Qualidade do projeto relacional - Normalização

Fábio Luiz Leite Júnior

Introdução

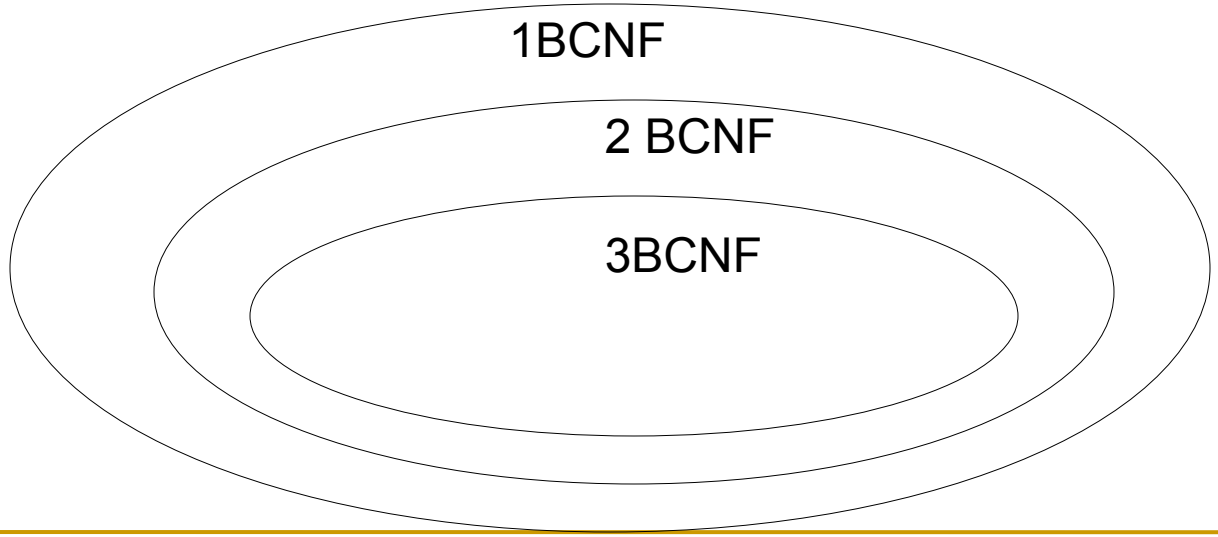
- Na concepção de um projeto de banco de dados podem ocorrer anomalias (redundância), podem ser do seguintes tipos:
  - Repetição desnecessária da informação em várias tuplas
  - Anomalias de atualização - o fato de ser necessário atualizar uma informação em um dado local e ser obrigado a atualizar a mesma informação em outro local
  - Anomalias de deleção - mesma situação acima para o caso de deleção
- Decomposição de tabelas

Exemplo de motivação

Titulo	ano	tipo	estudio	endEstudio
star wars	1970	cor	Lucas film	rancho skywalker
matrix	2001	cor	waner	hollywood
xmen	2003	cor	warner	hollywood
superman	2006	cor	warner	hollywood
superman	1970	p/b	disney	florida

Boyce-Codd Normal Form

- O objetivo da decomposição é evitar que tais tipos de anomalias ocorram
- Existe um conjunto de regras para garantir que essas anomalias não aocorram
- Relações simplificadas e estruturas regulares
- Aumento da integridade dos dados



Primeira forma normal

- Todos os atributos devem conter apenas valores atômicos

Matrícula	Nome	Cod Cargo	NomeCargo	CodProj	DataFim	Horas
120	João	1	Programador	01	17/07/95	37
120	João	1	Programador	08	12/01/96	12
121	Hélio	1	Programador	01	17/07/95	45
121	Hélio	1	Programador	08	12/01/96	21
121	Hélio	1	Programador	12	21/03/96	107
270	Gabriel	2	Analista	08	12/01/96	10
270	Gabriel	2	Analista	12	21/03/96	38
273	Silva	3	Projetista	01	17/07/95	22
274	Abraão	2	Analista	12	21/03/96	31
279	Carla	1	Programador	01	17/07/96	27
279	Carla	1	Programador	08	12/01/96	20
279	Carla	1	Programador	12	21/03/96	51
301	Ana	1	Programador	12	21/03/96	16
306	Manoel	3	Projetista	17	21/03/96	67

Primeira forma normal

- A chave primária para a tabela empregados é (Matrícula,CodProj)
- Vimos que um dos objetivos da normalização é reduzir a redundância de dados, porém com a tabela acima aumentamos a redundância ?!?!
- Precisamos realizar outros passos de normalização para termos um bom projeto.
- A 1FN possui características indesejáveis!

Primeira forma normal

- Anomalias da 1FN
  - Inserção: não podemos inserir um empregado sem que este esteja alocado num projeto, nem um projeto sem que haja um empregado trabalhando nele (integridade de entidade).
  - Remoção: se precisarmos remover um projeto, as informações de empregados que estiverem lotados apenas naquele projeto serão perdidas.
  - Atualização: se um empregado for promovido de cargo teremos que atualizar os atributos CodCargo e NomeCargo em todas as tuplas nas quais aquele empregado está presente

- Conclusão
  - Uma tabela em 1FN não evita, porém, anomalias de inclusão, atualização, e remoção. É preciso uma normalização mais “fina” , ou outras formas formas normais.
    - Segunda Forma Normal (2FN)
    - Terceira Forma Normal (3FN)
  - Esta normalização “fina” utiliza o conceito de *dependência funcional*

Dependência funcional

- A → B, lê - se:

  A *funcionalmente determina* B

  B é *funcionalmente dependente* de A

  B é *função* de A

Para cada valor de A só existe um valor de B.

A ¬→ B, negação de A → B.

Dependência funcional

- A ou B podem ser um conjunto de atributos.

  Identidade → Nome

  Identidade → Endereço

  Identidade ¬→ Habilidade

  Nome ¬→ Identidade

  Endereço ¬→ Identidade

  Habilidade ¬→ Identidade

  Identidade → Nome, Endereço

Dependência funcional

- Idéia de normalização “fina”: agrupar numa tabela somente dois conjuntos de atributos X e Y, com X → Y.

  X é então a *chave* da tabela, e Y é *complemento da chave*.

  Consequência das definições de dependência funcional e de chave:

  se X é chave então cada valor de X é *único*, e, consequentemente, um valor de X identifica uma linha da tabela.

Segunda forma normal

- Uma tabela está na Segunda Forma Normal (2FN) se ela é 1FN e todo atributo do complemento de uma chave candidata é totalmente funcionalmente dependente daquela chave
- A, B, C => D (D é totalmente funcionalmente dependente de {A, B, C}) se para todo valor de {A, B, C} só existe um valor de D, e se D não é funcionalmente dependente de A, ou B, ou C.

Segunda forma normal

A	B	C	D
a1	b3	c1	d4
a1	b1	c1	d2
a1	b3	c1	d1

- a1 b3 c1 d1 não pode existir
- A → D ((a1,d4), (a1, d2))
- B → D
- C → D

Segunda forma normal

- Exemplo 1
  - ESPORTISTA (Identidade, Nome, Endereço, Esporte)
  - Chaves candidatas
    - Identidade → Nome
    - Identidade → Endereço
    - Identidade → Esporte
    -
    - {Nome, Endereço} → Identidade
    - {Nome, Endereço} → Esporte

Segunda forma normal

- Conclusão: O atributo Esporte deve ser retirado da relação ESPORTISTA.
  - ESPORTISTA (Identidade, Nome, Endereço)
  - PRATICA-ESPORTE (Identidade, Esporte)
  - Um atributo sublinhado faz parte da chave.
  - Atualizar o endereço de Pelé: sem anomalia.
  - Incluir uma nova habilidade de Pelé: sem anomalia

Segunda forma normal

- Exemplo 2

E #	Enome	Sexo	Idade	D #	Dnome	Opinião
E 1	João	M	25	D 1	Mat	Boa
E 1	João	M	25	D 2	Quim	Má
E 1	João	M	25	D 3	Fis	Boa
E 2	Maria	F	22	D 2	Quim	Satisf.
E 2	Maria	F	22	D 3	Fis	Satisf.
E 2	Maria	F	22	D 4	Est	Má
E 3	João	M	27	D 2	Quim	Boa
E 3	João	M	27	D3	Fis	Boa

Segunda forma normal

- {E#, D# }
  - {E#, D#} => Enome (E# → Enome)
  - {E#, D#} => Sexo (E# → Sexo)
  - {E#, D#} => Idade (E# → Idade)
  - {E#, D#} => Dnome (D# → Dnome)
  - {E#, D#} => Opinião
- {E# , Dnome)
  - {E#, Dnome} => Enome (E# → Enome)
  - {E#, Dnome} => Sexo (E# → Sexo)
  - {E#, Dnome} => Idade (E# → Idade)
  - {E#, Dnome} => D# (Dnome → D#)
  - {E#, Dnome} => Opinião

Segunda forma normal

- Conclusão: Enome, Sexo, Idade e Dnome devem ser retirados de ESTUDANTE-DISCIPLINA

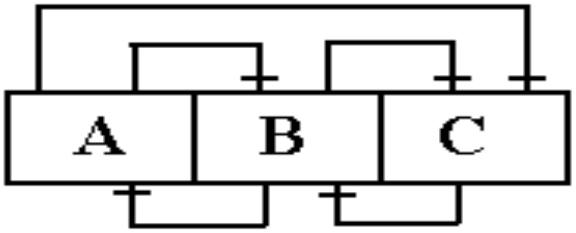
E #	Enome	Sexo	Idade
E1	João	M	25
E2	Maria	F	22
E3	João	M	27

D #	Dnome
D1	Mat
D2	Quim
D3	Fis
D4	Est

E #	D #	Opinião
E1	D1	Boa
E1	D2	Pobre
E1	D3	Boa
E2	D2	Satisfatória
E2	D3	Satisfatória
E2	D4	Pobre
E3	D2	Boa
E3	D3	Boa

Terceira Forma Normal (3FN)

- Envolve o conceito de dependência transitiva. Suponha que tenhamos uma tabela com colunas A, B e C.
- Se a coluna C é funcionalmente dependente de B e B é funcionalmente dependente de A, então C é funcionalmente dependente de A

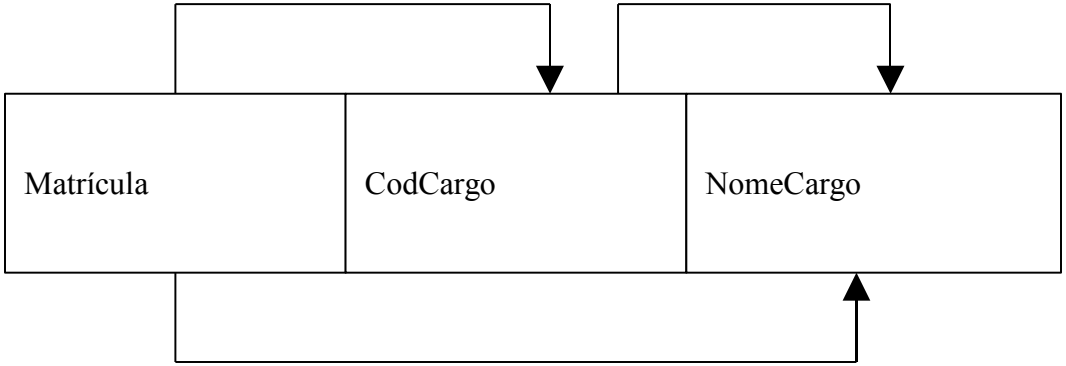


Terceira Forma Normal (3FN)

- Definição: Uma relação está em 3FN se, e somente se, estiver em 2FN e todos os atributos não-chave forem dependentes não-transitivos da chave primária.

Matrícula	Nome	CodCargo	NomeCargo
120	João	1	Programador
121	Hélio	1	Programador
270	Gabriel	2	Analista
273	Silva	3	Projetista
274	Abraão	2	Analista
279	Carla	1	Programador
301	Ana	1	Programador
306	Manuel	3	Projetista

Terceira Forma Normal (3FN)



- NomeCargo é dependente transitivo de Matrícula.
- Removendo esta dependência transitiva, obteremos,além das tabelas Projeto e Alocação, as seguintes tabelas:

Terceira Forma Normal (3FN)

Matrícula	Nome	CodCargo
120	João	1
121	Hélio	1
270	Gabriel	2
273	Silva	3
274	Abraão	2
279	Carla	1
301	Ana	1
306	Manuel	3

CodCargo	Nome
1	Programador
2	Analista
3	Projetista

"Uma relação está em 3FN se todas as colunas da tabela são funcionalmente dependentes da chave inteira e nada além da chave"

A 3FN elimina as características mais potencialmente indesejáveis dos dados que estão em 2FN ou 1FN.

Existem outros casos especiais que requerem mais níveis de normalização: Boyce-Codd, 4FN e 5FN