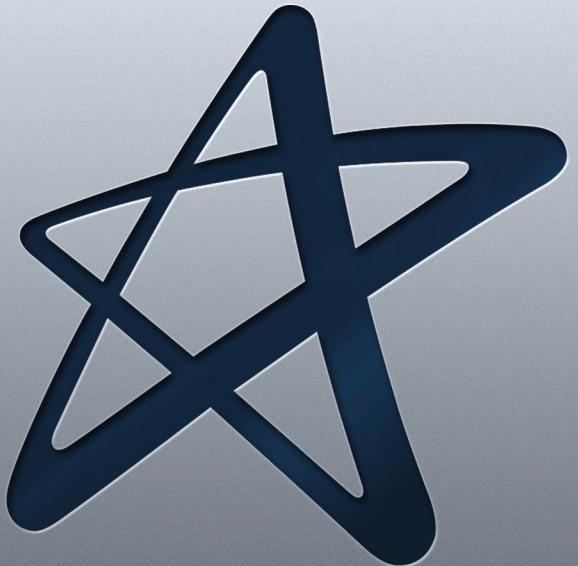


# Modelagem de Dados





# Material Teórico



**Implementando o Modelo Relacional**

**Responsável pelo Conteúdo:**

Prof. Esp. Hugo Fernandes

**Revisão Textual:**

Profa. Dra. Geovana Gentili Santos





- Normalização



### OBJETIVO DE APRENDIZADO

- Estudaremos os conceitos de Normalização do modelo relacional, suas regras de implantação e seus principais benefícios





# Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja uma maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:

Determine um horário fixo para estudar.

Mantenha o foco! Evite se distrair com as redes sociais.

Procure manter contato com seus colegas e tutores para trocar ideias! Isso amplia a aprendizagem.

Seja original! Nunca plágie trabalhos.

Aproveite as indicações de Material Complementar.

Conserve seu material e local de estudos sempre organizados.

Não se esqueça de se alimentar e se manter hidratado.

## Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como o seu “momento do estudo”.
- ✓ Procure se alimentar e se hidratar quando for estudar, lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo.
- ✓ No material de cada Unidade, há leituras indicadas. Entre elas: artigos científicos, livros, vídeos e sites para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados.
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e aprendizagem.

# Normalização

A normalização é um processo para avaliar e corrigir estruturas de tabela para minimizar redundâncias de dados, reduzindo, assim, a probabilidade de anomalias de dados. O processo de normalização envolve a atribuição de atributos a tabelas com base no conceito de determinação.

O processo de normalização deve ocorrer logo após a etapa de criação do modelo conceitual do banco de dados. Muitas vezes, após a normalização, ocorrerão atualizações no modelo conceitual.

A normalização funciona através de uma série de estágios chamados Formas normais. Os três primeiros estágios são descritos como primeira forma normal (1NF), segunda forma normal (2NF) e terceira forma normal (3NF). Do ponto de vista estrutural, podemos afirmar 2NF é melhor do que 1NF, e 3NF é melhor que 2NF.



## Importante!

Um problema óbvio com informações redundantes é que usamos mais memória do que é necessário. A redundância é um exemplo de uma anomalia que pode ocorrer em um SGBD do modelo relacional.

## A Necessidade de Normalização

Existem duas situações comuns em que os projetistas de banco de dados usam a normalização:

- Ao projetar uma nova estrutura de banco de dados com base nos requisitos de negócios dos usuários finais, o projetista de banco de dados construirá um modelo de dados usando o diagrama de entidade-relacionamento (DER). Após a conclusão do projeto inicial, o projetista pode usar a normalização para analisar as relações que existem entre os atributos dentro de cada entidade, para determinar se a estrutura pode ser melhorada através da normalização.
- Alternativamente, os projetistas de banco de dados são frequentemente solicitados a modificar estruturas de dados existentes que podem ser na forma de arquivos simples, planilhas ou estruturas de banco de dados mais antigas. Novamente, por meio de uma análise das relações entre os atributos ou campos na estrutura de dados, o projetista de banco de dados pode usar o processo de normalização para melhorar a estrutura de dados existente para criar um projeto de banco de dados apropriado.

## Anomalias

Existem três tipos de anomalias que ocorrem quando o banco de dados não é normalizado. Estas são:

- Inserção;
- Atualização; e
- Anomalia de Exclusão.

Vamos a um exemplo para entender isso! Suponha que em uma empresa desenvolvedora de *software*, as informações dos projetos executados são armazenadas em uma tabela chamada Projetos. Vejamos essa tabela abaixo:

Tabela 1

ID_Projeto	Nome_Projeto	ID_Empregado	Nome_Empregado	Cargo_Empregado	Valor_Hora	Horas_Trabalhadas
001	Manhattan	1	João da Silva	Programador Sênior	40.00	50
		2	Paulo Farias	Analista Sênior	40.00	30
		3	Carlos Alberto	Programador Sênior	40.00	50
		4	Maria Fernanda	Gerente	80.00	50

Os problemas com a tabela acima são:

1. O número do projeto destina-se a ser uma chave primária, mas contém nulos.
2. A tabela exibe redundâncias de dados.
3. As entradas de tabela permitem inconsistências de dados.
4. As redundâncias de dados produzem as seguintes anomalias:
  - a) Anomalias de Inserção: Não podemos armazenar os detalhes do Empregado até que o Projeto seja atribuído.
  - b) Anomalias de Atualização: Se o nome do projeto Manhattan precisar ser alterado, essa alteração deverá ser realizada em todos os quatro registros.
  - c) Anomalias de Exclusão: Se excluirmos o empregado 1, também iremos perder as informações sobre o Projeto.

Para superar essas anomalias, precisamos normalizar os dados. Nas próximas seções, iremos discutir sobre a normalização.

A chave primária de uma relação em um SGBD deve ser uma chave candidata, mas pode haver várias chaves candidatas para escolher. Quando se fala de normalização, é irrelevante qual chave é escolhida como chave primária.

## 1<sup>a</sup> Forma Normal (1FN)

Definimos que uma tabela está na primeira forma normal, se e somente se, todas as colunas possuem um único valor, e não existam grupos repetitivos (colunas) em uma linha ou atributos compostos. (MACHADO, 2014).

Para que uma tabela possa estar na 1FN, devemos seguir as seguintes regras:

1. Não devem existir colunas com dados repetidos ou similares;
2. Cada item de dados deve ser atômico (não possuir valores compostos);
3. Cada linha deve ser única, isto é, deve possuir uma chave primária;
4. Cada campo deve ter um nome exclusivo.



### Importante!

‘Atômico’ é o termo usado para descrever que um item de dados é único e indivisível.

Exemplo de dados repetidos:

Tabela 2

Tabela Cliente				
Id_Cliente	Nome_Cliente	Telefone1	Telefone2	Telefone3
123	João da Silva	1234-2356	8945-5689	2563-8996

Neste exemplo, o banco de dados está tentando armazenar números de telefones de contato para cada Cliente. O projetista criou três campos para manter números de telefone. Esse é um exemplo de “colunas com dados repetidos”. Os números de telefone são o mesmo tipo de dados.

Exemplo de dados “não atômicos”.

Tabela 3

Tabela Cliente		
Id_Cliente	Nome_Cliente	E-mail
123	João da Silva	joao@gmail.com; joaosilva@gmail.com

Por mais que a tabela acima possua um campo de chave primária (Id\_Cliente) e não existam dados repetidos, essa tabela não está na primeira forma normal (1FN) porque, como podemos notar, o cliente possui dois endereços de e-mail inseridos no campo “E-mail”. Desse modo, os dados inseridos neste campo não são atômicos.

Levando em consideração que existe a possibilidade de o cliente possuir mais que um (1) e-mail, a solução para esse cenário é criar uma nova entidade chamada E-mail e usar o campo chave (Id\_Cliente) como chave estrangeira para fazer a ligação entre as duas entidades. Podemos observar o resultado nas tabelas abaixo.

Tabela 4

Tabela Cliente	
Id_Cliente	Nome_Cliente
123	João da Silva
125	José Ferreira

Tabela 5

Tabela E-mail		
Id_Email	Id_Cliente	E-mail
1	123	joao@gmail.com
2	123	joaosilva@gmail.com
3	125	josefer@gmail.com

Com essa solução, não há problemas quanto a inserir mais que um (1) e-mail por cliente e confere a fácil extração de dados de e-mail, pois, além de existir apenas uma coluna para e-mail, todos os dados são atômicos.

Podemos afirmar que essas tabelas estão na primeira forma normal (1NF), dado que obedecem às seguintes regras:

- Cada tabela tem uma chave primária;
- Cada nome de campo é exclusivo;
- Os dados são atômicos;
- Não possui campos repetidos / redundantes.

## 2ª Forma Normal (2FN)

Uma tabela está na segunda forma normal (2FN) se estiver na 1FN e não possuir campos que sejam funcionalmente dependentes de parte da chave primária (MACHADO, 2014).

As regras para a segunda forma normal são:

1. A tabela deve estar já na primeira forma normal (1FN);
2. Todos os atributos não-chave devem depender da chave primária completa, ou seja, não contenham dependência parcial.



### Importante!

Dependência parcial ocorre quando uma coluna depende apenas de parte de uma chave primária composta (HEUSER, 2010).

A razão dessa regra é garantir que não haja dados redundantes sendo armazenados no banco de dados.

Vamos a um exemplo. Considere as seguintes tabelas.

Tabela 6

Tabela Projeto	
Id_Projeto	Nome_Projeto
103	Manhattan
104	Houston

Tabela 7

Tabela Empregado					
Id_Projeto	ID_Empregado	Nome_Empregado	Cargo_Empregado	Valor_Hora	Horas_Trabalhadas
103	1	João da Silva	Programador Junior	30.00	50
103	2	Maria Fernanda	Gerente	80.00	50
104	1	João da Silva	Programador Junior	30.00	20
104	3	Paulo Farias	Analista Sênior	40.00	10
104	4	Gustavo Fontes	Analista Sênior	40.00	20

No cenário acima, podemos perceber que a tabela **Projeto** está na segunda forma normal, pois todos os seus atributos dependem exclusivamente de sua chave primária. Contudo, a tabela **Empregado** não, posto que os atributos “Nome\_Empregado”, “Cargo\_Empregado” e “Valor\_Hora” dependem somente do campo chave “Id\_Empregado”. São atributos que não estão ligados diretamente do campo chave “Id\_Projeto”, ou seja, são atributos que não dependem da entidade **Projeto**. Por outro lado, o atributo “Horas\_Trabalhadas” depende exclusivamente da chave composta da Tabela (Id\_Projeto e Id\_Empregado). A informação guardada nesse atributo permite saber quantas horas o empregado trabalhou no projeto específico.

Para que a tabela **Empregado** passe para a segunda forma normal (2FN), devemos dividir a tabela em duas novas tabelas, criando, assim, uma nova tabela chamada **Projeto\_Horas\_Trabalhadas**. Podemos conferir o resultado abaixo:

Tabela 8

Tabela Projeto	
ID_Projeto	Nome_Projeto
103	Manhattan
104	Houston

TABELA PROJETO_HORAS_TRABALHADAS		
ID_Projeto	ID_Empregado	Horas_Trabalhadas
103	1	50
103	2	50
104	1	20
104	3	10

Podemos afirmar que as tabelas acima estão na segunda forma normal (2FN) porque todas as tabelas estão na 1FN e, além disso, os atributos não-chave de todas as tabelas possuem dependência exclusiva da chave primária de suas respectivas tabelas (sendo essas chaves primárias compostas ou não).

## 3<sup>a</sup> Forma Normal (3FN)

Uma tabela encontra-se na terceira forma normal quando, além de estar na 2FN, não possua dependências transitivas.



### Importante!

Dependência transitiva ocorre quando uma coluna, além de depender da chave primária da tabela, depende de outra coluna ou conjunto de colunas da tabela (HEUSER, 2010).

As regras para a terceira forma normal (3FN) são:

1. A tabela deve estar já na primeira forma normal (1FN);
2. Não existam atributos não-chave que dependam de outros atributos não-chave.

A razão dessa regra é detectar ainda outras fontes de dados redundantes. Se o valor de um atributo pode ser obtido simplesmente fazendo uso de outro atributo na tabela, então, ele não precisa estar na tabela. Criar uma tabela para armazenar esse tipo de atributo, tornará o banco de dados menor.

Desse modo, a tabela Empregado obtida na segunda forma normal (2FN), ainda possui dados redundantes. Perceba, nessa tabela, que o valor de hora de trabalho do empregado está vinculado ao seu cargo, ou seja, a informação do atributo “Valor\_Hora” depende da informação contida no atributo “Cargo\_Empregado”.

Nesse contexto, para que a tabela **Empregado** passe para a terceira forma normal (3FN), devemos dividir a tabela em duas novas tabelas, criando, portanto, uma nova tabela chamada **Cargo**. Podemos conferir o resultado abaixo:

Tabela 9

TABELA EMPREGADO		
ID_Empregado	Nome_Empregado	Id_Cargo
1	João da Silva	1
2	Maria Fernanda	2
3	Paulo Farias	3
4	Gustavo Fontes	3

Tabela 10

TABELA CARGO		
Id_Cargo	Cargo_Empregado	Valor_Hora
1	Programador Junior	30.00
2	Gerente	80.00
3	Analista Sênior	40.00

Tabela 11

TABELA PROJETO	
ID_Projeto	Nome_Projeto
103	Manhattan
104	Houston

Tabela 12

TABELA PROJETO		
ID_Projeto	ID_Empregado	Horas_Trabalhadas
103	1	50
103	2	50
104	1	20
104	3	10

Assim, podemos concluir que as tabelas acima estão na terceira forma normal (3FN), já que todas as tabelas estão na 2FN e, além disso, os atributos não-chave de todas as tabelas não possuem dependência transitiva de outros atributos não-chave em suas respectivas tabelas.

## Forma Normal Boyce/Codd (FNBC) e 4<sup>a</sup> e 5<sup>a</sup> Forma Normal (4FN, 5FN)

Para Heuser (2010), a decomposição das tabelas até 3FN é suficiente para obter o esquema de um banco de dados. Contudo, na literatura, nos deparamos com a forma normal Boyce/Codd (FNBC) e 4<sup>a</sup> e 5<sup>a</sup> Forma Normal (4FN, 5FN).

Definimos que uma tabela está na forma normal Boyce/Codd (FNBC), se e somente se, cada determinante é uma chave candidata. A maioria das entidades em 3NF já estão em BCNF.

A tabela está na quarta forma normal (4FN) se nenhuma instância da tabela do banco de dados contiver dois ou mais dados independentes e multivalorados descrevendo a entidade relevante, então ela estará na quarta forma normal.

Uma tabela está na quinta forma normal (5FN) somente se estiver em 4NF e não puder ser decomposta em qualquer número de tabelas menores sem perda de dados.

# Material Complementar

## Indicações para saber mais sobre os assuntos abordados nesta Unidade:



Sites

### Normalização de Bancos de Dados Relacionais

<https://goo.gl/F6ytEQ>



Livros

### Banco de Dados: Implementação em SQL, PL/SQL e Oracle 11g

Sandra Puga, Edson França e Milton Goya

Capítulo 5 – Normalização



Vídeos

### Normalizacao: Primeira Forma Normal - 1FN

<https://youtu.be/3kJKJNKiaD4>

### Normalizacao: Segunda Forma Normal - 2FN

<https://youtu.be/mHoZZUYVFzk>

### Normalizacao: Terceira Forma Normal - 3FN

<https://youtu.be/EZvrGEpyNbs>

## Referências

COUGO, S. Paulo. **Modelagem Conceitual e Projeto de Banco de Dados**. Rio de Janeiro: Campus. 1997.

HEUSER, C. A. **Projeto de banco de dados**. 6.ed. Porto Alegre: Bookman, 2010.

MACHADO, F. N. R. **Banco de Dados: projeto e implementação**. 3. ed. São Paulo: Érica, 2014.



**Cruzeiro do Sul Virtual**  
Educação a Distância

www.cruzeirodosulvirtual.com.br  
Campus Liberdade  
Rua Galvão Bueno, 868  
CEP 01506-000  
São Paulo - SP - Brasil  
Tel: (55 11) 3385-3000



**Cruzeiro do Sul**  
Educacional