1. Não está na 1 Forma , Pois Tem dados Repetidos

A **Tabela Contas** não está na **Primeira Forma Normal (1FN)** porque apresenta **valores multivalorados** dentro de uma única linha. O problema ocorre porque há múltiplas colunas para diferentes tipos de contas (ContaOrdem, ContaPoupança, ContaFuturoMais), em vez de armazenar as contas em linhas separadas.

### **Normalização para 1FN**

Para normalizar a tabela e garantir que ela esteja na **1FN**, devemos transformá-la em uma estrutura onde cada atributo contenha **apenas um valor atômico** por registro. Isso pode ser feito criando uma nova tabela que separa as contas de acordo com seu tipo.

Tabela Cliente

| **ID** | **Nome** |
| --- | --- |

| **3256** | **Marco Fonseca** |
| --- | --- |
| **2345** | **Raquel Castro** |
| **9514** | **Rui Pena** |
| **8524** | **Vitor Garcia** |
| **3698** | **Manuela Gouveia** |

**Tabela Contas**

| **ID Cliente** | **TipoConta** | **NúmeroConta** |
| --- | --- | --- |

| **3256** | **ContaOrdem** | **54165452** |
| --- | --- | --- |
| **3256** | **ContaPoupança** | **36987412** |
| **3256** | **ContaFuturoMais** | **85693125** |
| **2345** | **ContaOrdem** | **95184673** |
| **9514** | **ContaOrdem** | **73271001** |
| **9514** | **ContaPoupança** | **52145652** |
| **9514** | **ContaFuturoMais** | **84951673** |
| **8524** | **ContaOrdem** | **65278742** |
| **8524** | **ContaFuturoMais** | **56321479** |
| **3698** | **ContaPoupança** | **84095632** |

2. Está na 1 Forma , pois não tem campos repetidos.

3. Não Está na 1 Forma porque repete atributos.

A tabela **Produtos** não está na **Primeira Forma Normal (1FN)**, pois a coluna **Cor** contém **valores múltiplos** (exemplo: "vermelho, verde" no CódigoProduto 1 e "amarelo, azul" no CódigoProduto 4). Para estar na 1FN, cada campo deve conter **apenas um valor atômico**.

### **Normalização para 1FN**

Para normalizar a tabela, devemos **separar as cores em linhas individuais**, mantendo a relação com o CódigoProduto e o Preço:

Tabela Produtos

| **CodigoProduto** | **Preço** |
| --- | --- |

| **1** | **15,99** |
| --- | --- |
| **2** | **23,99** |
| **3** | **17,50** |
| **4** | **9,99** |
| **5** | **29,99** |

**Tabela Cores**

| **CodigoCor** | **Cor** |
| --- | --- |

| **1** | **vermelho** |
| --- | --- |
| **2** | **verde** |
| **3** | **amarelo** |
| **4** | **azul** |

**Tabela Produto\_Cor**

| **CodigoProduto** | **CodigoCor** |
| --- | --- |

| **1** | **1** |
| --- | --- |
| **1** | **2** |
| **2** | **3** |
| **3** | **2** |
| **4** | **3** |
| **4** | **4** |
| **5** | **1** |

4. Não está na 1 Forma

A tabela **ClientesAlugueres** não está na **Primeira Forma Normal (1FN)** porque contém **valores multivalorados** em algumas colunas. Especificamente:

* **ClienteID, Nome Cliente** e **PropriedadeID** não são atômicos, pois um cliente pode alugar várias propriedades, o que gera múltiplos valores na mesma linha.
* O mesmo acontece com **ProprietárioID e Nome Proprietário**, pois uma propriedade pode ter um proprietário diferente.

### **Normalização para 1FN:**

Para garantir a **1FN**, devemos eliminar os grupos repetitivos criando tabelas separadas. Podemos dividir a tabela em:

1. **Tabela Clientes**
   * ClienteID (PK)
   * NomeCliente
2. **Tabela Alugueres** (TABELA RELACIONAL ENTRE CLIENTES E PROPRIEDADES)
   * AluguerID (PK)
   * ClienteID (FK para Clientes)
   * PropriedadeID (FK para Propriedades)
   * InícioAluguer
   * FimAluguer

Isso elimina **valores multivalorados** e garante que cada coluna tenha apenas um único valor escalar, atendendo à **1FN**.

**Tabela Clientes**

| **ClienteID** | **NomeCliente** |
| --- | --- |

| **C05** | **Carla Santos** |
| --- | --- |
| **C019** | **Filipe Martins** |

**Tabela Alugueres**

| **ClienteID** | **PropriedadeID** | **Morada** | **Início**  **Aluguer** | **Fim**  **Aluguer** | **ProprietárioID** | **Nome**  **Proprietário** |
| --- | --- | --- | --- | --- | --- | --- |

| **C05** | **CBR004** | **Rua Carlos Seixas, 17** | **01/07/2000** | **31/08/2001** | **P009** | **Henrique Monteiro** |
| --- | --- | --- | --- | --- | --- | --- |
| **C05** | **CBR006** | **Alameda Marquesa de Pomares, 98, 1º D** | **01/09/2001** |  | **P001** | **Carlos Gonçalves** |
| **C019** | **CBR018** | **Rua Bartolomeu Dias, 8** | **01/09/2005** | **10/10/2005** | **P021** | **Mário Agulha** |
| **C019** | **CBR107** | **Rua do Brasil, 10, 8C** | **20/12/2005** | **31/12/2012** | **P020** | **Bárbara Silva** |
| **C019** | **CBR032** | **Rua Augusto de Matos, 15** | **01/01/2013** | **01/08/2013** | **P300** | **Rui Carneiro** |

Grupo 2

1.

### **1. Verificação da 1ª Forma Normal (1FN)**

Uma tabela está na 1FN se:

* Todos os atributos contêm valores atômicos (sem listas ou conjuntos de valores repetidos).
* Cada coluna contém um único tipo de dado.
* Existe uma chave primária identificando univocamente cada registo.

A tabela fornecida parece estar na 1FN, pois não há repetições dentro de células e cada campo tem um único valor.

### **2. Verificação da 2ª Forma Normal (2FN)**

Uma tabela está na 2FN se:

* Já está na 1FN.
* Todos os atributos não-chave dependem funcionalmente da chave primária completa, e não apenas de uma parte dela.

A chave primária aqui parece ser a combinação (EstudanteID, ProfessorID), pois um aluno pode ter várias classificações atribuídas por diferentes professores.

Agora, analisemos as dependências funcionais:

* NomeEstudante depende apenas de EstudanteID.
* NomeProfessor depende apenas de ProfessorID.
* Classificação depende de (EstudanteID, ProfessorID).

O problema é que NomeEstudante e NomeProfessor não dependem da chave primária completa, mas sim de partes dela. Isso significa que a tabela **não está na 2FN**.

### **3. Normalização para 2FN**

Para normalizar a tabela, devemos separar os dados em tabelas menores, garantindo que cada atributo dependa completamente da sua chave primária.

Podemos criar três tabelas:

Tabela Estudantes

| **EstudanteID** | **NomeEstudante** |
| --- | --- |

| **1** | **Fernando Miranda** |
| --- | --- |
| **2** | **Luísa Franco** |

**Tabela Professores**

| **ProfessorID** | **NomeProfessor** |
| --- | --- |

| **5** | **Miguel Alexandre** |
| --- | --- |
| **21** | **Paulo Ramos** |

**Tabela Classificações**

| **EstudanteID** | **ProfessorID** | **Classificação** |
| --- | --- | --- |

| **1** | **5** | **13** |
| --- | --- | --- |
| **2** | **21** | **8** |

1. **Não esta na 2 Forma**

### **1. Verificação da 1ª Forma Normal (1FN)**

**Para estar na 1FN, a tabela deve:**

* **Não ter atributos multivalorados ou repetidos.**
* **Ter uma chave primária definida.**

**A tabela tem a chave primária composta por (PeçaID, ArmazémID), pois uma peça pode estar armazenada em vários armazéns, e cada armazém pode conter várias peças.**

**Todos os atributos possuem valores atômicos, então a tabela está na 1FN.**

### **2. Verificação da 2ª Forma Normal (2FN)**

**Para estar na 2FN:**

* **Deve estar na 1FN.**
* **Todos os atributos não-chave devem depender da chave primária completa, não apenas de uma parte dela.**

#### **Analisemos as dependências funcionais:**

* **Quantidade depende de (PeçaID, ArmazémID), pois a quantidade de uma peça está associada a um armazém específico. ✅ (Depende da chave completa)**
* **EndereçoArmazém depende apenas de ArmazémID, pois o endereço pertence ao armazém e não a uma combinação de peça e armazém. ❌ (Depende apenas de parte da chave primária)**

**Como EndereçoArmazém depende apenas de ArmazémID, a tabela não está na 2FN.**

### **3. Normalização para 2FN**

**Para corrigir o problema, devemos separar os dados em duas tabelas:**

**Tabela Peças\_Armazéns**

| **PeçaID** | **ArmazémID** | **Quantidade** |
| --- | --- | --- |

| **101** | **5** | **50** |
| --- | --- | --- |
| **202** | **3** | **120** |

**Tabela Armazéns**

| **ArmazémID** | **EndereçoArmazém** |
| --- | --- |

| **5** | **Rua das Flores, 123** |
| --- | --- |
| **3** | **Av. Central, 45** |

1. **Nao está na 2 Forma**

### **1. Verificação da 1ª Forma Normal (1FN)**

**Para estar na 1FN, a tabela deve:**

* **Não ter atributos multivalorados ou repetidos.**
* **Ter uma chave primária bem definida.**

**A chave primária aqui é (Cliente\_ID, Loja\_ID), pois um cliente pode comprar em várias lojas e cada loja pode atender vários clientes.**

**A tabela não contém valores multivalorados ou repetidos, então está na 1FN.**

### **2. Verificação da 2ª Forma Normal (2FN)**

**Para estar na 2FN:**

* **Deve estar na 1FN.**
* **Todos os atributos não-chave devem depender da chave primária completa, e não apenas de uma parte dela.**

#### **Analisemos as dependências funcionais:**

* **LocalLoja depende apenas de Loja\_ID, pois a localização da loja não depende do cliente, apenas da própria loja. ❌ (Depende de parte da chave primária e não da chave composta completa).**

**Como LocalLoja depende apenas de Loja\_ID, a tabela não está na 2FN.**

### **3. Normalização para 2FN**

**Para corrigir o problema, devemos separar os dados em duas tabelas:**

**Tabela Vendas**

| **Cliente\_ID** | **Loja\_ID** |
| --- | --- |

| **1** | **1** |
| --- | --- |
| **1** | **3** |
| **2** | **1** |
| **3** | **2** |
| **4** | **3** |

**Tabela Lojas**

| **Loja\_ID** | **LocalLoja** |
| --- | --- |

| **1** | **Los Angeles** |
| --- | --- |
| **2** | **Pequim** |
| **3** | **São Francisco** |

**Grupo 3**

**1.**

**A tabela apresentada não está na 3ª Forma Normal (3FN), pois há uma dependência transitiva entre os atributos não-chave. O CustoUnitário depende diretamente do Combustível, e não apenas da chave primária composta (TipoVeículo, Combustível).**

**Tabela Abastecimento**

| **TipoVeículo** | **Combustível** |
| --- | --- |

| **Pesado** | **GPL** |
| --- | --- |
| **Ligeiro passageiros** | **Gasolina 95** |
| **Moto** | **Gasolina 98** |
| **Ligeiro mercadorias** | **Gasóleo** |

**Tabela Combustível**

| **Combustível** | **CustoUnitário** |
| --- | --- |

| **GPL** | **0,8** |
| --- | --- |
| **Gasolina 95** | **1,3** |
| **Gasolina 98** | **1,6** |
| **Gasóleo** | **1,1** |

**2.**

### **Exercício 2: Verificação e Normalização da Tabela Requisições**

#### **1. Verificação da 1ª Forma Normal (1FN)**

**Para estar na 1FN, a tabela não pode conter atributos multivalorados ou repetidos e deve ter uma chave primária bem definida.**

**Tabela inicial:  
Requisições (Livro\_ID, Título, Autor, Gênero, Leitor\_ID, Nome, Morada, Telefone)**

**Problemas:**

* **Nome, Morada e Telefone dependem apenas de Leitor\_ID e não de Livro\_ID.**
* **Cada leitor pode ter apenas um telefone, mas se um leitor requisitar vários livros, seu nome, morada e telefone podem se repetir.**

**Como solução, dividimos a tabela em:**

1. **Tabela Livros:**
   * **(Livro\_ID, Título, Autor, Gênero)**
2. **Tabela Leitores:**
   * **(Leitor\_ID, Nome, Morada, Telefone)**
3. **Tabela Requisições:**
   * **(Livro\_ID, Leitor\_ID) (chave primária composta)**

**Agora a tabela está na 1FN.**

#### **2. Verificação da 2ª Forma Normal (2FN)**

**Para estar na 2FN, a tabela precisa:**

* **Estar na 1FN.**
* **Todos os atributos não-chave devem depender da chave primária completa.**

**A nova estrutura já atende a isso, pois agora os atributos de livros dependem apenas de Livro\_ID, os atributos de leitores dependem apenas de Leitor\_ID, e a relação de requisições é representada pela tabela intermediária.**

**Agora a tabela está na 2FN.**

#### **3. Verificação da 3ª Forma Normal (3FN)**

**Para estar na 3FN:**

* **Deve estar na 2FN.**
* **Não pode haver dependências transitivas (um atributo não-chave não pode depender de outro atributo não-chave).**

**A estrutura já atende à 3FN, pois todas as dependências são diretas da chave primária.**

**Resultado final:  
A tabela foi normalizada para 3FN com as seguintes tabelas:**

1. **Livros (Livro\_ID, Título, Autor, Gênero)**
2. **Leitores (Leitor\_ID, Nome, Morada, Telefone)**
3. **Requisições (Livro\_ID, Leitor\_ID)**

**Tabela Livro**

| **ID** | **Titulo** | **Autor** | **Genero** |
| --- | --- | --- | --- |
| **-** | **-** | **-** | **-** |

**Tabela Leitor**

| **ID** | **Nome** | **Morada** | **Telefone** |
| --- | --- | --- | --- |
| **-** | **-** | **-** | **-** |

**Tabela Requisições**

| **ID** | **ID Livro** | **ID Leitor** |
| --- | --- | --- |
| **-** | **-** | **-** |

**3.**

### **Exercício 3: Normalização da Tabela Funcionários**

#### **1. Verificação da 1ª Forma Normal (1FN)**

**A tabela inicial:  
Funcionários (Funcionário\_ID, Nome, Categoria, Salário, Departamento\_ID, DesignaçãoDepartamento)**

**Problemas:**

* **Não há atributos multivalorados ou repetidos.**
* **A chave primária é Funcionário\_ID, então todos os outros atributos devem depender diretamente dela.**

**A tabela já está na 1FN.**

#### **2. Verificação da 2ª Forma Normal (2FN)**

**Para estar na 2FN:**

* **Deve estar na 1FN.**
* **Todos os atributos não-chave devem depender da chave primária completa.**

**Problema encontrado:**

* **DesignaçãoDepartamento depende apenas de Departamento\_ID, não de Funcionário\_ID.**
* **Salário depende apenas de Categoria, não de Funcionário\_ID.**

**Solução:  
Criamos duas tabelas separadas para essas dependências:**

1. **Tabela Funcionários:**
   * **(Funcionário\_ID, Nome, Categoria, Departamento\_ID)**
2. **Tabela Categorias:**
   * **(Categoria, Salário)**
3. **Tabela Departamentos:**
   * **(Departamento\_ID, DesignaçãoDepartamento)**

**Agora a estrutura está na 2FN.**

#### **3. Verificação da 3ª Forma Normal (3FN)**

**Para estar na 3FN:**

* **Deve estar na 2FN.**
* **Não pode haver dependências transitivas.**

**A nova estrutura já atende à 3FN, pois:**

* **Categoria → Salário (mantido na tabela Categorias).**
* **Departamento\_ID → DesignaçãoDepartamento (mantido na tabela Departamentos).**

**Resultado final:  
A tabela foi normalizada para 3FN com as seguintes tabelas:**

1. **Funcionários (Funcionário\_ID, Nome, Categoria, Departamento\_ID)**
2. **Departamentos(id, Salário)**
3. **Departamentos (Departamento\_ID, DesignaçãoDepartamento)**

**Tabela Funcionários**

| **id** | **nome** | **iddepartamento** | **idcategoria** |
| --- | --- | --- | --- |
| **-** | **-** | **-** | **-** |

**Tabela Departamento**

| **id** | **salário** |
| --- | --- |
| **A** | **4000,00€** |
| **B** | **3000,00€** |
| **C** | **2000,00€** |
| **D** | **1000,00€** |