

Trabalho George

Exercícios – Terceira Forma Normal (3FN)

Exercício 1 – Conceito

Explique com suas próprias palavras o que significa uma tabela estar na Terceira Forma Normal .

Em palavras simples, uma tabela está na Terceira Forma Normal (3FN) quando:

Primeiro, ela já está na Primeira e na Segunda Forma Normal (1FN e 2FN).

Além disso, não possui dependências transitivas.

A dependência transitiva é o ponto-chave. Ela ocorre quando uma coluna que não é a chave primária da tabela depende de uma outra coluna que também não é a chave primária, e não diretamente da chave primária.

Exercício 2 – Identificação

Considere a tabela abaixo:

| FuncionarioID | Nome | Departamento | GerenteDepartamento |
|---------------|-------|--------------|---------------------|
| 001 | Ana | Vendas | Carlos |
| 002 | Bruno | Compras | Carla |
| 003 | Carla | Vendas | Carlos |

A tabela lista três funcionários com seus IDs, nomes, departamentos e gerentes de departamento. Ana (ID 001) e Carla (ID 003) trabalham em Vendas sob o gerenciamento de Carlos, enquanto Bruno (ID 002) trabalha em Compras sob o gerenciamento de Carla. A tabela não fornece nenhuma instrução para o "Exercício 2 – Identificação", portanto, o que se pede é a análise da estrutura e dos dados da tabela.

Exercício 3 – Dependência Transitiva

Explique o que é dependência transitiva em uma tabela e por que ela viola a 3FN. Dê um exemplo.

A dependência transitiva ocorre quando um atributo não-chave de uma tabela depende de outro atributo que também não é a chave primária, e não da chave primária diretamente. Essa relação em cadeia ($A \rightarrow B$ e $B \rightarrow C$, então $A \rightarrow C$) é considerada uma violação da Terceira Forma Normal (3FN).

Por que a dependência transitiva viola a 3FN

A Terceira Forma Normal (3FN) estabelece que todos os atributos não-chave em uma tabela devem ser dependentes apenas da chave primária, e não de outros atributos não-chave. A dependência transitiva cria problemas como:

Redundância de dados: Os mesmos dados não-chave são repetidos em várias linhas, desperdiçando espaço de armazenamento.

Anomalias de atualização: Se for preciso atualizar a informação do atributo não-chave intermediário, será necessário modificar várias linhas da tabela, o que pode levar a inconsistências se a atualização não for completa.

Anomalias de exclusão: Ao apagar uma linha, pode-se perder dados importantes que estão associados ao atributo não-chave intermediário.

Exemplo de dependência transitiva

Considere uma tabela chamada Matriculas com as seguintes colunas:

ID_Matricula (Chave Primária)

ID_Aluno

Nome_Aluno

Codigo_Curso

Nome_Curso

Tabela: Matriculas

| ID_Matricula | ID_Aluno | Nome_Aluno | Codigo_Curso | Nome_Curso |
|--------------|----------|------------|--------------|--------------------------|
| 101 | 250 | João | INF101 | Introdução à Informática |
| 102 | 251 | Maria | MAT202 | Matemática Discreta |
| 103 | 250 | João | MAT202 | Matemática Discreta |
| 104 | 252 | Pedro | INF101 | Introdução à Informática |

Neste exemplo, a dependência transitiva é a seguinte:

ID_Matricula determina ID_Aluno, Nome_Aluno, Codigo_Curso e Nome_Curso (Dependência direta com a chave primária).

Codigo_Curso determina Nome_Curso (Dependência transitiva, pois Codigo_Curso não é a chave primária, mas determina outro atributo não-chave).

Como resolver a dependência transitiva (Aplicando a 3FN)

Para normalizar a tabela e remover a dependência transitiva, você deve dividir a tabela em duas:

1. Tabela Matriculas: Armazenará os dados relacionados ao aluno e ao curso matriculado.
2. Tabela Cursos: Armazenará as informações sobre os cursos, com uma chave primária para identificá-los.

Tabela: Matriculas (Normalizada)

| ID_Matricula | ID_Aluno | Nome_Aluno | Codigo_Curso |
|--------------|----------|------------|--------------|
| 101 | 250 | João | INF101 |
| 102 | 251 | Maria | MAT202 |

| | | | |
|-----|-----|-------|--------|
| 103 | 250 | João | MAT202 |
| 104 | 252 | Pedro | INF101 |

Tabela: Cursos (Normalizada)

Codigo_Curso (Chave Primária)

Nome_Curso

INF101

Introdução à Informática

MAT202

Matemática Discreta

Ao separar a tabela, cada atributo não-chave (como Nome_Aluno e Codigo_Curso) agora depende apenas da chave primária (ID_Matricula) na tabela Matriculas, e o Nome_Curso depende apenas da chave primária (Codigo_Curso) na tabela Cursos. Isso elimina a redundância e as anomalias de dados.

Exercício 4 – Conversão para 3FN

Transforme a tabela do exercício 2 em tabelas que respeitem a 3FN , mantendo as informações corretamente.

Considere a tabela abaixo:

| FuncionarioID | Nome | Departamento | GerenteDepartamento |
|---------------|-------|--------------|---------------------|
| 001 | Ana | Vendas | Carlos |
| 002 | Bruno | Compras | Carla |
| 003 | Carla | Vendas | Carlos |

Tabela 1: Funcionários

FuncionarioID 001,002,003

Nome: Ana, Bruno, Carla

Tabela 2: Departamentos

DepartamentoID: Vendas, Compras

NomeDepartamento: Vendas, Compras

GerenteID 001,003

Tabela 3: Gerentes

GerenteID:001,003

NomeGerente: Carlos, Carla

Exercício 5 – Tabela com Dependência Transitiva

A tabela a seguir contém informações de alunos e cursos:

| AlunoID | NomeAluno | CursoID | NomeCurso | CoordenadorCurso |
|---------|-----------|---------|------------|------------------|
| A01 | João | C01 | Matemática | Profa. Ana |
| A02 | Maria | CO2 | Física | Prof. Bruno |

Identifique as dependências transitivas e refatore a tabela para 3FN.

Refatoração para a 3ª Forma Normal (3FN)

Para remover a dependência transitiva e levar a tabela para a 3FN, é necessário dividi-la em tabelas separadas, de modo que cada atributo não-chave dependa diretamente apenas da chave primária.

Tabela de Alunos

Esta tabela armazenará os dados específicos de cada aluno.

Nome da Tabela: Alunos

Chave Primária: AlunoID

Atributos:

AlunoID (chave primária)

NomeAluno

CursoID (chave estrangeira)

| AlunoID | NomeAluno | CursoID |
|---------|-----------|---------|
|---------|-----------|---------|

| | | |
|-----|------|-----|
| A01 | João | C01 |
|-----|------|-----|

| | | |
|-----|-------|-----|
| A02 | Maria | C02 |
|-----|-------|-----|

Tabela de Cursos

Esta tabela armazenará os dados específicos de cada curso, eliminando a redundância de informações que existia na tabela original.

Nome da Tabela: Cursos

Chave Primária: CursoID

Atributos:

CursoID (chave primária)

NomeCurso

CoordenadorCurso

| CursoID | NomeCurso | CoordenadorCurso |
|---------|-----------|------------------|
|---------|-----------|------------------|

| | | |
|-----|------------|------------|
| C01 | Matemática | Profa. Ana |
|-----|------------|------------|

| | | |
|-----|--------|-------------|
| C02 | Física | Prof. Bruno |
|-----|--------|-------------|

Com essa refatoração, a dependência transitiva é removida e a estrutura do banco de dados atinge a 3ª Forma Normal. Cada tabela agora lida com uma única entidade, reduzindo a redundância e facilitando a manutenção dos dados.

Exercício 6 – Benefícios da 3FN

Lista três benefícios de se normalizar uma tabela para a 3FN.

1. Redução da redundância: A 3FN elimina as dependências transitivas, que são atributos não-chave que dependem de outros atributos não-chave, em vez de dependerem diretamente da chave primária. Isso evita que a mesma informação seja armazenada em vários lugares, o que economiza espaço de armazenamento e torna o banco de dados mais eficiente.
2. Melhora da integridade dos dados: Ao remover a redundância, a 3FN garante que cada fato seja armazenado em apenas um lugar. Isso ajuda a manter a consistência dos dados, pois qualquer atualização, inserção ou exclusão é feita em um único registro, evitando que a informação se torne inconsistente em diferentes partes do banco de dados.
3. Prevenção de anomalias de dados: A normalização para a 3FN previne anomalias de atualização, inserção e exclusão. Por exemplo:
 1. Anomalia de atualização: A 3FN evita que a alteração de um valor em um local exija a alteração em vários outros, o que minimiza o risco de erros.
 2. Anomalia de inserção: Permite a inserção de novos registros sem precisar incluir dados repetidos ou desnecessários.
 3. Anomalia de exclusão: Garante que a exclusão de uma linha não cause a perda de informações sobre outros dados relacionados.

Exercício 7 – Redundância e 3FN

Explícita como a existência de informações redundantes pode ser eliminada ao aplicar a 3FN. Dê um exemplo.

A Terceira Forma Normal (3FN) elimina a redundância de dados ao remover dependências transitivas. Uma dependência transitiva ocorre quando uma coluna que não é chave depende de outra coluna que também não é chave, em vez de depender diretamente da chave primária.

Para aplicar a 3FN, você deve:

Identificar as dependências transitivas na tabela.

Criar uma nova tabela para cada dependência transitiva.

Mover os atributos dependentes para a nova tabela.

Manter a coluna que determinava a dependência na tabela original como uma chave estrangeira.

Esse processo de decomposição garante que cada coluna não-chave em uma tabela dependa apenas da chave primária, eliminando a repetição desnecessária de informações.

Exercício 8 – Exercício Prático

Crie uma tabela de funcionários e departamentos que estejam na 3FN, considerando que cada departamento possui um gerente.

O modelo proposto para a 3FN é:

1. Tabela Departamentos

Esta tabela armazena as informações dos departamentos e, para manter a 3FN, inclui a chave estrangeira (`gerente_id`) que aponta para o funcionário que o gerencia.

`id_departamento` (Chave Primária): Identificador único do departamento.

`nome_departamento`: Nome do departamento.

`gerente_id` (Chave Estrangeira): Referencia o `id_funcionario` na tabela `Funcionarios`, indicando quem é o gerente do departamento.

2. Tabela Funcionarios

Esta tabela armazena as informações de cada funcionário, incluindo o departamento ao qual pertence.

id_funcionario (Chave Primária): Identificador único do funcionário.

nome_funcionario: Nome completo do funcionário.

cargo: Cargo ou função do funcionário (por exemplo, "Desenvolvedor", "Analista", "Gerente").

id_departamento (Chave Estrangeira): Referencia o id_departamento na tabela Departamentos.

Exercício 9 – Teoria e Prática

Dê um exemplo de tabela que você está na 2FN mas não está na 3FN , explicando a razão.

Para entender uma tabela que está na 2FN, mas não na 3FN, é preciso primeiro compreender o que cada uma significa.

Regras de normalização

2FN (Segunda Forma Normal): A tabela deve estar na 1FN e todos os atributos que não fazem parte da chave primária devem depender totalmente da chave primária completa. Isso significa que não deve haver dependência funcional de um atributo não-chave em relação a apenas uma parte da chave primária.

3FN (Terceira Forma Normal): A tabela deve estar na 2FN e não deve conter dependências funcionais transitivas. Uma dependência transitiva ocorre quando um atributo não-chave depende de outro atributo que também não é chave. Em outras palavras, todos os atributos não-chave devem depender diretamente da chave primária.

Exemplo: Tabela de Vendas

Considere a seguinte tabela VENDAS:

| ID_Venda | ID_Vendedor | Nome_Vendedor | Código_Regiao | Regiao | Total |
|----------|-------------|---------------|---------------|--------------|--------|
| 101 | 1 | João | R1 | Centro-Oeste | 250,00 |
| 102 | 2 | Maria | R2 | Sudeste | 150,00 |
| 103 | 1 | João | R1 | Centro-Oeste | 300,00 |

Exercício 10 – Revisão

Você recebeu a tabela abaixo:

| PedidoID | ID do produto | NomeProduto | Categoria | EstoqueCategoria |
|----------|---------------|-------------|-----------|------------------|
|----------|---------------|-------------|-----------|------------------|

| | | | | |
|------|-----|--------|------------|-----|
| P001 | 101 | Caneta | Escritório | 500 |
| P002 | 102 | Lápis | Escritório | 500 |

1. Verifique se a tabela está em 3FN.
2. Caso não esteja, reescreva-a em 3FN , separando as informações corretamente.

1. A tabela está em 3FN (Terceira Forma Normal)?

✗ Não está.

👉 Motivos:

NomeProduto, Categoria e EstoqueCategoria dependem do ID do produto, e não diretamente do PedidoID.

Há dependência transitiva, o que viola a 3ª forma normal.

(Exemplo: PedidoID → ID do produto → NomeProduto, Categoria, EstoqueCategoria)

2. Reescrevendo em 3FN:

Precisamos eliminar as dependências parciais e transitivas.

Então, separamos em duas tabelas:

📁 Tabela PedidoProduto

PedidoID ID do produto

P001 101

P002 102

📁 Tabela Produto

ID do produto NomeProduto Categoria EstoqueCategoria

| | | | |
|-----|--------|------------|-----|
| 101 | Caneta | Escritório | 500 |
| 102 | Lápis | Escritório | 500 |

✅ Agora está em 3FN, pois:

Cada tabela tem uma chave primária que identifica unicamente suas tuplas.

Todos os atributos dependem diretamente da chave primária da sua própria tabela.

Não há dependências transitivas.

Quer que eu te mostre como ficaria o modelo relacional (PK, FK) também?