

Banco de Dados

Normalização: Forma Normal Boyce Codd (FNBC)

Prof. Eldane Vieira

Forma normal Boyce Codd - FNBC

- Costuma-se dizer que a FNBC é a 3FN melhorada.
- A FNBC é mais simples que a 3FN, porém mais rígida.
- A 3FN só tratava dos casos de dependência de atributos fora de qualquer chave, mas se o atributo estiver em uma chave ele não trata.
- Então, na FNBC, verificam-se as dependências funcionais que se baseiam nas chaves.

Condições para a FNBC

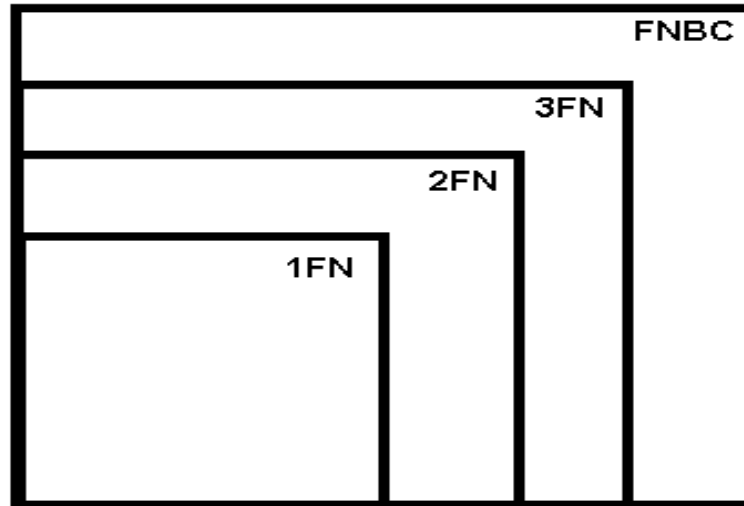
- Os casos cobertos pela FNBC são raros e somente ocorre quando três condições aparecem juntas:
 - A entidade tenha várias chaves candidatas.
 - Lembrando que chave candidata é um identificador único que poderia ser usado como chave primária.
 - As chaves sejam compostas.
 - As chaves candidatas compartilhem pelo menos um atributo.

Requisitos para FNBC

- Para uma tabela estar na FNBC é preciso atender duas condições:
 - A tabela precisa estar na 3FN;
 - Para qualquer dependência $A \rightarrow B$, A deve ser chave primária e B não.
 - Ou seja, A não pode ser um atributo não-chave e B não pode ser chave primária ou compor a chave primária.
- A normalização FNBC contribui para a escolha da chave primária da tabela, visto que ela ocorre em um cenário que existe ao menos duas chaves candidatas.

Níveis de normalização

- Atendendo a FNBC, atende-se os outros níveis de normalização



Recordando as dependências

Dependência funcional

chave primária → atributo não-chave

Dependência parcial

parte da chave primária → atributo não-chave

Viola a 2FN.

Dependência transitiva

atributo não-chave → atributo não-chave

Viola a 3FN.

Violação da FNBC

atributo não-chave —————▶ chave primária ou parte dela

*Isso não é possível!
Viola a FNBC.*

Exemplo

O Id_estudante não pode ser chave primária, porque seus valores se repetem em diferentes linhas, como ocorre com 101.

Id_Estudante	Disciplina	Professor
101	Java	P1
101	C++	P2
102	Java	P3
103	C#	P4
104	Java	P1

O campo Professor não pode ser chave primária, porque seus valores se repetem em diferentes linhas, como ocorre com P1.

O campo Disciplina não pode ser chave primária, porque seu valores se repetem em diferentes linhas, como ocorre com Java.

Outra característica a ser observada sobre Disciplina é que uma disciplina pode ser ministrada por mais de um professor, como ocorrer com os professores P1 e P3 que ministram Java, mas um professor só ministra uma disciplina.

Exemplo

Id_Estudante	Disciplina	Professor
101	Java	P1
101	C++	P2
102	Java	P3
103	C#	P4
104	Java	P1

Os campos Id_Estudante e Disciplina, juntos, podem ser chave primária, pois não existe repetição de nenhum par de valores envolvendo esses campos, e por isso podem ser usados para encontrar os demais campos da tabela, que no caso é Professor.

Logo, (Id_Estudante, Disciplina) são chaves candidatas a se tornarem chave primária, tendo a seguinte dependência:

(Id_Estudante, Disciplina) → Professor

Os campos Id_Estudante e Professor, juntos, podem ser chave primária, pois não existe repetição de nenhum par de valores envolvendo esses campos, e por isso podem ser usadas para encontrar os demais campos da tabela, que no caso é Disciplina.

Logo, (Id_Estudante, Professor) são chaves candidatas a se tornarem chave primária, tendo a seguinte dependência:

(Id_Estudante, Professor) → Disciplina

Exemplo

Id_Estudante	Disciplina	Professor
101	Java	P1
101	C++	P2
102	Java	P3
103	C#	P4
104	Java	P1

Como cada Professor só ministra uma disciplina. Então, com o identificador do professor é possível obter a disciplina que ele ministra. Tendo a seguinte dependência:

Professor → Disciplina

Esta dependência é restritiva para o problema, pois dependendo da chave primária escolhida pode gerar a situação (atributo não chave) → (atributo chave).

Exemplo

- Dependências obtidas do exemplo:

(Id_Estudante, Disciplina) → Professor

(Id_Estudante, Professor) → Disciplina

Professor → Disciplina

- Analisando as dependências, observa-se que as condições para ocorrência da FNBC foram atendidas:
 - A entidade tem várias chaves candidatas, no caso são 2 chaves.
 - As chaves candidatas são compostas, (Id_Estudante, Disciplina) e (Id_Estudante, Professor).
 - As chaves candidatas compartilhem pelo menos um atributo, que no caso é ID_Estudante.
- Outro ponto a ser observado é que se escolhermos (Id_Estudante, Disciplina) como chave primária, então a dependência (Professor → Disciplina) será do tipo (atributo não-chave → chave primária ou parte dela), isso viola a FNBC.

Exemplo

- Dependências obtidas do exemplo:

(Id_Estudante, Disciplina) → Professor

(Id_Estudante, Professor) → Disciplina

Professor → Disciplina

- Considerando a dependência (Professor → Disciplina) é preciso garantir que ela seja uma dependência funcional (chave primária → atributo não-chave).
- Para isso será criada uma tabela **Professor(Professor, Disciplina)**, considerando que o campo Professor seja a chave primária por ser um identificador único, visto que um professor só ministra uma disciplina, logo só terá uma ocorrência de cada professor.
- Dessa maneira, a dependência (Professor → Disciplina) se torna uma dependência funcional.
- Uma outra tabela que será usada é **Estudante(Id_Estudante, Professor)**, que possui chave primária composta.

Exemplo

- Com as tabelas:
 - Estudante(Id_Estudante, Professor)
 - Professor(Professor, Disciplina)
- Podemos dizer que elas satisfazem à FNBC.

Exercício

Considere a seguinte tabela *Projeto*(*idTutor*, *nomeProjeto*, *estudante*), sendo (*idTutor*, *estudante*) e (*idTutor*, *nomeProjeto*) chaves candidatas.

A tabela apresenta as seguintes dependências:

$(idTutor, nomeProjeto) \rightarrow estudante;$

$(idTutor, estudante) \rightarrow nomeProjeto$ e

$estudante \rightarrow nomeProjeto.$

Então, diga se esta tabela atende ou não à FNBC, explique sua resposta. Caso não atenda, normalize a tabela de modo que ela atenda à FNBC.

Referência

- R. Ramakrishnan e J. Gehrke, Database Management Systems, 3a Edição, McGraw-Hill, 2003.
- <https://www.youtube.com/watch?v=NNjUhvwvOrk>