



Modelo Lógico

Normalização de Dados

Prof. Fábio Procópio



2

Relembrando...

- Na [aula passada](#), iniciamos nossas discussões sobre o **Modelo Relacional** e vimos que um banco de dados baseado nesse modelo é composto de tabelas (ou relações);
- Vimos também os conceitos de:
 1. Tabela
 2. Chave primária
 3. Chave estrangeira
 4. Chave candidata
 5. Restrições de integridade





Introdução

- A **normalização de dados** é um conjunto de passos aplicados ao projeto de BD e objetiva armazenar os dados de forma consistente além de permitir o acesso eficiente a esses dados;
- Uma tabela de um banco de dados relacional está em uma determinada forma normal se atender a algumas condições;
- Inicialmente, foram definidas 3 formas normais, porém, atualmente, outras vêm sendo utilizadas:
 - Primeira forma normal (1FN)
 - Segunda forma normal (2FN)
 - Terceira forma normal (3FN)
 - Forma normal de Boyce-Codd (FNBC)
 - Quarta forma Normal (4FN)
 - Quinta forma Normal (5FN)

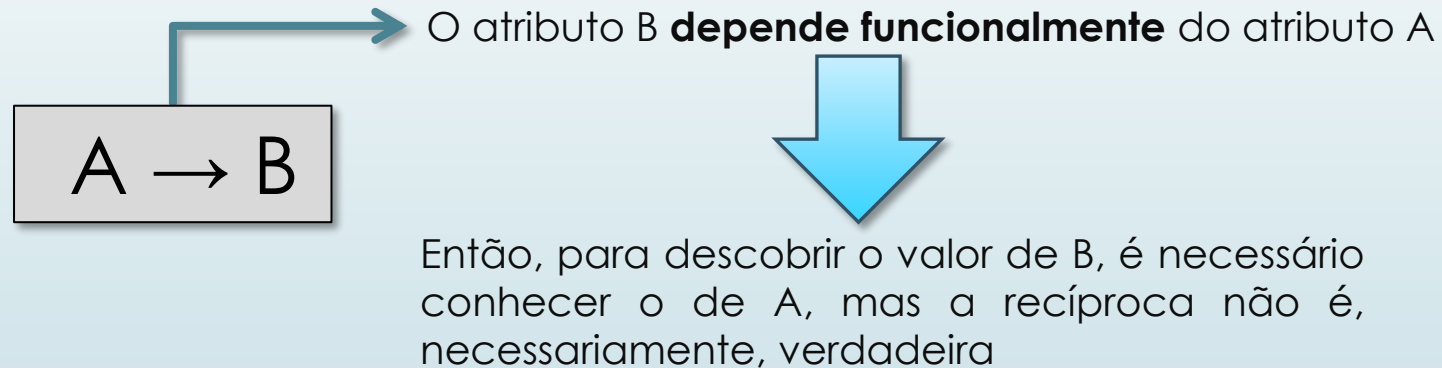


Dependência Funcional

Dependência Funcional (DF)

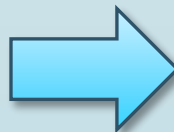
=

Relacionamento entre dois ou mais atributos de forma que o valor de um atributo identifique o valor para cada um dos outros atributos, ou seja, um atributo está relacionado a outro.



► Exemplo de uma DF com os dados de alunos:

Matrícula → Nome
Matrícula → Curso
Matrícula → Celular
Matrícula → Endereço



Matrícula → Nome, Curso, Celular, Endereço



5

Dependência Funcional Parcial – 1 de 2

Dependência Funcional Parcial (DFP)

=

Os atributos não-chave não dependem funcionalmente de toda a chave primária quando esta for composta.

- Assim, nas tabelas onde a chave primária for composta, todos os atributos não-chave devem depender de toda a chave primária;
- Quando a dependência é de parte da chave, existe uma dependência funcional parcial:

$AB \rightarrow C, D$



Imagine que C dependa funcionalmente de AB, mas D depende apenas de B. Assim, há uma **Dependência Funcional Parcial** de D em relação à AB.



6

Dependência Funcional Parcial – 2 de 2

- A tabela **BOLETIM** possui uma chave primária composta representada pelos campos **Matricula_Aluno**, **Periodo** e **CodDisciplina**;
- Percebe-se que o campo Nota depende de toda a chave primária;
- Porém, NomeDisciplina depende apenas de parte da chave primária, ou seja, do campo CodDisciplina:

Matricula_Aluno	Periodo	CodDisciplina	NomeDisciplina	Nota
123	1	8	Engenharia de Requisitos	4,0
123	1	9	Qualidade de Software	10,0
123	1	5	Engenharia de Software	7,0
123	2	8	Engenharia de Requisitos	9,0

- A tabela acima, é configurado um caso de **Dependência Funcional Parcial** pois o campo NomeDisciplina depende apenas de **parte da chave primária**.



7 Dependência Funcional Transitiva – 1 de 2

Dependência Funcional Transitiva (DFT)

=

Quando um campo não chave depende de um outro campo que também não é chave primária.

- É IMPORTANTE destacar a diferença entre as dependências funcional Parcial e Transitiva em uma tabela:
 - Parcial: pelo menos um atributo não chave depende de parte da chave primária;
 - Transitiva: pelo menos um atributo não chave depende de outro atributo também não chave primária.



8 Dependência Funcional Transitiva – 2 de 2

- A tabela **FUNCIONARIO** é composta pelos campos **Matricula** (PK), NomeFuncionario, CodCargo, NomeCargo e SalarioCargo;
- Percebe-se que o campo **Matricula** determina apenas os atributos NomeFuncionario e CodCargo;
- Enquanto que CodCargo (que **não é PK**) determina NomeCargo e SalarioCargo;

Matricula	NomeFuncionario	CodCargo	NomeCargo	SalarioCargo
1	Ary	1	Professor	R\$ 7.500,00
2	Tatiana	2	Advogado	R\$ 6.900,00
3	Ana	3	Secretária	R\$ 1.550,00
4	Luis	4	Analista de Sistemas	R\$ 8.000,00
5	Rodrigo	1	Professor	R\$ 7.500,00

- A tabela acima, configura um caso de **Dependência Funcional Transitiva** pois os campos NomeCargo e SalarioCargo dependem funcionalmente de CodCargo, que não é uma PK.



Primeira Forma Normal (1FN) – 1 de 2

Primeira Forma Normal

=

Uma tabela está na 1FN quando todos os seus atributos possuem apenas valores atômicos e monovalorados

➤ Para deixar uma tabela na 1FN deve ser executado o seguinte algoritmo:

se (existirem atributos compostos) **então**

dividir esses atributos em outros atômicos

fimse

se (existirem atributos multivalorados) **então**

colocá-los em uma outra tabela e relacioná-la com a tabela original

fimse



10

Primeira Forma Normal (1FN) – 2 de 2

- Baseado na definição da 1FN, percebe-se que a tabela **PESSOA** não está na 1FN porque Telefone é um atributo **multivalorado** e Endereco um atributo **composto**:

Codigo	Nome	Telefone	Endereco
1	Ary	(34) 3821-0000 (34) 9979-0000 (34) 9964-0000	Av. Getúlio Vargas, 1000, apto 201 – Centro – Patos de Minas-MG
2	Tatiana	(34) 3822-0000 (34) 9976-0000	Av. Brasil, 966 – Centro – Belo Horizonte-MG
3	Ana	(11) 3184-0000	Rua Minas Gerais, 100 – Bairro Brasil – Recife-PE
4	Luis	(31) 3257-0000	Praça da Liberdade, 27 – Bairro Esperança – São Paulo-SP

Multivalorado



1FN

Composto

Codigo_tel	Nrotel	Codigo
1	(34) 3821-0000	1
2	(34) 9979-0000	1
3	(34) 9964-0000	1
4	(34) 3822-0000	2
5	(34) 9976-0000	2
6	(11) 3184-0000	3
7	(31) 3257-0000	4

Codigo	Nome	Tipo Logradouro	Logradouro	Nro	Comp	Bairro	Cidade	UF
1	Ary	Avenida	Getúlio Vargas	1000	Apto 201	Centro	Patos de Minas	MG
2	Tatiana	Avenida	Brasil	966		Centro	Belo Horizonte	MG
3	Ana	Rua	Minas Gerais	100		Brasil	Recife	PE
4	Luis	Praça	Liberdade	27		Esperança	São Paulo	SP



Segunda Forma Normal (2FN) – 1 de 2

Segunda Forma Normal

=

Uma tabela está na 2FN se estiver na 1FN e não possuir dependência funcional parcial.

➤ Para deixar uma tabela na 2FN deve ser executado o seguinte algoritmo:

*/*existe uma DF Parcial*/*

se (há atributos que não dependem integralmente da chave primária) **então**
retirá-los todos da tabela e criar uma nova tabela;

fimse

12

Segunda Forma Normal (2FN) – 2 de 2

- A tabela **VENDA** tem como PK composta os atributos **Nro** e **Codp** e percebe-se que ela está na 1FN porque não há atributos compostos e nem multivalorados;
- Contudo, ela não está na 2FN porque há uma DF parcial, ou seja, existem atributos que dependem parcialmente da PK (Codp → Nomep, Vunit):

PK **DFP** **DF**

Nro	Codp	Nomep	Vunit	Qdade	Vtot
1	1	Sabão em pó	R\$ 5,50	2	R\$ 11,00
2	2	Sabonete	R\$ 1,10	5	R\$ 5,50
3	1	Sabão em pó	R\$ 5,50	3	R\$ 16,50
3	2	Sabonete	R\$ 1,10	2	R\$ 2,20



Nro	Codp	Qdade	Vtot
1	1	2	R\$ 11,0
2	2	5	R\$ 5,50

Codp	NomeP	Vunit
1	Sabão em pó	R\$ 5,50
2	Sabonete	R\$ 1,10



Terceira Forma Normal (3FN) – 1 de 3

Terceira Forma Normal

=

Uma tabela está na 3FN se estiver na 2FN e não possuir dependência funcional transitiva.

➤ Para deixar uma tabela na 3FN deve ser executado o seguinte algoritmo:

se (há atributos que dependem de campos não-chave) **então**

retirá-los todos da tabela e criar uma nova tabela;

fimse



Terceira Forma Normal (3FN) – 2 de 3

- A tabela **FUNCIONARIO** tem como chave primária o atributo **Matricula** e percebe-se:
 - está na 1FN porque **só existem** atributos **atômicos** e **monovalorados**;
 - está na 2FN porque **não há atributos que dependam parcialmente** da PK, uma vez que ela não é composta;
 - porém, **não está na 3FN** porque há uma DF transitiva: CodCargo → NomeCargo, SalarioCargo

Matricula	NomeFuncionario	CodCargo	NomeCargo	SalarioCargo
1	Ary	1	Professor	R\$ 7.500,00
2	Tatiana	2	Advogado	R\$ 6.900,00
3	Ana	3	Secretária	R\$ 1.550,00
4	Luis	4	Analista de Sistemas	R\$ 8.000,00
5	Rodrigo	1	Professor	R\$ 7.500,00



15

Terceira Forma Normal (3FN) – 3 de 3

Matricula	NomeFuncionario	CodCargo	NomeCargo	SalarioCargo
1	Ary	1	Professor	R\$ 7.500,00
2	Tatiana	2	Advogado	R\$ 6.900,00
3	Ana	3	Secretária	R\$ 1.550,00
4	Luis	4	Analista de Sistemas	R\$ 8.000,00
5	Rodrigo	1	Professor	R\$ 7.500,00



Matricula	NomeFuncionario	CodCargo
1	Ary	1
2	Tatiana	2
3	Ana	3
4	João	4

CodCargo	NomeCargo	SalarioCargo
1	Professor	R\$ 7.500,00
2	Advogado	R\$ 6.900,00
3	Secretária	R\$ 1.550,00
4	Analista de Sistemas	R\$ 8.000,00



Resumo das Formas Normais

FN	Condição	Solução (normalização)
1	A relação não deve conter os atributos não atômicos ou as relações aninhadas (só conter atributos atômicos)	Criar uma nova relação para cada atributo não-atômico ou para cada relação aninhada
2	Para as relações que possuam chaves primárias com vários atributos, nenhum atributo externo à chave deve depender parcialmente da chave primária	Decompor e montar uma nova relação para cada chave parcial com seu(s) atributo(s) dependente(s). Assegurar-se de que manteve a relação com a chave primária original e com todos os atributos que possuam dependência funcional total com ela
3	As relações não devem ter atributos que não pertençam a uma chave, funcionalmente determinados por outro atributo que também não pertença a uma chave (ou por um conjunto de atributos não-chave). Isto é, não deve haver dependência transitiva entre um atributo não-chave e uma chave primária	Decompor e montar uma relação que contenha o(s) atributo(s) não-chave que determina(m) funcionalmente o(s) outro(s) atributo(s)



Exercícios de Fixação – 1 de 2

Nos esquemas abaixo, os campos acompanhados pelo símbolo de arroba identificam a chave primária e os campos envolvidos por chaves {...} são multivalorados. Nenhum esquema está normalizado, isto é, eles estão na 0FN.

De acordo com as técnicas de normalização, para cada um dos esquemas, apresente a sequência de normalização desde a 1FN até a 3FN. Ao fim, desenhe o Diagrama de Entidade-Relacionamento (DER) correspondente a cada um dos esquemas.

- 1) **Empregado** = (@MatriculaEmpregado, NomeEmpregado, CodigoDepartamento, NomeDepartamento, CodigoGerente, NomeGerente, {NúmeroProjeto, HorasTrabalhadasNoProjeto, NomeProjeto, DataInicioProjeto})
- 2) **Ordem_Compra** = (@NumeroOrdemCompra, DataEmissao, CodigoFornecedor, NomeFornecedor, EnderecoFornecedor, ValorTotalOrdemCompra, {CoProduto, DescricaoProduto, ValorUnitario, QtdeComprada})



Exercícios de Fixação – 2 de 2

- 3) **Matricula** = (@MatriculaAluno, NomeAluno, FoneAluno, AnoIngresso, {CodigoDisciplina, NomeDisciplina}, CoCurso, NomeCurso, DataMatricula)
- 4) **Paciente** = (@MatriculaPaciente, NomePaciente, CodigoPlanoSaude, NomePlanoSaude, {DataEntradaQuarto, NumeroQuarto, DescricaoQuarto, NumeroComodos})



Principais Referências

- 1) JÚNIOR, Ary. **Normalização de Dados**. Revista SQL Magazine, ed. 47.
- 2) HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. 6ª edição. Bookman, Porto Alegre, 2009.
- 3) Blog do Luis. **Normalização de dados e as formas normais**. Acessado em: 28 jan. 2019.
Disponível em: <http://www.luis.blog.br/normalizacao-de-dados-e-as-formas-normais.aspx>.
- 4) GONÇALVES, Mauri. **Modelagem de Dados – Final (Normalização)**. Acessado em: 28 jan. 2019.
Disponível em: http://imasters.uol.com.br/artigo/7020/bancodedados/modelagem_de_dados_final_normalizacao/.
- 5) ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 4ª edição. Pearson. São Paulo, 2005.