

Formação Desenvolvedor Moderno Módulo: Lógica de Programação

Capítulo: Normalização

Exercícios

https://devsuperior.com.br

1

Exercício 1:

Normalize a seguinte relação:

Matricula

cod aluno	cod turma	cod_discipina	nome_disciplina	nome_aluno	cod_local_nasc	nome_local_nasc
111	001	1	Lógica de programação	Alex	94	Uberlândia/MG
111	001	2	Banco de dados	Alex	94	Uberlândia/MG
222	002	3	Java Script	Ari	88	Indaiatuba/SP
222	002	4	POO	Ari	88	Indaiatuba/SP
333	001	2	Banco de dados	Ana	90	São Paulo/SP
333	001	4	POO	Ana	90	São Paulo/SP
444	002	1	Lógica de programação	Pedro	75	Brasília/DF

Verificação 1FN

A relação não pode possuir atributos compostos ou multivalorados

cod_aluno	cod_turma	cod_discipina	nome_disciplina	nome_aluno	cod_local_nasc	nome_local_nasc
111	001	1	Lógica de programação	Alex	94	Uberlândia/MG
111	001	2	Banco de dados	Alex	94	Uberlândia/MG
222	002	3	Java Script	Ari	88	Indaiatuba/SP
222	002	4	P00	Ari	88	Indaiatuba/SP
333	001	2	Banco de dados	Ana	90	São Paulo/SP
333	001	4	POO	Ana	90	São Paulo/SP
444	002	1	Lógica de programação	Pedro	75	Brasília/DF

3

3

Exercício 1:

Relações resultantes após aplicar os passos 1FN

Matricula

cod_aluno	cod_turma	cod_discipina	nome_disciplina	nome_aluno	cod_local_nasc	cidade	UF
111	001	1	Lógica de programação	Alex	94	Uberlândia	MG
111	001	2	Banco de dados	Alex	94	Uberlândia	MG
222	002	3	Java Script	Ari	88	Indaiatuba	SP
222	002	4	POO	Ari	88	Indaiatuba	SP
333	001	2	Banco de dados	Ana	90	São Paulo	SP
333	001	4	POO	Ana	90	São Paulo	SP
444	002	1	Lógica de programação	Pedro	75	Brasília	DF

Verificação 2FN

Para relações onde a chave primária é composta, não pode haver dependência funcional parcial em relação a chave.

DF Funcionais

cod_aluno	cod_turma	cod_discipina	nome_disciplina	nome_aluno	cod_local_nasc	cidade	UF
111	001	1	Lógica de programação	Alex	94	Uberlândia	MG
111	001	2	Banco de dados	Alex	94	Uberlândia	MG
222	002	3	Java Script	Ari	88	Indaiatuba	SP
222	002	4	POO	Ari	88	Indaiatuba	SP
333	001	2	Banco de dados	Ana	90	São Paulo	SP
333	001	4	POO	Ana	90	São Paulo	SP
444	002	1	Lógica de programação	Pedro	75	Brasília	DF

5

5

Exercício 1:

Verificação 2FN

Para relações onde a chave primária é composta, não pode haver dependência funcional parcial em relação a chave.

cod_turma	cod_discipina	nome_disciplina	nome_aluno	cod_local_nasc	cidade	UF
001	1	Lógica de programação	Alex	94	Uberlândia	MG
001	2	Banco de dados	Alex	94	Uberlândia	MG
002	3	Java Script	Ari	88	Indaiatuba	SP
002	4	POO	Ari	88	Indaiatuba	SP
001	2	Banco de dados	Ana	90	São Paulo	SP
001	4	POO	Ana	90	São Paulo	SP
002	1	Lógica de programação	Pedro	75	Brasília	DF
	001 001 002 002 001	001 1 1 001 2 002 3 002 4 001 2 001 4	001 1 Lógica de programação 001 2 Banco de dados 002 3 Java Script 002 4 POO 001 2 Banco de dados 001 4 POO	001 1 Lógica de programação Alex 001 2 Banco de dados Alex 002 3 Java Script Ari 002 4 POO Ari 001 2 Banco de dados Ana 001 4 POO Ana	001 1 Lógica de programação Alex 94 001 2 Banco de dados Alex 94 002 3 Java Script Ari 88 002 4 POO Ari 88 001 2 Banco de dados Ana 90 001 4 POO Ana 90	001 1 Lógica de programação Alex 94 Uberlândia 001 2 Banco de dados Alex 94 Uberlândia 002 3 Java Script Ari 88 Indaiatuba 002 4 POO Ari 88 Indaiatuba 001 2 Banco de dados Ana 90 São Paulo 001 4 POO Ana 90 São Paulo

(cod_aluno, cod_turma) -> cod_disciplina

c

Verificação 2FN

Para relações onde a chave primária é composta, não pode haver dependência funcional parcial em relação a chave.

	cod_aluno	cod_turma	cod_discipina	nome_disciplina	nome_aluno	cod_local_nasc	cidade	UF
	111	001	1	Lógica de programação	Alex	94	Uberlândia	MG
	111	001	2	Banco de dados	Alex	94	Uberlândia	MG
	222	002	3	Java Script	Ari	88	Indaiatuba	SP
	222	002	4	POO	Ari	88	Indaiatuba	SP
	333	001	2	Banco de dados	Ana	90	São Paulo	SP
	333	001	4	POO	Ana	90	São Paulo	SP
	444	002	1	Lógica de programação	Pedro	75	Brasília	DF
-			^					

(cod_aluno, cod_turma) -> cod_disciplina

cod_aluno -> nome_aluno

cod_aluno -> cod_local_nasc

7

7

Exercício 1:

Verificação 2FN

Para relações onde a chave primária é composta, não pode haver dependência funcional parcial em relação a chave.

cod_aluno	cod_turma	cod_discipina	nome_disciplina	nome_aluno	cod_local_nasc	cidade	UF
111	001	1	Lógica de programação	Alex	94	Uberlândia	MG
111	001	2	Banco de dados	Alex	94	Uberlândia	MG
222	002	3	Java Script	Ari	88	Indaiatuba	SP
222	002	4	P00	Ari	88	Indaiatuba	SP
333	001	2	Banco de dados	Ana	90	São Paulo	SP
333	001	4	P00	Ana	90	São Paulo	SP
444	002	1	Lógica de programação	Pedro	75	Brasília	DF
			1	1			

 $(cod_aluno, cod_turma) \rightarrow cod_disciplina$

cod_aluno -> nome_aluno

cod_aluno -> cod_local_nasc, cidade, UF

Relações resultantes após aplicar os passos 2FN

Aluno

cod_aluno	nome_aluno	cod_local_nasc	cidade	UF	
111	Alex	94	Uberlândia	MG	
222	Ari	88	Indaiatuba	SP	
333	Ana	90	São Paulo	SP	
444	Pedro	75	Brasília	DF	
A					

Matricula

cod_aluno	cod_turma	cod_discipina	nome_disciplina
111	001	1	Lógica de programação
111	001	2	Banco de dados
222	002	3	Java Script
222	002	4	POO
333	001	2	Banco de dados
333	001	4	POO
444	002	1	Lógica de programação

9

9

Exercício 1:

Verificação 3FN

A relação não pode ter atributos não chave que dependem funcionalmente de outros atributos não chave.

Aluno

cod_aluno	nome_aluno	cod_local_nasc	cidade	UF
111	Alex	94	Uberlândia	MG
222	Ari	88	Indaiatuba	SP
333	Ana	90	São Paulo	SP
444	Pedro	75	Brasília	DF
		^		

Matricula

cod_aluno	cod turma	cod_discipina	nome_disciplina
111	001	1	Lógica de programação
111	001	2	Banco de dados
222	002	3	Java Script
222	002	4	POO
333	001	2	Banco de dados
333	001	4	POO
444	002	1	Lógica de programação

10

cod_aluno -> cod_local_nasc

Verificação 3FN

A relação não pode ter atributos não chave que dependem funcionalmente de outros atributos não chave.

cod_aluno -> cod_local_nasc

Aluno

cod_aluno	nome_aluno	cod_local_nasc	cidade	UF
111	Alex	94	Uberlândia	MG
222	Ari	88	Indaiatuba	SP
333	Ana	90	São Paulo	SP
444	Pedro	75	Brasília	DF
		<u> </u>	1	

Dependência transitiva

cod_aluno -> cod_local_nasc -> cidade, UF

Matricula

cod_aluno	cod_turma	cod_discipina	nome_disciplina
111	001	1	Lógica de programação
111	001	2	Banco de dados
222	002	3	Java Script
222	002	4	POO
333	001	2	Banco de dados
333	001	4	POO
444	002	1	Lógica de programação

11

11

Exercício 1:

Verificação 3FN

A relação não pode ter atributos não chave que dependem funcionalmente de outros atributos não chave.

Aluno

cod_aluno	nome_aluno	cod_local_nasc	
111	Alex	94	
222	Ari	Ari 88	
333	Ana	90	
444	Pedro	75	

Local

cod_local	cidade	UF
94	Uberlândia	MG
88	Indaiatuba	SP
90	São Paulo	SP
75	Brasília	DF

Matricula

cod_aluno	cod turma	cod_discipina	nome_disciplina
111	001	1	Lógica de programação
111	001	2	Banco de dados
222	002	3	Java Script
222	002	4	POO
333	001	2	Banco de dados
333	001	4	POO
444	002	1	Lógica de programação

Verificação 3FN

A relação não pode ter atributos não chave que dependem funcionalmente de outros atributos não chave.

(cod_aluno, cod_turma) -> cod_disciplina

Aluno

cod_aluno	nome_aluno	cod_local_nasc
111	Alex	94
222	Ari	88
333	Ana	90
444	Pedro	75

Local

cod_local	cidade	UF
94	Uberlândia	MG
88	Indaiatuba	SP
90	São Paulo	SP
75	Brasília	DF

Matricula

cod_aluno	cod_turma	cod_discipina	nome_disciplina
111	001	1	Lógica de programação
111	001	2	Banco de dados
222	002	3	Java Script
222	002	4	POO
333	001	2	Banco de dados
333	001	4	POO
444	002	1	Lógica de programação

13

13

Exercício 1:

Verificação 3FN

A relação não pode ter atributos não chave que dependem funcionalmente de outros atributos não chave.

(cod_aluno, cod_turma) -> cod_disciplina cod_disciplina -> nome_disciplina

Aluno

cod_aluno	nome_aluno	cod_local_nasc
111	Alex	94
222	Ari	88
333	Ana	90
444	Pedro	75

Local

cod_local	cidade	UF
94	Uberlândia	MG
88	Indaiatuba	SP
90	São Paulo	SP
75	Brasília	DF

Matricula

cod_aluno	cod_turma	cod_discipina	nome_disciplina
111	001	1	Lógica de programação
111	001	2	Banco de dados
222	002	3	Java Script
222	002	4	POO
333	001	2	Banco de dados
333	001	4	POO
444	002	1	Lógica de programação
		D	ependência transitiva

1./

Verificação 3FN

A relação não pode ter atributos não chave que dependem funcionalmente de outros atributos não chave.

Aluno

cod_aluno	nome_aluno	cod_local_nasc	
111	Alex	94	
222	Ari	88	
333	Ana	90	
444	Pedro	75	

Local

cod_local	cidade	UF
94	Uberlândia	MG
88	Indaiatuba	SP
90	São Paulo	SP
75	Brasília	DF

Matricula

cod_aluno	cod_turma	cod_discipina
111	001	1
111	001	2
222	002	3
222	002	4
333	001	2
333	001	4
444	002	1

Disciplina

cod_discipina	nome_disciplina
1	Lógica de programação
2	Banco de dados
3	Java Script
4	POO

15

15

Resultado final do exercício 1 (parte 1: especificação textual do esquema)

Local (cod_local, cidade, UF)

Disciplina (cod_disciplina, nome_disciplina)

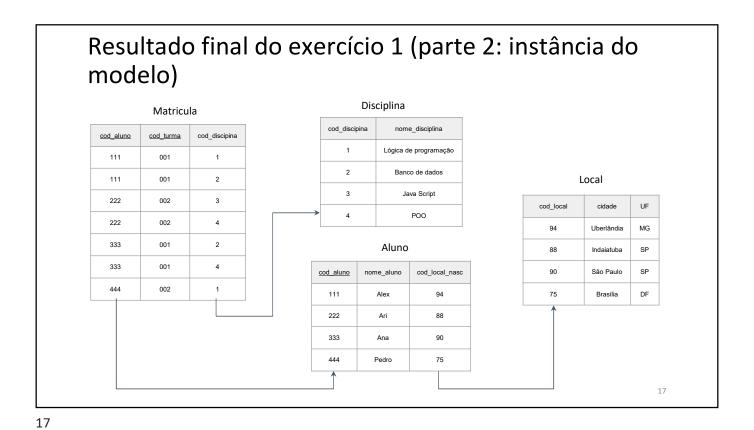
Aluno (<u>cod_aluno</u>, nome_aluno, cod_local_nasc)

cod_local_nasc referencia Local

Matricula (<u>cod_aluno</u>, <u>cod_turma</u>, cod_disciplina)

cod_aluno referencia Aluno

cod_disciplina referencia Disciplina



Normalize a seguinte relação:

Pedido (<u>nro-pedido</u>, data, <u>nro-peca</u>, descricao, qtd_comprada, preco_comprado)

Pedido

nro-pedido	data	nro-peca	descricao	qtd_comprada	preco_cotado
1031	28/03/2022	001	parafuso	20	10.00
1031	28/03/2022	002	conectores	5	25.00
1031	28/03/2022	003	engrenagem	3	30.00
1035	28/03/2022	005	porca	10	5.00
1015	29/03/2022	001	parafuso	100	50.00
1015	29/03/2022	004	conector USB	2	40.00

Relação já está na 1FN

Verificação 1FN

A relação não pode possuir atributos compostos ou multivalorados

nro-pedido	data	nro-peca	descricao	qtd_comprada	preco_cotado
1031	28/03/2022	001	parafuso	20	10.00
1031	28/03/2022	002	conectores	5	25.00
1031	28/03/2022	003	engrenagem	3	30.00
1035	28/03/2022	005	porca	10	5.00
1015	29/03/2022	001	parafuso	100	50.00
1015	29/03/2022	004	conector USB	2	40.00

19

19

Exercício 2:

Verificação 2FN

Para relações onde a chave primária é composta, não pode haver dependência funcional parcial em relação a chave.

nro-pedido	data	nro-peca	descricao	qtd_comprada	preco_cotado
1031	28/03/2022	001	parafuso	20	10.00
1031	28/03/2022	002	conectores	5	25.00
1031	28/03/2022	003	engrenagem	3	30.00
1035	28/03/2022	005	porca	10	5.00
1040	29/03/2022	001	parafuso	100	50.00
1040	29/03/2022	004	conector USB	2	40.00

Verificação 2FN

Para relações onde a chave primária é composta, não pode haver dependência funcional parcial em relação a chave.

nro-pedido -> data

data	nro-peca	descricao	qtd_comprada	preco_cotado
28/03/2022	001	parafuso	20	10.00
28/03/2022	002	conectores	5	25.00
28/03/2022	003	engrenagem	3	30.00
28/03/2022	005	porca	10	5.00
29/03/2022	001	parafuso	100	50.00
29/03/2022	004	conector USB	2	40.00
	28/03/2022 28/03/2022 28/03/2022 28/03/2022 29/03/2022	28/03/2022 001 28/03/2022 002 28/03/2022 003 28/03/2022 005 29/03/2022 001	28/03/2022 001 parafuso 28/03/2022 002 conectores 28/03/2022 003 engrenagem 28/03/2022 005 porca 29/03/2022 001 parafuso	28/03/2022 001 parafuso 20 28/03/2022 002 conectores 5 28/03/2022 003 engrenagem 3 28/03/2022 005 porca 10 29/03/2022 001 parafuso 100

21

21

Exercício 2:

Verificação 2FN

Para relações onde a chave primária é composta, não pode haver dependência funcional parcial em relação a chave.

nro-pedido -> data

nro-pedido	data	nro-peca	descricao	qtd_comprada	preco_cotado
1031	28/03/2022	001	parafuso	20	10.00
1031	28/03/2022	002	conectores	5	25.00
1031	28/03/2022	003	engrenagem	3	30.00
1035	28/03/2022	005	porca	10	5.00
1040	29/03/2022	001	parafuso	100	50.00
1040	29/03/2022	004	conector USB	2	40.00

Verificação 2FN

Para relações onde a chave primária é composta, não pode haver dependência funcional parcial em relação a chave.

nro-pedido -> data

nro-peca -> descricao

nro-pedido	data	nro-peca	descricao	qtd_comprada	preco_cotado
1031	28/03/2022	001	parafuso	20	10.00
1031	28/03/2022	002	conectores	5	25.00
1031	28/03/2022	003	engrenagem	3	30.00
1035	28/03/2022	005	porca	10	5.00
1040	29/03/2022	001	parafuso	100	50.00
1040	29/03/2022	004	conector USB	2	40.00
	<u> </u>		<u> </u>	1	1

23

23

Exercício 2:

Verificação 2FN

Para relações onde a chave primária é composta, não pode haver dependência funcional parcial em relação a chave.

nro-pedido -> data

nro-peca -> descricao

nro-pedido	data	nro-peca	descricao	qtd_comprada	preco_cotado
1031	28/03/2022	001	parafuso	20	10.00
1031	28/03/2022	002	conectores	5	25.00
1031	28/03/2022	003	engrenagem	3	30.00
1035	28/03/2022	005	porca	10	5.00
1040	29/03/2022	001	parafuso	100	50.00
1040	29/03/2022	004	conector USB	2	40.00
				1	

Verificação 2FN

Para relações onde a chave primária é composta, não pode haver dependência funcional parcial em relação a chave.

nro-pedido -> data

nro-peca -> descricao

nro-pedido	data	nro-peca	descricao	qtd_comprada	preco_cotado
1031	28/03/2022	001	parafuso	20	10.00
1031	28/03/2022	002	conectores	5	25.00
1031	28/03/2022	003	engrenagem	3	30.00
1035	28/03/2022	005	porca	10	5.00
1040	29/03/2022	001	parafuso	100	50.00
1040	29/03/2022	004	conector USB	2	40.00
	†		<u> </u>		1

25

25

Exercício 2:

Verificação 2FN

Para relações onde a chave primária é composta, não pode haver dependência funcional parcial em relação a chave.

nro-pedido	data	nro-peca	descricao	qtd_comprada	preco_cotado
1031	28/03/2022	001	parafuso	20	10.00
1031	28/03/2022	002	conectores	5	25.00
1031	28/03/2022	003	engrenagem	3	30.00
1035	28/03/2022	005	porca	10	5.00
1040	29/03/2022	001	parafuso	100	50.00
1040	29/03/2022	004	conector USB	2	40.00
				1	1

nro-pedido -> data

nro-peca -> descricao

Resultado final do exercício 2 (parte 1: especificação textual do esquema)

Pedido (nro-pedido, data)

Peca (nro-peca, descricao)

Pedido_peca (nro-peca, qtd_comprada, preco_cotado) nro-pedido referencia Pedido nro-peca referencia Peca

27

27

Resultado final do exercício 2 (parte 2: instância do modelo)

Pedido

28/03/2022

28/03/2022

29/03/2022

nro-pedido

1031

1035 1040

Peca

nro-peca	descricao
001	parafuso
002	conectores
003	engrenagem
004	conector USB
005	porca
1	'

Pedido_peca

nro-pedido	nro-peca	qtd_comprada	preco_cotado
1031	001	20	10.00
1031	002	5	25.00
1031	003	3	30.00
1035	005	10	5.00
1040	001	100	50.00
1040	004	2	40.00
		-	

Normalize a seguinte relação:

Cliente (nro-cli, nome, end_entrega)

Cliente

nro-cli	nome	end_entrega
124	João dos Santos	{ Rua 10, 1024 Rua 24, 1356 }
311	José Ferreira Neves	{ Rua 46, 1344 Rua 98, 4456 }

29

29

Exercício 3:

Normalize a seguinte relação:

Não está na 1FN: Presença de atributo multivalorado.

Verificação 1FN

A relação não pode possuir **atributos compostos** ou **multivalorados**

Cliente

nro-cli	nome	end_entrega
124	João dos Santos	{ Rua 10, 1024 Rua 24, 1356 }
311	José Ferreira Neves	{ Rua 46, 1344 Rua 98, 4456 }

Resultado final do exercício 3 (parte 1: especificação textual do esquema)

Cliente (<u>nro-cli</u>, nome)

Cliente-entrega (<u>nro-cli</u>, <u>end entrega</u>)

nro-cli referencia Cliente

31

31

Resultado final do exercício 3 (parte 2: instância do modelo)

Cliente

nro-cli nome

124 João dos Santos

311 José Ferreira Neves

entrega

nro-cli	end entrega
124	Rua 10, 1024
124	Rua 24, 1356
311	Rua 46, 1344
311	Rua 98, 4456

Cliente-