

Trabalho George

Exercício 1 – Conceito

Explique com suas próprias palavras o que significa que uma tabela esteja na Primeira Forma Normal .

Cada célula da tabela deve conter apenas um valor único e indivisível. Imagine uma célula como uma pequena caixa. Ela só pode ter um item. Por exemplo, em uma tabela de clientes, a coluna "Telefone" não deve conter mais de um número de telefone na mesma linha. Se o cliente tiver dois telefones, cada um deve ter sua própria linha ou coluna. Cada linha da tabela deve ser única. A combinação dos valores em cada linha deve ser distinta, para que seja possível identificar e referenciar cada registro de forma exclusiva. Isso geralmente é alcançado pela definição de uma chave primária.

Exercício 2 – Identificação

Considere a tabela abaixo:

Aluno	Disciplinas
João	Matemática, Física
Maria	Química
Pedro	História, Geografia, Filosofia

A tabela está em 1FN? Justifique sua resposta.

Não, a tabela não está em 1FN (Primeira Forma Normal).

Justificativa

Para uma tabela estar na Primeira Forma Normal, ela precisa seguir a regra principal de que todos os atributos (colunas) devem ser atômicos, ou seja, devem conter apenas um único valor indivisível por célula.

Na tabela apresentada, a coluna "Disciplinas" viola essa regra. Observe que:

Para o aluno "João", a célula da coluna "Disciplinas" contém dois valores: "Matemática, Física".

Para o aluno "Pedro", a célula contém três valores: "História, Geografia, Filosofia".

Essas células com múltiplos valores são chamadas de "valores multivalorados" e impedem que a tabela esteja em 1FN.

Exercício 3 – Conversão

Transforme a tabela do exercício 2 em uma tabela que respeite a 1FN .

Para normalizar a tabela para 1FN, os dados devem ser reestruturados de modo que cada célula contenha apenas um valor atômico. A forma mais comum de fazer isso é criando múltiplas linhas para cada aluno, uma para cada disciplina, ou criando uma tabela separada para as disciplinas.

Um exemplo de correção seria:

Aluno	Disciplina
João	Matemática
João	Física
Maria	Química
Pedro	História
Pedro	Geografia
Pedro	Filosofia

Exercício 4 – Valores Atômicos

Explique o que significa que os atributos de uma tabela devem ter valores atômicos na 1FN. Dê um exemplo.

Na Primeira Forma Normal (1FN) da normalização de bancos de dados, o requisito de valores atômicos significa que cada célula de uma tabela deve conter um único valor indivisível. Um valor atômico não pode ser decomposto em unidades menores que tenham significado individual no contexto do banco de dados.

Para cumprir a 1FN, é preciso evitar dois problemas principais:

Atributos multivalorados: Quando uma única célula contém uma lista de valores (por exemplo, vários números de telefone, habilidades ou cores).

Atributos compostos: Quando uma única célula armazena vários dados que podem ser divididos em colunas separadas (por exemplo, um endereço completo que inclui rua, cidade e estado). Exemplo

Imagine uma tabela de Funcionários que não está na 1FN porque viola a regra de atomicidade.

Tabela original (não está na 1FN):

ID	Nome	Endereco	Habilidades
1	Maria da Silva	Rua A, 123, São Paulo	Programação, Inglês

Essa tabela viola a 1FN por dois motivos:

1. Atributo composto: A coluna Endereco não é atômica, pois pode ser dividida em Rua, Numero, Cidade, etc..
2. Atributo multivalorado: A coluna Habilidades contém múltiplos valores para cada funcionário.

Exercício 5 – Tabela com Repetição

Analise a tabela:

Pedido	Produto	Quantidade
001	Caneta, Lápis	10, 5
002	Caderno	2

Ela está na 1FN? Caso negativo, refaça-a para que você esteja.

Não, a tabela não está na 1ª Forma Normal (1FN) porque as colunas "Produto" e "Quantidade" contêm múltiplos valores (multi-valoradas) em uma única célula. Para normalizá-la, é preciso decompor a tabela em linhas separadas para cada item e reestruturar as colunas para ter um único valor por célula, criando uma nova coluna para cada item do pedido.

Exercício 6 – Identificador

Por que cada tabela deve possuir uma chave primária para estar em 1FN? Explique com exemplo

Em um banco de dados relacional, cada tabela deve ter uma chave primária para estar na Primeira Forma Normal (1FN), pois a chave primária garante a identificação exclusiva de cada registro (linha). Isso é fundamental para evitar a redundância de dados e as chamadas "anomalias de dados" (inserção, exclusão e atualização), que são problemas que surgem quando há dados duplicados ou inconsistentes.

A 1FN define que todos os atributos (colunas) de uma tabela devem ser atômicos e monovalorados, ou seja, cada célula deve conter apenas um único valor. Embora essa seja a regra mais citada, a existência de uma chave primária é intrínseca ao conceito de uma "relação" (que, no modelo relacional, é o que chamamos de tabela). Sem uma chave primária, não há como distinguir um registro de outro de forma única, e a tabela perde a sua definição como uma relação.

Exemplo: Tabela sem chave primária e suas anomalias

Considere a seguinte tabela de Pedidos que não possui uma chave primária:

Nome do Cliente	Endereço do Cliente	Nome do Produto	Preço do Produto
-----------------	---------------------	-----------------	------------------

Ana	Rua A	Camiseta	50.00
Ana	Rua A	Calça	100.00
João	Rua B	Camiseta	50.00
João	Rua B	Sapato	150.00
Ana	Rua A	Saia	80.00

Exercício 7 – Multivalorados

A tabela a seguir contém informações de um paciente:

Paciente	Alergias
Ana	Pólen, Penicilina
Bruno	Frutos do mar

Identifique os problemas de 1FN e proponha uma solução.

O problema da tabela apresentada é a violação da Primeira Forma Normal (1FN), que exige que todos os atributos (colunas) contenham apenas valores atômicos, ou seja, indivisíveis e únicos.

Problema identificado: Atributo multivalorado

O atributo Alergias na linha do paciente "Ana" contém dois valores distintos ("Pólen" e "Penicilina"), o que impede a tabela de estar na 1FN.

Solução

Para resolver esse problema, a tabela deve ser decomposta em duas tabelas separadas que se relacionam, eliminando o atributo multivalorado e garantindo que cada coluna contenha apenas um valor atômico por linha.

Tabela 1: Paciente

Esta tabela armazenará informações sobre os pacientes e terá um identificador único (ID_Paciente) como chave primária.

ID_Paciente	Nome
1	Ana
2	Bruno

Tabela 2: Alergia

Esta tabela armazenará as alergias, cada uma com seu próprio identificador (ID_Alergia) e o nome da alergia.

ID_Alergia	Nome_Alergia
------------	--------------

1	Pólen
2	Penicilina
3	Frutos do mar

Tabela 3: Paciente_Alergia

Esta é uma tabela de relacionamento que liga os pacientes às suas alergias. Ela usa as chaves primárias das outras duas tabelas para estabelecer a relação de muitos-para-muitos (um paciente pode ter várias alergias, e uma alergia pode afetar vários pacientes).

ID_Paciente	ID_Alergia
1	1
1	2
2	3

Com essa normalização, cada célula das tabelas contém um único valor, resolvendo o problema do atributo multivalorado e colocando o banco de dados em conformidade com a 1FN.

Exercício 8 – Exercício Prático

Crie uma tabela de vendas de produtos que você está em 2FN, considerando que a tabela terá chave primária composta (PedidoID + ProdutoID).

Para que uma tabela de vendas esteja na Segunda Forma Normal (2FN) com uma chave primária composta (PedidoID + ProdutoID), é preciso seguir estas regras:

Estar na Primeira Forma Normal (1FN), ou seja, cada coluna deve conter apenas valores atômicos (não repetidos).

Eliminar dependências funcionais parciais, o que significa que todos os atributos não chave devem depender totalmente da chave primária composta.

Uma tabela que não está na 2FN teria informações sobre o pedido ou o produto que dependem apenas de uma parte da chave primária (PedidoID ou ProdutoID), gerando redundância de dados. Para corrigir isso, criamos tabelas separadas para as informações que têm dependência parcial.

Exemplo prático

Vamos começar com uma tabela não normalizada e, em seguida, separá-la em três tabelas para atingir a 2FN.

Cenário inicial (não na 2FN)

Imagine a seguinte tabela VENDAS, que viola a 2FN:

PedidoID	ProdutoID	NomeCliente	DataPedido	NomeProduto	PreçoProduto	Quantidade
----------	-----------	-------------	------------	-------------	--------------	------------

1	101	Maria	2025-10-27	Caneta Azul	R\$ 2,00	5
1	102	Maria	2025-10-27	Caderno A4	R\$ 15,00	2
2	101	João	2025-10-27	Caneta Azul	R\$ 2,00	10

Nesta tabela, a chave primária é composta por (PedidoID, ProdutoID). No entanto: NomeCliente e DataPedido dependem apenas de PedidoID. NomeProduto e PrecoProduto dependem apenas de ProdutoID. Isso causa redundância de dados, já que as informações de clientes e produtos são repetidas para cada item de pedido.

Tabelas na 2FN

Para normalizar, separamos a tabela em três:

1. Tabela PEDIDOS

Esta tabela armazena as informações que dependem apenas de PedidoID.

Chave primária: PedidoID

Colunas: PedidoID, NomeCliente, DataPedido

PedidoID	NomeCliente	DataPedido
1	Maria	2025-10-27
2	João	2025-10-27

2. Tabela PRODUTOS

Esta tabela armazena as informações que dependem apenas de ProdutoID.

Chave primária: ProdutoID

Colunas: ProdutoID, NomeProduto, PrecoProduto

ProdutoID	NomeProduto	PrecoProduto
101	Caneta Azul	R\$ 2,00
102	Caderno A4	R\$ 15,00

3. Tabela DETALHES_PEDIDO (ou Vendas)

Esta tabela central, que mantém a chave primária composta original, armazena apenas as informações que dependem da combinação de PedidoID e ProdutoID.

Chave primária: (PedidoID, ProdutoID)

Chaves estrangeiras: PedidoID (referencia PEDIDOS), ProdutoID (referencia PRODUTOS)

Colunas: PedidoID, ProdutoID, Quantidade

PedidoID	ProdutoID	Quantidade
1	101	5

1	102	2
2	101	10

Vantagens da normalização para 2FN

Redução da redundância: Informações de cliente e produto não são mais repetidas, economizando espaço em disco.

Melhora na integridade dos dados: É mais difícil cometer erros de digitação, pois a informação do produto e do cliente é atualizada em apenas um lugar.

Melhora o desempenho: As tabelas menores e mais específicas são mais rápidas de pesquisar e atualizar.

Exercício 9 – Teoria e Prática

Dê um exemplo de tabela que você está na 1FN mas não está na 2FN , explicando a razão.

Para dar um exemplo de uma tabela que está na Primeira Forma Normal (1FN) mas não na Segunda Forma Normal (2FN), primeiro precisamos entender as regras de cada uma.

1FN: Todos os atributos (colunas) devem ser atômicos, ou seja, não podem conter valores repetidos ou múltiplos em uma mesma célula.

2FN: Para uma tabela com chave primária composta (formada por duas ou mais colunas), todos os atributos não chave (colunas que não compõem a chave primária) devem depender da chave primária inteira, e não apenas de parte dela.

A violação da 2FN ocorre quando há uma dependência funcional parcial, onde um atributo não chave depende de apenas uma parte da chave primária composta.

Exemplo: Tabela de Inscrições em Cursos

Imagine uma tabela chamada Inscricoes que registra quais alunos se inscreveram em quais cursos.

ID_Aluno	Nome_Aluno	ID_Curso	Nome_Curso
101	Ana Silva	C100	Banco de Dados
101	Ana Silva	C200	Programação Web
102	João Santos	C100	Banco de Dados
103	Pedro Costa	C300	Estrutura de Dados

Análise da tabela:

Chave Primária Composta: A chave primária é a combinação de (ID_Aluno, ID_Curso), pois é a única maneira de identificar de forma única cada inscrição.

Atributos não chave: Nome_Aluno e Nome_Curso.

Por que a tabela está na 1FN:

Cada célula contém um único valor, sem repetições.

Por que a tabela não está na 2FN:

O atributo não chave Nome_Aluno depende apenas de ID_Aluno, que é apenas uma parte da chave primária composta.

O atributo não chave Nome_Curso depende apenas de ID_Curso, que é também apenas uma parte da chave primária composta.

Problemas causados pela violação da 2FN

Essa estrutura causa redundância e anomalias de dados:

Redundância: O nome do aluno ("Ana Silva") é repetido para cada curso em que ela se inscreve. O mesmo ocorre com o nome do curso ("Banco de Dados").

Anomalia de atualização: Se o nome de um aluno mudar, teríamos que atualizar todas as linhas em que ele aparece, correndo o risco de deixar alguma desatualizada.

Anomalia de inserção: Não podemos inserir um novo curso na tabela sem que ele seja associado a um aluno.

Como resolver: Normalizando para 2FN

Para normalizar a tabela e colocá-la na 2FN, dividimos a tabela original em duas:

1. Alunos, que armazena informações sobre os alunos.
2. Cursos, que armazena informações sobre os cursos.
3. Inscricoes, que armazena as associações entre alunos e cursos, mantendo apenas as chaves primárias de cada tabela.

Nova Tabela Alunos:

ID_Aluno (PK)	Nome_Aluno
101	Ana Silva
102	João Santos
103	Pedro Costa

Nova Tabela Cursos:

ID_Curso (PK)	Nome_Curso
C100	Banco de Dados
C200	Programação Web
C300	Estrutura de Dados

Nova Tabela Inscricoes:

ID_Aluno (FK)	ID_Curso (FK)
101	C100
101	C200
102	C100

Com essa nova estrutura, a dependência funcional parcial é eliminada e a tabela Inscricoes (que tem chave primária composta) não contém atributos não chave, cumprindo a regra da 2FN.

Exercício 10 – Revisão

Você recebeu a tabela abaixo:

VendaID	ID do produto	NomeProduto	PrecoUnitário	Quantidade
V001	201	Caneta	2,50	10
V001	202	Lápis	1,50	5

1. Verifique se a tabela está em 2FN.
2. Caso não esteja, reescreva-a em 2FN , separando as informações corretamente.

Verificação da 2FN

A tabela não está na Segunda Forma Normal (2FN2 FN

2FN). A chave primária é composta por VendaID e ID do produto. Os atributos NomeProduto e PrecoUnitário dependem apenas de ID do produto, que é uma parte da chave primária.

Normalização para 2FN

A tabela é dividida em duas novas tabelas para atender à 2FN.

Tabela de Vendas

Esta tabela contém informações sobre as vendas.

VendaID	ID do produto	Quantidade
V001	201	10
V001	202	5

Tabela de Produtos

Esta tabela contém informações sobre os produtos.

ID do produto	NomeProduto	PrecoUnitário
201	Caneta	2,50
202	Lápis	1,50

Resposta Final

A tabela original não está em 2FN. A normalização para 2FN resulta em duas tabelas: uma para vendas e outra para produtos.