

Álgebra relacional

Prof.: Maiquel de Brito

BLU3024

Departamento de Engenharia de Controle, Automação e Computação (CAC)

UFSC Blumenau

1. Introdução
2. Relações
3. Teoria dos conjuntos
4. Álgebra Relacional

Introdução

- O que é um banco de dados?

- O que é um banco de dados?
- Quando usar banco de dados?

- O que é um banco de dados?
- Quando usar banco de dados?
- Por que usar banco de dados?

- O que é um banco de dados?
- Quando usar banco de dados?
- Por que usar banco de dados?

Banco de dados:

“... coleção de dados inter-relacionados, representando informações sobre um domínio específico”

SGBD (ou RDBMS) i.e. Sistema Gerenciador de Banco de Dados

software que possui recursos para manipular as informações contidas em um banco de dados

Ex.: MySQL, MariaDB, PostgreSQL, Oracle, SQL Server, DB2, Paradox, Access etc.

Relações

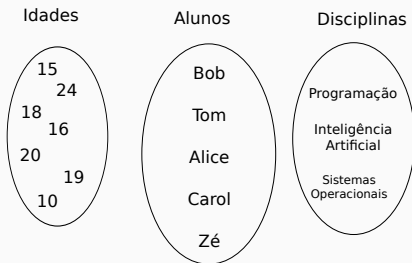
Teoria dos conjuntos

Teoria dos conjuntos

Conjuntos: agrupam elementos pertencentes à realidade

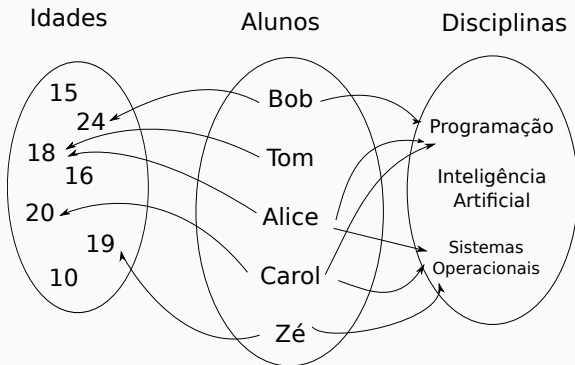
Exemplos:

- Conjunto de pessoas = {bob, tom, alice, carol, zé}
- Conjunto de disciplinas: {Programação, Inteligência Artificial, Sistemas Operacionais}
- Conjunto de idades: $\{x \in \mathbb{Z} | 0 \leq x \leq 100\}$



Relações

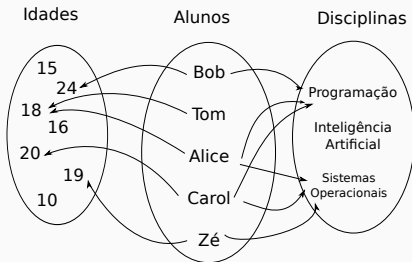
Relações: são combinações entre elementos de dois conjuntos



Qual disciplina tem mais alunos acima dos 20 anos?

Relações

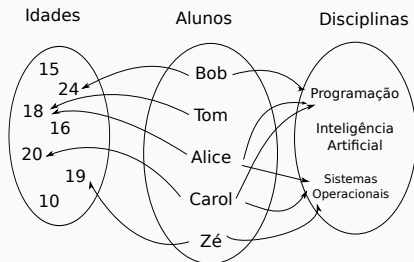
Aluno X Idade	
Aluno	Idade
Bob	24
Tom	18
Alice	18
Alice	18
Carol	20
Carol	20
Zé	19



Relações

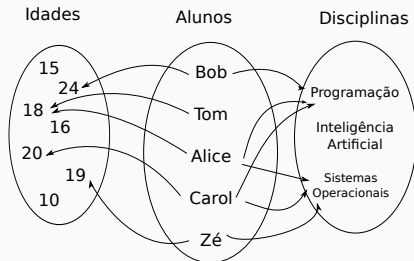
Aluno X Disciplina

Aluno	Disciplina
Bob	Programação
Alice	Programação
Alice	Sistemas Operacionais
Carol	Programação
Carol	Sistemas Operacionais
Zé	Sistemas Operacionais



Relações

Aluno	Idade	Disciplina
Bob	24	Programação
Tom	18	Programação
Alice	18	Programação
Alice	18	Sistemas Operacionais
Carol	20	Programação
Carol	20	Sistemas Operacionais
Zé	19	Sistemas Operacionais



Álgebra Relacional

Álgebra relacional: define operações sobre relações

- seleção
- projeção
- união
- diferença
- produto cartesiano
- rename

Álgebra Relacional - Seleção

$\sigma_{predicado}(relação)$: retorna as tuplas da *relação* que atendem ao *predicado*

Relação *Aluno*

Id	Nome	Idade	Disciplina	Professor	Nota 1	Nota 2	Nota 3
1	Bob	24	Programação	Yoda	9,5	8,5	9,0
2	Tom	18					
3	Alice	18	Programação	Yoda	7,5	9,5	8,0
4	Alice	18	Sistemas Operacionais	Gandalf	5,5	9,5	9,0
5	Carol	20	Programação	Yoda	9,0	8,0	9,5
6	Carol	20	Sistemas Operacionais	Gandalf	9,0	8,0	9,0
7	Zé	19	Sistemas Operacionais	Gandalf	7,5	8,5	8,0

$\sigma_{(Idade \geq 20)}(Aluno)$:

1	Bob	24	Programação	Yoda	9,5	8,5	9,0
5	Carol	20	Programação	Yoda	9,0	8,0	9,5
6	Carol	20	Sistemas Operacionais	Gandalf	9,0	8,0	9,0

Álgebra Relacional - Seleção

$\sigma_{predicado}(\text{relação})$: predicado admite comparações ($=, \neq, <, \leq, >, \geq$) e conectivos lógicos \wedge (“e”), \vee (“ou”) e \neg (“não”)

Relação *Aluno*

Id	Nome	Idade	Disciplina	Professor	Nota 1	Nota 2	Nota 3
1	Bob	24	Programação	Yoda	9,5	8,5	9,0
2	Tom	18					
3	Alice	18	Programação	Yoda	7,5	9,5	8,0
4	Alice	18	Sistemas Operacionais	Gandalf	5,5	9,5	9,0
5	Carol	20	Programação	Yoda	9,0	8,0	9,5
6	Carol	20	Sistemas Operacionais	Gandalf	9,0	8,0	9,0
7	Zé	19	Sistemas Operacionais	Gandalf	7,5	8,5	8,0

$\sigma_{(Idade < 20 \wedge nota3 \geq 9)}(\text{Aluno})$:

Id	Nome	Idade	Disciplina	Professor	Nota 1	Nota 2	Nota 3
4	Alice	18	Sistemas Operacionais	Gandalf	5,5	9,5	9,0

Álgebra Relacional - Projeção

$\Pi_{campo_1, \dots, campo_n}(relação):$

retorna os campos $campo_1, \dots, campo_n$ das tuplas de *relação*

Relação *Alunos*

Id	Nome	Idade	Disciplina	Professor	Nota ₁	Nota ₂	Nota ₃
1	Bob	24	Programação	Yoda	9,5	8,5	9,0
2	Tom	18					
3	Alice	18	Programação	Yoda	7,5	9,5	8,0
4	Alice	18	Sistemas Operacionais	Gandalf	5,5	9,5	9,0
5	Carol	20	Programação	Yoda	9,0	8,0	9,5
6	Carol	20	Sistemas Operacionais	Gandalf	9,0	8,0	9,0
7	Zé	19	Sistemas Operacionais	Gandalf	7,5	8,5	8,0

$\Pi_{(Nome, Idade, Nota_1)}(Alunos):$

Bob	24	9,5
Tom	18	
Alice	18	7,5
Alice	18	5,5
Carol	20	9,0
Carol	20	9,0
Zé	19	7,5

Composição de operadores relacionais

Ex.: encontrar o nome dos alunos do professor Yoda.

Relação *Alunos*

Id	Nome	Idade	Disciplina	Professor	Nota ₁ ...
1	Bob	24	Programação	Yoda	9,5
2	Tom	18			
3	Alice	18	Programação	Yoda	7,5
4	Alice	18	Sistemas Operacionais	Gandalf	5,5
5	Carol	20	Programação	Yoda	9,0
6	Carol	20	Sistemas Operacionais	Gandalf	9,0
7	Zé	19	Sistemas Operacionais	Gandalf	7,50

$\Pi_{(Nome)}(Alunos)$:

Nome
Bob
Tom
Alice
Alice
Carol
Carol
Zé

Composição de operadores relacionais

Ex.: encontrar o nome dos alunos do professor Yoda.

Relação *Alunos*

$\Pi_{(Nome)}(\sigma_{professor=Yoda}(Alunos))$:

Id	Nome	Idade	Disciplina	Professor	Nota ₁ ...
1	Bob	24	Programação	Yoda	9,5
2	Tom	18			
3	Alice	18	Programação	Yoda	7,5
4	Alice	18	Sistemas Operacionais	Gandalf	5,5
5	Carol	20	Programação	Yoda	9,0
6	Carol	20	Sistemas Operacionais	Gandalf	9,0
7	Zé	19	Sistemas Operacionais	Gandalf	7,50

Nome
Bob
Alice
Carol

Relação₁ \cup Relação₂

Relação: Alunos

Nome	Idade
Bob	24
Tom	18
Alice	18
Carol	20
Zé	19

Relação: Professores

Nome	Idade
Yoda	900
Gandalf	3.000
Neo	25

Alunos \cup Professores

Nome	Idade
Bob	24
Tom	18
Alice	18
Carol	20
Zé	19
Yoda	900
Gandalf	3.000
Neo	25

Álgebra relacional - Diferença

$Relação_1 - Relação_2$

Retorna uma relação com as tuplas que estão $Relação_1$ e que **não estão** em $Relação_2$

Relação: Alunos

Nome	Idade
Bob	24
Tom	18
Alice	18
Carol	20
Zé	19
Neo	25

Relação: Professores

Nome	Idade
Yoda	900
Gandalf	3.000
Neo	25

Alunos – Professores

Nome	Idade
Bob	24
Tom	18
Alice	18
Carol	20
Zé	19

Álgebra relacional - Produto Cartesiano

$Relação_1 \times Relação_2$

Combina, em uma única relação, as informações de $Relação_1$ e $Relação_2$

Relação: Alunos

Nome	Idade	Curso
Bob	24	Computação
Tom	18	Engenharia
Alice	18	Matemática

Relação: Professores

Nome	Idade	Disciplina
Yoda	900	Programação
Gandalf	3.000	S.O.

$Alunos \times Professores$

Nome	Idade	Curso	Nome	Idade	Disciplina
Bob	24	Computação	Yoda	900	Programação
Tom	18	Engenharia	Yoda	900	Programação
Alice	18	Matemática	Yoda	900	Programação
Bob	24	Computação	Gandalf	3.000	S.O.
Tom	18	Engenharia	Gandalf	3.000	S.O.
Alice	18	Matemática	Gandalf	3.000	S.O.

Álgebra relacional - Produto Cartesiano

$Relação_1 \times Relação_2$

Combina, em uma única relação, as informações de $Relação_1$ e $Relação_2$

- Seja $|R|$ o número de tuplas de uma relação R .
 $|Relação_1 \times Relação_2| = |Relação_1| \times |Relação_2|$
- Seja $|R|_c$ o número de colunas de uma relação R .
 $|Relação_1 \times Relação_2|_c = |Relação_1|_c + |Relação_2|_c$
- A operação $R \times R$ é possível para qualquer relação R .

Relação: Professores

$Professores \times Professores$

Nome	Idade	Professor.Nome	Professor.Idade	Professor1.Nome	Professor1.Idade
Yoda	900	Yoda	900	Yoda	900
Gandalf	3.000	Yoda	900	Gandalf	3.000
		Gandalf	3.000	Gandalf	3.000
		Gandalf	3.000	Yoda	900

$\rho_x(E)$: retorna o resultado da expressão relacional E sob o nome x .

Professores

Nome	Idade	Disciplina
Yoda	900	Programação
Gandalf	3.000	S.O.
Neo	25	Algoritmos

$\rho_{DiscProf}(\Pi_{Disciplina}(Professores))$

DiscProf
Programação
S.O.
Algoritmos

Sejam E_1 e E_2 duas expressões em álgebra relacional.

- $\sigma_p(E_1)$: retorna as tuplas de E_1 que satisfazem o predicado p
- $\Pi_S(E_1)$: retorna os atributos $s \in S$ de todas as tuplas de E_1
- $\rho_x(E_1)$: retorna E_1 sob o nome x
- $E_1 \cup E_2$: retorna todas as tuplas de E_1 e todas as tuplas de E_2
- $E_1 - E_2$: retorna todas as tuplas que estão em E_1 mas não estão em E_2
- $E_1 \times E_2$: retorna a combinação das tuplas de E_1 e E_2

Álgebra Relacional - Intersecção

$E_1 \cap E_2$: Retorna as tuplas que estão tanto na relação E_1 quanto na relação E_2 .

Relação: Alunos

Nome	Idade
Bob	24
Tom	18
Alice	18
Carol	20
Zé	19
Neo	25

Relação: Professores

Nome	Idade
Yoda	900
Gandalf	3.000
Neo	25

$Alunos \cap Professores$

Nome	Idade
Neo	25

Álgebra Relacional - Join

$E_1 \bowtie_{\theta} E_2$: Retorna, de $E_1 \times E_2$, as tuplas que satisfazem a condição θ

Relação: Professores		
Nome	Idade	CodDisc
Yoda	900	2
Gandalf	3.000	3
Neo	25	

Relação: Disciplinas	
Código	Nome
1	Algoritmos
2	S.O
3	Programação

Professores $\bowtie_{Professor.CodDisc=Disciplina.Codigo}$ *Disciplinas*

Professor.Codigo	Professor.Idade	Professor.CodDisc	Disciplinas.Codigo	Disciplinas.Nome
Yoda	900	2	2	SO
Gandalf	3.000	3	3	Programação

$E_1 \bowtie_{\theta} E_2$: Retorna, de $E_1 \times E_2$, as tuplas que satisfazem a condição θ

- $E_1 \bowtie_{\theta} E_2 = \sigma_{\theta}(E_1 \times E_2)$

Álgebra Relacional - Left Outer Join

$E_1 \bowtie_{\theta} E_2$: Retorna, de $E_1 \times E_2$, as tuplas que satisfazem a condição θ , bem como as de E_1 que não satisfazem θ

Relação: Professores

Nome	Idade	CodDisc
Yoda	900	2
Gandalf	3.000	3
Neo	25	

Relação: Disciplinas

Código	Nome
1	Algoritmos
2	S.O
3	Programação

$Professores \bowtie_{Professor.CodDisc=Disciplina.Codigo} Disciplinas$

Professor.Codigo	Professor.Idade	Professor.CodDisc	Disciplinas.Codigo	Disciplinas.Nome
Yoda	900	2	2	SO
Gandalf	3.000	3	3	Programação
Neo	25	null	null	null

Álgebra Relacional - Right Outer Join

$E_1 \bowtie_{\theta} E_2$: Retorna, de $E_1 \times E_2$, as tuplas que satisfazem a condição θ , bem como as de E_2 que não satisfazem θ

Relação: Professores

Nome	Idade	CodDisc
Yoda	900	2
Gandalf	3.000	3
Neo	25	

Relação: Disciplinas

Código	Nome
1	Algoritmos
2	S.O
3	Programação

$Professores \bowtie_{Professor.CodDisc=Disciplina.Codigo} Disciplinas$

Professor.Codigo	Professor.Idade	Professor.CodDisc	Disciplinas.Codigo	Disciplinas.Nome
Yoda	900	2	2	SO
Gandalf	3.000	3	3	Programação
null	null	null	1	Algoritmos

Álgebra Relacional - Full Outer Join

$E_1 \bowtie_{\theta} E_2$: Retorna, de $E_1 \times E_2$, as tuplas que satisfazem a condição θ , bem como as de E_1 e E_2 que não satisfazem θ

Relação: Professores

Nome	Idade	CodDisc
Yoda	900	2
Gandalf	3.000	3
Neo	25	

Relação: Disciplinas

Código	Nome
1	Algoritmos
2	S.O
3	Programação

$Professores \bowtie_{Professor.CodDisc=Disciplina.Codigo} Disciplinas$

Professor.Codigo	Professor.Idade	Professor.CodDisc	Disciplinas.Codigo	Disciplinas.Nome
Yoda	900	2	2	SO
Gandalf	3.000	3	3	Programação
Neo	25	null	null	null
null	null	null	1	Algoritmos