Universidade do Minho

Ano Letivo: 2023/24
Turnos: PL3/PL7

Bases de Dados

PL07 – Normalização e Álgebra Relacional

Docente: Diana Ferreira
Email: diana.ferreira@algoritmi.uminho.pt

Horário de Atendimento:
5ª feira 16h-17h

Sumário

Bibliografia:

- Connolly, T., Begg, C., Database Systems, A Practical Approach to Design, Implementation, and Management , Addison-Wesley, 4a Edição, 2004. **(Chapter 4/5; Chapter 14/15)**
- Belo, O., "Bases de Dados Relacionais: Implementação com MySQL", FCA – Editora de Informática, 376p, Set 2021. ISBN: 978-972-722-921-5.

Teoria da Normalização

Normalização de Dados

A normalização de dados baseia-se na análise das **chaves primárias** e das **dependências funcionais** de todos os seus atributos.

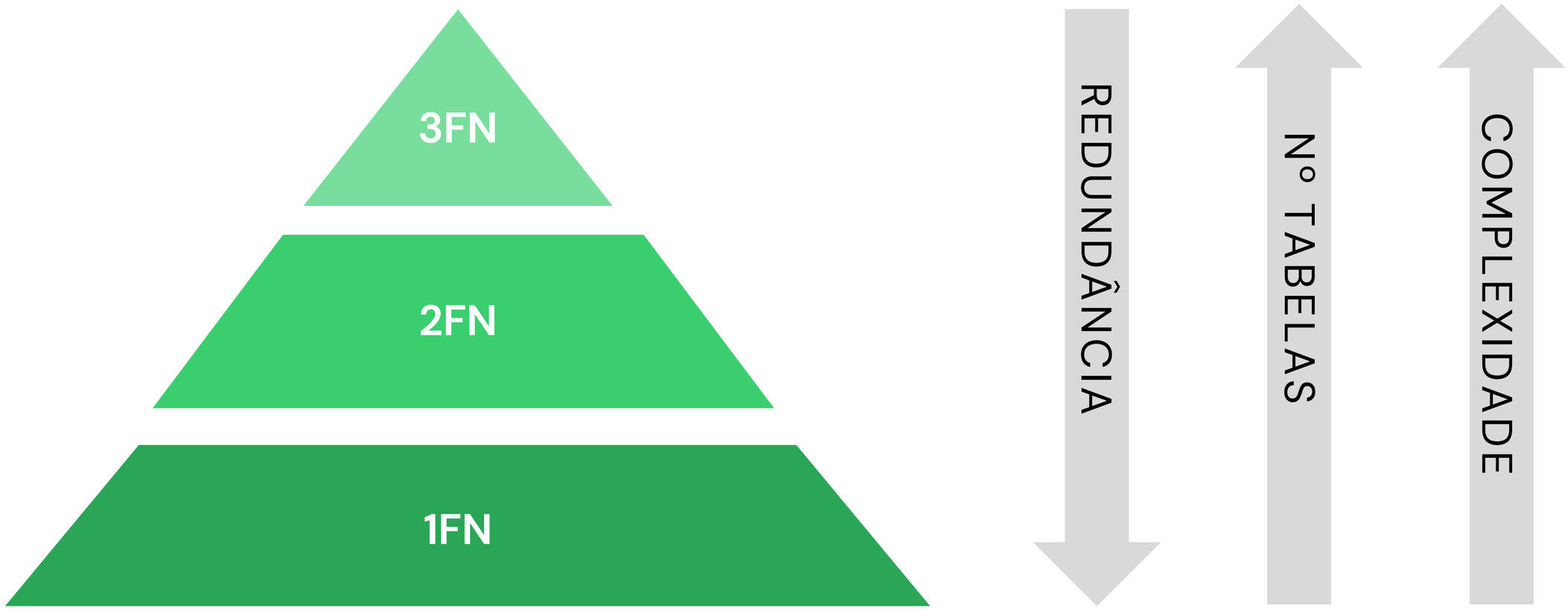
É um processo **progressivo**, que assenta na execução de uma série de etapas, cada uma delas correspondendo a uma **forma normal** específica com critérios de validação cada vez mais fortes.

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 duplicação de dados**, **melhorando a eficiência de armazenamento**, a **integridade** e a **escalabilidade** dos dados.

Teoria da Normalização

Formas Normais

O processo de normalização é **progressivo**, ou seja, cada um dos níveis superiores de normalização é um subconjunto do respetivo nível inferior.



Teoria da Normalização

Primeira Forma Normal – 1FN

Diz-se que uma relação está na 1FN se:

1. Possuir uma chave primária.
2. Todos os seus atributos forem atómicos. Não são permitidos atributos que implicitamente codificam subatributos (atributos compostos) ou atributos multivalor.
3. Não possuir grupos de dados repetitivos.

Na prática, podemos dizer que uma relação está na 1FN se as interseções entre colunas (atributos) e linhas (registos) possuírem um único valor – um valor atómico.

Teoria da Normalização

Primeira Forma Normal – 1FN

Aplicação da 1FN:

Passo 1: Uma das chaves candidatas é escolhida para chave primária.

Passo 2: Atributos multivalor são convertidos em novas relações com chave externa referindo a chave primária da tabela original.

Passo 3: Cada atributo composto é mapeado em vários sub-atributos atómicos.

Teoria da Normalização

Segunda Forma Normal – 2FN

Diz-se que uma relação está na segunda forma normal (2FN) se:

1. A relação estiver também na 1FN.

2. Todos os seus atributos não-primos forem **totalmente dependentes** da sua chave primária. Isto é, não podem existir **dependências parciais**. Diz-se que um atributo é não-primo quando este não faz parte de uma chave primária.

Teoria da Normalização

Segunda Forma Normal – 2FN

O que é uma dependência funcional?

As dependências funcionais determinam a forma como se pode interpretar e relacionar os dados e permitem especificar medidas formais sobre a correção dos esquemas relacionais.

Na prática, a dependência funcional **A1** → **A2** entre dois conjuntos de atributos de uma relação significa que:

- para cada valor de A1, existe apenas um valor possível para A2, por isso diz-se que A2 é funcionalmente dependente de A1 ou que A1 determina funcionalmente A2;

- valores iguais para A1, determinam valores iguais para A2.

Teoria da Normalização

Segunda Forma Normal – 2FN

Aplicando a análise de dependências funcionais:

Alunos(id_aluno, nome_aluno, cod_curso, nome_curso)

Notas(id_aluno, cod_dis, nome_dis, nota, cod_prof, nome_prof, dep_prof)

Diagrama de Dependências

id_aluno → nome_aluno, cod_curso

cod_curso → nome_curso

{id_aluno, cod_dis} → nota

cod_disc → nome_dis, cod_prof

cod_prof → nome_prof, dep_prof

A tabela Alunos está na 2FN, mas a Notas não!

Teoria da Normalização

Segunda Forma Normal – 2FN

Diagrama de Dependências

{id_aluno, cod_dis} → nota

cod_disc → nome_dis, cod_prof

cod_prof → nome_prof, dep_prof

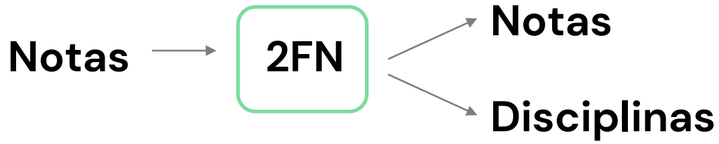
Notas(<u>id_aluno, cod_dis</u>, nome_dis, nota, cod_prof, nome_prof, dep_prof)

<u>id_aluno</u>	**<u>cod_dis</u>**	**nome_dis**	**nota**	**cod_prof**	**nome_prof**	**dep_prof**
001	D01	Bases de Dados	16	P01	Maria do Carmo	Dep. Informática
001	D02	Criptografia	12	P02	Paulo Gomes	Dep. Informática
002	D01	Bases de Dados	17	P01	Maria do Carmo	Dep. Informática
002	D03	Lógica Computacional	14	P03	Tiago Pinho	Dep. Sistemas
...	...	...	...	...	...	...

Dependência Parcial

Teoria da Normalização

Segunda Forma Normal – 2FN



Disciplinas(<u>cod_dis</u>, nome_dis, cod_prof, nome_prof, dep_prof)

Notas
<u>id_aluno</u>
<u>cod_dis</u>
nota

(001, ‘D01’, 16)
(001,’D02’, 12)
(002,’D01’, 17)
(002,’D03’, 14)
…

Disciplinas
<u>cod_dis</u>
nome_dis
cod_prof
nome_prof
dep_prof

Teoria da Normalização

Terceira Forma Normal – 3FN

Diz-se que uma relação está na 3FN se:

1. A relação estiver também na 1FN e na 2FN.
2. Todos os seus atributos que não sejam chaves primárias sejam mutuamente independentes, não havendo assim **dependências funcionais transitivas**. Por outras palavras, numa relação na 3FN, todos os atributos dependem única e exclusivamente da chave primária.

Na prática, isto significa que os atributos que não dependam da chave primária devem ser "eliminados" da relação, ou seja, devem ser transferidos para outra tabela.

Teoria da Normalização

Terceira Forma Normal – 3FN

Na prática, a dependência funcional **A1** → **A2, A3** e **A3** → **A4** entre os atributos de uma relação significa que:

- Existe uma dependência funcional transitiva entre A1 e A4. Ou seja os atributos que não são chave primária, não são mutuamente independentes entre si.

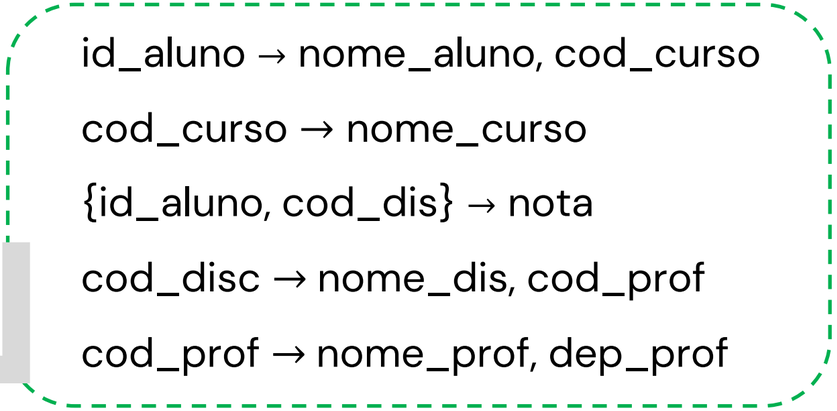
Aplicando a análise de dependências funcionais ao caso de estudo anterior:

Alunos(id_aluno, nome_aluno, cod_curso, nome_curso)

Notas(id_aluno, cod_dis, nota)

Disciplinas(cod_dis, nome_dis, cod_prof, nome_prof, dep_prof)

Diagrama de Dependências



A tabela Alunos e Disciplinas não estão na 3FN!

Teoria da Normalização

Terceira Forma Normal – 3FN

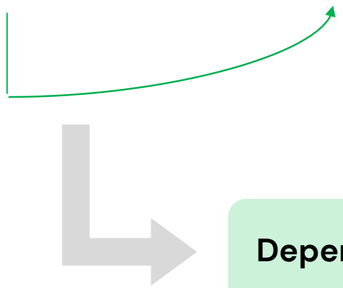
Diagrama de Dependências

id_aluno → nome_aluno, cod_curso

cod_curso → nome_curso

Alunos(id_aluno, nome_aluno, cod_curso, nome_curso)

id_aluno	nome_aluno	cod_curso	nome_curso
001	João Ferreira	C01	MIEI
001	João Ferreira	C01	MIEI
002	Rita Abreu	C01	MIEI
002	Rita Abreu	C01	MIEI
...	...	...	...



Dependência Transitiva

Teoria da Normalização

Terceira Forma Normal – 3FN

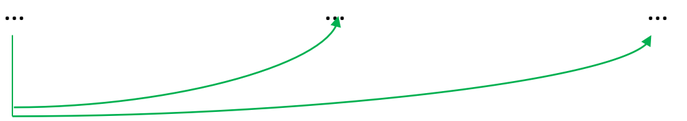
Diagrama de Dependências

cod_disc → nome_dis, cod_prof

cod_prof → nome_prof, dep_prof

Disciplinas(<u>cod_dis</u>, nome_dis, cod_prof, nome_prof, dep_prof)

<u>cod_dis</u>	**nome_dis**	**cod_prof**	**nome_prof**	**dep_prof**
DO1	Bases de Dados	PO1	Maria do Carmo	Dep. Informática
DO2	Criptografia	PO2	Paulo Gomes	Dep. Informática
DO1	Bases de Dados	PO1	Maria do Carmo	Dep. Informática
DO3	Lógica Computacional	PO3	Tiago Pinho	Dep. Sistemas
...	...	...	...	...



Dependência Transitiva

Teoria da Normalização

Terceira Forma Normal – 3FN

Alunos(id_aluno, nome_aluno, cod_curso)

Alunos
id_aluno
nome_aluno
cod_curso

Disciplinas(cod_dis, nome_dis, cod_prof)

Disciplinas
cod_dis
nome_dis
cod_prof

Cursos(cod_curso, nome_curso)

Cursos
cod_curso
nome_curso

Professores(cod_prof, nome_prof, dep_prof)

Professores
cod_prof
nome_prof
dep_prof

Teoria da Normalização

Terceira Forma Normal – 3FN

Alunos(<u>id_aluno</u>, nome_aluno, cod_curso)

Alunos
<u>id_aluno</u>
nome_aluno
cod_curso

Cursos(cod_curso, nome_curso)

Cursos
<u>cod_curso</u>
nome_curso

Diagrama de Dependências

id_aluno → nome_aluno, cod_curso

cod_curso → nome_curso

{id_aluno, cod_dis} → nota

cod_disc → nome_dis, cod_prof

cod_prof → nome_prof, dep_prof

Disciplinas
<u>cod_dis</u>
nome_dis
cod_prof

Disciplinas
<u>cod_prof</u>
nome_prof
dep_prof

As relações encontram-se na 3FN!

FASE 4: Modelação Lógica

Resolução de Exercícios

Ficha de Excercícios PL06:

Questão 2

Ficha de Excercícios PL07:

Questão 1, 2 e 3

Álgebra Relacional

O que é?

A Álgebra Relacional é uma linguagem teórica com operações que podem ser realizadas numa ou mais relações, definindo uma nova relação sem que as relações originais sejam modificadas.

Em álgebra relacional, podemos distinguir entre operações **unárias** (operações que ocorrem apenas sobre uma única relação) e **binárias** (operações que ocorrem sobre duas relações).

Álgebra Relacional

Operações

OPERAÇÕES UNÁRIAS

σ (“sigma”)	selecção
π (“pi”)	projecção
$\longleftarrow$	atribuição
δ (“delta”) ρ (“rho”)	renomeação
τ (“tau”)	ordenação
γ (“gama”)	agregação

OPERAÇÕES BINÁRIAS

$\times$	produto cartesiano
$\cup$	união
$\cap$	intersecção
$-$	diferença
$\bowtie$	junção
$\div /$	divisão/quociente

Álgebra Relacional

Operações de Junção

A operação de Junção é utilizada para combinação da informação contida entre duas ou mais tabelas. Uma junção pode ser definida como um produto cartesiano seguido por operações de seleção e de projeção.

Inner joins

$\bowtie$	junção natural
$\bowtie_{A\theta B}$	teta-junção
$\bowtie_{A=B}$	equi-junção
$\ltimes$	semi-junção

Outer joins

$\overset{\frown}{\bowtie}$	junção externa esquerda
$\overset{\frown}{\bowtie}$	junção externa direita
$\overset{\frown}{\bowtie}$	junção externa completa

Álgebra Relacional

Operadores

= → igual

<> → diferente

!= → diferente

< → inferior a

<= → igual ou inferior a

> → superior a

>= → igual ou superior a

∧ (AND) → para condições conjuntas

∨ (OR) → para condições disjuntas

¬ (NOT) → para negação de condições

NULL → para verificar o preenchimento de uma coluna

IN → para determinar se um valor especificado corresponde a qualquer valor de uma lista de valores

BETWEEN → para determinar se um valor está contido num intervalo de valores

LIKE → para consultar dados com base num padrão especificado (% → qualquer sequência de zero ou mais caracteres; _ → caracter único).

Álgebra Relacional

Funções de Datas

CURDATE – Retorna a data atual;

NOW/ SYSDATE – Retorna a data e hora atuais;

DAY - Obtém o dia do mês de um DATE/DATETIME;

DAYOFWEEK – Obtém o índice do dia da semana de um DATE/DATETIME;

MONTH - Retorna um inteiro que representa o mês de um DATE/DATETIME;

WEEK - Retorna um número de semana de um DATE/DATETIME;

WEEKDAY - Retorna um índice de dia da semana para um DATE/DATETIME;

YEAR - Retorna o ano de um DATE/DATETIME;

HOUR – Retorna a hora de um DATETIME/TIME;

MINUTE – Retorna os minutos de um DATETIME/TIME;

SECOND – Retorna os segundos de um DATETIME/TIME;

Álgebra Relacional

Funções de Datas

DATEDIFF – Calcula o número de dias entre dois valores DATE/DATETIME;

DATE_ADD – Adiciona um valor de tempo a um valor DATE/DATETIME;

DATE_SUB – Subtrai um valor de tempo a um valor DATE/DATETIME;

DATE_FORMAT – Formata um valor de data com base em um formato de data especificado;

STR_TO_DATE – Converte uma string num valor de data e hora com base num formato especificado;

TIMEDIFF – Calcula a diferença entre dois valores DATETIME/TIME;

TIMESTAMPDIFF – Calcula a diferença entre dois valores DATE/DATETIME.

Álgebra Relacional

Operações

EXEMPLOS:

- Quais são os pacientes de Braga?

$$\sigma_{localidade="Braga"}\ (pacientes)$$

- Liste os pacientes que não são do sexo masculino, indicando o seu nome e sexo.

$$\pi_{nome,\ sexo}\ (\sigma_{sexo\neq"M"}\ (pacientes))$$

- Renomear os atributos dos procedimentos para codigo, descricao e valor.

$$\rho_{(codigo,\ descricao,\ valor)}\ (procedimentos)$$

ou

$$\delta_{cod\_proc\leftarrow codigo,des\_proc\leftarrow descricao,preco\leftarrow valor)}\ (procedimentos)$$

Álgebra Relacional

Operações

EXEMPLOS:

- Liste os pacientes com menos de 18 anos.

$$\sigma_{TIMESTAMPDIFF(YEAR,dta\_nascimento,CURDATE())<18}\ (pacientes)$$

- Qual é o ano de validade da última prescrição?

$$\pi_{YEAR(data\_validade)}\ (\sigma_{MAX(data\_prescricao)}\ (prescricao))$$

- Quantas consultas estão agendadas para Maio de 2024?

$$\rho_{nr\_consultas}(\gamma_{COUNT(*)}\ (\sigma_{YEAR(data\_agenda)=2024 \wedge MONTH(data\_agenda)=05}\ (consultas)))$$

Álgebra Relacional

Operações

EXEMPLOS:

- Liste os pacientes do mais jovem para o mais velho cujo ano de nascimento seja igual ou superior a 1990.

$$\tau_{dta\_nascimento\ DESC}\ (\sigma_{YEAR(dta\_nascimento)\geq 1990}(\text{pacientes})$$

- Qual é o valor máximo e mínimo dos procedimentos?

$$\gamma_{MAX(preco),\ MIN(preco)}\ (\text{procedimentos})$$

- Liste as prescrições efetuadas e respetivos fármacos.

$$medicamentos \bowtie prescricoes$$

- Liste os emails, tanto dos pacientes como dos funcionários, numa única relação.

$$emails\_pac \cup emails\_func$$

Álgebra Relacional

Operações

EXEMPLOS:

- Quais os médicos que deram consultas?

$$R_0 \leftarrow \delta_{id\_medico \leftarrow num\_mec}(consultas)$$

$$R \leftarrow Medicos \ltimes R_0$$

- Liste os nomes dos pacientes de Braga que não foram consultados.

$$R_0 \leftarrow \sigma_{localidade="Braga"}(pacientes)$$
$$R_1 \leftarrow \pi_{nome}(R_0)$$

Nomes dos Pacientes de Braga

$$R_2 \leftarrow pacientes \bowtie_{pacientes.nr\_sequencial=consultas.id\_paciente} consultas$$
$$R_3 \leftarrow \pi_{nome}(R_2)$$

Nomes dos pacientes que foram consultados

$$R_4 \leftarrow R_1 - R_3$$

FASE 4: Modelação Lógica

Resolução de Exercícios

Ficha de Excercícios PL07:

Questão 4