Nota: Este material representa uma cópia resumida do capítulo 4 do livro Sistemas de Gerenciamento Sistemas de Gerenciamento de Banco de Dados - Ramakrishnan Gehrke (3ª edição), versão disponível em pdf na internet.

Introdução à Álgebra Relacional

Álgebra Relacional

Ao definir a álgebra e o cálculo relacionais, a alternativa de referenciar os campos pela posição é mais conveniente do que referenciá-los pelo nome: as consultas normalmente envolvem a computação de resultados intermediários, que também são instâncias de relações; e se usamos nomes de campo para referenciar os campos, a definição dos construtores da linguagem de consulta deve especificar os nomes dos campos para todas as instâncias de relação intermediárias. Isso pode ser tedioso e é realmente um item secundário, porque podemos referenciar os campos pela posição de qualquer maneira. Por outro lado, os nomes de campo tornam as consultas mais legíveis.

Por essas considerações, usamos a notação posicional para definir a álgebra e cálculo relacionais. Também introduzimos convenções simples que permitem que relações intermediárias 'herdem' os nomes de campo, por conveniência. Apresentaremos algumas consultas exemplo usando o seguinte esquema:

Marinheiros(id-marin: integer, nome-marin: string, avaliação: integer, idade: real)
Barcos(id-barco: integer, nome-barco: string, cor: string)
Reservas(id-marin: integer, id-barco: integer, dia: date)

Os campos-chaves encontram-se sublinhados, e o domínio de cada campo encontra-se listado após o nome do campo. Assim, id-marin é a chave de Marinheiros, id-barco é a chave de Barcos e todos os três campos juntos formam a chave de Reservas. Os campos em uma instância de uma dessas relações são referenciados pelo nome, ou posicionalmente, usando a ordem pela qual elas foram listadas. Em diversos exemplos ilustrando os operadores da álgebra relacional, usamos as instâncias M1 e M2 (de Marinheiros) e R1 (de Reservas) ilustrados nas Figuras 4.1, 4.2 e 4.3, respectivamente.

id-marin	*nome-marin*	*avaliação*	*idade*
22	Dustin	7	45,0
31	Lubber	8	55,5
58	Rusty	10	35,0

Figura 4.1 Instância $M1$ de Marinheiros.

id-marin	*nome-marin*	*avaliação*	*idade*
28	Yuppy	9	35,0
31	Lubber	8	55,5
44	Guppy	5	35,0
58	Rusty	10	35,0

Figura 4.2 Instância $M2$ de Marinheiros.

id-marin	*id-barco*	*dia*
22	101	10/10/96
58	103	11/12/96

Figura 4.3 Instância $R1$ de Reservas.

A **álgebra relacional** é uma das duas linguagens de consulta formais associadas ao modelo relacional. As consultas na álgebra são compostas usando uma coleção de operadores. Uma propriedade fundamental é a de que todo operador na álgebra aceita (uma ou duas) instâncias de relação como argumentos e retorna uma instância de relação como resultado. Esta propriedade facilita a composição de operadores para formar uma consulta complexa — uma **expressão de álgebra relacional** é:

- recursivamente definida como uma relação,
- um operador de álgebra unário aplicado a uma única expressão,
- ou um operador de álgebra binário aplicado a duas expressões.

Nas seções seguintes, descreveremos os operadores **básicos da álgebra (seleção, projeção, união, produto cartesiano e diferença)**, assim como alguns operadores adicionais que podem ser definidos em termos dos operadores básicos, mas que aparecem com frequência suficiente para justificar atenção especial. A natureza procedural da álgebra nos permite considerar uma expressão algébrica como uma receita, ou um plano, para avaliar uma consulta, e os sistemas relacionais realmente usam as expressões algébricas para representar os planos de avaliação das consultas.

Seleção e Projeção

https://www.youtube.com/watch?v=n8JrEmlORAY

A álgebra relacional inclui operadores para selecionar linhas de uma relação (σ) e para projetar colunas (π). Essas operações nos permitem manipular os dados em uma única relação. Considere a instância da relação Marinheiros ilustrada na Figura 4.2, denotada como M2. Podemos recuperar as linhas correspondentes aos marinheiros experientes usando o operador σ. A expressão resulta na relação ilustrada na Figura 4.4. O subscrito avaliação>8 especifica o critério de seleção a ser aplicado ao recuperar tuplas.

$$\sigma_{avaliação>8}(M2)$$

id-marin	*nome-marin*	*avaliação*	*idade*
28	Yuppy	9	35,0
58	Rusty	10	35,0

Figura 4.4 $\sigma_{avaliação>8}(M2)$.

O operador de seleção σ especifica as tuplas a serem mantidas por meio de uma condição de seleção. Em geral, a condição de seleção é uma combinação Booleana (isto é, uma expressão usando os conectivos lógicos ∧ e ∨) de termos que têm a forma atributo op constante ou atributo1 op atributo2, sendo op um dos operadores de comparação <, <=, =, π, >=, ou >. A referência a um atributo pode ser pela posição (na forma .i ou i) ou pelo nome (na forma .nome ou nome). O esquema do resultado de uma seleção é o esquema da instância de relação de entrada.

O operador de projeção π nos permite extrair colunas de uma relação; por exemplo, podemos localizar todos os nomes e avaliações dos marinheiros usando π. A expressão resulta na relação ilustrada na Figura 4.5. O subscrito nome-marin,avaliação especifica os campos a serem mantidos; os demais campos são 'lançados fora'. O esquema do resultado de uma projeção é determinado, obviamente, pelos campos que são projetados.

$$\pi_{nome\text{-}marin, avaliação}(M2)$$

nome-marin	*avaliação*
Yuppy	9
Lubber	8
Guppy	5
Rusty	10

Figura 4.5 $\pi_{nome\text{-}marin,avaliação}(M2)$.

Suponha que desejemos localizar apenas as idades dos marinheiros. A expressão resulta na relação ilustrada da Figura 4.6. O ponto importante a observar é que, embora haja três marinheiros de idade 35, uma única tupla com idade=35,0 aparece no resultado da projeção. Isso é consequência da definição de uma relação como um conjunto de tuplas. Na prática, os sistemas reais normalmente omitem a etapa custosa de eliminação de tuplas duplicadas, gerando relações que são multiconjuntos. Entretanto, nossa discussão de álgebra e cálculo relacionais supõe que a eliminação de duplicatas é sempre realizada de forma que as relações são sempre conjuntos de tuplas.

$$\pi_{idade}(M2)$$

idade
35,0
55,5

Figura 4.6 $\pi_{idade}(M2)$.

Como o resultado de uma expressão da álgebra relacional é sempre uma relação, podemos usar uma expressão sempre que uma relação é esperada. Por exemplo, podemos computar os nomes e avaliações dos marinheiros bem avaliados combinando duas das consultas anteriores. A expressão produz o resultado ilustrado na Figura 4.7. Ele é obtido aplicando-se a seleção a M2 (para obter a relação ilustrada na Figura 4.4) e depois aplicando-se a projeção.

$$\pi_{nome\text{-}marin, avaliação}(\sigma_{avaliação>8}(M2))$$

nome-marin	*avaliação*
Yuppy	9
Rusty	10

Figura 4.7 $\pi_{nome\text{-}marin, avaliação}(\sigma_{avaliação>8}(M2))$.

Operações de Conjunto

As seguintes operações padrão sobre conjuntos também estão disponíveis na álgebra relacional:
união (∪), intersecção (∩), diferença de conjunto (–) e produto cartesiano (x).

- **União**: R ∪ M retorna uma instância de relação contendo todas as tuplas que ocorrem na instância de relação R ou na instância de relação M (ou em ambas). R e M devem ser compatíveis à união, e o esquema do resultado é definido de forma idêntica ao esquema de R. Duas instâncias de relação são consideradas **compatíveis à união** se as seguintes condições forem satisfeitas:
  - elas têm o mesmo número de campos, e
  - os campos correspondentes, considerados na ordem, da esquerda para a direita, têm o mesmo domínio.
- **Intersecção**: R ∩ M retorna uma instância de relação contendo todas as tuplas que ocorrem em ambas R e M. As relações R e M devem ser compatíveis à união, e o esquema do resultado é definido de forma idêntica ao esquema de R.
- **Diferença de conjunto**: R – M retorna uma instância de relação contendo todas as tuplas que ocorrem em R, mas não em M. As relações R e M devem ser compatíveis à união, e o esquema do resultado é definido de forma idêntica ao esquema de R.
- **Produto cartesiano**: R x M retorna uma instância de relação cujo esquema contém todos os campos de R (na mesma ordem em que eles aparecem em R) seguidos de todos os campos de M (na mesma ordem em que eles aparecem em M). O resultado de R x M contém uma tupla <r, s> (a concatenação das tuplas r e s) para cada par de tuplas r ∈ R, s ∈ M.

Usamos a convenção de que os campos de R x M herdam nomes dos campos correspondentes de R e M. É possível que ambos R e M contenham um ou mais campos com o mesmo nome; esta situação cria um conflito de nomeação. Os campos correspondentes em R x M não são nomeados e são referenciados somente pela posição. Nas definições anteriores, observe que cada operador pode ser aplicado a instâncias de relação que são computadas usando uma (sub)expressão de álgebra relacional.

Ilustramos agora essas definições através de diversos exemplos. A união de M1 e M2 é ilustrada na Figura 4.8. Os campos são listados na ordem; os nomes de campos também são herdados de M1. M2 têm os mesmos nomes de campo, é claro, uma vez que ela também é uma instância de Marinheiros. Em geral, os campos de M2 podem ter nomes diferentes; lembre-se de que exigimos apenas correspondência dos domínios. Observe que o resultado é um conjunto de tuplas. As tuplas que aparecem em ambas M1 e M2 aparecem apenas uma vez em M1 ∪ M2. Além disso, M1 ∪ R1 não é uma operação válida porque as duas relações não são compatíveis à união. A intersecção de M1 e M2 é ilustrada na Figura 4.9, e a diferença de conjunto M1 – M2 é ilustrada na Figura 4.10.

id-marin	*nome-marin*	*avaliação*	*idade*
22	Dustin	7	45,0
31	Lubber	8	55,5
58	Rusty	10	35,0
28	Yuppy	9	35,0
44	Guppy	5	35,0

Figura 4.8 $M1 \cup M2$.

id-marin	*nome-marin*	*avaliação*	*idade*
31	Lubber	8	55,5
58	Rusty	10	35,0

Figura 4.9 $M1 \cap M2$.

id-marin	*nome-marin*	*avaliação*	*idade*
22	Dustin	7	45

Figura 4.10 $M1 - M2$.

O resultado do produto cartesiano M1 x R1 é ilustrado na Figura 4.11. Como ambas R1 e M1 têm um campo chamado id-marin, pela nossa convenção de nomes de campo, os dois campos correspondentes em M1 x R1 não são nomeados, e são referenciados somente pela posição em que eles aparecem na Figura 4.11. Os campos em M1 x R1 têm os mesmos domínios que os campos correspondentes em R1 e M1. Na Figura 4.11, id-marin é listado entre parênteses para enfatizar que não é um nome de campo herdado; apenas o domínio correspondente é herdado.

(id-marin)	*nome-marin*	*avaliação*	*idade*	*(id-marin)*	*id-barco*	*dia*
22	Dustin	7	45,0	22	101	10/10/96
22	Dustin	7	45,0	58	103	11/12/96
31	Lubber	8	55,5	22	101	10/10/96
31	Lubber	8	55,5	58	103	11/12/96
58	Rusty	10	35,0	22	101	10/10/96
58	Rusty	10	35,0	58	103	11/12/96

Figura 4.11 $M1 \times R1$.

Renomear

Introduzimos o operador **renomear** ρ para evitar conflitos de nome após uma expressão de álgebra. Por exemplo, a expressão ρ(C(1 → id-marin1, 5 → id-marin2), M1 x R1) retorna uma relação que contém as tuplas ilustradas na Figura 4.11 e tem o seguinte esquema:

C(id-marin1: integer, nome-marin: string, avaliação: integer, idade: real, idmarin2: integer, id-barco: integer, dia: date)

É normal incluir alguns operadores adicionais na álgebra, mas todos eles podem ser definidos em termos dos operadores que definimos até agora. (De fato, o operador renomear é necessário apenas por conveniência sintática, e mesmo o operador é redundante; R M pode ser definido como R – (R – M).) Consideramos esses operadores adicionais e suas definições em termos dos operadores básicos nas duas próximas subseções.

Junções

A operação **junção** é uma das operações mais úteis na álgebra relacional e a maneira mais comumente usada para combinar informações de duas ou mais relações.

Junções Condicionais

A **Junção Theta,** também conhecida como **Junção Condicional** é representada pelo símbolo ⋈**,** e baseia-se em uma combinação dos operadores produto cartesiano e seleção. Ela cria uma relação pela combinação dos campos de uma relação com aquelas de uma outra baseada em uma comparação de valores entre colunas que não necessariamente tem o mesmo nome. A sintaxe da junção theta é:

<tabela1> ⋈ *<critério de seleção> <tabela2>,* Onde:

<tabela> é o nome de uma tabela, ou expressão algébrica que resulte em uma tabela
<critério de seleção> é uma expressão booleana envolvendo literais e valores das colunas das duas tabelas envolvidas.

$$R \bowtie_C M = \sigma_C (R \times M)$$

(id-marin)	*nome-marin*	*avaliação*	*idade*	*(id-marin)*	*id-barco*	*dia*
22	Dustin	7	45,0	58	103	11/12/96
31	Lubber	8	55,5	58	103	11/12/96

Figura 4.12 $M1 \bowtie_{M1.id\text{-}marin < R1.id\text{-}marin} R1$.

Assim, ⋈ é definida como um produto cartesiano seguido de uma seleção. Observe que a condição c pode (e normalmente o faz) referenciar os atributos de ambas R e M. A referência a um atributo de uma relação, digamos, R, pode ser pela posição (no formato R.i) ou pelo nome (no formato R.nome). Como um exemplo, o resultado de M1 ⋈ M1.id-marin < R1.id-marin R1 é ilustrado na Figura 4.12.

TabelaA

c	d
ValorA	50
ValorB	65
ValorC	70

TabelaB

e	f
ValorD	50
ValorE	80
ValorF	70

Com a relação abaixo iremos reproduzir um produto cartesiano das duas tabelas selecionando apenas as linhas em que a coluna "d" da tabela "TabelaA" seja igual a coluna "f" da tabela "TabelaB".

TabelaA ⋈ TabelaA.d = TabelaB.f TabelaB

c	d	e	f
ValorA	50	ValorD	50
ValorC	70	ValorF	70

Eqüijunção

Ilustramos M1 ⋈ M1.id-marin=R1.id-marin R1 na Figura 4.13. Observe que apenas um campo chamado id-marin aparece no resultado.

(id-marin)	*nome-marin*	*avaliação*	*idade*	*id-barco*	*dia*
22	Dustin	7	45,0	101	10/10/96
58	Rusty	10	35,0	103	11/12/96

Figura 4.13 $M1 \bowtie_{M1.id\text{-}marin=R1.id\text{-}marin} R1$.

Junção Natural

Um caso ainda mais especial da operação junção R ⋈ M é uma eqüijunção na qual as igualdades são especificadas para todos os campos que têm os mesmos nomes em R e M. Neste caso, podemos simplesmente omitir a condição de junção; o padrão é que a condição de junção seja uma coleção de igualdades em todos os campos comuns. Chamamos este caso especial de **junção natural**, e ela tem a boa propriedade de garantir que o resultado não tenha dois campos com os mesmos nomes.

A expressão de eqüijunção M1 ⋈ M1.id-marin=R1.id-marin R1 é, de fato, uma junção natural e pode simplesmente ser denotada como M1 ⋈ R1, uma vez que o único campo comum é id-marin. Se as duas relações não tiverem nenhum atributo em comum, M1 ⋈ R1 é apenas o produto cartesiano.

Divisão

Curso Álgebra Relacional: Operador Divide

O operador **divisão** ÷ é útil para expressar certos tipos de consultas, como, por exemplo, “Localize os nomes dos marinheiros que reservaram todos os barcos”. Note que os sistemas de banco de dados não tentam explorar a semântica da divisão implementando-o como um operador distinto.

Finalizado

Estudante	Tarefa
Fred	Basedados1
Fred	Basedados2
Fred	Compiladores1
Pedro	Basedados1
Pedro	Compiladores1
Sara	Basedados1
Sara	Basedados2

ProjectoBD

Tarefa
Basedados1
Basedados2

Finalizado ÷ ProjectoBD

Estudante
Fred
Sara

Se ProjectoBD contém todas as tarefas do projecto Basedados, deduz-se que o resultado da divisão do exemplo contém exactamente os estudantes que completaram ambas as tarefas no projecto Basedados. É normalmente requerido que os nomes do atributo no cabeçalho de S são subsets do mesmo que R porque de outra maneira o resultado da operação vai ser vazio.

A operação de divisão entre duas relações A ÷ B traz uma nova relação C com todas as tuplas que possuem campos em comum nas duas relações.

Ex: (cliente) ÷ (cpfProcurado)

nome	cpf
Daniel	1234
Carlos	2233
Diego	9908
Robert	3276

÷

cpf
1234
9908

=

nome
Daniel
Diego

Referencias:

- Fundamentos de Sistemas de Gerenciamento de Banco de Dados - Elmasri & Navathe (6ª edição)
- https://spaceprogrammer.com/bd/aprendendo-as-principais-operacoes-da-algebra-relacional/
- Sistemas de Gerenciamento Sistemas de Gerenciamento de Banco de Dados - Ramakrishnan Gehrke (3ª edição)

Isenção de Responsabilidade: