

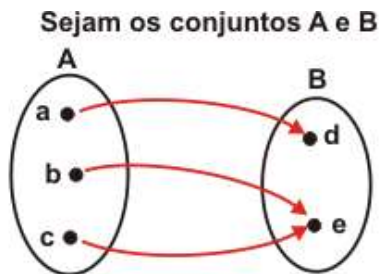
Unidade 5.2: DML – JUNÇÕES DE TABELAS (JOIN)

Prof. Rodrigo Baroni

SQL-DML: Junções

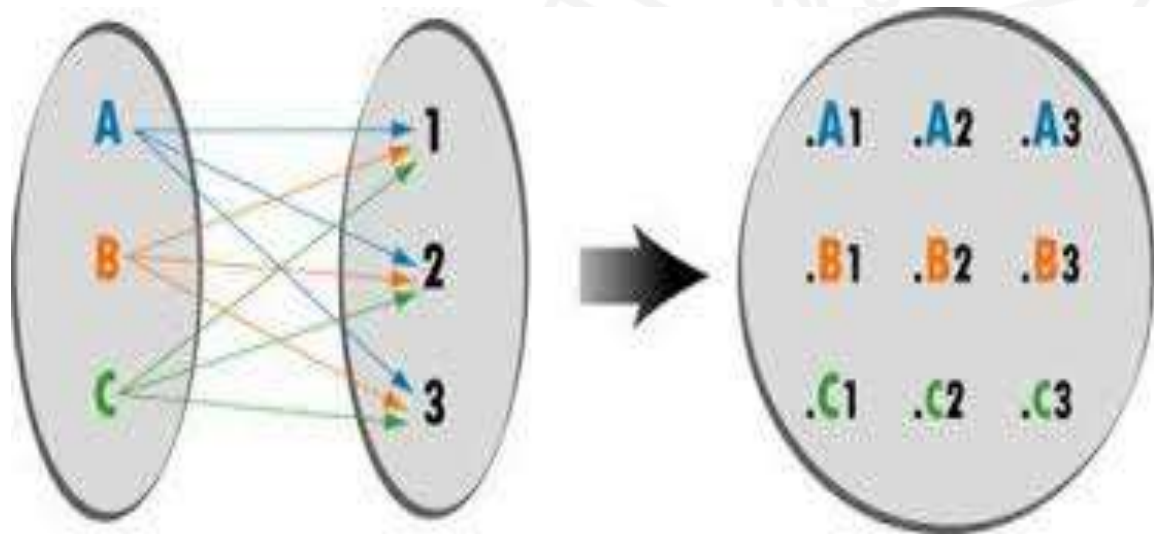
- Usados em consultas (SELECT) que combinam linhas de duas ou mais tabelas ou visões.
- As tabelas e/ou visões são especificadas na cláusula FROM
- As condições de ligação são especificadas na cláusula WHERE. A ligação é usualmente feita através do relacionamento da chave primária com a chave estrangeira e é denominada condição de junção.
- A junção é um subconjunto do **produto cartesiano**.

Produto Cartesiano



O diagram de flechas representa o produto cartesiano
 $A \times B$

$$A \times B = \{(a, d), (b, e), (c, e)\}$$



Tabelas de Exemplo

Funcionario

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
10	Ana	analista	2800	1	100	null	null
20	Jose	prog.	1500	1	200	MG	BH
30	Paulo	gerente	4000	1	100	RJ	Rio
40	Lucas	contador	1400	2	300	RJ	Rio
50	Luiz	gerente	3500	2	200	MG	Betim
60	Joel	analista	3200	1	100	ES	Vitoria

Dependente

matricula	Num-dependente	Nome-dependente
10	1	Gabriel
10	2	Luiza
20	1	Guilherme

Depto

cod-depto	nome-depto
1	Informatica
2	Contabilidade

Produto Cartesiano

- ❖ Os registros são relacionados sem levar em consideração as chaves estrangeiras, gerando todas as combinações possíveis
- ❖ $Qtde_registros \text{ do produto cartesiano} = Qtde_registros \text{ da tabela A} \times Qtde_registros \text{ da Tabela B}$
- ❖ No exemplo, $12 = 6 \times 2$
- ❖ Por isso, cuidado deve ser tomado para não esquecer a condição de junção, pois sem a mesma o comando é executado, mas traz o produto cartesiano.

**Select nome_depto, nome
From Depto A, Funcionario B**

Nome_depto	nome
Informatica	Ana
Informatica	Jose
Informatica	Paulo
Informatica	Lucas
Informatica	Luiz
Informatica	Joel
Contabilidade	Ana
Contabilidade	Jose
Contabilidade	Paulo
Contabilidade	Lucas
Contabilidade	Luiz
Contabilidade	Joel

Junções: Nomes de Atributos Ambíguos

- Duas colunas podem ter o mesmo nome, desde que estejam em tabelas diferentes.
- O nome da coluna é então prefixado com o nome da tabela.

```
SELECT empregado.nome, departamento.nome  
FROM funcionario, departamento  
WHERE funcionario.dep_id=departamento.id;
```

Junções: Apelidos

- Nas consultas com JOIN, colunas de quaisquer tabelas podem ser selecionadas. Pode-se utilizar apelidos (ALIAS) para os nomes das tabelas. Se um campo aparece nas 2 tabelas e é selecionado, é preciso informar de qual tabela o mesmo será exibido.
- Apelidos podem ser criados em qualquer consulta SQL, mesmo que a tabela não seja referenciada mais de uma vez
- Facilita a escrita da consulta

```
SELECT F.Pnome, F.Unome  
FROM funcionario F, departamento D  
WHERE D.Dnome='Pesquisa' AND D.Dnumero=F.Dnr;
```

Tabelas de Exemplo

Funcionario

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
10	Ana	analista	2800	1	100	null	null
20	Jose	prog.	1500	1	200	MG	BH
30	Paulo	gerente	4000	1	100	RJ	Rio
40	Lucas	contador	1400	2	300	RJ	Rio
50	Luiz	gerente	3500	2	200	MG	Betim
60	Joel	analista	3200	1	100	ES	Vitoria

Dependente

matricula	Num-dependente	Nome-dependente
10	1	Gabriel
10	2	Luiza
20	1	Guilherme

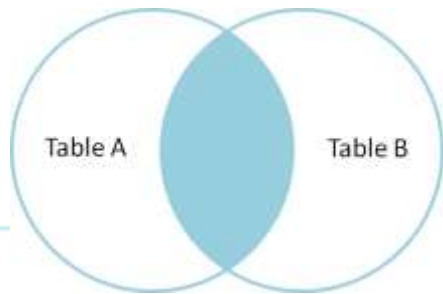
Depto

cod-depto	nome-depto
1	Informatica
2	Contabilidade

SQL-DML: Junções

- Ex: Selecione os nomes dos depts e de seus funcionários com os cargos, em ordem alfabética do nome do depto e do funcionário

```
Select nome_depto, nome, cargo  
From Depto A, Funcionario B  
where A.cod_depto = B.cod_depto  
order by 1, 2
```



Nome_depto	nome	cargo
Contabilidade	Lucas	contador
Contabilidade	Luiz	gerente
Informatica	Ana	analista
Informatica	Joel	Analista
Informatica	Jose	prog.
Informatica	Paulo	gerente

SQL-DML: Junções

Ex: Selecione a média de salários dos funcionários do depto de informática (sem considerar o gerente)

Select avg (salario) as media

From Depto A, Funcionario B

where A.cod_depto = B.cod_depto and nome_depto = 'Informatica'

and cargo <> 'gerente'

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
10	Ana	analista	2800	1	100	null	null
20	Jose	prog.	1500	1	200	MG	BH
30	Paulo	gerente	4000	1	100	RJ	Rio
40	Lucas	contador	1400	2	300	RJ	Rio
50	Luiz	gerente	3500	2	200	MG	Betim
60	Joel	analista	3200	1	100	ES	Vitoria

Media

2500

SQL-DML: Junções

- Ex: Selecione o somatório de salários por nome do departamento, em ordem decrescente do somatório

```
Select nome_depto, sum (salario) as Soma  
From Depto A, Funcionario B  
where A.cod_depto = B.cod_depto  
group by nome_depto  
order by 2 desc
```

Nome_depto	Soma
Informatica	11500
Contabilidade	4900

Tabelas de Exemplo

Funcionario

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
10	Ana	analista	2800	1	100	null	null
20	Jose	prog.	1500	1	200	MG	BH
30	Paulo	gerente	4000	1	100	RJ	Rio
40	Lucas	contador	1400	2	300	RJ	Rio
50	Luiz	gerente	3500	2	200	MG	Betim
60	Joel	analista	3200	1	100	ES	Vitoria

Dependente

matricula	Num-dependente	Nome-dependente
10	1	Gabriel
10	2	Luiza
20	1	Guilherme

Depto

cod-depto	nome-depto
1	Informatica
2	Contabilidade

SQL-DML: Junções com 3 Tabelas

- Ex: Selecione os nomes dos depts e de seus funcionários que tenham dependentes, em ordem alfabética do nome do depto, do funcionário e do nome do dependente

```
Select nome_depto, nome, nome_dependente  
From Depto A, Funcionario B, Dependente C  
where A.cod_depto = B.cod_depto AND B.matricula = C.matricula  
order by 1, 2, 3
```

Nome_depto	nome	Nome_dependente
Informatica	Ana	Gabriel
Informatica	Ana	Luiza
Informatica	Jose	Guilherme

SQL-DML: Junções

– TIPOS DE JUNÇÕES:

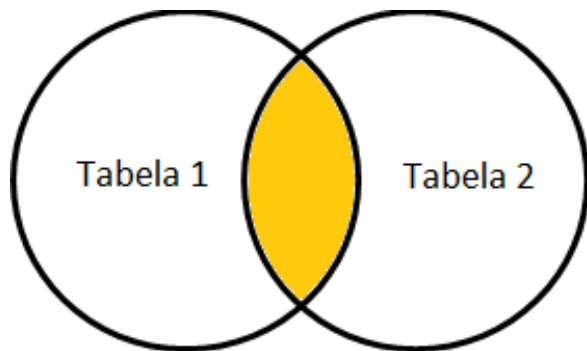
- inner join (equivale à sintaxe tradicional da junção)
- left outer join
- right outer join
- full outer join

– CONDIÇÕES:

- on <predicado>



INNER JOIN



```
SELECT coluna1, coluna2, ... colunan  
FROM tabela1 INNER JOIN tabela2  
ON tabela1.coluna = tabela2.coluna;
```

Sintaxe alternativa

```
SELECT coluna1, coluna2, ... colunan  
FROM tabela1, tabela2  
WHERE tabela1.coluna = tabela2.coluna;
```

INNER JOIN

Ex: Selecione os nomes dos depts e de seus funcionários com os cargos, em ordem alfabética do nome do depto e do funcionário

```
Select nome_depto, nome, cargo  
From Depto A, Funcionario B  
where A.cod_depto = B.cod_depto  
order by 1, 2
```

É equivalente a:

```
Select nome_depto, nome, cargo  
From Depto A INNER JOIN Funcionario B  
ON A.cod_depto = B.cod_depto  
order by 1, 2
```

Obs: INNER JOIN pode ser substituído apenas pela palavra JOIN, já que é o join *default*

Nome_depto	nome	cargo
Contabilidade	Lucas	contador
Contabilidade	Luiz	gerente
Informatica	Ana	analista
Informatica	Joel	Analista
Informatica	Jose	prog.
Informatica	Paulo	gerente

Tabelas de Exemplo

Funcionario

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
10	Ana	analista	2800	1	100	null	null
20	Jose	prog.	1500	1	200	MG	BH
30	Paulo	gerente	4000	1	100	RJ	Rio
40	Lucas	contador	1400	2	300	RJ	Rio
50	Luiz	gerente	3500	2	200	MG	Betim
60	Joel	analista	3200	1	100	ES	Vitoria

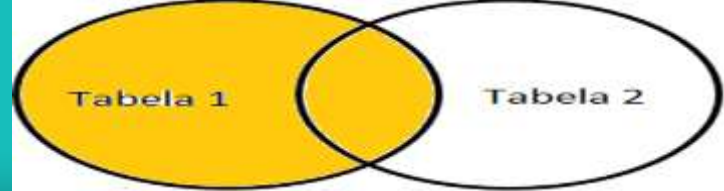
Dependente

matricula	Num-dependente	Nome-dependente
10	1	Gabriel
10	2	Luiza
20	1	Guilherme

Depto

cod-depto	nome-depto
1	Informatica
2	Contabilidade

LEFT JOIN



Ex: Selecione os nomes dos funcionários e de seus dependentes, exibindo também funcionários sem dependentes, em ordem alfabética dos nomes

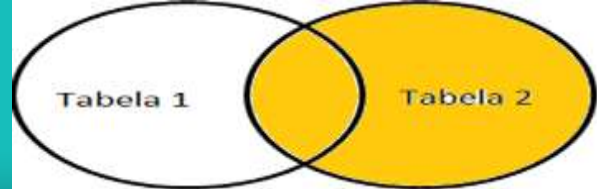
```
Select nome, nome_dependente
From Funcionario A LEFT OUTER JOIN Dependente B
ON A.matricula = B.matricula
order by 1, 2
```

Pergunta: Quantos registros seriam exibidos se fosse um INNER JOIN ?

O número de registros exibidos em um Left Join é no mínimo igual a um Inner Join, sendo na maioria das vezes maior, pois o Left Join engloba o Inner Join

nome	Nome_dependente
Ana	Gabriel
Ana	Luiza
Joel	
Jose	Guilherme
Lucas	
Luiz	
Paulo	

RIGHT JOIN



Ex: Selecione os nomes dos funcionários e de seus dependentes, exibindo também funcionários sem dependentes, em ordem alfabética dos nomes

```
Select nome, nome_dependente
From Dependente A RIGHT OUTER JOIN Funcionario B
ON A.matricula = B.matricula
order by 1, 2
```

No RIGHT JOIN, a tabela à direita é quem manda.

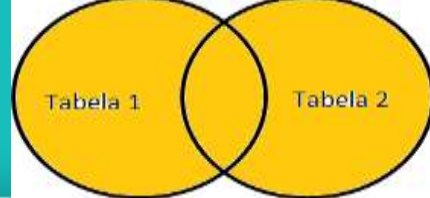
Tabela1 LEFT JOIN Tabela2

é igual a Tabela2 RIGHT JOIN Tabela1

nome	Nome_dependente
Ana	Gabriel
Ana	Luiza
Joel	
Jose	Guilherme
Lucas	
Luiz	
Paulo	

Obs: Expressão OUTER pode ser suprimida no Left, Right e FullJoins

FULL JOIN



Ex: Selecione os nomes dos departamentos e dos funcionários, exibindo departamentos sem funcionários e também funcionários sem departamentos, em ordem alfabética dos nomes

```
Select nome_depto, nome  
From Depto A FULL OUTER JOIN Funcionario B  
ON A.cod_depto = B.cod_depto  
order by 1, 2
```

Resultado:

Full Join é a soma do Left Join com o Right Join

Full Join é diferente do Produto cartesiano pois não gera combinações aleatórias

Nome_depto	Nome
Contabilidade	Lucas
Contabilidade	Luiz
Informática	Ana
Informática	Joel
Informática	Jose
Informática	Paulo
Patrimonio	
	Joaquim

UNION (União)

- Agrega resultados de comandos SELECT. Deve existir compatibilidade de colunas e as linhas duplicadas são desprezadas.
- UNION ALL: não elimina tuplas duplicadas
- Equivale a operação de união da teoria dos conjuntos. Pode atuar em tabelas diferentes desde que a compatibilidade seja respeitada.

UNION (União)

- Ex:Selecione os nomes e estados dos funcionários de MG e do RJ, marcando os de MG com um asterisco
- `Select nome, estado, '*' From Funcionario where estado = 'MG' UNION
select nome, estado, '' from funcionario where estado = 'RJ'`

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
10	Ana	analista	2800	1	100	null	null
20	Jose	prog.	1500	1	200	MG	BH
30	Paulo	gerente	4000	1	100	RJ	Rio
40	Lucas	contador	1400	2	300	RJ	Rio
50	Luiz	gerente	3500	2	200	MG	Betim
60	Joel	analista	3200	1	100	ES	Vitoria

Nome	Estado	
Jose	MG	*
Paulo	RJ	
Lucas	RJ	
Luiz	MG	*

UNION (União)

Selecione os funcionários recém-admitidos com uma mensagem de Bem-Vindo e os que completaram 5 anos de casa com Parabéns

```
SELECT MATRIC_TEMP, NOME, 'BEM-VINDO'  
FROM ENTREVISTAS  
WHERE POSICAO = 'ADM' AND  
UNION  
SELECT MATRICULA, NOME, 'PARABENS'  
FROM FUNCIONARIOS  
WHERE ANOS = 5  
ORDER BY 2
```



PUC Minas
Virtual