

# SQL

- **Aninhamentos: IN, ANY, ALL, SOME**
- **JOINs**

Carina Friedrich Dorneles  
dorneles@inf.ufsc.br

*Banco de Dados*

# IN

---

- ▶ Nomes dos pacientes que tem consulta em janeiro de 2002

```
SELECT p.nome
FROM paciente p
WHERE p.codigo IN
      (SELECT c.codpac
       FROM consulta c
       WHERE c.data BETWEEN
        '01-01-2002' AND '31-01-2002')
```



## NOT IN

---

- ▶ Nomes dos pacientes que **não** tem consulta em janeiro de 2002

```
SELECT p.nome
FROM paciente p
WHERE p.codigo NOT IN
      (SELECT c.codpac
       FROM consulta c
       WHERE c.data BETWEEN
            '01-01-2002' AND '31-01-2002')
```



# IN para conjunto de valores

---

- ▶ Selecionar o nome das cidades cujo estado seja 'RS', 'SC' ou 'PR'

```
SELECT c.nome  
FROM cidade c  
WHERE UF IN ( 'RS' , 'SC' , 'PR' )
```

Pode substituir o OR

```
WHERE UF = 'RS' OR  
      UF = 'SC' OR  
      UF = 'PR'
```

---

# IN para subconsultas

---

- ▶ Selecionar o nome do produto que possua o maior preço de custo.



# IN para subconsultas

---

- ▶ Selecionar o nome do produto que possua o maior preço de custo.

```
SELECT p.nome  
FROM produto p  
WHERE p.precoCusto IN (SELECT max(precoCusto)  
                        FROM produto)
```



# IN para subconsultas

---

- ▶ Selecionar o nome do produto que possua o maior preço de custo.

```
SELECT p.nome  
FROM produto p  
WHERE p.precoCusto IN (SELECT max(precoCusto)  
                        FROM produto)
```

- ▶ Isso não existe:

```
SELECT p.nome  
FROM produto p  
WHERE p.precoCusto = max(precoCusto)
```



# IN para subconsultas

---

- ▶ Selecionar o nome do produto que possua o maior preço de custo.

```
SELECT p.nome  
FROM produto p  
WHERE p.precoCusto IN (SELECT max(precoCusto)  
                        FROM produto)
```

- ▶ Isso não existe:

```
SELECT p.nome  
FROM produto p  
WHERE p.precoCusto = max(precoCusto)
```





# SQL

## ANY, ALL e SOME

Carina Friedrich Dorneles  
dorneles@inf.ufsc.br

*Banco de Dados*

# Subconsultas

## ▶ Consultas mais internas podem retornar

- ▶ Uma única linha
- ▶ Várias linhas

## ▶ Uma linha:

```
SELECT nome
FROM produto
WHERE precoCusto = (SELECT MIN(valorcusto)
                    FROM produto)

SELECT data, sum(precoVenda)
FROM venda
GROUP BY data
HAVING sum(precoVenda) = (SELECT MIN(valorcusto)
                        FROM produto)
```

```
SELECT nome
FROM produto
WHERE precoCusto > (SELECT AVG(valorcusto)
                   FROM produto)
```

▶ > , < , >= , <= , <>

# Subconsultas

---

## ► Várias linhas:

- *Retornar nome do produto cujo preço de custo se iguale a preços de custo dos produtos de vestuário.*

```
SELECT nome
FROM produto
WHERE precoCusto IN (SELECT valorcusto
                       FROM produto
                       WHERE categoria = 'vestuário')
```



# Subconsultas

---

## ▶ Várias linhas:

- ▶ *Retornar nome do produto cujo preço de custo se iguale a preços de custo dos produtos de vestuário.*

```
SELECT nome
FROM produto
WHERE precoCusto IN (SELECT valorcusto
                      FROM produto
                      WHERE categoria = 'vestuário')
```

Como usar  $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $<>$  para conjunto de valores?

---



# Subconsultas

---

- ▶ Operadores >, <, >=, <= e <> combinados com **ANY** (ou **SOME**) ou **ALL**:

```
SELECT nome
FROM produto
WHERE precoCusto > ANY (SELECT valorcusto
                           FROM produto
                           WHERE categoria = 'vestuário')
```

```
SELECT nome
FROM produto
WHERE precoCusto > ALL (SELECT valorcusto
                        FROM produto
                        WHERE categoria = 'vestuário')
```

---



# SQL

## JOIN(S)

Carina Friedrich Dorneles  
dorneles@inf.ufsc.br

*Banco de Dados*

# Introdução

---

```
SELECT m.nome  
FROM medico m JOIN consulta c  
                ON m.codigo = c.codmed  
WHERE m.idade > 30
```

- Junção vista até agora em aula



# Tipos de Joins no SQL

---

- ▶ Dois principais tipos de JOIN no SQL

- ▶ **INNER JOINS**

- ▶ JOIN
    - ▶ NATURAL JOIN

- ▶ **OUTER JOINS**

- ▶ RIGHT JOIN
    - ▶ LEFT JOIN
    - ▶ FULL JOIN





# INNER JOIN (JOIN)

---

```
SELECT m.nome, COUNT(c.data) AS qtdConsultas  
FROM medico m JOIN consulta c ON m.codigo = c.codmed  
WHERE m.idade > 30  
GROUP BY m.nome
```



# INNER JOIN (JOIN)

---

```
SELECT m.nome, COUNT(c.data) AS qtdConsultas
FROM medico m JOIN consulta c ON m.codigo = c.codmed
WHERE m.idade > 30
GROUP BY m.nome
```

- ▶ Esta consulta irá listar os médicos e a quantidade de consultas dele

<i><b>nome</b></i>	<i><b>qtdConsultas</b></i>
Ana	10
Carlos	23
Paula	45
Everton	16



# INNER JOIN (NATURAL JOIN)

---

- ▶ Junta as tabelas pelas colunas de mesmo nome

Medico

CodMed	Nome
1	Ana
2	Carlos
3	Paula

Ender\_medico

CodMed	logradouro	cep
1	Av. dos Bandeirantes	99800222
2	Rua Dona Bela	98000020
3	Rua Dom Seastiao	89660000



# INNER JOIN (NATURAL JOIN)

---

- ▶ Junta as tabelas pelas colunas de mesmo nome

Medico

CodMed	Nome
1	Ana
2	Carlos
3	Paula

Ender\_medico

CodMed	logradouro	cep
1	Av. dos Bandeirantes	99800222
2	Rua Dona Bela	98000020
3	Rua Dom Seastiao	89660000

```
SELECT m.nome, em.logradouro, em.cep
FROM medico m NATURAL JOIN ender_medico em
WHERE m.idade > 30
```

nome	logradouro	cep
Ana	Av. dos Bandeirantes	99800222
Carlos	Rua Dona Bela	98000020
Paula	Rua Dom Seastiao	89660000



# USING

---

## ► Similar ao natural JOIN

Medico

CodMed	Nome
1	Ana
2	Carlos
3	Paula

Ender\_medico

CodMed	logradouro	cep
1	Av. dos Bandeirantes	99800222
2	Rua Dona Bela	98000020
3	Rua Dom Seastiao	89660000

```
SELECT m.nome, em.logradouro, em.cep
FROM medico m NATURAL JOIN ender_medico em USING (codmed)
WHERE m.idade > 30
```

nome	logradouro	cep
Ana	Av. dos Bandeirantes	99800222
Carlos	Rua Dona Bela	98000020
Paula	Rua Dom Seastiao	89660000



# INNER JOIN (JOIN)

---

```
SELECT m.nome, COUNT(c.data) AS qtdConsultas  
FROM medico m JOIN consulta c ON m.codigo = c.codmed  
WHERE m.idade > 30  
GROUP BY m.nome
```

- ▶ Esta consulta irá listar os médicos e a quantidade de consultas dele

<i><b>nome</b></i>	<i><b>qtdConsultas</b></i>
Ana	10
Carlos	23
Paula	45
Everton	16



# JOIN

---

- ▶ Mas e se quisermos que apareça assim:

Mostrando, inclusive, os nomes daqueles médicos que não possuem consultas

<b><i>nome</i></b>	<b><i>qtdConsultas</i></b>
Ana	10
Carlos	23
Paula	45
Everton	16
Betania	NULL
Tito	NULL
Pedro	NULL
Laura	NULL



# JOIN

---

- ▶ Mas e se quisermos que apareça assim:

Mostrando, inclusive, os nomes daqueles médicos que não possuem consultas

**USAR:    OUTER JOIN**

<i><b>nome</b></i>	<i><b>qtdConsultas</b></i>
Ana	10
Carlos	23
Paula	45
Everton	16
Betania	NULL
Tito	NULL
Pedro	NULL
Laura	NULL





# OUTER JOIN

---

- ▶ Três tipos:

- ▶ **LEFT OUTER JOIN**

- ▶ ou simplesmente **LEFT JOIN**
    - ▶ **LEFT = esquerda**

- ▶ **RIGHT OUTER JOIN**

- ▶ ou simplesmente **RIGHT JOIN**
    - ▶ **RIGHT = direita**

- ▶ **FULL OUTER JOIN**

- ▶ ou simplesmente **FULL JOIN**
    - ▶ **FULL = completa**



# LEFT OUTER JOIN

---

- ▶ Traz todos os dados da tabela da **esquerda**, não importando o que tem na da tabela da **direita**

- ▶ Exemplo:

```
SELECT m.nome, COUNT(c.data) AS qtdConsultas  
FROM medico m LEFT OUTER JOIN consulta c  
ON m.codigo = c.codmed  
GROUP BY m.nome
```



# LEFT OUTER JOIN

---

- ▶ Traz todos os dados da tabela da **esquerda**, não importando o que tem na da tabela **direita**

Traz todos os  
médicos

Independente se ele tem consulta

- ▶ Exemplo:

```
SELECT m.nome, COUNT(c.data) AS qtdConsultas
FROM medico m LEFT OUTER JOIN consulta c
ON m.codigo = c.codmed
GROUP BY m.nome
```



# LEFT OUTER JOIN

---

```
SELECT m.nome, COUNT(c.data) AS qtdConsultas
FROM medico m LEFT OUTER JOIN consulta c
ON m.codigo = c.codmed
GROUP BY m.nome
```

<i><b>nome</b></i>	<i><b>qtdConsultas</b></i>
Ana	10
Carlos	23
Paula	45
Everton	16
Betania	NULL
Tito	NULL
Pedro	NULL
Laura	NULL



# RIGHT OUTER JOIN

---

► Traz todos os dados da tabela da **direita**, não importando o que tem na da tabela da **esquerda**

► Exemplo:

```
SELECT c.data, c.hora, r.descricao
FROM consulta c RIGHT OUTER JOIN receita r
ON c.codReceita = r.codigo
```



# RIGHT OUTER JOIN

---

- ▶ Traz todos os dados da tabela da **direita**, não importando o que tem na da tabela da **esquerda**

Independente se indicadas em consultas

Traz todas as receitas

- ▶ Exemplo:

```
SELECT c.data, c.hora, r.descricao
FROM consulta c RIGHT OUTER JOIN receita r
ON c.codReceita = r.codigo
```



# RIGHT OUTER JOIN

---

```
SELECT c.data, c.hora, r.descricao  
FROM consulta c RIGHT OUTER JOIN receita r  
ON c.codReceita = r.codigo
```

<i><b>data</b></i>	<i><b>hora</b></i>	<i><b>descricao</b></i>
10/10/2006	9h	Tylenol - 1 vez ao dia
10/11/2006	9h30	Xarope TT - 3x ao dia
10/12/2006	9h30	Creme gel - 1x dia
23/10/2006	14h20	Comp. HH - 2x ao dia
21/11/2006	15h30	Doril - 1x ao dia
NULL	NULL	Claritin - 2x ao dia
NULL	NULL	Sorine - 1x ao dia



# FULL OUTER JOIN

---

- ▶ Traz todos os dados da tabela da **direita**, e todos da **esquerda**
- ▶ Exemplo:

```
SELECT c.data, c.hora, r.descricao  
FROM consulta c FULL OUTER JOIN receita r  
ON c.codReceita = r.codigo
```





# FULL OUTER JOIN

---

```
SELECT c.data, c.hora, r.descricao
FROM consulta c FULL OUTER JOIN receita r
ON c.codReceita = r.codigo
```

<i><b>data</b></i>	<i><b>hora</b></i>	<i><b>descricao</b></i>
10/10/2006	9h	Tylenol - 1 vez ao dia
10/11/2006	9h30	Xarope TT - 3x ao dia
10/12/2006	9h30	Creme gel - 1x dia
23/10/2006	14h20	Comp. HH - 2x ao dia
21/11/2006	15h30	Doril - 1x ao dia
NULL	NULL	Claritin - 2x ao dia
NULL	NULL	Sorine - 1x ao dia
10/11/2006	14h30	NULL
10/11/2006	15h30	NULL
10/11/2006	16h	NULL



# LEFT OUTER JOIN

---

**Anulando uma junção externa. Supondo o exemplo:**

```
SELECT m.nome, c.data  
FROM medico m LEFT OUTER JOIN consulta c  
ON m.codigo = c.codmed  
GROUP BY m.nome, c.data
```

<i>nome</i>	<i>data</i>
Ana	10/10/2009
Carlos	23/05/2009
Paula	04/05/2009
Everton	16/12/2009
Betania	NULL
Tito	NULL
Pedro	NULL
Laura	NULL



# LEFT OUTER JOIN

## Anulando uma junção externa

```
SELECT m.nome, c.data
FROM medico m LEFT OUTER JOIN consulta c
ON m.codigo = c.codmed
WHERE data < '01/01/2009'
GROUP BY m.nome
```

*Filtra* as linhas em  
cima da junção já  
feita

<i>nome</i>	<i>data</i>
Ana	10/10/2009
Carlos	23/05/2009
Paula	04/05/2009
Everton	16/12/2009
Botania	NULL
Tito	NULL
Pedro	NULL
Laura	NULL

# LEFT OUTER JOIN

---

## Anulando uma junção externa

```
SELECT m.nome, c.data  
FROM medico m LEFT OUTER JOIN consulta c  
ON m.codigo = c.codmed AND data < '01/01/2009'  
GROUP BY m.nome
```

O filtro é feito  
*durante a junção*

<i>nome</i>	<i>data</i>
Ana	10/10/2009
Carlos	23/05/2009
Paula	04/05/2009
Everton	16/12/2009
Betania	NULL
Tito	NULL
Pedro	NULL
Laura	NULL