

BANCO DE DADOS 1 – Prof. Giovani
Exercício de Fixação 08 – EF08 – RESOLVIDO

1. A tabela a seguir não está na primeira forma normal (1FN).

CLIENTE (cpf, nome, telefones, endereco)

CLIENTE			
CPF	NOME	TELEFONES	ENDERECO
1	Tilo Gerhold	43 3524-5987 43 3524-4176 43 99185-4876	Rua das Couves, 154, Casa, Jd. Bandeirantes, 86305-157
2	Lisa Reimann	41 4398-8743 41 4398-0948 41 99723-7219	Rua Butiá, 189, Ed. Jujuba - Apto 29, Centro, 86304-654
3	Gerhard Hüttig	11 3524-1584 11 99745-1587 11 99185-4876	Av. Jacarandá, 158, Casa, São Sebastião, 86352-147

Os registros contidos na tabela têm o propósito de auxiliar na visualização das redundâncias e/ou dependências presentes.

Realize os procedimentos necessários para levar a tabela não normalizada à sua 1FN, descrevendo as etapas realizadas.

Considere `nome`¹ e `endereco`² como atributos compostos e, `telefone`^{3,4}, como coluna (atributo) multivalorada (há mais de um valor de telefone para um único registro). Dê a eles o tratamento adequado conforme estabelece a definição de 1FN.

RESPOSTA:

Procedimento em duas etapas:

ETAPA 1 – Elimine o(s) grupo(s) de repetição, eventualmente existente(s), criando nova(s) tabela(s).

Em relação à tabela principal (não normalizada), o atributo `telefone` é uma coluna (atributo) multivalorada, isto é, um grupo de repetição. O tratamento a ser dado a um grupo de repetição existente em uma tabela, conforme visto em aula, é a eliminação do mesmo por meio da criação de uma nova tabela.

Em relação ao atributo `nome`, este não é um atributo simples, e sim, composto (vide subatributos na nota de rodapé). Logo, deve ser decomposto e, em seu lugar, criados os atributos `pnome` e `unome` (primeiro e último nome, respectivamente) na tabela inicial `CLIENTE`.

Em relação ao atributo `endereco`, da mesma forma como ocorreu com o atributo `nome`, o mesmo será decomposto em: `logradouro`, `numero`, `complemento`, `bairro` e `cep`, a serem criados na tabela inicial `CLIENTE`.

O atributo `telefone` é uma coluna (atributo) multivalorado, isto é, um grupo de repetição. A forma mais adequada de tratamento é a criação de uma nova tabela para armazenar os seus valores.

Assim, após a execução da ETAPA 1, resultam a tabela inicial, `CLIENTE`, e uma nova tabela denominada `TELEFONE` (lembrando, que não precisamos nomear a nova tabela nesta ETAPA 1), que contém os atributos:

¹ Atributo composto por: primeiro nome e último nome.

² Atributo composto por: logradouro, número, complemento, bairro e cep.

³ Atributo composto por: DDD e o número do telefone.

⁴ Atributo multivalorado que necessita armazenar telefones: residencial, profissional ou móvel.

BANCO DE DADOS 1 – Prof. Giovani
Exercício de Fixação 08 – EF08 – RESOLVIDO

- `cpf`, para vincular o cadastro de um novo telefone a um cliente existente;
- `ddd_telefone`, para armazenar o DDD do telefone a ser cadastrado;
- `nro_telefone`, para armazenar o número do telefone a ser cadastrado; e
- `tipo_telefone`, para armazenar o tipo do telefone a ser cadastrado (residencial, profissional ou móvel).

`CLIENTE (cpf, pnome, unome, logradouro, numero, complemento, bairro, cep)`

`TELEFONE (cpf, ddd_telefone, nro_telefone, tipo_telefone)`

ETAPA 2 – Identifique as chaves primárias das tabelas

Além da chave primária (`pk`), também foi assinalada a chave estrangeira (`fk`) nas tabelas, com o objetivo de facilitar a visualização dos relacionamentos existentes entre as tabelas.

Conforme prevê esta ETAPA 2, sua chave primária da nova tabela (`TELEFONE`) é composta pela chave primária da tabela inicial (`CLIENTE`), mais a chave primária da tabela aninhada.

`CLIENTE (cpf (pk), pnome, unome, logradouro, numero, complemento, bairro, cep, cod_cidade)`

`TELEFONE (cpf (pk, fk), ddd_telefone (pk), nro_telefone (pk), tipo_telefone)`

BANCO DE DADOS 1 – Prof. Giovani

Exercício de Fixação 08 – EF08 – RESOLVIDO

2. As tabelas a seguir abstraem uma realidade que permite armazenar as vendas (rastreadáveis por meio de um número de nota fiscal de venda) que um determinado empregado realizou em uma determinada data. Todos os campos relacionados encontram-se estruturados na tabela `VENDA`. Já na tabela `ITEMVENDA`, é possível armazenar a quantidade de itens vendidos a um determinado preço (por item), de um determinado produto, vinculado a um número de nota fiscal.

```
VENDA (numero_nf (pk), data_venda, cod_emp, nome_emp)
```

```
ITEMVENDA (numero_nf (pk, fk), cod_prod (pk), descricao_prod, qtde_item, preco_item)
```

As tabelas estão na 1FN. No entanto, a presença de dependências funcionais parciais e transitivas impede elas de estarem na 2FN e 3FN, respectivamente.

Realize os procedimentos de passagem necessários, descrevendo as etapas realizadas e apresentando, ao seu final, o resultado alcançado por cada forma normal executada.

RESPOSTA:

Passagem da 1FN para a 2FN
Conforme visto em aula, a passagem à 2FN consta da eliminação de dependências parciais, isto é, de atributos não-chave que dependem apenas de parte da chave primária composta, e não da chave primária como um todo.
A seguir são apresentadas as etapas realizadas na passagem para a 2FN.
<p>ETAPA 1 – Copiar para a 2FN a(s) tabela(s) que tiver(em) chave primária simples e/ou que, em relação as quais, não tenha sido identificada dependência parcial (no caso de tabela que tenha chave primária composta).</p> <pre>VENDA (numero_nf (pk), data_venda, cod_emp, nome_emp)</pre> <p><u>Comentário:</u> a tabela <code>VENDA</code> possui chave primária simples. Logo, pode-se afirmar que ela está na 2FN e a mesma comporá o resultado final, da passagem da 1FN para a 2FN. Já a tabela <code>ITEMVENDA</code>, por outro lado, possui uma chave primária composta, requerendo que a tabela seja analisada com o objetivo de identificar se há, ou não, presença de dependência funcional parcial.</p> <p>ETAPA 2 – Caso a tabela com chave primária composta tiver dependência parcial (requer que a dependência já tenha sido identificada), criar uma nova tabela que tenha como chave primária a parte da chave composta da tabela original (atributo determinante, em relação ao qual a dependência esteja ocorrendo), que deve permanecer também na tabela original.</p> <p><u>Comentário:</u> inicialmente, é necessário analisar a tabela <code>ITEMVENDA</code>, apresentada a seguir:</p> <pre>ITEMVENDA (numero_nf (pk, fk), cod_prod (pk), descricao_prod, qtde_item, preco_item)</pre> <p>Da análise resultante, percebe-se que <code>descricao_prod</code> depende apenas de <code>cod_prod</code>, que é parte da chave primária composta. Já os atributos <code>qtde_item</code> e <code>preco_item</code>, dependem da chave primária composta como um todo (<code>numero_nf</code> e <code>cod_prod</code>).</p> <p>Para facilitar a visualização das dependências⁵, a sua descrição textual é apresentada a seguir:</p> <pre>cod_prod → descricao_prod numero_nf, cod_prod → qtde_item, preco_item</pre>

⁵ Conforme dica (ou seja, não é obrigatório apresentar na resolução deste exercício) passada em aula, para melhor visualizar os atributos dependentes e realizar a sua distribuição, considere (i) traçar o diagrama de dependências ou, (ii) descrevê-las textualmente conforme o conceito de Determinação, ambos vistos em aula.

BANCO DE DADOS 1 – Prof. Giovani

Exercício de Fixação 08 – EF08 – RESOLVIDO

Uma vez identificada a dependência parcial, torna-se necessária a criação de uma nova tabela (conforme prevê a Etapa 2), que terá, como chave primária, a parte da chave primária composta da tabela original em relação a qual a dependência parcial esteja ocorrendo. A parte da chave primária composta deve permanecer também na tabela original (a ver se o nome ITEMVENDA permanecerá adequado). Assim, teremos a seguinte nova tabela, além da tabela original:

```
<nome_da_nova_tabela> (cod_prod (pk), ???, ...)  
<nome_da_tabela_original> (numero_nf (pk, fk), cod_prod (pk), ???, ...)
```

Os demais atributos (não-chave) serão distribuídos conforme previsto na ETAPA 3. Nessa etapa também passaremos a ter uma melhor visualização acerca dos nomes mais adequados para as tabelas resultantes.

ETAPA 3 - Remover da tabela original todos os atributos dependentes (parciais) que foram identificados, criando-os na nova tabela.

Em relação à **nova tabela**, criamos nela o atributo dependente apenas de `cod_prod`, ou seja, `descricao_prod`, e removemos o mesmo da tabela original (ITEMVENDA).



```
<nome_da_nova_tabela> (cod_prod (pk), descricao_prod)
```

Por fim, em relação à **tabela original**, restaram dois atributos (`qtde_item`, `preco_item`), dependendo da chave primária composta (`numero_nf`, `cod_prod`) como um todo:

```
<nome_da_tabela_original> (numero_nf (pk, fk), cod_prod (pk), qtde_item, preco_item)
```

Resultado da passagem da 1FN para a 2FN, após a eliminação das dependências parciais:

```
VENDA (numero_nf (pk), data_venda, cod_emp, nome_emp)  
PRODUTO (cod_prod (pk), descricao_prod)  
ITEMVENDA (numero_nf (pk, fk), cod_prod (pk, fk), qtde_item, preco_item)
```

Em relação a VENDA, esta já estava na 2FN, conforme constatado na ETAPA 1. A tabela PRODUTO, foi o resultado da decomposição/separação da tabela original ITEMVENDA, na qual foi identificada uma dependência parcial.

Note que, `cod_prod`, em ITEMVENDA, está agora definido também como chave estrangeira, uma vez que referencia PRODUTO por meio de `cod_prod`.

Continuação na próxima página...

BANCO DE DADOS 1 – Prof. Giovani

Exercício de Fixação 08 – EF08 – RESOLVIDO

Passagem da 2FN para a 3FN

A passagem à 3FN consta da eliminação das dependências transitivas, isto é, de atributos não-chave que dependem funcionalmente de outro atributo não-chave.

A seguir são apresentadas as etapas realizadas na passagem para a 3FN.

ETAPA 1 - Copiar para a 3FN a(s) tabela(s) que tiver(em) menos que dois atributos não-chave

Do resultado da passagem da 1FN para a 2FN, percebe-se que a tabela **PRODUTO** possui apenas um atributo não-chave. Assim, pode-se afirmar que a mesma já está na 3FN, pois não há a possibilidade de nela existir uma dependência transitiva. Da mesma forma, na tabela **ITEMVENDA** não identificamos uma dependência transitiva entre os dois atributos não-chave existentes. Assim, em relação às tabelas **PRODUTO** e **ITEMVENDA**, pode-se afirmar que as mesmas já estão na 3FN.


```
PRODUTO (cod_prod (pk), descricao_prod)
ITEMVENDA (numero_nf (pk, fk), cod_prod (pk, fk), qtde_item, preco_item)
```

ETAPA 2 - Para todas as dependências transitivas identificadas, apresente o atributo determinante como chave primária de uma nova tabela a ser criada. O atributo determinante, em relação ao qual a dependência esteja ocorrendo, deve permanecer também na tabela original.

Do resultado da passagem da 1FN para a 2FN, percebe-se que a tabela **VENDA** possui dois atributos não-chave e, entre eles, há uma dependência funcional transitiva.

Para facilitar a visualização, o diagrama de dependência é apresentado a seguir:

```
VENDA (numero_nf (pk), data_venda, cod_emp, nome_emp)
```



Identificada a dependência transitiva, torna-se necessária a criação de uma nova tabela (conforme prevê a ETAPA 2), que deverá ter, como chave primária, o atributo não-chave determinante em relação ao qual a dependência esteja ocorrendo. O mesmo deverá permanecer também na tabela original (**VENDA**, onde a dependência transitiva está ocorrendo). Assim, teremos a seguinte (nova) tabela:


```
<nome_da_nova_tabela> (cod_emp (pk), ???, ...)
```

O atributo (não-chave) determinado (**nome_emp**) será distribuído conforme previsto na ETAPA 3.

ETAPA 3 - Remover da tabela original (**VENDA**) todos os atributos dependentes (transitivos) que foram identificados, criando-os na nova tabela.

Em relação à **nova tabela**, criamos nela o atributo dependente de **cod_emp**, ou seja, **nome_emp**, e removemos o mesmo da tabela original (**VENDA**).

```
VENDA (numero_nf (pk), data_venda, cod_emp, nome_emp)
```



```
<nome_da_nova_tabela> (cod_emp (pk), nome_emp)
```

BANCO DE DADOS 1 – Prof. Giovani

Exercício de Fixação 08 – EF08 – RESOLVIDO

Resultado da passagem da 2FN para a 3FN, após a eliminação das dependências transitivas:

```
VENDA (numero_nf (pk), data_venda, cod_emp (fk))  
PRODUTO (cod_prod (pk), descricao_prod)  
ITEMVENDA (numero_nf (pk, fk), cod_prod (pk, fk), qtde_item, preco_item)  
EMPREGADO (cod_emp (pk), nome_emp)
```

Em relação a **PRODUTO** e **ITEMVENDA**, ambas já estavam na 3FN, conforme constatado na ETAPA 1. A tabela **EMPREGADO**, foi o resultado da decomposição/separação da tabela **VENDA**, na qual foi identificada uma dependência transitiva.

Note que, **cod_emp**, em **VENDA**, está agora definido como chave estrangeira, uma vez que referencia **EMPREGADO** por meio de **cod_emp**.