**NORMALIZAÇÃO DE DADOS**

A normalização tem o objetivo de evitar anomalias nas tabelas. Ela é dividida em três formas normais.

Na Primeira Forma Normal (1FN), não existem grupos de atributos repetidos e as suas relações não possuem atributos multivalorados ou relações aninhadas.

 Para que uma tabela esteja na 1FN, cada linha deve representar um registro e cada célula deve conter um valor único.

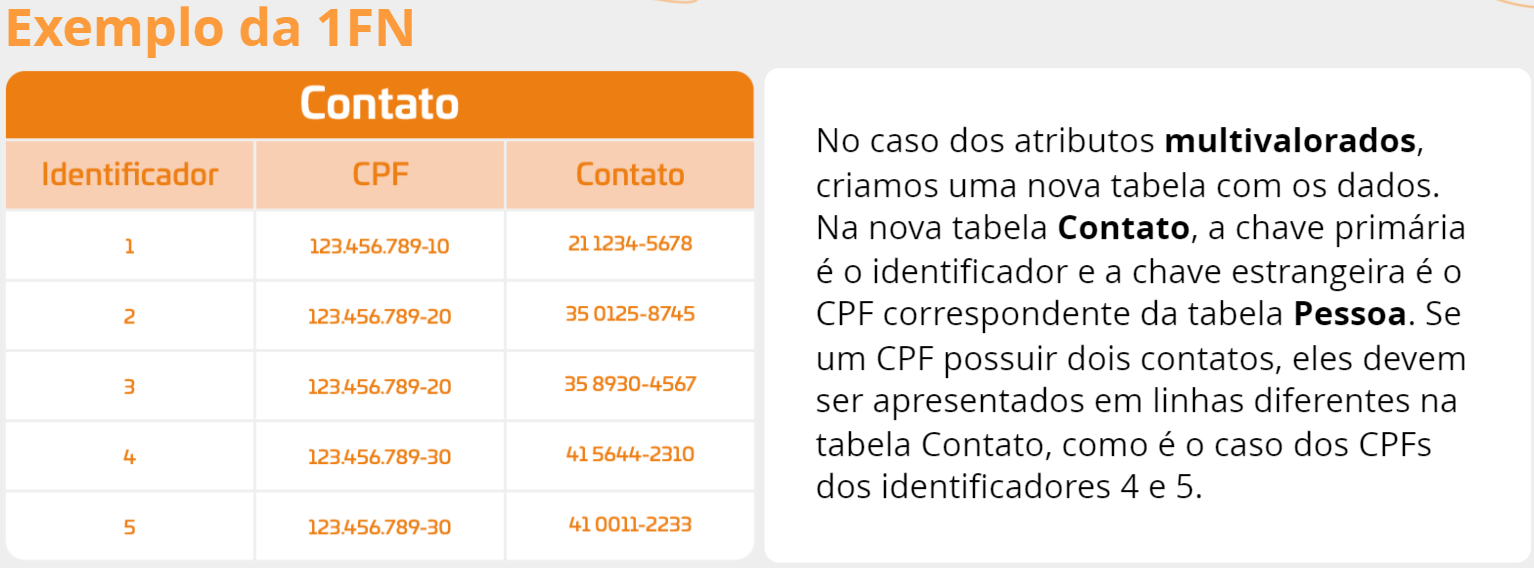
1FN – Primeira Forma Normal

Uma tabela 1FN deve estar de acordo com as seguintes informações:

* Os dados devem estar em valores atômicos, ou seja, indivisíveis;
* Ter apenas um dado por coluna;
* Deve existir pelo menos uma chave primária;
* Se existirem atributos multivalorados, uma nova tabela será criada;
* Se existirem atributos compostos, eles serão desmembrados em novas colunas de dados.







**2FN – Segunda Forma Normal**

A [Segunda Forma Normal](https://www.youtube.com/watch?v=mHoZZUYVFzk) (2FN) tem que estar, obrigatoriamente, na 1FN. Sendo assim, não deve ter atributos compostos ou multivalorados e nem dependências parciais.

No caso das [chaves primárias compostas](https://www.youtube.com/watch?v=ao4S-ZxAZbM), a 2FN exige que todas as relações tenham dependência total. Ou seja, uma chave primária composta determina funcionalmente os atributos, não podendo existir um atributo que depende apenas de uma parte dessa chave.

Portanto, na 2FN, os dados:

* Já estão na 1FN;
* A chave primária precisa ser simples.



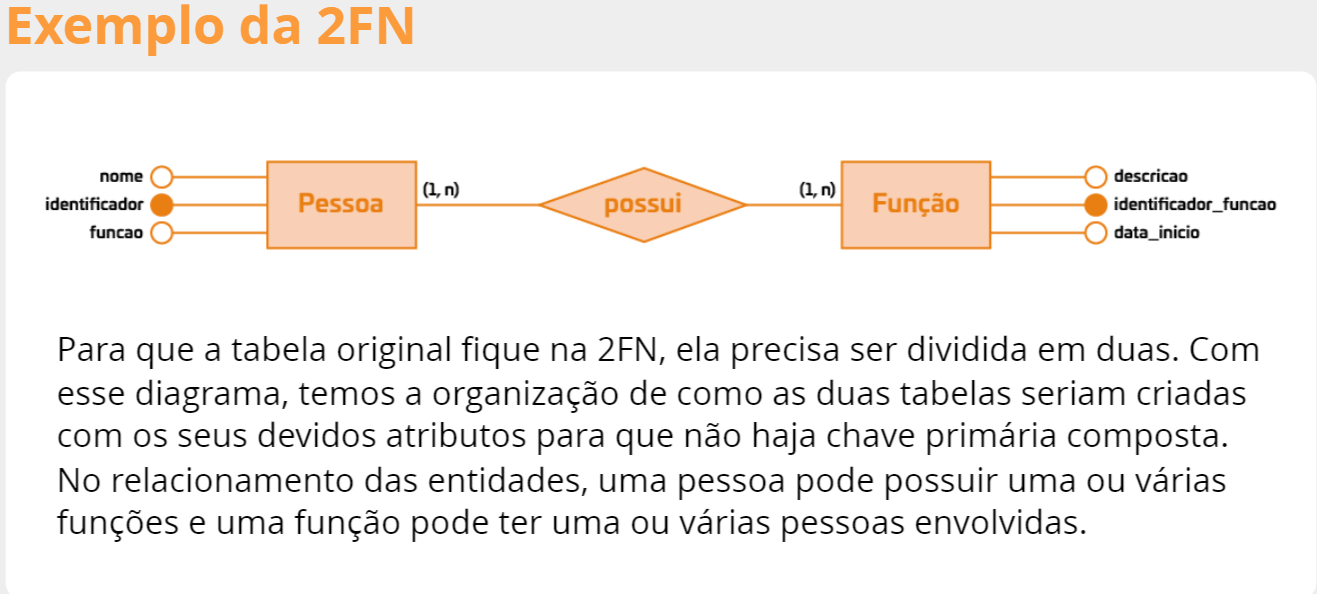
Essa tabela está na 1FN, não possuindo dados multivalorados e nem atributos compostos. Portanto, como não está na 2FN, ela precisa de ajustes.

Nesta tabela, temos chaves primárias compostas, o **IDENTIFICADOR**e o **IDENTIFICADOR\_FUNÇÃO**. Com o atributo NOME, conseguimos saber qual é o nome da pessoa, sendo um atributo que depende parcialmente da chave primária.

Também conseguimos saber qual é a função da pessoa pelo número do identificador e a descrição pelo identificador da função. Com isso, podemos ter duas tabelas ao invés de apenas uma.

Para transformar essa tabela na 2FN, será necessário desmembrar alguns dados e construir outra tabela. Assim, nenhuma coluna depende parcialmente de outra.

Para construir a outra tabela, é interessante montar o modelo conceitual para entendermos quais atributos ficarão em cada uma das tabelas.



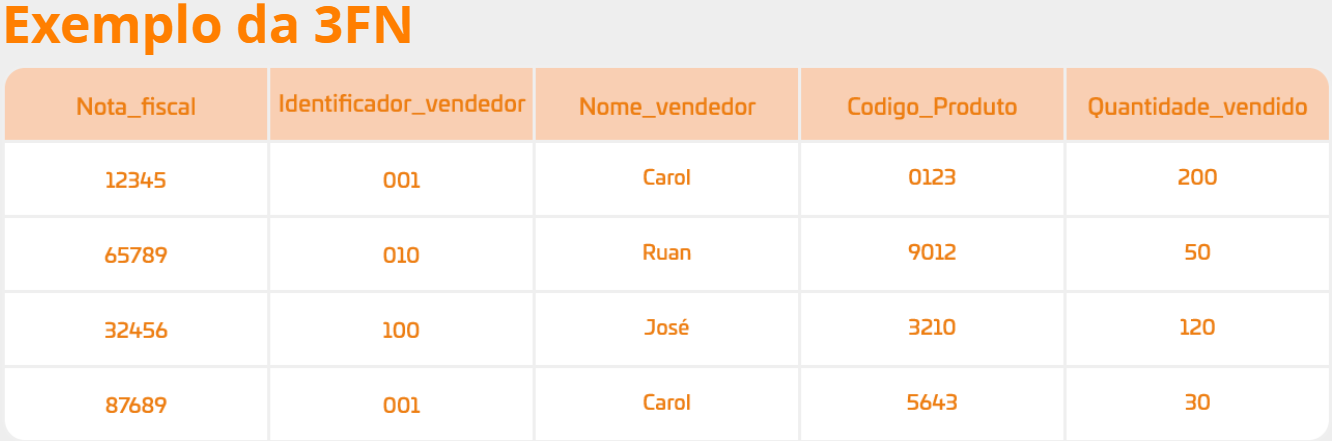
**3FN – Terceira Forma Normal**

A [Terceira Forma Normal](https://www.youtube.com/watch?v=EZvrGEpyNbs) (3FN) tem que estar, obrigatoriamente, na 2FN e possuir dependência exclusiva da chave primária da tabela.

Todos os seus atributos  devem ser funcionalmente independentes.

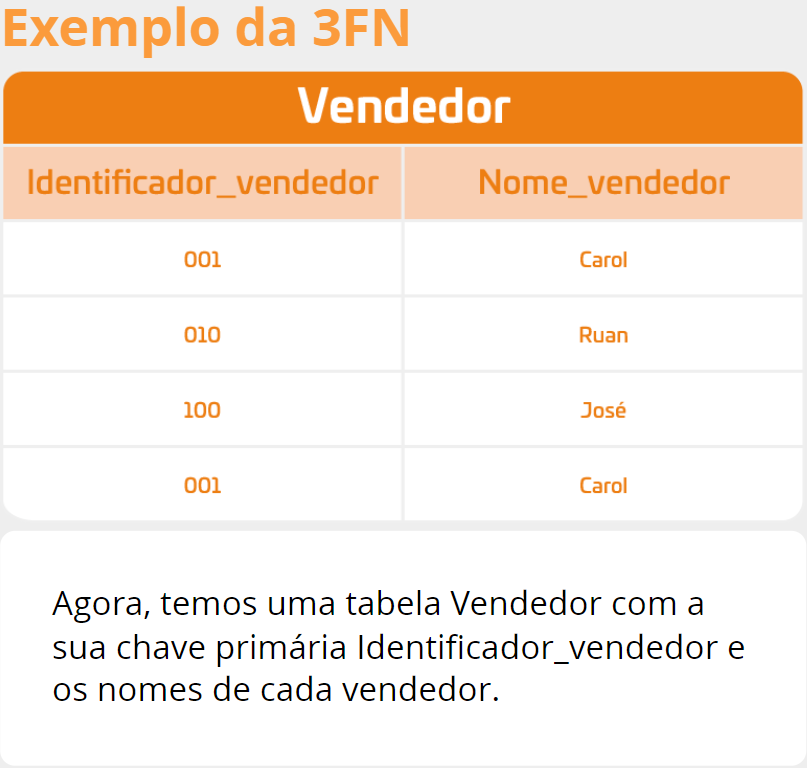
Na 3FN, os dados:

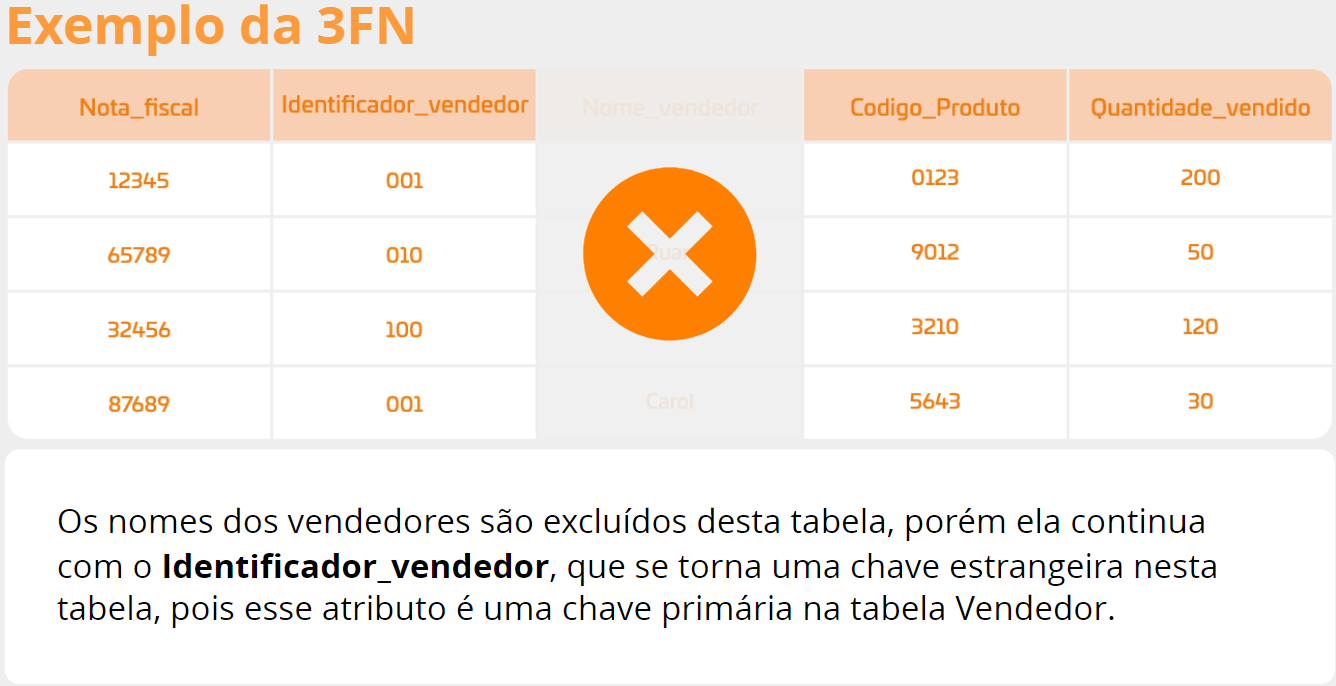
* Devem estar na 2FN;
* Os atributos não podem depender de outros que não são chaves.



Essa tabela não está na 3FN, pois os seus atributos não são todos dependentes. A **Nota\_fiscal** é a chave primária e sabemos o **Nome\_vendedor** através do **Identificador\_vendedor**. Com isso, temos uma dependência.

Para resolver isso, precisamos criar uma nova tabela com esses atributos que são dependentes.





**Forma Normal de Boyce-Codd (FNBC)**

Cada forma normal é estritamente mais forte do que a anterior, então:

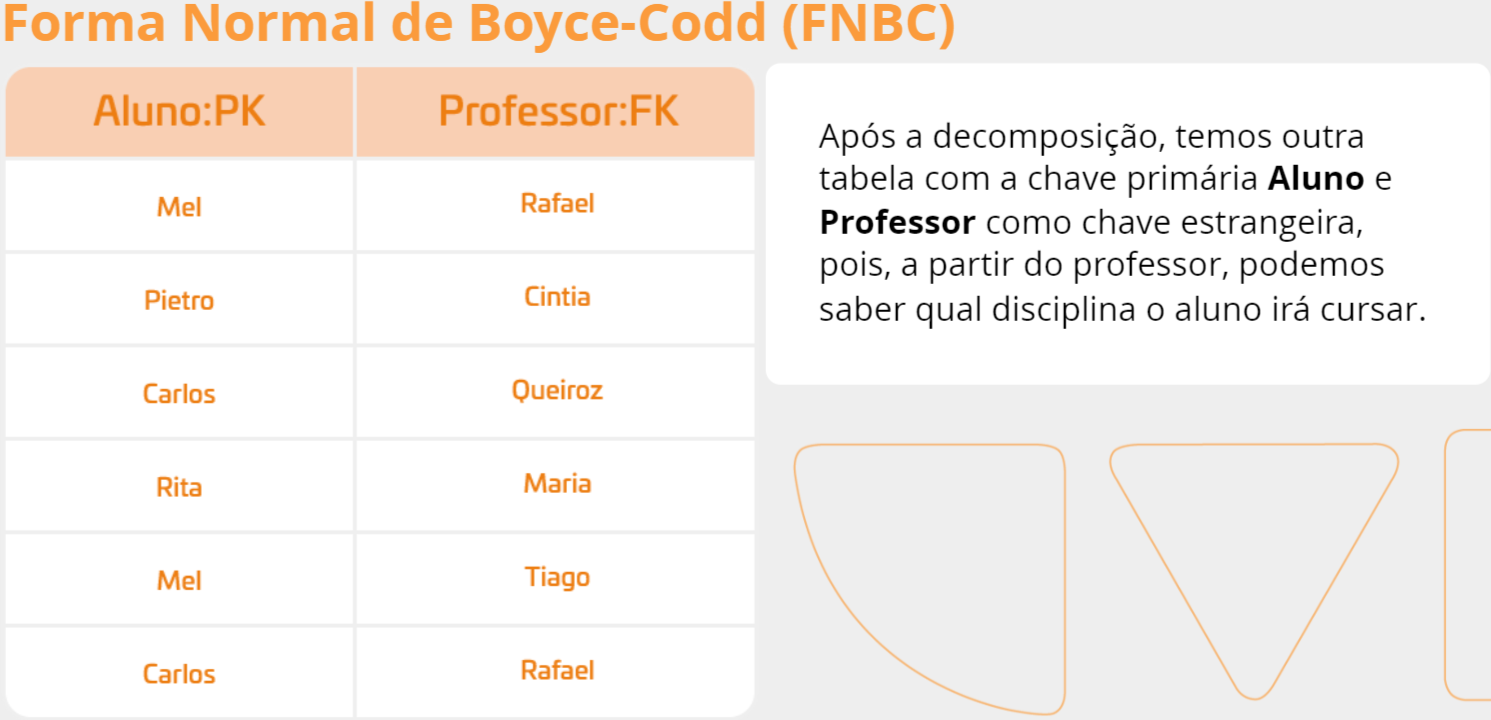
* Toda relação da 2FN encontra-se na 1FN; e
* Toda relação na 3FN encontra-se na 2FN.

O mesmo acontece  com a Forma Normal de Boyce-Codd (FNBC). Toda relação FNBC está na 3FN.

Ela é uma melhoria da 3FN utilizada em casos onde a tabela apresenta mais de uma chave candidata.  
A FNBC serve para separar, em novas tabelas, valores que ainda estejam redundantes do seu banco de dados.







Vimos no exemplo anterior como transformar uma tabela para a FNBC, utilizando a função de decompor os atributos, construindo novas tabelas para eliminar as dependências.