

Ao final deste assunto, você será capaz de:

- Compreender os objetivos do Projeto Lógico de um banco de dados
- Fazer a modelagem lógica de um banco de dados, reconhecendo os elementos desse modelo através do uso do Diagrama de Estruturas de Dados, Diagrama de Acesso Imediato a Dados e Mapeamento de Estruturas
- Obter o Projeto Lógico através das técnicas de normalização

2.3.1. Características

- Fase do projeto de BD propriamente dito.
- Voltado para o ambiente de implementação (aplicação e SGBD)
- Depende da Estrutura Lógica do SGBD

Estrutura Lógica: forma ou desenho lógico que determina como os dados são armazenados e recuperados logicamente.

As estruturas lógicas de SGBD vieram evoluindo de acordo com a abordagem adotada:

- 1) Orientação a Registro:
 - a. Abordagem Hierárquica;
 - b. Abordagem em Redes;
 - c. Abordagem Relacional.
- 2) Orientação a Objetos:
 - a. Abordagem Orientada a Objetos.

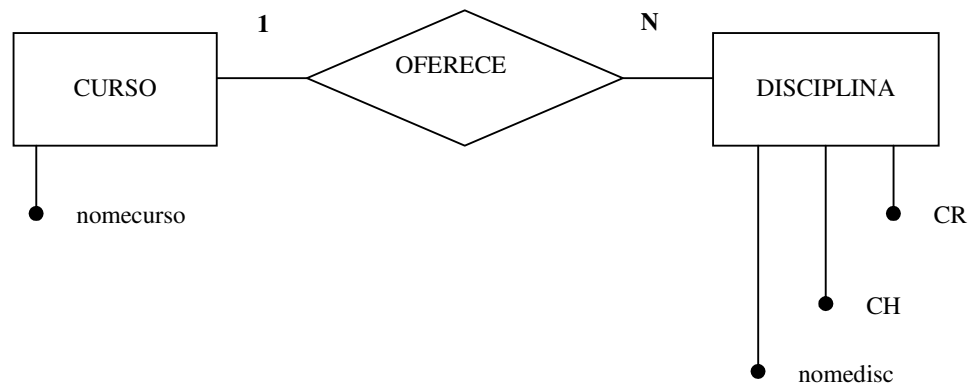
Mais tarde estaremos conceituando cada uma das abordagens.

2.3.2. Construção do Projeto Lógico

Converte-se o Modelo Conceitual para o Modelo Lógico:

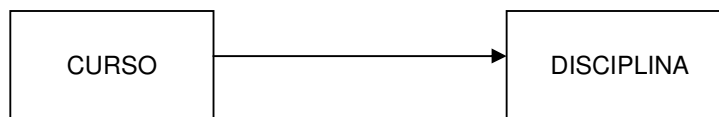
- 1) Dado o mapeamento de Entidades e Relacionamentos apresentados no MER (DER).

Exemplo 1:



- 2) Os elementos armazenadores de dados passam a ser conhecidos como estruturas de dados.
- 3) As estruturas de dados identificadas são representadas por retângulos e deve existir uma linha ligando as estruturas que tiverem relacionamentos entre si.
- 4) Uma seta indica o sentido do relacionamento entre as estruturas.
- 5) O número de instâncias relacionadas define o sentido do relacionamento.

Diagrama de Estruturas de Dados (DED)



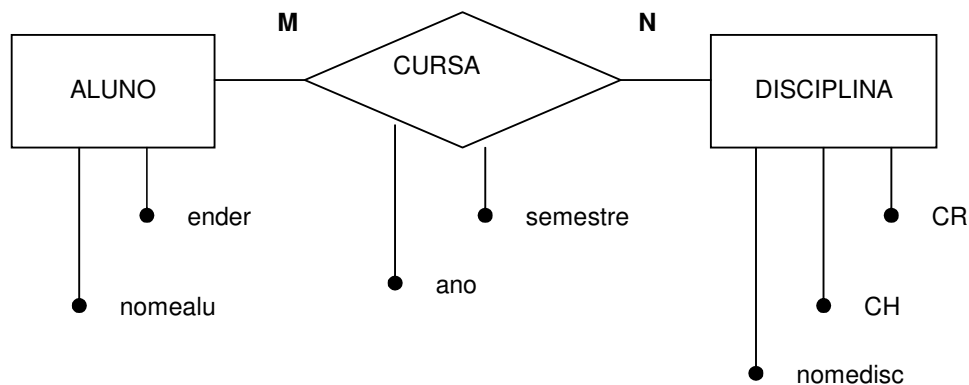
Mapeamento de Estruturas

6) Identifica-se e lista os atributos de cada estrutura:

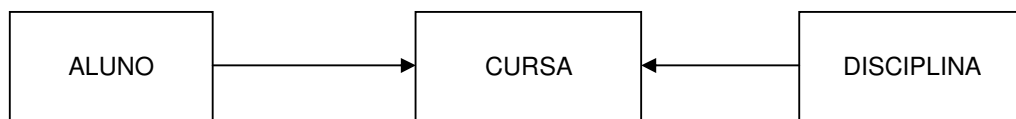
CURSO
nomecurso

DISCIPLINA
nomedisc
CH
CR

Exemplo 2:



Neste caso, o relacionamento contém atributos, ou seja, é necessário armazenar dados que dizem respeito ao relacionamento e não às entidades. Deste modo, entendemos que o relacionamento precisa converter-se em uma estrutura de dados. Logo, o diagrama de estruturas ficaria da seguinte forma.



O mapeamento das estruturas é listado a seguir:

ALUNO
nomealu
Ender
CNPJ

DISCIPLINA
nomedisc
CH

CURSA
ano
semestre

Suponhamos que as estruturas apresentadas fossem utilizadas para implementar um banco de dados, como faríamos para obter os dados?

Antecipamos que pela abordagem relacional, bastaria verificar o dado por qualquer atributo. Mas, pelas outras abordagens, isso não é possível. Além do mais, se o volume de dados for muito grande, a pesquisa diretamente no arquivo físico acaba por tornar-se muito lenta. Logo, precisamos de um atributo que possa ser transformado em um índice de pesquisa. Esse atributo de pesquisa é chamado de atributo **chave** (key, em inglês).

Portanto, podemos escolher um ou mais atributos para ser a nossa chave de pesquisa.

Mas, se quisermos recuperar um e somente um registro ao fazermos uma pesquisa? Ou seja, a chave escolher deve ser tal que ao informarmos o seu valor seja obtido somente um registro.

No caso de Disciplina, não existe um atributo que possa representar unicamente uma disciplina. Neste caso, ou seja, se não tivermos um atributo que possa cumprir esse papel podemos criá-lo. No exemplo, essa chave poderia se chamar coddisc (código da disciplina).

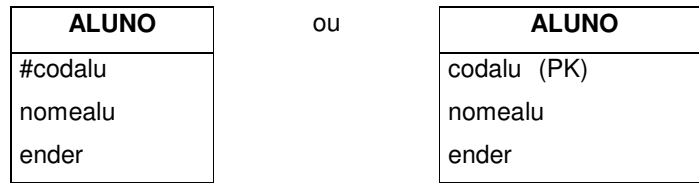
No caso de Aluno, existe o atributo CNPF, que poderia cumprir essa função mas, se existirem alunos sem CNPF, o atributo não alcançará o objetivo. Portanto, poderíamos criar o atributo codalu.

No caso de Aluno, ficamos com duas possibilidades de obter um aluno: pelo atributo codalu ou CNPF. Os atributos que permitem a recuperação de registros únicos de um arquivo são chamados de Chaves Candidatas (CANDIDATE KEY).

ALUNO	DISCIPLINA
Chaves candidatas: codalu, CNPF	Chaves candidatas: coddisc

A estrutura Aluno apresenta duas chaves candidatas, mas o atributo CNPF pode não ter valor caso o aluno não tenha CNPF, portanto, efetivamente, a chave que representa qualquer aluno é codalu. Esse tipo de chave, ou seja, a chave que representa qualquer registro de um arquivo é

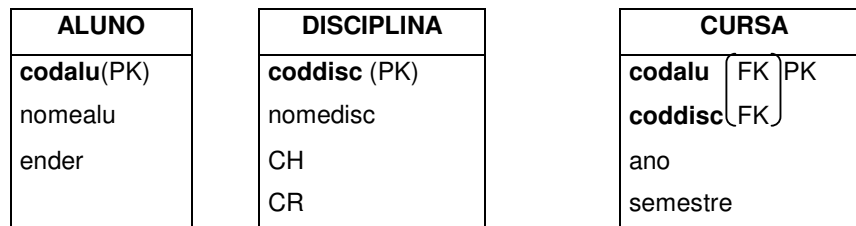
conhecida como Chave Primária (PRIMARY KEY), que pode ser indicada colocando-se um # ou PK ao lado do atributo. Exemplo:



Se desejarmos recuperar com frequência registros de um banco de dados através de um atributo que não a chave primária, pode ser importante que criemos um índice auxiliar para esse atributo a fim de acelerar essa pesquisa. Esses atributos são conhecidos como Chaves Secundárias (SECONDARY KEY – SK). Normalmente, pode ser interessante definir as chaves candidatas como SK.

Como definiríamos a chave primária de uma estrutura resultante de um relacionamento?

Neste caso, o relacionamento não tem chave própria e a chave primária terá que ser obtida a partir das chaves primárias das entidades relacionadas. Ou seja, no exemplo 2 as chaves primárias de Aluno (codalu) e de Disciplina (coddisc) migram para a estrutura Cursa, de modo que passa a ser (codalu, coddisc).



Portanto, as setas indicam o deslocamento da chave primária de uma estrutura para a outra. A chave primária de uma estrutura quando é definida na estrutura para a qual ela se deslocou chamamos de Chave Estrangeira (FOREIGN KEY – FK)

Exemplo:

Construir o projeto lógico do problema obtendo-o a partir do seu Modelo Conceitual.

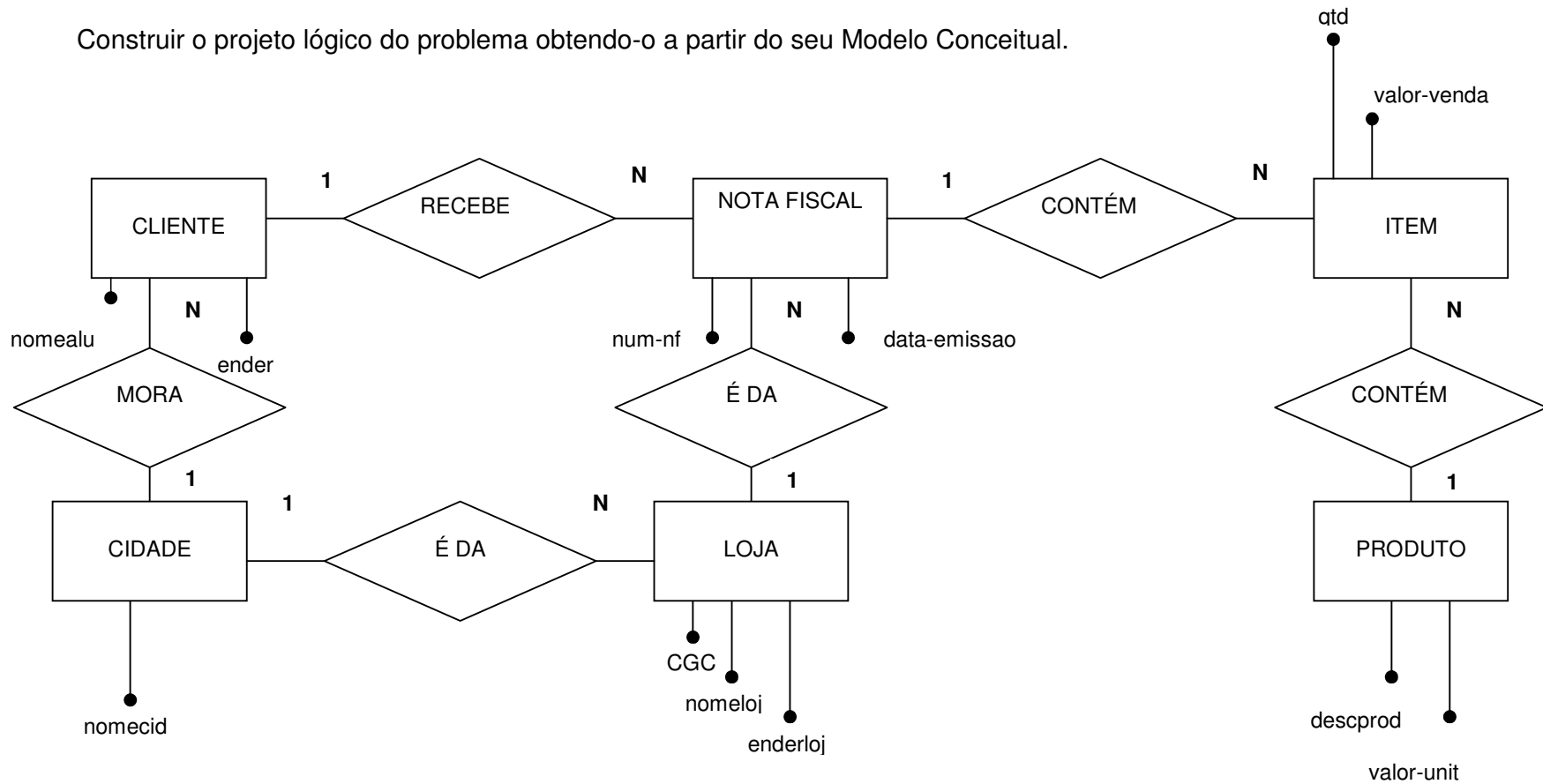
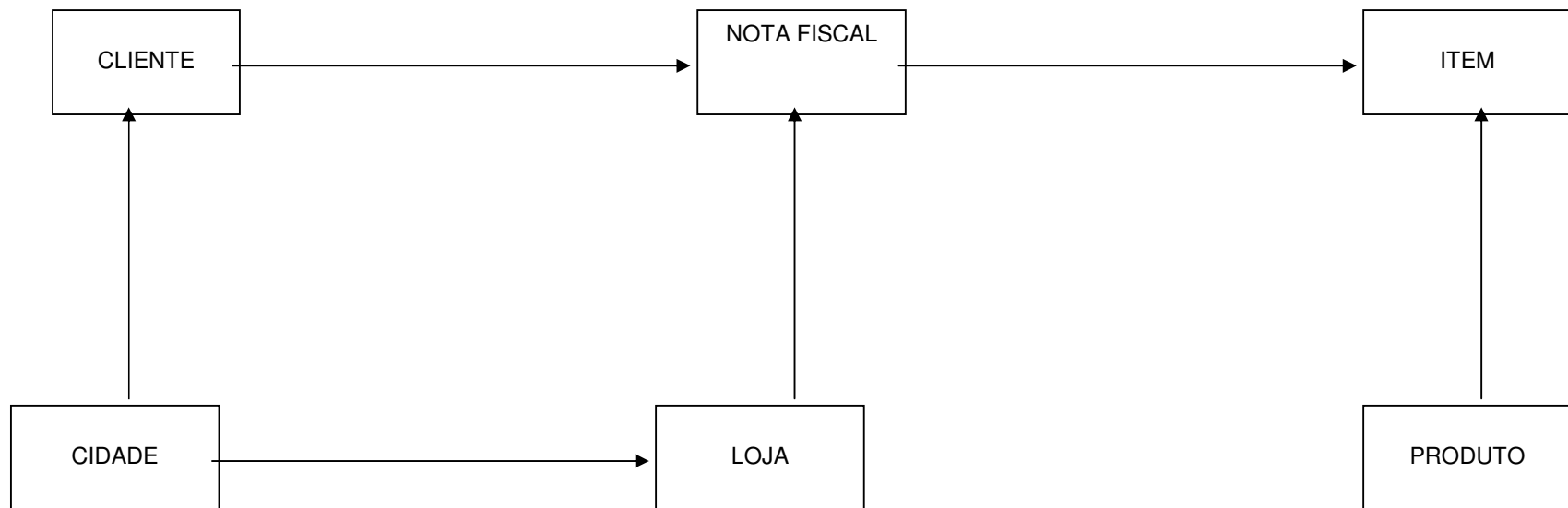


Diagrama de Estruturas

Mapeamento de Estruturas

CIDADE	
codcid	PK
nomecid	

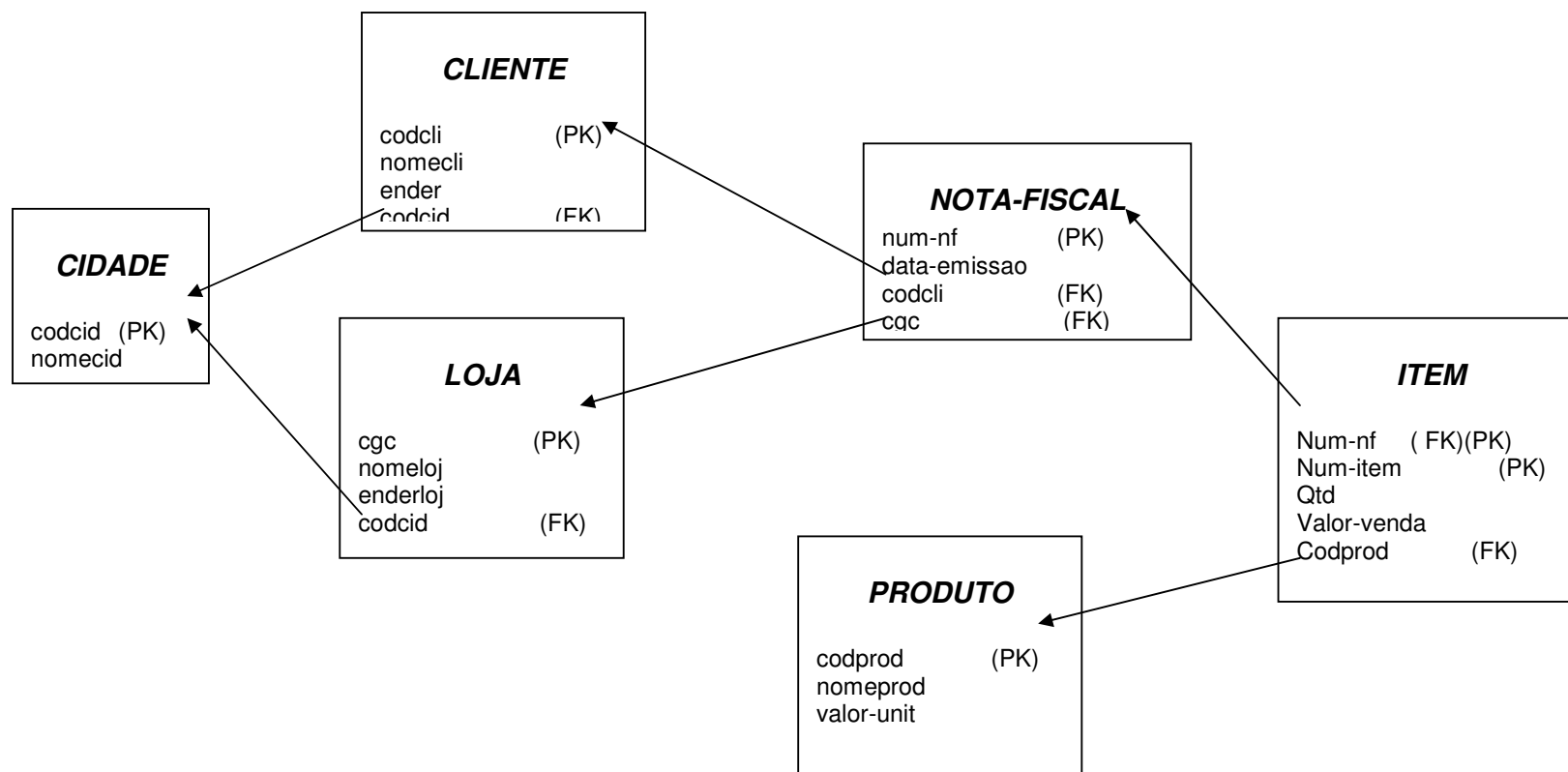
NOTA-FISCAL	
num-nf	PK
data-emissao	
codcli	FK
cgc	FK

CLIENTE	
codcli	PK
nomecli	
ender	
codcid	FK

ITEM			
Num-nf	FK	PK	
Num-item		PK	
Qtd			
Valor-venda			
Codprod	FK		

LOJA	
cgc	PK
nomejoj	
enderjoj	
codcid	FK

PRODUTO	
codprod	PK
nomeprod	
valor-unit	

Diagrama de Acesso Imediato a Dados (DAID)

2.3.3. Normalização

Conceito

Organização dos dados de forma mais simples a fim de ter o projeto lógico o menos complexo possível e a fim de minimizar o gasto com recursos no projeto físico.

Busca evitar que o projeto lógico dê margem para redundâncias e inconsistências de dados.

Por quê normalização?

O termo normalização é uma alusão a ação de tornar algo normal. Quando produzimos um projeto lógico que dê margem a repetições de valores, redundâncias e inconsistências de dados, dizemos que nosso projeto apresenta anomalias, pois esses são exatamente os problemas que mais nos atraíram quando partimos para a tecnologia de banco de dados.

Se eliminarmos as anomalias de nosso projeto este se tornará normal.

O ponto de partida para a aplicação da normalização é, normalmente, o conjunto de documentos, arquivos, relatórios, etc, de uma organização, que já dão pistas de vários dados relacionados e de interesse. Estes dados estão normalmente em um formato estruturado. A utilização das regras de normalização, neste caso, nivela estruturas aninhadas (mantém apenas atributos atômicos) e após, decompõe em tantas tabelas quantas forem necessárias para evitar os problemas comentados anteriormente. A normalização pode também ser aplicada após a realização da etapa de projeto lógico de BD, para validar tabelas geradas a partir do mapeamento do diagrama ER. Na verdade, o ideal é que a normalização e o projeto conceitual/lógico de BD sejam técnicas complementares, isto é, sejam aplicadas em paralelo e depois comparadas, visando alcançar uma organização melhor possível dos dados em um BD relacional.

Formas Normais (FN)

O processo de normalização é feito de forma gradual, de modo que a cada fase ou forma estaremos simplificando o nosso modelo até que não hajam mais anomalias.

A normalização se processa projetando-se para fora da estrutura a anomalia e, para não perdermos o dado, ele é re-ligado através do atributo ao qual os atributos projetados têm afinidade.

Primeira Forma Normal (1FN)

Buscamos eliminar os atributos multivalorados das nossas estruturas. Uma estrutura está na primeira forma normal (1FN) quando não tem atributos multivalorados.

Segunda Forma Normal (2FN)

Buscamos retirar da estrutura todos os atributos que tiverem dependência funcional de parte da chave. Portanto, só vale a pena verificar se a estrutura está na 2FN se a chave da mesma não for simples, isto é, deve ser pelo menos dupla.

Terceira Forma Normal (3FN)

Buscamos retirar da estrutura todos os atributos que tiverem dependência funcional de outro atributo que não a chave ou parte da chave. Portanto, se a estrutura não tiver esse tipo de dependência funcional, ela estará na 3FN.

Forma Normal Boyce-Codd (FNBC)

Buscamos retirar da estrutura todos os atributos que tiverem dependência funcional de um outro atributo que não a chave primária. Diferentemente da 3FN, neste caso, a dependência deve ser de uma chave candidata.

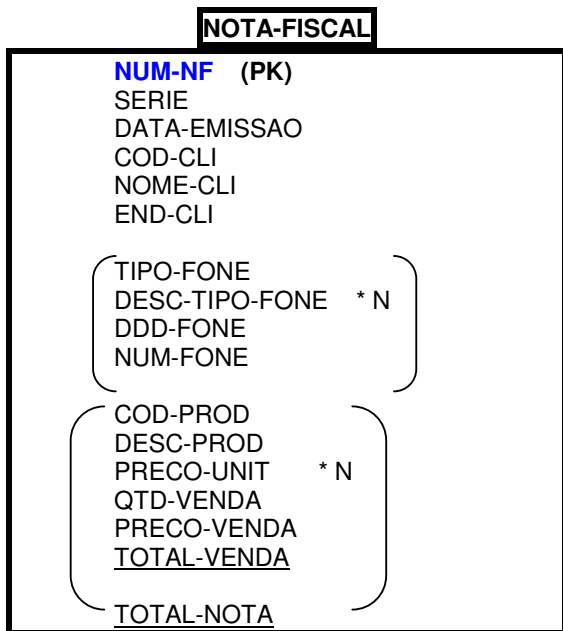
Existem ainda a 4FN e a 5FN, que só devem ser usadas para estruturas que tenham mais que dois atributos concatenados na chave.

Exemplo 1:

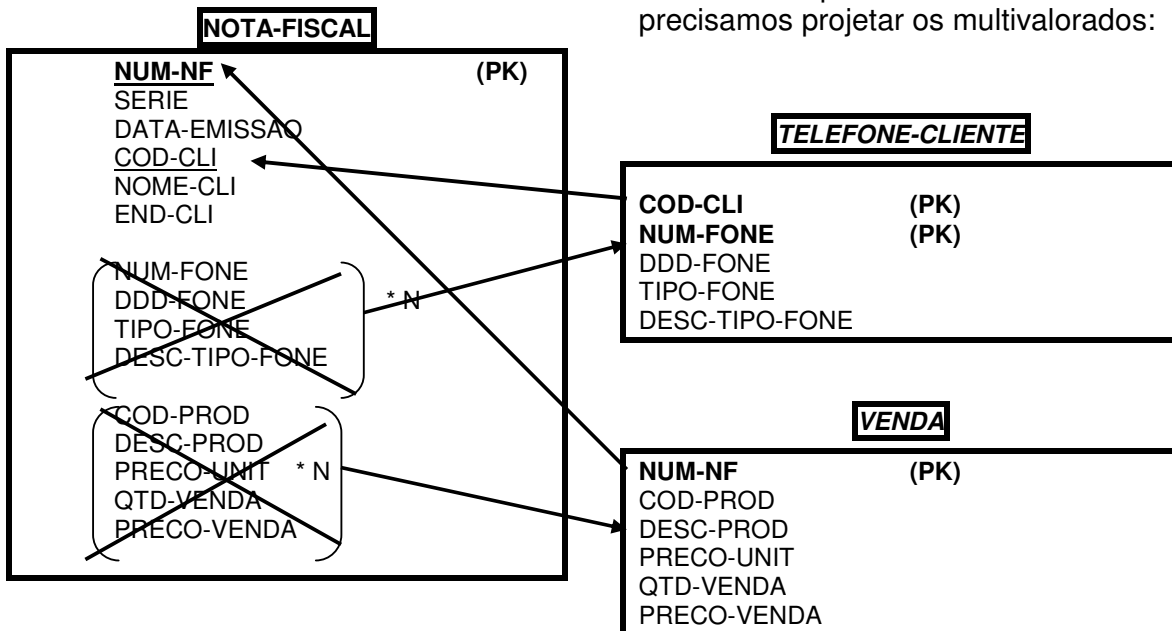
Considere o seguinte documento que será usado para a produção das estruturas de dados por normalização:

NOTA FISCAL																																															
Número NF: _____ (NUM-NF)		Série: _____ (SERIE)		Emissão: ____/____/____ (DATA-EMISSAO)																																											
Cliente: <div style="margin-left: 40px;"> Nome: _____ (COD-CLI, NOME-CLI) Endereço: _____ (END-CLI) Telefones: <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> Tipo: _____ (TIPO-FONE, DESC-TIPO-FONE) DDD: _____ Número: _____ (DDD-FONE) (NUM-FONE) </div> <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> Tipo: _____ (TIPO-FONE, DESC-TIPO-FONE) DDD: _____ Número: _____ (DDD-FONE) (NUM-FONE) </div> <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> Tipo: _____ (TIPO-FONE, DESC-TIPO-FONE) DDD: _____ Número: _____ (DDD-FONE) (NUM-FONE) </div> </div>																																															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Cód (COD-PROD)</th> <th style="width: 40%;">Descrição (DESC-PROD)</th> <th style="width: 10%;">Quant (QTD-VENDA)</th> <th style="width: 15%;">Preço Unitário (PRECO-UNIT)</th> <th style="width: 15%;">Preço Venda (PRECO-VENDA)</th> <th style="width: 10%;">Total Venda (TOTAL-VENDA)</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr> <td colspan="5" style="text-align: right;">Total (TOTAL-NOTA)</td> <td> </td> </tr> </tbody> </table>						Cód (COD-PROD)	Descrição (DESC-PROD)	Quant (QTD-VENDA)	Preço Unitário (PRECO-UNIT)	Preço Venda (PRECO-VENDA)	Total Venda (TOTAL-VENDA)																															Total (TOTAL-NOTA)					
Cód (COD-PROD)	Descrição (DESC-PROD)	Quant (QTD-VENDA)	Preço Unitário (PRECO-UNIT)	Preço Venda (PRECO-VENDA)	Total Venda (TOTAL-VENDA)																																										
Total (TOTAL-NOTA)																																															

Aplicando a Primeira Forma Normal (1FN)



O que resulta em:



Antes de aplicarmos a 1FN,
 devemos verificar se existem atributos

redundantes, os quais podem ser obtidos a partir de outros atributos. Essas redundâncias podem provocar inconsistências e, portanto, devem ser eliminadas.

A estrutura apresenta o atributo

$TOTAL-VENDA = QTD-VENDA \times PRECO-VENDA$.

Ou seja, o atributo pode ser obtido através de operação entre outros dois atributos. Portanto, é desnecessário manter o atributo TOTAL-VENDA, além do fato que o mesmo pode conduzir a inconsistência de dados.

O mesmo podemos dizer de TOTAL-NOTA, que é o somatório dos valores de TOTAL-VENDA. Pelo mesmo motivo anterior, este atributo deve ser desconsiderado, pois pode ser calculado a partir de atributos existentes.

Pela primeira forma normal precisamos projetar os multivalorados:

Resultado da Primeira Forma Normal (1FN)

NOTA-FISCAL

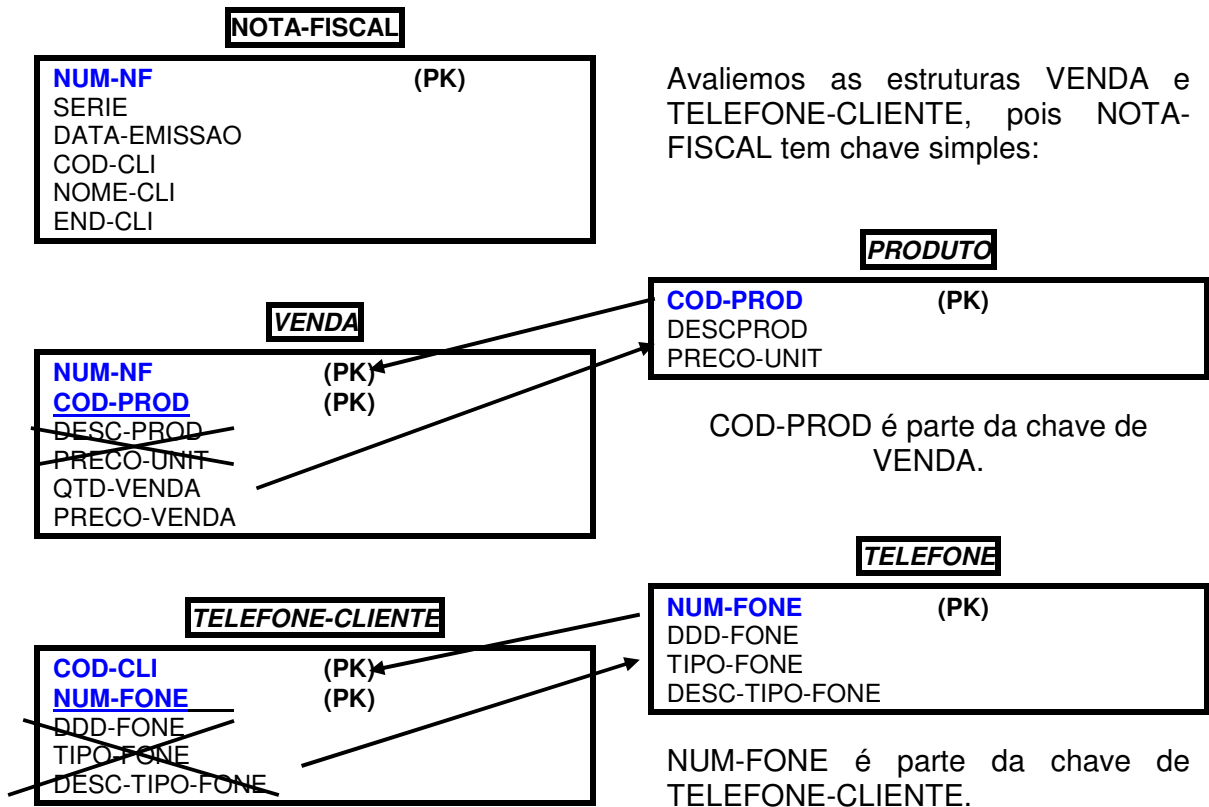
NUM-NF	(PK)
SERIE	
DATA-EMISSAO	
COD-CLI	
NOME-CLI	
END-CLI	

TELEFONE-CLIENTE

COD-CLI	(PK)
NUM-FONE	(PK)
DDD-FONE	
TIPO-FONE	
DESC-TIPO-FONE	

VENDA

NUM-NF	(PK) (FK)
COD-PROD	(PK) (FK)
DESC-PROD	
PRECO-UNIT	
QTD-VENDA	
PRECO-VENDA	

Aplicando a Segunda Forma Normal (2FN)

Resultado da Segunda Forma Normal (2FN)

a chave primária de TELEFONE passe a ser DDD-FONE + NUM-FONE.

NOTA-FISCAL

NUM-NF	(PK)
SERIE	
DATA-EMISSAO	
COD-CLI	
NOME-CLI	
END-CLI	

TELEFONE-CLIENTE

COD-CLI	(PK)	(FK)
NUM-FONE	(PK)	(FK)

TELEFONE

NUM-FONE	(PK)
DDD-FONE	
TIPO-FONE	
DESC-TIPO-FONE	

VENDA

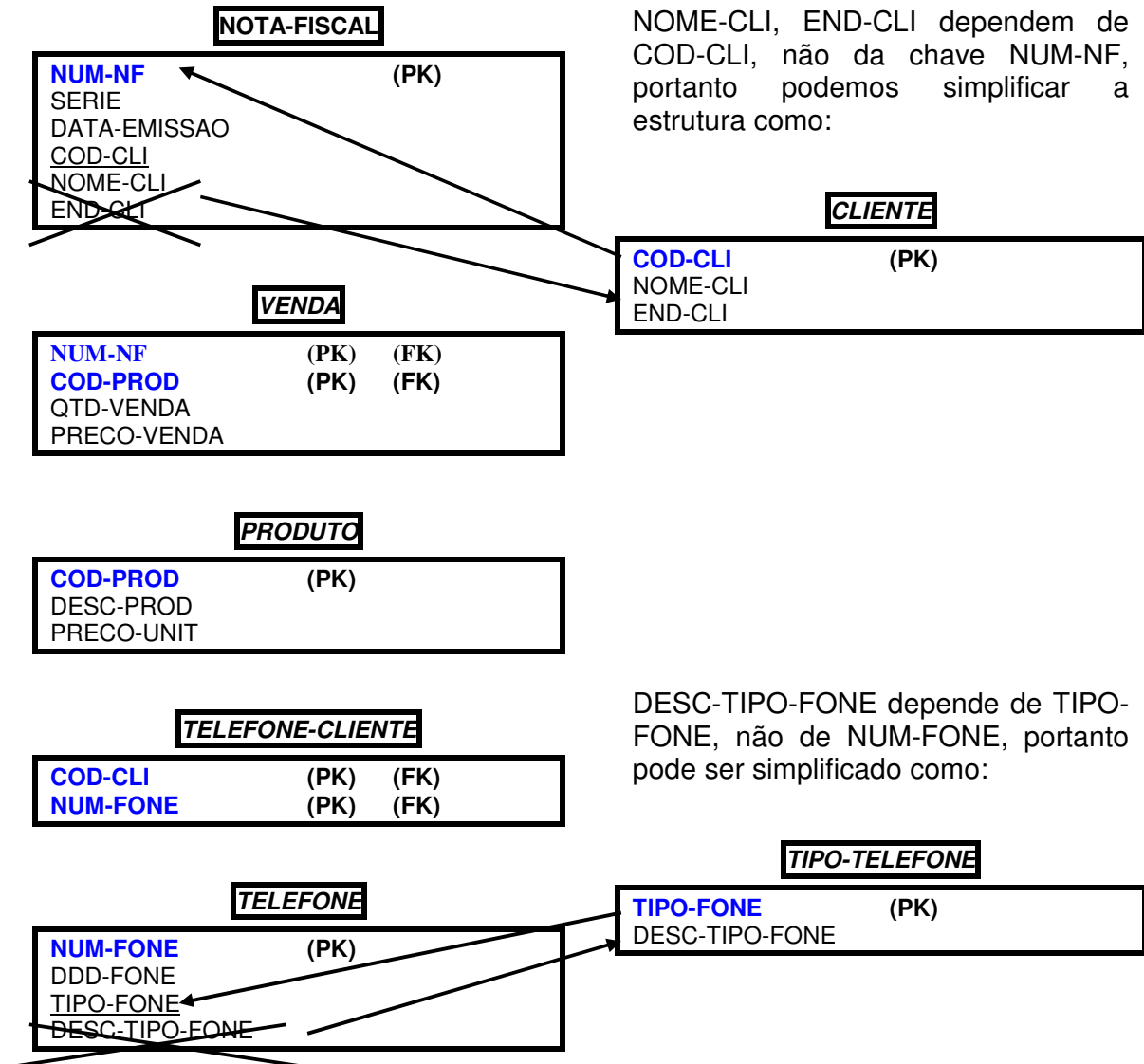
NUM-NF	(PK)	(FK)
COD-PROD	(PK)	(FK)
QTD-VENDA		
PRECO-VENDA		

PRODUTO

COD-PROD	(PK)
DESCPROD	
PRECO-UNIT	

* Se houver a possibilidade de existirem NUM-FONE idênticos em outros DDD-FONE, é necessários que

Aplicando a Terceira Forma Normal (3FN)



Resultado da Terceira Forma Normal (3FN)

NOTA-FISCAL

NUM-NF	(PK)
SERIE	
DATA-EMISSAO	
COD-CLI	(FK)

TIPO-TELEFONE

TIPO-FONE	(PK)
DESC-TIPO-FONE	

PRODUTO

COD-PROD	(PK)
DESC-PROD	
PRECO-UNIT	

TELEFONE-CLIENTE

COD-CLI	(PK)	(FK)
NUM-FONE	(PK)	(FK)

CLIENTE

COD-CLI	(PK)
NOME-CLI	
END-CLI	

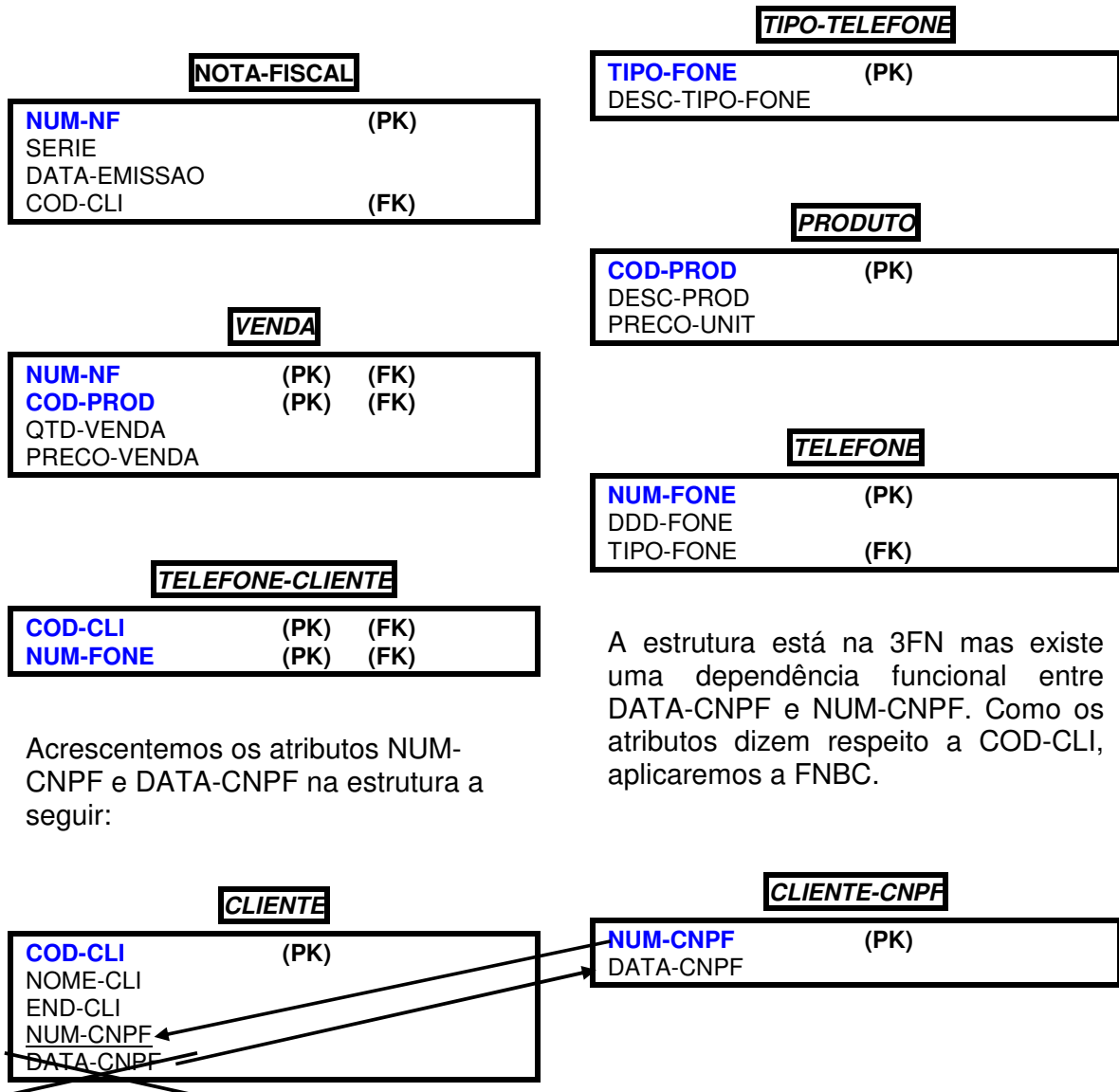
VENDA

NUM-NF	(PK)	(FK)
COD-PROD	(PK)	(FK)
QTD-VENDA		
PRECO-VENDA		

TELEFONE

NUM-FONE	(PK)
DDD-FONE	
TIPO-FONE	(FK)

Aplicando a Forma Normal Boyce-Codd (FNBC)



Resultado da Forma Normal Boyce-Codd (FNBC)

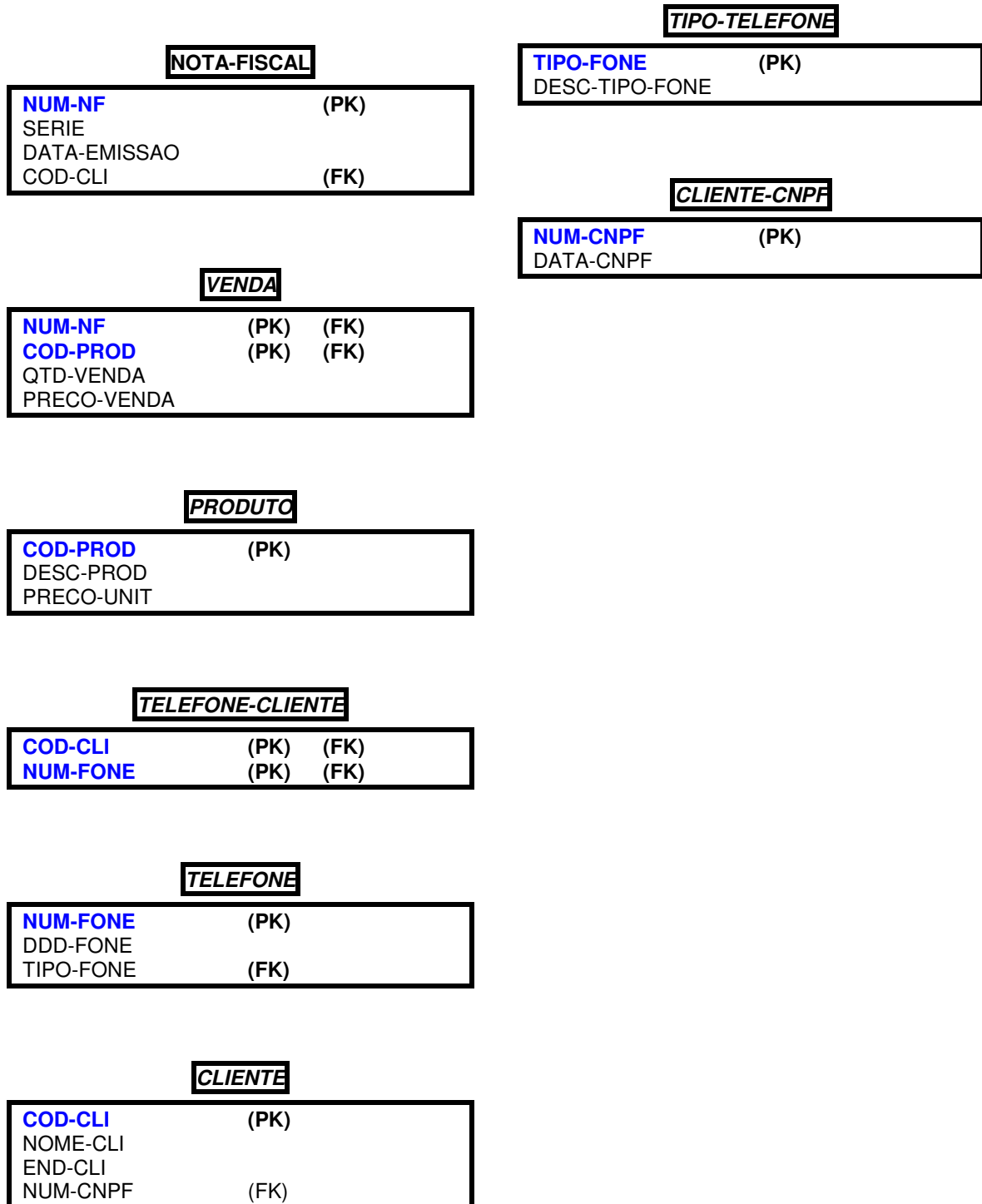
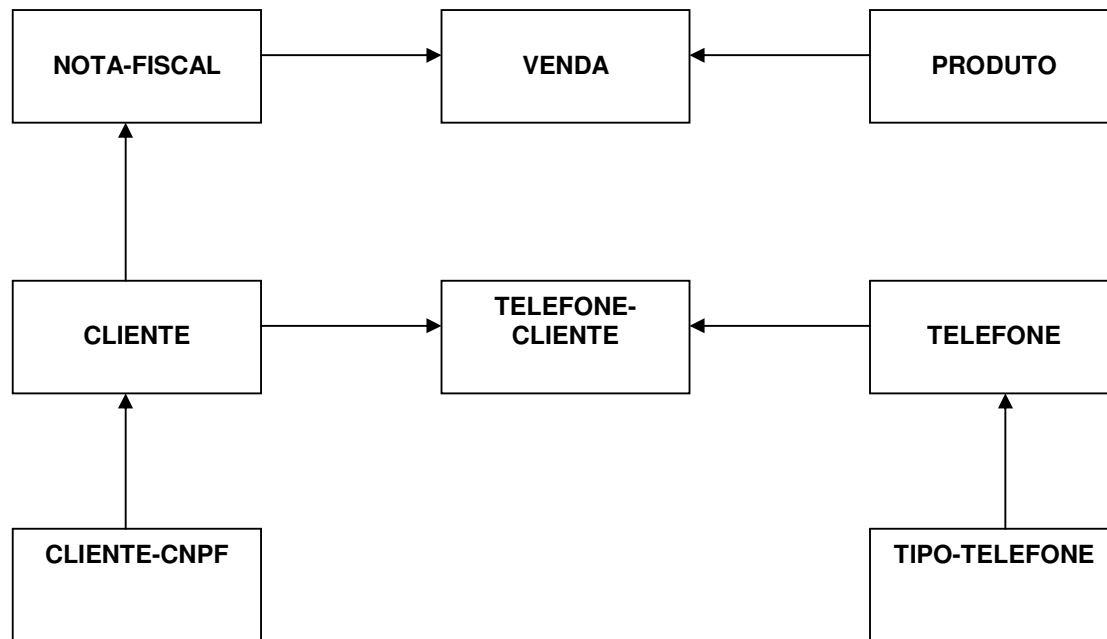


Diagrama de Estruturas de Dados (DED)**Outras formas normais**

Normalmente uma relação na BCFN também já se encontra na 4FN e 5FN. Estas últimas surgem para resolver casos muito raros.

Enquanto que nas formas reais 1FN até a FNBC o que se busca retirar são redundâncias que puderem ser detectadas a partir das estruturas de dados apresentadas, ou seja, conhecendo-se o esquema a ser definido para o banco de dados pode-se fazer a normalização, nas demais formas normais (4FN e 5FN) é importante perceber como ficarão as ligações entre as estruturas normalizadas após a normalização também do ponto de vista de possíveis conteúdos, ou seja, neste caso, é necessário conhecermos as possíveis instâncias dos atributos chaves a considerar.

Em resumo:

1FN -> FNBC

Dependências Funcionais

4FN e 5FN

Dependências Multivaloradas

Observações:

Para verificarmos se uma estrutura precisará ser avaliada quanto a 4FN e 5FN basta observarmos se a estrutura em questão atende as seguintes condições:

- 1) A estrutura tem a chave primária composta por menos de três atributos;
- 2) A estrutura tem a chave primária composta por mais de dois atributos, mas não tem atributos não chave.

Se uma estrutura estiver na FNBC e atender as condições apresentadas nos itens anteriores, pode-se afirmar que a mesma está na 4FN e na 5FN.

Quarta Forma Normal (4FN)

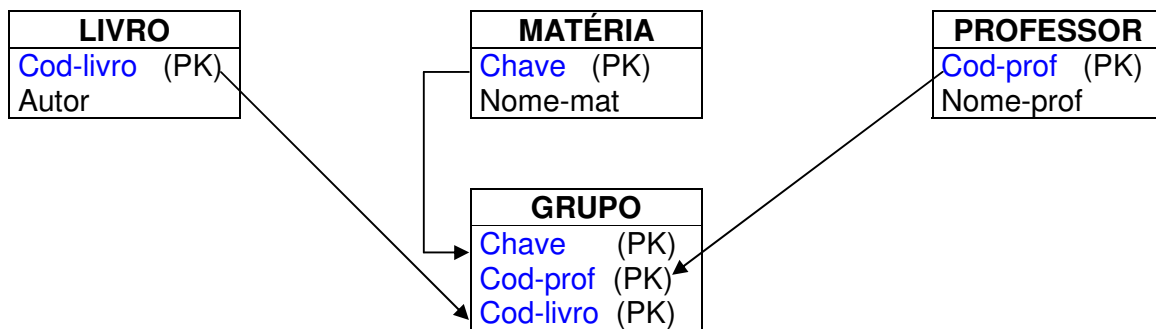
Ocorre quando se tem tabelas que mantêm relacionamentos ternários ou superiores e se detecta dependências funcionais multivaloradas, ou seja, um ou mais atributos determinam vários valores de um outro atributo. São consideradas, para efeito de análise da 4FN, apenas tabelas com Chave Primária formada por uma tripla (ou quádrupla, etc) que não tenham atributos não-chave. Normalmente, se detecta a necessidade da aplicação da 4FN quando se tem mais de um atributo multivalorado em uma tabela não normalizada. Estes atributos acabam fazendo parte da Chave Primária desta tabela na 1FN (pois cada um dos seus valores deve estar associado aos demais atributos da tabela), porém, isolados, eles não determinam nenhum atributo não-chave. Isto resulta em uma tabela com Chave Primária tripla (no mínimo), onde um deles é o identificador da tabela propriamente dito. Assim, quando se detecta uma Dependência Funcional (DF) multivalorada entre um atributo e o identificador da tabela, deve-se gerar uma tabela com apenas uma Chave Primária formada pelo par (identificador da tabela, atributo multivalorado). Caso esta decomposição não fosse feita, haveria muita redundância destes pares na tabela do relacionamento triplo ou superior (para representar todas as combinações possíveis dos três ou mais atributos, o que caracteriza uma dependência funcional multivalorada).

Exemplo:

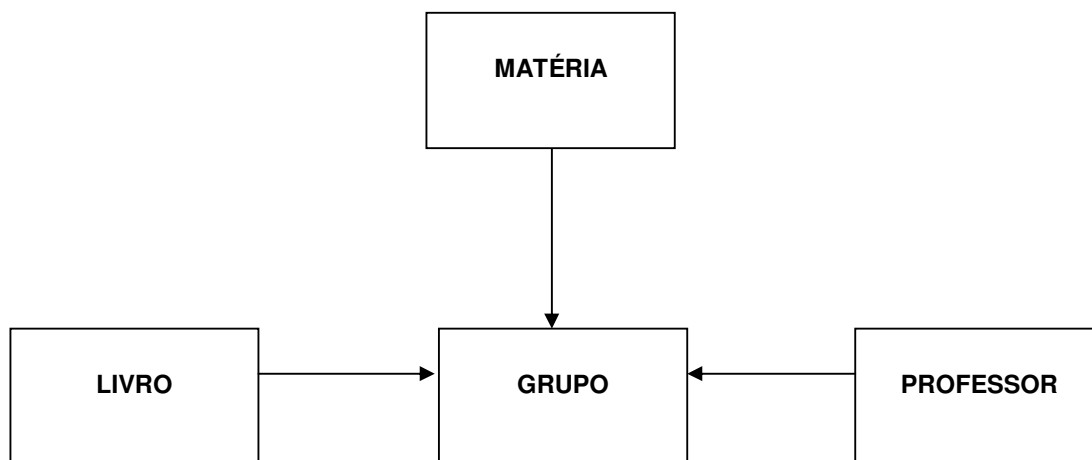
Em uma universidade se deseja fazer o controle de matérias, professores e livros de texto.

- Uma matéria pode ser ministrada por um ou mais professores.
- Uma matéria tem associados um ou mais livros de texto.
- Um professor pode ministrar uma ou mais matérias.
- Um professor ao ministrar uma matéria usa todos os livros de texto dessa matéria.
- Um livro de texto pode ser utilizado em uma ou mais matérias.

Deste modo, um possível desenho seria o que é mostrado na figura a seguir:



Cujo diagrama de estruturas de dados (DED) é o seguinte:



Analisando detalhadamente a relação GRUPO, nós podemos perceber que essa estrutura é toda composta por chaves, uma vez que um mesmo livro pode se repetir para diferentes combinações de professor e matéria:

- Um mesmo professor pode se repetir para diferentes combinações de matérias e livros.
- Uma mesma matéria pode se repetir para diferentes combinações de professor e livro.

Neste sentido, como a estrutura GRUPO é toda composta por chaves, já está na FNBC, mas tem alguns problemas:

Ao se incluir um novo professor para uma matéria temos o problema de precisar incluir tantos registros de livros quantos hajam para a matéria.

Ao se incluir um livro para uma matéria, tem que ser incluídos tantos registros de professores quantos hajam para a matéria..

A FNBC não nos ajudar a corrigir este problema, mas o que se requer é uma forma normal adicional:

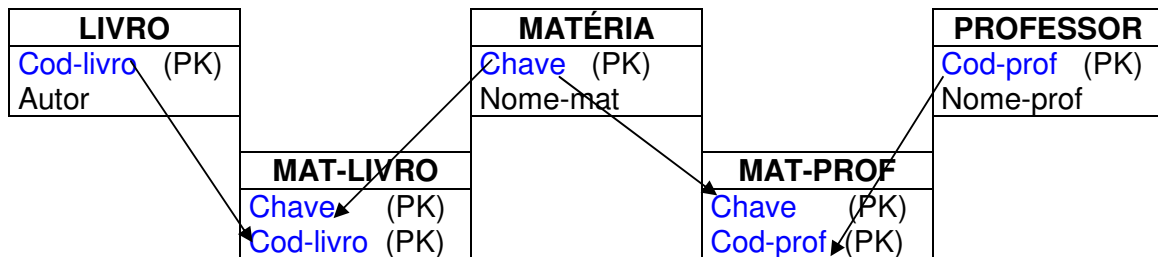
Uma relação está na Quarta Forma Normal (4FN) se, ao ter dependências multivaloradas da forma A-B, todas as dependências funcionais dependem de A.

Quer dizer, que todas as dependências multivaloradas são dependências funcionais. Para o caso da estrutura analisada temos:

CHAVE - COD-PROF

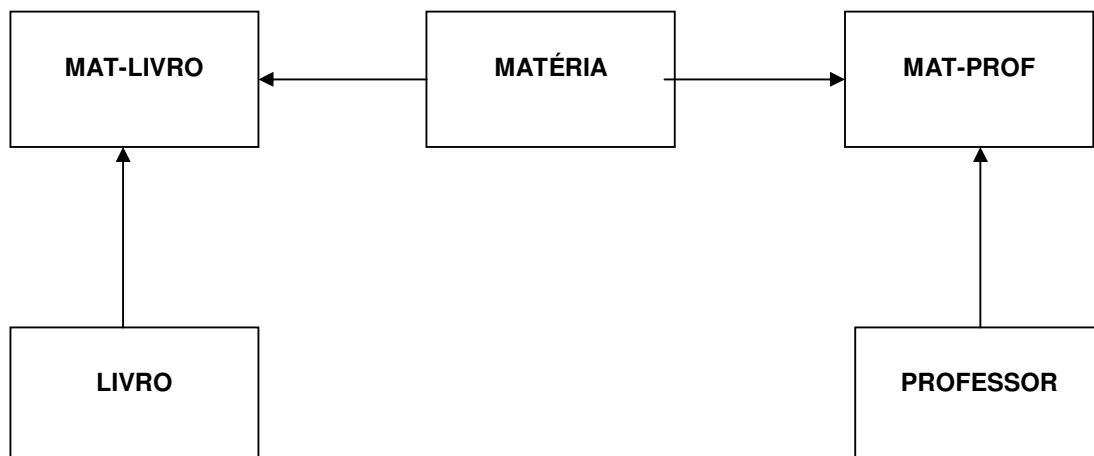
CHAVE - COD-LIVRO

Para este caso, um melhor desenho separaria as dependências multivaloradas conforme a figura a seguir:



Ou seja, as estruturas LIVRO e PROFESSOR são independentes, de modo que matéria parece um vetor que é listado para os valores de LIVRO e PROFESSOR.

O diagrama estrutura após a normalização ficaria o seguinte:



Quinta Forma Normal (5FN) ou Forma Normal Projeção/Junção (FNPJ)

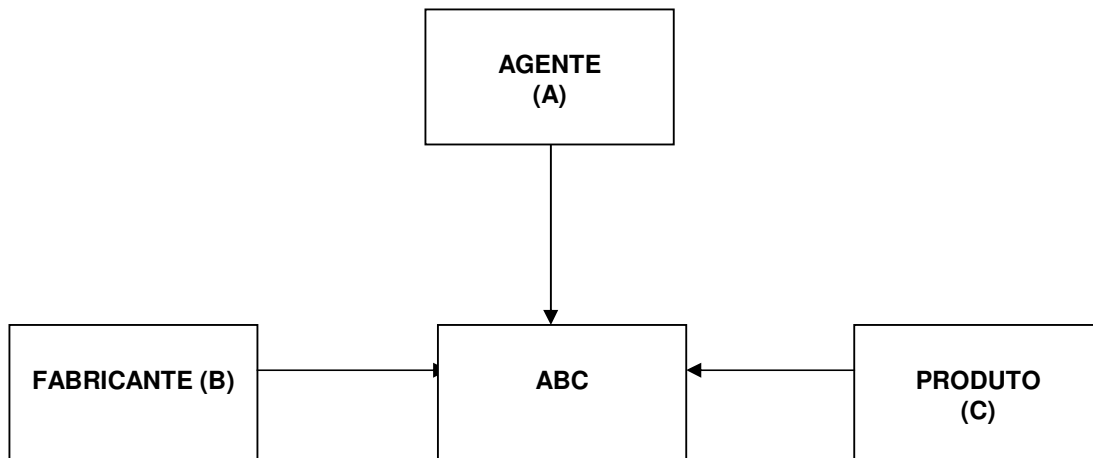
Esta regra também se aplica apenas a tabelas que representam relacionamentos ternários, sem atributos não-chave. É dito que uma tabela está na 5FN se um relacionamento triplo puder ser decomposto em 3 tabelas de relacionamentos binários (a Chave Primária é um par) sem que isto gere dados incorretos quando os mesmos forem combinados novamente em uma tabela de relacionamento triplo. Esta garantia de consistência exige que uma premissa seja sempre verdadeira: toda vez que um dado A se relaciona com outros dois (B e C), e estes outros também, por sua vez (gerando 3 relações binárias; (A,B), (A,C) e (B,C)), então é verdade que ocorre na realidade os 3 dados combinados (A,B,C).

Um exemplo é pensar em A como sendo um agente de vendas, B como sendo uma empresa fabricante e C um produto. A premissa seria: se A representa a empresa B, A é especialista na venda do produto C e B produz o produto C, então é verdade que A vende o produto C para a empresa B. Esta forma normal elimina o relacionamento ternário e a redundância de pares de valores relacionados, como por exemplo, a redundância da afirmação de que o agente A representa a empresa B. Ou seja, uma relação estará na 5FN se não puder ser mais decomposta sem perda de informação.

Dada a estrutura a seguir:

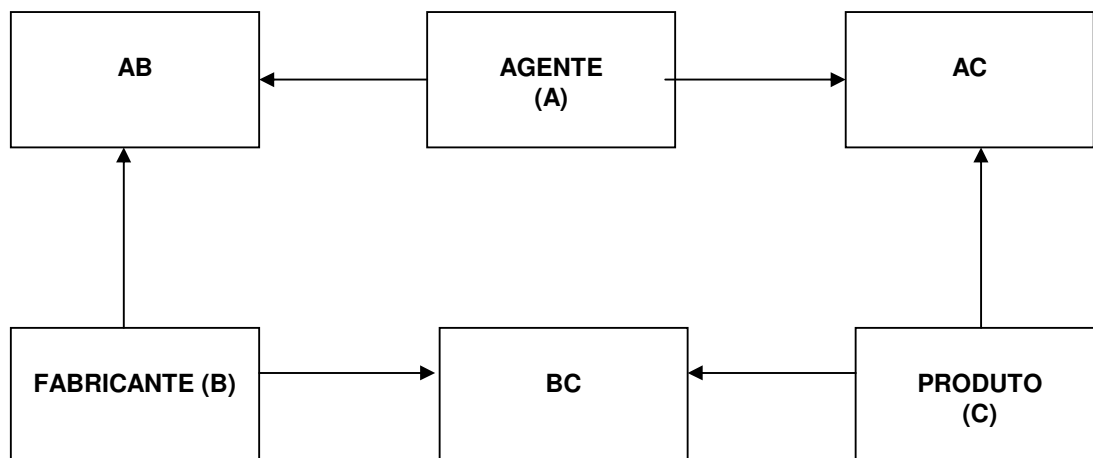
ABC	
CodAg	(PK)
CodFab	(PK)
CodProd	(PK)

Ou seja, as estruturas estão relacionadas conforme o DED abaixo:

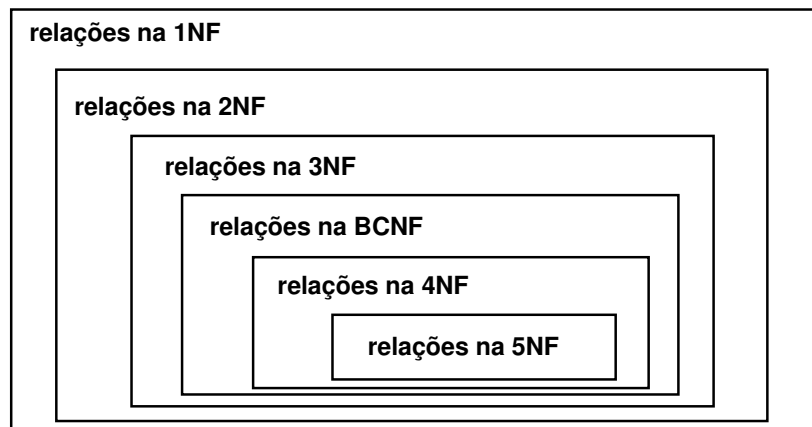


Se a estrutura ABC não puder ser decomposta em três estruturas AB, AC, BC sem perdas, então a estrutura não está na 5FN e a normalização é essa decomposição.

Ou seja, o resultado dessa normalização gerará as seguintes estruturas:



Resumidamente: a seqüência de normalizações é feita conforme o diagrama a seguir:



Fonte: Prof. FÁBIO. Apresentação PPT - Normalização: Introdução – Formas Normais, UFSC/INE

Desnormalização ⁽¹⁾

- *Nem sempre a normalização (eliminação de redundância) deve ser levada ao limite!*
- *Pode interessar armazenar informação redundante para acelerar as consultas:*
 - dados calculados: total de uma fatura (soma dos valores linha a linha)
 - dados duplicados: data de uma fatura repetida nas linhas da fatura, para acelerar pesquisas por artigo e data
- relações não decompostas, para evitar junções nas consultas, quando estas são muito mais freqüentes do que as atualizações
- *Pode interessar armazenar relações não decompostas, para acelerar a verificação de restrições de integridade*
- ficar pela 3NF quando a passagem à BCNF não preserva dependências funcionais, pois estas só se verificam na junção!

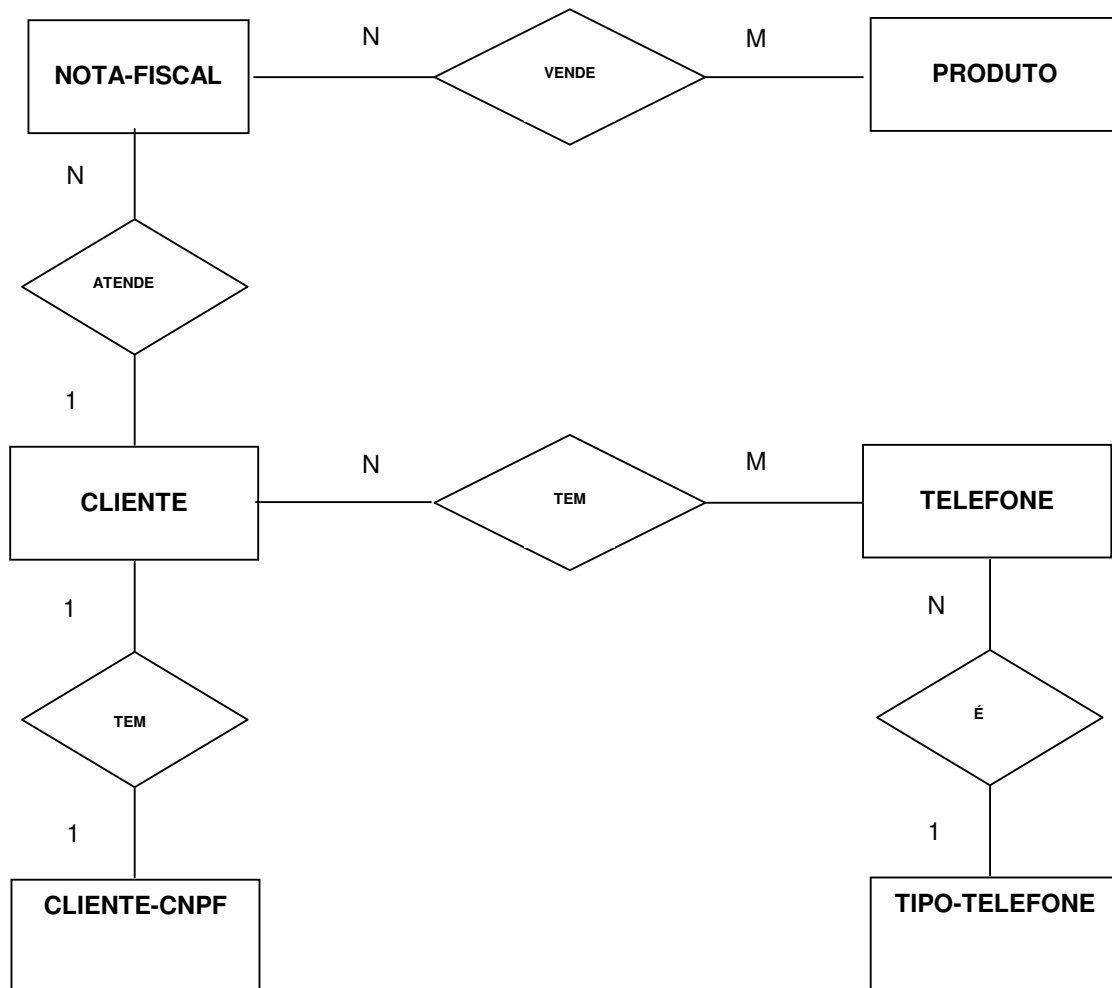
¹ DAVID, Gabriel & FÁRIA, João Pascoal. Apresentação PPT: Projeto de bases de dados relacionais - Normalização, Faculdade de Engenharia da Universidade do Porto, 2002.

2.3.4. Obtenção do MER a partir das Estruturas de Dados

Para ser possível a construção do MER é necessário que os elementos do mesmo sejam conhecidos, ou seja, entidades, relacionamentos, cardinalidades e atributos. Para identificar cada um desses elementos a partir das estruturas de dados mapeadas pode-se seguir o roteiro a seguir:

- 1) As entidades são todas aquelas estruturas cujas chaves primárias sejam compostas por atributos próprios da estruturas. Portanto, estruturas cujas chaves primárias sejam compostas somente por um atributo representam entidades. Estruturas cujas chaves primárias são compostas e que contém chaves estrangeiras em sua composição, somente são consideradas entidades se pelo menos um desses atributos for próprio da estrutura.
- 2) Os relacionamentos N:M são todas as estruturas de dados que tem chaves primárias compostas somente de chaves estrangeiras. Os relacionamentos 1:N são demonstrados nas estruturas de dados através de chaves estrangeiras, onde o N indicará que a entidade de onde se originou a chave estrangeira e 1 indicará a entidade que contém a chave estrangeira.
- 3) Os atributos das entidades e relacionamentos são todos os atributos das estruturas que representam essas entidades e relacionamentos, exceto as chaves estrangeiras.

Obter o MER a partir do Mapeamento das Estruturas de Dados produzido por Normalização no Exemplo 1.



ATIVIDADE

Uma empresa de eventos deseja fazer o controle das inscrições dos participantes de um congresso em palestras e, para tanto, indicou a ficha de inscrição, apresentada a seguir. Usando esse documento com base para as informações que o cliente precisa ter, obtenha por normalização:

- a) O mapeamento das estruturas de dados. Ou seja, apresentar a relação de estruturas, indicando os seus atributos, chave primária e chaves estrangeiras;
- b) O Diagrama de Acesso Imediato a Dados (DAID);
- c) Diagrama de Estruturas de Dados (DED) que represente o problema;
- d) O Modelo Entidade- Relacionamento (MER) correspondente ao problema.

FICHA DE INSCRIÇÃO NAS PALESTRAS
Número de Inscrição: _____

Participante
Nome: _____
E-mail: _____ Data de Nascimento: ____/____/____ Estado: _____
Endereço: _____ Estado: _____ CEP: _____

Escolaridade
Curso: _____ Ano Término: _____
Tipo de Graduação: _____
Instituição: _____ Cidade: _____ Estado: _____
Curso: _____ Ano Término: _____
Tipo de Graduação: _____
Instituição: _____ Cidade: _____ Estado: _____

Palestras
Data: ____/____/____ Hora: ____:____ Sala: _____ Título da Palestra: _____
Palestrante: _____
Hora: ____:____ Sala: _____ Título da Palestra: _____
Palestrante: _____

Para ver a solução consulte o documento [Exemplo – Projeto Lógico \(apend-b\)](#).