

BASE DE DADOS



NORMALIZAÇÃO

Teórico-Práticas
Ano Lectivo 2018/2019
Nuno Escudeiro/Melo e Castro/
Rui Coentro/Silva Pereira

Normalização

Processo de Normalização

- ★ Técnica formal para analisar uma **relação com base na sua chave primária** e nas dependências funcionais entre os atributos dessa relação.
- ★ Identificar um conjunto de dependências funcionais para uma relação tem como objetivo especificar o conjunto de restrições de integridade que devem ser titulares de uma relação.
- ★ Uma restrição de integridade importante a considerar em **primeiro lugar é a identificação das chaves candidatas, uma das quais será selecionada para ser a chave primária para a relação.**
- ★ O processo de normalização é executado a partir de uma série de passos. Cada passo corresponde a uma **forma normal**, com propriedades específicas.

Primeira Forma Normal

- ★ Uma relação está na 1FN se:
 - Os atributos chave estão definidos
 - Não existem grupos repetitivos
 - Todos os atributos estão definidos em domínios que contêm apenas valores atômicos, isto é, cada atributo só pode admitir valores elementares e não conjunto de valores
 - Todos os atributos dependem funcionalmente da chave primária

- ★ **Visa eliminar a existência de grupos de valores repetidos**
 - A uma ocorrência da chave só pode corresponder uma ocorrência dos outros atributos não chave

Normalização

Primeira Forma Normal

Suponhamos

Aluno (idAluno, nome, morada, (idDisciplina, nomeDisciplina))

Esta estrutura não se encontra na 1FN, uma vez que as colunas **idDisciplina** e **nomeDisciplina** admitem um conjunto de valores

<u>idAluno</u>	nome	morada	idDisciplina	nomeDisciplina
A1	João	Rua A	D1, D2, D3	Matemática, Economia, Direito
A2	Ana	Rua B	D1, D4	Matemática, Física
A3	Pedro	Rua C	D1, D2	Matemática, Economia
A4	Filipa	Rua D	D1	Matemática

Normalização

Primeira Forma Normal

Aluno (idAluno, nome, morada)

<u>Id aluno</u>	nome	morada
A1	João	Rua A
A2	Ana	Rua B
A3	Pedro	Rua C
A4	Filipa	Rua D

AlunoInscrito (idAluno, idDisciplina, nomeDisciplina)

<u>Id aluno</u>	<u>idDisciplina</u>	nomeDisciplina
A1	D1	Matematica
A1	D2	Economia
A1	D3	Direito
A2	D1	Matematica
A2	D4	Fisica
A3	D1	Matematica
A3	D2	Economia
A4	D1	Matematica

Segunda Forma Normal

- ★ Uma relação está na 2FN se:
 - Estiver na 1FN
 - Cada atributo não chave depende funcionalmente da totalidade da chave
 - ✧ Não existem dependências parciais
 - ✓ Todos os atributos que não pertencem à chave dependem funcionalmente da chave no seu conjunto e
 - ✓ Não dependem de nenhum dos seus elementos ou subconjuntos tomados isoladamente

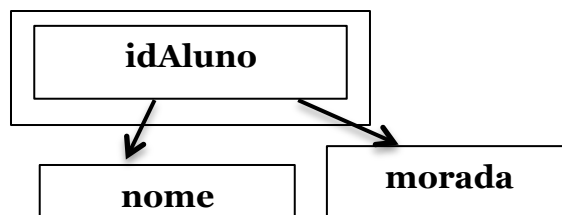
Normalização

Conversão da estrutura para a 2 FN

1. Se a relação só tem um atributo como chave primária e se essa relação já estiver na 1FN, então a relação também se encontra na 2FN
2. **Se a chave primária é composta e** se algum atributo não-chave depende apenas de uma parte da chave primária, então a relação deverá ser decomposta, para que cada atributo dependa da totalidade da chave primária

Exemplo 1

- A tabela **Aluno** já está na 1ª FN e a chave primária contém apenas um atributo ⇒ também está na 2ª FN




<u>Id aluno</u>	nome	morada
A1	João	Rua A
A2	Ana	Rua B
A3	Pedro	Rua C
A4	Filipa	Rua D

Normalização

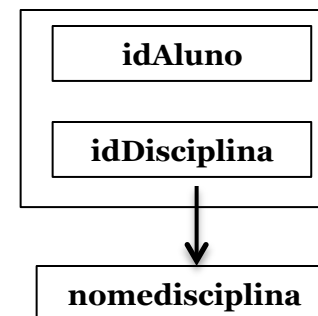
Conversão da estrutura para a 2 FN

- ★ A tabela ***AlunoInscrito*** encontra-se na 1ª FN mas a sua chave primária é composta
- ★ Necessário decompor a tabela ***AlunoInscrito*** pois existe uma dependência funcional entre o atributo não-chave ***nomeDisciplina*** e apenas parte da chave primária, com o atributo ***idDisciplina***

$idDisciplina \rightarrow nomeDisciplina$



<u>Id aluno</u>	<u>idDisciplina</u>	nomeDisciplina
A1	D1	Matemática
A1	D2	Economia
.....



Normalização

Conversão da estrutura para a 2 FN

Aluno(idAluno, nome, morada)

<u>Id aluno</u>	nome	morada
A1	João	Rua A
A2	Ana	Rua B
A3	Pedro	Rua C
A4	Filipa	Rua D

Disciplina(idDisciplina, nomeDisciplina)

<u>idDisciplina</u>	nomeDisciplina
D1	Matemática
D2	Economia
D3	Direito
D4	Física

AlunoInscrito(idAluno(FK), idDisciplina(FK))

<u>Id aluno</u>	<u>idDisciplina</u>
A1	D1
A1	D2
A1	D3
A2	D1
A2	D4
A3	D1
A3	D2
A4	D1

Normalização

Terceira Forma Normal

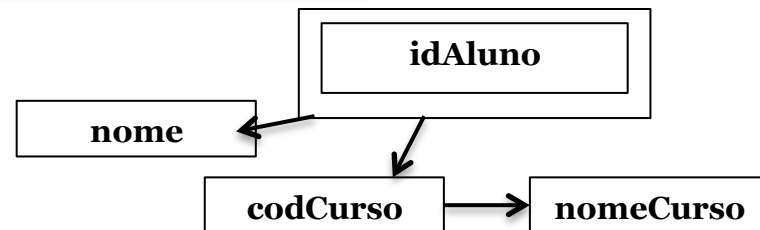
★ Uma relação está na 3FN se:

➤ Estiver na 2FN

➤ Nenhum dos seus atributos depende funcionalmente de atributos não chave

✓ Nenhum dos atributos que não fazem parte da chave pode ser funcionalmente dependente de qualquer combinação dos restantes

✓ Cada atributo depende apenas da chave e não de qualquer outro atributo ou conjunto de atributos



Aluno(idAluno, nome, codCurso, nomeCurso)

EXEMPLO

★ Esta tabela não se encontra na 3FN porque o atributo não-chave **nomeCurso** depende funcionalmente do atributo **codCurso**

<u>Id aluno</u>	nome	codCurso	nomeCurso
A1	João	01	Informática
A2	Ana	02	Civil
A3	Pedro	01	Informática
A4	Filipa	03	Química

Conversão da estrutura para a 3FN

1. Procurar dependências funcionais entre os atributos não-chave da relação
2. Se a relação que já está na 2FN e **tiver apenas um atributo não-chave**, então a relação também já se encontra na 3FN
3. Se existir algum conjunto de atributos não-chave na relação que tenha dependência funcional em relação a um outro conjunto de atributos não-chave da mesma relação, então a relação deve ser decomposta de modo a que qualquer atributo não-chave da relação só dependa da chave primária da relação

Normalização

Conversão da estrutura para a 3FN

- ★ A tabela está na 2FN mas não está na 3FN
- ★ Necessário decompor a tabela **Aluno** pois existe uma dependência funcional (transitiva) entre o atributo não-chave **codCurso** e o atributo **nomeCurso**

Aluno(idAluno, nome, codCurso(FK))

<u>Id_aluno</u>	nome	codCurso
A1	João	01
A2	Ana	02
A3	Pedro	01
A4	Filipa	03

Curso(codCurso, nomecurso)

<u>codCurso</u>	nomeCurso
01	Informática
02	Civil
01	Informática
03	Quimica

Conclusão

- ★ O nível de normalização deve ser pensado contra outros critérios
 - Por exemplo, um nível de normalização exagerado pode originar problemas de performance
- ★ A redundância entre os dados não pode ser completamente eliminada
 - de facto, as chaves estrangeiras são também uma forma de redundância
 - Problemas que a redundância pode trazer
 - ✧ Custo de espaço de armazenamento - a redundância implica ocupar espaço adicional com algo que não acrescenta nada ao que já existe armazenado
 - ✧ Manutenção - uma simples alteração ou remoção pode implicar o acesso a várias tabelas, tornando-se difícil manter a coerência dos dados armazenados
 - ✧ Desempenho - Se a redundância for significativa, isso implicará mais acessos a disco para trazer os mesmos dados

Normalização

Exercício 1

Desenhe o diagrama de dependências funcionais e normalize a estrutura apresentada



Nr Factura	Data	codCliente	NomeCliente	CodProd	Descricao Produto	Valor	Quantidade	Desconto
000257	01-07-2016	1234567	João Gomes	12	Lápis Bic	100	250	5%
000257	01-07-2016	1234567	João Gomes	13	Bloco de notas	1000	200	5%
000257	01-07-2016	1234567	João Gomes	15	Caneta	70	50	0%
000258	01-07-2016	1234568	Ana Marques	12	Lápis Bic	100	400	6%
000258	01-07-2016	1234568	Ana Marques	16	Caderno	500	350	6%
000258	01-07-2016	1234568	Ana Marques	17	Régua	100	20	0%

Normalização

Exercício 2

Desenhe o diagrama de dependências funcionais e normalize a estrutura apresentada



FICHA DE DESLOCAÇÃO A CLIENTES

Numero Deslocação: _____

Data: __ / __ / __

Custo total: _____

Número do local: _____

Designação: _____

Número do Cliente: _____

Nome: _____

Número de país do Cliente: _____

Nome: _____

Telefone do Cliente: _____

FUNCIONÁRIOS QUE SE DESLOCARAM

Número funcionário	Nome funcionário

Normalização

Exercício 3



Desenhe o diagrama de dependências funcionais e normalize a estrutura apresentada

Considere a estrutura de dados seguinte, referente ao planeamento de produção de uma fábrica de artigos de plástico. A fábrica está estruturada em secções e cada secção é composta por diferentes centros de trabalho. Uma ordem de produção pode ser realizada em diversos centros de trabalho e utiliza diversas matérias-primas.

Planeamento de produção = {ordem_prod, produto, nome_produto, qtd_a_produzir, data_prev_inicio, data_prev_fim, data_real_inicio, data_real_fim, {secção, nome_secção, {centro_trabalho, desc_centro_trabalho}, localização}, {mat_prima, descrição_mp, qtd_mp}, percent_execução}