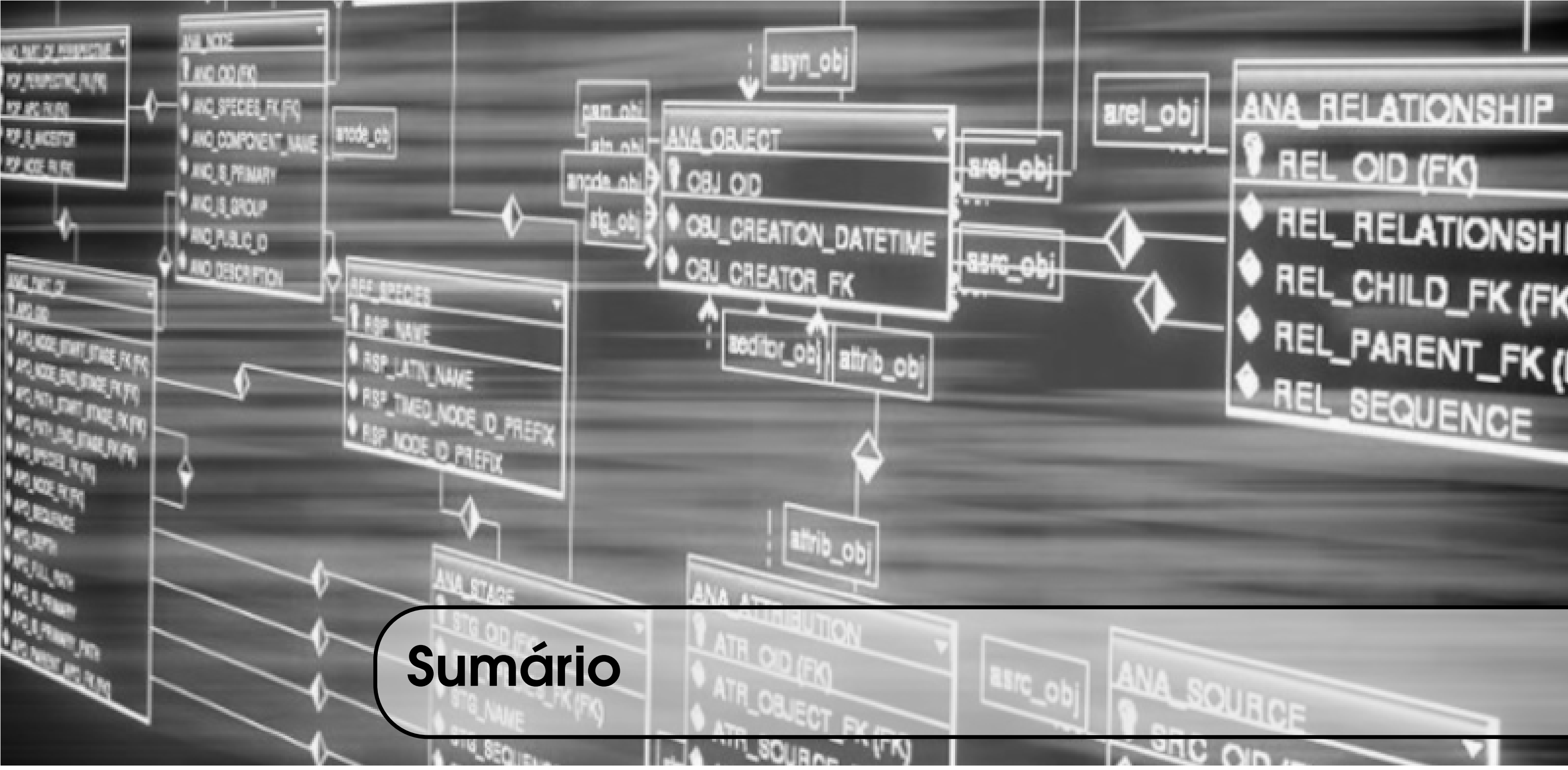
Operações da Álgebra Relacional

Newton Carlos Will


asyn_obj
arel_obj
ANA_RELATIONSHIP
REL_OID (FK)
ANA_OBJECT
OBJ_OID
OBJ_CREATION_DATETIME
OBJ_CREATOR_FK
srel_obj
asrc_obj
attrib_obj
REL_SEQUENCE
ANA_SOURCE

Sumário

1. Introdução

A álgebra relacional é muito importante por diversos motivos. Primeiro, ela oferece um alicerce formal para as operações do modelo relacional. Segundo, e talvez mais importante, ela é usada como base para a implementação e otimização de consultas nos módulos de otimização e processamento de consultas, que são partes integrais dos sistemas de gerenciamento de banco de dados relacional (SGBDRs). Terceiro, alguns de seus conceitos são incorporados na linguagem de consulta padrão SQL para SGDBRs.

Embora a maioria dos SGBDRs comerciais em uso não ofereça interfaces de usuário para consultas da álgebra relacional, as operações e funções essenciais nos módulos internos da maioria dos sistemas relacionais são baseadas nas operações da álgebra relacional.

A álgebra relacional normalmente é considerada uma parte integral do modelo de dados relacional. Suas operações podem ser dividas em dois grupos. Um grupo inclui conjunto de operações da teoria de conjunto da matemática; estas são aplicáveis porque cada relação é definida como um conjunto de tuplas no modelo relacional formal. As operações de conjunto incluem UNIÃO, INTERSECÇÃO, DIFERENÇA DE CONJUNTO e PRODUTO CARTESIANO (também conhecida como PRODUTO CRUZADO). O outro grupo consiste em operações desenvolvidas especificamente para bancos de dados relacionais - entre elas estão SELEÇÃO, PROJEÇÃO e JUNÇÃO, entre outras.

Algumas solicitações de banco de dados comuns não podem ser realizadas com as operações originais da álgebra relacional, de modo que operações adicionais foram criadas para expressá-las. Estas incluem funções de agregação, que são operações que podem resumir dados das tabelas, bem como tipos adicionais de operações JUNÇÃO e UNIÃO, conhecidas como JUNÇÃO EXTERNA e UNIÃO EXTERNA.

1.1 Modelo do Banco de Dados Utilizado nos Exemplos

Para os exemplos que se seguirão nos próximos capítulos, será utilizado, basicamente, o modelo descrito pelo Diagrama Entidade-Relacionamento mostrado na Figura 1.1, que representa o modelo conceitual do banco de dados. Os registros (ou tuplas) contidos em cada relação são apresentados nas Tabelas 1.1-1.6.

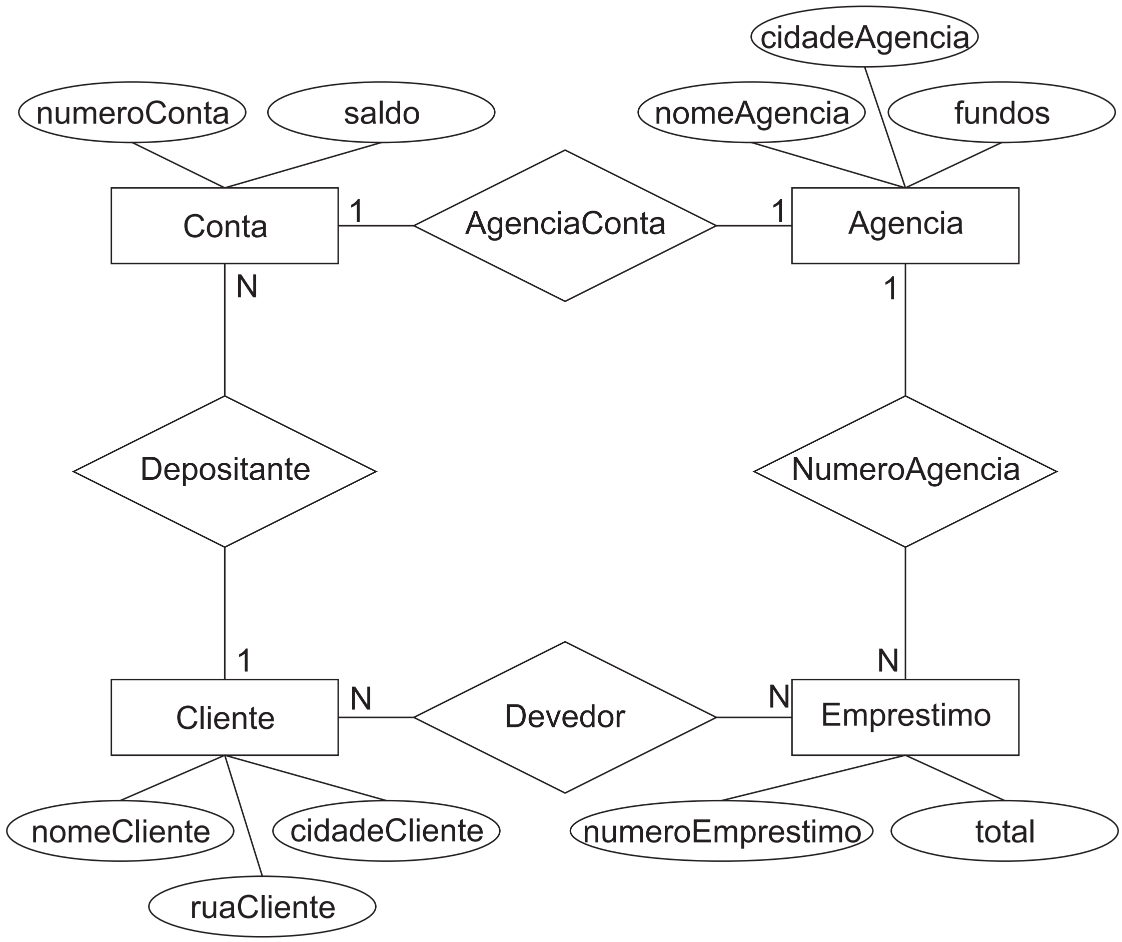


Figura 1.1: Diagrama E-R para empresa da área bancária, utilizado nos exemplos descritos nos próximos capítulos

Tabela 1.1: Registros da relação Agencia

nomeAgencia	**cidadeAgencia**	**fundos**
Downtown	Brooklyn	9000000
Redwood	Palo Alto	2100000
Perryridge	Horseneck	1700000
Mianus	Horseneck	400000
Round Hill	Horseneck	8000000
Pownal	Bennington	300000
North Town	Rye	3700000
Brighton	Brooklyn	7100000

Tabela 1.2: Registros da relação Cliente

nomeCliente	ruaCliente	cidadeCliente
Jones	Main	Harrison
Smith	North	Rye
Hayes	Main	Harrison
Curry	North	Rye
Lindsay	Park	Pittsfield
Turner	Putnam	Stamford
Williams	Nassau	Princeton
Adams	Spring	Pittsfield
Johnson	Alma	Palo Alto
Glenn	Sand Hill	Woodside
Brooks	Senator	Brooklyn
Green	Walnut	Stamford

Tabela 1.3: Registros da relação Conta

nomeAgencia	numeroConta	saldo
Downtown	A-101	500
Mianus	A-215	700
Perryridge	A-102	400
Round Hill	A-305	350
Brighton	A-201	900
Redwood	A-222	700
Brighton	A-217	750

Tabela 1.4: Registros da relação Depositante

nomeCliente	numeroConta
Johnson	A-101
Smith	A-215
Hayes	A-102
Turner	A-305
Johnson	A-201
Jones	A-217
Lindsay	A-222

Tabela 1.5: Registros da relação Emprestimo

nomeAgencia	numeroEmprestimo	total
Downtown	L-17	1000
Redwood	L-23	2000
Perryridge	L-15	1500
Downtown	L-14	1500
Mianus	L-93	500
Round Hill	L-11	900
Perryridge	L-16	1300
Mianus	L-20	500

Tabela 1.6: Registros da relação Devedor

nomeCliente	**numeroEmprestimo**
Jones	L-17
Smith	L-23
Hayes	L-15
Jackson	L-14
Curry	L-93
Smith	L-11
Williams	L-17
Adams	L-16
Turner	L-21

2. Operações Relacionais Unárias: Seleção

2.1 A Operação Seleção

A operação SELEÇÃO é usada para escolher um subconjunto de tuplas de uma relação que satisfaça uma condição de seleção. Pode-se considerar que a operação SELEÇÃO seja um filtro que mantém apenas as tuplas que satisfaçam uma condição qualificadora. Como alternativa, podemos considerar que essa operação restringe as tuplas em uma relação para apenas aquelas que satisfaçam a condição. A operação SELEÇÃO também pode ser visualizada como uma partição horizontal da relação em dois conjuntos de tuplas - aquelas que satisfazem a condição e são selecionadas, e aquelas que não satisfazem a condição e são descartadas.

Em geral, a operação SELEÇÃO é indicada por

$$\sigma_{<condicao\_selecao>}(R) \tag{2.1}$$

onde o símbolo σ (sigma) é usado para indicar o operador SELEÇÃO e a condição de seleção é uma expressão booleana (condição) especificadas nos atributos da relação R. Observe que R costuma ser uma expressão da álgebra relacional cujo resultado é uma relação - a mais simples expressão desse tipo é apenas o nome de uma relação no banco de dados. A relação resultante da operação SELEÇÃO tem os mesmos atributos de R.

A expressão booleana especificada em $< condicao\_selecao >$ é composta de uma série de cláusulas da forma

$$< nome\_atributo >< operador\_comparacao >< valor\_constante > \tag{2.2}$$

ou

$$< nome\_atributo >< operador\_comparacao >< nome\_atributo > \tag{2.3}$$

onde $< nome\_atributo >$ é o nome de um atributo de R, $< operador\_comparacao >$ em geral é um dos operadores $\{=, <, \leq, >, \geq, \neq\}$ e $< valor\_constante >$ é um valor constante do domínio do atributo. As cláusulas podem ser conectadas pelos operadores booleanos padrão **AND** ($\wedge$), **OR** ($\vee$) e **NOT** ($\neg$) para formar uma condição de seleção geral. Por exemplo, para encontrar as tuplas que contenham empréstimos acima de 1200 dólares, feitos na agência Perryridge, escrevemos:

$$\sigma_{nomeAgencia=\text{“}Perryridge''\wedge total>1200}(Emprestimo) \tag{2.4}$$

o que nos dá como resultado a relação apresentada na Tabela 2.1. Observe que os atributos que constam na relação de resultado são exatamente os mesmos que constam na relação Emprestimo (Tabela 1.5).

Tabela 2.1: Resultado de $\sigma_{nomeAgencia=\text{"}Perryridge'' \wedge total>1200}(Emprestimo)$

nomeAgencia	**numeroEmprestimo**	**total**
Perryridge	L-15	1500
Perryridge	L-16	1300

A fim de facilitar a leitura, a Expressão 2.5 pode ser reescrita da seguinte forma:

$$\sigma_{nomeAgencia=\text{"}Perryridge''\,\mathbf{AND}\,total>1200}(Emprestimo) \tag{2.5}$$

Observe que todos os operadores de comparação no conjunto $\{=, <, \leq, >, \geq, \neq\}$ podem ser aplicados aos atributos cujos domínios são valores ordenados, como domínios numéricos ou datas. Os domínios de cadeias de caracteres também são considerados ordenados com base na ordem alfabética dos caracteres. Se o domínio de um atributo for um conjunto de valores desordenados, então somente os operadores de comparação no conjunto $\{=, \neq\}$ podem ser usados. Um exemplo de domínio desordenado é Cor = { “Vermelho”, “Azul”, “Verde”, “Branco”, “Amarelo”, ... }, onde nenhuma ordem é especificada entre as diversas cores.

O operador SELEÇÃO é unário; ou seja, ele é aplicado a uma única relação. Além do mais, a operação de seleção é aplicada a cada tupla individualmente; logo, as condições de seleção não podem envolver mais de uma tupla. O grau da relação resultante de uma operação SELEÇÃO - seu número de atributos - é o mesmo que o grau de R. O número de tuplas na relação resultante é sempre menor ou igual ao número de tuplas em R.

Observe que a operação SELEÇÃO é cumulativa, ou seja,

$$\sigma_{<cond1>}(\sigma_{<cond2>}(R)) = \sigma_{<cond2>}(\sigma_{<cond1>}(R)) \tag{2.6}$$

Portanto, uma sequência de SELEÇÃO pode ser aplicada em qualquer ordem. Além disso, sempre podemos combinar uma cascata (ou sequência) de operações SELEÇÃO a uma única operação SELEÇÃO com uma condição conjuntiva AND; ou seja,

$$\sigma_{<cond1>}(\sigma_{<cond2>}(...(\sigma_{<condn>}(R))...)) = \sigma_{<cond1>\wedge<cond2>\wedge...\wedge<condn>}(R)) \tag{2.7}$$

2.2 A Operação Projeção

Se pensarmos em uma relação como uma tabela, a operação SELEÇÃO escolhe algumas linhas da tabela enquanto descarta outras linhas. A operação PROJEÇÃO, por sua vez, seleciona certas colunas da tabela enquanto descarta outras. Se estivermos interessados apenas em certos atributos de uma relação, usamos a operação PROJEÇÃO para projetar a relação apenas por esses atributos. Portanto, o resultado da operação PROJEÇÃO pode ser visualizado como uma partição vertical da relação em duas relações: uma tem as colunas (atributos) necessários e contém o resultado da operação, e a outra contém as colunas descartadas.

A forma geral da operação PROJEÇÃO é

$$\pi_{<lista\_atributos>}(R) \tag{2.8}$$

onde π (pi) é o símbolo usado para representar a operação PROJEÇÃO, e $< lista\_atributos >$ é a sublista desejada de atributos da relação R. Mais uma vez, observe que R, em geral, é uma expressão da álgebra relacional cujo resultado é uma relação, que no caso mais simples é apenas o nome de uma relação do banco de dados. O resultado da operação PROJEÇÃO tem apenas os atributos especificados em $< lista\_atributos >$ na mesma ordem em que eles aparecem na lista. Logo, seu grau é igual ao número de atributos em $< lista\_atributos >$.

Como exemplo, a consulta para relacionar todos os nomes de agências e totais dos empréstimos, com base nos registros contidos na relação Emprestimo, pode ser escrita da seguinte forma:

$$\pi_{nomeAgencia,total}(Emprestimo) \tag{2.9}$$

sendo o resultado dessa operação apresentado na Tabela 2.2.

Tabela 2.2: Resultado de $\pi_{nomeAgencia,total}(Emprestimo)$

nomeAgencia	**total**
Downtown	1000
Redwood	2000
Perryridge	1500
Downtown	1500
Mianus	500
Round Hill	900
Perryridge	1300

Observe que a tupla <*"Mianus", 500*> só aparece uma vez na Tabela 2.2, embora essa combinação de valores apareça duas vezes na relação Emprestimo, apresentada na Tabela 1.5. Isso ocorre porque a operação PROJEÇÃO remove quaisquer tuplas duplicadas, de modo que o resultado dessa operação é um conjunto de tuplas distintas, e, portanto, uma relação válida. Isso é conhecido como eliminação de duplicatas.

A eliminação de duplicatas envolve a classificação ou alguma outra técnica para detectar duplicatas e, portanto, aumenta o processamento. Se as duplicatas não fossem eliminadas, o resultado seria um multiconjunto ou bag de tupla, em vez de um conjunto. Isso não era permitido no modelo relacional formal.

2.3 A Operação Relacional de Comparação

O fato de o resultado de uma operação relacional ser uma relação é importante. Consideremos uma consulta mais complexa como “encontre o nome dos clientes que moram em ’Harrison”’. Escrevemos

$$\pi_{nomeCliente}(\sigma_{cidadeCliente=\text{``}Harrison''}(Cliente)) \tag{2.10}$$

Note que, em vez de dar o nome da relação como argumento na operação de projeção, foi criada uma expressão que tem como resultado uma relação.

Em geral, desde que o resultado de uma operação em álgebra relacional seja do mesmo tipo que sua entrada (relação), as operações em álgebra relacional podem ser compostas juntas em uma expressão em álgebra relacional. A composição de operações em álgebra relacional em expressões é similar à composição de operações aritméticas (como $+$, $-$, $*$ e $\div$) em expressões aritméticas.

3. Operações de Álgebra Relacional com B

O próximo grupo de operações da álgebra relacional são as operações matemáticas padrão sobre conjuntos.

Várias operações de teoria de conjuntos são usadas para mesclar os elementos de dois conjuntos de diversas maneiras, incluindo UNIÃO, INTERSECÇÃO e DIFERENÇA DE CONJUNTO. Estas são operações binárias, ou seja, cada uma é aplicada a dois conjuntos (de tuplas). Quando essas operações são adaptadas aos bancos de dados relacionais, as duas relações sobre as quais qualquer uma dessas três operações são aplicadas precisam ter o mesmo tipo de tuplas, essa condição é chamada de *compatibilidade de união* ou *compatibilidade de tipo*. Duas relações $R(A_1, A_2, ..., A_n)$ e $S(B_1, B_2, ..., B_n)$ são consideradas compatíveis na união (ou compatíveis no tipo) se tiverem o mesmo grau n e se $dominio(A_i) = dominio(B_i)$, para $1 \leq i \leq n$. Isso significa que as duas relações têm o mesmo número de atributos e cada par correspondente de atributos tem o mesmo domínio.

3.1 A Operação União

O resultado da operação UNIÃO, indicada por $R \cup S$, é uma relação que inclui todas as tuplas que estão em R ou em S ou tanto em R quanto em S. As tuplas duplicadas são eliminadas.

Considere a consulta para encontrar o nome de todos os clientes do banco que tenham uma conta, um empréstimo ou ambos. Note que a relação Cliente não possui essa informação, desde que um cliente não precise, necessariamente, ter uma conta ou um empréstimo para "existir" no banco.

Para responder a essa consulta, precisamos de informações da relação Depositante (Tabela 1.4) e Devedor (Tabela 1.6). Sabemos como encontrar todos os clientes com um empréstimo no banco:

$$\pi_{nomeCliente}(Devedor) \tag{3.1}$$

Também sabemos como encontrar os nomes de todos os clientes com uma conta no banco:

$$\pi_{nomeCliente}(Depositante) \tag{3.2}$$

Para responder à consulta, precisamos da união desses dois conjuntos, isto é, precisamos de todos os nomes dos clientes que aparecem em uma ou em ambas as relações. Logo, a expressão necessária é:

$$\pi_{nomeCliente}(Devedor) \cup \pi_{nomeCliente}(Depositante) \tag{3.3}$$

A relação resultante dessa consulta aparece na Tabela 3.1. Note que há dez tuplas no resultado, apesar de haver sete devedores e seis depositantes. Essa discrepância ocorre porque Smith, Jones e Hayes são devedores e depositantes. Como as relações são conjuntos, valores duplicados são eliminados.

Tabela 3.1: Nomes de todos os clientes que tenham um empréstimo ou uma conta.

nomeCliente
Johnson
Smith
Hayes
Turner
Jones
Lindsay
Jackson
Curry
Williams
Adams

3.2 A Operação Intersecção

O resultado da operação INTERSECÇÃO, indicada por $R \cap S$, é uma relação que inclui todas as tuplas que estão tanto em R quanto em S.

Suponha que desejamos encontrar todos os clientes que tenham tanto empréstimo quanto conta. Usando a intersecção de conjuntos, podemos escrever:

$$\pi_{nomeCliente}(Devedor) \cap \pi_{nomeCliente}(Depositante) \tag{3.4}$$

O resultado dessa consulta aparece na Tabela 3.2.

Tabela 3.2: Clientes com contas e empréstimo no banco.

nomeCliente
Hayes
Jones
Smith

3.3 A Operação Diferença entre Conjuntos

A operação DIFERENÇA ENTRE CONJUNTOS, denotada por $-$, permite-nos encontrar as tuplas que estão em uma relação, mas não em outra. A expressão $R - S$ resulta na relação que contém as tuplas que estão em R, mas não em S.

Podemos encontrar todos os clientes do banco que possuem contas mas não possuem empréstimos escrevendo:

$$\pi_{nomeCliente}(Depositante) - \pi_{nomeCliente}(Devedor) \tag{3.5}$$

Tabela 3.3: Clientes com contas, mas sem empréstimos.

nomeCliente
Johnson
Turner
Lindsay

O resultado dessa relação aparece na Tabela 3.3.

A operação DIFERENÇA ENTRE CONJUNTOS não é comutativa, ou seja, em geral $(R-S) \neq (S-R)$.

3.4 A Operação Produto Cartesiano

A operação PRODUTO CARTESIANO, representada por $\times$, permite-nos combinar informações de duas relações quaisquer. Representamos o produto cartesiano das relações $R1$ e $R2$ por $R1 \times R2$.

Lembre-se que uma relação é definida por um subconjunto de um produto cartesiano de um conjunto de domínios. A partir dessa definição, é possível, intuitivamente, definir a operação do produto cartesiano. Entretanto, desde que um mesmo nome de atributo pode aparecer tanto em $R1$ quanto em $R2$, precisamos estabelecer um nome de esquema para diferenciar esses dois atributos. Faremos isso anexando um nome de atributo à relação na qual esse atributo se origina. Por exemplo, o esquema de relação para $R = Devedor \times Emprestimo$ é *(Devedor.nomeCliente, Devedor.numeroEmprestimo, Emprestimo.nomeAgencia, Emprestimo.numeroEmprestimo, Emprestimo.total)*.

Nesse esquema podemos diferenciar *Devedor.numeroEmprestimo* de *Emprestimo.numeroEmprestimo*. Para aqueles atributos que aparecem apenas uma vez nos dois esquemas, normalmente omitimos o prefixo do nome da relação. Essa simplificação não leva a qualquer ambiguidade. Podemos então escrever o esquema de relação para R como *(nomeCliente, Devedor.numeroEmprestimo, nomeAgencia, Emprestimo.numeroEmprestimo, total)*.

Agora que sabemos o esquema de relação para $R = Devedor \times Emprestimo$, quais as tuplas que aparecerão em R? Constituímos uma tupla de R por meio de cada par de tuplas possível: um da relação Devedor e outro da relação Emprestimo. Assim, R é uma relação grande, como pode ser visto na Tabela 3.4, onde estão incluídas apenas uma parte das tuplas que compõem R.

Assuma que podemos ter n_1 tuplas em Devedor e n_2 tuplas em Emprestimo. Então, existem $n_1 * n_2$ modos de escolher um par de tuplas - uma tupla de cada relação; assim há $n_1 * n_2$ tuplas em R. Particularmente, note que, para algumas tuplas t em R, podemos ter *t[Devedor.numeroEmprestimo]* $\neq$ *t[Emprestimo.numeroEmprestimo]*.

Suponha que queiramos encontrar os nomes de todos os clientes que tenham um empréstimo na agência Perryridge. Podemos precisar, para isso, de informações das relações Devedor e Emprestimo. Se escrevemos:

$$\sigma_{nomeAgencia=\text{“}Perryridge\text{''}}(Devedor \times Emprestimo) \tag{3.6}$$

então o resultado da relação é mostrado na Tabela 3.5. Teremos uma relação dos devedores ligados apenas à agência Perryridge. Entretanto, a coluna *nomeCliente* pode conter clientes que não tenham um empréstimo na agência de Perryridge. (Se você não consegue ver por que isso é verdade, relembre que o produto cartesiano determina todos os pares possíveis de uma tupla de Devedor com uma tupla de Emprestimo).

Como a operação PRODUTO CARTESIANO associa todas as tuplas de Emprestimo a todas as tuplas de Devedor, sabemos que, se um cliente efetua um empréstimo na agência

Tabela 3.4: Resultado de $Devedor \times Emprestimo$.

nomeCliente	**Devedor. numeroEmprestimo**	**nomeAgencia**	**Emprestimo. numeroEmprestimo**	**total**
Jones	L-17	Downtown	L-17	1000
Jones	L-17	Redwood	L-23	2000
Jones	L-17	Perryridge	L-15	1500
Jones	L-17	Downtown	L-14	1500
Jones	L-17	Mianus	L-93	500
Jones	L-17	Round Hill	L-11	900
Jones	L-17	Perryridge	L-16	1300
Smith	L-23	Downtown	L-17	1000
Smith	L-23	Redwood	L-23	2000
Smith	L-23	Perryridge	L-15	1500
Smith	L-23	Downtown	L-14	1500
Smith	L-23	Mianus	L-93	500
Smith	L-23	Round Hill	L-11	900
Smith	L-23	Perryridge	L-16	1300
Hayes	L-15	Downtown	L-17	1000
Hayes	L-15	Redwood	L-23	2000
Hayes	L-15	Perryridge	L-15	1500
Hayes	L-15	Downtown	L-14	1500
Hayes	L-15	Mianus	L-93	500
Hayes	L-15	Round Hill	L-11	900
Hayes	L-15	Perryridge	L-16	1300
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
Williams	L-17	Downtown	L-17	1000
Williams	L-17	Redwood	L-23	2000
Williams	L-17	Perryridge	L-15	1500
Williams	L-17	Downtown	L-14	1500
Williams	L-17	Mianus	L-93	500
Williams	L-17	Round Hill	L-11	900
Williams	L-17	Perryridge	L-16	1300
Adams	L-16	Downtown	L-17	1000
Adams	L-16	Redwood	L-23	2000
Adams	L-16	Perryridge	L-15	1500
Adams	L-16	Downtown	L-14	1500
Adams	L-16	Mianus	L-93	500
Adams	L-16	Round Hill	L-11	900
Adams	L-16	Perryridge	L-16	1300

Perryridge, então existem algumas tuplas em $Devedor \times Emprestimo$ que contêm seu nome, e *Devedor.numeroEmprestimo = Emprestimo. numeroEmprestimo*. Assim, se escrevemos:

$$\sigma_{Devedor.numeroEmprestimo=Emprestimo.numeroEmprestimo} (\sigma_{nomeAgencia=\text{“}Perryridge''}(Devedor \times Emprestimo)) \quad (3.7)$$

pegamos apenas aquelas tuplas de $Devedor \times Emprestimo$ que pertençam aos que tenham um

Tabela 3.5: Resultado de $\sigma_{nomeAgencia=\text{``}Perryridge''}(Devedor \times Emprestimo)$.

nomeCliente	**Devedor. numeroEmprestimo**	**nomeAgencia**	**Emprestimo. numeroEmprestimo**	**total**
Jones	L-17	Perryridge	L-15	1500
Jones	L-17	Perryridge	L-16	1300
Smith	L-23	Perryridge	L-15	1500
Smith	L-23	Perryridge	L-16	1300
Hayes	L-15	Perryridge	L-15	1500
Hayes	L-15	Perryridge	L-16	1300
Jackson	L-14	Perryridge	L-15	1500
Jackson	L-14	Perryridge	L-16	1300
Curry	L-93	Perryridge	L-15	1500
Curry	L-93	Perryridge	L-16	1300
Smith	L-11	Perryridge	L-15	1500
Smith	L-11	Perryridge	L-16	1300
Williams	L-17	Perryridge	L-15	1500
Williams	L-17	Perryridge	L-16	1300
Adams	L-16	Perryridge	L-15	1500
Adams	L-16	Perryridge	L-16	1300

empréstimo na agência Perryridge, como mostrado na Tabela 3.6.

Tabela 3.6: Resultado da Expressão 3.7.

nomeCliente	**Devedor. numeroEmprestimo**	**nomeAgencia**	**Emprestimo. numeroEmprestimo**	**total**
Hayes	L-15	Perryridge	L-15	1500
Adams	L-16	Perryridge	L-16	1300

Finalmente, uma vez que queiramos apenas *nomeCliente*, podemos fazer uma projeção:

$$\begin{aligned}&\pi_{nomeCliente}(\sigma_{Devedor.numeroEmprestimo=Emprestimo.numeroEmprestimo}\\&(\sigma_{nomeAgencia=\text{``}Perryridge''}(Devedor \times Emprestimo)))\end{aligned} \tag{3.8}$$

O resultado dessa expressão é mostrado na Tabela 3.7 e é a resposta correta à nossa consulta.

Tabela 3.7: Resultado da Expressão 3.8.

nomeCliente
Hayes
Adams

4. Operações Relacionais Binárias: Junção e

4.1 A Operação Junção

A operação JUNÇÃO, indicada por $\bowtie$, é usada para combinar tuplas relacionadas de duas relações em uma única tupla "maior". Esta operação é muito importante para qualquer banco de dados relacional com mais de uma relação única, porque nos permite processar relacionamentos entre as relações.

Frequentemente, é desejável simplificar certas consultas que exijam um produto cartesiano. Tipicamente, uma consulta em um produto cartesiano envolve uma seleção de operações sobre o resultado desse produto cartesiano. Considere a consulta "encontrar todos os nomes dos clientes que tenham um empréstimo no banco e encontrar o total emprestado". Primeiro, formamos o produto cartesiano das relações Devedor e Emprestimo. Então, selecionamos aquelas tuplas que pertençam somente ao mesmo numeroEmprestimo e, a seguir, projetamos o resultado para nomeCliente, numeroEmprestimo e total:

$$\pi_{nomeCliente,Emprestimo.numeroEmprestimo,total} \\ (\sigma_{Devedor.numeroEmprestimo=Emprestimo.numeroEmprestimo}(Devedor \times Emprestimo)) \tag{4.1}$$

A JUNÇÃO NATURAL é uma operação binária que nos permite combinar certas seleções e um produto cartesiano dentro de uma operação. As operações de JUNÇÃO NATURAL formam um produto cartesiano de seus dois argumentos, promovem uma seleção obedecendo à equivalência dos atributos que aparecem em ambos os esquemas de relação e, finalmente, removem os atributos em duplicidade.

Embora a definição de junções naturais seja complicada, a operação é fácil de ser aplicada. Como ilustração, consideremos novamente o exemplo "encontrar os nomes de todos os clientes que tenham um empréstimo no banco e encontrar o total emprestado". Essa consulta pode ser expressa usando uma junção natural, como segue:

$$\pi_{nomeCliente,numeroEmprestimo,total}(Devedor \bowtie Emprestimo) \tag{4.2}$$

Desde que os esquemas para Devedor e Emprestimo têm o atributo numeroEmprestimo em comum, a operação JUNÇÃO NATURAL considera somente pares de tuplas que têm o mesmo valor em numeroEmprestimo. Ela combina cada um desses pares de tuplas em uma única tupla por meio da união dos dois esquemas. Após a projeção, obtemos a relação mostrada na Tabela 4.1.

Tabela 4.1: Resultado de $\pi_{nomeCliente,numeroEmprestimo,total}(Devedor \bowtie Emprestimo)$

nomeCliente	**numeroEmprestimo**	**total**
Jones	L-17	1000
Smith	L-23	2000
Hayes	L-15	1500
Jackson	L-14	1500
Curry	L-93	500
Smith	L-11	900
Williams	L-17	1000
Adams	L-16	1300

4.1.1 Junção Externa

A operação de JUNÇÃO EXTERNA é uma extensão da operação de JUNÇÃO para tratar informações omitidas. Podemos usar a operação de JUNÇÃO EXTERNA para evitar a perda de informações. Existem três formas de usar essa operação: JUNÇÃO EXTERNA À ESQUERDA, denotada por ⟕; JUNÇÃO EXTERNA À DIREITA, denotada por ⟖; e JUNÇÃO EXTERNA TOTAL, denotada por ⟗.

A JUNÇÃO EXTERNA À ESQUERDA toma todas as tuplas da relação à esquerda que não encontraram par entre as tuplas da relação à direita, preenche a tupla com valores nulos para todos os outros atributos da relação à direita e a adiciona ao resultado da junção natural. Todas as informações da relação à esquerda são apresentadas no resultado da junção externa à esquerda.

Um exemplo de junção externa à esquerda poderia ser aplicado para listar todos os clientes com os seus possíveis empréstimos. Esta consulta é semelhante à apresentada na Expressão 4.2, mas agora não queremos omitir da listagem os clientes que não possuem empréstimos. Desta forma, temos:

$$\pi_{nomeCliente,numeroEmprestimo,total}(Devedor ⟕ Emprestimo) \tag{4.3}$$

O resultado da consulta pode ser visualizado na Tabela 4.2, onde percebe-se que a tupla <*Turner, L-21*> da relação Devedor, que não tem correspondência na relação Emprestimo, é relacionada no resultado tendo o atributo *total* preenchido com valor *nulo*.

Tabela 4.2: Resultado de $\pi_{nomeCliente,numeroEmprestimo,total}(Devedor ⟕ Emprestimo)$

nomeCliente	**numeroEmprestimo**	**total**
Jones	L-17	1000
Smith	L-23	2000
Hayes	L-15	1500
Jackson	L-14	1500
Curry	L-93	500
Smith	L-11	900
Williams	L-17	1000
Adams	L-16	1300
Turner	L-21	*nulo*

A JUNÇÃO EXTERNA À DIREITA é simétrica à junção externa à esquerda: as tuplas da relação à direita que não encontraram par na relação da esquerda são preenchidas com nulos e adicionadas ao resultado da junção natural. Assim, todas as informações da relação da direita se apresentam no resultado da junção externa à direita.

Aplicando a junção externa à direita na Expressão 4.2, temos:

$$\pi_{nomeCliente,numeroEmprestimo,total}(Devedor ⟖ Emprestimo) \tag{4.4}$$

que resultará na relação mostrada na Tabela 4.3, onde percebemos que a tupla <*Mianus, L-20, 500*> da relação Emprestimo é apresentada no resultado mesmo não estando relacionada com alguma tupla da relação Devedor.

Tabela 4.3: Resultado de $\pi_{nomeCliente,numeroEmprestimo,total}(Devedor ⟖ Emprestimo)$

nomeCliente	**numeroEmprestimo**	**total**
Jones	L-17	1000
Smith	L-23	2000
Hayes	L-15	1500
Jackson	L-14	1500
Curry	L-93	500
Smith	L-11	900
Williams	L-17	1000
Adams	L-16	1300
nulo	L-20	500

A JUNÇÃO EXTERNA TOTAL faz ambas as operações, preenche as tuplas da relação da esquerda que não encontraram par na relação da direita, assim como também preenche as tuplas da relação da direita que não encontraram par na relação da esquerda, adicionando-as ao resultado da junção.

Novamente, aplicando a junção externa total à Expressão 4.2, temos:

$$\pi_{nomeCliente,numeroEmprestimo,total}(Devedor ⟗ Emprestimo) \tag{4.5}$$

que trará o resultado mostrado na Tabela 4.4, onde, tanto a tupla <*Turner, L-21*> da relação Devedor, quanto a tupla <*Mianus, L-20, 500*> da relação Emprestimo, são listadas.

Tabela 4.4: Resultado de $\pi_{nomeCliente,numeroEmprestimo,total}(Devedor ⟗ Emprestimo)$

nomeCliente	**numeroEmprestimo**	**total**
Jones	L-17	1000
Smith	L-23	2000
Hayes	L-15	1500
Jackson	L-14	1500
Curry	L-93	500
Smith	L-11	900
Williams	L-17	1000
Adams	L-16	1300
Turner	L-21	*nulo*
nulo	L-20	500

Assim, o resultado da junção externa total, nada mais é do que a união do resultado apresentado pela junção externa à esquerda com o resultado apresentado pela junção externa à direita.

4.2 A Operação de Divisão

A operação de divisão, simbolizada por $\div$, é usada nas consultas nas quais se emprega a frase "para todos". Suponha que desejemos encontrar **todos** os clientes que tenham conta em **todas** as agências localizadas no Brooklyn. Podemos obter todas as agências no Brooklyn pela expressão:

$$r_1 = \pi_{nomeAgencia}(\sigma_{cidadeAgencia=\text{``Brooklyn''}}(Agencia)) \tag{4.6}$$

A relação resultante dessa expressão aparece na Tabela 4.5

Tabela 4.5: Resultado de $\pi_{nomeAgencia}(\sigma_{cidadeAgencia=\text{``Brooklyn''}}(Agencia))$

nomeAgencia
Brighton
Downtown

Podemos encontrar todos os pares (nomeCliente, nomeAgencia) para os quais os clientes têm uma conta em uma agência, escrevendo:

$$r_2 = \pi_{nomeCliente,nomeAgencia}(Depositante \bowtie Conta) \tag{4.7}$$

A Tabela 4.6 mostra a relação resultante dessa expressão.

Tabela 4.6: Resultado de $\pi_{nomeCliente,nomeAgencia}(Depositante \bowtie Conta)$

nomeCliente	**nomeAgencia**
Jones	Downtown
Smith	Mianus
Hayes	Perryrudge
Turner	Round Hill
Williams	Perryridge
Lindsay	Redwood
Johson	Brighton
Jones	Brighton

Agora, precisamos encontrar os clientes que aparecem em r_2, com **todos** os nomes de agência em r_1 (ou seja, os clientes listados em r_2 que contém contas em **todas** as agências listadas em r_1). A operação que proporciona exatamente os clientes em questão é a operação de divisão.

A fórmula para a consulta é escrita da seguinte forma:

$$\begin{aligned}&\pi_{nomeCliente,nomeAgencia}(Depositante \bowtie Conta)\\&\div\pi_{nomeAgencia}(\sigma_{cidadeAgencia=\text{``Brooklyn''}}(Agencia))\end{aligned} \tag{4.8}$$

O resultado para essa expressão é uma relação que possui o esquema (nomeCliente) e que contém a tupla (Johnson), assim como mostrado na Tabela 4.7

Tabela 4.7: Resultado da divisão entre a relação da Tabela 4.6 pela relação da Tabela 4.5, resultando na listagem de clientes que possuem contas em **todas** as agências do Brooklyn

nomeCliente
Johnson

5. Outras Operações Relacionais

5.1 A Operação Designação

É conveniente, às vezes, escrever expressões em álgebra relacional com uma designação para a relação, de modo a usá-la como uma variável temporária. A operação de DESIGNAÇÃO, denotada por $\leftarrow$, trabalha de maneira similar à designação (*assignment*) em linguagens de programação. Para ilustrar esta operação, consideremos a definição de divisão apresentada na Seção 4.2. Podemos escrever a expressão com o seguinte conjunto:

$$\begin{aligned}
&r_1 \leftarrow \pi_{nomeAgencia}(\sigma_{cidadeAgencia=\text{``}Brooklyn''}(Agencia))\\
&r_2 \leftarrow \pi_{nomeCliente,nomeAgencia}(Depositante \bowtie Conta)\\
&r_2 \div r_1
\end{aligned} \tag{5.1}$$

A aplicação de uma DESIGNAÇÃO não resulta em uma relação sendo exibida para o usuário. O resultado da expressão à direita do $\leftarrow$ é a designação de uma variável relação à esquerda do $\leftarrow$. Esta variável relação pode ser usada em expressões subsequentes.

Com a operação de designação, uma consulta pode ser escrita como um programa sequencial, consistindo de uma série de designações seguidas por uma expressão cujo valor é exibido como o resultado daquela consulta. Para consultas em álgebra relacional, a designação deve ser feita sempre para uma variável relação temporária. Designações para relações permanentes constituem uma modificação no banco de dados.

5.2 A Operação Renomear

Ao contrário das relações de um banco de dados, o resultado de uma expressão em álgebra relacional não possui um nome que possa ser usado para referenciá-la. Seria útil se pudéssemos dar nome a elas; o operador RENOMEAR, representado pela letra grega *rho* (ρ), permite-nos executar esse tipo de tarefa. Dada a expressão em álgebra relacional E, a expressão

$$\rho_x(E) \tag{5.2}$$

tem por resultado a expressão E sob o nome x.

Uma relação R é considerada como uma expressão (trivial) de álgebra relacional. Assim, podemos também aplicar a operação RENOMEAR à relação R para obter a mesma relação sob um novo nome.

A segunda forma de usar a operação RENOMEAR é a que segue. Assuma que uma expressão E em álgebra relacional seja de ordem primária. Então a expressão:

$$\rho_{x(A_1,A_2,\ldots,A_n)}(E) \tag{5.3}$$

retorna o resultado da expressão E sob o nome x, com os atributos recebendo novos nomes, $A_1, A_2, \ldots, A_n$.

6. Resumo das Operações Relacionais

A Tabela 6.1 apresenta um resumo das operações da álgebra relacional que foram descritas nos capítulos anteriores.

Tabela 6.1: Operadores da Álgebra Relacional

Símbolo	**Operação**	**Sintaxe**	**Tipo**
π	Projeção (project)	$\pi_{atributos}(Relacao)$	Primitiva - Unária
σ	Seleção (select)	$\sigma_{condicao}(Relacao)$	Primitiva - Unária
$\times$	Produto Cartesiano (cartesian product)	$Relacao1 \times Relacao2$	Primitiva - Binária
$\cup$	União (union)	$Relacao1 \cup Relacao2$	Primitiva - Binária
$-$	Diferença entre Conjuntos (set difference)	$Relacao1 - Relacao2$	Primitiva - Binária
$\cap$	Conjunto Intersecção (set intersection)	$Relacao1 \cap Relacao2$	Derivada - Binária
$\bowtie$	Junção Natural (natural join)	$Relacao1 \bowtie Relacao2$	Derivada - Binária
$\div$	Divisão (division)	$Relacao1 \div Relacao2$	Derivada - Binária
ρ	Renomear Relação ou Atributo	$\rho_{Nome}(Relacao)$	Especial - Unária
$\leftarrow$	Designação, ou Atribuição (assignment)	$variavel \leftarrow Relacao$	Especial - Unária

7. Referências

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson, 2011.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados**. 3. ed. São Paulo: Makron, 1999.