



A grande maioria das aplicações que desenvolvemos atualmente utilizam um banco de dados relacional o que implica na utilização de consultas para obtenção de resultados.

Para isso usamos a linguagem SQL que é a linguagem de pesquisa declarativa padrão para banco de dados relacional. Muitas das características originais do SQL foram inspiradas na [álgebra relacional](#) e neste artigo eu procuro abordar os conceitos básicos da álgebra relacional.

Curso C# Vídeo Aulas
Do básico ao intermediário

Por um preço justo

A **Álgebra Relacional** é uma linguagem de consulta formal, porém procedimental, ou seja, o usuário dá as instruções ao sistema para que o mesmo realize uma sequência de operações na base de dados para calcular o resultado desejado.

Na terminologia formal de modelo relacional temos os seguintes conceitos:

1. Uma linha é chamada de **tupla**;
2. O cabeçalho da coluna é chamado de **atributo**;
3. Tabela é chamada de **relação**;
4. O tipo de dados que descreve os tipos de valores que podem aparecer em cada coluna é chamado de **domínio**;

A álgebra relacional é uma forma de cálculo sobre **conjuntos ou relações**.

A álgebra relacional recebia pouca atenção até a publicação do modelo relacional de dados de [E.F Codd](#), em 1970. Codd propôs tal álgebra como uma base para linguagens de consulta em banco de dados.

(fonte: http://pt.wikipedia.org/wiki/%C3%81lgebra_relacional - consultado em novembro de 2012)

Neste artigo eu vou me ater às principais operações fundamentais da álgebra relacional.

Há seis operações fundamentais na álgebra relacional:

1. **Seleção**
2. **Projeção**
3. **Produto cartesiano**
4. **União**
5. **Diferença entre conjuntos**
6. **Renomear**

1- Seleção σ : Seleciona tuplas (linhas) que satisfazem um certo predicado ou condição.

Indicada por σ (letra grega sigma), é uma operação que para um conjunto inicial fornecido como argumento, produz um subconjunto estruturalmente idêntico, mas apenas com os elementos do conjunto original que atendem a uma determinada condição (chamada de predicado). A seleção pode ser entendida como uma operação que filtra as linhas de uma relação(tabela), e é uma operação unária, pois opera sobre um único conjunto de dados.

Notação - σ **predicado (relação)**

Alunos

| id | nome | sexo |
|-----|-----------|------|
| 123 | Macoratti | M |
| 234 | Miriam | F |
| 456 | Jefferson | M |
| 567 | Janice | F |

Ex 1: Selecionar tuplas de Alunos cujo nome = Macoratti

σ **nome = 'Macoratti' (Alunos)** \implies produz o conjunto dos elementos de alunos que atendem ao predicado [Nome = 'Macoratti'], ou seja, representa um subconjunto dos alunos para o qual essa condição é avaliada como verdadeira.

| id | nome | sexo |
|----|------|------|
| | | |

123 | Macoratti | M

- Resultado - subconjunto horizontal de uma relação
- Operadores de comparação : =, <, <=, >, >=, ≠
- Operadores lógicos: ^ (and) V (or) ¬ (not)

Ex 2: Selecionar as tuplas de Alunos com id > 123 e id < 567

$\sigma \text{ id} > 123 \wedge \text{id} < 567$

| id | nome | sexo |
|-----|-----------|------|
| 234 | Miriam | F |
| 456 | Jefferson | M |

O operador de seleção é comutativo $\Rightarrow \sigma_{\langle \text{condição1} \rangle} (\sigma_{\langle \text{condição2} \rangle}) = \sigma_{\langle \text{condição2} \rangle} (\sigma_{\langle \text{condição1} \rangle})$

2- Projeção π : Gera novas relações excluindo alguns atributos

Indicada por π (a letra grega pi) produz um conjunto onde há um elemento para cada elemento do conjunto de entrada, sendo que a estrutura dos membros do conjunto resultante é definida nos argumentos da operação. Pode ser entendida como uma operação que filtra as colunas de uma tabela. Por operar sobre apenas um conjunto de entrada é classificada como uma operação unária.

Notação: $\pi \text{ lista_nome_atributos}$ (Relação)

Ex. 1 - projete o atributo nome sobre a relação Alunos

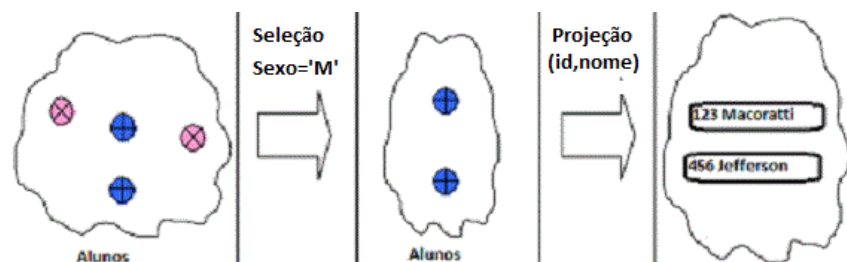
$\pi \text{ nome}$ (Alunos)

| nome |
|-----------|
| Macoratti |
| Miriam |
| Jefferson |
| Janice |

Ex. 2 : Descobrir o nome e o id de todos os alunos do sexo masculino

Neste caso será necessário combinar uma projeção com uma seleção.

Se decidirmos projetar as colunas desejadas diretamente a partir da relação alunos, estaremos considerando também os elementos do sexo feminino o que não queremos. Como a projeção não permite descartar linhas, apenas colunas, deveremos fornecer a essa operação o subconjunto resultante de uma filtragem (seleção) da relação de alunos original, como mostram as figuras abaixo, que representam as relações e as operações de duas maneiras diferentes.



Definindo a expressão que atende aos requisitos temos:

$\pi \text{ id,nome} (\sigma \text{ sexo} = 'M')$ (Alunos)

| id | nome |
|-----|-----------|
| 123 | Macoratti |
| 456 | Jefferson |

O operador Projeção não é comutativo.

A álgebra relacional empresta da teoria de conjuntos quatro operadores: União, Intersecção, Diferença e Produto Cartesiano que veremos a seguir.

3- Produto Cartesiano X : Retorna todas as combinações de tuplas de duas R1 e R2.

O resultado do produto cartesiano de duas relações é uma terceira relação contendo todas as combinações possíveis entre os elementos das relações originais.

Essa relação resultante possuirá um número de colunas que é igual à soma das quantidades de colunas das duas relações iniciais, e um número de linhas igual ao produto do número de suas linhas. Portanto, se fizermos o produto cartesiano de uma relação A que possua 5 colunas e 10 linhas com uma relação B onde existem 3 colunas e 8 linhas, a relação resultante terá **5+3= 8 colunas e 10*8= 80 linhas**.

- **Total de colunas do produto cartesiano : Número colunas da primeira tabela + número de colunas da segunda tabela**
- **Número de linhas do produto cartesiano: Número de linhas da primeira tabela x número de linhas da segunda tabela**

Assim, cada linha dessa relação corresponderá à concatenação de uma linha da primeira relação com uma linha da segunda.

Notação : relação1 X relação2 (R1 x R2)

Ex 1: Descobrir o nome do aluno, sexo e o nome do curso para cada aluno

Alunos

| id | nome | sexo | curso |
|-----|-----------|------|-------|
| 123 | Macoratti | M | 100 |
| 234 | Miriam | F | 110 |
| 456 | Jefferson | M | 120 |
| 567 | Janice | F | 100 |

Cursos

| id | nome |
|-----|------------|
| 100 | Quimica |
| 110 | Inglês |
| 120 | Matemática |
| 130 | Física |

π nome, sexo, curso (σ Alunos.curso = Cursos.id (Alunos x Cursos)

Resultado:

| id | nome | sexo | nome |
|-----|-----------|------|------------|
| 123 | Macoratti | M | Quimica |
| 234 | Miriam | F | Inglês |
| 456 | Jefferson | M | Matemática |
| 567 | Janice | F | Quimica |

Note que primeiro fizemos o produto cartesiano (**Alunos x Cursos**) que resulta em uma relação com **6 colunas e 16 linhas**:

| id | nome | sexo | curso | id | nome |
|-----|-----------|------|-------|-----|------------|
| 123 | Macoratti | M | 100 | 100 | Quimica |
| 123 | Macoratti | M | 100 | 110 | Inglês |
| 123 | Macoratti | M | 100 | 120 | Matematica |
| 123 | Macoratti | M | 100 | 130 | Fisica |
| 234 | Miriam | F | 110 | 100 | Quimica |
| 234 | Miriam | F | 110 | 110 | Inglês |
| 234 | Miriam | F | 110 | 120 | Matematica |
| 234 | Miriam | F | 110 | 130 | Fisica |
| ... | ... | ... | ... | ... | ... |

Depois fizemos uma seleção pelo código do curso : **σ Alunos.curso = Cursos.id**

| id | nome | sexo | curso | id | nome |
|-----|-----------|------|-------|-----|------------|
| 123 | Macoratti | M | 100 | 100 | Quimica |
| 123 | Miriam | F | 110 | 110 | Inglês |
| 123 | Jefferson | M | 120 | 120 | Matematica |
| 123 | Janice | F | 100 | 100 | Quimica |

Em seguida fizemos um projeção de **nome, sexo e curso**:

| nome | sexo | nome |
|-----------|------|---------|
| Macoratti | M | Quimica |
| | | |

| | | |
|-----------|---|------------|
| Miriam | F | Inglês |
| Jefferson | M | Matemática |
| Janice | F | Química |

4- União \cup : Retorna a união das tuplas de duas relações R1 e R2 com eliminação automática de duplicatas;

Produz como resultado uma Relação que contém todas as linhas da primeira Relação seguidas de todas as linhas da segunda tabela. A Relação resultante possui a mesma quantidade de colunas que as relações originais, e tem um número de linhas que é no máximo igual à soma das linhas das relações fornecidas como operandos, já que as linhas que são comuns a ambas as relações aparecem uma única vez no resultado.

Notação: Relação1 \cup Relação2 (R1 \cup R2)

Obs: As relações devem possuir o mesmo número de atributos.

Alunos

Professores

Funcionarios

| id | nome | idade | curso |
|----|-----------|-------|---------|
| 10 | Macoratti | 45 | Química |
| 20 | Miriam | 43 | Artes |
| 30 | Bianca | 21 | Física |

| id | nome | idade | setor |
|-----|--------|-------|---------|
| 100 | Pedro | 50 | Química |
| 200 | Maria | 45 | Física |
| 300 | Bianca | 21 | Artes |

| id | nome | setor | idade |
|----|-----------|---------|-------|
| 10 | Margarida | Química | 46 |
| 20 | Jamil | Física | 32 |

Domínio:

id = int
nome = varchar(30)
idade = int
curso = varchar(30)
setor = varchar(30)

A relação **Alunos** é compatível com **Professores** mas não é compatível com **Funcionarios**.

Ex1: Encontre uma relação com todos os alunos e com todos os professores:

resultado: **Alunos \cup Professores**

| id | nome | idade | curso |
|-----|-----------|-------|---------|
| 10 | Macoratti | 45 | Química |
| 20 | Miriam | 43 | Artes |
| 30 | Bianca | 21 | Física |
| 100 | Pedro | 50 | Química |
| 200 | Maria | 45 | Física |
| 300 | Bianca | 21 | Artes |

A operação de união é comutativa $\Rightarrow R1 \cup R2 = R2 \cup R1$

5- Diferença -- : Retorna as tuplas presentes em R1 e ausentes em R2;

É uma operação que requer como operandos duas relações união-compatíveis, ou seja, estruturalmente idênticas. O resultado é uma relação que possui todas as linhas que existem na primeira relação e não existem na segunda.

Notação : relação1 - relação2 (R1 - R2)

Alunos (R1)

Professores(R2)

| id | nome | idade | curso |
|----|-----------|-------|---------|
| 10 | Macoratti | 45 | Química |
| 20 | Miriam | 43 | Artes |
| 30 | Bianca | 21 | Física |

| id | nome | idade | setor |
|-----|--------|-------|---------|
| 100 | Pedro | 50 | Química |
| 200 | Maria | 45 | Artes |
| 300 | Bianca | 21 | Física |

Domínio:

id = int
nome = varchar(30)
idade = int
curso = varchar(30)
setor = varchar(30)

Ex1 : Apresente uma relação de todos os alunos que não são professores

Resultado : Aluno - Professor

| id | nome | idade | curso |
|----|-----------|-------|---------|
| 10 | Macoratti | 45 | Química |
| 20 | Miriam | 43 | Artes |

Note-se que a DIFERENÇA não é comutativa !

Resultado : Professor - Aluno

| id | nome | idade | setor |
|-----|-------|-------|---------|
| 100 | Pedro | 50 | Química |

200 | Maria | 45 | Artes

6 - Interseção \cap : Retorna as tuplas comuns a R1 e R2;

Esta é uma operação adicional que produz como resultado uma tabela que contém, sem repetições, todos os elementos que são comuns às duas tabelas fornecidas como operandos. As tabelas devem ser união-compativeis.

Notação : relação1 \cap relação2 (R1 \cap R2)

Alunos(R1)

Professores(R2)

| id | nome | idade | curso |
|----|-----------|-------|---------|
| 10 | Macoratti | 45 | Quimica |
| 20 | Miriam | 43 | Artes |
| 30 | Bianca | 21 | Fisica |

| id | nome | idade | setor |
|-----|--------|-------|---------|
| 100 | Pedro | 50 | Quimica |
| 200 | Maria | 45 | Artes |
| 300 | Bianca | 21 | Fisica |

Domínio:

id = int
nome = varchar(30)
idade = int
curso = varchar(30)
setor = varchar(30)

Ex1 : Apresente uma relação de todos os alunos que são professores;

Resultado : Alunos \cap Professores

| id | nome | idade | curso |
|----|--------|-------|--------|
| 30 | Bianca | 21 | Fisica |

Existem operadores de álgebra que são deriváveis de outros. A operação de intersecção é derivável de união e diferença: $A \cap B = A - (A - B)$

A operação de intersecção é comutativa $\Rightarrow R1 \cap R2 = R2 \cap R1$

7 - Junção Natural \bowtie : Retorna a combinação de tuplas de duas relações R1 e R2 que satisfazem um predicado;

O resultado da operação junção natural é uma relação com todas as combinações das tuplas na relação1 (R1) e relação2 (R2) nas quais os seus atributos em comum são iguais.

É uma operação que produz uma combinação entre as linhas de uma relação com as linhas correspondentes de outra relação, sendo em princípio correspondente a uma seleção pelos atributos de relacionamento sobre um produto cartesiano dessas relações:

A operação de junção foi criada porque esse tipo de combinação de tabelas é muito comum, facilitando com isso a escrita de expressões. A tabela resultante de uma junção tem todas as colunas da primeira tabela e todas da segunda tabela.

Notação: R1 \bowtie R2

No exemplo a seguir temos as relações **Empregados** e **Setores** a sua junção natural :

Empregados

Setores

Empregados \bowtie Setores

| id | nome | setor |
|-----|-----------|--------|
| 100 | Macoratti | Admin |
| 200 | Jefferson | Contab |
| 300 | Bianca | Admin |
| 400 | Janice | Contab |

| setor | gerente |
|--------|-----------|
| Admin | Paulino |
| Contab | Amelia |
| RH | Francisca |

\bowtie

| id | nome | setor | gerente |
|-----|-----------|--------|---------|
| 100 | Macoratti | Admin | Paulino |
| 200 | Jefferson | Contab | Amelia |
| 300 | Bianca | Admin | Paulino |
| 400 | Janice | Contab | Amelia |

A junção natural pode ser vista como uma combinação de uma operação de seleção aplicada sobre uma operação de produto cartesiano:

σ

<critério> (<relação1> X <relação2>)

8 - Renomeação ρ : Altera o nome de uma relação e/ou dos seus atributos

Esta operação unária primitiva redefine o nome de uma tabela em um determinado contexto. É útil para auto-relacionamentos, onde precisamos fazer a junção de uma tabela com ela mesma, e nesse caso cada versão da tabela precisa receber um nome diferente da outra.

Notação: ρ <novo nome> (R)

Ex1: ρ <empregados> (funcionarios)

Renomeia a relação funcionarios parra empregados.

9 - Divisão \div : É uma operação adicional que produz como resultado a projeção de todos os elementos da primeira relação que se relacionam com todos os elementos da segunda relação.

Divisão é uma operação da álgebra relacional utilizada quando se deseja extrair de uma relação R1 uma determinada parte que possui as características (valores de atributos) da relação R2.

Notação: $R1 \div R2$

Ex1: Dada as relações:

Equipes

Projetos

| id | NomeProjeto | NomeProjeto | descricao |
|-----|-------------|-------------|-----------------|
| 100 | Projeto1 | Projeto1 | Suporte |
| 200 | Projeto2 | Projeto2 | Desenvolvimento |
| 300 | Projeto3 | Projeto3 | Manutenção |
| 400 | Projeto4 | | |

Resultado:

Equipes \div (π <nome_projeto> (Projetos))

| id | NomeProjeto | | NomeProjeto | | id |
|-----|-------------|--------|-------------|---------------|-----|
| 100 | Projeto1 | \div | Projeto1 | \Rightarrow | 100 |
| 200 | Projeto2 | | Projeto2 | | |
| 300 | Projeto3 | | Projeto3 | | |
| 400 | Projeto4 | | | | |

10 - Atribuição \leftarrow : Permite que o conteúdo de uma relação seja atribuído (colocado) em uma variável especial, oferecendo a possibilidade de um tratamento até certo ponto algorítmico para algumas seqüências de operações.

Atribui-se a relação resultante de uma operação à direita de \leftarrow , a uma variável temporária, à esquerda, a qual poderá ser utilizada em relações subseqüentes.

Notação: **variável \leftarrow operação**

Ex1: Resultado \leftarrow Equipes \div (π <nome_projeto> (Projetos))

A seguir temos uma tabela com um resumo das operações vistas neste artigo:

| <i>Símbolo</i> | <i>Operação</i> | <i>Sintaxe</i> | <i>Tipo</i> |
|----------------|------------------------|-------------------------------|-------------|
| σ | Seleção / Restrição | σ condição (Relação) | Primitiva |
| π | Projeção | π expressões (Relação) | Primitiva |
| \cup | União | Relação1 \cup Relação2 | Primitiva |
| \cap | Intersecção | Relação1 \cap Relação2 | Adicional |
| - | Diferença de conjuntos | Relação1 - Relação2 | Primitiva |
| x | Produto cartesiano | Relação1 x Relação2 | Primitiva |
| x | Junção | Relação1 x Relação2 | Adicional |
| \div | Divisão | Relação1 \div Relação2 | Adicional |
| ρ | Renomeação | ρ nome (Relação) | Primitiva |
| \leftarrow | Atribuição | variável \leftarrow Relação | Adicional |

Rom 10:9 Porque, se com a tua boca confessares a Jesus como Senhor, e em teu coração creres que Deus o ressuscitou dentre os mortos, será salvo;

Rom 10:10 pois é com o coração que se crê para a justiça, e com a boca se faz confissão para a salvação.

Rom 10:11 Porque a Escritura diz: Ninguém que nele crê será confundido.

Rom 10:12 Porquanto não há distinção entre judeu e grego; porque o mesmo Senhor o é de todos, rico para com todos os que o invocam.

Rom 10:13 Porque: Todo aquele que invocar o nome do Senhor será salvo.

Referências:

- [Seção VB .NET do Site Macoratti.net](#)
- [Super DVD .NET - A sua porta de entrada na plataforma .NET](#)
- [Super DVD Vídeo Aulas - Vídeo Aula sobre VB .NET, ASP .NET e C#](#)
- [Seção C# do site Macoratti.net](#)
- [Bases de Dados Parte IV Álgebra e Cálculo Relacional](#)
- [Álgebra relacional – Wikipédia, a enciclopédia livre](#)
- [Modelo Relacional - Manipulação Álgebra ... - UDESC Joinville](#)

[José Carlos Macoratti](#)