



Formação Desenvolvedor Moderno

Módulo: Lógica de Programação

Capítulo: Normalização

Exercícios

<https://devsuperior.com.br>

1

Exercício 1:

Normalize a seguinte relação:

Matricula

<u>cod_aluno</u>	<u>cod_turma</u>	<u>cod_disciplina</u>	<u>nome_disciplina</u>	<u>nome_aluno</u>	<u>cod_local_nasc</u>	<u>nome_local_nasc</u>
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

2

2

Exercício 1:

Verificação 1FN

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

<u>cod_aluno</u>	<u>cod_turma</u>	<u>cod_disciplina</u>	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

3

3

Exercício 1:

Relações resultantes após aplicar os passos 1FN

Matricula

<u>cod_aluno</u>	<u>cod_turma</u>	<u>cod_disciplina</u>	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

4

4

Exercício 1:

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_disciplina	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_aluno	cod_turma	cod_disciplina	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



6

6

Exercício 1:

Verificação 2FN

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

<u>cod_aluno</u>	<u>cod_turma</u>	<u>cod_disciplina</u>	<u>nome_disciplina</u>	<u>nome_aluno</u>	<u>cod_local_nasc</u>	<u>cidade</u>	<u>UF</u>
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.

<u>cod_aluno</u>	<u>cod_turma</u>	<u>cod_disciplina</u>	<u>nome_disciplina</u>	<u>nome_aluno</u>	<u>cod_local_nasc</u>	<u>cidade</u>	<u>UF</u>
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, cidade, UF

8

8

Exercício 1:

Relações resultantes após aplicar os passos 2FN

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

cod_aluno	cod_turma	cod_disciplina	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.

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

cod_aluno -> cod_local_nasc

cod_aluno	cod_turma	cod_disciplina	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

10

Exercício 1:

Verificação 3FN

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

Aluno

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

Matricula

<u>cod_aluno</u>	<u>cod_turma</u>	<u>cod_disciplina</u>	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

$\text{cod_aluno} \rightarrow \text{cod_local_nasc}$

$\text{cod_aluno} \rightarrow \text{cod_local_nasc} \rightarrow \text{cidade, UF}$

Dependência transitiva

11

11

Exercício 1:

Verificação 3FN

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

Aluno

<u>cod_aluno</u>	nome_aluno	<u>cod_local_nasc</u>
111	Alex	94
222	Ari	88
333	Ana	90
444	Pedro	75

Matricula

<u>cod_aluno</u>	<u>cod_turma</u>	<u>cod_disciplina</u>	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

Local

<u>cod_local</u>	cidade	UF
94	Uberlândia	MG
88	Indaiatuba	SP
90	São Paulo	SP
75	Brasília	DF

12

12

Exercício 1:

Verificação 3FN

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

Aluno

<u>cod_aluno</u>	nome_aluno	cod_local_nasc
111	Alex	94
222	Ari	88
333	Ana	90
444	Pedro	75

Local

<u>cod_local</u>	cidade	UF
94	Uberlândia	MG
88	Indaiatuba	SP
90	São Paulo	SP
75	Brasília	DF

(cod_aluno, cod_turma) -> cod_disciplina

Matricula

<u>cod_aluno</u>	<u>cod_turma</u>	<u>cod_disciplina</u>	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.

Aluno

<u>cod_aluno</u>	nome_aluno	cod_local_nasc
111	Alex	94
222	Ari	88
333	Ana	90
444	Pedro	75

Local

<u>cod_local</u>	cidade	UF
94	Uberlândia	MG
88	Indaiatuba	SP
90	São Paulo	SP
75	Brasília	DF

(cod_aluno, cod_turma) -> cod_disciplina

cod_disciplina -> nome_disciplina

Matricula

<u>cod_aluno</u>	<u>cod_turma</u>	<u>cod_disciplina</u>	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

Dependência transitiva

14

14

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	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_disciplina
111	001	1
111	001	2
222	002	3
222	002	4
333	001	2
333	001	4
444	002	1

Disciplina

cod_disciplina	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)

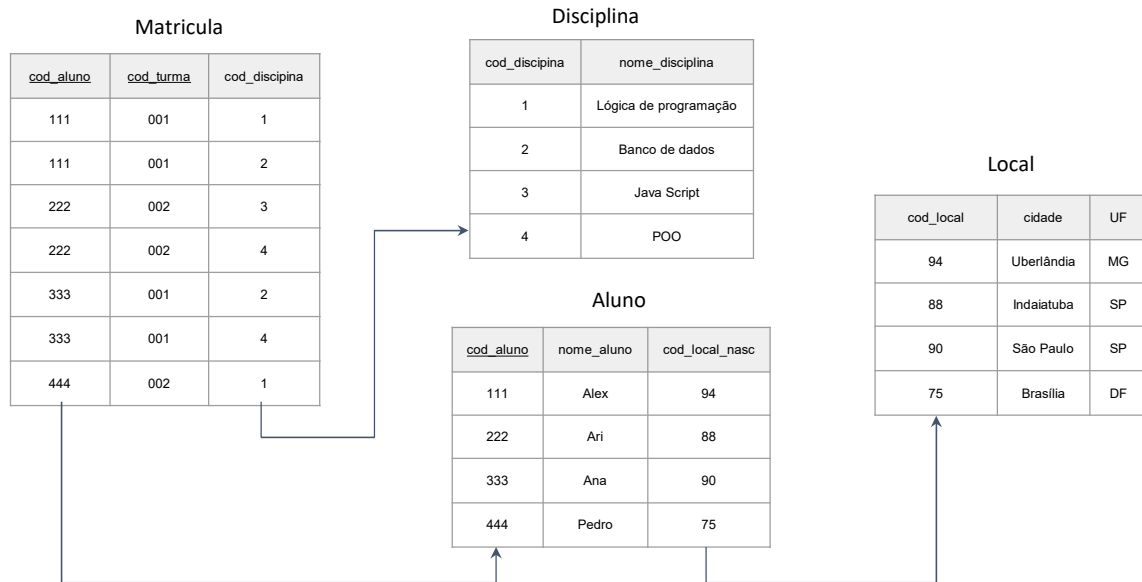
Aluno (cod_aluno, nome_aluno, cod_local_nasc)
cod_local_nasc referencia Local

Matricula (cod_aluno, cod_turma, cod_disciplina)
cod_aluno referencia Aluno
cod_disciplina referencia Disciplina

16

16

Resultado final do exercício 1 (parte 2: instância do modelo)



17

17

Exercício 2:

Normalize a seguinte relação:

Pedido (nro_pedido, data, nro_peca, descricao, qtd_comprada, preco_comprado)

Pedido

<u>nro_pedido</u>	data	<u>nro_peca</u>	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

18

18

Exercício 2:

Relação já está na 1FN

Verificação 1FN

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

<u>nro-pedido</u>	data	<u>nro-peca</u>	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.

<u>nro-pedido</u>	<u>data</u>	<u>nro-peca</u>	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

20

20

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

<u>nro-pedido</u>	<u>data</u>	<u>nro-peca</u>	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



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

<u>nro-pedido</u>	<u>data</u>	<u>nro-peca</u>	<u>descricao</u>	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



22

22

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

<u>nro-pedido</u>	<u>data</u>	<u>nro-peca</u>	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



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

<u>nro-pedido</u>	<u>data</u>	<u>nro-peca</u>	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



24

24

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

<u>nro-pedido</u>	<u>data</u>	<u>nro-peca</u>	<u>descricao</u>	<u>qtd_comprada</u>	<u>preco_cotado</u>
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

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

{nro-pedido, nro-peca} -> {qtdade_comprada,
preco_cotado}

<u>nro-pedido</u>	<u>data</u>	<u>nro-peca</u>	<u>descricao</u>	<u>qtd_comprada</u>	<u>preco_cotado</u>
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

26

26

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

Pedido (nro-pedido, data)

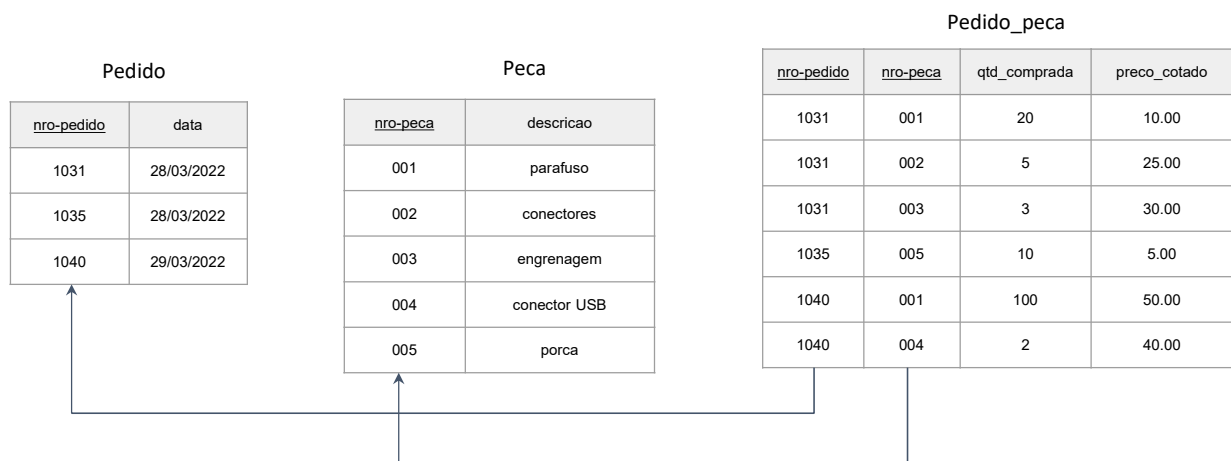
Peca (nro-peca, descricao)

Pedido_peca (nro-pedido, 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)



28

28

Exercício 3:

Normalize a seguinte relação:

Cliente (nro-cli, nome, end_entrega)

Cliente

<u>nro-cli</u>	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:

Verificação 1FN

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

Cliente

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

Não está na 1FN: Presença de atributo **multivalorado**.

30

30

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

Cliente (nro-cli, nome)

Cliente-entrega (nro-cli, end_entrega)
nro-cli referencia Cliente

31

31

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

Cliente		Cliente-entrega	
<u>nro-cli</u>	nome	<u>nro-cli</u>	<u>end_entrega</u>
124	João dos Santos	124	Rua 10, 1024
311	José Ferreira Neves	124	Rua 24, 1356
		311	Rua 46, 1344
		311	Rua 98, 4456

32

32