

Normalização no modelo relacional

Dependências funcionais

Existe uma dependência funcional $X \rightarrow Y$ entre dois conjuntos de atributos X e Y, se uma instância de valores dos atributos de X determina ou identifica univocamente uma instância de valores dos atributos de Y.

Formas Normais

- Uma relação diz-se que está na Primeira Forma Normal (1FN) quando:
 - Não contém atributos multivalor;
 - Não contém grupos repetitivos.
- Uma relação diz-se que está na Segunda Forma Normal (2FN) quando:
 - Está na Primeira Forma Normal (1FN);
 - Todos os atributos não chave dependem funcionalmente da totalidade da chave.
- Uma relação diz-se que está na Terceira Forma Normal (3FN) quando:
 - Está na Segunda Forma Normal (2FN);
 - Todos os atributos não chave não dependem funcionalmente uns dos outros.

Ficha Prática 3

Exercício 1

NOTAS (Numero_aluno, Disciplina, Semestre, Nota)

Hipótese 1: Considerando que uma mesma disciplina pode existir no 1.º e no 2.º semestre:

Exemplo:

<u>Numero_aluno</u>	<u>Disciplina</u>	<u>Semestre</u>	Nota
10123	Bases de Dados	1	15
10123	Álgebra	2	9
10123	Álgebra	1	13
11245	Bases de Dados	1	14
...			

NOTAS (Numero_aluno, Disciplina, Semestre, Nota)

1FN ✓, 2FN ✓, 3FN ✓

Hipótese 2: Considerando que uma disciplina está sempre associada a um semestre:

Exemplo:

<u>Numero_aluno</u>	<u>Disciplina</u>	Semestre	Nota
10123	Bases de Dados	1	15
10123	Álgebra	2	13
11245	Bases de Dados	1	14
...			

NOTAS (Numero_aluno, Disciplina, Semestre, Nota)

1FN ✓, 2FN ✗

Dependências funcionais existentes na tabela NOTAS:

Disciplina → Semestre

Numero_aluno, Disciplina → Nota

Assim a tabela original deve ser substituída pelas seguintes que obedecem à 2FN:

DISCIPLINAS (Disciplina, Semestre)

NOTAS (Numero_aluno, Disciplina, Nota)

Disciplina referencia DISCIPLINAS

1FN ✓, 2FN ✓, 3FN ✓

Exercício 2

SOFTWARE (SoftwareHouse, Produto, Release, ReqSys, Preco, Garantia)

Exemplo:

<u>SoftwareHouse</u>	<u>Produto</u>	<u>Release</u>	<u>ReqSys</u>	Preco	Garantia
Microsoft	Windows	7	1	80	1
Microsoft	Windows	8	1	100	1
Microsoft	Windows	10	2	120	1
Oracle	OracleServer	9	2	220	2
...					

Vamos considerar para os requisitos do sistema (ReqSys):

<u>ReqSys</u>	Processador	Memoria	Disco
1	Pentium II	128	10
2	Pentium IV	512	80
...			

Assim, temos:

REQUISITOS (ReqSys, Processador, Memoria, Disco)

SOFTWARE (SoftwareHouse, Produto, Release, ReqSys, Preco, Garantia)

ReqSys referencia REQUISITOS

1FN ✓, 2FN ✗

Dependências funcionais na tabela SOFTWARE:

SoftwareHouse → Garantia

SoftwareHouse, Produto, Release → ReqSys, Preco

REQUISITOS (...) *Nota: esta notação significa que a tabela se mantém.*

GARANTIAS (SoftwareHouse, Garantia)

SOFTWARE (SoftwareHouse, Produto, Release, ReqSys, Preco)

SoftwareHouse referencia GARANTIAS

ReqSys referencia REQUISITOS

1FN ✓, 2FN ✓, 3FN ✓

Exercício 3

PRODUTO_LOJA (id_produto, id_promocao, marca, estilo, preco)

Dependências funcionais na tabela PRODUTO_LOJA:

id_produto, id_promocao → marca, estilo, preco

id_produto → marca, estilo

Então:

PRODUTO_LOJA (id_produto, id_promocao, marca, estilo, preco)

1 FN ✓, 2 FN ✗

Pelas dependências funcionais obtidas a partir da 2FN na tabela PRODUTO_LOJA, nomeadamente:

id_produto → marca, estilo

id_produto, id_promocao → preco

Resulta nas seguintes tabelas:

PRODUTO (id_produto, marca, estilo)

PRODUTO_LOJA (id_produto, id_promocao, preco)

id_produto referencia PRODUTO

1FN ✓, 2 FN ✓, 3 FN ✓

Exercício 4a

R1 (H, I, J, K, L, M, N, O)

Dependências funcionais na tabela R1:

$H, I \rightarrow J, K, L$

$J \rightarrow M$

$K \rightarrow N$

$L \rightarrow O$

Então:

R1 (H, I, J, K, L, M, N, O)

1 FN ✓, 2 FN ✓, 3 FN ✗

Pelas dependências funcionais fornecidas, resulta nas seguintes tabelas:

R11 (J, M)

R12 (K, N)

R13 (L, O)

R1 (H, I, J, K, L)

J referencia R11

K referencia R12

L referencia R13

1 FN ✓, 2 FN ✓, 3 FN ✓

Exercício 4b

R2 (D, O, N, T, C, R, Y)

Dependências funcionais na tabela R2:

$D, O \rightarrow N, T, C, R, Y$

$D \rightarrow N$

Então:

R2 (D, O, N, T, C, R, Y)

1 FN ✓, 2 FN ✗

Pelas dependências funcionais obtidas a partir da 2FN na tabela R2, nomeadamente:

$D \rightarrow N$

$D, O \rightarrow T, C, R, Y$

Resulta nas seguintes tabelas:

R21 (D, N)

R2 (D, O, T, C, R, Y)

D referencia R21

1 FN ✓, 2 FN ✓, 3 FN ✓

Exercício 5b

APOLICE (num_apolice, data, matricula, marca_viatura, cod_cliente, nome_cliente, localidade_cliente, cod_tipo, tipo, num_pagamento, data_pagamento, valor_pagamento, estado_pagamento)

Exemplo:

NA	D	M	MV	CC	NC	LC	CT	T	NP	DP	VP	EP
1	20.01.2004	01-01-AA	X	1	AA	VR	1	TR	1	20.01.2004	50	P
1	20.01.2004	01-01-AA	X	1	AA	VR	1	TR	2	20.01.2005	45	NP
2	01.01.1998	02-02-BB	Z	2	BB	PT	2	T	3	01.01.1998	10	P
2	01.01.1998	02-02-BB	Z	2	BB	PT	2	T	4	01.01.1999	10	P
2	01.01.1998	02-02-BB	Z	2	BB	PT	2	T	5	01.01.2000	9	P
2	01.01.1998	02-02-BB	Z	2	BB	PT	2	T	6	01.01.2001	8	P

Legenda: TR – Todos os Riscos; T – Terceiros; P – Pago; NP – Não Pago.

De forma simplificada temos a seguinte tabela:

APOLICE (NA, D, M, MV, CC, NC, LC, CT, T, NP, DP, VP, EP)

1 FN ✗

Analisando o exemplo é possível definir várias linhas identificadas pelo NP referentes a cada apólices com o mesmo NA. Assim, podemos decompor a tabela APOLICE nas seguintes tabelas:

APOLICE (NA, D, M, MV, CC, NC, LC, CT, T)

LINHAS_APOLICE (NA, NP, DP, VP, EP)

NA referencia APOLICE

1 FN ✓, 2 FN ✓, 3 FN ✗

Dependências funcionais na tabela APOLICE:

M → MV

CC → NC, LC

CT → T

NA → D, M, CC, CT

Então resulta nas seguintes tabelas:

VEICULO (M, MV)

CLIENTE (CC, NC, LC)

TIPO (CT, T)

APOLICE (NA, D, ~~M~~, ~~CC~~, ~~CT~~)

M referencia VEICULO

CC referencia CLIENTE

CT referencia TIPO

LINHAS_APOLICE (...)

1 FN ✓, 2 FN ✓, 3 FN ✓