



UNIVERSIDADE  
FEDERAL  
DE PERNAMBUCO



**Centro de  
Informática**  
UFPE

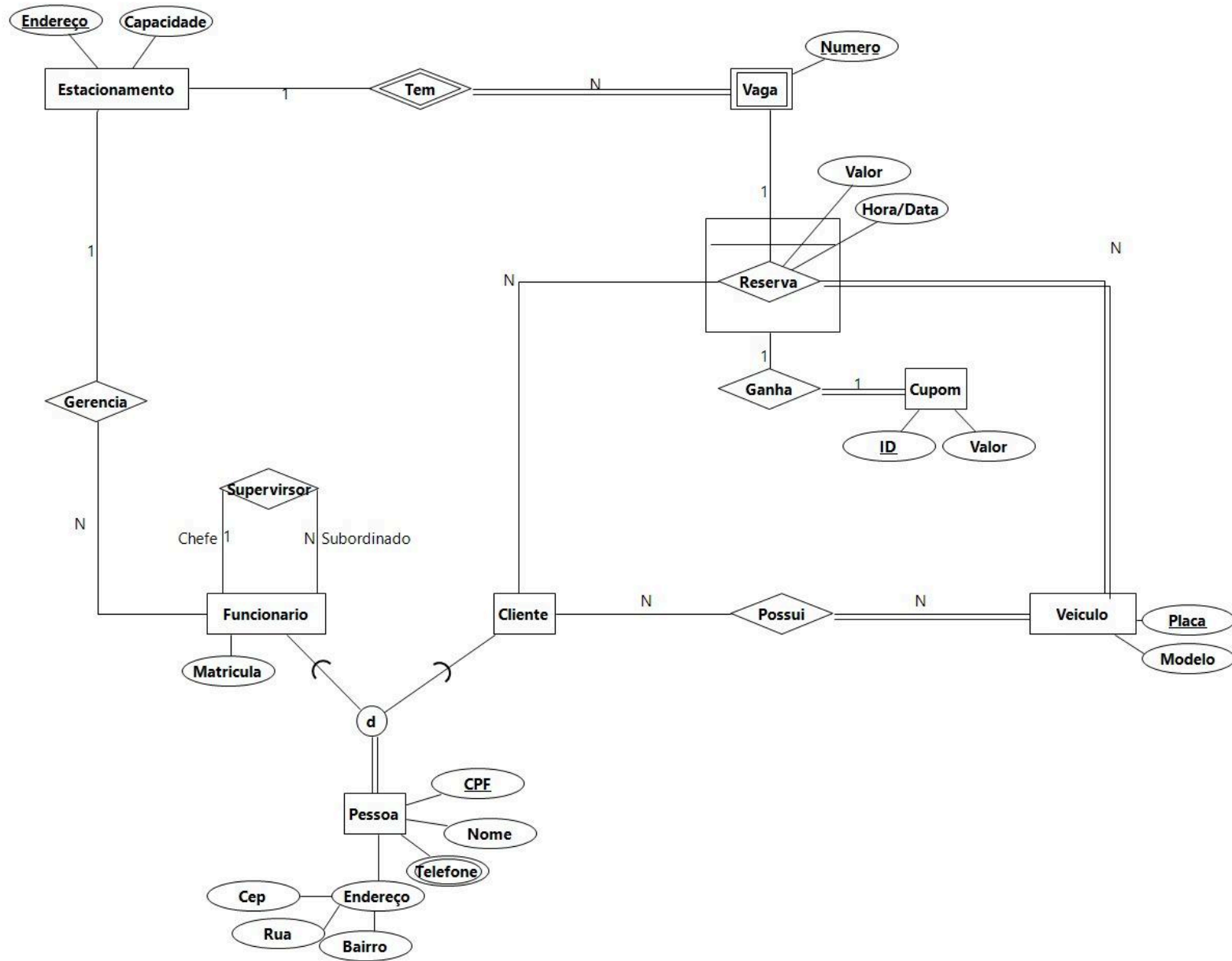
**Universidade Federal de Pernambuco  
Centro de Informática - CIn**

**Sistema de Gerenciamento de Estacionamento  
Modelo Físico de Banco de Dados**

Humberto Campos  
Carlos Kaynan  
Guilherme Caio  
Diego Rafael  
Alexandre Souza  
Eraldo Cassimiro  
Izack Angelo

Recife, 06 de Outubro de 2024

# Projeto Conceitual Atualizado



## Projeto Lógico Atualizado

Pessoa(CPF, Nome, End\_CEP, End\_Bairro, End\_Rua)

Telefone(CPF, Telefone)

CPF → Pessoa(CPF)

Cliente(CPF)

CPF → Pessoa(CPF)

Estacionamento(Endereço, Capacidade)

Funcionário(CPF, Matrícula, CPF\_Chefe, Endereço\_estacionamento!)

CPF → Pessoa(CPF)

CPF\_Chefe → Funcionário(CPF)

Endereço\_estacionamento → Estacionamento(Endereço)

Veículo(Placa, Modelo)

Vaga(Endereço\_estacionamento, Número)

Endereço\_estacionamento → Estacionamento(Endereço)

Reserva(CPF\_Cliente, Placa\_veiculo, [Endereço\_estacionamento, Número]!, Valor, Hora/Data)

CPF\_Cliente → Cliente(CPF)

Placa\_veiculo → Veículo(Placa)

Endereço\_estacionamento, Número → Vaga(Endereço\_estacionamento, número)

Cupom(ID, Valor, [CPF\_Cliente, Placa\_veiculo]!)

CPF\_Cliente, Placa\_Veiculo → Reserva(CPF\_Cliente, Placa\_Veiculo)

Possui(CPF\_cliente, Placa\_veiculo)

CPF\_cliente → Cliente(CPF)

Placa\_veiculo → Veículo(Placa)

## Projeto Físico

### Criação das tabelas

```
CREATE TABLE Pessoa (  
    CPF CHAR(11) PRIMARY KEY,  
    Nome CHAR(50) NOT NULL,  
    End_CEP NUMBER(8),  
    End_Rua CHAR(50) ,  
    End_Bairro CHAR(50)  
);
```

```
CREATE TABLE Telefone (  
    CPF CHAR(11),  
    Telefone CHAR(14),  
    PRIMARY KEY (CPF, Telefone),  
    FOREIGN KEY (CPF) REFERENCES Pessoa(CPF)  
);
```

```
CREATE TABLE Cliente (  
    CPF CHAR(11) PRIMARY KEY,  
    FOREIGN KEY (CPF) REFERENCES Pessoa(CPF)  
);
```

```
CREATE TABLE Estacionamento (  
    Endereco CHAR(30) PRIMARY KEY,  
    Capacidade INT NOT NULL  
);
```

```
CREATE TABLE Funcionario (  
    CPF CHAR(11) PRIMARY KEY,  
    Matricula INT,  
    Endereco_estacionamento CHAR(30),
```

```

CPF_CHEFE CHAR(11) NULL,
FOREIGN KEY (CPF) REFERENCES Pessoa(CPF),
        FOREIGN KEY (Endereco_estacionamento) REFERENCES
Estacionamento(Endereco),
        CONSTRAINT AUTORELACIONAMENTO FOREIGN KEY (CPF_CHEFE)
REFERENCES Funcionario(CPF)
);

```

```

CREATE TABLE Veiculo (
    Placa CHAR(7) PRIMARY KEY,
    Modelo VARCHAR(30) NOT NULL
);

```

```

CREATE TABLE Vaga (
    Endereco_estacionamento CHAR(30),
    Numero INT,
    PRIMARY KEY (Endereco_estacionamento, Numero),
        FOREIGN KEY (Endereco_estacionamento) REFERENCES
Estacionamento(Endereco)
);

```

```

CREATE TABLE Reserva (
    CPF_Cliente CHAR(11),
    Placa_veiculo CHAR(7),
    Endereco_estacionamento CHAR(30),
    Numero INT,
    Valor DECIMAL(10, 2),
    DataHora TIMESTAMP,
    PRIMARY KEY (CPF_Cliente, Placa_veiculo),
    FOREIGN KEY (CPF_Cliente) REFERENCES Cliente(CPF),
    FOREIGN KEY (Placa_veiculo) REFERENCES Veiculo(Placa),
        FOREIGN KEY (Endereco_estacionamento, Numero) REFERENCES
Vaga(Endereco_estacionamento, Numero)
);

```

```

CREATE TABLE Cupom (
    ID INT PRIMARY KEY,
    Valor DECIMAL(10, 2) NOT NULL,
    CPF_Cliente CHAR(11),
    Placa_veiculo CHAR(7),
    FOREIGN KEY (CPF_Cliente, Placa_veiculo) REFERENCES
Reserva(CPF_Cliente, Placa_veiculo)
);

```

```

CREATE TABLE Possui (
    CPF_cliente CHAR(11),
    Placa_veiculo CHAR(7),
    PRIMARY KEY (CPF_cliente, Placa_veiculo),
    FOREIGN KEY (CPF_cliente) REFERENCES Cliente(CPF),
    FOREIGN KEY (Placa_veiculo) REFERENCES Veiculo(Placa)
);

```

## Povoamento

```

INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('12345678901', 'Ana Souza', '50000000', 'Rua A, 123', 'Centro');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('23456789012', 'Bruno Lima', '51000100', 'Av. B, 456', 'Jardim das Flores');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('34567890123', 'Carlos Silva', '52000200', 'Rua C, 789', 'Centro');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('45678901234', 'Daniela Santos', '53000300', 'Rua D, 654', 'COHAB');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('56789012345', 'Eduardo Costa', '54000400', 'Av. Nove, 12', 'Jardim das Flores');

```

```
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('67890123456', 'Fernanda Oliveira', '55000500', 'Rua E, 43', 'Bairro Velho');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('78901234567', 'Gabriel Almeida', '56000600', 'Rua F, 323', 'Bairro Novo');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('89012345678', 'Heloisa Pereira', '57000700', 'Av. Sete, 72', 'Centro');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('90123456789', 'Igor Martins', '58000800', 'Rua G, 74', 'COHAB');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('01234567890', 'Juliana Fernandes', '59000900', 'Rua M, 66', 'Jardim das Flores');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('11122334455', 'Karine Souza', '60000000', 'Av. Dez, 77', 'Bairro Novo');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('22233445566', 'Lucas Carvalho', '61000100', 'Rua Paris, 99', 'COHAB');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('33344556677', 'Maria Ferreira', '63000300', 'Av. Rio, 27', 'Bairro Velho');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('44455667788', 'Natália Costa', '64000400', 'Rua Cinco, 47', 'Jardim das Flores');
INSERT INTO Pessoa (CPF, Nome, End_CEP, End_Rua, End_Bairro) VALUES
('55566778899', 'Olavo Santos', '62000200', 'Rua Luar, 555', 'Bairro Novo');
```

```
INSERT INTO Telefone (CPF, Telefone) VALUES ('12345678901', '(07)99999-1111');
INSERT INTO Telefone (CPF, Telefone) VALUES ('23456789012', '(07)99999-2222');
INSERT INTO Telefone (CPF, Telefone) VALUES ('34567890123', '(07)99999-3333');
INSERT INTO Telefone (CPF, Telefone) VALUES ('45678901234', '(07)99999-4444');
INSERT INTO Telefone (CPF, Telefone) VALUES ('56789012345', '(07)99999-5555');
INSERT INTO Telefone (CPF, Telefone) VALUES ('67890123456', '(07)99999-6666');
INSERT INTO Telefone (CPF, Telefone) VALUES ('78901234567', '(07)99999-7777');
INSERT INTO Telefone (CPF, Telefone) VALUES ('89012345678', '(07)99999-8888');
INSERT INTO Telefone (CPF, Telefone) VALUES ('90123456789', '(07)99999-9999');
INSERT INTO Telefone (CPF, Telefone) VALUES ('01234567890', '(07)88888-1111');
INSERT INTO Telefone (CPF, Telefone) VALUES ('11122334455', '(07)88888-2222');
INSERT INTO Telefone (CPF, Telefone) VALUES ('22233445566', '(07)88888-3333');
INSERT INTO Telefone (CPF, Telefone) VALUES ('33344556677', '(07)88888-4444');
```

```
INSERT INTO Telefone (CPF, Telefone) VALUES ('44455667788', '(07)88888-5555');
INSERT INTO Telefone (CPF, Telefone) VALUES ('55566778899', '(07)88888-6666');
```

```
INSERT INTO Cliente (CPF) VALUES ('12345678901');
INSERT INTO Cliente (CPF) VALUES ('23456789012');
INSERT INTO Cliente (CPF) VALUES ('34567890123');
INSERT INTO Cliente (CPF) VALUES ('45678901234');
INSERT INTO Cliente (CPF) VALUES ('56789012345');
```

```
INSERT INTO Estacionamento (Endereco, Capacidade) VALUES ('Av. Principal,
1000', 50);
INSERT INTO Estacionamento (Endereco, Capacidade) VALUES ('Rua Secundária,
200', 75);
INSERT INTO Estacionamento (Endereco, Capacidade) VALUES ('Av. Beira Mar,
300', 100);
INSERT INTO Estacionamento (Endereco, Capacidade) VALUES ('Rua do Parque,
400', 45);
INSERT INTO Estacionamento (Endereco, Capacidade) VALUES ('Av.
Independência, 500', 80);
```

```
INSERT INTO Funcionario (CPF, Matricula, Endereco_estacionamento) VALUES
('11122334455', 0006, 'Av. Principal, 1000');
INSERT INTO Funcionario (CPF, Matricula, Endereco_estacionamento) VALUES
('22233445566', 0007, 'Rua Secundária, 200');
INSERT INTO Funcionario (CPF, Matricula, Endereco_estacionamento) VALUES
('33344556677', 0008, 'Av. Beira Mar, 300');
INSERT INTO Funcionario (CPF, Matricula, Endereco_estacionamento) VALUES
('44455667788', 0009, 'Rua do Parque, 400');
INSERT INTO Funcionario (CPF, Matricula, Endereco_estacionamento) VALUES
('55566778899', 0010, 'Av. Independência, 500');
```

```
INSERT INTO Funcionario (CPF, Matricula, Endereco_estacionamento,
CPF_CHEFE) VALUES ('67890123456', 0001, 'Av. Principal, 1000', '11122334455');
```



```
INSERT INTO Funcionario (CPF, Matricula, Endereco_estacionamento, CPF_CHEFE) VALUES ('78901234567', 0002, 'Rua Secundária, 200', '22233445566');
```

```
INSERT INTO Funcionario (CPF, Matricula, Endereco_estacionamento, CPF_CHEFE) VALUES ('89012345678', 0003, 'Av. Beira Mar, 300', '33344556677');
```

```
INSERT INTO Funcionario (CPF, Matricula, Endereco_estacionamento, CPF_CHEFE) VALUES ('90123456789', 0004, 'Rua do Parque, 400', '44455667788');
```

```
INSERT INTO Funcionario (CPF, Matricula, Endereco_estacionamento, CPF_CHEFE) VALUES ('01234567890', 0005, 'Av. Independência, 500', '55566778899');
```

```
INSERT INTO Veiculo (Placa, Modelo) VALUES ('ABC1234', 'Toyota Corolla');
```

```
INSERT INTO Veiculo (Placa, Modelo) VALUES ('XYZ5678', 'Honda Civic');
```

```
INSERT INTO Veiculo (Placa, Modelo) VALUES ('DEF4321', 'Ford Focus');
```

```
INSERT INTO Veiculo (Placa, Modelo) VALUES ('GHI8765', 'Chevrolet Onix');
```

```
INSERT INTO Veiculo (Placa, Modelo) VALUES ('JKL9876', 'VW Gol');
```

```
INSERT INTO Vaga (Endereco_estacionamento, Numero) VALUES ('Av. Principal, 1000', 7);
```

```
INSERT INTO Vaga (Endereco_estacionamento, Numero) VALUES ('Av. Beira Mar, 300', 8);
```

```
INSERT INTO Vaga (Endereco_estacionamento, Numero) VALUES ('Rua do Parque, 400', 5);
```

```
INSERT INTO Vaga (Endereco_estacionamento, Numero) VALUES ('Av. Independência, 500', 6);
```

```
INSERT INTO Vaga (Endereco_estacionamento, Numero) VALUES ('Rua Secundária, 200', 9);
```

```
INSERT INTO Reserva (CPF_Cliente, Placa_veiculo, Endereco_estacionamento, Numero, Valor, DataHora)
```

```
VALUES ('12345678901', 'ABC1234', 'Av. Principal, 1000', 7, 30.00, TO_DATE('2024-08-01 08:00:00', 'YYYY-MM-DD HH24:MI:SS'));
```

```
INSERT INTO Reserva (CPF_Cliente, Placa_veiculo, Endereco_estacionamento,
Numero, Valor, DataHora)
```

```
VALUES ('23456789012', 'XYZ5678', 'Rua Secundária, 200', 9, 25.00,
TO_DATE('2024-08-03 10:00:00', 'YYYY-MM-DD HH24:MI:SS'));
```

```
INSERT INTO Reserva (CPF_Cliente, Placa_veiculo, Endereco_estacionamento,
Numero, Valor, DataHora)
```

```
VALUES ('34567890123', 'DEF4321', 'Av. Beira Mar, 300', 8, 60.00,
TO_DATE('2024-08-05 12:00:00', 'YYYY-MM-DD HH24:MI:SS'));
```

```
INSERT INTO Reserva (CPF_Cliente, Placa_veiculo, Endereco_estacionamento,
Numero, Valor, DataHora)
```

```
VALUES ('45678901234', 'GHI8765', 'Rua do Parque, 400', 5, 30.00,
TO_DATE('2024-08-07 10:00:00', 'YYYY-MM-DD HH24:MI:SS'));
```

```
INSERT INTO Reserva (CPF_Cliente, Placa_veiculo, Endereco_estacionamento,
Numero, Valor, DataHora)
```

```
VALUES ('56789012345', 'JKL9876', 'Av. Independência, 500', 6, 40.00,
TO_DATE('2024-08-10 12:00:00', 'YYYY-MM-DD HH24:MI:SS'));
```

```
INSERT INTO Cupom (ID, Valor, CPF_Cliente, Placa_veiculo) VALUES (001, 25.00,
'12345678901', 'ABC1234');
```

```
INSERT INTO Cupom (ID, Valor, CPF_Cliente, Placa_veiculo) VALUES (002, 30.00,
'23456789012', 'XYZ5678');
```

```
INSERT INTO Cupom (ID, Valor, CPF_Cliente, Placa_veiculo) VALUES (003, 15.00,
'34567890123', 'DEF4321');
```

```
INSERT INTO Cupom (ID, Valor, CPF_Cliente, Placa_veiculo) VALUES (004, 10.00,
'45678901234', 'GHI8765');
```

```
INSERT INTO Cupom (ID, Valor, CPF_Cliente, Placa_veiculo) VALUES (005, 35.00,
'56789012345', 'JKL9876');
```

```
INSERT INTO Possui (CPF_cliente, Placa_veiculo) VALUES ('12345678901',
'ABC1234');
```

```
INSERT INTO Possui (CPF_cliente, Placa_veiculo) VALUES ('23456789012',
'XYZ5678');
```

```
INSERT INTO Possui (CPF_cliente, Placa_veiculo) VALUES ('34567890123',
'DEF4321');
```

```
INSERT INTO Possui (CPF_cliente, Placa_veiculo) VALUES ('45678901234',  
'GHI8765');
```

```
INSERT INTO Possui (CPF_cliente, Placa_veiculo) VALUES ('56789012345',  
'JKL9876');
```

## Consultas

1. Uma consulta que possua as cláusulas WHERE e ORDER BY

A consulta retorna uma lista de clientes (incluindo seus nomes, ruas, bairro e telefones) que não residem no bairro "COHAB", ordenados pelo nome.

```
SELECT
    p.Nome,
    p.End_Rua,
    p.End_Bairro,
    t.Telefone
FROM
    Cliente c
JOIN
    Pessoa p ON c.CPF = p.CPF
JOIN
    Telefone t ON p.CPF = t.CPF
WHERE
    p.End_Bairro <> 'COHAB'
ORDER BY
    p.Nome;
```

### Filtrar Resultados:

- WHERE p.End\_Bairro <> 'COHAB': Exclui todos os registros em que o bairro é "COHAB", ou seja, apenas clientes que moram em outros bairros serão incluídos nos resultados.

### Ordenar Resultados:

- ORDER BY p.Nome: Organiza os resultados em ordem alfabética pelo nome da pessoa.

2. Uma consulta que possua as cláusulas COUNT/SUM e GROUP BY:

A consulta abaixo conta o número de vendas e calcula o total de receita por produto:

```
SELECT
    r.CPF_Cliente,
    COUNT(r.Placa_veiculo) AS total_reservas,
    SUM(r.Valor) AS valor_total_reservas
FROM
    Reserva r
GROUP BY
    r.CPF_Cliente;
```

- COUNT(id\_venda): Conta o número total de vendas para cada produto.
- SUM(quantidade \* preco): Calcula a receita total gerada por cada produto.
- GROUP BY id\_produto: Agrupa os resultados por id\_produto, permitindo que as funções de agregação sejam aplicadas a cada grupo.

3. Uma consulta que possua as cláusulas HAVING e LIMIT:

Vamos contar quantos veículos cada cliente possui na tabela Possui e filtrar aqueles que têm pelo menos um veículo, limitando o resultado a 3 clientes.

```
SELECT
    cl.CPF,
    COUNT(ph.Placa_veiculo) AS total_veiculos
FROM
    Cliente cl
LEFT JOIN
    Possui ph ON cl.CPF = ph.CPF_cliente
GROUP BY
    cl.CPF
HAVING
    COUNT(ph.Placa_veiculo) > 0
FETCH FIRST 3 ROWS ONLY;
```

- HAVING COUNT(ph.Placa\_veiculo) > 1: Filtra os clientes que possuem mais de um veículo.
- FETCH FIRST 5 ROWS ONLY: Limita os resultados a 5 clientes.

#### 4. Uma consulta que possua um JOIN:

A consulta retornará uma lista de nomes de clientes e seus respectivos telefones, ordenados pelo nome.

```
SELECT
    p.Nome,
    t.Telefone
FROM
    Cliente c
JOIN
    Pessoa p ON c.CPF = p.CPF
JOIN
    Telefone t ON p.CPF = t.CPF
ORDER BY
    p.Nome;
```

#### **Junções:**

- JOIN Pessoa p ON c.CPF = p.CPF: Faz uma junção entre a tabela Cliente e a tabela Pessoa usando o CPF como chave.
- JOIN Telefone t ON p.CPF = t.CPF: Faz uma junção adicional entre a tabela Pessoa e a tabela Telefone para obter os números de telefone.

5. Uma consulta que possua uma subconsulta:

A consulta retornará uma lista de nomes de clientes que têm pelo menos um telefone registrado, ordenados pelo nome.

```
SELECT
    p.Nome
FROM
    Cliente c
JOIN
    Pessoa p ON c.CPF = p.CPF
WHERE
    c.CPF IN (SELECT t.CPF FROM Telefone t)
ORDER BY
    p.Nome;
```

**Subconsulta:**

(SELECT t.CPF FROM Telefone t): Está subconsulta retorna todos os CPFs que têm pelo menos um telefone registrado na tabela 'Telefone'.



## **Apresentação**

Link da Apresentação: [Modelo Fisco](#)

Link GitHub: [ProjetoBD](#)