

Normalização de Tabelas

Crícia Felício



Exemplo de esquema relacional “ruim”

Vendas

nomeC	<u>CPF</u>	endereco	fone	<u>codP</u>	nomeP	Vunit	qtd	total
Zé	111	ABC	123	A	Lápis	0,50	2	1,00
Ana	222	XYZ	456	B	Caneta	1,00	3	3,00
João	333	XPT	789	C	Régua	1,00	2	2,00
Pedro	444	KZZ	Null	A	Lápis	0,50	20	10,00

Histórico sobre Normalização

- Década de 70
 - Surgimento e o aperfeiçoamento do sistema relacional
 - Formas normais: São regras que foram definidas para a simplificação de tabelas.
 - Normalização:
 - Processo de adequação de tabelas as formas normais.
 - É um **processo formal**, passo a passo, de **análise** dos **atributos** de uma relação.

Normalização de Tabelas

Objetivo:

- evitar redundância,
 - inconsistência e
 - perda de informação no banco de dados.
-
- Teoria proposta por E.F. Codd no início dos anos 70.

Anomalias

Inclusão

Exclusão

Modificação

Ex: Quais problemas são decorrentes da relação Vendas?

nomeC	<u>CPE</u>	endereco	fone	<u>codP</u>	nomeP	Vunit	qtd	total
Zé	111	ABC	123	A	Lápis	0,50	2	1,00
Ana	222	XYZ	456	B	Caneta	1,00	3	3,00
João	333	XPT	789	C	Régua	1,00	2	2,00
Pedro	444	KZZ	Null	A	Lápis	0,50	20	10,00

Anomalias

- Problemas na atualização:
 - uma mudança no nome do produto A requer várias mudanças
- Inconsistência:
 - não há nada no projeto impedindo que o produto A tenha dois ou mais nomes diferentes no BD

nomeC	<u>CPE</u>	endereco	fone	<u>codP</u>	nomeP	Vunit	qtd	total
Zé	111	ABC	123	A	Lápis	0,50	2	1,00
Ana	222	XYZ	456	B	Caneta	1,00	3	3,00
João	333	XPT	789	C	Régua	1,00	2	2,00
Pedro	444	KZZ	Null	A	Lápis	0,50	20	10,00

Anomalia

- Inserção:
 - a inserção de um novo produto sem um pedido correspondente causa problema
- Exclusão:
 - se o cliente ANA fosse eliminado seria perdida a informação de que o produto B é chamado *caneta* e custa R\$ 1,00

nomeC	<u>CPE</u>	endereco	fone	<u>codP</u>	nomeP	Vunit	qtd	total
Zé	111	ABC	123	A	Lápis	0,50	2	1,00
Ana	222	XYZ	456	B	Caneta	1,00	3	3,00
João	333	XPT	789	C	Régua	1,00	2	2,00
Pedro	444	KZZ	Null	A	Lápis	0,50	20	10,00

Normalização de dados

- Como evitar problemas na criação de um novo banco de dados?
 - Definindo corretamente o modelo conceitual do projeto de banco de dados do sistema.
 - Transformando adequadamente o modelo conceitual em modelo lógico relacional.
- E quando o banco de dados já existir fisicamente?
 - Aplicar o processo de normalização.

Processo de Normalização

- Inicia com uma relação ou coleção de relações
- Produz uma nova coleção de relações:
 - equivalente a coleção original (representa a mesma informação)
 - livre de problemas
- **Significado:** as novas relações estarão normalizadas, pelo menos até a 3FN (Terceira Forma Normal)

Exemplo

- Qual esquema necessita de maior espaço de armazenamento?

Empregado(enome, CPF, datanasc, endereço, dnumero)

Projeto(pnome, pnumero, plocalizacao, dnum)

Trabalha-em(CPF, pnúmero, horas)

CPF referencia Empregado

pnumero referencia Projeto

Ou

EMP_PROJ (CPF, pnumero, horas, enome, pnome, Plocalizacao,dnumero,dnum)

Exemplo

- Identifique Anomalias de Inserção, Deleção e Atualização que podem ocorrer na relação abaixo:

EMP_PROJ (CPF, pnumero, horas, enome, pnome,
Plocalizacao, dnumero, dnum)

Exemplo

- Anomalia de Inserção:
 - Não se pode inserir um projeto a menos que um empregado esteja associado. Por outro lado, não se pode inserir um empregado a menos que ele esteja associado a um projeto
- Anomalia de Remoção:
 - Quando um projeto é removido, todos os empregados que trabalham no projeto serão removidos.
 - Da mesma forma, se um empregado for o único empregado do projeto, a remoção desse empregado resultará na remoção do projeto correspondente
- Anomalia de Atualização:
 - A mudança de nome de um projeto provoca alterações em todos os empregados que trabalham nesse projeto.

Formas Normais

- O conceito de normalização foi introduzido por E. F. Codd em 1972
 - Codd propôs:
 - Primeira forma normal (1FN)
 - Segunda forma normal (2FN)
 - Terceira forma normal (3FN)
 - Forma Normal de Boyce-Codd(FNBC)
 - Quarta Forma Normal(4FN)

Primeira forma normal

- Uma relação está na primeira forma normal se todos os seus atributos são **monovalorados** e **simples** e não possui relações aninhadas.
 - Monovalorados x Multivalorados
 - Exemplo: CPF é um atributo monovalorado
 - Simples X Compostos
 - Exemplo: salário é um atributo simples

Primeira Forma Normal

- O que deve ser feito para colocar uma relação na 1FN?
 - Quando encontrarmos um atributo(ou conjunto de atributos) multivalorado, deve-se:
 - Criar uma nova entidade com a chave primária da entidade anterior e o grupo repetitivo
 - Caso o grupo repetitivo possua chave primária, acrescentá-lo também como chave na nova relação
 - Quando encontrarmos um atributo composto, deve-se:
 - Dividi-lo em outros atributos que sejam simples

Exemplo

- Colocar na 1FN (presença de relações aninhadas)
 - **Projeto (codProj, tipo, descr, {codEmp, nome, categ, sal, dataIni, tempoAloc})**

CodProj	Tipo	descricao	Alocacao
001	Financeiro	Projeto SisFIN	{012,Joao,Contador,3000,15/03/2011,3 meses}, {013,maria,Contador,3000,15/03/2011,2 meses}
002	Informatica	Projeto Informatização	...
003	Infra-estrutur a	Projeto InfraNow	...

Exemplo

- Solução: Criar uma relação para as alocações

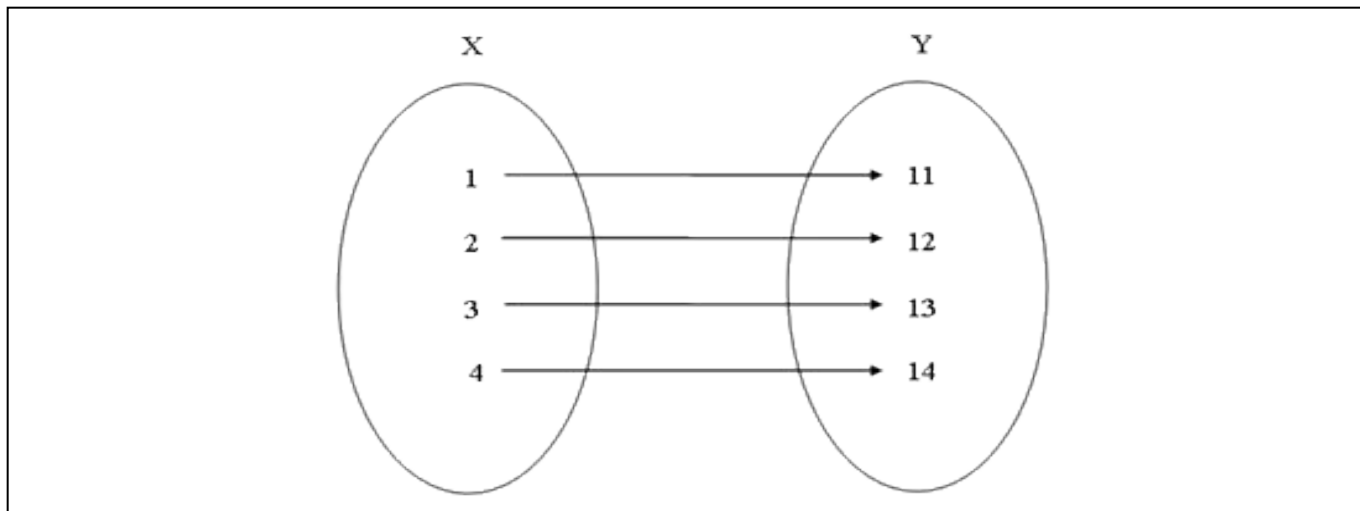
Projeto (CodProj, Tipo, Descr)

Alocacao(CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempoAloc)

Codproj referencia Projeto

Dependência Funcional

- O Modelo Relacional pegou emprestado da teoria de funções da matemática o conceito de **dependência funcional**
- Exemplo
 - Considere os seguintes conjuntos de valores

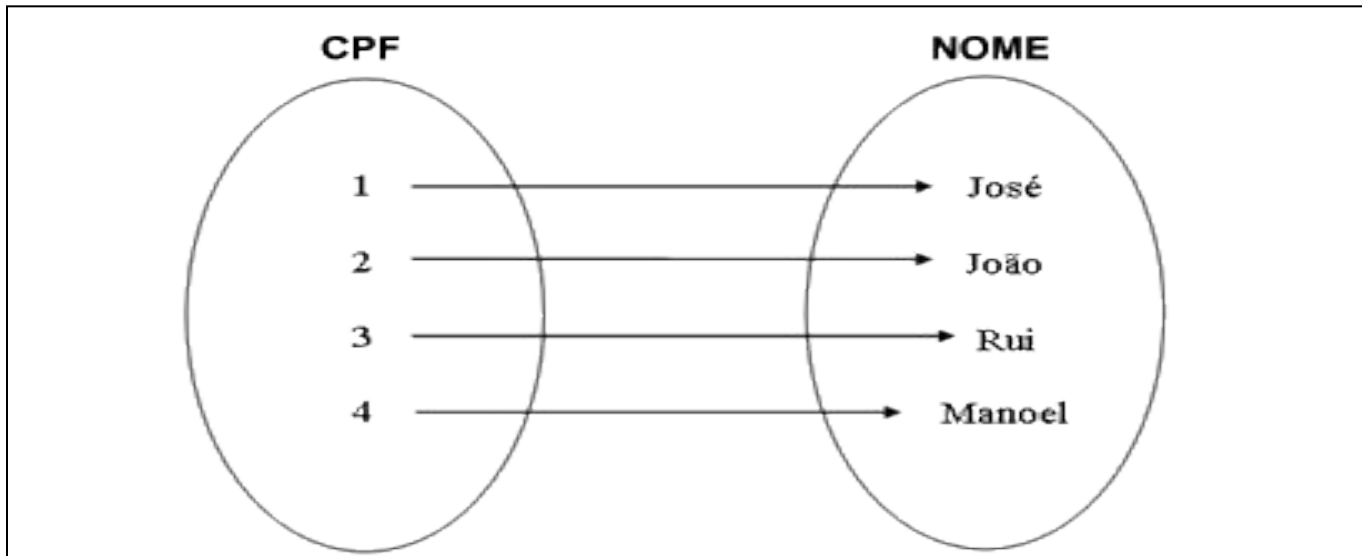


Dependência Funcional

- Nos conjuntos representados anteriormente há uma dependência de valores entre x e y , que pode ser expressa da seguinte forma:
 - $f(x) = x + 10$, ou seja, y é função de x , ou seja, $y = f(x) = x + 10$

Dependência Funcional

- Transpondo essa idéia para o modelo relacional
 - $f(\text{CPF}) = \text{nome}$
 - Através do CPF é possível determinar o nome da pessoa
 - O nome depende funcionalmente do CPF



Dependência Funcional

- Um subconjunto de atributos B de um esquema de relação R é funcionalmente dependente de um outro subconjunto de atributos A de R se o valor para A determinar um único valor para B

Segunda Forma Normal

- Uma relação está na segunda forma normal quando duas condições são satisfeitas:
 - A relação estiver na primeira forma normal
 - Todos os atributos da relação que não for chave primária dependem funcionalmente de toda a chave primária
 - Não pode haver dependência parcial

Segunda Forma Normal

- O que deve ser feito para colocar uma relação na 2FN?
 - Identificar todos os atributos que não possuem dependência total da chave
 - Remover os atributos nessa situação ou criar novas relações de forma a não haver dependências parciais

Exemplo

R=Fornecimento(CodFornecedor,CodMaterial, Preço, TelefoneF)

CodFornecedor	CodMaterial	Preço	TelefoneF
1234567	47	10,00	23 45678909
1234568	65	60,00	21 45678909
1234569	93	100,00	22 45678909

Telefonef depende funcionalmente apenas do atributo CodFornecedor

Exemplo

- Como o atributo telefonef já existe na tabela Fornecedor, esse atributo pode ser removido da tabela Fornecimento

CodFornecedor	CodMaterial	Preco
1234567	47	10,00
1234568	65	60,00
1234569	93	100,00

Tabela esta na
2FN

Exercício

- Colocar na 2FN

Projeto (CodProj, Tipo, Descr)

Alocacao (CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempoAloc)

A relação Alocação não esta na 2FN!

CodEmp \rightarrow Nome, Cat, Sal (DF Parcial!)

(CodProj, CodEmp) \rightarrow DataIni, TempoAloc

Exercício

- Solução: Criar uma relação que armazene as informações dos empregados

Projetos (CodProj, Tipo, Descr)

Empregado (CodEmp, Nome, Cat, Sal)

Alocacao (CodProj, CodEmp, DataIni, TempoAloc)

CodProj referencia Projeto

CodEmp referencia Empregado

Terceira Forma Normal

- Uma relação R está na Terceira Forma Normal se:
 - Está na Segunda Forma Normal
 - Não existem atributos não chave que sejam dependentes de outros atributos não chave
 - Atributos não chave devem depender exclusivamente de atributos chave

Terceira Forma Normal

- O que deve ser feito para colocar uma relação na 3FN?
 - Identificar todos os atributos que são funcionalmente dependentes de outros atributos não chave
 - Remover os atributos com dependência transitiva ou separar as relações

Exemplo

- R=Funcionario(CPF,RG,nome,cod_depto,nome_depto)

CPF	RG	Nome	Cod_depto	Nome_depto
1234567	m45678	Carlos	001	Informatica
1234568	m45679	Roberto	001	Informatica
1234569	m45680	Maria	003	Recurso Humanos

Exemplo

- Funcionario(CPF,RG,nome,cod_depto) e
- Departamento(cod_depto,nome_depto)

CPF	RG	Nome	Cod_depto
1234567	m45678	Carlos	001
1234568	m45679	Roberto	001
1234569	m45680	Maria	003

Cod_depto	Nome_depto
001	Informatica
002	Financeiro
003	Recurso Humanos

As relações
estão na 3FN

Exercício

- Colocar na 3FN

Projeto (CodProj, Tipo, Descr)

Empregado (CodEmp, Nome, Cat, Sal)

Alocacao (CodProj, CodEmp, DataIni, TempoAloc)

CodProj referencia Projeto

CodEmp referencia Empregado

Analizando as Dependências Funcionais:

CodEmp \rightarrow Sal

CodEmp \rightarrow Cat \rightarrow Sal (DF Transitiva!)

Exercício

■ Solução:

Projeto (CodProj, Tipo, Descr)

CategoriasFuncionais(Cat, Sal)

Empregado(CodEmp, Nome, Cat)

Cat referencia CategoriasFuncionais

Alocacao (CodProj, CodEmp, DataIni, TempoAloc)

CodProj referencia Projeto

CodEmp referencia Empregado

Exercício

- Normalize a seguinte relação: Slides baseados no material do Prof. André Santaché*

Membro						
IdMembro	Nome	Papel	Tarefa			
			IdTarefa	Descricao	DataInicio	HorasAlocadas
mel	Melissa	Gerente	1700	Planejamento e Orçamento	15/01/2012	80
mel	Melissa	Gerente	1701	Projeto do Sistema	15/02/2012	120
asd	Asdrúbal	Analista	1701	Projeto do Sistema	15/02/2012	180
asd	Asdrúbal	Analista	1705	Especificação da Arquitetura	01/03/2012	120
asd	Asdrúbal	Analista	1705	Detalhamento de Modelos	30/03/2012	200
dor	Doriana	Programador	1705	Detalhamento de Modelos	30/03/2012	120
dor	Doriana	Programador	1850	Implementação de componentes	15/04/2012	1200
qui	Quincas	Programador	1850	Implementação de componentes	15/04/2012	2400

*<https://www.ic.unicamp.br/~santanch/teaching/db/2013-1/slides/bd07-normalizacao-v01.pdf>

Exercício

- `Membro(idMembro, Nome, Papel, {idTarefa, Descricao, DataInicio, HorasAlocadas})`
 - Colocando na 1FN

Exercício

- Colocando na 2FN

Questões

- Análise de chaves primárias (Pks)

- *tabelas podem ou não ter atributos que garantam a identificação única de suas tuplas ou ter uma CP muito extensa*

sugestão: definir uma Chave Primária

ÑN: Projeto (CodProj, Tipo, Descr, {Nome, Cat, Sal, DataIni, TempoAl})



ÑN: Projeto (CodProj, Tipo, Descr, {CodEmp, Nome, Cat, Sal, DataIni, TempoAl})

Questões

- Dados irrelevantes
 - tabelas podem ter atributos que não precisam ser mantidos necessariamente no BD

sugestão: eliminar estes atributos

ÑN: Projetos (CodProj, Tipo, Descr, NroEmps, (CodEmp, Nome, Cat, Sal, DataIni, TempoAloc))



ÑN: Projetos (CodProj, Tipo, Descr, (CodEmp, Nome, Cat, Sal, DataIni, TempoAloc))

Questões

- Dados relevantes, porém implícitos

sugestão: definir tais dados

ÑN: Aprovação (CodCurso, Nome, (CodCand, Nome, Endereço))

a ordem determina
a classificação
do candidato

ÑN: Aprovação (CodCurso, Nome, (CodCand, Nome, Endereço, OrdemClass))

Exemplo

ÑN - Matricula (cod_aluno, cod_turma, cod_disciplina,
nome_disciplina, nome_aluno, cod_local_nasc, nome_local_nasc)

- Dependências ?

Exemplo

ÑN - Matricula (cod_aluno, cod_turma, cod_disciplina,
nome_disciplina, nome_aluno, cod_local_nasc, nome_local_nasc)

- Dependências:
 - (cod_aluno, cod_turma) -> cod_disciplina
 - cod_aluno -> nome_aluno, cod_local_nasc, nome_local_nasc
 - cod_disciplina -> nome_disciplina
 - cod_local_nasc -> nome_local_nasc

Exemplo

Matricula (cod_aluno, cod_turma, cod_disciplina, nome_disciplina, nome_aluno, cod_local_nasc, nome_local_nasc)

- Satisfaz a 1FN, mas não a 2FN
 - Apresenta dependências parciais em relação a chave primária

Exemplo

- Normalização

2FN

Matricula (cod_aluno, cod_turma, cod_disciplina, nome_disciplina)

Aluno(cod_aluno, nome_aluno, cod_local_nasc, nome_local_nasc)

Não satisfaz a 3FN pois apresenta dependencia funcional entre atributos não chave.

Exemplo

3FN

Disciplina (cod_disciplina, nome_disciplina)

Local(cod_local_nasc, nome_local_nasc)

Aluno (cod_aluno, nome_aluno, cod_local_nasc)

Cod_local_nasc referencia Local

Matricula (cod_aluno, cod_turma, cod_disciplina)

cod_aluno referencia aluno, co_disciplina referencia Disciplina

Revisão

Forma Normal	Verificação	Correção
1 FN	A relação não pode possuir atributos compostos ou multivalorados e não pode conter relações aninhadas	<ul style="list-style-type: none">• Separar em uma nova relação os atributos multivalorados ou relações aninhadas. Levando o campo chave da relação original.• Quebrar os atributos compostos.
2FN	Para relações que possuem a chave primária composta por mais de um atributo, não pode haver dependência funcional parcial em relação a chave.	<ul style="list-style-type: none">• Separar em uma nova relação a dependência parcial.• Manter a relação original com a sua chave e seus atributos funcionalmente dependentes.
3FN	A relação não pode ter atributos não chave que dependem funcionalmente de outros atributos não chave.	<ul style="list-style-type: none">• Separar em uma nova relação a dependência funcional entre atributos não chave.• Manter na relação original o campo chave da nova relação como chave estrangeira.