



UAlg FCT

UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Bases de Dados

ÁLGEBRA RELACIONAL

Álgebra Relacional

Uma linguagem procedimental que consiste num conjunto de operações que recebem uma ou duas relações como entrada e produzem uma nova relação como resultado.

Seis operadores básicos

- seleção: σ (*Sigma*)
- projeção: Π (*Pi*)
- união: \cup
- diferença: $-$
- Produto cartesiano: \times
- renomear: ρ (*Ró*)

Operador de Seleção

A operação **select (seleção)** seleciona tuplos que satisfazem um determinado predicado.

Notação: $\sigma_p(r)$

p é chamado de **predicado de seleção**

Exemplo: selecione os tuplos da relação do *professor* nos quais o professor pertence ao departamento de “Física”.

- Query

$\sigma_{dept_name = \text{“Física”}}(professor)$

- Resultado

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
33456	Gold	Physics	87000

Operador de Selecção (continuação)

Permitimos comparações usando

$=, \neq, >, \geq, <, \leq$

no predicado de seleção.

Podemos combinar vários predicados num predicado maior usando:

\wedge (**and**), \vee (**or**), \neg (**not**)

Exemplo: Encontre os instrutores de Física com um salário maior que 90.000, escrevemos:

$\sigma_{dept_name = \text{"Física"} \wedge salario > 90.000} (instrutor)$

O predicado **select** pode incluir comparações entre dois atributos.

- Exemplo: encontre todos os departamentos cujo nome é o mesmo que o nome do edifício:
- $\sigma_{nome_departamento = edificio} (departamento)$

Operador de Projeção

Uma operação unária que retorna a relação do argumento, apenas com um conjunto de atributos.

Notação:

$$\Pi_{A_1, A_2, A_3 \dots A_k} (r)$$

onde A_1, A_2, \dots, A_k são nomes de atributos e r é um nome de relação.

O resultado é definido como a relação de k colunas obtidas apagando as colunas que não estão listadas

Linhas duplicadas removidas do resultado, pois as relações são conjuntos

Exemplo de operação do projeção

Exemplo: elimine o atributo *dept_name* do *instrutor*

Consulta :

$\Pi_{ID, nome, salário} (instrutor)$

Resultado:

<i>ID</i>	<i>name</i>	<i>salary</i>
10101	Srinivasan	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

Composição de Operadores

O resultado de uma operação de álgebra relacional é uma relação e, portanto, as operações de álgebra relacional podem ser compostas numa **expressão de álgebra relacional** .

Considere a consulta -- Encontre os nomes de todos os instrutores do departamento de Física.

$$\Pi_{nome} (\sigma_{nome_departamento = "Física" } (instrutor))$$

Em vez de fornecer o nome de uma relação como argumento da operação de projeção, fornecemos uma expressão que é avaliada como uma relação.

Cuidado com a utilização dos parênteses.

Operação de produto cartesiano

A operação do produto cartesiano (denotada por \times) permite combinar informações de quaisquer duas relações.

Exemplo: o produto cartesiano das relações *instructor* e *teaches* é escrito como:

instructor \times *teaches*

Construímos um tuplo de resultado para cada par possível de tuplos: um da relação do *instructor* e um da relação *teaches* (ver o próximo slide)

Como o *ID* do *instructor* aparece em ambas as relações, distinguimos entre esses atributos anexando ao atributo o nome da relação da qual o atributo veio originalmente.

- *instructor.ID*
- *teaches.ID*

Tabela (relação) instrutor X teaches

instructor.ID	name	dept_name	salary	teaches.ID	course_id	sec_id	semester	year
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	12121	FIN-201	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	15151	MU-199	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	22222	PHY-101	1	Fall	2017
...
...
12121	Wu	Finance	90000	10101	CS-101	1	Fall	2017
12121	Wu	Finance	90000	10101	CS-315	1	Spring	2018
12121	Wu	Finance	90000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
12121	Wu	Finance	90000	15151	MU-199	1	Spring	2018
12121	Wu	Finance	90000	22222	PHY-101	1	Fall	2017
...
...
15151	Mozart	Music	40000	10101	CS-101	1	Fall	2017
15151	Mozart	Music	40000	10101	CS-315	1	Spring	2018
15151	Mozart	Music	40000	10101	CS-347	1	Fall	2017
15151	Mozart	Music	40000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
15151	Mozart	Music	40000	22222	PHY-101	1	Fall	2017
...
...
22222	Einstein	Physics	95000	10101	CS-101	1	Fall	2017
22222	Einstein	Physics	95000	10101	CS-315	1	Spring	2018
22222	Einstein	Physics	95000	10101	CS-347	1	Fall	2017
22222	Einstein	Physics	95000	12121	FIN-201	1	Spring	2018
22222	Einstein	Physics	95000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
...
...

Operação de junção (*join*)

O produto cartesiano

instrutor X teaches

associa cada tuplo de *instrutor* com cada tuplo de *teaches*.

- A maioria das linhas resultantes contém informações sobre professores que NÃO ministraram uma disciplina específica.

Para obter apenas os tuplos de “*instrutor X teaches*” que pertencem aos instrutores e às disciplinas que eles ministraram, escrevemos:

$\sigma_{instrutor.id = teaches.id} (instrutor \times teaches)$

- Obtemos apenas os tuplos de “*instrutor X teaches*” que pertencem aos instrutores e às disciplinas que eles ministraram.

O resultado desta expressão, é mostrado no próximo slide

Operação de junção – *join* (continuação)

A tabela correspondente a:

$$\sigma_{instructor.id = ensina.id} (instructor \times teaches)$$

<i>instructor.ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>	<i>teaches.ID</i>	<i>course_id</i>	<i>sec_id</i>	<i>semester</i>	<i>year</i>
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
32343	El Said	History	60000	32343	HIS-351	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-101	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-319	1	Spring	2018
76766	Crick	Biology	72000	76766	BIO-101	1	Summer	2017
76766	Crick	Biology	72000	76766	BIO-301	1	Summer	2018
83821	Brandt	Comp. Sci.	92000	83821	CS-190	1	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-190	2	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-319	2	Spring	2018
98345	Kim	Elec. Eng.	80000	98345	EE-181	1	Spring	2017

Operador join

A operação ilustrada no slide anterior corresponde a um Operador adicional (não dá mais capacidade à álgebra relacional), que será apresentado mais à frente.

Operação União

A operação de união permite-nos combinar duas relações

Notação: $r \cup s$

Para $r \cup s$ ser válido.

1. r, s devem ter a *mesma aridade* (mesmo número de atributos)
2. Os domínios de atributos devem ser **compatíveis** (exemplo: 2ª coluna de r com o mesmo tipo de valores que a 2ª coluna de s)

Exemplo: para encontrar todos os cursos ministrados no semestre do outono de 2017, ou no semestre da primavera de 2018, ou em ambos

$$\Pi_{course_id} (\sigma_{semestre= "Outono" \wedge ano=2017} (section)) \cup \\ \Pi_{course_id} (\sigma_{semestre= "Primavera" \wedge ano=2018} (section))$$

Operação União (Cont.)

Resultado de:

$$\Pi_{course_id} (\sigma_{semestre = "Outono" \wedge ano = 2017} (section)) \cup \\ \Pi_{course_id} (\sigma_{semestre = "Primavera" \wedge ano = 2018} (section))$$

<i>course_id</i>
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

Operação de intersecção de conjuntos

A operação de intersecção de conjuntos permite encontrar tuplos que estão em ambas as relações de entrada.

Notação: $r \cap s$

Assumir:

- r, s têm a *mesma aridade*
- atributos de r e s são compatíveis

Exemplo: encontre o conjunto de todos os cursos ministrados nos semestres do outono de 2017 e da primavera de 2018.

$$\prod_{course_id} (\sigma_{semestre = "Outono" \wedge ano = 2017} (seção)) \cap \prod_{course_id} (\sigma_{semestre = "Primavera" \wedge ano = 2018} (seção))$$

- Resultado

<i>course_id</i>
CS-101

Operação de diferença de conjuntos

A operação de diferença de conjuntos permite encontrar tuplos que estão numa relação, mas não estão noutra.

Notação $r - s$

As diferenças definidas devem ser consideradas entre relações **compatíveis**.

- r e s devem ter a **mesma** aridade
- os domínios de atributos de r e s devem ser compatíveis

Exemplo: para encontrar todos os cursos ministrados no semestre do outono de 2017, mas não no semestre da primavera de 2018

$$\Pi_{course_id} (\sigma_{semestre = "Outono" \wedge ano = 2017} (seção)) - \\ \Pi_{course_id} (\sigma_{semestre = "Primavera" \wedge ano = 2018} (seção))$$

<i>course_id</i>
CS-347
PHY-101

A operação de renomeação

Os resultados das expressões de álgebra relacional não têm um nome que possamos usar para nos referirmos a eles. O operador de renomeação, ρ , é fornecido para esse propósito

A expressão:

$$\rho_x (E)$$

retorna o resultado da expressão E sob o nome x

Outra forma da operação de renomeação:

$$\rho_{x(A1,A2, ..An)} (E)$$

Expressões Equivalentes

Há mais de uma forma de escrever uma consulta em álgebra relacional.

Exemplo: Encontre informações sobre cursos ministrados por instrutores do departamento de Física com salário superior a 90.000

Consulta 1

$$\sigma_{dept_name = \text{"Física"} \wedge salário > 90.000} (instrutor)$$

Consulta 2

$$\sigma_{dept_name = \text{"Física"}} (\sigma_{salário > 90.000} (instrutor))$$

As duas consultas não são idênticas; no entanto, são equivalentes: elas fornecem o mesmo resultado em qualquer base de dados.

Exemplo

agencia (agencia-nome, agencia-cidade, orcamento)

cliente (cliente-nome, cliente-rua)

conta (conta-numero, agencia-nome, saldo)

emprestimo (emprestimo-numero, agencia-nome, valor)

titular (cliente-nome, conta-numero)

titular-emprestimo (cliente-nome, emprestimo-numero)

Exemplos de Queries

Empréstimos com valor superior a 1200€

$$\sigma_{valor > 1200} (emprestimo)$$

Números dos empréstimos com valor superior a 1200€

$$\Pi_{emprestimo-numero} (\sigma_{valor > 1200} (emprestimo))$$

Exemplos de Queries

Nomes dos clientes que têm um empréstimo, uma conta, ou ambos.

$$\Pi_{\text{cliente-nome}}(\text{titular-emprestimo}) \cup \Pi_{\text{cliente-nome}}(\text{titular})$$

Nomes dos clientes que têm um empréstimo e uma conta

$$\Pi_{\text{cliente-nome}}(\text{titular-emprestimo}) \cap \Pi_{\text{cliente-nome}}(\text{titular})$$

Exemplos de Queries

Nome de todos os clientes que têm um empréstimo na agência de nome “Faro 2”

$$\Pi_{\text{cliente-nome}} (\sigma_{\text{agencia-nome}=\text{“Faro 2”}} \\ (\sigma_{\text{titular-emprestimo.emprestimo-numero}=\text{emprestimo.emprestimo-numero}} \\ (\text{titular-emprestimo} \times \text{emprestimo})))$$

Nomes de todos os clientes que têm um empréstimo na agência de “Faro 2”, e não têm nenhuma conta numa agência do banco

$$\Pi_{\text{cliente-nome}} (\sigma_{\text{agencia-nome}=\text{“Faro 2”}} \\ (\sigma_{\text{titular-emprestimo.emprestimo-numero}=\text{emprestimo.emprestimo-numero}} \\ (\text{titular-emprestimo} \times \text{emprestimo}))) - \Pi_{\text{cliente-nome}} (\text{titular})$$

Exemplos de Queries

Qual é o maior saldo numa conta?

$$\Pi_{saldo(conta)} - \Pi_{conta.saldo} \\ (\sigma_{conta.saldo < d.saldo} (conta \times \rho_d(conta)))$$