

4.4.2. Normalização

O processo de **normalização** pode ser visto como o processo no qual são eliminados esquemas de relações (tabelas) não satisfatórios, decompondo-os, através da separação de seus atributos em esquemas de relações menos complexas, mas que satisfaçam as propriedades desejadas.

O processo de normalização como foi proposto inicialmente por Ted Codd conduz um esquema de relação através de um bateria de testes para certificar se o mesmo está na **1ª, 2ª e 3ª Formas Normais**. Estas três formas normais partem do princípio de que um conjunto de dependências funcionais é dado para cada relação, e que cada relação tem uma chave primária especificada. Por isto são chamadas de Formas Normais Baseadas em Chaves Primárias.

Quando definimos cuidadosamente uma Diagrama ER, identificando todas as entidades corretamente, os esquemas de relação gerados do diagrama ER não precisam de muito mais normalização. Entretanto, pode haver dependências funcionais entre os atributos de uma entidade. Por exemplo: Suponha que uma entidade **funcionário** tenha atributos *numero_depto* e *endereco_depto* e que exista uma dependência funcional *numero_depto* → *endereco_depto*. Poderíamos então normalizar a relação gerada de funcionário.

A maioria dos exemplos dessas dependências surge de um projeto ruim de diagrama ER. Nesse exemplo, se tivéssemos projetado o diagrama ER corretamente, teríamos criado uma entidade *depto* com o atributo *endereco_depto* e uma relação entre **funcionário** e **depto**. Da mesma forma, uma relação envolvendo mais de duas entidades pode não estar em uma forma normal desejável. Como a maioria dos relacionamentos é binária, esses casos são relativamente raros. (Na verdade, algumas variantes do diagrama ER realmente dificultam ou impossibilitam especificar relações não binárias).

As dependências funcionais podem ajudar a detectar um mau projeto ER. Se as relações geradas não estiverem na forma normal desejada, o problema pode de ser corrigido no diagrama E-R. Ou seja, a normalização pode ser deixada por conta da intuição do projetista durante a modelagem ER, e pode ser feita formalmente nas relações geradas do modelo ER.

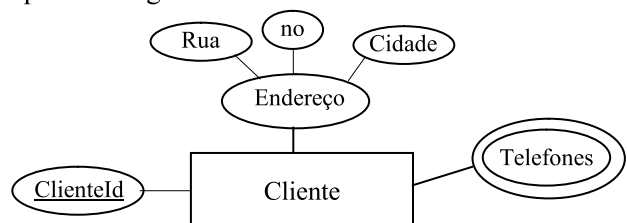
Inicialmente Ted Codd propôs três formas normais, mais tarde uma 3FN mais sólida chamada de forma normal de Boyce-Codd (FNBC) foi proposta. Todas estas formas normais são baseadas nas dependências funcionais entre atributos de uma relação. Posteriormente, foram propostos uma **quarta forma normal (4FN)** e uma **quinta forma normal (5FN)**, baseadas nos conceitos de dependências multivaloradas e dependências de junção respectivamente.

4.4.2.1. 1ª Forma Normal

A **1ª Forma Normal** prega que todos os atributos de uma tabela devem ser **atômicos** (indivisíveis), ou seja, não são permitidos atributos multivalorados, atributos compostos ou atributos multivalorados compostos. Leve em consideração o esquema a seguir:

- **CLIENTE**

1. ClienteId
2. { Telefone }
3. Endereço: (Rua, Número, Cidade)



gerando a tabela resultante:

<i>Cliente</i>	<u>ClienteId</u>	Telefone 1	Endereco		
		Telefone n	Rua	No	Cidade

sendo que a mesma não está na 1ª Forma Normal pois seus atributos não são atômicos. Para que a tabela acima fique na 1ª Forma Normal temos que eliminar os atributos não atômicos, gerando as seguintes tabelas como resultado:

<i>Cliente</i>	<u>ClienteId</u>	Rua	Número	Cidade
----------------	------------------	-----	--------	--------

<i>Cliente_Telefone</i>	<u>ClienteId</u>	<u>Telefone</u>
-------------------------	------------------	-----------------

4.4.2.2. 2ª Forma Normal

A 2ª Forma Normal prega o conceito da **dependência funcional total**. Uma dependência funcional $X \rightarrow Y$ é **total** se removemos um atributo **A** qualquer do componente **X** e desta forma, a dependência funcional deixa de existir. A dependência funcional $X \rightarrow Y$ é uma **dependência funcional parcial** se existir um atributo **A** qualquer do componente **X** que pode ser removido e a dependência funcional $X \rightarrow Y$ não deixa de existir.

Veja a dependência funcional 3 do item 4.4.1. **Dependência Funcional:**

{ EmpregadoId , ProjetoId } \rightarrow Horas (2FN)

é uma dependência funcional total, pois se removermos o atributo **EmpregadoId** ou o atributo **ProjetoId**, a dependência funcional deixa de existir.

Uma tabela **T** está na 2ª Forma Normal se estiver na 1ª Forma Normal e todo atributo que não compõem a chave primária **C** for totalmente funcionalmente dependente da chave primária **C**. Se uma tabela não está na 2ª Forma Normal à mesma pode ser normalizada gerando outras tabelas cujos atributos que não façam parte da chave primária sejam totalmente funcionalmente dependente da mesma, ficando a tabela na 2ª Forma Normal.

4.4.2.3. 3ª Forma Normal

A 3ª Forma Normal prega o conceito de **dependência transitiva**. Uma dependência funcional $X \rightarrow Y$ em uma tabela **T** é uma dependência transitiva se existir um conjunto de atributos **Z** que não é um subconjunto de chaves de **T** e as dependências $X \rightarrow Z$, $Z \rightarrow Y$, são válidas. Considere a seguinte tabela como exemplo:

<i>Empregado</i>	<u>EmpregadoId</u>	Nome	DeptoId	NomeDepto	GerenteId
------------------	--------------------	------	---------	-----------	-----------

onde temos a seguinte dependência transitiva:

Empregado \rightarrow { Nome , DeptoId}

DeptoId \rightarrow { Nome, GerenteId}

Outro Exemplo:

<i>NotaFiscal</i>	<u>NumeroNF</u>	Total	ClienteId	NomeCliente	Item	ProdutoId	NomeProduto	QtdeVendida
-------------------	-----------------	-------	-----------	-------------	------	-----------	-------------	-------------

onde temos a seguinte dependência transitiva:

NumeroNF \rightarrow { ClienteId, Total } (1FN)

ClienteId \rightarrow { Nome } (1FN)

{NumeroNF, Item} \rightarrow {ProdutoId, QtdeVendida} (2FN)

ProdutoId \rightarrow { Nome} (3FN)

Uma tabela está na 3ª Forma Normal se estiver na 2ª Forma Normal e não houver dependência transitiva entre atributos não chave.

4.4.3. Resumo das Formas

A Tabela 1 abaixo informalmente resume as três formas normais em chaves primárias, os testes utilizados em cada caso e a normalização ou “remédio” correspondente para se alcançar a forma normal.

Forma Normal	Teste	Solução(Normalização)
Primeira(1FN)	A relação não deve ter qualquer atributo não atômico nem relações agrupadas(multivalores).	Forme novas relações para cada atributo não atômico ou relação agrupada(multivalorada).
Segunda(2FN)	Para relações nas quais a chave primária contém múltiplos atributos, nenhum atributo não chave deve ser funcionalmente dependente de uma parte da chave primária.	Decomponha e monte uma relação para cada chave parcial com seu(s) atributo(s) dependente(s). Certifique-se de manter uma relação com a chave primária original e quaisquer atributos que sejam completamente dependentes dela em termos funcionais.
Terceira(3FN)	A relação não deve ter um atributo não chave funcionalmente determinado por um outro atributo não chave (ou por um conjunto de atributos não chave). Ou seja, não deve haver dependências transitivas de um atributo não chave na chave primária.	Decomponha e monte uma relação que inclua o(s) atributo(s) não chave que funcionalmente determine(m) outros atributos não chave.

Tabela 1 - Resumo das Formas Normais

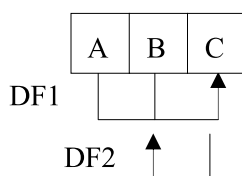
4.4.4. Outras Formas

4.4.4.1. Forma Normal de Boyce-Codd

A forma normal de Boyce-Codd(FNBC) foi proposta como uma forma mais simples da 3FN, mas foi considerada mais restrita do que ela, porque toda relação na FNBC também está na 3FN; entretanto, uma relação na 3FN não está necessariamente na FNBC.

Um esquema de relação R está na FNBC se, sempre que uma dependência funcional não-trivial $X \rightarrow A$ se mantiver em R, X for uma superchave de R. A única diferença entre as definições da FNBC e da 3FN é que a condição da 3FN que diz que permite que A seja principal, está ausente da FNBC.

Na prática, a maioria dos esquemas de relação que estão na 3FN também estão na FNBC. Somente se $X \rightarrow A$ se mantiver em um esquema de relação R, não sendo X uma superchave e sendo A um atributo principal, R estará na 3FN mas não na FNBC.



DF1 : {aluno, disciplina} → instrutor

DF2 : instrutor → disciplina

4.4.4.2. Quarta Forma Normal

Agora a definição da quarta forma normal (4FN), que é violada quando uma relação que tem dependências multivaloradas indesejáveis, e portanto pode ser utilizada para identificar e decompor relações. Um esquema de relação R está na 4FN com relação a um conjunto de dependências F (que inclui dependências funcionais e dependências multivaloradas) se, para toda dependência multivalorada não-trivial $X \twoheadrightarrow Y$ em F^+ , X for uma superchave de R .

Por exemplo, considere a relação EMPREGADO mostrada na Figura 38. Uma tupla nessa relação EMPREGADO representa o fato de que um empregado cujo nome seja NomeEmpregado trabalha no projeto cujo nome é NomeProjeto e tem um dependente cujo nome é NomeDependente. Um empregado pode trabalhar em diversos projetos e pode ter diversos dependentes, e os projetos e dependentes do empregado são independentes um do outro. Em um diagrama ER, cada qual seria representado como um atributo multivalorado ou como um tipo de entidade fraca.

Relação EMPREGADO		
NomeEmpregado	NomeProjeto	NomeDependente
Joao	X	Pedro
Joao	Y	Maria
Joao	X	Maria
Joao	Y	Pedro

Relação EMPREGADO_PROJETOS	
NomeEmpregado	NomeProjeto
Joao	X
Joao	Y

Relação DEPENDENTES	
NomeEmpregado	NomeDependente
Joao	Pedro
Joao	Maria

Figura 38 - Quarta Forma Normal

A relação EMPREGADO da Figura 38 não está na 4FN porque nas DMVs não triviais $\text{NomeEmpregado} \twoheadrightarrow \text{NomeProjeto}$ e $\text{NomeEmpregado} \twoheadrightarrow \text{NomeDependente}$, NomeEmpregado não é a superchave de EMPREGADO. Decompomos EMPREGADO em EMPREGADO_PROJETOS e DEPENDENTES. Tanto EMPREGADO_PROJETOS como DEPENDENTES são DMVs triviais. Nenhuma outra DMV não trivial se mantém em EMPREGADO_PROJETOS e DEPENDENTES.

4.4.4.3. Quinta Forma Normal

Também chamada *forma normal de junção de projeto*. Um esquema de relação R está na quinta forma normal (5FN) (ou forma normal de projeção de junção [FNPJ]) com respeito a um conjunto F de dependências funcionais, multivaloradas e de junção se, para toda dependência de junção não trivial $DJ(R_1, R_2, \dots, R_n)$ em F^+ (ou seja, implicada por F), todo R_i for uma superchave de R .

Relação FORNECIMENTO		
NomeFornecedor	NomePeca	NomeProjeto
Joao	Quadrado	X
Joao	Círculo	Y
Antonio	Quadrado	Y
Wilson	Círculo	Z
Antonio	Triângulo	X
Antonio	Quadrado	X
Joao	Quadrado	Y

Relação R1	
NomeFornecedor	NomePeca
Joao	Quadrado
Joao	Círculo
Antonio	Quadrado
Wilson	Círculo
Antonio	Triângulo

Relação R2	
NomeFornecedor	NomeProjeto
Joao	X
Joao	Y
Antonio	Y
Wilson	Z
Antonio	X

Relação R3	
NomePeca	NomeProjeto
Quadrado	X
Círculo	Y
Quadrado	Y
Círculo	Z
Triângulo	X

Figura 39 - Quinta Forma Normal

Para exemplo de uma DJ, considere a relação toda-chave FORNECIMENTO da Figura 39. Suponha que sempre se mantenha a seguinte restrição adicional: sempre que um fornecedor f fornece a peça p , e um projeto j utiliza a peça p , e o fornecedor f fornece pelo menos uma peça para o projeto j , então o fornecedor f também estará fornecendo a peça p para o projeto j . Essa restrição pode ser reafirmada de outras maneiras, e especificada uma dependência de junção DJ(R1,R2,R3) entre as três projeções R1(NomeFornecedor, NomePeca), R2(NomeFornecedor, NomeProjeto) e R3(NomePeca, NomeProjeto) de FORNECIMENTO. Se essa restrição se mantém, as tuplas abaixo da linha pontilhada na Figura 39 deve existir em qualquer estado válido (legal) da relação FORNECIMENTO que também contém as tuplas acima da linha pontilhada.

A

Figura 39 mostra como a relação FORNECIMENTO *com a dependência de junção* é decomposta em três relações R1, R2 e R3, que estão cada qual na 5FN. Note que aplicar a JUNÇÃO NATURAL a quaisquer duas dessas relações produz tuplas inválidas, mas com uma JUNÇÃO NATURAL aplicada as três conjuntamente não o fazem.

Na prática, descobrir DJs em banco de dados com centenas de atributos só é possível com um elevado grau de intuição sobre os dados por parte do projetista. Portanto, a prática atual de projetos de banco de dados dedica pouca atenção a elas.