

DOCUMENTAÇÃO DO BANCO DE DADOS - Comercial CADSPM LTDA

ENTIDADES E ATRIBUTOS

Cliente

- id_cliente (PK)
- nome

Funcionario

- id_funcionario (PK)
- nome
- cargo

Produto

- id_produto (PK)
- nome_prod
- preco

Pedido

- id_pedido (PK)
- data_pedido

ItemPedido

- id_itemPedido (PK)
- fk_Pedido_id_pedido (FK)
- fk_Produto_id_produto (FK)
- quantidade
- subtotal

RELACIONAMENTOS

Faz (Cliente ↔ Pedido)

- Um cliente pode fazer vários pedidos.
- Um pedido é feito por exatamente um cliente.

Atende_ (Funcionario ↔ Pedido)

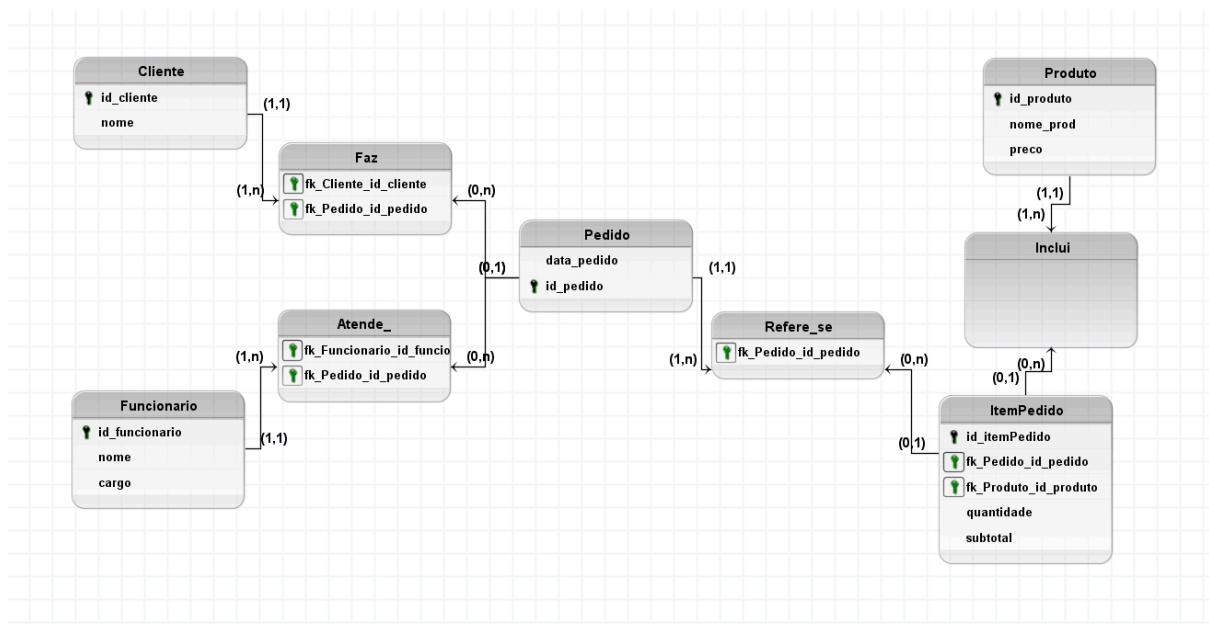
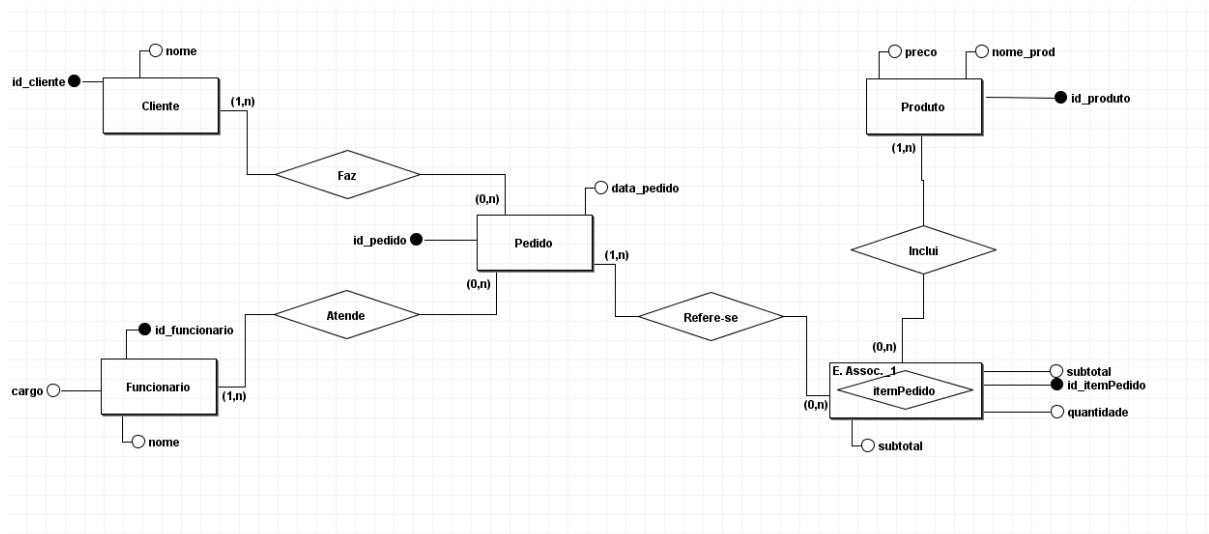
- Um pedido pode ser atendido por vários funcionários.
- Um funcionário pode atender vários pedidos.

Refere_se (Pedido ↔ ItemPedido)

- Relaciona cada pedido aos seus itens (linha do pedido).

Inclui (Produto ↔ ItemPedido)

- Cada itemPedido inclui um produto.
- Um produto pode aparecer em vários itemPedido.



REGRAS DE NEGÓCIO

- Todo pedido deve estar associado a pelo menos um cliente e um ou mais itens.
- Cada itemPedido é específico de um pedido e produto.
- Subtotal é calculado com base na quantidade × preço do produto.

—

-- Base de Dados: Comercial CADSPM LTDA

-- 1. 2 consultas com SELECT e WHERE

```
SELECT * FROM Cliente WHERE nome LIKE 'A%';
```

```
SELECT * FROM Produto WHERE preco > 100;
```

-- 2. 2 consultas com GROUP BY e ORDER BY com funções de agregação

```
SELECT fk_Pedido_id_pedido, SUM(subtotal) AS total_pedido
```

```
FROM ItemPedido
```

```
GROUP BY fk_Pedido_id_pedido
```

```
ORDER BY total_pedido DESC;
```

```
SELECT cargo, COUNT(*) AS qtd_funcionarios
```

```
FROM Funcionario
```

```
GROUP BY cargo
```

```
ORDER BY qtd_funcionarios DESC;
```

-- 3. 2 consultas com operadores aritméticos (+, -, *, /)

```
SELECT nome_prod, preco, preco * 0.90 AS preco_com_desconto FROM Produto;
```

```
SELECT quantidade, subtotal, subtotal / quantidade AS preco_unitario FROM ItemPedido;
```

-- 4. 3 consultas com operadores de comparação (=, !=, <, >, etc.)

```
SELECT * FROM Pedido WHERE id_pedido = 3;
```

```
SELECT * FROM Produto WHERE preco != 39.90;
```

```
SELECT * FROM ItemPedido WHERE quantidade > 2;
```

-- 5. 3 consultas com operadores lógicos (AND, OR)

```
SELECT * FROM Produto WHERE preco > 100 AND preco < 800;
```

```
SELECT * FROM Cliente WHERE nome LIKE 'J%' OR nome LIKE 'F%';
```

```
SELECT * FROM Funcionario WHERE cargo = 'Vendedor' AND nome LIKE 'A%';
```

-- 6. 2 consultas com operadores lógicos e negação (NOT)

```
SELECT * FROM Produto WHERE NOT preco < 100;
```

```
SELECT * FROM Cliente WHERE NOT nome LIKE '%a';
```

-- 7. 3 consultas com operadores auxiliares (IS NULL, BETWEEN, LIKE, IN)

```
SELECT * FROM Produto WHERE preco BETWEEN 50 AND 500;
```

```
SELECT * FROM Cliente WHERE nome LIKE '%a%';
```

```
SELECT * FROM Funcionario WHERE cargo IN ('Vendedor', 'Gerente');
```

-- 8. 3 consultas com funções de agregação (SUM(), AVG(), etc.)

```
SELECT SUM(preco) AS total_estoque FROM Produto;
```

```
SELECT AVG(preco) AS preco_medio FROM Produto;
```

```
SELECT MAX(subtotal) AS maior_subtotal FROM ItemPedido;
```

-- 9. 2 consultas com funções de datas (NOW(), DATE(), YEAR(), etc.)

```
SELECT NOW() AS data_atual;
```

```
SELECT * FROM Pedido WHERE YEAR(data_pedido) = 2025;
```

-- 10. 3 subconsultas com agrupamento e união de dados

```
SELECT nome, cargo FROM Funcionario
```

```
WHERE id_funcionario IN (
```

```
    SELECT fk_Funcionario_id_funcionario FROM Atende_
```

```
    WHERE fk_Pedido_id_pedido IN (
```

```
        SELECT id_pedido FROM Pedido WHERE data_pedido >= '2025-07-05'
```

```
    )
```

```
);
```

```
SELECT nome_prod, preco FROM Produto
```

```
WHERE id_produto IN (
```

```
    SELECT fk_Produto_id_produto FROM ItemPedido
```

```
    WHERE quantidade >= 2
```

```
);
```

```
SELECT nome FROM Cliente
```

```
WHERE id_cliente IN (
```

```
    SELECT fk_Cliente_id_cliente FROM Faz
```

```
    WHERE fk_Pedido_id_pedido IN (
```

```
        SELECT id_pedido FROM Pedido WHERE id_pedido <= 5
```

```
    )
```

```
);
```

-- 11. 3 consultas com JOIN e visualização de tabelas

```
SELECT c.nome AS cliente, p.id_pedido, p.data_pedido
```

```
FROM Cliente c
```

```
JOIN Faz f ON c.id_cliente = f.fk_Cliente_id_cliente
```

```
JOIN Pedido p ON f.fk_Pedido_id_pedido = p.id_pedido;
```

```
SELECT i.id_itemPedido, pr.nome_prod, i.quantidade, i.subtotal
```

```
FROM ItemPedido i
```

```
JOIN Produto pr ON i.fk_Produto_id_produto = pr.id_produto;
```

```
SELECT f.nome AS funcionario, p.id_pedido
```

```
FROM Funcionario f
```

```
JOIN Atende_ a ON f.id_funcionario = a.fk_Funcionario_id_funcionario
```

```
JOIN Pedido p ON a.fk_Pedido_id_pedido = p.id_pedido;
```

-- 12. 4 consultas com tipos de JOIN: INNER, LEFT, RIGHT

-- INNER JOIN

```
SELECT c.nome, p.id_pedido FROM Cliente c
```

```
INNER JOIN Faz f ON c.id_cliente = f.fk_Cliente_id_cliente
```

```
INNER JOIN Pedido p ON f.fk_Pedido_id_pedido = p.id_pedido;
```

-- LEFT JOIN

```
SELECT f.nome, p.id_pedido FROM Funcionario f
```

```
LEFT JOIN Atende_ a ON f.id_funcionario = a.fk_Funcionario_id_funcionario  
LEFT JOIN Pedido p ON a.fk_Pedido_id_pedido = p.id_pedido;
```

```
-- RIGHT JOIN
```

```
SELECT p.id_pedido, f.nome FROM Pedido p  
RIGHT JOIN Atende_ a ON p.id_pedido = a.fk_Pedido_id_pedido  
RIGHT JOIN Funcionario f ON a.fk_Funcionario_id_funcionario = f.id_funcionario;
```

```
-- INNER JOIN Produto com ItemPedido
```

```
SELECT pr.nome_prod, ip.quantidade FROM Produto pr  
INNER JOIN ItemPedido ip ON pr.id_produto = ip.fk_Produto_id_produto;
```