

ÁLGEBRA E CÁLCULO RELACIONAL*

Giselle Palermo e Evandro Bacarin

9 de setembro de 2005

1 Introdução

Sistemas de Gerenciamento de Banco de Dados (SGBD) utilizam linguagens de consulta para recuperar dados armazenados num banco de dados. A *álgebra relacional* e o *cálculo relacional* são as linguagens de consultas formais associadas ao modelo relacional.

O texto aborda estas duas linguagens. A seção 2 é dedicada à álgebra relacional, descrevendo seus principais operadores. A seção 3 apresenta o cálculo relacional em suas duas vertentes: o cálculo relacional de tuplas e o cálculo relacional de domínios. As três linguagens são comparadas na seção 4, na qual são apresentados alguns exemplos de consultas escritas nestes três dialetos. Por fim, a seção 5 analisa se estas três linguagens possuem o mesmo poder de expressão.

2 Álgebra Relacional

A *álgebra relacional* é uma linguagem de consulta que utiliza operadores para selecionar linhas ou colunas de uma relação. Cada consulta descreve um procedimento para computar a resposta desejada. Na álgebra relacional, as instâncias de relação aparecem tanto como argumentos dos operadores, como resultados das consultas. A seguir, são descritos os seus principais operadores.

2.1 Seleção (σ)

O operador σ seleciona um subconjunto de linhas da relação que satisfazem uma certa condição, e tem como resultado uma nova relação cujo esquema é igual ao da relação de entrada.

Considere a relação S1 (Fig. 1) que contém dados sobre marinheiros. A coluna *sid* contém um código que identifica unicamente um marinheiro na relação, as outras colunas contém o seu nome, a sua idade e um atributo que indica o seu grau de experiência.

A consulta $\sigma_{rating > 8}(S1)$ seleciona os marinheiros da relação S1 que satisfazem a condição *rating* > 8. O resultado é apresentado na Fig. 2. Observe que o esquema resultante é idêntico ao esquema S1.

*Este trabalho foi produzido como parte da avaliação do curso MO-410 Introdução a Banco de Dados oferecido pelo Instituto de Computação da Unicamp em 2005/1 e teve como fonte primária o livro - Ramakrishnan, R. and Gehrke, J. (2003) "Data Management Systems", McGraw-Hill, 3rd edition.

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

Figura 1: Instância S1 de marinheiros

sid	sname	rating	age
28	yuppy	9	35.0
58	rusty	10	35.0

Figura 2: $\sigma_{rating>8}(S1)$

2.2 Projeção (π)

Uma relação pode possuir muitas colunas, porém, para uma certa consulta, apenas são de interesse algumas delas. Para estes casos, a operação de projeção produz uma nova relação que contém um subconjunto das colunas da relação original.

Por exemplo, a consulta $\pi_{age}(S1)$ retorna somente as idades dos marinheiros (Fig. 3). Note que, embora três marinheiros da relação S1 tenham 35 anos, apenas uma tupla com o valor 35 aparece no resultado da projeção.

age
35.0
55.5

Figura 3: $\pi_{age}(S1)$

A consulta $\pi_{sname, rating}(\sigma_{rating>8}(S1))$ combina os operadores de seleção e projeção e retorna o nome e classificação dos marinheiros com classificação superior a oito. Ela pode ser entendida com se fosse resolvida em dois passos: no primeiro passo é feita a seleção e, em seguida, são projetadas as colunas de interesse (*sname* e *rating*). O seu resultado é mostrado na Fig. 4.

2.3 União (\cup)

O operador de união recebe como argumentos duas relações compatíveis e produz uma terceira relação que contém todas as tuplas das duas relações, sem repetição. As duas relações são compatíveis se possuem o mesmo número de colunas e as respectivas colunas são do mesmo domínio.

Considere outra relação de marinheiros S2 (Fig. 5).

A união das instâncias S2 (Fig. 5) e S1 (Fig. 1) resulta em uma instância de relação com as linhas de S2 seguidas das linhas de S1, como na Fig. 6. O esquema da relação resultante é o mesmo de S2.

sname	rating
yuppy	9
rusty	10

Figura 4: $\pi_{sname, rating}(\sigma_{rating > 8}(S1))$

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

Figura 5: Instância S2 de marinheiros

2.4 Intersecção (\cap)

A operação de intersecção é similar à operação de união. Também tem como argumentos duas relações compatíveis e o seu resultado é uma nova instância contendo todas as tuplas que ocorrem em ambas as relações. Por exemplo, a Fig. 7 mostra o resultado da intersecção das relações S1 e S2.

2.5 Diferença de Conjuntos ($-$)

A diferença entre duas relações, R e S, retorna uma relação contendo todas as tuplas que ocorrem em R, mas não ocorrem em S. Semelhante à operação de união, ambas relações devem ser compatíveis.

A Fig. 8 mostra o resultado da consulta S2 - S1. Note que os marinheiros *lubber* e *rusty* aparecem na relação S1 e, por isso, não aparecem na relação resultante.

2.6 Renomeação (ρ)

O operador ρ é usado para atribuir um novo nome a uma relação ou a algumas de suas colunas. A operação de renomeação é da forma:

$$\rho(R_2(n_1, n_2, \dots, n_k), R_1)$$

tal que cada renomeação n_i tem a forma:

$$nome_antigo \rightarrow novo_nome \text{ ou } posicao \rightarrow novo_nome$$

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	guppy	5	35.0
28	yuppy	9	35.0

Figura 6: S2 \cup S1

sid	sname	rating	age
31	lubber	8	55.5
58	rusty	10	35.0

Figura 7: $S1 \cap S2$

sid	sname	rating	age
22	dustin	7	45.0

Figura 8: $S2 - S1$

e significa que é produzida uma nova relação, denominada R_2 , com os mesmos campos de R_1 e que algumas de suas colunas são renomeadas.

Considere, novamente, a relação da Fig. 8. A expressão abaixo, atribui o nome *RelDif* ao resultado da consulta $S2 - S1$ e renomeia a coluna *sname* para *name* e a terceira coluna (*rating*) para *rat*.

$$\rho(\text{RelDif}(sname \rightarrow name, 3 \rightarrow rat), S2 - S1)$$

A relação resultante é mostrada na Fig. 9.

sid	name	rat	age
22	dustin	7	45.0

Figura 9: Relação *RelDif*

2.7 Produto Cartesiano (\times)

No produto cartesiano $R \times S$, cada linha de R é emparelhada com todas as linha de S , resultando em um esquema composto por campos de R (na mesma ordem de R) seguido por campos de S (na mesma ordem de S). A relação resultante contém uma tupla (r,s) para cada par de tuplas $r \in R$ e $s \in S$.

Considere uma nova relação $R1$ que armazena as reservas de barcos por marinheiros (Fig. 10). Esta relação contém o código do marinheiro (*sid*) que fez a reserva, o código do barco (*bid*) que foi reservado e o dia (*day*) em que o barco será usado.

Na Fig. 11 é apresentado o resultado da consulta $S2 \times R1$. Note que o marinheiro *dustin* é combinado com cada uma das tuplas da relação $R1$. O mesmo acontece com os marinheiros *lubber* e *rusty*. Como existem 3 tuplas na relação $S2$ e 2 tuplas na relação $R1$, a relação resultante contém 6 tuplas.

Observe que ocorre um conflito de nomes, pois, tanto $S2$ e $R1$ têm uma coluna chamada *sid*. Para resolver este problema, é utilizado a operação de renomeação:

$$\rho(C(1 \rightarrow sid1, 5 \rightarrow sid2), S2 \times R1)$$

A relação resultante é mostrada na Fig. 12. Note que na primeira linha desta relação, os campos *sid1* e *sid2* possuem o mesmo valor (22). O campo *sid1* é oriundo da tabela $S2$ e contém o código do marinheiro *dustin*. Entretanto, o campo *sid2* é oriundo da tabela $R1$ e indica o código do marinheiro que reservou o barco *101*. Como os valores dos campos *sid1*

sid	bid	day
22	101	10/10/96
58	103	11/12/96

Figura 10: Instância R1 de reservas

(sid)	sname	rating	age	(sid)	bid	day
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 11: $S2 \times R1$

e $sid2$ são iguais, conclui-se que o barco 101 foi reservado por $dustin$ para o dia $10/10/96$. Por outro lado, na segunda linha, os valores de $sid1$ e $sid2$ são diferentes. Assim, **não** se pode concluir que o barco 103 foi reservado por $dustin$. Note que estes campos novamente terão o mesmo valor apenas na última linha da relação, indicando que o marinheiro $rusty$ reservou o barco 103 .

sid1	sname	rating	age	sid2	bid	day
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 12: $\rho(C(1 \rightarrow sid1, 5 \rightarrow sid2), S2 \times R1)$

Vale observar que, das seis linhas contidas na relação resultante, apenas duas contêm informação relevante. Para selecioná-las pode-se utilizar a operação de seleção:

$$\sigma_{sid1=sid2} \rho(C(1 \rightarrow sid1, 5 \rightarrow sid2), S2 \times R1)$$

O resultado produzido por esta consulta é mostrado na Fig. 13. A seção 2.8 apresenta consultas mais compactas que obtém o mesmo resultado.

2.8 Junções (\bowtie)

O produto cartesiano, como já mencionado, resulta numa relação que pode conter muitos dados incoerentes. Por isso, é necessária uma seleção posterior, descartando as tuplas indesejadas. A operação de junção é mais eficiente, e pode ser definida como um produto cartesiano seguido por seleções e projeções. Os tipos de junção são: condicional, equi-junção e natural.

sid1	sname	rating	age	sid2	bid	day
22	dustin	7	45.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

Figura 13: $\sigma_{sid1=sid2} \rho(C(1 \rightarrow sid1, 5 \rightarrow sid2), S2 \times R1)$

2.8.1 Junção Condicional

A junção condicional é a forma mais geral de junção. É equivalente a um produto cartesiano seguido por uma seleção que respeita uma condição c :

$$R \bowtie_c S = \sigma_c(R \times S)$$

A Fig. 14 mostra o resultado da consulta:

$$S2 \bowtie_{S2.sid < R1.sid} R1$$

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	58	103	11/12/96

Figura 14: $S2 \bowtie_{S2.sid < R1.sid} R1$

2.8.2 Equi-Junção

A junção condicional aceita como condição de junção expressões lógicas, como a da Fig. 14. Porém, em geral, a condição da junção requer apenas a igualdade entre os campos da relação, como, por exemplo:

$$S2 \bowtie_{S2.sid = R1.sid} R1$$

Em tais casos, a junção é denominada *equi-junção*. A Fig. 15 mostra um exemplo deste tipo de junção. Note que, é exibida apenas uma coluna *sid*, pois $R1.sid$ e $S2.sid$ possuem o mesmo valor.

sid	sname	rating	age	bid	day
22	dustin	7	45.0	101	10/10/96
58	rusty	10	35.0	103	11/12/96

Figura 15: $S2 \bowtie_{S2.sid = R1.sid} R1$

2.8.3 Junção Natural

A junção natural é uma equi-junção na qual a igualdade se aplica a todos os campos de mesmo nome das relações de entrada. Assim, a equi-junção

$$S2 \bowtie_{S2.sid = R1.sid} R1$$

pode ser simplesmente expressa por

$$S2 \bowtie R1$$

2.9 Divisão (/)

Considere duas instâncias de relação A e B, na qual A tem exatamente dois campos x e y , e B tem só um campo y , com o mesmo domínio do campo y de A. A divisão de A/B é o conjunto de todos valores x (na forma de tuplas unárias) tal que para cada valor de y em (uma tupla de) B, existe uma tupla $\langle x, y \rangle$ em A.

A Fig. 16 mostra alguns exemplos do uso da operação de divisão. Note que A/B1 produziu uma tabela com quatro linhas, uma vez que na tabela A, existem as tuplas $\langle s1, p2 \rangle$, $\langle s2, p2 \rangle$, $\langle s3, p2 \rangle$ e $\langle s4, p2 \rangle$. Note, também, que A/B3 produziu uma tabela com uma única linha contendo $s1$, uma vez que, na tabela A, apenas o valor $s1$ está relacionado com $p1$, $p2$ e $p4$ (por exemplo, $s2$ não está relacionado com $p4$).

A	<table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr><th>sno</th><th>pno</th></tr> </thead> <tbody> <tr><td>s1</td><td>p1</td></tr> <tr><td>s1</td><td>p2</td></tr> <tr><td>s1</td><td>p3</td></tr> <tr><td>s1</td><td>p4</td></tr> <tr><td>s2</td><td>p1</td></tr> <tr><td>s2</td><td>p2</td></tr> <tr><td>s3</td><td>p2</td></tr> <tr><td>s4</td><td>p2</td></tr> <tr><td>s4</td><td>p4</td></tr> </tbody> </table>	sno	pno	s1	p1	s1	p2	s1	p3	s1	p4	s2	p1	s2	p2	s3	p2	s4	p2	s4	p4	B1	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><th>pno</th></tr> <tr><td>p2</td></tr> </table>	pno	p2	A/B1	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><th>sno</th></tr> <tr><td>s1</td></tr> <tr><td>s2</td></tr> <tr><td>s3</td></tr> <tr><td>s4</td></tr> </table>	sno	s1	s2	s3	s4
sno	pno																															
s1	p1																															
s1	p2																															
s1	p3																															
s1	p4																															
s2	p1																															
s2	p2																															
s3	p2																															
s4	p2																															
s4	p4																															
pno																																
p2																																
sno																																
s1																																
s2																																
s3																																
s4																																
		B2	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><th>pno</th></tr> <tr><td>p2</td></tr> <tr><td>p4</td></tr> </table>	pno	p2	p4	A/B2	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><th>sno</th></tr> <tr><td>s1</td></tr> <tr><td>s4</td></tr> </table>	sno	s1	s4																					
pno																																
p2																																
p4																																
sno																																
s1																																
s4																																
		B3	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><th>pno</th></tr> <tr><td>p1</td></tr> <tr><td>p2</td></tr> <tr><td>p4</td></tr> </table>	pno	p1	p2	p4	A/B3	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><th>sno</th></tr> <tr><td>s1</td></tr> </table>	sno	s1																					
pno																																
p1																																
p2																																
p4																																
sno																																
s1																																

Figura 16: Exemplos do uso da divisão

Esta operação é útil para expressar certos tipos de consultas como, por exemplo, “Encontre o nome dos marinheiros que tenham reservado **todos** os barcos.”

2.10 Exemplos adicionais

Esta seção apresenta alguns exemplos adicionais. Os esquemas das relações utilizadas são similares aos das relações já apresentadas anteriormente e podem ser facilmente inferidos do contexto.

- *Encontre os nomes dos marinheiros que reservaram um barco vermelho ou um barco verde.*

$$\boxed{\begin{array}{l} \rho(\text{Tempboats}, (\sigma_{\text{color}='red' \vee \text{color}='green'} \text{Boats})) \\ \pi_{\text{sname}} (\text{Tempboats} \bowtie \text{Reserves} \bowtie \text{Sailors}) \end{array}}$$

Primeiro, são selecionados apenas os barcos que sejam vermelhos ou verdes. Para facilitar legibilidade, a resposta foi nomeada *Tempboats*. A seguir, a relação *Reserves* foi usada para estabelecer a ligação entre os barcos e os marinheiros que os reservaram. Note que ocorrem três junções simultâneas: barcos são ligados às reservas por meio do campo *bid* comum a ambos; ao mesmo tempo, reservas são ligadas aos respectivos marinheiros por meio do campo *sid*. Por fim, são apenas projetados os nomes dos marinheiros da relação resultante destas junções.

- *Encontre os nomes dos marinheiros que reservaram um barco vermelho e um barco verde.*

$$\begin{array}{c} \rho(\text{Tempred}, \pi_{sid}((\sigma_{color='red'} \text{Boats}) \bowtie \text{Reserves})) \\ \rho(\text{Tempgreen}, \pi_{sid}((\sigma_{color='green'} \text{Boats}) \bowtie \text{Reserves})) \\ \pi_{sname}((\text{Tempred} \cap \text{Tempgreen}) \bowtie \text{Sailors}) \end{array}$$

A estratégia, neste exemplo, é obter o código dos marinheiros que reservaram barcos vermelhos, armazenados na relação temporária *Tempred*. A seguir, são obtidos os códigos dos marinheiros que reservaram barcos verdes, armazenados em *Tempgreen*. Para determinar os marinheiros que reservaram barcos verdes e vermelhos, basta fazer a intersecção entre estes dois conjuntos (*Tempgreen* e *Tempred*). Note que esta intersecção produz apenas os códigos destes marinheiros. Assim, é feita a junção do resultado da intersecção com a relação dos marinheiros. Para finalizar, são projetados apenas os nomes dos marinheiros encontrados.

Note que um barco não pode ser simultaneamente verde e vermelho. Por isso, não bastaria fazer uma consulta semelhante ao exemplo anterior, trocando o operador \vee por \wedge .

- *Encontre os sids dos marinheiros com idade superior a 20 anos que não tenham reservado um barco vermelho.*

$$\begin{array}{c} \pi_{sid}(\sigma_{age>20} \text{Sailors}) - \\ \pi_{sid}((\sigma_{color='red'} \text{Boats}) \bowtie \text{Reserves} \bowtie \text{Sailors}) \end{array}$$

O primeiro passo, neste exemplo, é determinar quais são os marinheiros com idade superior a 20 anos. A seguir, são retirados aqueles que reservaram algum barco vermelho.

- *Encontre os nomes dos marinheiros que reservaram todos os barcos da marca Interlake.*

$$\begin{array}{c} \rho(\text{Tempsid}, (\pi_{sid,bid} \text{Reserves}) / (\pi_{bid}(\sigma_{bname='Interlake'} \text{Boats}))) \\ \pi_{sname}(\text{Tempsid} \bowtie \text{Sailors}) \end{array}$$

Num primeiro passo são calculadas duas tabelas: uma contendo os códigos de barcos (*bid*) e os códigos dos marinheiros (*sid*) que os reservaram; outra, contendo apenas os códigos dos barcos da marca *Interlake*. É fácil, agora, aplicar o operador de divisão a estas duas tabelas para obter os códigos dos marinheiros que reservaram todos os barcos da marca desejada. Estes códigos são armazenados na relação temporária *Tempsid*.

A seguir, é feita a junção desta tabela temporária com a tabela de marinheiros a fim de que se obtenha os nomes dos marinheiros.

3 Cálculo Relacional

O cálculo relacional é uma linguagem de consulta declarativa. Por meio dela podemos descrever o conjunto das respostas esperadas sem a necessidade de explicitar como este conjunto é computado. Existem duas variantes de cálculo relacional: o *cálculo relacional de tuplas* (CRT) e o *cálculo relacional de domínio* (CRD).

3.1 Cálculo Relacional de Tuplas

Uma consulta no cálculo relacional de tuplas é da forma:

$$\{T \mid p(T)\}$$

tal que T é uma variável de tupla e $p(T)$ denota uma *fórmula* que descreve T .

Uma *variável de tupla* é uma variável que pode armazenar tuplas de um esquema de relação particular, ou seja, pode armazenar uma linha de uma relação. O resultado da consulta é o conjunto de todas as tuplas t para as quais a fórmula $p(T)$ é verdadeira. Por exemplo, a consulta abaixo encontra todos os marinheiros da relação $S1$ com *rating* superior ao valor 7.

$$\{S \mid S \in S1 \wedge S.rating > 7\}$$

3.1.1 Sintaxe das Consultas

Seja Rel o nome de uma relação, R e S variáveis de tupla, a e b atributos de R e S (respectivamente). Seja op um operador do conjunto $\{<, >, \leq, \geq, \neq, =\}$. Uma *fórmula atômica* é uma dentre:

- $R \in Rel$
- $R.a \ op \ S.b$
- $R.a \ op \ cte$ ou $cte \ op \ R.a$, cte é uma constante

No exemplo anterior, as fórmulas atômicas são:

- $S \in S1$
- $S.rating > 7$

Uma *fórmula* é definida recursivamente pelas regras abaixo, tal que p e q são fórmulas e $p(R)$ denota uma fórmula na qual a variável R é utilizada. Uma fórmula é:

- qualquer fórmula atômica
- $\neg p$, $p \wedge q$, $p \vee q$ ou $p \Rightarrow q$
- $\exists R(p(R))$, R é uma variável de tupla
- $\forall R(p(R))$, R é uma variável de tupla

3.1.2 Semântica das Consultas

A resposta de uma consulta CRT

$$\{T \mid p(T)\}$$

é o conjunto de todas as tuplas t para as quais a fórmula $p(T)$ é verdadeira, quando t é atribuída à variável T . Neste sentido, T é uma *variável livre* que pode ser associada a diferentes tuplas da relação. Por exemplo, considere a relação S1 (Fig. 1) e a consulta

$$\{S \mid S \in S1 \wedge S.rating > 7\}$$

Note que as tuplas de ordem 1,2 ou 4 da relação S1 podem ser atribuídas à variável S, pois elas satisfazem as condições descritas na fórmula ($rating > 7$), resultando na relação mostrada na Fig. 17.

Neste exemplo, é fácil perceber quais são as tuplas da relação que satisfazem a fórmula, entretanto, é necessário definir precisamente quando uma fórmula é verdadeira.

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.0
58	rusty	10	35.0

Figura 17: $\{S \mid S \in S1 \wedge S.rating > 7\}$

Formalmente, uma consulta é avaliada usando uma dada instância de um banco de dados. Cada variável livre numa fórmula F é atribuída a uma tupla.¹ Para cada atribuição de tuplas às variáveis, F é verdadeira se alguma das condições abaixo se verifica:

- F é uma fórmula atômica $R \in Rel$, e uma tupla de uma instância da relação Rel é atribuída à R. Por exemplo, todas as tuplas da relação S1 satisfazem à consulta $\{S \mid S \in S1\}$.
- F é da forma $\neg p$ e p não é verdadeira.
- F é da forma $p \wedge q$ e, ambas, p e q são verdadeiras.
- F é da forma $p \vee q$ e uma das duas é verdadeira.
- F é da forma $p \Rightarrow q$ e q é verdadeira sempre que p o for.
- F é uma comparação $R.a \text{ op } S.b$ ou $R.a \text{ op } cte$ ou $cte \text{ op } R.a$,² e as tuplas atribuídas a R e a S possuem os atributos a e b com valores que satisfazem a comparação. Por exemplo, $\{S \mid S \in S1 \wedge S.age < 40.0\}$.
- F é da forma $\exists R(p(R))$, e existe alguma atribuição de tuplas às variáveis livres em p(R), incluindo a variável R, que faz que fórmula p(R) seja verdadeira.
- F é da forma $\forall R(p(R))$ e existe alguma atribuição de tuplas às variáveis livres em p(R) que faz que a fórmula p(R) seja verdadeira para qualquer tupla que seja atribuída a R. Os quantificadores \exists e \forall vinculam a variável R.

¹O termo original: *bound*.

²*cte* denota uma constante

3.2 Cálculo Relacional de Domínios

Uma consulta do Cálculo Relacional de Domínios (CRD) é da forma:

$$\{\langle x_1, x_2, \dots, x_n \rangle \mid p(x_1, x_2, \dots, x_n)\}$$

tal que cada x_i é uma variável de domínio ou uma constante e $p(x_1, x_2, \dots, x_n)$ denota uma *fórmula* do CRD cujas variáveis livres são algumas das variáveis x_i , $1 \leq i \leq n$.

Uma *fórmula atômica* do CRD é uma dentre as seguintes:

- $\langle x_1, x_2, \dots, x_n \rangle \in Rel$, tal que Rel é uma relação com n atributos; cada x_i , $1 \leq i \leq n$, é uma variável ou uma constante
- $x_r \text{ op } x_s$ (op é um operador relacional)
- $x_i \text{ op } cte$ ou $cte \text{ op } x_i$ (cte é uma constante)

Uma *fórmula* do CRD é definida recursivamente como um dos seguintes:

- qualquer fórmula atômica
- $\neg p$, $p \wedge q$, $p \vee q$, $p \Rightarrow q$
- $\exists x(p(x))$, tal que x é uma variável de domínio
- $\forall x(p(x))$, tal que x é uma variável de domínio

Por exemplo, a consulta abaixo retorna o nome e a idade dos marinheiros com rating superiores a 7.

$$\{\langle sname, age \rangle \mid \langle sid, sname, rating, age \rangle \in S1 \wedge rating > 7\}$$

4 Exemplos

A seguir são apresentados alguns exemplos de consultas na três linguagens apresentadas.

- **Projeção:** Retorne os nomes e as idades dos marinheiros.

- **álgebra:** $\pi_{name, age} S1$
- **CRT:** $\{P \mid \exists S \in S1 (P.name = S.name \wedge P.age = S.age)\}$
- **CRD:** $\{\langle sname, age \rangle \mid \langle sid, sname, rating, age \rangle \in S1\}$

- **Seleção:** Retorne os marinheiros com rating superior a 7.

- **álgebra:** $\sigma_{rating > 7}(S1)$
- **CRT:** $\{S \mid S \in S1 (S.rating > 7)\}$
- **CRD:** $\{\langle sid, sname, rating, age \rangle \mid \langle sid, sname, rating, age \rangle \in S1 \wedge rating > 7\}$

- **Junção:** Retorne os dias em que cada marinheiro possui reserva.

- **álgebra:** $\pi_{sname,day}(S1 \bowtie R1)$
- **CRT:** $\{P | \exists R \in R1 \exists S \in S1 (R.sid = S.sid \wedge P.name = S.sname \wedge P.day = R.day)\}$
- **CRD:** $\{\langle sname, day \rangle | \exists sid, sname (\langle sid, sname, rating, age \rangle \in S1 \wedge \exists day (\langle sid, bid, day \rangle \in R1))\}$

Note que o valor atribuído à variável *sid* de *S1* vincula o valor da variável *sid* de *R1*.

- Retorne o nome e a idade dos marinheiros com rating igual a 7.

- **álgebra:** $\pi_{sname,age}(\sigma_{rating=7}(S1))$
- **CRT:** $\{P | \exists S \in S1 (S.rating = 7 \wedge P.name = S.sname \wedge P.age = S.age)\}$
- **CRD:** $\{\langle sname, age \rangle | \langle sid, sname, 7, age \rangle \in S1\}$

5 Poder de expressão da álgebra e do cálculo

A princípio, o cálculo relacional possui maior expressividade que a álgebra relacional, pois usando-o é possível escrever a seguinte consulta:

$$\{S | \neg(S \in S1)\}$$

que denota todas as tuplas *S* que não estão na relação *S1*. O conjunto destas tuplas é infinito. Note, por exemplo, que a relação *S1* possui o campo *rating* cujo domínio é de valores inteiros e o conjunto dos inteiros é um conjunto infinito. Assim, tal consulta – denominada *insegura* – não pode ser respondida e deve ser proibida.

Pode ser mostrado que toda consulta que pode ser escrita usando uma consulta segura do cálculo relacional, também pode ser escrita como uma consulta da álgebra relacional. Desta forma, podemos concluir que o poder de expressão da álgebra e do cálculo relacional são equivalentes.

Se uma linguagem de consulta permite que sejam construídas todas as consultas que podem ser expressas usando a álgebra relacional, tal linguagem é dita ser *relacionalmente completa*.³

³Termo original: relationally complete.