

SQL - PARTE 3

Sérgio Mergen

Operações de Junção

- O produto cartesiano permite combinar registros de duas instâncias de tabelas
 - E logo após aplicar critérios de junção com a cláusula WHERE
- O mesmo pode ser obtido com o operador de junção
 - E seus derivados

Junção versus Produto Cartesiano

- Listar os dados de projeto e do seu respectivo departamento para projetos com custo estimado superior a R\$15.000.

```
select  p.*, d.nome  
from    projeto p, depto d  
where   p.idDepto = d.idDepto  
and     custo > 15000
```

Versus

```
select  p.*, d.nome  
from    projeto p natural inner join depto d  
where   custo > 15000
```

- A última separa claramente onde ir buscar os dados de onde se colocam condições “de filtragem”

Tipos de Junção

- Três tipos básicos de junção.
 - Junção Cruzada (Cross Join)
 - Realiza um **produto cartesiano**
 - Junção Interna (Inner Join)
 - Realiza um **produto cartesiano** seguido da aplicação de **critérios de junção**
 - Um registro de uma tabela **só é retornado se ele possuir correspondências** com a tabela do outro lado
 - Junção Externa (Outer Join)
 - Realiza um produto cartesiano seguido da aplicação de critérios de junção
 - Um registro de uma tabela **é retornado mesmo que não possua correspondências** com a tabela do outro lado

Tipos de Junção

- Junção Cruzada (Cross Join)
 - Forma de uso
 - Tabela1 **Cross join** tabela2
 - Realiza um **produto cartesiano**
 - Combina todos registros da tabela do lado esquerdo da junção com todos registros do lado direito da junção
 - Por isso, não requer critérios de junção

Exemplo

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	3	12.000	1
2	Lucrei	2	30.000	2

Depto		
idDepto	nome	predio
1	TI	3
2	Marketing	2

Projeto p **cross join** depto

Resposta							
idProj	nome	duracao	custo	idDepto	idDepto	nome	predio
1	ABC	3	12.000	1	1	TI	3
1	ABC	3	12.000	1	2	Marketing	2
2	Lucrei	2	30.000	2	1	TI	3
2	Lucrei	2	30.000	2	2	Marketing	2

Tipos de Junção

- Junção Interna (Inner Join):
 - Forma de uso
 - Tabela1 (**inner**) join tabela2
 - Realiza um **produto cartesiano** seguido da aplicação de **critérios de junção**
 - Um registro de uma tabela **só é retornado se ele possuir correspondências** com a tabela do outro lado
 - Deve-se especificar os critérios de junção da seguinte forma:

Critérios de Junção
on critérios using (A_1, A_2, \dots, A_n)

onde $A_1 \dots A_n$ são atributos

Exemplo

Projeto				
idProj	nome	duracao	custo	idDeppto
1	ABC	3	12.000	1
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	
4	Caos	10	100.000	1

Deppto		
idDeppto	nome	predio
1	TI	3
2	Marketing	2
3	RH	2

Projeto p **inner join** deppto d **on** p.idDeppto = d.idDeppto

Resposta							
idProj	nome	duracao	custo	idDeppto	idDeppto	nome	predio
1	ABC	3	12.000	1	1	TI	3
2	Lucrei	2	30.000	2	2	Marketing	2
4	Caos	10	100.000	1	1	TI	3

Exemplo

Projeto				
idProj	nome	duracao	custo	idDeppto
1	ABC	3	12.000	1
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	
4	Caos	10	100.000	1

Deppto		
idDeppto	nome	predio
1	TI	3
2	Marketing	2
3	RH	2

Projeto p **inner join** deppto d **using (idDeppto)**

Resposta							
idProj	nome	duracao	custo	idDeppto	idDeppto	nome	predio
1	ABC	3	12.000	1	1	TI	3
2	Lucrei	2	30.000	2	2	Marketing	2
4	Caos	10	100.000	1	1	TI	3

Tipos de Junção

- Junção Externa (Outer Join):
 - Realiza um **produto cartesiano** seguido da aplicação de **critérios de junção**
 - Um registro de uma tabela **é retornado mesmo que não possua correspondências** com a tabela do outro lado
 - Pode-se definir critérios de junção da mesma forma que a junção interna
- Formas de uso da junção externa
 - Tabela1 **Left (outer) join** tabela2
 - Todos registros da tabela da esquerda da junção são retornados
 - Tabela1 **Right (outer) join** tabela2
 - Todos registros da tabela da direita da junção são retornados
 - Tabela1 **Full outer join** tabela2
 - Todos registros das duas tabelas são retornados

Exemplo

Projeto				
idProj	nome	duracao	custo	idDeppto
1	ABC	3	12.000	1
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	null
4	Caos	10	100.000	1

Deppto		
idDeppto	nome	predio
1	TI	3
2	Marketing	2
3	RH	2

Projeto p **left outer join** depto d **on** p.idDeppto = d.idDeppto

Resposta							
idProj	nome	duracao	custo	idDeppto	idDeppto	nome	predio
1	ABC	3	12.000	1	1	TI	3
2	Lucrei	2	30.000	2	2	Marketing	2
4	Caos	10	100.000	1	1	TI	3
3	Genesis	2	15.000	null	null	null	nulll

Operações de Junção

- Junção natural
 - Modificador que pode ser aplicado em uma junção
 - Implica que a junção será realizada com base nos atributos de mesmo nome nas tabelas envolvidas
 - Por esse motivo, não requer critérios de junção explícitos
- Pode ser aplicado nos seguintes tipos de junção
 - Junção interna
 - Natural (Inner) join
 - Junção externa
 - Natural Left (outer) join
 - Natural Right (outer) join
 - Natural Full outer join

Exemplo

Projeto				
idProj	nomeP	duracao	custo	idDepto
1	ABC	3	12.000	1
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	null
4	Caos	10	100.000	1

Depto		
idDepto	nomeD	predio
1	TI	3
2	Marketing	2
3	RH	2

Projeto p **natural join** depto d

Resposta							
idProj	nomeP	duracao	custo	idDepto	idDepto	nomeD	predio
1	ABC	3	12.000	1	1	TI	3
2	Lucrei	2	30.000	2	2	Marketing	2
4	Caos	10	100.000	1	1	TI	3

Exemplo

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	3	12.000	1
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	null
4	Caos	10	100.000	1

Depto		
idDepto	nome	predio
1	TI	3
2	Marketing	2
3	RH	2

Projeto p **natural join** depto d

Exemplo de uso equivocado da junção natural.

Por quê?

Valores Nulos

- As tuplas podem conter valores nulos, denotado por *null*, em alguns dos seus atributos.
 - *null* significa um valor desconhecido ou que não existe.

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	3	12.000	1
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	null
4	Caos	10	100.000	1

Valores Nulos

- O predicado **is null** pode ser utilizado para testar a existência de valores nulos.
 - E.g. mostrar todos os projetos que não possuem custo definido.

```
select nome  
from projeto  
where custo is null
```

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	3	12.000	1
2	Lucrei	2	30.000	2
3	Genesis	2	null	2
4	Caos	10	100.000	1

Valores Nulos

- O resultado de uma expressão aritmética com *null* é *null*
- E.g.
 - $5 + \text{null} = \text{null}$
 - $10 * \text{null} = \text{null}$
 - $\text{Null} / \text{null} = \text{null}$
 - $(\text{null} + 3 * 14) / 4 = \text{null}$

Valores Nulos e Expressões Aritméticas

- Listar os projetos com tempo de duração estimado em número de meses

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	12.000	1
3	Show	null	5.000	2

Valores Nulos e Expressões Aritméticas

- Listar os projetos com tempo de duração estimado em número de meses

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	12.000	1
3	Show	null	5.000	2

```
select    nome, duracao * 12 as tempo
from      projeto p
```

Resposta	
nome	tempo
1	24
3	null

Valores Nulos e Comparações

- Qualquer comparação com *null* retorna *unknown*
 - *E.g.*
 - $5 < \text{null} = \text{unknown}$
 - $\text{null} <> \text{null} = \text{unknown}$
 - $\text{null} = \text{null} = \text{unknown}$
- Resultado da condição da cláusula **where** é tratado como **false** quando o seu valor é *unknown*

Valores Nulos e Comparações

- Listar os projetos com duração estimada superior a 24 meses

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	12.000	1
2	Lucrei	3	30.000	1
3	Show	null	5.000	2

Valores Nulos e Comparações

- Listar os projetos com duração estimada superior a 24 meses

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	12.000	1
2	Lucrei	3	30.000	1
3	Show	null	5.000	2

```
select      *  
from        projeto p  
where       duracao * 12 > 24
```

Resposta				
idProj	nome	duracao	custo	idDepto
2	Lucrei	3	30.000	1

Valores Nulos e Lógica Ternária

- Lógica trivalente usando o valor lógico *unknown*:
 - OR:
 - (unknown **or** true) = true
 - (unknown **or** false) = unknown
 - (unknown **or** unknown) = unknown
 - AND:
 - (true **and** unknown) = unknown
 - (false **and** unknown) = false
 - (unknown **and** unknown) = unknown
 - NOT:
 - (**not** unknown) = unknown
 - **P is unknown**
 - é verdade se o valor de P é unknown

Valores Nulos e Lógica Ternária

- Listar os projetos com duração estimada inferior a 3 anos e custo estimado inferior a 15.000

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	12.000	1
2	Lucrei	3	30.000	1
3	Show	1	null	2

```
select      *  
from        projeto p  
where       duracao < 3 and custo < 15.000
```

Valores Nulos e Lógica Ternária

- Listar os projetos com duração estimada inferior a 3 anos e custo estimado inferior a 15.000

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	12.000	1
2	Lucrei	3	30.000	1
3	Show	1	null	2

```
select      *  
from        projeto p  
where       duracao < 3 and custo < 15.000
```

Resposta				
idProj	nome	duracao	custo	idDepto
1	ABC	2	12.000	1

Valores Nulos e Lógica Ternária

- Listar os projetos com duração estimada inferior a 3 anos ou custo estimado inferior a 15.000

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	12.000	1
2	Lucrei	3	30.000	1
3	Show	1	null	2

```
select      *  
from        projeto p  
where       duracao < 3 or custo < 15.000
```

Valores Nulos e Lógica Ternária

- Listar os projetos com duração estimada inferior a 3 anos ou custo estimado inferior a 15.000

Projeto				
idProj	nome	duracao	custo	idDeppto
1	ABC	2	12.000	1
2	Lucrei	3	30.000	1
3	Show	1	null	2

```
select      *
from        projeto p
where       duracao < 3 or custo < 15.000
```

Resposta				
idProj	nome	duracao	custo	idDeppto
1	ABC	3	12.000	1
3	Show	1	null	2

Valores Nulos e Lógica Ternária

- Listar os projetos com custo diferente de 30.000

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	12.000	1
2	Lucrei	3	30.000	1
3	Show	1	null	2

```
select *  
from projeto p  
where custo <> 30.000
```

Resposta				
idProj	nome	duracao	custo	idDepto
1	ABC	3	12.000	1

Valores Nulos e Lógica Ternária

- Listar os projetos com custo diferente de 30.000

Projeto				
idProj	nome	duracao	custo	idDeppto
1	ABC	2	12.000	1
2	Lucrei	3	30.000	1
3	Show	1	null	2

```
select      *
from        projeto p
where       custo <> 30.000 or (custo <> 30.000) is unknown
```

Resposta				
idProj	nome	duracao	custo	idDeppto
1	ABC	3	12.000	1
3	Show	1	null	2

Valores Nulos e Agregações

- Agrupamento
 - Valores nulos são agrupados em um só valor
- Agregação
 - As funções de agregação ignoram os nulos
 - Caso específico
 - Se todos os valores a serem agregados forem nulos
 - O resultado da agregação é nulo

Valores Nulos e Agregações

- Listar quantas vezes cada custo foi aplicado

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	20.000	1
2	Lucrei	3	null	1
3	ACME	3	20.000	1
4	Show	1	null	2

```
select    custo, count(*) as qtd
from      projeto p
group by  custo
```

Resposta	
custo	qtd
null	2
20.000	2

Valores Nulos e Agregações

- Listar o custo total de todos os projetos

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	12.000	1
2	Lucrei	3	30.000	1
3	Show	1	null	2

```
select    sum(custo)
from      projeto p
```

Valores Nulos e Agregações

- Listar o custo total de todos os projetos

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	12.000	1
2	Lucrei	3	30.000	1
3	Show	1	null	2

```
select    sum(custo)
from      projeto p
```

Resposta
Sum(custo)
42.000

Valores Nulos e Agregações

- Listar o custo total de todos os projetos

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	null	1
2	Lucrei	3	null	1
3	Show	1	null	2

```
select    sum(custo)
from      projeto p
```

Valores Nulos e Agregações

- Listar o custo total de todos os projetos

Projeto				
idProj	nome	duracao	custo	idDepto
1	ABC	2	null	1
2	Lucrei	3	null	1
3	Show	1	null	2

```
select    sum(custo)
from      projeto p
```

Resposta
Sum(custo)
null

Atividade Individual

- Execute o script fornecido para criar o banco de dados de filmes no MySQL.
- Resolva as consultas solicitadas no moodle.
 - Apenas as **10 primeiras consultas** serão avaliadas
 - As seis últimas não serão avaliadas, mas podem ser entregues junto com as 10 primeiras
 - É interessante tentar resolvê-las
 - Elas podem explorar comandos não vistos durante a aula
 - Essa é uma boa forma de aprender mais a respeito dos recursos da linguagem SQL

SQL – PARTE 3
