

Banco de Dados

Modelagem Lógica

Normalização

O problema

- Bancos de dados mal projetados podem ter **anomalias de modificação**
 - Efeitos colaterais inesperados que decorrem da modificação do conteúdo de uma tabela que possui redundâncias excessivas.

O problema

<u>NumAluno</u>	NomeAluno	Ingresso	<u>Turma</u>	PeriodoLetivo	Nota	CodDisc	NomeDisc
1	João	2007/2	1	2007/2	8,5	C1	Banco
1	João	2007/2	2	2008/1	7,5	C2	Cálculo
2	Maria	2007/2	3	2008/1	4,8	C3	Engenharia
2	Maria	2007/2	2	2008/1	6,8	C2	Cálculo

- Essa tabela possui **anomalias de inserção**
 - Ex: Para inserir uma disciplina é necessário inserir os outros dados, especialmente o número do aluno.

O problema

<u>NumAluno</u>	NomeAluno	Ingresso	<u>Turma</u>	PeriodoLetivo	Nota	CodDisc	NomeDisc
1	João	2007/2	1	2007/2	8,5	C1	Banco
1	João	2007/2	2	2008/1	7,5	C2	Cálculo
2	Maria	2008/1	3	2008/1	4,8	C3	Engenharia
2	Maria	2008/1	2	2008/1	6,8	C2	Cálculo

- Essa tabela possui **anomalias de atualização**
 - Ex: Se o nome do aluno 1 precisar ser alterado será necessário mudar duas linhas. Se esse aluno estiver matriculado em dez turmas, dez linhas terão de ser mudadas.

O problema

<u>NumAluno</u>	NomeAluno	Ingresso	<u>Turma</u>	PeriodoLetivo	Nota	CodDisc	NomeDisc
1	João	2007/2	1	2007/2	8,5	C1	Banco
1	João	2007/2	2	2008/1	7,5	C2	Cálculo
2	Maria	2008/1	3	2008/1	4,8	C3	Engenharia
2	Maria	2008/1	2	2008/1	6,8	C2	Cálculo

- Essa tabela possui **anomalias de exclusão**
 - Ex: Se o registro do aluno 2 na disciplina C3 (terceira linha) for excluída, as informações sobre a própria disciplina serão perdidas.

A solução

- O processo de **normalização de dados**
 - Permite remover as redundâncias das tabelas para evitar as anomalias de modificação.
 - Existem três formas normais comumente utilizadas: a 1FN, a 2FN e a 3FN.
 - Existem ainda determinadas “condições de exceção” que são chamadas de formas normais: a Forma Normal de Boyce-Codd, a 4FN e a 5FN, menos comuns na prática.

Normalização

Definição:

Consiste em definir o formato lógico adequado para as estruturas de dados que foram identificadas no projeto lógico do sistema, com o objetivo de minimizar o espaço utilizado pelos dados e garantir a integridade e confiabilidade das informações.

Normalização

- Desenvolvida por E.F. (Ted) Codd da IBM.
- Baseada na teoria de conjuntos.
- Tem como objetivo a organização dos dados em tabelas, garantindo a não redundância dos mesmos.
- Garante a obtenção de tabelas estáveis, evolutivas e consistentes.
- Processo formal de decomposição de dados em tabelas, seguindo determinados passos chamados de Formas Normais.

Normalização



- Normalmente aplica-se até a 3a. Forma Normal

Normalização

A normalização é feita através da análise dos dados que compõem as estruturas utilizando o conceito chamado "Formas Normais (FN)".

As FN são conjuntos de restrições aos quais os dados devem estar sujeitos.

O processo de Normalização permite remover as redundâncias das tabelas para evitar as anomalias de modificação (inserção e exclusão).

Normalização

Pode-se dizer que uma estrutura está na **primeira forma normal** (1FN), se os dados que a compõem satisfizerem as restrições definidas para esta etapa.

A normalização completa dos dados é feita seguindo as restrições das 5 formas normais existentes, sendo que a passagem de uma FN para outra só é feita quando a etapa anterior foi cumprida.

Apesar das 5 formas existentes, na prática, usa-se até a 3ª forma normal (3FN).

Normalização

Para realizar a normalização dos dados é primordial que seja definido um **campo chave** para a estrutura, campo este que irá identificar todos os demais campos que compõem os registros.

Formas Normais:

- Primeira Forma Normal (1FN)
- Segunda Forma Normal (2FN)
- Terceira Forma Normal (3FN)

Normalização

Primeira Forma Normal (1FN)

- Uma tabela está na **Primeira Forma Normal**, se não contiver:
 - Atributos compostos
 - Atributos multivalorados
- Para cada chave há a ocorrência de um e somente um dado de cada atributo
- Não devem existir grupos repetitivos.

Normalização

Primeira Forma Normal

- Para se obter tabelas na 1FN é necessário decompor cada tabela não normalizada em tantas tabelas quantos forem o número de conjuntos de atributos repetitivos.
- Nas novas chaves criadas, a chave primária será a concatenação da chave primária da tabela original, mais o(s) atributo(s) do grupo repetitivo visualizado como chave primária deste grupo.
- Após a aplicação da Primeira Forma Normal tem-se duas tabelas:

Normalização

Exemplo

CODIGO	NOME	TELEFONE
1	JOAO	31247891
		99150080
2	MARIA	55051133
3	PAULO	99601293



COD CLI	NOME
1	JOAO
2	MARIA
3	PAULO

COD CLI	TELEFONE	
1	31247891	
1	99150080	
2	55051133	
3	99601293	

Normalização

Considere a seguinte relação:

cliente = {CPF, nome, endereço, (telefone)}

A relação '*cliente*' não está na 1FN, porque **nome** e **endereço** são atributos **compostos** e **telefone** é um atributo **multivalorado**.

A normalização desta relação resulta nas seguintes relações:

cliente = {CPF, nome, sobrenome, rua, numero, bairro}

cliente_telefone = {CPF, DDD, fone}

Note que telefone foi para uma nova relação composta pela chave primária de cliente mais o telefone (decomposto).

Todos os atributos da relação *cliente_telefone* fazem parte da chave primária, ou seja, é criada uma chave primária **composta**.

Normalização

Considere a seguinte situação referente a um ALUNO:

Codigo Aluno

Nome Aluno

Sexo Aluno

Codigo Disciplina 1

Nome Disciplina 1

Carga Horária Disciplina 1

Nota-1 Disciplina 1

Nota-2 Disciplina 1

Média Disciplina 1

Numero Curso

Nome Curso

Duração Curso

7 vezes

Normalização

Aplicando-se a Primeira Forma Normal

ALUNO

Codigo Aluno

Nome Aluno

Sexo Aluno

Numero Curso

Nome Curso

Duração Curso

ALUNO-DISCIPLINA

Codigo Aluno

Codigo Disciplina

Nome Disciplina

Carga Horária Disciplina

Nota-1 Disciplina

Nota-2 Disciplina

Normalização

Segunda Forma Normal (2FN)

- Uma tabela está na **Segunda Forma Normal**, se estiver na Primeira Forma Normal, e
- Todos os atributos/colunas não-chave forem **totalmente** dependentes da chave composta.
- Os atributos/colunas não-chave que tiverem **dependencia parcial** em relação a chave composta devem ser retirados da tabela.

Normalização

Segunda Forma Normal

- Para se obter tabelas na 2FN, deve-se observar se alguma tabela possui chave primária concatenada, e analisar se existe algum atributo ou conjunto de atributos **com dependência parcial** em relação a alguma parte da chave primária concatenada.

ALUNO

Codigo Aluno

Nome Aluno

Sexo Aluno

Numero Curso

Nome Curso

Duração Curso

ALUNO-DISCIPLINA

Codigo Aluno

Codigo Disciplina

Nome Disciplina

Carga Horária Disciplina

Nota-1 Disciplina

Nota-2 Disciplina

Normalização

Segunda Forma Normal

- A aplicação da 2FN gera novas tabelas que herdarão a chave parcial e todos os atributos que dependem apenas desta chave parcial.
- Uma entidade para estar na 2FN não pode ter atributos com dependência parcial em relação à chave primária.

ALUNO

Codigo Aluno

Nome Aluno

Sexo Aluno

Numero Curso

Nome Curso

Duração Curso

ALUNO-DISCIPLINA

Codigo Aluno

Codigo Disciplina

Nota-1 Disciplina

Nota-2 Disciplina

DISCIPLINA

Codigo Disciplina

Nome Disciplina

Carga Horária Disciplina

Normalização

Considere:

Aluno_Disciplina = {CodigoAluno, CodigoDisciplina,
NomeAluno, NomeDisciplina, CargaHorariaDisciplina,
Nota1Disciplina, Nota2Disciplina }

As dependências funcionais são:

CodigoAluno \rightarrow NomeAluno

CodigoDisciplina \rightarrow NomeDisciplina, CargaHorariaDisciplina

(CodigoAluno, CodigoDisciplina) \rightarrow Nota1Disciplina,
 \rightarrow Nota2Disciplina

Normalização (2FN)

A normalização desta relação ficaria da seguinte forma:

AlunoDisciplina = {CodigoAluno, CodigoDisciplina,
Nota1Disciplina, Nota2Disciplina}

Disciplina = {CodigoDisciplina, NomeDisciplina,
CargaHorariaDisciplina}

Aluno = {CodigoAluno, NomeAluno, SexoAluno, NumCurso,
NomeCurso, CargaHorariaCurso}

A **normalização** foi feita da seguinte forma:

1. Mantém-se na tabela original as chaves e os atributos que dependem totalmente dela.
2. Para cada chave que possua atributos dependentes, cria-se uma nova relação, neste caso Aluno e Disciplina.

Normalização – 3FN

Terceira Forma Normal:

- Uma tabela está na **Terceira Forma Normal**, se estiver na Segunda Forma Normal, e não contiver atributos dependentes de atributos que não sejam chaves.
- Todos os atributos desta tabela devem depender somente da chave primária.

Normalização

Terceira Forma Normal

- Uma tabela está na 3FN se nenhum de seus atributos possui dependência em relação a outro atributo da tabela que não participe da chave primária, ou seja, não existe nenhum atributo intermediário entre a chave primária e o próprio atributo observado.

ALUNO

Codigo Aluno

Nome Aluno

Sexo Aluno

Numero Curso

Nome Curso

Duração Curso

ALUNODISCIPLINA

Codigo Aluno

Codigo Disciplina

Nota-1 Disciplina

Nota-2 Disciplina

DISCIPLINA

Codigo Disciplina

Nome Disciplina

Carga Horária Disciplina

Normalização

Terceira Forma Normal

- Ao retirar a dependência, deve ser criada uma nova tabela que tenha como chave primária o atributo que causou esta dependência, e como demais atributos todos aqueles que forem dependentes funcionalmente desta nova chave.

ALUNO

Codigo Aluno

Nome Aluno

Sexo Aluno

Numero Curso

ALUNODISCIPLINA

Codigo Aluno

Codigo Disciplina

Nota1 Disciplina

Nota2 Disciplina

DISCIPLINA

Codigo Disciplina

Nome Disciplina

Carga Horária Disciplina

CURSO

Numero Curso

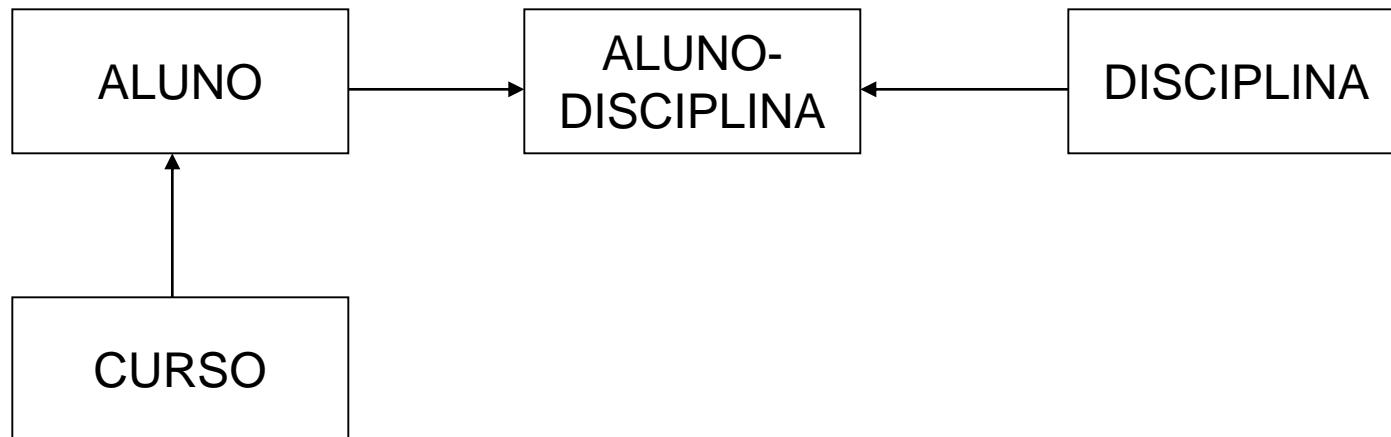
Nome Curso

Duração Curso

Normalização

Formas Normais

- Após a aplicação das Formas Normais teremos o seguinte modelo:



Normalização

Forma Normal Boyce_Codd (BCFN)

Uma entidade está na forma BCFN **se, e somente se**, estiver na 3FN e todos os atributos não chave dependerem apenas da chave primária.

Exemplo:

Cliente = {cod_cliente, nome_cliente, email_cliente}

Sendo cod_cliente a chave primária

cod_cliente → nome_cliente

cod_cliente → email_cliente

Todos os atributos não chave dependem funcionalmente apenas da chave primária. Logo está em BCFN

Exercícios

- Dado o modelo de nota fiscal a seguir ,
apresente o modelo lógico normalizado até a
3FN.

Nota Fiscal

Nº Nota Fiscal :	Série:	Data:		
Dados do Cliente				
Código:	Nome:	CGC:		
Endereço:				
Itens da Nota fiscal				
Código	Descrição	Preço Unitário	Quantidade	Total
Total Geral				

1ª Forma Normal

Analizando a estrutura anterior, observamos que existem várias produtos em uma única Nota Fiscal, sendo portanto, elementos repetitivos que deverão ser retirados.

Estrutura na primeira forma normal (1FN):

Arquivo de Notas Fiscais (Num. NF, Série, DataEmissão, CódigoCliente, NomeCliente, EndereçoCliente, CGCCliente e TotalGeralNota)

Arquivo de Vendas (Num. NF, CódigoProduto, DescriçãoProduto, PreçoUnitário, QuantidadeVendida e TotalVendaProduto)

Obs. Os campos sublinhados identificam as chaves das estruturas.

2ª Forma Normal

Consiste em retirar das estruturas que possuem **chaves compostas** (campo chave sendo formado por mais de um campo), os elementos que são funcionalmente dependentes de parte da chave.

Podemos afirmar que uma estrutura está na 2FN, se ela estiver na 1FN e não possuir campos que são funcionalmente dependentes de parte da chave.

2ª Forma Normal

Estrutura na segunda forma normal (2FN):

Arquivo de Notas Fiscais (Num. NF, Série, DataEmissão, CódigoCliente, NomeCliente, EndereçoCcliente, CGCCcliente e TotalGeralNota)

Arquivo de Vendas (Num. NF, CódigoProduto, QuantidadeVendida e Total VendaProduto)

Arquivo de Produtos (CódigoProduto, DescriçãoProduto, PreçoUnitário)

2ª Forma Normal

Como resultado desta etapa, houve um desdobramento do arquivo de Vendas (o arquivo de Notas Fiscais, não foi alterado, por não possuir chave composta) em duas estruturas a saber:

- Primeira estrutura (**Arquivo de Vendas**): Contém os elementos originais, sendo excluídos os dados que são dependentes apenas do campo Código do Produto.
- Segundo estrutura (**Arquivo de Produtos**): Contém os elementos que são identificados apenas pelo Código do Produto, ou seja, independentemente da Nota Fiscal, a descrição e o preço unitário serão constantes.

3ª Forma Normal

Consiste em retirar das estruturas os campos que **são funcionalmente dependentes de outros campos que não são chaves**. Podemos afirmar que uma estrutura está na 3FN, se ela estiver na 2FN e não possuir campos dependentes de outros campos não chaves.

Estrutura na terceira forma normal (3FN):

Arquivo de Notas Fiscais (Num. NF, Série, DataEmissão, CódigoCliente e TotalGeralNota)

Arquivo de Vendas (Num. NF, CódigoProduto, QuantidadeVendida e TotalVendaProduto)

Arquivo de produtos (CódigoProduto, DescriçãoProduto, PreçoUnitário)

Arquivo de Clientes (CódigoCliente, NomeCliente, EndereçoCliente e CGCCliente)

3ª Forma Normal

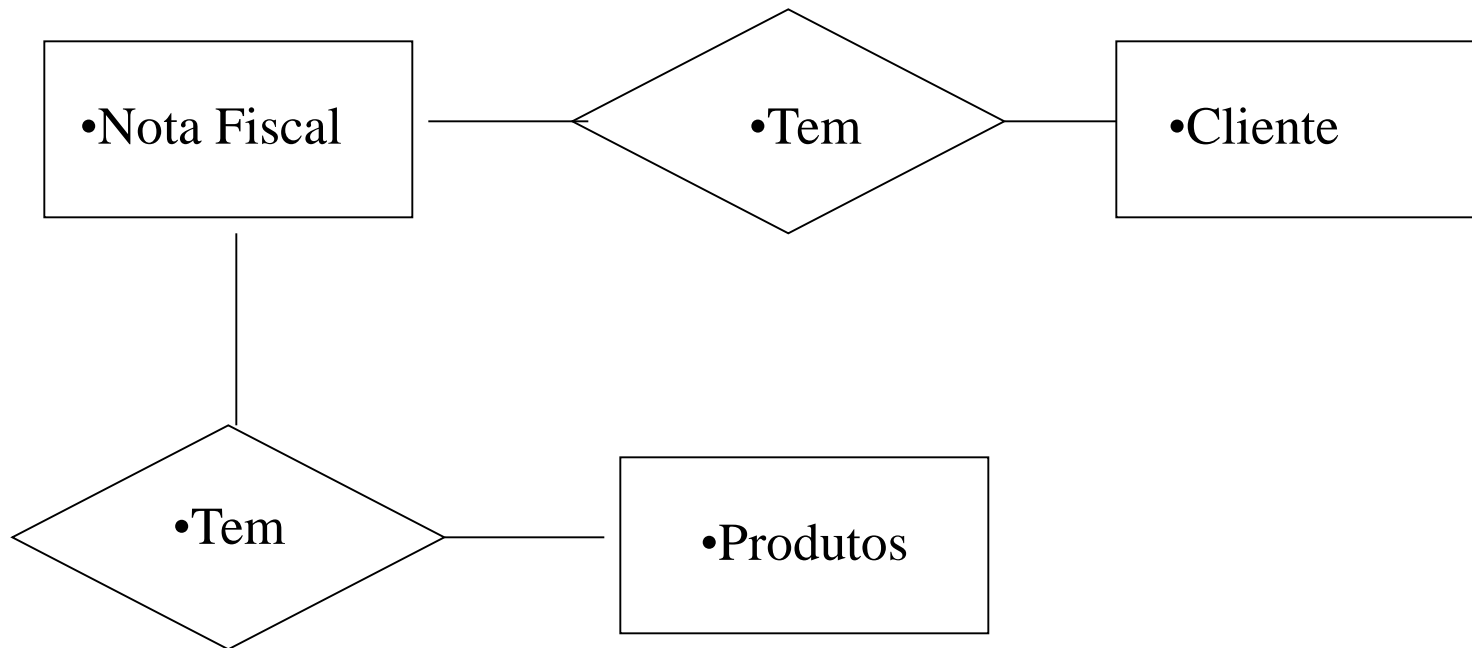
- Como resultado desta etapa, houve um desdobramento do arquivo de Notas Fiscais por ser o único que possuía campos que não eram dependentes da chave principal (Num. NF), uma vez que os dados dos clientes são independentes da Nota Fiscal.
- Este procedimento permite evitar inconsistência nos dados e economiza espaço pela eliminação das repetições, não havendo a necessidade de repetir dados dos clientes a cada nova compra.

3ª Forma Normal

As estruturas foram alteradas pelos motivos, a saber:

- **Arquivo de Notas Fiscais:** Contém os elementos originais, sendo excluídos os dados que são dependentes apenas do campo Código do Cliente (informações referentes ao cliente).
- **Arquivo de Clientes:** Contém os elementos que são identificados apenas pelo Código do Cliente, ou seja, independentemente da Nota Fiscal; o Nome, Endereço e CGC dos clientes serão constantes.
- Após a normalização, as estruturas dos dados estão projetadas para eliminar as inconsistências e redundâncias dos dados, eliminando desta forma qualquer problema de atualização e operacionalização do sistema.
- *A versão final dos dados ainda poderá sofrer alguma alteração para atender as necessidades específicas do sistema, a critério do analista de desenvolvimento durante o projeto físico do sistema.*

Diagrama Entidade - Relacionamento



Modelo de Dados

