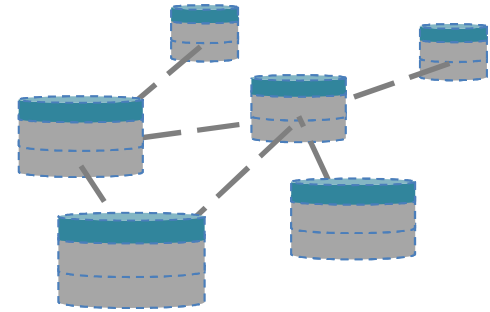


Sistemas de Bases de Dados

Notas de Leitura



07 >> Normalização de Dados

Orlando Belo

Departamento de Informática, Escola de Engenharia, Universidade do Minho
PORTUGAL

> www.di.uminho.pt/~omb

> www.researchgate.net/profile/Orlando_Belo

2020

07



Resumo

A **normalização de dados** é uma técnica especialmente orientada para a organização dos dados numa base de dados relacional. Através da sua aplicação, os atributos de um dado modelo de dados são organizados para assegurar **a coesão dos tipos das entidades envolvidas**, minimizando ou mesmo eliminando eventuais casos de duplicação de dados, melhorando os níveis de **eficiência de armazenamento**, bem como a **integridade e a escalabilidade dos dados**. Nesta unidade de ensino em sistemas de bases de dados abordaremos os aspetos mais importantes relacionados com a **normalização de dados**, focando a nossa atenção, em particular, **no processo de normalização de dados, nas diversas formas normais, dependências funcionais** e, por fim, numa eventual **desnormalização de uma base de dados**.



Estrutura da Apresentação

- Introdução
- Normalização de dados
- Anomalias nos dados
- Níveis de normalização
- Da 1ª à 5ª forma normal
- Dependências funcionais
- Axiomas de inferência
- Desnormalização de dados



Introdução

- O processo de desenvolvimento de um sistema de bases de dados, em particular do esquema da sua base de dados **não é um problema simples**.
- O projeto e a implementação de um esquema de uma base de dados **requer conhecimento para além do simples saber desenhar** com uma dada notação e construir a base de dados física através de uma linguagem específica, como a SQL.
- Esta atividade está rodeada de inúmeros outros aspetos que levantam alguns **cuidados e preocupações**, em especial aqueles que se relacionam com a **redundância, integridade e complexidade dos dados**.



Introdução

- Em particular, no que diz respeito a questões de redundância de dados (e também de integridade), temos algumas ferramentas e mecanismos que nos permitem controlar e gerir a duplicação de dados numa dada base de dados, através da análise do seu esquema e da semântica associada.
- A Teoria da Normalização de Dados é um desses mecanismos.



Codd e a Normalização

- Codd no seu artigo de 1970 - *A Relational Model of Data for Large Shared Data Banks* - propôs o processo de normalização e aquilo que hoje reconhecemos como a primeira forma normal.

“There is, in fact, a very simple elimination procedure which we shall call normalization. Through decomposition nonsimple domains are replaced by ‘domains whose elements are atomic (nondecomposable) values.’”

(Codd, 1970)



A Normalização de Dados

- Num processo de desenvolvimento de uma base de dados a normalização pode ser vista como **um mero processo de organização** que visa minimizar o impacto da duplicação de dados.
- Um processo de normalização implica normalmente **dividir uma base de dados em tabelas mais pequenas e definir os vários relacionamentos existentes entre elas.**



A Normalização de Dados

- O objetivo de qualquer processo de normalização é fazer o **isolamento dos dados** de forma a que eventuais **operações de inserção, modificação ou remoção** realizadas sobre um dado atributo sejam realizadas numa **única tabela** e depois **propagadas** pelo resto da base de dados seguindo os relacionamentos estabelecidos.
- De facto, o que a normalização de dados nos permite é o desenvolvimento de um modelo de dados lógico para uma base de dados relacional com **uma representação dos dados precisa**, bem como dos próprios relacionamentos e restrições associadas.



A Normalização de Dados

- A normalização de bases de dados é, pois, **um processo de remoção de dados redundantes das tabelas** que a constituem, com o intuito de aumentar a **eficiência de armazenamento, a integridade dos dados e as suas características de escalabilidade**.
- No modelo relacional existem métodos específicos para quantificar quanto eficiente é uma base de dados, que são usualmente reconhecidos por **Formas Normais (FN)**, havendo algoritmos específicos para converter uma dada base de dados nas diversas formas normais.



Dados Redundantes

AvaliaçãoAlunos

AlunoNr	Nome	Ano	Se	Disciplin a Nr	Designação	Nota	Professor Nr	Professor
A12345	João Costa	2014	1	R34V12	Engenharia de Software	18	1	Ribeiro Teixeira
A45231	Ana Pascoal	2014	1	R34V12	Engenharia de Software	12	1	Ribeiro Teixeira
A45231	Ana Pascoal	2014	1	T34D12	Comunicações por Computador	15	2	Costa Santos
A12345	João Costa	2014	1	BD4653	Bases de Dados	10	3	Narciso Pereira
A98976	Soraia Rodrigues	2014	1	T34D12	Comunicações por Computador	15	2	Costa Santos
A98976	Soraia Rodrigues	2014	1	T34D12	Comunicações por Computador	15	2	Costa Santos
(...)	(...)	(...)	(...)	(...)	(...)	(...)		

Ocorrência de dados
redundantes



Motivos para Normalizar

- Existem vários motivos que nos podem impelir para a aplicação de um processo de normalização sobre uma base de dados, em particular:
 - Controlo efetivo da redundância dos dados.
 - Flexibilidade, existem diversas formas de “olharmos” para os dados.
 - Garantia de integridade de dados.
 - Anomalias de modificações, na execução de operações de inserção, atualização e remoção.
 - Eficiência, ao se eliminar dados redundantes poupamos algum espaço de armazenamento.



Alguns Aspectos Importantes

- Num processo de normalização de dados, devemos ter em consideração alguns **aspectos importantes**. A saber:
 - Cada registo numa tabela deve representar uma entidade ou uma instância de um relacionamento.
 - A **referenciação** de outras entidades apenas deve ser realizada através de chaves estrangeiras.
 - Os atributos das entidades e dos relacionamentos devem ser **mantidos, tanto quanto possível, à parte**.
 - O desenvolvimento de um esquema deve poder ser explicado tabela a tabela.



Alguns Aspectos Importantes

- e também:
 - A semântica dos atributos deve ser de fácil interpretação.
 - Misturar atributos de várias entidades pode originar variadíssimos problemas – e.g. gastar maior espaço de armazenamento com informação redundante ou ter que lidar com anomalias de atualização de dados.



Anomalias nos Dados

- As tabelas de uma base de dados podem conter informação redundante que potencialmente pode sofrer processos de atualização anómalos.
- As anomalias dos dados podem ser provocadas por **problemas de inconsistência dos dados** armazenados numa base de dados, resultantes de um processo qualquer de inserção, atualização ou remoção de dados.
- Essas situações de inconsistência podem ser provocadas, por exemplo, por uma **atualização deficiente de um dado registo**, replicado por várias localizações, cujas réplicas não foram todas afetadas pela operação realizada.



Anomalias nos Dados

- As anomalias podem ser evitadas se implementarmos os diferentes **níveis de normalização** (ou formas normais) que a teoria da normalização considera.
- As anomalias que podem surgir durante a fase de construção e desenvolvimento de um esquema de uma base de dados podem ser vários **tipos**: inserção, **atualização ou remoção**.
- Nenhum destes tipos de anomalias é desejado num ambiente de bases de dados. As anomalias **podem ser evitadas através do processo de normalização** de uma base de dados.



Anomalia de Inserção

- Uma anomalia de **inserção** ocorre quando não se é capaz de se inserir um dado registo numa tabela por causa da inexistência de outros dados.

<u>FuncionárioNr</u>	Tipo	Nome	Função	ProjetoNr
1	N	João Francisco	Gestor de Projeto	P1
2	N	Maria Antonieta	Programador	P1
3	N	Ana Castro e Sousa	Programador	P2
4	N	Feliz da Silva	Administrador	P2
5	T	Joana Teles	Operador	NULL
(...)	(...)	(...)	(...)	(...)

Este atributo não admite valores nulos.

O que é que aconteceria se tentássemos inserir este registo?



Anomalia de Atualização

- Uma anomalia de **atualização** sucede quando se atualizam apenas alguns elementos de dados (**atualização parcial**) contidos numa base de dados, que estão replicados numa ou em mais tabelas – situação que conduz a base de dados a um estado inconsistente.

<u>Disciplina Nr</u>	<u>Sala Nr</u>	Designação	Capacidade	Área
A12345	101	Laboratório 1	30	100
A45231	001	Auditório 2	20	80
A76340	001	Auditório 2	20	80
A98976	234	Laboratório 2	30	120
(...)	(...)	(...)	(...)	(...)

O que é que aconteceria se modificássemos a designação ou a capacidade da sala apenas neste registo?



Anomalia de Remoção

- Uma anomalia de **remoção** ocorre quando se perde informação base por causa da remoção de um ou outro registo numa mesma tabela.

<u>Disciplina Nr</u>	Sala Nr	Designação	Capacidade	Área
A12345	101	Laboratório 1	30	100
A45231	001	Auditório 2	20	80
A76340	001	Auditório 2	20	80
A98976	234	Laboratório 2	30	120
(...)	(...)	(...)	(...)	(...)

O que é que aconteceria se removêssemos este registo?



A Influência dos Valores Nulos

- As tabelas devem ser construídas de forma a que os seus registos contenham o menor número de valores nulos possível.
- Os atributos de uma dada tabela que contenham frequentemente valores nulos devem ser avaliados no sentido de serem colocados numa outra tabela com a mesma chave primária.
- Os valores nulos podem ser provocados por várias situações, destacando-se as seguintes: atributo inválido ou não aplicável, valor de um atributo desconhecido, valor que existe mas que de momento não está disponível.



O Impacto de Operações de Junção

- Algumas operações de junção podem conduzir a resultados enganadores por terem sido realizadas sobre tabelas mal definidas.
- A propriedade junção sem perdas - *lossless join* - é usada para garantir que os resultados de um junção tenham significado.
- As tabelas de uma base de dados devem ser definidas de forma a satisfazerem a condição de junção sem perdas.
- Uma junção natural entre duas tabelas não deve produzir resultados com registos adulterados.



Propriedades da Decomposição

- Uma técnica que se utiliza para resolver algumas das anomalias referidas baseia-se definição de novos esquemas através da decomposição das tabelas que o constituem.
- Um processo de **decomposição** de uma dada tabela origina um novo conjunto de tabelas deve, que deve respeitar as seguintes **propriedades**:
 - **Decomposição sem perdas** (*loss-less join*) – garante-nos que o um esquema derivado a partir do esquema original contém a mesma informação que este último.
 - **Preservação de Dependências** (*dependency preservation*) – as dependências funcionais das tabelas do esquema original devem-se verificar também no esquema derivado.



Transformação de um Esquema

- O processo de **transformação de um dado esquema** de uma base de dados num segundo esquema equivalente deve obedecer aos seguintes princípios orientadores (Beeri et al., 1978):
 - **Princípio da Representação** – o esquema derivado e o esquema original devem ser equivalentes, contendo ambos o mesmo conjunto de atributos.
 - **Princípio da Separação** – um esquema derivado de um esquema original é superior se contiver a informação não relacionada contida em tabelas distintas.
 - **Princípio da Redundância Mínima** – um esquema derivado de um esquema original deve conter toda a informação deste última sem incorporação de situações de redundância de dados.



O Processo de Normalização

- O processo de normalização é **uma técnica formal de verificação** das diversas tabelas base de um esquema de uma base de dados, que se baseia na análise das **chaves primárias e das dependências funcionais** de todos os seus atributos.
- É um **processo progressivo**, que assenta na execução de uma série de etapas, cada uma delas correspondendo a uma **forma normal** específica – um **critério concreto de validação** -, que vai transformando o esquema de uma base de dados sucessivamente em direção a **um estado mais robusto e menos vulnerável**.



As Formas Normais

- 1FN - Primeira Forma Normal.
- 2FN - Segunda Forma Normal.
- 3FN - Terceira Forma Normal.
- FNBC - Forma Normal de Boyce-Codd.
- 4FN - Quarta Forma Normal.
- 5FN - Quinta Forma Normal.
- FNDC - Forma Normal Domínio-chave.
- 6FN - Sexta Forma Normal.



As Formas Normais

- 1FN – Primeira Forma Normal, Codd (1970) e Date (2003).
 - O domínio de cada atributo de uma relação apenas pode conter valores atómicos, e o valor de cada atributo é constituído apenas por um simples valor do seu domínio.
- 2FN – Segunda Forma Normal, Codd (1971).
 - Numa tabela nenhum atributo não primo é funcionalmente dependente num subconjunto próprio de qualquer chave candidata.



As Formas Normais

- 3FN – Terceira Forma Normal.
 - A versão de Codd (1971). Todo o atributo não primo de uma relação não é transitivamente dependente de uma chave candidata da tabela. Os atributos que não contribuam para a descrição da chave primária deverão ser removidos da tabela, o que significa que **não são permitidas dependências transitivas** entre os atributos de uma relação.
 - A versão de Zaniolo (1982) - FNCE – Forma normal da chave elementar (*Elementary Key Normal Form*). Toda a dependência funcional não trivial que exista uma tabela é tanto uma dependência de um atributo chave-elementar como uma dependência de uma super-chave.



As Formas Normais

- **FNBC** – **Forma Normal de Boyce–Codd**, Boyce e Codd (1974).
 - Numa relação qualquer dependência funcional não trivial é dependente de uma super-chave.
- **4FN** – **Quarta Forma Normal**, Fagin (1977).
 - Numa relação qualquer dependência multi-valor não trivial é uma dependência de uma super-chave.



As Formas Normais

- 5FN – Quinta Forma Normal, Fagin (1979).
 - Todas as dependências de junção não triviais numa tabela devem ser implicadas pelas suas super-chaves, ou seja que todos os determinantes funcionais das dependências de junção são super-chaves.
- FNDC – Forma Normal Domínio-Chave, Fagin (1981).
 - Qualquer restrição definida numa tabela é uma consequência lógica das restrições de chave e de domínio dessa tabela.



As Formas Normais

- 6FN – Sexta Forma Normal, Date, Darwen e Lorentzos (2002).
 - Todas as dependências de junção devem ser triviais.



Regras da Normalização

5

RULES OF DATA NORMALIZATION

1 ELIMINATE REPEATING GROUPS

Make a separate table for each set of related attributes, and give each table a primary key.

FIRST NORMAL FORM

Unnormalized Data Items For Puppies

Puppy Number
Puppy Name
Kennel Code
Kennel Name
Kennel Location
Trick ID 1..n
Trick Name 1..n
Trick When Learned 1..n
Skill Level 1..n

In the original list of data above, each puppy's description is followed by a list of tricks the puppy has learned. Some might know ten tricks, some might know none. To answer the question, "Can I tell you?" we need to first find if a puppy knows a trick, then we can list the tricks and the skill level of that trick. This is awkward, but it is the only way to represent the data.

Moving the tricks into a separate table helps considerably. Separating the repeating groups of tricks from the puppy information results in first normal form. The puppy number in the trick table matches the primary key in the puppy table, providing

Trick Table

Puppy #	Trick ID	Trick Name	When Learned	Skill Level
52	27	Roll Over	1	5
55	18	Stare Ahead	9	9
54	25	Roll Over	5	5

This table will hold a row for every puppy.

SECOND NORMAL FORM

In the Trick Table in first normal form, the primary key is made up of the puppy number and the trick ID. If you remove the puppy number, you are left with the "Where's Waldo?" effect. The "roll over" trick is now listed twice. This is a problem because the two puppy numbers are different. To avoid this, we need to separate the puppy number from the trick information. This is done by creating a separate table for the puppy information. This table will hold a row for every puppy. The puppy number in the trick table matches the primary key in the puppy table, providing

Puppy Table

Puppy Number	Puppy Name	Kennel Code	Kennel Name	Kennel Location
52	Spot	ABC	Alpha Kennel	London
55	Spot	DEF	Beta Kennel	Paris
54	Spot	GHI	Gamma Kennel	New York

Trick Table

Puppy Number	Trick ID	Trick Name	When Learned	Skill Level
52	27	Roll Over	1	5
55	18	Stare Ahead	9	9
54	25	Roll Over	5	5

2 ELIMINATE REDUNDANT DATA

If an attribute depends on only part of a multi-valued key, remove it to a separate table.

"As anything at all, perfection is finally attained not when there is no longer anything to add, but when there is no longer anything to take away."

Trick Table

Puppy #	Trick ID	Trick Name	When Learned	Skill Level
52	27	Roll Over	1	5
55	18	Stare Ahead	9	9
54	25	Roll Over	5	5

Puppy Table

Puppy Number	Puppy Name	Kennel Code	Kennel Name	Kennel Location
52	Spot	ABC	Alpha Kennel	London
55	Spot	DEF	Beta Kennel	Paris
54	Spot	GHI	Gamma Kennel	New York

Trick Table

Puppy Number	Trick ID	Trick Name	When Learned	Skill Level
52	27	Roll Over	1	5
55	18	Stare Ahead	9	9
54	25	Roll Over	5	5

3 ELIMINATE COLUMNS NOT DEPENDENT ON KEY

If attributes do not contribute to a description of the key, remove them to a separate table.

THIRD NORMAL FORM

Puppy Table

Puppy Number	Puppy Name	Kennel Code	Kennel Name	Kennel Location
52	Spot	ABC	Alpha Kennel	London
55	Spot	DEF	Beta Kennel	Paris
54	Spot	GHI	Gamma Kennel	New York

Trick Table

Puppy Number	Trick ID	Trick Name	When Learned	Skill Level
52	27	Roll Over	1	5
55	18	Stare Ahead	9	9
54	25	Roll Over	5	5

Puppy Table

Puppy Number	Puppy Name	Kennel Code	Kennel Name	Kennel Location
52	Spot	ABC	Alpha Kennel	London
55	Spot	DEF	Beta Kennel	Paris
54	Spot	GHI	Gamma Kennel	New York

Trick Table

Puppy Number	Trick ID	Trick Name	When Learned	Skill Level
52	27	Roll Over	1	5
55	18	Stare Ahead	9	9
54	25	Roll Over	5	5

4 ISOLATE INDEPENDENT MULTIPLE RELATIONSHIPS

No table may contain two or more 1:n or m:n relationships that are not directly related.

Trick Table

Puppy #	Trick ID	Trick Name	When Learned	Skill Level
52	27	Roll Over	1	5
55	18	Stare Ahead	9	9
54	25	Roll Over	5	5

Puppy Table

Puppy Number	Puppy Name	Kennel Code	Kennel Name	Kennel Location
52	Spot	ABC	Alpha Kennel	London
55	Spot	DEF	Beta Kennel	Paris
54	Spot	GHI	Gamma Kennel	New York

Trick Table

Puppy Number	Trick ID	Trick Name	When Learned	Skill Level
52	27	Roll Over	1	5
55	18	Stare Ahead	9	9
54	25	Roll Over	5	5

5 ISOLATE SEMANTICALLY RELATED MULTIPLE RELATIONSHIPS

There may be practical constraints on information that justify separating logically related many-to-many relationships.

Trick Table

Puppy #	Trick ID	Trick Name	When Learned	Skill Level
52	27	Roll Over	1	5
55	18	Stare Ahead	9	9
54	25	Roll Over	5	5

Puppy Table

Puppy Number	Puppy Name	Kennel Code	Kennel Name	Kennel Location
52	Spot	ABC	Alpha Kennel	London
55	Spot	DEF	Beta Kennel	Paris
54	Spot	GHI	Gamma Kennel	New York

Trick Table

Puppy Number	Trick ID	Trick Name	When Learned	Skill Level
52	27	Roll Over	1	5
55	18	Stare Ahead	9	9
54	25	Roll Over	5	5

DATABASE Programming & Design

1 ELIMINATE REPEATING GROUPS

Make a separate table for each set of related attributes, and give each table a primary key.

FIRST NORMAL FORM

Unnormalized Data Items For Puppies

Puppy Number
Puppy Name
Kennel Code
Kennel Name
Kennel Location
Trick ID 1..n
Trick Name 1..n
Trick When Learned 1..n
Skill Level 1..n

In the original list of data above, each puppy's description is followed by a list of tricks the puppy has learned. Some might know ten tricks, some might know none. To answer the question, "Can I tell you?" we need to first find if a puppy knows a trick, then we can list the tricks and the skill level of that trick. This is awkward, but it is the only way to represent the data.

Moving the tricks into a separate table helps considerably. Separating the repeating groups of tricks from the puppy information results in first normal form. The puppy number in the trick table matches the primary key in the puppy table, providing

Trick Table

Puppy #	Trick ID	Trick Name	When Learned	Skill Level
52	27	Roll Over	1	5
55	18	Stare Ahead	9	9
54	25	Roll Over	5	5

SECOND NORMAL FORM

In the Trick Table in first normal form, the primary key is made up of the puppy number and the trick ID. If you remove the puppy number, you are left with the "Where's Waldo?" effect. The "roll over" trick is now listed twice. This is a problem because the two puppy numbers are different. To avoid this, we need to separate the puppy number from the trick information. This is done by creating a separate table for the puppy information. This table will hold a row for every puppy. The puppy number in the trick table matches the primary key in the puppy table, providing

Puppy Table

Puppy Number	Puppy Name	Kennel Code	Kennel Name	Kennel Location
52	Spot	ABC	Alpha Kennel	London
55	Spot	DEF	Beta Kennel	Paris
54	Spot	GHI	Gamma Kennel	New York

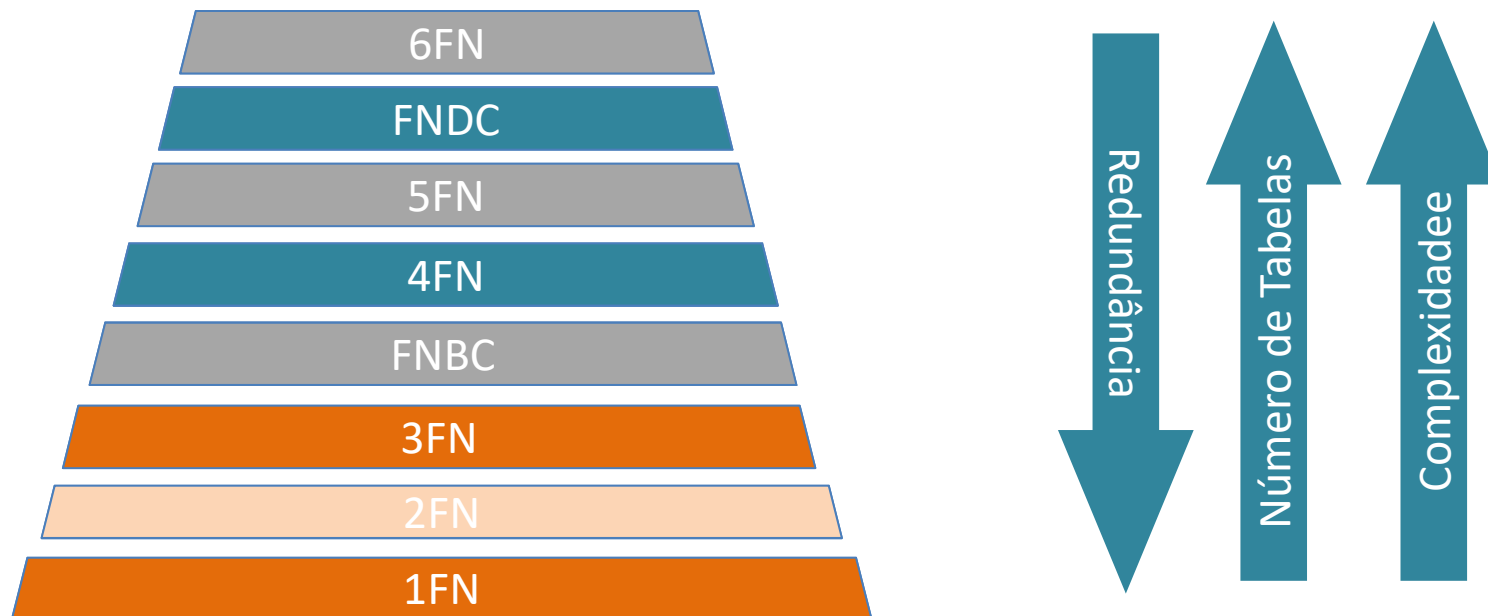
Trick Table

Puppy Number	Trick ID	Trick Name	When Learned	Skill Level
52	27	Roll Over	1	5
55	18	Stare Ahead	9	9
54	25	Roll Over	5	5

Fonte: <https://www.tes.com/lessons/eaxN1kTnVqp05w/ais275-week-5-normalization>



Níveis de Normalização



Obs.: Cada um dos níveis superiores de normalização é um subconjunto do respectivo nível inferior - o processo de normalização é progressivo.



Uma Base de Dados Normalizada

- Uma base de dados que foi **adequadamente normalizada**, usualmente, apresenta características como as seguintes:
 - Valores atômicos em todos os atributos.
 - Ausência de redundância.
 - Utilização mínima de valores nulos.
 - Perdas de informação mínimas.
- Além disso, uma base de dados normalizada permite **acomodar novos tipos de dados** sem que a sua estrutura base seja afetada – as aplicações que interagem com a base de dados não são afetadas.
- As tabelas normalizadas são muito convenientes para processos de **interrogação de carácter geral** - *general-purpose querying*.



Forma não Normalizada

- Uma **tabela não está normalizada** se contiver um ou mais grupos de dados repetidos e se se verificar a possibilidade de ocorrência de uma série de anomalias que podem conduzir a própria base de dados a um estado de inconsistência.
- Uma base de dados não está normalizada se existir pelos menos uma tabela não normalizada.



A Primeira Forma Normal

- Uma relação está na primeira forma normal (1FN) se todos os valores dos seus **atributos forem atômicos**, não contendo grupos de dados repetidos.
 - Um atributo atômico é aquele que não pode ser decomposto.
 - Um grupo repetitivo é um conjunto de um ou mais atributos com vários valores que estão relacionados.
- Na prática, podemos dizer que uma relação na qual as interseções entre colunas (atributos) e linhas (registos) apenas contêm um único valor – um valor atômico – estão na 1FN.



Requisitos da 1FN

- Os requisitos para que uma dada tabela esteja na 1FN são os seguintes:
 - Cada tabela tem uma chave primária.
 - Os valores de cada um dos atributos de uma tabela são todos atributos atômicos.
 - Não existem grupos de dados repetidos.



Exemplo de Aplicação da 1FN

Avaliação Alunos

AlunoNr	Nome	Ano	Se	Disciplina Nr	Designação	Nota	Professor Nr	Professor
A12345	João Costa	2014	1	R34V12	Engenharia de Software	18	1	Ribeiro Teixeira
A45231	Ana Páscoal	2014	1	R34V12	Engenharia de Software	12	1	Ribeiro Teixeira
A45231	Ana Pascoal	2014	1	T34D12	Comunicações por Computador	15	2	Costa Santos
A12345	João Costa	2014	1	BD4653	Bases de Dados	10	3	Narciso Pereira
A98976	Soraia Rodrigues	2014	1	T34D12	Comunicações por Computador	15	2	Costa Santos
A98976	Soraia Rodrigues	2014	1	T34D12	Lógica Computacional	15	2	Valença Oliveira.
(...)	(...)	(...)	(...)	(...)	(...)	(...)		

Grupo repetitivo



Exemplo de Aplicação da 1FN

Alunos

<u>AlunoNr</u>	Nome	Ano	Semestre
A12345	João Costa	2014	1
A45231	Ana Pascoal	2014	1

Avaliação

		AlunoNr	Disciplina Nr	Designação	Nota	Professor Nr	Professor
A98976	Sor						
(...)	(...)	A12345	R34V12	Engenharia de Software	18	1	Ribeiro Teixeira
AvaliaçãoAlunos 1FN Alunos Avaliação							
A98976	T34D12	Comunicações por Computador	15	2	Costa Santos		
A98976	T34D12	Comunicações por Computador	15	2	Costa Santos		
(...)	(...)	(...)	(...)	(...)			



A Segunda Forma Normal

- Uma **relação está na segunda forma normal (2FN)** se todos os seus atributos não-**primos** (que não fazem parte de uma chave primária) **foram** totalmente dependentes da sua chave primária.
 - Para cada **dependência funcional** $A1 \rightarrow A2$ de uma relação na 2FN, $A2$ é primo, ou $A1$ é uma super chave, ou $A1$ não estiver contido em nenhuma chave.
- Assim, Se tivermos o seguinte conjunto de dependências:
 - $\{A1, A2\} \rightarrow A3, A4, A5$
 - $A2 \rightarrow A4$a segunda dependência funcional é uma **dependência parcial**.



Dependências Funcionais

- As **dependências funcionais** (Codd, 1969) (Codd, 1970) são mecanismos que nos permitem especificar medidas formais sobre a correção dos esquemas relacionais.
- As **dependências funcionais são tipicamente restrições** que podem ser extraídas a partir do significado dos atributos e dos relacionamentos que sobre eles podem estar definidos.
- Juntamente com as chaves das relações de um esquemas, as dependências funcionais são utilizadas para **definir as diversas formas normais**, mecanismos de verificação da correção dos esquemas.



Dependências Funcionais

- As dependências funcionais são **aspectos chave** na compreensão do processo de normalização, uma vez que determinam a forma como se pode interpretar e relacionar os dados.
- As dependências funcionais são definidas com base nas **restrições aplicacionais** que se verifiquem sobre os atributos no contexto de um dado problema real.



Dependências Funcionais

- De uma forma um pouco mais formal, quando estabelecemos uma **dependência funcional** ($A1 \rightarrow A2$) entre dois conjuntos de atributos de uma dada tabela, X e Y, isso significa que:
 - existe apenas um valor possível para A2 para cada valor de A1, então A2 é funcionalmente dependente de A1 (ou que A1 determina funcionalmente A2)
 - valores iguais para A1, determinam valores iguais para A2.



Exemplos de Dependências Funcionais

- O número de um aluno de uma instituição universitária, ou o seu número de contribuinte, **determinam funcionalmente** o seu nome.
 - $\text{AlunoNr} \rightarrow \text{Nome}$
 - $\text{NrContribuinte} \rightarrow \text{Nome}$
- O número de um aluno e o número de uma disciplina determinam funcionalmente a nota que o aluno obteve a essa disciplina.
 - $\{\text{AlunoNr}, \text{DisciplinaNr}\} \rightarrow \text{Nota}$



Diagramas de Dependências

Alunos = {Número, Nome, DataNascimento, Curso, Designação, NrAlunos}

Discipli

Alunos

Conjunto de Dependências Funcionais

Número → Nome, DataNascimento, Curso

Curso → Designação, NrAlunos

Id → designação, Responsável

Responsável → Nome, departamento

{Aluno, Disciplina} → Nota, Data

Aluno → Nome

Disciplina → Designação

ta}



A Segunda Forma Normal

- Tal como a 1FN trata da questão da redundância de dados ao nível do registo, a 2FN trabalha a redundância dos dados em termos de atributos.
- Relembra-se aqui que, a aplicação das diversas formas normais é progressiva. Assim, para que uma dada relação esteja na 2FN tem de estar também na 1FN.



O Que Evita a 2FN

- Tanto a 2FN com a 3FN (que veremos de seguida) foram definidas para **evitar anomalias que ocorrem em processos de atualização** que envolvam dados redundantes e alguns efeitos colaterais que possam resultar da modificação de uma relação.
- A aplicação da 2FN permite **aumentar a eficiência no armazenamento de dados e reduz a repetição de dados**.



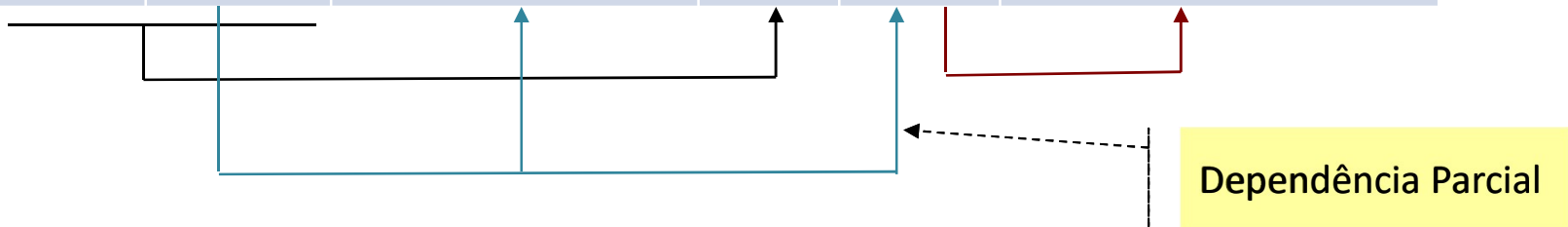
Exemplo de Aplicação da 2FN

Avaliação

<u>AlunoNr</u>	<u>DisciplinaNr</u>	Designação	Nota	ProfessorNr	Professor
A12345	R34V12	Engenharia de Software	18	1	Ribeiro Teixeira
A45231	R34V12	Engenharia de Software	12	1	Ribeiro Teixeira
A45231	T34D12	Comunicações por Computador	15	2	Silva Teixeira
A12345	BD4653	Bases de Dados	10	3	Alves Teixeira
A98976	T34D12	Comunicações por Computador	12	2	Silva Teixeira
A98976	T34D12	Comunicações por Computador	10	2	Silva Teixeira
(...)	(...)	(...)	(...)	(...)	(...)

Assumindo-se como verdadeiras as seguintes Dependências Funcionais:

- $\{\text{AlunoNr}, \text{DisciplinaNr}\} \rightarrow \text{Nota}$
- $\text{DisciplinaNr} \rightarrow \text{Designação}, \text{ProfessorNr}$
- $\text{ProfessorNr} \rightarrow \text{Professor}$



Resultado da Aplicação da 2FN

Avaliação

<u>AlunoNr</u>	<u>Disciplina Nr</u>	Nota
A12345	R34V12	18

Disciplinas

<u>Disciplina Nr</u>	Designação	Professor Nr	Professor
R34V12	Engenharia de Software	1	Ribeiro Teixeira
T34D12	Comunicações por Computador	2	Costa Santos
BD4653			
A98970			
A98970			
(...)			
T34D12	Comunicações por Computador	2	Costa Santos
T34D12	Comunicações por Computador	2	Costa Santos
(...)	(...)	(...)	(...)

Avaliação

2FN

Avaliação

Disciplinas



A Terceira Forma Normal

- Uma tabela está na 3FN se todos os atributos que não sejam chaves primárias sejam mutuamente independentes, não havendo assim dependências funcionais transitivas.
 - Para cada dependência $A1 \rightarrow A2 \in F^+$, $A2 \in A1$ (uma dependência trivial), ou $A1$ é uma super-chave, ou $A1$ é um atributo primo.
- Assim, se tivermos o seguinte conjunto de dependências:
 - $A1 \rightarrow A2, A3$
 - $A3 \rightarrow A4$a segunda dependência funcional revela uma dependência transitiva entre $A1$ e $A4$.



Requisitos da 3FN

- Os requisitos para que uma tabela esteja na 3FN são os seguintes:
 - Todos os requisitos da 1FN e da 2FN devem-se verificar.
 - Os atributos que não dependam da chave primária devem ser “eliminados”:
 - qualquer atributo que é dependente não só da chave primária como também de um outro qualquer atributo, deve ser transferido para outra tabela.
 - Não devem existir dependências transitivas.



O Que Evita a 3FN

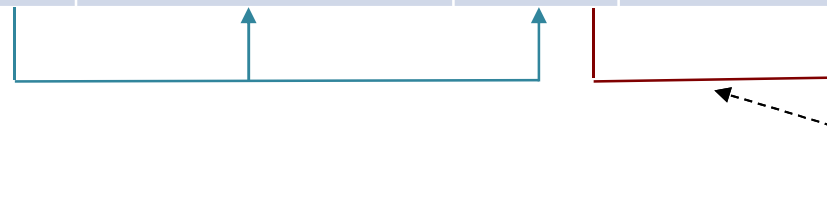
- A 3FN requer que não existam dependências funcionais de atributos não-chave que não sejam chaves candidatas, o que permite **evitar dados que não são “desejados”** numa dada tabela.



Exemplo de Aplicação da 3FN

Disciplinas

Disciplina Nr	Designação	Professor Nr	Professor
R34V12	Engenharia de Software	1	Ribeiro Teixeira
R34V12	Engenharia de Software	1	Ribeiro Teixeira
T34D12	Comunicações por Computador	2	Costa Santos
BD4653	Bases de Dados	3	Narciso Pereira
T34D12	Comunicações por Computador	2	Costa Santos
T34D12	Comunicações por Computador	2	Costa Santos
(...)	(...)	(...)	(...)



Dependência Transitiva



Resultado da Aplicação da 3FN

Disciplinas

Disciplina Nr	Designação	Professor Nr
R34V12	Engenharia de Software	1
BD4653	Bases de Dados	3
T34D1		
(...)		

Professores

Professor Nr	Professor
(...)	(...)
(...)	(...)
(...)	(...)
(...)	(...)

Disciplinas

3FN

Disciplinas

Professores



O Processo de Normalização

AlunoNr	Nome	Ano	Se	Disciplina Nr	Designação	Nota	Professor Nr	Professor
A12345	João Costa	2014	1	R34V12	Engenharia de Software	18	1	Ribeiro Teixeira
A45231	Ana Pascoal	2014	1	R34V12	Engenharia de Software	12	1	Ribeiro Teixeira
A45231	Ana Pascoal	2014	1	T34D12	Comunicações por Computador	15	2	Costa Santos
A12345	João Costa	2014	1	BD4653	Bases de Dados	10	3	Narciso Pereira
A98976	Soraia Rodrigues	2014	1	T34D12	Comunicações por Computador	15	2	Costa Santos
A98976	Soraia Rodrigues	2014	1	T34D12	Comunicações por Computador	15	2	Costa Santos
(...)	(...)	(...)	(...)	(...)	(...)	(...)	(...)	(...)

1FN

AlunoNr	Nome	Ano	Semestre
A12345	João Costa	2014	1
A45231	Ana Pascoal	2014	1
A98976	Soraia Rodrigues	2014	1
(...)	(...)	(...)	(...)

AlunoNr	Disciplina Nr	Designação	Nota	Professor Nr	Professor
A12345	R34V12	Engenharia de Software	18	1	Ribeiro Teixeira
A45231	R34V12	Engenharia de Software	12	1	Ribeiro Teixeira
A45231	T34D12	Comunicações por Computador	15	2	Costa Santos
A12345	BD4653	Bases de Dados	10	3	Narciso Pereira
A98976	T34D12	Comunicações por Computador	15	2	Costa Santos
A98976	T34D12	Comunicações por Computador	15	2	Costa Santos
(...)	(...)	(...)	(...)	(...)	(...)

2FN

AlunoNr	Disciplina Nr	Nota
A12345	R34V12	18
A45231	R34V12	12
A45231	T34D12	15
A12345	BD4653	10
A98976	T34D12	15
A98976	T34D12	15
(...)	(...)	(...)

Disciplina Nr	Designação	Professor Nr	Professor
R34V12	Engenharia de Software	1	Ribeiro Teixeira
R34V12	Engenharia de Software	1	Ribeiro Teixeira
T34D12	Comunicações por Computador	2	Costa Santos
BD4653	Bases de Dados	3	Narciso Pereira
T34D12	Comunicações por Computador	2	Costa Santos
T34D12	Comunicações por Computador	2	Costa Santos
(...)	(...)	(...)	(...)

3FN

Disciplina Nr	Designação	Professor Nr
R34V12	Engenharia de Software	1
BD4653	Bases de Dados	3
T34D12	Comunicações por Computador	2
(...)	(...)	(...)

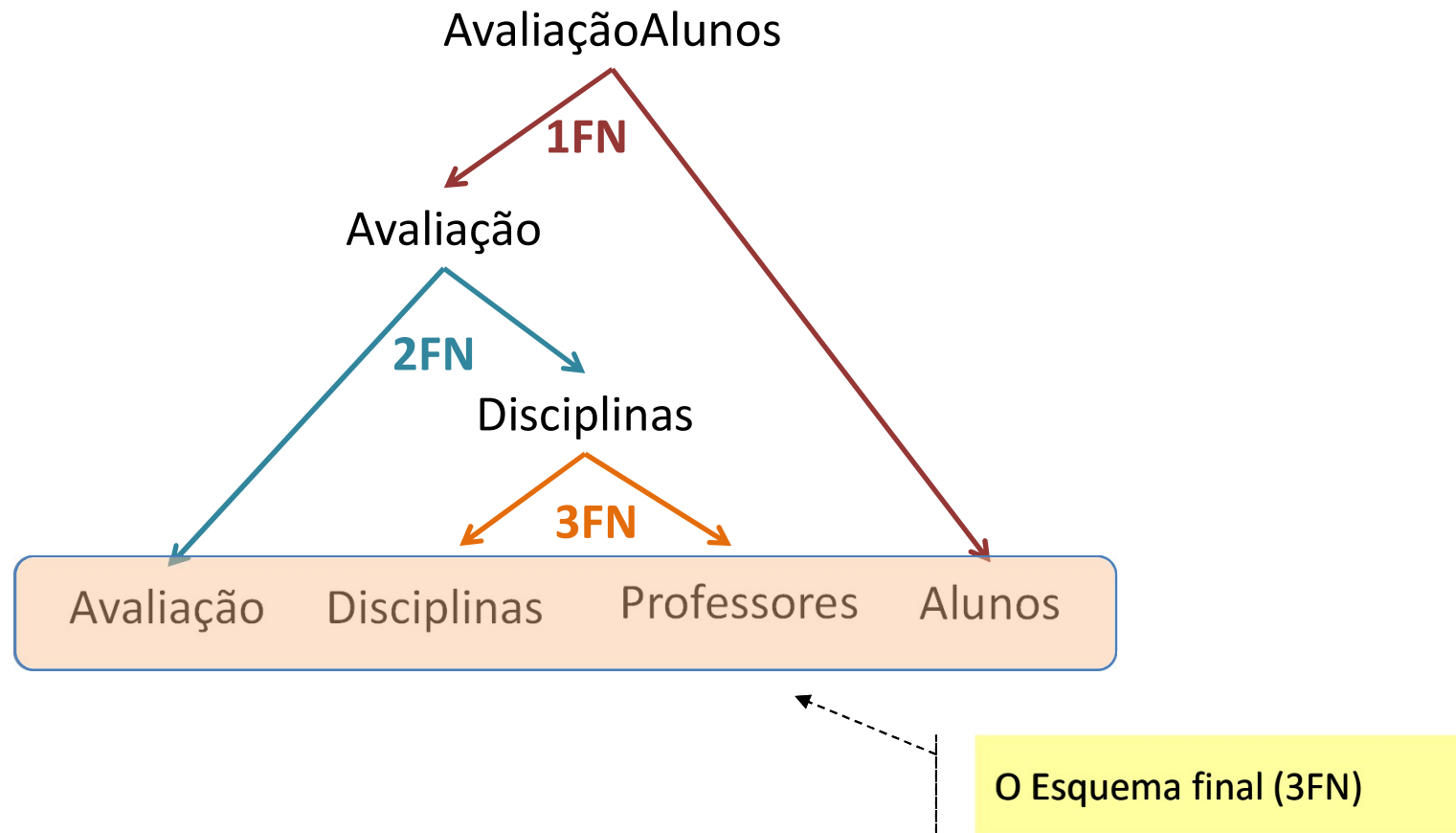
Professor Nr	Professor
1	Ribeiro Teixeira
2	Costa Santos
3	Narciso Pereira
(...)	(...)

Após ter sido normalizado até à 3FN, o esquema final integra 4 tabelas:

- Alunos
- Avaliação
- Disciplinas
- Professores



O Processo de Normalização Realizado



A Forma Normal de Boyce-Codd

- Uma tabela está na **forma normal de Boyce-Codd (FNBC)** (Boyce & Codd, 1974) se estiver na 3FN e se para cada dependência funcional não trivial $A1 \rightarrow A2$, $A1$ for uma super-chave:
 - Para cada dependência funcional $A1 \rightarrow A2$ pertencente ao fecho de dependências (F^+), $A2$ pertence a $A1$ (a dependência é trivial) ou $A1$ é uma super-chave.



A Forma Normal de Boyce-Codd

- A FNBC não permite pois a ocorrência de dependências entre atributos que pertençam a chaves candidatas.
- Esta forma normal é um refinamento da 3FN que, basicamente, invalida a possibilidade de um atributo funcionalmente dependente ser um atributo primo - como tal é mais restritiva que a 3FN.



Aplicação da FNBC

- A aplicação da FNBC só se aplicará se:
 - As tabelas tenham pelo menos duas chaves candidatas;
 - Que pelo menos uma chave candidata seja composta por dois ou mais atributos;
 - Que as chaves candidatas não sejam disjuntas, que tenham pelo menos um atributo em comum.



Exemplo de Aplicação da FNBC

ProjetosInvestigação

Projeto	ÁreaInvestigação	Professor
Sketching	Big Data	Xavier Garcia
Otimização Operacional	Bases de Dados	Ribeiro Teixeira

Sketching
Tunning ON
OffShore
Cypher
(...)

Assumindo as seguintes dependências funcionais:

(1) {Projeto, ÁreaInvestigação} → Professor

Cada projeto tem vários professores envolvidos e pode estar enquadrado em mais do que uma área de investigação; porém, um projeto apenas tem um professor por área de investigação.

(2) Professor → ÁreaInvestigação (não FNBC)

Cada professor desenvolve trabalho de investigação numa dada área de investigação;

Existem duas chaves candidatas: {Projeto, ÁreaInvestigação} e {Projeto, Professor}.



Resultado da Aplicação da FNBC

ProfessoresÁreaInvestigação

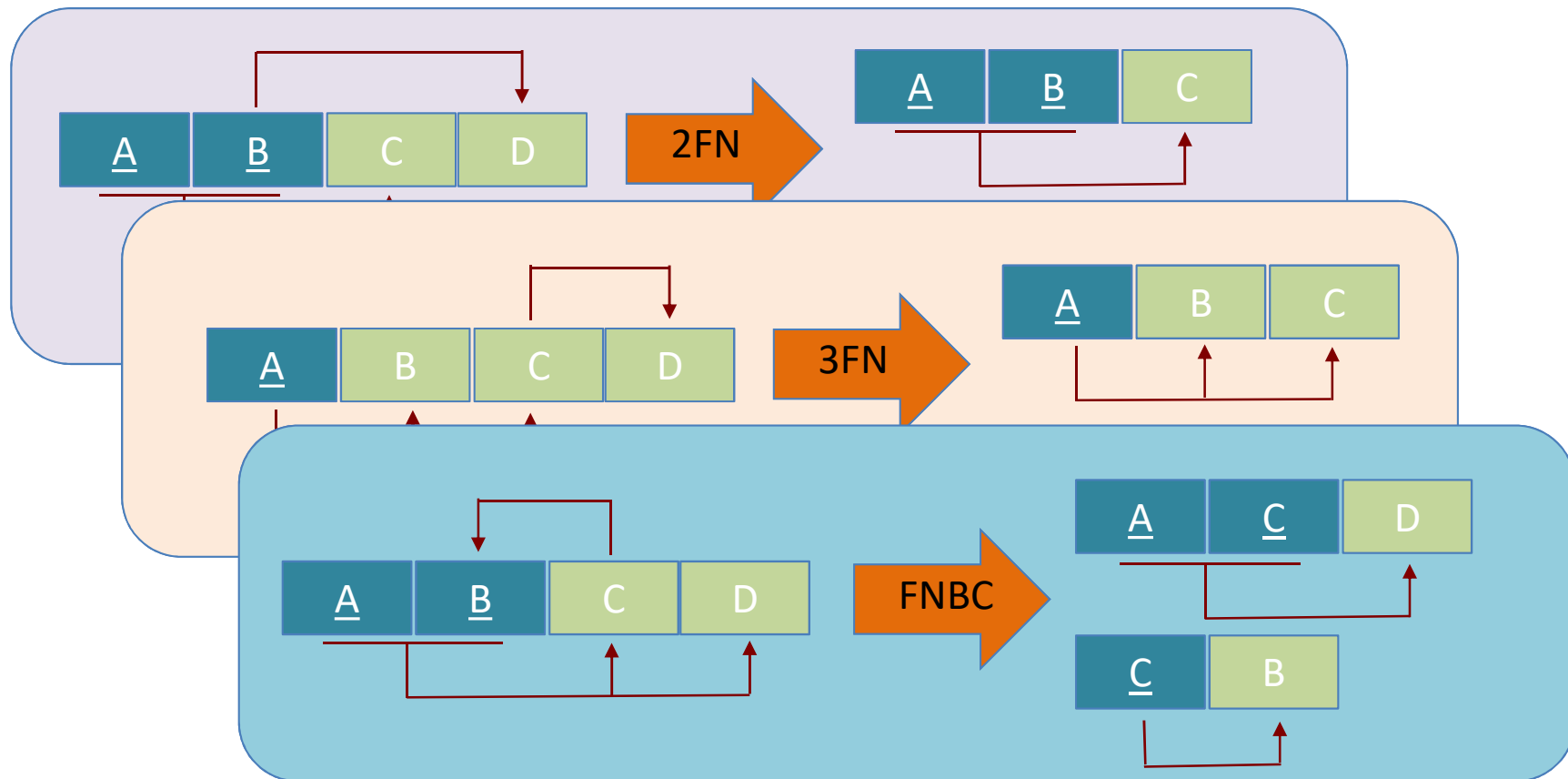
<u>Professor</u>	<u>ÁreaInvestigação</u>
Xavier Garcia	Big Data
Ribeiro Teixeira	Bases de Dados
Renata Prata	Cloud Computing
Costantino Santi	Comunicações por Computador
(...)	(...)

ProjetosProfessor

<u>Projeto</u>	<u>Professor</u>
Sketching	Xavier Garcia
Otimização Operacional	Ribeiro Teixeira
Sketching	Ribeiro Teixeira
Tunning ON	Renata Prata
OffShore	Renata Prata
Cypher	Costantino Santi
(...)	(...)



Reverendo as DF – 2FN, 3FN e FNBC



A Quarta Forma Normal (4FN)

- A aplicação da **quarta forma normal** (4FN) (Fagin, 1977) elimina relacionamentos muitos-para-um (N:1) entre atributos, envolvendo um tipo de dependência funcional mais geral: **a dependência multivalor**.
- Uma relação está na 4FN se estiver na FNBC e se não contiver mais do que um atributo multivalor, ou seja:
 - Se e só se, para cada uma dessas dependências multivalor triviais $X \twoheadrightarrow Y$, X é uma superchave, isto é, X é uma chave candidata ou um seu superconjunto.



Exemplo de Aplicação da 4FN

AlunosModalidadesLínguas

Aluno	Modalidade	Língua
Francisco	Atletismo	Português
Francisco	Ténis	Inglês
Francisco	Natação	NULL
Geraldina	Tiro	Alemão
Geraldina	NULL	Francês
António	Futebol	NULL
(...)	(...)	(...)

Aluno → Modalidade

Aluno → Língua

Alunos	Modalidade
Francisco	Atletismo
Francisco	Ténis
Francisco	Natação
Geraldina	Tiro
António	Futebol
(...)	(...)

Alunos	Língua
Francisco	Português
Francisco	Inglês
Geraldina	Alemão
Geraldina	Francês
(...)	(...)



A Quinta Forma Normal (5FN)

- A **quinta forma normal** (5FN) (Fagin, 1979) verifica-se quando todas as tabelas estão “partidas” em tantas relações quantas possíveis com o objetivo de evitar casos de redundância nos dados que contêm.
- A 5NF também conhecida por a **forma normal projeção-junção** (FNPJ) e tem como objetivo reduzir a redundância em bases de dados relacionais que armazenam factos multivalor isolando semanticamente relacionamentos múltiplos.
- Uma tabela diz-se que está na 5FN, se e só se qualquer dependência de junção não trivial definida nessa tabela pelas chaves candidatas – uma **dependência de junção** $\{A, B, \dots Z\}$ numa relação R é implicada pelas suas chaves candidatas se e só se cada $A, B, \dots, Z$ for uma chave em R .



A Quinta Forma Normal (5FN)

- Uma relação estando na 5FN não pode ser mais fragmentada em relações mais pequenas sem que os dados sejam modificados ou o seu próprio significado.
- Assim, a 5FN aborda, pois, os casos nos quais **a informação pode ser reconstruída** a partir de pequenas partes de informação que pode ser mantida com menos redundância.
- De referir que a 3FN e a 4FN também abordam a questão da redundância, porém a 5FN trata de casos que as anteriores não abordaram.



Exemplo de Aplicação da 5FN

EmpresaArtigosVendedores		
Empresa	Artigo	Agente
A Metalurgia Ideal	Fogareiros	Justino & Filhos, Lda.
A Metalurgia Ideal	Lareiras	Lareiras e Fogões, Lda.
Carvoeira do Tojo	Carvão	A Braseira, SA.
Quintas do Vale	Frango da Cerca	Frescos & Companhia, Lda.
Costa e Canto	Alf...	
Quintas do Vale	Fra...	
(...)	(...)	(...)

Porém, se tomarmos em consideração um requisito como, por exemplo: “se um dado agente representa uma empresa, então ele vende sem exceção todos os seus produtos”, como é que organizaríamos os nossos dados?



Exemplo de Aplicação da 5FN

EmpresasArtigos	
Empresa	Artigo
A Metalurgia Ideal	Fogareiros
A Metalurgia Ideal	Lareiras
Carvoeira do Tojo	Carvão
Quintas do Vale	Frango
Costa e Canto	Alface
(...)	(...)

EmpresasAgentes	
Empresa	Agente
A Metalurgia Ideal	Justino & Filhos, Lda.
A Metalurgia Ideal	Lareiras e Fogões, Lda.
Carvoeira do Tojo	A Braseira, SA.
Quintas do Vale	Frango da Cerca
Costa e Canto	(...)

ArtigosAgentes	
Artigo	Agente
Fogareiros	Justino & Filhos, Lda.
Lareiras	Lareiras e Fogões, Lda.
Carvão	A Braseira, SA.
Frango da Cerca	A Braseira, SA.
(...)	(...)

Estas três relações, que resultaram basicamente da aplicação de três projeções, estão na 5FN. A relação anterior não estava.



Como Melhorar um Esquema

- Existem algumas **ações** que podem ser realizadas no sentido de melhorar um esquema de uma base de dados, em particular:
 - Avaliar as definições das chaves primárias.
 - Avaliar as convenções de nomes que foram adotadas.
 - Refinar a atomicidade dos atributos.
 - Identificar novos atributos.
 - Identificar novos relacionamentos.



Como Melhorar um Esquema

- Refinar as chaves primárias de acordo com a granularidade estabelecida.
- Manter a correção histórica.
- Avaliar a utilização de atributos derivados.



Como Melhorar um Esquema

- Além das ações anteriores devemos ter também em consideração que:
 - As estruturas das tabelas usualmente são limpas para eliminar dependências problemáticas, parciais e transitivas.
 - A normalização, por si só, não garante a produção de bons esquemas para bases de dados, apesar de se ter que reconhecer o seu valor num processo de eliminação de dados redundantes.



Quando “não Normalizar”

- Há situações em que temos vantagens em **desnormalizar** uma parte de um esquema de uma base de dados, em particular quando:
 - Queremos manter as tabelas a um nível (simples) que permita satisfazer só por si as *queries* dos seus utilizadores.
 - Pretendemos evitar o processamento de mais do que uma tabela simultaneamente.
 - Precisamos de manter um qualquer sistema de arquivo.
 - Melhorar o desempenho do sistema em circunstâncias algo particulares.
 - (...)



Quando “não Normalizar”

- Queremos evitar a junção de um número grande de tabelas que envolve operações de I/O adicionais e mais processamento lógico, reduzindo consequentemente o desempenho do sistema.
- Pretendemos evitar potenciais conflitos em termos de eficiência de desenho e projeto, requisitos, velocidade de processamento.



Efeitos da Desnormalização

- As tabelas desnormalizadas numa base de dados que esteja em produção tende a apresentar alguns “defeitos”:
 - As operações de atualização de dados são menos eficientes, uma vez que os programas têm que manipular tabelas de maior dimensão.
 - As operações de indexação e, conseqüentemente, os índices tornam-se mais pesados.
 - As tabelas desnormalizadas impõe estratégias mais complicadas em situações em que é necessário criar e manter vistas – tabelas virtuais.



Algumas Recomendações

- A possibilidade de desnormalizar uma ou mais tabelas deve ser **analisada com cuidado**, avaliando as suas potenciais implicações sobre a base de dados e os as aplicações cliente.
- Em **casos particulares, de acordo com algumas circunstâncias**, as tabelas desnormalizadas são uma boa opção. Mas isso deve ser avaliado e comprovado.



Referências

- Beerli, C., Bernstein, P., Goodman, N., **A Sophisticate's Introduction to Database Normalization Theory**, In Proceedings of Fourth International Conference of very Large Databases, Berlin, Alemanha, 1978.
- Kent, W., **A Simple Guide to Five Normal Forms in Relational Database Theory**, Communications of the ACM, vol. 26, pp. 120–125, 1983
- Codd, E.F., **A relational model of data for large shared data banks**. In Communications of the ACM, Vol. 13 Issue 6, pp 377-387, June 1970.
- Codd, E.F., **Normalized Data Base Structure: A Brief Tutorial**, ACM SIGFIDET Workshop on Data Description, Access, and Control, Nov. 11-12, San Diego, California, 1971.



Referências

- Codd, E., **Further Normalization of the Data Base Relational Model**, R. Rustin (ed.), Data Base Systems (Courant Computer Science Symposia 6), Prentice-Hall, 1972.
- Connolly, T., Begg, C., **Database Systems: A Practical Approach to Design, Implementation, and Management**, 6th edition, Pearson, January, 2014.
- Date, C., **An Introduction to Database Systems**, 8th Edition, Pearson, July, 2003.
- Garcia-Molina, H., Ullman, J., Widom, J., **Database Systems: The Complete Book**, 2nd Edition, Pearson, June, 2008.
- Codd, E.F., **Recent Investigations into Relational Data Base Systems**, IBM Research Report RJ1385, April 23, 1974.



Referências

- Fagin, R., **Multivalued Dependencies and a New Normal Form for Relational Databases**, ACM Transactions on Database Systems 2 (3), Sept. 1977.
- Fagin, R., **Normal Forms and Relational Database Operators**, ACM SIGMOD International Conference on Management of Data , Boston, Mass, May 31-June 1, 1979.
- Fagin, R., **A Normal Form for Relational Databases That Is Based on Domains and Keys**, ACM Transactions on Database Systems. 6 (3): 387–415, 1981.
- Date, C.J., Darwen, H., Lorentzos, N., **Temporal data and the relational model**, Elsevier, 2002.



Referências Web

- http://en.wikipedia.org/wiki/Database_normalization
- <https://www.tes.com/lessons/eaxN1kTnVqp05w/ais275-week-5-normalization>
- <http://www.bkent.net/Doc/simple5.htm>
- <https://beginnersbook.com/2015/05/normalization-in-dbms/>
- <https://www.guru99.com/database-normalization.html#:~:text=Normalization%20is%20a%20database%20design,and%20links%20them%20using%20relationships>

