLje4AQPIAQD4AQGYAgigApglwglIEEAAYR8ICBxAhGKABGARCAgUQIRifBZgDAOIDBRIBMSBAiAYBkAYIkgcDMS43oAeuG7IHazAuN7gHkAjCBwUwLjUuM8gHGA&sclient=gws-wiz-serp)

<https://www.google.com/search?client=opera-gx&q=normalização+banco+de+dados&sourceid=opera&ie=UTF-8&oe=UTF-8>

[file:///C:/Users/USER/Downloads/Documentação\\_Projeto\\_Integrador%20\(2\).pdf](file:///C:/Users/USER/Downloads/Documentação_Projeto_Integrador%20(2).pdf)