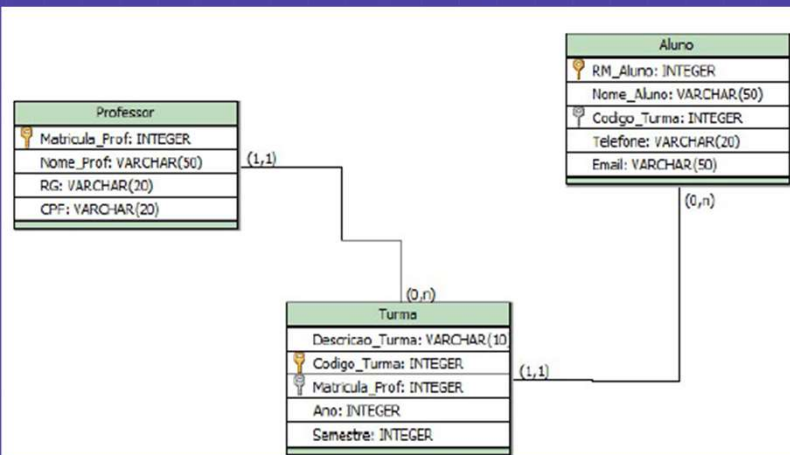
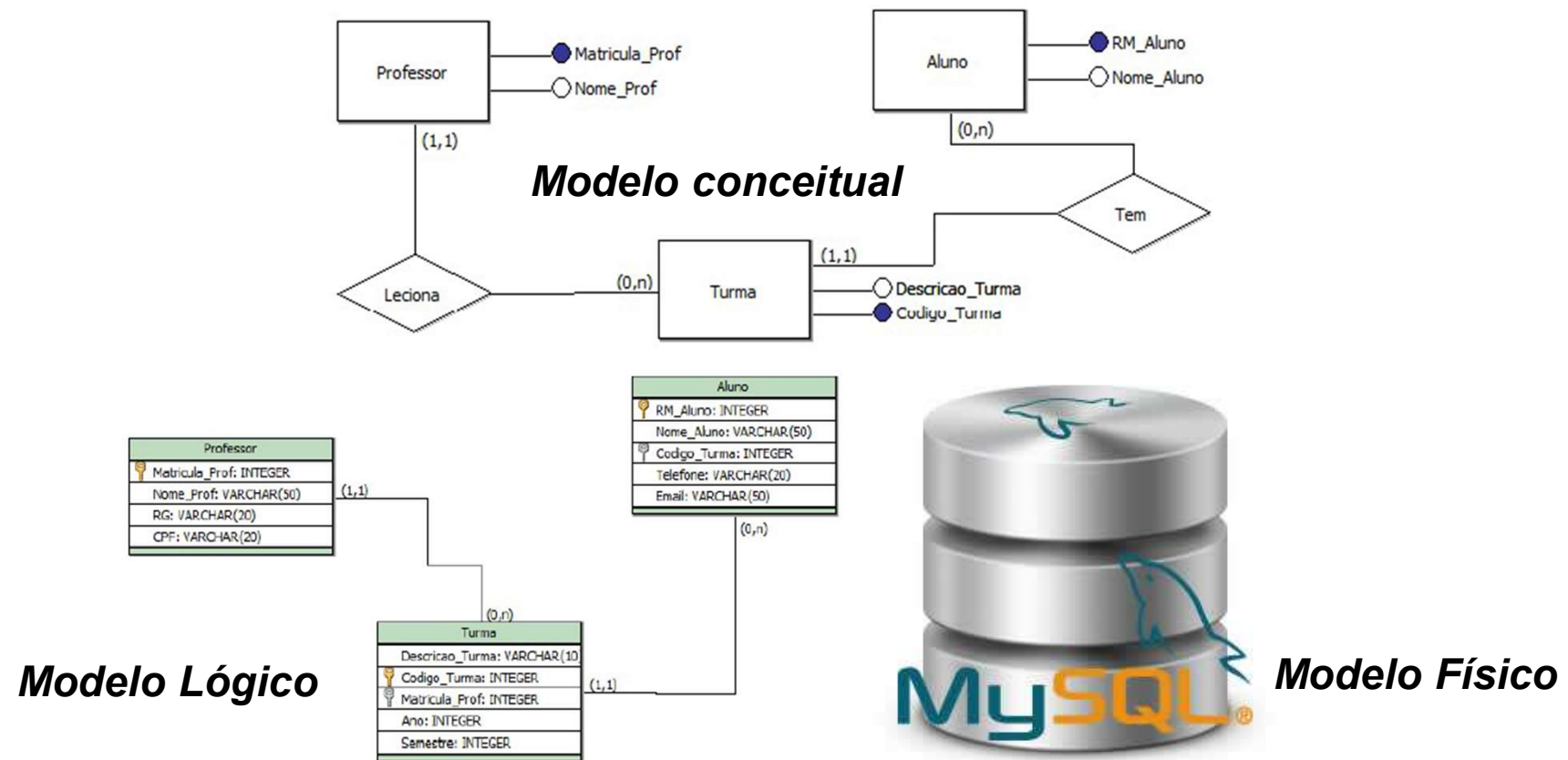


R

Modelagem e Banco de Dados + SQL




Banco de dados Relacional Introdução





Normalização de Dados: Construindo um Banco de Dados Sólido e Eficiente

A normalização é um processo crucial no design de bancos de dados que visa organizar as informações de forma eficiente, eliminando redundâncias e garantindo a integridade dos dados. As três primeiras formas normais (1FN, 2FN e 3FN) são os pilares desse processo, estabelecendo regras para a estrutura das tabelas e seus relacionamentos.



1ª Forma Normal (1FN): A Base da Organização

A 1FN exige que cada atributo de uma tabela contenha apenas um valor atômico, ou seja, indivisível. Isso significa que não podemos ter listas ou grupos de valores em um único atributo.

Exemplo sem 1FN (Sistema de Vendas):

ID Cliente	Nome Cliente	Telefones
1	João Silva	(11) 98765-4321
2	Maria Souza	(21) 91234-5678, (21) 99999-8888

Exemplo com 1FN (Sistema de Vendas):

ID Cliente	Nome Cliente	Telefone
1	João Silva	(11) 98765-4321
2	Maria Souza	(21) 91234-5678
2	Maria Souza	(21) 99999-8888

2ª Forma Normal (2FN): Eliminando Redundâncias Parciais

A 2FN entra em ação quando temos uma chave primária composta por mais de um atributo. Ela exige que todos os atributos não chave dependam funcionalmente da chave primária completa, e não apenas de parte dela.

Exemplo 1 sem 2FN (Sistema de Vendas):

ID Pedido	ID Produto	Nome Produto	Preço Unitário	Quantidade
1	10	Camiseta	30.00	2
1	20	Calça	50.00	1
2	10	Camiseta	30.00	3
2	30	Meia	10.00	5

Observe que o nome do produto e o preço unitário se repetem para o mesmo produto em diferentes pedidos. Isso causa redundância e pode levar a inconsistências se os dados forem atualizados apenas em uma das linhas.

Exemplo 1 com 2FN (Sistema de Vendas):

Tabela Produto:

ID Produto	Nome Produto	Preço Unitário
10	Camiseta	30.00
20	Calça	50.00
30	Meia	10.00

Tabela Pedido:

ID Pedido	ID Produto	Quantidade
1	10	2
1	20	1
2	10	3
2	30	5

Agora, cada produto é armazenado apenas uma vez na tabela "Produto", e a tabela "Pedido" armazena apenas as informações relevantes para o pedido em si, evitando redundância.

Exemplo 2: Sistema de Biblioteca

Tabela sem 2FN (Empréstimo):

ID Empréstimo	ID Livro	Título Livro	Autor Livro	ID Leitor	Nome Leitor	Data Empréstimo	Data Devolução
1	100	Dom Quixote	Cervantes	25	João Silva	2024-07-01	2024-07-15
2	100	Dom Quixote	Cervantes	30	Maria Souza	2024-07-05	2024-07-20

Observe que o título e o autor do livro se repetem para cada empréstimo do mesmo livro.

Tabelas com 2FN (Livro e Empréstimo):

Tabela Livro:

ID Livro	Título Livro	Autor Livro
100	Dom Quixote	Cervantes

Tabela Empréstimo:

ID Empréstimo	ID Livro	ID Leitor	Data Empréstimo	Data Devolução
1	100	25	2024-07-01	2024-07-15
2	100	30	2024-07-05	2024-07-20

Agora, cada livro é armazenado apenas uma vez na tabela "Livro", e a tabela "Empréstimo" armazena apenas as informações relevantes para o empréstimo em si.

Exemplo 3: Sistema de Controle de Estoque

Tabela sem 2FN (Produto_Fornecedor):

ID Produto	Nome Produto	ID Fornecedor	Nome Fornecedor	Preço Compra
1	Camiseta	10	Fornecedor A	15.00
1	Camiseta	20	Fornecedor B	14.50
2	Calça	10	Fornecedor A	30.00

Tabelas com 2FN (Produto, Fornecedor e Produto_Fornecedor):

Tabela Produto:

ID Produto	Nome Produto
1	Camiseta
2	Calça

Tabela Fornecedor:

ID Fornecedor	Nome Fornecedor
10	Fornecedor A
20	Fornecedor B

Tabela Produto_Fornecedor:

ID Produto	ID Fornecedor	Preço Compra
1	10	15.00
1	20	14.50
2	10	30.00

3ª Forma Normal (3FN): Removendo Dependências Transitivas

A 3FN busca eliminar as dependências transitivas, ou seja, quando um atributo não chave depende de outro atributo não chave.

Exemplo 1 sem 3FN (Sistema de Vendas):

ID Cliente	Nome Cliente	CEP	Cidade	Estado
1	João Silva	01234-567	São Paulo	SP
2	Maria Souza	01234-567	São Paulo	SP

Observe que a cidade e o estado dependem do CEP, que não é a chave primária.

Exemplo com 3FN (Sistema de Vendas):

Tabela Cliente:

ID Cliente	Nome Cliente	CEP
1	João Silva	01234-567
2	Maria Souza	01234-567

Tabela CEP:

CEP	Cidade	Estado
01234-567	São Paulo	SP

Ao criar uma tabela separada para o CEP, eliminamos a dependência transitiva e evitamos a repetição de informações sobre a cidade e o estado para cada cliente com o mesmo CEP.

Exemplo 2: Sistema de Recursos Humanos

Tabela sem 3FN (Funcionário):

ID Funcionário	Nome Funcionário	Departamento	Gerente Departamento
1	João Silva	Vendas	Maria Souza
2	Carlos Santos	Vendas	Maria Souza

Observe que o gerente do departamento depende do departamento, que não é a chave primária.

Tabelas com 3FN (Funcionário e Departamento):

Tabela Funcionário:

ID Funcionário	Nome Funcionário	ID Departamento
1	João Silva	10
2	Carlos Santos	10

Tabela Departamento:

ID Departamento	Departamento	Gerente Departamento
10	Vendas	Maria Souza

Exemplo 3: Sistema de Controle de Pedidos

Tabela sem 3FN (Pedido):

ID Pedido	ID Cliente	Nome Cliente	Endereço Cliente	Cidade Cliente	Estado Cliente
1	100	João Silva	Rua A, 123	São Paulo	SP
2	100	João Silva	Rua A, 123	São Paulo	SP

Observe que o endereço, a cidade e o estado do cliente dependem do ID do cliente, que não é a chave primária do pedido.

Tabelas com 3FN (Pedido, Cliente e Endereço):

Tabela Pedido:

ID Pedido	ID Cliente
1	100
2	100

Tabela Cliente:

ID Cliente	Nome Cliente	ID Endereço
100	João Silva	1

Tabela Endereço:

ID Endereço	Endereço Cliente	Cidade Cliente	Estado Cliente
1	Rua A, 123	São Paulo	SP

Ao aplicar a normalização, garantimos um banco de dados mais organizado, eficiente e livre de anomalias, facilitando a manutenção e o desenvolvimento de aplicações.