

1. Introdução :

A álgebra relacional é uma linguagem de consultas formal, porém procedural. Consiste em um conjunto de operações tendo como operando uma ou duas relações e produzindo, como resultado, uma nova relação. Assim sendo a relação resultante pode tornar-se um operando de uma outra operação.

As operações fundamentais na álgebra relacional são seleção, projeção, união, diferença e produto cartesiano. Além dessas operações fundamentais existem algumas outras operações – renomeação, interseção, junção natural, divisão, e atribuição, que são definidas em termos das operações fundamentais.

As operações seleção, projeção e renomeação são chamadas de operações unárias, pois operam sobre uma única relação. As outras três operações fundamentais operam um par de relações e são portanto chamadas de operações binárias. As operações de união, diferença e interseção devem ser compatíveis de união.

Tanto os operandos como o resultado das operações da álgebra relacional são uma relação, portanto o resultado das operações está suscetível a ser aplicado a qualquer operação da álgebra;

A álgebra relacional foi o meio encontrado para provar matematicamente que o modelo relacional funcionava com perfeição.

2. Operações:

Dada as seguintes tabelas, teremos as possíveis operações:

PECAS

Codpeca	Descr	Qtde	Codfor
A	Aaa	30	1
B	Bbb	40	1
C	Ccc	50	2

FORNECEDORES

CodFor	Nome	Tel	Uf
1	SHELL	721	RJ
2	ESSO	722	SP
3	IPIRANGA	723	SP

1.1. Projeção (π)

A operação projeção é usada para obter determinados atributos de uma relação. A operação é executada em apenas uma relação e o resultado é uma nova relação contendo apenas os atributos especificados, eliminando-se as eventuais duplicidades entre as tuplas. É denotada pela letra grega π e são listados, subscritos a letra π , os atributos exibidos no resultado.

O argumento da relação vem a seguir entre parênteses.

π descr, qtde(pecas)

R

Descr	Qtde
Aaa	30
Bbb	40
Ccc	50

1.2. Seleção (δ)

A operação seleção retorna um subconjunto de tuplas que satisfaçam um determinado predicado, usando a letra grega sigma (σ) para denotar seleção. O predicado aparece subscrito à letra σ . O argumento da relação é dado entre parênteses, seguindo o predicado. Para especificar o predicado, pode-se usar operadores relacionais ($>$, $>=$, $<$, $<=$, $=$) e conectivos lógicos (\wedge , \vee , \neg).

Exemplo:

π nome (δ uf='SP'(fornecedores))

Nome
ESSO
IPIRANGA

1.3. Produto Cartesiano (\times)

A operação produto cartesiano permite combinar informações de duas relações quaisquer. No resultado para cada tupla da primeira relação haverá a combinação com todas as tuplas da segunda relação ou vice-versa.

ex: Pecas \times fornecedores \Leftrightarrow select * from pecas, fornecedores

PECAS

Codpeca	Descr	Qtde	Codfor	Codfor	Nome	Tel	Uf
A	Aaa	30	1	1	SHELL	721	RJ
A	Aaa	30	1	2	ESSO	722	SP
A	Aaa	30	1	3	IPIRANGA	723	SP
B	Bbb	40	1	1	SHELL	721	RJ
B	Bbb	40	1	2	ESSO	722	SP
B	Bbb	40	1	3	IPIRANGA	723	SP
C	Ccc	50	2	1	SHELL	721	RJ
C	Ccc	50	2	2	ESSO	722	SP
C	Ccc	50	2	3	IPIRANGA	723	SP

Mostre a descrição das peças e o nome do fornecedor que a fornece:

1.4. Junção ($\triangleright\triangleleft$)

A operação de junção é utilizada para combinar tuplas de duas relações que atendam o predicado de junção especificado partindo dos atributos comuns a ambas. O resultado conterá os atributos das duas relações que estão participando da junção. A operação é de extrema importância em bancos de dados relacionais, pois é através dela que é permitida a realização de relacionamentos

A operação junção natural é uma operação binária que permite combinar as operações de seleção e produto cartesiano dentro de uma única operação. A operação de junção natural gera como resultado uma relação composta por todos os atributos da primeira relação mais os atributos da segunda relação, exceto aqueles usados na cláusula de junção. A cláusula de junção é determinada implicitamente pela igualdade dos atributos comuns das duas relações. O resultado apresentado não repete as colunas em comum, ou seja, apresenta todas as colunas da primeira relação mais as colunas da Segunda relação não usadas na junção.

A operação de junção a união dos dados de uma tabela com o de outras tabelas respeitando a igualdade das chaves primárias e estrangeiras de cada uma.

ex:

π pecas.desc, forn.nome (δ pecas.codfor=forn.codfor(pecas x forn))
 \leftrightarrow (é equivalente a operação abaixo)
 π pecas.desc, forn.nome (pecas $\triangleright\triangleleft$ fornecedores)

Variações do Join

$\triangleright\triangleleft$	inner join
$] \triangleright\triangleleft$	left join
$\triangleright\triangleleft [$	right join
$] \triangleright\triangleleft [$	full join

Alunos			Cursos	
Nome	Curso		Curso	Descr
JOAO	DIR		ADM	ADMINISTRACAO
MARIA	-		DIR	DIREITO
PEDRO	INF		INF	INFORMATICA

Apostila de Álgebra Relacional

Alunos x cursos

Alunos	Cursos		
Nome	Curso	Curso	Descr
JOAO	DIR	ADM	ADMINISTRACAO
JOAO	DIR	DIR	DIREITO
JOAO	DIR	INF	INFORMATICA
MARIA	-	ADM	ADMINISTRACAO
MARIA	-	DIR	DIREITO
MARIA	-	INF	INFORMATICA
PEDRO	INF	ADM	ADMINISTRACAO
PEDRO	INF	DIR	DIREITO
PEDRO	INF	INF	INFORMATICA

Em SQL

Select *

From alunos _____ join cursos on (aluno.curso=cursos.curso)

Left
Right
Full

ORACLE

Select *

From alunos, cursos

Where alunos.curso (+)= cursos.curso(+)

...

1.5. União (\cup)

Operação União

A operação de união cria uma relação partindo de outras duas, levando as tuplas comuns e não-comuns a ambas. Desta forma, aparecerão no resultado somente linhas únicas de uma ou outra relação e as informações duplicadas aparecerão somente uma vez. A operação somente é possível caso as relações de origem possuam compatibilidade de união, ou seja, possuam o mesmo grau e atributos correspondentes definidos num domínio comum.

A união entre duas consultas em álgebra relacional só é possível se elas possuírem os mesmos atributos e o resultado desta união será correspondente as tuplas da primeira consulta adicionada as tuplas da segunda.

Exemplo:

π descr, qtde (δ codfor=2 (pecas))

\cup

π descr, qtde (δ qtde<35 (pecas))

suponha a tabela abaixo:

alunos

nome	nota	status
JOAO	8	APR
PEDRO	4	REP
MARIA	6	APR

```
Select nome,nota,'APR' as status  
From alunos  
Where nota >= 6
```

Union

```
Select nome,nota,'REP' as status  
From alunos  
Where nota < 6
```

π nome, nota, status \leftarrow 'APR' (δ nota \geq 6(alunos))

\cup

π nome, nota, status \leftarrow 'REP' (δ nota<6(alunos))

1.6. Interseção (\cap) (em sql é intersect)

A relação criada pela operação interseção será o resultado de todas as tuplas que pertençam a ambas as relações presentes na operação. A operação de interseção usa o símbolo \cap entre os nomes das relações envolvidas. Ela é comutativa, isto é, $r \cap s$ é equivalente a $s \cap r$. Porém, não é uma operação fundamental e não imprime maior poder à álgebra relacional. Simplesmente é mais conveniente escrever $r \cap s$ que $r - (r - s)$. Exige que as relações operandos sejam união-compatíveis, ou seja a interseção entre duas consultas só se dará se elas possuírem os mesmos atributos e terá como resultado as tuplas que fizerem parte de ambas simultaneamente.

Deverão manter o mesmo domínio.

Alunos_inf		Alunos_dir
Nome		Nome_aluno
Dt_nasc		Dt_nasc

π nome, dt_nasc (alunos_inf)
 \cap
 π nome \leftarrow nome_aluno, dt_nasc (alunos_dir)

1.7. Subtração (-) (em sql é minus)

A operação diferença entre conjuntos, denotada por -, permite encontrar as tuplas que estão no primeiro operando, mas não no segundo. A operação de diferença não é comutativa. Ela exige que as relações operandos sejam união- compatíveis.

A subtração assim como as operações anteriores nos obriga que tenhamos os mesmos atributos nas consultas envolvidas. O seu resultado é correspondente as tuplas da primeira que não foram encontradas na segunda consulta.

Exemplo:

Mostrar o código das peças que não são fornecidas por fornecedores de São Paulo.

π codpeca (pecas)
-
 π pecas.codpeca (δ uf = 'SP' (pecas \triangleright \triangleleft fornecedores))

Alunos	Notas	Disciplinas
Matric	Ano	Disc
Nome	Semestre	Descrição
	Matric	
	Disc	
	Nota	

Utilizando o Minus

```
Select matricula
From alunos
Minus
Select matricula
From notas
```

Where ano=2001

π matricula (alunos)

-

π matricula (δ matricula (δ ano = 2001 (notas))

Select matricula

From alunos

Where matricula not exist (select matricula from notas where ano = 2001)

Select nome

From alunos

Minus

Select nome

From notas, alunos

Where ano=2001 and alunos.matric=notas.matric

π matricula (alunos)

-

π matricula (δ matricula (δ ano = 2001 (notas $\triangleright\triangleleft$ alunos))

1.8. *Pertence (\in) IN*

```
SELECT MATRICULA
FROM alunos
Where matricula NOT IN
(Select matricula
from notas
where ano=2001)
```

```
select matricula
from alunos
where NOT EXIST (
select matricula from where ano=2001 and alunos.matricula=notas.matricula)
```

a vantagem do pertence com relação ao minus é que os campos das tabelas deverão se idênticos, enquanto que no pertence basta eu colocar os relacionamentos e indicar quais campos eu preciso.

π matricula (δ matricula \notin (π matricula (δ ano = 2001(notas))) (alunos))

1.9. atribuição (\leftarrow)

Utilidades da função renomear:

a) Quebra consultas complexas:

alunos2001 \leftarrow π matricula (δ ano = 2001 (notas))

π nome (δ matricula \notin alunos_2001 (alunos))

b) Inserir dados em tabelas

Insert into alunos values (5,'SILVIA')

ALUNOS \leftarrow ALUNOS \cup {(5,'SILVIA')}

c) Alterar dados em tabelas

UPDATE ALUNOS SET NOME = 'SILVIA ANDREA' WHERE MAT = 5;

ALUNOS \leftarrow π mat, nome \leftarrow 'SILVIA ANDREA' (δ mat = 5 (alunos))

\cup
 π mat, nome (δ mat \neq 5 (alunos))

d) Apagar o aluno 5

ALUNOS \leftarrow ALUNOS - δ mat = 5 (alunos)

1.10. Renomeação (ρ)

A operação renomeação, representada pela letra minúscula grega rho (ρ), permite renomear uma relação com outro nome, permitindo desta forma o uso desta como primeiro operando (nome original) e segundo operando (nome alternativo) de uma operação binária, ou viceversa. Desta forma, o resultado apresentará o nome qualificado do primeiro e segundo operandos. Permite também renomear atributos.

Apostila de Álgebra Relacional

π p.descr, f.nome (δ p.codfor = f.codfor (ρ p (pecas) x ρ f (fornecedores))

```
select p.descr, f.nome
from pecas p, fornecedores f
where p.codfor=f.codfor
```

mostrar os funcionários que trabalham no mesmo setor do João.
mostrar os funcionários que ganham mais que Maria.

28/08/2002

Funcionários		Participam		Projetos
Mat		Projeto		Projeto
Nome		Mat		Descrição
Tel		Ano		
Salário		N_horas		
Chefe				

Mostrar os funcionários que ganham mais que o 'João'

Em SQL

```
Select f2.nome
From funcionários f1, funcionários f2
Where f1.nome='Joao' and f2.salário>f1.salario
```

π f2.nome (δ f1.nome = 'Joao' ^ f2.salário>f1.salário (ρ f1 (funcionarios) x ρ f2 (funcionarios))

Mostrar os funcionários que participam dos mesmos projetos de 'Maria'

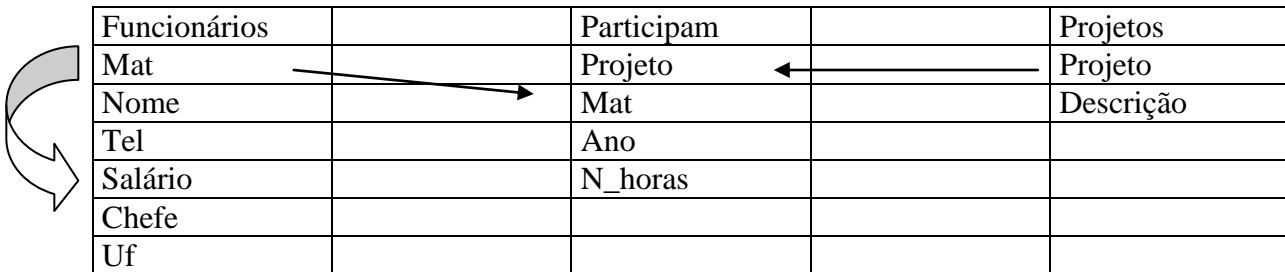
```
Select f2.nome
From funcionários f1, participam p1, participam p2, funcionários f2
Where f1.nome='Maria'
And p1.mat=f1.mat
And f1.nome='Maria'
And p2.projeto=p1.projeto
And p2.mat=f2.mat
```

$\pi_{f2.nome} (\delta_{f1.nome = 'Joao' \wedge f2.salário > f1.salário} (\rho_{f1}(\text{funcionarios}) \times \rho_{f2}(\text{funcionarios})))$

Obs.: os relacionamentos de primária para estrangeira posso fazer join no caso do exemplo temos aqui: $p1.mat=f1.mat$ e aqui: $p2.mat=f2.mat$ relacionamento chave primária para chave estrangeira.

$\pi_{f2.nome} (\delta_{f1.nome = 'Maria' \wedge p2.projeto=p1.projeto} (\rho_{f1}(\text{funcionarios}) \bowtie \rho_{p1}(\text{participam}) \times \rho_{f2}(\text{funcionarios}) \bowtie \rho_{p2}(\text{participam})))$

1.11. Funções de Grupo (Min, Max, Avg, Count, Sum)



Funcionários		Participam		Projetos
Mat		Projeto		Projeto
Nome		Mat		Descrição
Tel		Ano		
Salário		N_horas		
Chefe				
Uf				

Mostrar total de Salarios

```
Select Sum(salário)
From funcionarios
```

$\pi_{G} \text{ Sum(salário)} (\text{funcionários})$

Mostrar o total e a media dos salários por estado

```
Select uf, sum(salário), avg(salário)
From funcionarios
Group by uf
```

$\pi_{uf} \text{ G sum(salário), avg(salário)} (\text{funcionários})$

Mostrar os estados cujo total de salários seja maior que 10000

```
Select uf, sum(salário)
From funcionarios
Group by uf
Having sum(salário)>10000
```

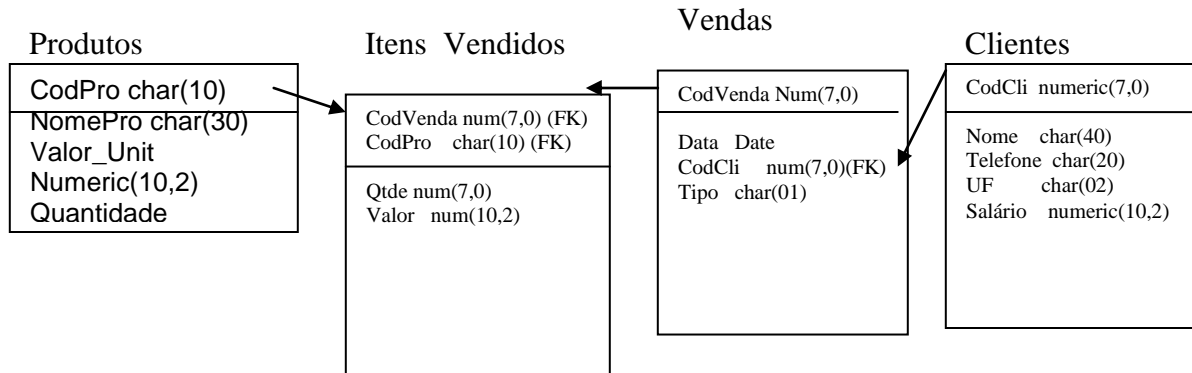
$\delta \text{ sum(salario) > 10000 } (\pi \text{ uf G sum(salário) (funcionários)})$

1.12. Operação Divisão

Dadas duas relações R e S, com grau m e n respectivamente, esta operação apresentará como resultado uma relação com grau (m-n).

A operação de divisão, simbolizada por \div , é usada nas consultas nas quais se exige que todos os valores dos atributos do divisor estejam presentes no dividendo. Produz como resultado a projeção de todos os atributos da primeira relação exceto aqueles presentes no divisor. Não é um operação básica da álgebra.

Exercícios Consultas



- 1) Mostrar o nome e o preço de todos os produtos.
- 2) Mostrar nome, uf e o salário de todos os clientes.
- 3) Mostrar todos os campos da tabela de produtos ordenados em ordem alfabética.
- 4) Mostrar o nome do produto e o seu inventário=(preço* estoque) ordenados em ordem do maior para o menor inventário.
- 5) Mostrar o nome dos clientes, o código da venda que ele realizou e a data ordenados por clientes.
- 6) Mostrar o código, o nome e o estoque dos produtos que custam menos que 1 real.
- 7) Mostrar o nome dos produtos vendidos a clientes de São Paulo obs: eliminar colunas repetidas se houver.
- 8) Mostrar o código dos produtos vendidos no dia 15/05/2000.
- 9) Mostrar o nome dos clientes que realizaram compras à vista e o nome dos produtos comprados nestas compras.
- 10) Mostrar o nome e o telefone dos clientes que compraram arroz.
- 11) Mostrar os produtos que contenham as letras rr no seus nomes.
- 12) Mostrar o nome convertido para máscula de todos os produtos.
- 13) Mostrar a média de salário dos cariocas.
- 14) Mostrar o total de salário por estado.
- 15) Mostrar o código do cliente e a quantidade de vendas que cada um deles realizou.
- 16) Mostrar o nome do produto e o total de quantidades vendidas daquele produto.

- 17) Mostrar o código dos produtos e a quantidade total vendida em ordem decrescente do total de quantidade. Devem ser exibidos apenas os produtos cujo total de quantidades seja superior a 10.
- 18) Mostrar o nome dos clientes e o total em reais de compras que cada um realizou (total= qtde*valor).
- 19) Fazer uma listagem de produtos mais vendidos (Código, Descrição, Total Vendas) ordenado decrescentemente pelo total de vendas.
- 20) Listar o nome dos produtos que nunca foram vendidos
- 21) Inserir o produto 21, 'Refrigerante', Estoque=10, Preço R\$1.90
- 22) Apagar os clientes da Bahia
- 23) Aumentar o preço dos produtos em 10%
- 24) Alterar o salário do cliente 3 para R\$ 1300,00
- 25) Apagar o cliente 5