

Normalização de Banco de Dados



Índice

- ▶ Introdução
- ▶ O que é Normalização de Dados
 - ▶ Definição
 - ▶ Objetivo
- ▶ Formas de Normalização de Dados
 - ▶ Tipos de Formas de Normalização
 - ▶ 1FN - 1ª Forma Normal;
 - ▶ 2FN - 2ª Forma Normal;
 - ▶ 3FN - 3ª Forma Normal;

Introdução

A normalização de dados é um tema muito importante para a criação de um projeto de banco de dados, é a forma de organizar o seu banco de dados.

O que é Normalização de banco de dados?

Definição: A Normalização é um processo que aplicamos um conjunto de regras para todas as entidades de um banco de dados.

Objetivo: Evitar inconsistências no projeto de Banco de Dados, tais como: anomalias, redundância de dados, mistura de dados distintos na mesma tabela, nome de tabelas e de campos sem sentido.

Exemplos de Dados sem Normalização

Cod_Cli	cliente
01	jhon
02	MAX
03	maria
04	JULIANA

Cd_Prod	PROD
100	Teclado
101	Mouse
102	sdd
103	HD

notaFiscal	itnota
000001	Teclado
000002	Mouse
000003	sdd
000004	HD

O banco de Dados deve ter uma organização impecável, os dados devem ser normalizados, ou seja, ter um padrão, seguir uma linha de raciocínio, a nomenclatura dos campos deve ter sentido de acordo com os dados da tabela, existem boas práticas a serem adotadas nas tabelas acima.

Formas de Normalização de Dados 1.1

A adequação de um projeto de banco de dados deve-se basear em 05 perguntas que são nomeadas como Formas Normais, que correspondem a um conjunto de regras (popularmente chamada de regras de negócios) simplificando e adequando as entidades. O banco de dados do tipo Relacional está na forma correta quando satisfaz as regras de negócios.

Formas de Normalização de Dados 1.2

No slide anterior foi mencionado que existem 05 formas de normalização, existem outras formas de normalização que são aceitas.

O trabalho original de Edgar Frank Codd definiu apenas 03 dessas formas:

- ▶ 1FN - 1ª Forma Normal;
- ▶ 2FN - 2ª Forma Normal;
- ▶ 3FN - 3ª Forma Normal;

1FN - 1ª Forma Normal 1.1

Todos os campos de uma tabela deverão ser atômicos, ou seja, não pode conter grupos repetidos e nem campo multivalorado. Nesta forma normal, é necessário identificar a primary key da tabela, e identificar as colunas com dados repetidos e removê-las, em seguida criar uma nova tabela com a primary key para armazenar o dado repetido e, por fim, criar uma relação entre a tabela principal e a tabela secundária, conforme os exemplos a seguir.

1FN - 1ª Forma Normal 1.2

Figura 1 - Tabela sem normalização

codCliente	Cliente	Localizacao	Telefone
01	Joao	Sao Paulo, Sao Paulo, Brasil	11953861745, 1158551490
02	Maria	Fortaleza, Ceara, Brasil	85975455651, 8558178932
03	Jose	Teresina, Piaui, Brasil	8698974521, 8651112565

Os campos Localização e Telefone são multivalorado, contém mais de um dado.

1FN - 1ª Forma Normal 1.3

Figura 2 - Normalização do tipo 1FN

codCliente	Cliente	Cidade	UF	Pais
01	Joao	Sao Paulo	SP	Brasil
02	Maria	Fortaleza	CE	Brasil
03	Jose	Teresina	PI	Brasil

codCliente	Telefone
01	11953861745
01	1158551490
02	85975455651
02	8558178932
03	86989745210
03	8651112565

Os campos Cidade, UF, Pais foram adicionados na tabela principal e uma nova tabela foi com uma chave estrangeira referenciando a primeira tabela para armazenar o número de telefone.

2FN - 2ª Forma Normal 1.1

Para criar a normalização 2FN é necessário primeiramente estar na normalização 1FN, todos os atributos não chaves da tabela dependem unicamente apenas da chave primária e não pode depender de parte dela, é necessário identificar as colunas que não são funcionalmente dependentes da chave primária da tabela, em seguida, remover essa coluna da tabela principal e criar uma nova tabela com esses dados, veja a figura de exemplo a seguir.

2FN - 2ª Forma Normal 1.2

Figura3 - Tabela normalizada com a forma 1FN

codFuncionario	codProjeto	Funcionario	Projeto	localProjeto	Horas
100	0001	Fernanda	Versao 3.2	Santos	6
101	0001	Marcelo	Versao 3.2	Santos	12
103	0002	Joao	Versao 3.3	Belo Horizonte	18

A tabela possui dados distintos.
Os campos **Funcionario**, **Projeto**, **LocalProjeto**
e **Horas** devem ser separados por tabelas.

2FN - 2ª Forma Normal 1.3

Figura3 - Tabelas normalizadas com a forma 2FN

Projeto		
codProjeto	Projeto	localProjeto
0001	Versao 3.2	Santos
0001	Versao 3.3	Belo horizonte

Funcionario	
codFuncionario	Funcionario
0001	Versao 3.2
0001	Versao 3.3

Projeto

Funcionario_Projeto		
codFuncionario	codProjeto	Horas
100	100	6
101	101	12
102	102	18

Os dados foram separados por tabelas e os relacionamentos foram mantidos, duas tabelas (Funcionario e Projeto) separadas por grupo de dados e uma tabela Funcionario_Projeto com o agrupamento dos dados das tabelas apartadas.

3FN - 3ª Forma Normal 1.1

A Normalização 3FN só é possível quando temos um caso de um banco de dados na 2FN, os atributos não chave devem ser independentes e depender apenas da chave primária (B depende de A, e somente para cada valor de A existe um valor em B), identificamos os campos que são dependentes de outros campos que não são chaves e extraí-las para outra entidade.

3FN - 3ª Forma Normal 1.2

Figura 4 - Tabela normalizada com a forma 2FN

codFuncionario	Funcionario	Nascimento	codDepartamento	Departamento	codGerente
01	Maria	01-09-1989	100	Vendas	010
02	Joao	12-12-2000	100	Vendas	010
03	Fernanda	23-05-1998	101	Marketing	011

A tabela possui dados que não dependem diretamente da chave primaria, são os campos Departamento e codGerente.

3FN - 3ª Forma Normal 1.3

Figura 5 - Tabelas normalizadas com a forma 3FN

Funcionario_Departamento

codFuncionario	Funcionario	Nascimento	codDepartamento
01	Maria	01-09-1989	100
02	Joao	12-12-2000	100
03	Fernanda	23-05-1998	101

Departamento

codDepartamento	Departamento	codGerente
100	Vendas	010
100	Vendas	010
101	Marketing	011

Os dados foram separados em duas tabelas e o relacionamento da entidade Funcionario_Departamento é a chave estrangeira com a entidade Departamento.

Resumindo a Normalização

A forma da normalização para ser aplicada devemos realizar avaliações da desnormalização da relação para normalizar o banco de dados. Veja a tabela abaixo:

Forma Normal	Avaliação	Resolução
1FN	Sem atributos atômicos e nenhuma relação agrupada	Crie relações novas para cada campo atômico ou faça relação aninhada.
2FN	Em relações onde vários atributos dependem da chave primária, identifique atributos não-chave que não dependem da chave primária.	Separe os atributos por tabelas e crie relacionamentos mantendo a relação com a chave primária.
3FN	Os atributos não-chave que não dependem da chave primária, ou seja, atributo não chave não pode depender da chave primária.	Separe os dados e crie uma relação que faz com que os atributos não-chave que determinam outros atributos não-chave.

Exercícios de Normalização

Primeira Forma Normal

Uma tabela está na 1FN, se e somente se, todos os valores das colunas da tabela forem **atômicos**. Assim, podemos dizer que os relacionamentos, como definidos acima, estão necessariamente na 1FN. Uma relação está na 1FN quando todos os atributos da relação estiverem baseados em um domínio simples, não contendo grupos ou valores repetidos.

Passagem à 1FN

- 1) Encontre a chave primária da tabela;
- 2) Fique ciente de quais são as colunas da tabela que apresentam dados repetidos para sejam removidas;
- 3) Crie uma tabela para esses dados repetidos, com a chave primária da anterior;
- 4) Por fim, estabeleça relação entre a nova tabela e a principal.

Exercício de Normalização - 01.

Analise a tabela abaixo e faça a normalização para a 1FN.

Veículos				
cod	Veiculo	proprietario	ano_modelo	Características
V01	Gol- VW	José Garcia	2010/2010	verde -2 portas
V02	Uno - FIAT	Julian Almeida	2013/2013	azul - 4 portas
V03	Hb20 - Hyundai	Lia Itachi	2013/2014	roxo - 4 portas
V03	City - Honda	Priscila Kravinsky	2014/2014	preto - 4 portas

Resumindo a Normalização

Segunda Forma Normal [\[editar | editar código-fonte \]](#)

Definição

Uma relação está na 2FN se, e somente se, estiver na 1FN e cada atributo não-chave for dependente da chave primária inteira, isto é, cada atributo não-chave não pode ser dependente de apenas parte da chave.

No caso de tabelas com chave primária composta, se um atributo depende apenas de uma parte primária, então esse atributo deve ser colocado em outra tabela.

Passagem à 2FN

- Geração de novas tabelas com DFs (Dependências Funcionais) completas.
- Análise de dependências funcionais:
 - *tipo* e *descrição* dependem de *codp*;
 - *nome*, *categ* e *salário* dependem de *code*;
 - *data_início* e *tempo_aloc* dependem de toda a chave.

Exercício de Normalização - 02.

Analise a tabela abaixo e faça a normalização para a 2FN.

Veículos					
codigo	veiculo	proprietario	cod_prop	marca	ID-marca
V01	Gol	José Garcia	25	VW	1
V02	Uno	Julian Almeida	30	fiat	2
V03	Hb20	Lia Itachi	35	Hyundai	3
V03	City	Priscila Kravinsky	40	Honda	5

Exercício de Normalização - 03.

Terceira Forma Normal

Definição

Uma relação R está na 3FN, se estiver na 2FN e cada atributo não-chave de R não possuir dependência transitiva, para cada chave candidata de R .

Passagem à 3FN

- Geração de novas tabelas com DF diretas
- Análise de dependências funcionais entre atributos não chave:
 - *salário* depende funcionalmente de *categ.*

Exercício de Normalização - 03.

Aplique as três primeiras Formas Normais à tabela PEDIDOS e produza um D.E.R. normalizado na 3FN:

PEDIDOS									
nr_pedido	data_pedido	id_cliente	nome_cliente	cod_prod	nome_prod	quant	vl_unit	total_prod	total_ped
001	10/01/2011	1003	Ernesto	31	Caderno	2	15,00	30,00	38,00
				42	Caneta	1	3,00	3,00	
				67	Lápis	5	1,00	5,00	
002	11/01/2001	1007	Fabiana	42	Caneta	2	3,00	6,00	19,00
				67	Lápis	3	1,00	3,00	
				85	Lapiseira	2	5,00	10,00	