

Álgebra Relacional

■ Tópicos:

- * Definição da Álgebra Relacional
- * Operadores básicos da Álgebra Relacional
- * Operadores adicionais
- * Operações estendidas em Álgebra Relacional
- * Modificações de Bases de Dados
- * Vistas

■ Bibliografia:

- * Secções 2.5, 2.6 e 6.1 do livro recomendado
- * Capítulos 2 e 3 do livro *The theory of relational databases*

Definição Formal

- Uma expressão básica na álgebra relacional é:
 - ✦ Uma relação na base de dados
 - ✦ Uma relação constante
- Sejam E_1 e E_2 expressões de álgebra relacional; então todas as expressões abaixo são expressões de álgebra relacional:
 - ✦ $E_1 \cup E_2$
 - ✦ $E_1 - E_2$
 - ✦ $E_1 \times E_2$
 - ✦ $\sigma_P(E_1)$, P é um predicado nos atributos de E_1
 - ✦ $\Pi_S(E_1)$, S é uma lista com alguns dos atributos de E_1
 - ✦ $\rho_x(E_1)$, x é um novo nome para o resultado de E_1

Operações Adicionais

- Definem-se outras operações que não aumentam o poder expressivo da álgebra relacional, mas simplificam algumas consultas habituais.
 - ✦ Interseção de conjuntos
 - ✦ Junção Natural
 - ✦ Divisão
 - ✦ Atribuição

Operação de Interseção de Conjuntos

- Notação:

$$r \cap s$$

- Definida por:

$$r \cap s = \{ t \mid t \in r \text{ e } t \in s \}$$

- Assume:

- ✦ r, s têm a mesma aridade
- ✦ Os atributos de r e s são compatíveis

- Notar que:

$$r \cap s = r - (r - s) = s - (s - r)$$

Interseção de Conjuntos - Exemplo

- Relações r, s :

r		s	
A	B	A	B
α	1	α	2
α	2	β	3
β	1		

- $r \cap s$

$r \cap s$	
A	B
α	2

Operação de Junção Natural

- Notação: $r \bowtie s$
- Sejam r e s relações nos esquemas R e S , respetivamente. O resultado é uma relação no esquema $R \cup S$ que é obtido considerando cada par de tuplos t_r de r e t_s de s .
- Se t_r e t_s têm o mesmo valor em cada um dos atributos em $R \cap S$, um tuplo t é adicionado ao resultado, em que
 - ✦ t tem os mesmos valores que t_r em r
 - ✦ t tem os mesmos valores que t_s em s
- Exemplo: $R = (A, B, C, D)$ $S = (E, B, D)$
 - ✦ Esquema resultado: (A, B, C, D, E)
 - ✦ $r \bowtie s$ define-se por:

$$\Pi_{r.B, r.D, r.A, r.C, s.E} (\sigma_{r.B = s.B \wedge r.D = s.D} (r \times s))$$

Junção Natural – Exemplo

■ Relações r, s:

r

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

s

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ϵ

■ $r \bowtie s$

$r \bowtie s$

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

Exemplo Clínica

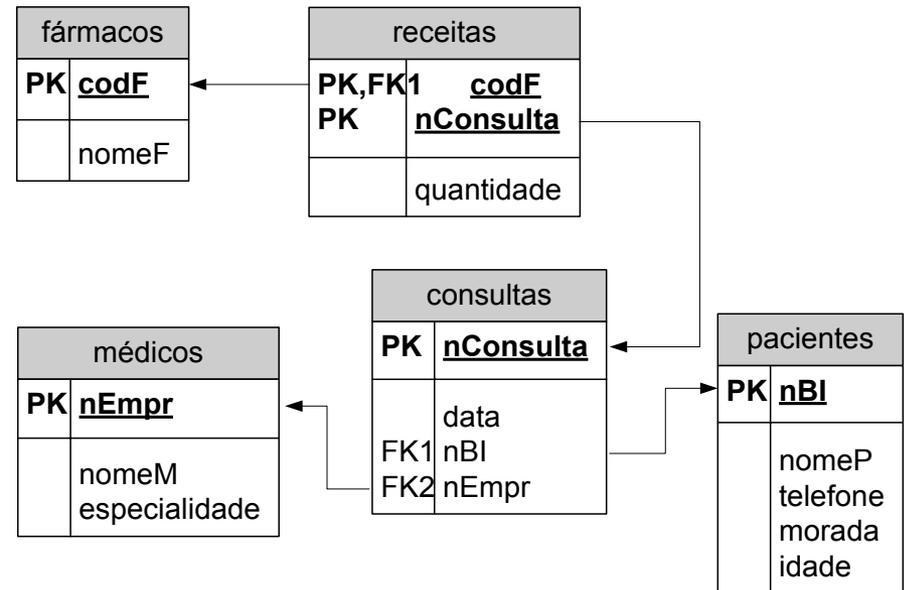
$médicos(nEmpr, nomeM, especialidade)$

$fármacos(codF, nomeF)$

$pacientes(nBI, nomeP, telefone, morada, idade)$

$consultas(nConsulta, data, nBI, nEmpr)$

$receitas(codF, nConsulta, quantidade)$



- Quais os (nomes dos) fármacos alguma vez prescritos por cardiologistas?

$\Pi_{nomeF} (fármacos \bowtie receitas \bowtie consultas \bowtie \sigma_{esp = "car..."} (médicos))$

ou

$\Pi_{nomeF} (\sigma_{esp = "car..."} (fármacos \bowtie receitas \bowtie consultas \bowtie médicos))$

Operação de Divisão

- Notação: $r \div s$
- Adequada para consultas que incluam a frase “para todo”.
- Sejam r e s relações nos esquemas R e S respetivamente, com
 - ★ $R = (A_1, \dots, A_m, B_1, \dots, B_n)$
 - ★ $S = (B_1, \dots, B_n)$
- O resultado de $r \div s$ é uma relação no esquema

$$R - S = (A_1, \dots, A_m)$$

e

$$r \div s = \{ t \mid t \in \Pi_{R-S}(r) \wedge \forall u \in s : tu \in r \}$$

onde tu é o tuplo resultante da concatenação dos tuplos t e u

Operação de Divisão – Exemplo

r

A	B
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
ϵ	6
ϵ	1
β	2

s

B
1
2

r ÷ s

A
α
β

Outro exemplo de divisão

r

A	B	C	D	E
α	a	α	a	1
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

s

D	E
a	1
b	1

r ÷ s

A	B	C
α	a	γ
γ	a	γ

Exemplo Clínica

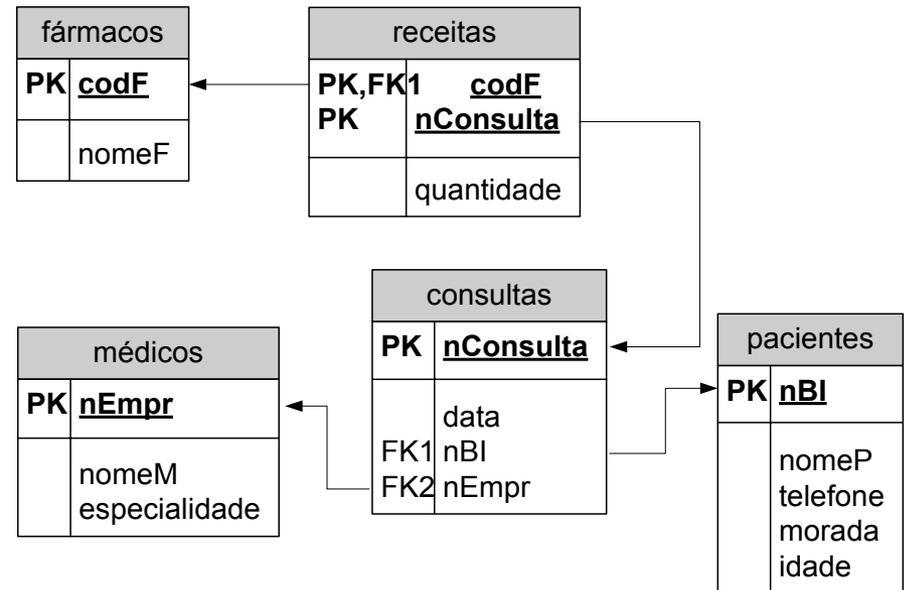
médicos(*nEmpr*, *nomeM*, *especialidade*)

fármacos(*codF*, *nomeF*)

pacientes(*nBl*, *nomeP*, *telefone*, *morada*, *idade*)

consultas(*nConsulta*, *data*, *nBl*, *nEmpr*)

receitas(*codF*, *nConsulta*, *quantidade*)



- Quais os códigos dos fármacos que já foram receitados por todos os médicos da clínica?

$$\prod_{codF, nEmpr} (\text{consultas} \times \text{receitas}) \div \prod_{nEmpr} (\text{médicos})$$

Operação de Divisão (Cont.)

■ Propriedade

★ Seja $q = r \div s$

★ Então q é a maior relação satisfazendo $q \times s \subseteq r$

■ Definição em termos de operações básicas da álgebra relacional Sejam $r(R)$ e $s(S)$ relações, com $S \subseteq R$

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - r)$$

Porquê?

★ $\Pi_{R-S}(r) \times s$ dá os elementos de r com todos os valores de S

★ $\Pi_{R-S}(\Pi_{R-S}(r) \times s) - r$ dá os tuplos t em $\Pi_{R-S}(r)$ tal que para algum tuplo $u \in s$, $tu \notin r$

★ Finalmente, a expressão toda dá os restantes tuplos, ou seja aqueles tal que para todo $u \in s$, se verifica que $tu \in r$

Operação de Atribuição

- A operação de atribuição (\leftarrow) permite-nos expressar consultas complexas de uma forma muito conveniente. Escreve-se a consulta como um programa sequencial constituído por uma sequência de atribuições terminada com uma expressão cujo valor é o resultado da consulta.
- A atribuição é sempre efetuada para uma variável de relação temporária.
- Exemplo: escrever $r \div s$ como

$$temp1 \leftarrow \prod_{R-S}(r)$$

$$temp2 \leftarrow \prod_{R-S}((temp1 \times s) - r)$$

$$resultado \leftarrow temp1 - temp2$$

- O resultado à direita de \leftarrow é atribuído à variável que se encontra à esquerda de \leftarrow .
- Pode-se utilizar a variável nas expressões seguintes.

Exemplo Clínica

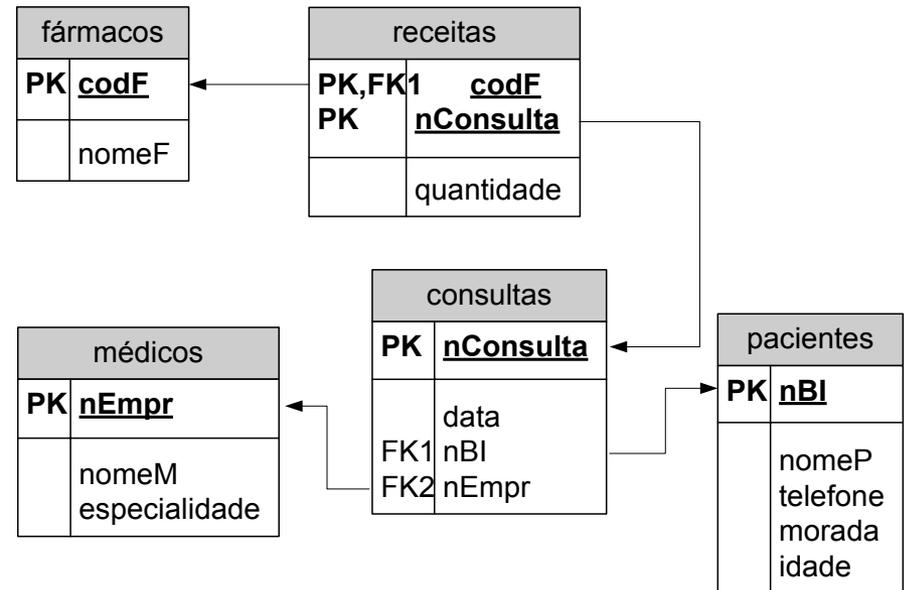
$médicos(nEmpr, nomeM, especialidade)$

$fármacos(codF, nomeF)$

$pacientes(nBl, nomeP, telefone, morada, idade)$

$consultas(nConsulta, data, nBl, nEmpr)$

$receitas(codF, nConsulta, quantidade)$



- Quais os (nomes dos) fármacos que já foram receitados por todos os médicos da clínica?

$$r \leftarrow \prod_{codF, nEmpr} (consultas \times receitas) \div \prod_{nEmpr} (médicos)$$

$$\prod_{nomeF} (fármacos \times r)$$

Algumas propriedades

- Se $R \cap S = \{ \}$ então $r(R) \bowtie s(S) = r(R) \times s(S)$
- $s \bowtie (\sigma_{a=v}(r)) = \sigma_{a=v}(s \bowtie r)$
- $(r \cup r') \bowtie s = (r \bowtie s) \cup (r' \bowtie s)$
- $\sigma_{a=v}(r \text{ op } s) = \sigma_{a=v}(r) \text{ op } \sigma_{a=v}(s)$ para $\text{op} \in \{ \cup, \cap, - \}$

Operações Estendidas da Álgebra Relacional

- Aumentam a expressividade da Álgebra Relacional:
 - ✦ Projeção Generalizada
 - ✦ Funções de Agregação
 - ✦ Junção Externa

Projeção Generalizada

- Permite a utilização de funções aritméticas na lista de projeção.

$$\Pi_{F_1, F_2, \dots, F_n}(E)$$

- E é uma expressão arbitrária de álgebra relacional.
- Cada uma das expressões F_1, F_2, \dots, F_n é uma expressão aritmética envolvendo constantes e atributos no esquema de E .
- Dada a relação *credit-info(customer-name, limit, credit-balance)*, encontrar quanto cada cliente pode ainda gastar:

$$\Pi_{customer-name, limit - credit-balance}(credit-info)$$

- Há quantos dias foi cada uma das consultas

$$\Pi_{nConsulta, today - data}(consultas)$$

Funções de Agregação e Operações

- **Funções de Agregação** aplicam-se a uma coleção de valores e devolvem um único valor como resultado

avg: média dos valores

min: mínimo dos valores

max: máximo dos valores

sum: soma dos valores

count: número dos valores distintos

- **Operação de Agregação** na álgebra relacional

$$G_1, G_2, \dots, G_n \quad \mathcal{G} \quad F_1(A_1), F_2(A_2), \dots, F_n(A_n) \quad (E)$$

✦ E é uma expressão de álgebra relacional

✦ G_1, G_2, \dots, G_n é uma lista de atributos de agrupamento (pode ser vazia)

✦ Cada F_i é uma função de agregação

✦ Cada A_i é um nome de um atributo

- Intuitivamente:

✦ Agrupa a relação resultado de E em conjuntos que tenham valores iguais nos atributos G_1, \dots, G_n (se $n = 0$, faz um grupo com todo o E)

✦ Para cada grupo, devolve um tuplo

✦ Esse tuplo tem os valores dos atributos G_1, \dots, G_n do grupo, mais o resultado de aplicar as várias $F_i(A_i)$ ao conjunto de valores do grupo

Operação de Agregação - Exemplo

r		
A	B	C
α	α	7
α	β	7
β	β	3
β	β	10

$G_{\text{sum}(C)}(r)$
sum(C)
27

Operação de Agregação - Exemplo

- Relação *account* agrupada por *branch-name*:

contas

balcão	número-conta	saldo
Sete Rios	A-102	400
Sete Rios	A-202	900
Benfica	A-217	750
Benfica	A-216	750
Almada	A-222	700

balcão $G_{\text{sum(saldo)}}$ (contas)

balcão	sum(saldo)
Sete Rios	1300
Benfica	1500
Almada	700

Funções de Agregação (Cont.)

- O resultado da agregação não tem um nome
 - ✦ Pode-se recorrer à operação de renomeação para lhe dar um nome
 - ✦ Por conveniência, permite-se a renomeação de atributos na operação de agregação

branch-name **G** *sum(balance)* **as** *sum-balance* (*account*)

Exemplo Clínica

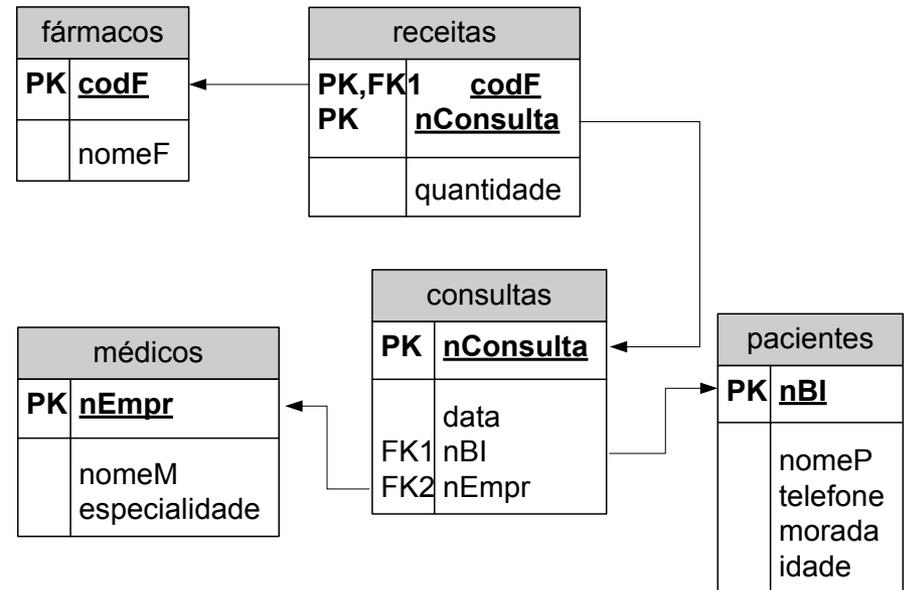
médicos(*nEmpr*, *nomeM*, *especialidade*)

fármacos(*codF*, *nomeF*)

pacientes(*nBl*, *nomeP*, *telefone*, *morada*, *idade*)

consultas(*nConsulta*, *data*, *nBl*, *nEmpr*)

receitas(*codF*, *nConsulta*, *quantidade*)



- Qual a média de idades dos pacientes de cada um dos médicos?

$nEmpr \mathcal{G} avg(idade) \text{ as } média (\Pi_{nBl, nEmpr, idade} (consultas \bowtie pacientes))$

Exemplo Clínica

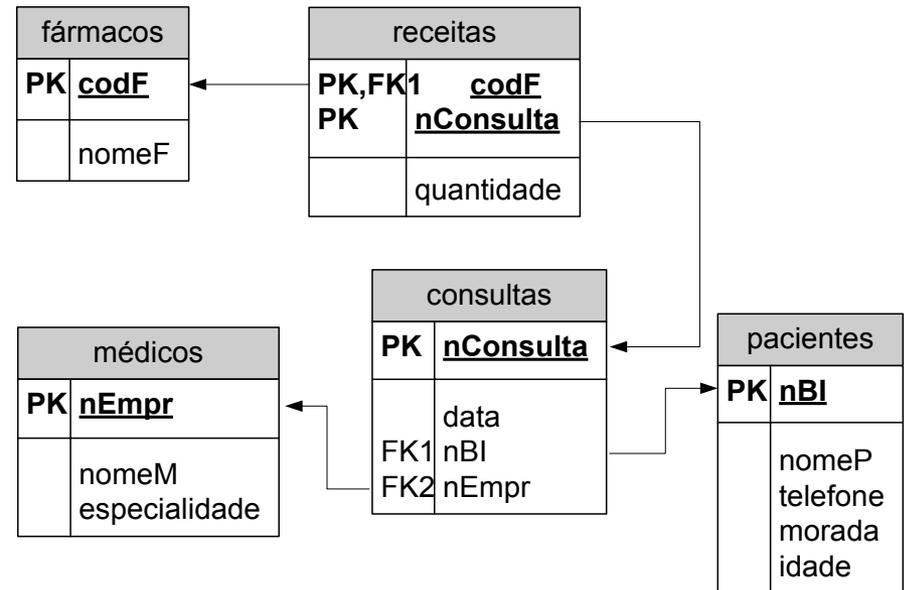
médicos(nEmpr, nomeM, especialidade)

fármacos(codF, nomeF)

pacientes(nBl, nomeP, telefone, morada, idade)

consultas(nConsulta, data, nBl, nEmpr)

receitas(codF, nConsulta, quantidade)



- Quantos fármacos diferentes foram receitados em cada uma das consultas?

nconsulta \mathcal{G} *count*(codF) *as* *quantos* (*receitas*)

Exemplo Clínica

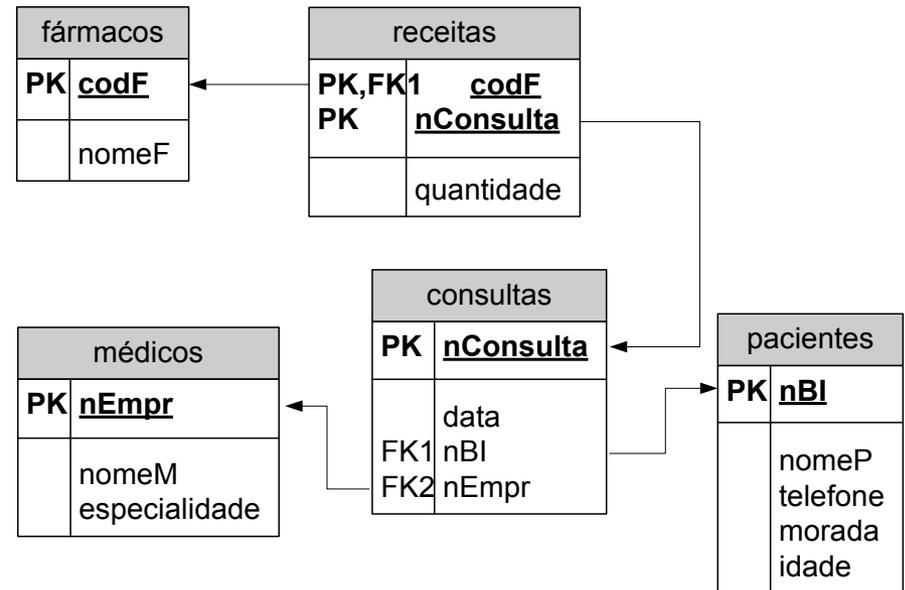
médicos(*nEmpr*, *nomeM*, *especialidade*)

fármacos(*codF*, *nomeF*)

pacientes(*nBl*, *nomeP*, *telefone*, *morada*, *idade*)

consultas(*nConsulta*, *data*, *nBl*, *nEmpr*)

receitas(*codF*, *nConsulta*, *quantidade*)



- Para cada médico, qual a quantidade média de fármacos receitados por consulta?

$quantCons \leftarrow_{nconsulta} \mathcal{G} \text{sum}(quant) \text{ as } soma (receitas)$

$nEmpr \mathcal{G} \text{avg}(soma) (quantCons \bowtie consultas)$

Exemplo Clínica

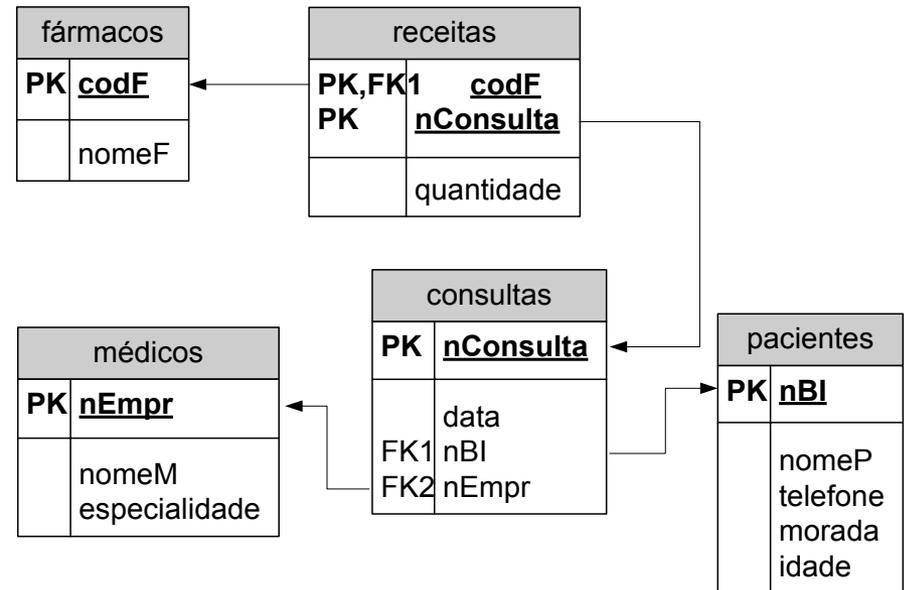
médicos(nEmpr, nomeM, especialidade)

fármacos(codF, nomeF)

pacientes(nBI, nomeP, telefone, morada, idade)

consultas(nConsulta, data, nBI, nEmpr)

receitas(codF, nConsulta, quantidade)



- Qual a idade do paciente mais velho?

$\mathcal{G} \max(idade) \text{ as } idade \text{ (pacientes)}$

Junção Externa (ou exterior)

- Uma extensão da operação de junção que evita a perda de informação.
- Calcula a junção e depois adiciona ao resultado os tuplos de uma relação que não estão relacionados com a outra relação na junção.
- Utiliza valores *nulos* :
 - ✦ *null* significa que o valor é desconhecido ou que não existe
 - ✦ Simplificadamente, todas as comparações com *null* são **falsas** por definição.
 - ❖ Estudaremos já de seguida o significado preciso das comparações com nulos,

Junções Externas - Exemplos

contas

número-conta	balcão	saldo
A-102	Sete Rios	400
A-202	Benfica	900
A-216	Almada	750

titulares

nome-cliente	número-conta
Joana Sobral	A-102
Pedro Silva	A-202
Ana Dias	A-310

■ Junção interna (natural)

contas × titulares

número-conta	balcão	saldo	nome-cliente
A-102	Sete Rios	400	Joana Sobral
A-202	Benfica	900	Pedro Silva

Junções Externas - Exemplos

contas

número-conta	balcão	saldo
A-102	Sete Rios	400
A-202	Benfica	900
A-216	Almada	750

titulares

nome-cliente	número-conta
Joana Sobral	A-102
Pedro Silva	A-202
Ana Dias	A-310

■ Junção externa esquerda

contas ⋈ titulares

número-conta	balcão	saldo	nome-cliente
A-102	Sete Rios	400	Joana Sobral
A-202	Benfica	900	Pedro Silva
A-216	Almada	750	NULL

Junções Externas - Exemplos

contas

número-conta	balcão	saldo
A-102	Sete Rios	400
A-202	Benfica	900
A-216	Almada	750

titulares

nome-cliente	número-conta
Joana Sobral	A-102
Pedro Silva	A-202
Ana Dias	A-310

■ Junção externa direita

Contas ⋈ titulares

número-conta	balcão	saldo	nome-cliente
A-102	Sete Rios	400	Joana Sobral
A-202	Benfica	900	Pedro Silva
A-310	NULL	NULL	Ana Dias

Junções Externas - Exemplos

contas

número-conta	balcão	saldo
A-102	Sete Rios	400
A-202	Benfica	900
A-216	Almada	750

titulares

nome-cliente	número-conta
Joana Sobral	A-102
Pedro Silva	A-202
Ana Dias	A-310

■ Junção externa total

contas ⋈ titulares

número-conta	balcão	saldo	nome-cliente
A-102	Sete Rios	400	Joana Sobral
A-202	Benfica	900	Pedro Silva
A-216	Almada	750	NULL
A-310	NULL	NULL	Ana Dias

Exemplo Clínica

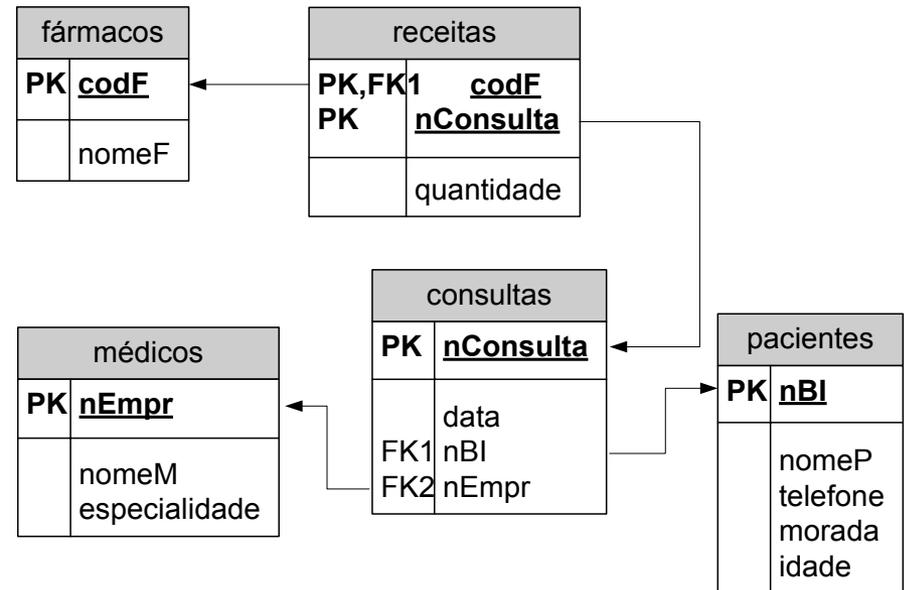
médicos(nEmpr, nomeM, especialidade)

fármacos(codF, nomeF)

pacientes(nBl, nomeP, telefone, morada, idade)

consultas(nConsulta, data, nBl, nEmpr)

receitas(codF, nConsulta, quantidade)



- Quantos fármacos diferentes foram receitados em cada uma das consultas?

nconsulta \mathcal{G} **count**(codF) **as** *quantos* (*consultas* \bowtie *receitas*)

Valores Nulos

- É possível que um tuplo tenha um valor nulo, denotado por *null*, para algum dos seus atributos
- *null* significa um valor desconhecido ou que não existe.
- O resultado de qualquer expressão aritmética envolvendo um *null* é *null*.
- As funções de agregação ignoram os valores nulos
 - ✦ Decisão arbitrária. Alternativamente, poder-se-ia retornar *null*.
 - ✦ Segue a semântica do standard SQL no tratamento de valores nulos.
- Na eliminação de duplicados e agrupamento, um *null* é tratado como um outro valor qualquer, assumindo-se que dois *nulls* são o mesmo
 - ✦ Alternativa: assumir que cada *null* é diferente de todos os outros
 - ✦ Ambas são decisões arbitrárias, portanto seguem a do SQL

Valores Nulos

- **Comparações com valores nulos devolvem o valor de verdade *unknown***
 - * Se *false* fosse usado em vez *unknown*, então $\text{not } (A < 5)$
não seria equivalente a $A \geq 5$
- **Lógica a três valores com o valor de verdade *unknown*:**
 - * OR: $(\text{unknown or true}) = \text{true}$,
 $(\text{unknown or false}) = \text{unknown}$
 $(\text{unknown or unknown}) = \text{unknown}$
 - * AND: $(\text{true and unknown}) = \text{unknown}$,
 $(\text{false and unknown}) = \text{false}$,
 $(\text{unknown and unknown}) = \text{unknown}$
 - * NOT: $(\text{not unknown}) = \text{unknown}$
- **Como só são devolvidos tuplos para os quais a condição é verdadeira, o valor de verdade *unknown* é tratado para este efeitos como *false*.**

Modificação da Base de Dados

- O conteúdo da base de dados pode ser modificado através das seguintes operações:
 - ✦ Remoção
 - ✦ Inserção
 - ✦ Atualização
- Todas estas operação são expressas por intermédio do operador de atribuição.

Remoção

- Uma operação de remoção é expressa de uma maneira semelhante a uma consulta, exceto que os tuplos selecionados são removidos da base de dados.
- Só se podem remover tuplos integralmente; não se podem apagar valores de determinados atributos (isso é uma operação de atualização, que veremos à frente)
- Uma remoção é expressa em álgebra relacional por:

$$r \leftarrow r - E$$

em que r é uma relação e E é uma operação de álgebra relacional.

Exemplos de Remoção

- Apagar todas as contas na agência de Perryridge.

$$account \leftarrow account - \sigma_{branch-name = 'Perryridge'}(account)$$

- Apagar todos os registos de empréstimos de montante entre 0 e 50

$$loan \leftarrow loan - \sigma_{amount \geq 0 \text{ and } amount \leq 50}(loan)$$

- Apagar todas as contas de balcões localizados em Needham.

$$r_1 \leftarrow \sigma_{branch-city = 'Needham'}(account \bowtie branch)$$
$$r_2 \leftarrow \Pi_{branch-name, account-number, balance}(r_1)$$
$$account \leftarrow account - r_2$$
$$r_3 \leftarrow \Pi_{customer-name, account-number}(r_2 \bowtie depositor)$$
$$depositor \leftarrow depositor - r_3$$

Exemplos de Remoção (cont)

- Apagar toda a informação relativa a consultas anteriores a 2010:

$$r_1 \leftarrow \sigma_{data < 01-01-2010} (consultas)$$

$$r_2 \leftarrow \Pi_{codF, nConsulta, quant} (receitas \bowtie r_1)$$

$$consultas \leftarrow consultas - r_1$$

$$receitas \leftarrow receitas - r_2$$

Inserção

- Para inserir informação numa relação podemos:
 - ✦ especificar um tuplo a ser inserido
 - ✦ escrever uma consulta cujo resultado é um conjunto de tuplos a inserir

- Na álgebra relacional, uma inserção é expressa por:

$$r \leftarrow r \cup E$$

em que r é uma relação e E é uma expressão de álgebra relacional.

- A inserção de um único tuplo é efetuada quando a expressão E é a relação constante contendo esse tuplo.

Exemplos de Inserção

- Inserir informação na base de dados especificando que o cliente Smith tem €1200 na conta A-973 na agência de Perryridge.

$$account \leftarrow account \cup \{('Perryridge', A-973, 1200)\}$$
$$depositor \leftarrow depositor \cup \{('Smith', A-973)\}$$

- Dar um bônus a todos os mutuários na agência de Perryridge: uma conta de poupança de €200. O número do empréstimo é utilizado para número da conta de poupança.

$$r_1 \leftarrow (\sigma_{branch-name = "Perryridge"}(borrower \bowtie loan))$$
$$account \leftarrow account \cup \Pi_{loan-number, branch-name, 200}(r_1)$$
$$depositor \leftarrow depositor \cup \Pi_{customer-name, loan-number}(r_1)$$

Exemplos de Inserção

- Inserir informação na base de dados especificando que um novo paciente, com BI nº 10000000 e nome Paulo, teve uma consulta (nº1000) no dia 30-09-2015 com o médico João (assumindo que só há um médico com esse nome)

$pacientes \leftarrow pacientes \cup \{(10000000, 'Paulo', null, null, null)\}$

$consultas \leftarrow consultas \cup$

$\cup \Pi_{1000,30-09-2015,1000,nEmpr} (\sigma_{nomeM = "João"} (médicos))$

Atualização

- Um mecanismo para alterar um valor de um tuplo sem alterar *todos* os valores do tuplo.
- Recorre-se ao operador de projeção generalizada para efetuar este tipo de tarefa

$$r \leftarrow \Pi_{F_1, F_2, \dots, F_i} (r)$$

- Cada F_i ou é o i -ésimo atributo de r , se o i -ésimo atributo não for alterado, ou
- uma expressão F_i , envolvendo apenas constantes e atributos de r , que permite calcular o novo valor do atributo.

Exemplos de Atualizações

- Pague juros de 5% em todas as contas

$account \leftarrow \Pi_{AN, BN, BAL} * 1.05 (account)$

em que AN , BN e BAL são abreviaturas para *account-number*, *branch-name* e *balance*, respetivamente.

- Pague 6% de juros em todas as contas com saldo superior a €10,000 e 5% às restantes contas.

$account \leftarrow \Pi_{AN, BN, BAL} * 1.06 (\sigma_{BAL > 10000} (account))$
 $\cup \Pi_{AN, BN, BAL} * 1.05 (\sigma_{BAL \leq 10000} (account))$

Exemplos Adicionais

Exemplo Bancário

branch (branch-name, branch-city, assets)

customer (customer-name, customer-street, customer-only)

account (account-number, branch-name, balance)

loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)

borrower (customer-name, loan-number)

Exemplo Bancário

branch (branch-name, branch-city, assets)

customer (customer-name, customer-street, customer-only)

account (account-number, branch-name, balance)

loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)

borrower (customer-name, loan-number)

- Determinar todos os empréstimos superiores a €1200

$$\sigma_{amount > 1200} (loan)$$

- Encontrar os números dos empréstimos de montante superior a €1200

$$\Pi_{loan-number} (\sigma_{amount > 1200} (loan))$$

Exemplo Bancário

branch (branch-name, branch-city, assets)

customer (customer-name, customer-street, customer-only)

account (account-number, branch-name, balance)

loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)

borrower (customer-name, loan-number)

- Listar os nomes de todos os clientes que têm um empréstimo, uma conta, ou ambas as coisas

$\Pi_{customer-name} (borrower) \cup \Pi_{customer-name} (depositor)$

- Encontrar os clientes que têm um empréstimo e uma conta no banco.

$\Pi_{customer-name} (borrower) \cap \Pi_{customer-name} (depositor)$

Exemplo Bancário

branch (branch-name, branch-city, assets)

customer (customer-name, customer-street, customer-only)

account (account-number, branch-name, balance)

loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)

borrower (customer-name, loan-number)

- Determinar todos os clientes que têm um empréstimo na agência de Perryridge.

$\Pi_{customer-name} (\sigma_{branch-name='Perryridge'} (borrower \bowtie loan))$

Exemplo Bancário

branch (branch-name, branch-city, assets)

customer (customer-name, customer-street, customer-only)

account (account-number, branch-name, balance)

loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)

borrower (customer-name, loan-number)

- Listar os nomes dos clientes que possuem um empréstimo na agência de Perryridge mas que não têm nenhuma conta no banco.

$$\Pi_{customer-name} (\sigma_{branch-name='Perryridge'} (borrower \bowtie loan)) \\ - \Pi_{customer-name} (depositor)$$

Exemplo Bancário

branch (branch-name, branch-city, assets)

customer (customer-name, customer-street, customer-only)

account (account-number, branch-name, balance)

loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)

borrower (customer-name, loan-number)

- Quais os clientes que têm um empréstimo na agência de Perryridge.

$\Pi_{\text{customer-name}}(\sigma_{\text{branch-name} = \text{'Perryridge'}}(\text{borrower} \bowtie \text{loan}))$

ou

$\Pi_{\text{customer-name}}(\sigma_{\text{branch-name} = \text{'Perryridge'}}(\text{loan}) \bowtie \text{borrower})$

Exemplo Bancário

branch (branch-name, branch-city, assets)

customer (customer-name, customer-street, customer-only)

account (account-number, branch-name, balance)

loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)

borrower (customer-name, loan-number)

- Determinar o saldo mais elevado entre todas as contas

- * Renomear a relação *account* como *d*

$\Pi_{balance}(account) - \Pi_{account.balance}$

$(\sigma_{account.balance < d.balance} (account \times \rho_d (account)))$

ou, simplesmente

$\mathcal{G} \max(balance) (account)$

Exemplo Bancário

branch (branch-name, branch-city, assets)

customer (customer-name, customer-street, customer-only)

account (account-number, branch-name, balance)

loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)

borrower (customer-name, loan-number)

- Encontrar os clientes que têm uma conta pelo menos nas agências de “Downtown” e “Uptown”. (onde CN= *customer-name* e BN = *branch-name*)

$$\Pi_{CN}(\sigma_{BN='Downtown'}(depositor \bowtie account)) \cap \\ \cap \Pi_{CN}(\sigma_{BN='Uptown'}(depositor \bowtie account))$$

ou

$$\Pi_{CN,BN}(depositor \bowtie account) \div \rho_{t(BN)}(\{('Downtown'),('Uptown')\})$$

Exemplo Bancário

branch (branch-name, branch-city, assets)

customer (customer-name, customer-street, customer-only)

account (account-number, branch-name, balance)

loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)

borrower (customer-name, loan-number)

- Listar todos os clientes que têm uma conta em todas as agências localizadas na cidade de Brooklyn.

$$\begin{aligned} & \Pi_{customer-name, branch-name} (depositor \bowtie account) \div \\ & \div \Pi_{branch-name} (\sigma_{branch-city = 'Brooklyn'} (branch)) \end{aligned}$$