

Databases

SQL - Junções

Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2023/2024





Outline

- Produto cartesiano
- Junção interna
- Junção externa
- Junção com a própria tabela

Operadores relacionais (breve introdução)



Operadores relacionais

Operadores unários

- Restrição
- Projeção

Operadores binários

- União
- Intersecção
- Diferença
- Divisão
- Prod. cartesiano
- Junção

Produto cartesiano

O produto cartesiano de duas tabelas R1 e R2 é uma nova tabela R3 na qual cada registo de R1 é adicionado a todos os registos de R2

Cliente

Nº cliente	Endereço	Telefone
1022	Dias da Silva, 159	711228
1025	Av. Brasil, 21, 1º	722119

Factura

ID Cliente	Factura
1022	101
1025	321

SELECT *

FROM Cliente, Factura

Produto1

Nº cliente	Endereço	Telefone	ID Cliente	Factura
1022	Dias da Silva, 159	711228	1022	101
1022	Dias da Silva, 159	711228	1025	321
1025	Av. Brasil, 21, 1º	722119	1022	101
1025	Av. Brasil, 21, 1º	722119	1025	321

O resultado do produto cartesiano é de utilidade muito limitada uma vez que os registos são associados sem qualquer critério

É **extremamente importante** entender o funcionamento/necessidade dos joins!

Junção (equi-junção)

Exemplo:

```
SELECT emp.nome, funcao, dep.nome  
FROM emp, dep  
WHERE emp.ndep = dep.ndep;
```

```
SELECT emp.nome, funcao, dep.nome  
FROM emp JOIN dep  
ON emp.ndep = dep.ndep;
```

Condição de junção. Só os registros que satisfazem esta condição de junção é que aparecem no resultado

Os dados destes dois atributos vêm da tabela **emp**

Os dados deste atributo vêm da tabela **dep**

NOME	FUNCAO	NOME
Jorge Sampaio	Presidente	Contabilidade
Silvia Teles	Encarregado	Contabilidade
Olga Costa	Continuo	Contabilidade
Augusto Reis	Encarregado	Investigação
Rita Pereira	Continuo	Investigação

Distinguir colunas com o mesmo nome em tabelas diferentes

```
SELECT emp.NOME, FUNCAO, dep.NOME
FROM emp, dep
WHERE dep.NDEP = emp.ndep;
```

Como o atributo nome aparece nas duas tabelas, para os distinguir é necessário inserir o nome da tabela antes

```
SELECT e.NOME, e.FUNCAO, d.NOME
FROM emp AS e, dep AS d
WHERE d.NDEP = e.ndep;
```

Para facilitar, podemos dar um **pseudônimo** ao nome da tabela de modo a ser mais fácil escrever o comando. Por exemplo, a tabela **emp** passa a ser designada por **e** e **dep** por **d**

Junção interna (*inner join*)

A junção interna (*inner join*) é tipo de junção por defeito. Apenas os registos que satisfazem a condição de junção aparecem no resultado

```
SELECT emp.nome, emp.ndep, dep.ndep, dep.nome
FROM emp, dep
WHERE emp.ndep = dep.ndep;
```

Neste exemplo pode ver-se que só os registos de ambas as tabelas que têm o mesmo valor de **ndep** é que aparecem no resultado

NOME	NDEP	NDEP	NOME
-----	-----	-----	-----
Jorge Sampaio	10	10	Contabilidade
Silvia Teles	10	10	Contabilidade
Olga Costa	10	10	Contabilidade
Augusto Reis	20	20	Investigação
Rita Pereira	20	20	Investigação
Catarina Silva	20	20	Investigação
Maria Dias	20	20	Investigação
Antonio Silva	20	20	Investigação
Duarte Guedes	30	30	Vendas
Joana Mendes	30	30	Vendas
Nelson Neves	30	30	Vendas
Ana Rodrigues	30	30	Vendas
Manuel Madeira	30	30	Vendas
Tome Ribeiro	30	30	Vendas
14 rows selected.			



Junção externa (*outer join*)

Para fazer a junção das tabelas Emp e Dep, independentemente de alguns empregados não pertencerem a departamento nenhum, ou de alguns departamentos não terem empregados, é necessário uma junção externa

A junção externa pode ser aplicada à tabela do lado esquerdo ou do lado direito da condição de junção (*LEFT JOIN* ou *RIGHT JOIN*)

Exemplo de junção externa (à direita)

```
SELECT emp.NOME, emp.NDEP, dep.NDEP, dep.NOME  
FROM dep LEFT JOIN emp ON emp.NDEP = dep.NDEP;
```

Tabela da esquerda

Tabela da direita

O departamento 40 não tem empregados mas aparece também no resultado da junção externa

NOME	NDEP	NDEP	NOME
Jorge Sampaio	10	10	Contabilidade
Silvia Teles	10	10	Contabilidade
Olga Costa	10	10	Contabilidade
Augusto Reis	20	20	Investigação
Rita Pereira	20	20	Investigação
Catarina Silva	20	20	Investigação
Maria Dias	20	20	Investigação
Antonio Silva	20	20	Investigação
Duarte Guedes	30	30	Vendas
Joana Mendes	30	30	Vendas
Nelson Neves	30	30	Vendas
Ana Rodrigues	30	30	Vendas
Manuel Madeira	30	30	Vendas
Tome Ribeiro	30	30	Vendas
		40	Planeamento

15 rows selected.

Junção, Projeção e Restrição

Vulgarmente a junção usa-se em conjunto com a Projeção e a restrição, de modo a eliminar colunas e registos desnecessários para o resultado.

```
SELECT emp.nome, funcao, dep.nome  
FROM emp, dep  
WHERE emp.ndep = dep.ndep  
And sal > 220000;
```

Junção: mostra dados das duas tabelas, emp e dep

Projeção: só mostra alguns atributos

Restrição: só mostra alguns registos

NOME	FUNCAO	NOME
Jorge Sampaio	Presidente	Contabilidade
Silvia Teles	Encarregado	Contabilidade
Augusto Reis	Encarregado	Investigação
Catarina Silva	Analista	Investigação
Maria Dias	Analista	Investigação
Duarte Guedes	Encarregado	Vendas
Ana Rodrigues	Vendedor	Vendas

7 rows selected.

Não equi-junção

```
SELECT NOME, SAL, ESCALAO  
FROM emp, descontos  
WHERE sal between salinf and salsup;
```

NOME	SAL	ESCALAO
Tome Ribeiro	56950	1
Rita Pereira	65100	1
Olga Costa	68300	1
Antonio Silva	70800	1
Joana Mendes	145600	2
Manuel Madeira	157800	2
Nelson Neves	212250	3
Ana Rodrigues	221250	3
Silvia Teles	279450	3
Duarte Guedes	380850	4
Catarina Silva	435000	4
Augusto Reis	450975	4
Maria Dias	565000	5
Jorge Sampaio	890000	5

14 rows selected.

Neste caso a condição de junção não é uma igualdade (como quando é feita através das chaves primária + estrangeira) mas sim um intervalo



Self (inner) join

- Mostrar os nomes dos empregados e dos seus respetivos encarregados

```
select e1.nemp, e1.nome, e1.encar, e2.nemp, e2.nome  
from emp as e1, emp as e2  
where e1.encar=e2.nemp;
```

Self (outer) join

- Incluir os que não têm supervisor

```
SELECT e1.nemp, e1.nome, e1.encar, e2.nemp, e2.nome  
FROM emp e1  
LEFT JOIN emp e2 ON e1.encar=e2.nemp;
```

- Incluir o nome de departamento?

```
SELECT e1.nome, e1.encar, e2.nome, d1.nome  
FROM emp AS e1  
JOIN dep AS d1 ON e1.ndep=d1.ndep  
LEFT JOIN emp AS e2 ON e1.encar=e2.nemp
```

Self (outer) join

- Não misturem sintaxes:

```
SELECT e1.nome, e1.encar, e2.nome, d1.nome
FROM emp AS e1, dep AS d1
LEFT JOIN emp AS e2 ON e1.encar=e2.nemp
WHERE e1.ndep=d1.ndep
```

Não funciona!

```
SELECT e1.nome, e1.encar, e2.nome, d1.nome
FROM dep AS d1, emp AS e1
LEFT JOIN emp AS e2 ON e1.encar=e2.nemp
WHERE e1.ndep=d1.ndep
```

Mas este funciona! (precedências)

Com OUTER JOIN usar sempre explicit (ON) JOIN

Operadores relacionais (revisitados)



Operadores relacionais

Operadores unários

- **Restrição**
- **Projeção**


Operadores binários

- **União**
- **Intersecção**
- **Diferença**
- **Divisão**
- **Prod. cartesiano**
- **Junção**

União


A união de duas tabelas resulta numa nova tabela que contém todos os registos existentes nas duas tabelas

```
SELECT nome, funcao, sal
FROM emp
WHERE ndep = 30 AND sal > 220000;
```




NOME	FUNCAO	SAL
-----	-----	-----
Duarte Guedes	Encarregado	380850
Ana Rodrigues	Vendedor	221250

```
SELECT nome, funcao, sal
FROM emp
WHERE funcao = 'Vendedor';
```



NOME	FUNCAO	SAL
-----	-----	-----
Joana Mendes	Vendedor	145600
Nelson Neves	Vendedor	212250
Ana Rodrigues	Vendedor	221250
Manuel Madeira	Vendedor	157800

```
SELECT nome, funcao, sal
FROM emp
WHERE ndep = 30 AND sal > 220000
UNION
SELECT nome, funcao, sal
FROM emp
WHERE funcao = 'Vendedor';
```



NOME	FUNCAO	SAL
-----	-----	-----
Ana Rodrigues	Vendedor	221250
Duarte Guedes	Encarregado	380850
Joana Mendes	Vendedor	145600
Manuel Madeira	Vendedor	157800
Nelson Neves	Vendedor	212250



Compatibilidade em união

As tabelas que participam na união têm de ser **compatíveis em união**:

- Ambas têm de ter o mesmo grau (mesmo número de colunas)
- Colunas correspondentes em cada relação têm de ter o mesmo domínio (o mesmo tipo de dados)

Muitas vezes as uniões são feitas sobre a mesma tabela, de modo a combinar os resultados de duas ou mais instruções SELECT

União com ALL

```
SELECT nome, funcao, sal
FROM emp
WHERE ndep = 30
AND sal > 220000
```

UNION ALL

```
SELECT nome, funcao, sal
FROM emp
WHERE funcao = 'Vendedor'
```

NOME	FUNCAO	SAL
-----	-----	-----
Duarte Guedes	Encarregado	380850
Ana Rodrigues	Vendedor	221250

NOME	FUNCAO	SAL
-----	-----	-----
Joana Mendes	Vendedor	145600
Nelson Neves	Vendedor	212250
Ana Rodrigues	Vendedor	221250
Manuel Madeira	Vendedor	157800

NOME	FUNCAO	SAL
-----	-----	-----
Ana Rodrigues	Vendedor	221250
Duarte Guedes	Encarregado	380850
Joana Mendes	Vendedor	145600
Ana Rodrigues	Vendedor	221250
Manuel Madeira	Vendedor	157800
Nelson Neves	Vendedor	212250

Intersecção

```
SELECT nome, funcao, sal
FROM emp
WHERE ndep = 30
AND sal > 220000
```

INTERSECT

```
SELECT nome, funcao, sal
FROM emp
WHERE funcao = 'Vendedor'
```

NOME	FUNCAO	SAL
-----	-----	-----
Duarte Guedes	Encarregado	380850
Ana Rodrigues	Vendedor	221250

NOME	FUNCAO	SAL
-----	-----	-----
Joana Mendes	Vendedor	145600
Nelson Neves	Vendedor	212250
Ana Rodrigues	Vendedor	221250
Manuel Madeira	Vendedor	157800

NOME	FUNCAO	SAL
-----	-----	-----
Ana Rodrigues	Vendedor	221250

Diferença

```
SELECT nome, funcao, sal
FROM emp
WHERE ndep = 30 AND sal
> 220000
```

EXCEPT

```
SELECT nome, funcao, sal
FROM emp
WHERE funcao =
'Vendedor'
```

NOME	FUNCAO	SAL
-----	-----	-----
Duarte Guedes	Encarregado	380850
Ana Rodrigues	Vendedor	221250

NOME	FUNCAO	SAL
-----	-----	-----
Joana Mendes	Vendedor	145600
Nelson Neves	Vendedor	212250
Ana Rodrigues	Vendedor	221250
Manuel Madeira	Vendedor	157800

NOME	FUNCAO	SAL
-----	-----	-----
Duarte Guedes	Encarregado	380850



Q&A



Databases

SQL - Junções

Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2023/2024

