

SQL – DML

Select – from - where

Carina F. Dorneles
dorneles@inf.ufsc.br

INE5423 – Banco de Dados I

SQL - DDL e DML

- ▶ **DDL:** possui comandos para *implementar* o modelo lógico
 - ▶ Comando para **criação** das tabelas
 - ▶ Comando para **alteração** das tabelas ✓
 - ▶ Comando para **remoção** das tabelas
 - ▶ **DML:** possui comandos para *manipular* os dados armazenados no Banco de Dados
 - ▶ Comando para **inserção** de dados nas tabelas
 - ▶ Comando para **alteração** de dados nas tabelas ✓
 - ▶ Comando para **remoção** de dados nas tabelas
 - ▶ Comando para **consulta** aos dados nas tabelas
-

Sintaxe básica SQL

- Sintaxe básica de uma instrução de consulta:

SELECT <Lista_de_colunas>

FROM <Lista_de_tabelas>

[WHERE <Condição>]

Cláusula
WHERE é
opcional



- O resultado de uma consulta SQL é uma relação (uma **tabela** - *resultset*)



Instruções básicas SQL

paciente

| <i>codigo</i> | <i>nome</i> | <i>endereco</i> | <i>email</i> | <i>cpf</i> | <i>profissao</i> |
|----------------------|--------------------|------------------------|---------------------|-------------------|-------------------------|
| 1 | Ana | Rua 9n | ana@... | 1234 | advogado |
| 2 | Carla | Av. B | ca@... | 9990 | professor |
| 3 | Joana | Rua A | jo@... | 9090 | professor |
| 4 | Peter | Rua 17 | pet@... | 9191 | empresario |



Tabela
original



Instruções básicas SQL

paciente

| <i>codigo</i> | <i>nome</i> | <i>endereco</i> | <i>email</i> | <i>cpf</i> | <i>profissao</i> |
|----------------------|--------------------|------------------------|---------------------|-------------------|-------------------------|
| 1 | Ana | Rua 9n | ana@... | 1234 | advogado |
| 2 | Carla | Av. B | ca@... | 9990 | professor |
| 3 | Joana | Rua A | jo@... | 9090 | professor |
| 4 | Peter | Rua 17 | pet@... | 9191 | empresario |

→ Tabela original

- O resultado de uma consulta SQL é uma tabela. Exemplo



Instruções básicas SQL

paciente

| <i>codigo</i> | <i>nome</i> | <i>endereco</i> | <i>email</i> | <i>cpf</i> | <i>profissao</i> |
|---------------|-------------|-----------------|--------------|------------|------------------|
| 1 | Ana | Rua 9n | ana@... | 1234 | advogado |
| 2 | Carla | Av. B | ca@... | 9990 | professor |
| 3 | Joana | Rua A | jo@... | 9090 | professor |
| 4 | Peter | Rua 17 | pet@... | 9191 | empresario |

→ Tabela original

- O resultado de uma consulta SQL é uma tabela. Exemplo

```
SELECT nome, email
```

```
FROM Paciente
```

```
WHERE profissão = 'professor'
```



Instruções básicas SQL

paciente

| <i>codigo</i> | <i>nome</i> | <i>endereco</i> | <i>email</i> | <i>cpf</i> | <i>profissao</i> |
|---------------|-------------|-----------------|--------------|------------|------------------|
| 1 | Ana | Rua 9n | ana@... | 1234 | advogado |
| 2 | Carla | Av. B | ca@... | 9990 | professor |
| 3 | Joana | Rua A | jo@... | 9090 | professor |
| 4 | Peter | Rua 17 | pet@... | 9191 | empresario |

Tabela
original

- O resultado de uma consulta SQL é uma tabela. Exemplo

```
SELECT nome, email
```

```
FROM Paciente
```

```
WHERE profissão = 'professor'
```

Tabela
resultante

paciente

| <i>nome</i> | <i>email</i> |
|-------------|--------------|
| Carla | ca@... |
| Joana | jo@... |

Exemplo passo-a-passo

- Exemplo:

“Recuperar o nome dos pacientes, com email ap@abc.de.br” e idade >30



Exemplo passo-a-passo

- Exemplo:

“Recuperar o nome dos pacientes, com email ap@abc.de.br” e idade >30

- Passo 1: Identificar qual (is) a **tabela**(s) envolvida (s)

FROM paciente



Exemplo passo-a-passo

■ Exemplo:

“Recuperar o nome dos pacientes, com email ap@abc.de.br” e idade >30

- Passo 1: Identificar qual (is) a **tabela**(s) envolvida (s)

FROM paciente

- Passo 2: Identificar quais **atributos** devem ser retornados

SELECT nome

FROM paciente



Exemplo passo-a-passo

■ Exemplo:

“Recuperar o nome dos pacientes, com email ap@abc.de.br” e idade >30

- Passo 1: Identificar qual (is) a **tabela**(s) envolvida (s)

FROM paciente

- Passo 2: Identificar quais **atributos** devem ser retornados

SELECT nome

FROM paciente

- Passo 3: Verificar a existência de **filtro**

SELECT nome

FROM paciente

WHERE email = 'ap@abc.de.br' AND idade >30

Exemplos

- Exemplo:

*“Recuperar **todos os dados** dos pacientes, com email ap@a.d.br”*

```
SELECT *  
FROM paciente  
WHERE email = 'ap@a.d.br'
```

*“Recuperar **o nome** dos pacientes, com email ap@abc.de.br”*

```
SELECT nome  
FROM paciente  
WHERE email = 'ap@abc.de.br'
```



Exemplos

- Exemplo:

*“Recuperar **nome**, **endereco** e **email** dos pacientes que não sejam engenheiros*

```
SELECT nome, endereco, email
FROM paciente
WHERE profissao <> 'engenheiro'
```

| <i>nome</i> | <i>endereco</i> | <i>email</i> |
|-------------|-----------------|--------------|
| Ana | Rua 9n | ana@... |
| Carla | Av. B | ca@... |
| Joana | Rua A | jo@... |
| Peter | Rua 17 | pet@... |

Instruções básicas SQL

■ Uso da expressão WHERE

✓ Operadores de comparação:

=, <>, <, <=, >, >=

<coluna> **BETWEEN** 'valor' **AND** 'valor'

<coluna> **IN** ('valor1', ..., 'valorn')

<coluna> **LIKE** '%banco de %'

<coluna> **LIKE** '19_ _'

<coluna> **IS** [NOT] NULL

"_" = um caracter

"%" = sequência de caracteres

✓ Operadores lógicos:

AND

OR



NULL

- ▶ Valores NULL indicam inexistência de valor
- ▶ Não indica vazio!!

Pessoa

| Codigo | Nome | Salário | Endereco |
|---------------|-------------|----------------|-----------------|
| 1 | Ana | 100 | |
| 2 | Juca | 0 | Av. Brasil |

Pessoa

| Codigo | Nome | Salário | Endereco |
|---------------|-------------|----------------|-----------------|
| 1 | Ana | 100 | NULL |
| 2 | Juca | NULL | Av. Brasil |



NULL

- ▶ Valores NULL indicam inexistência de valor
- ▶ Não indica vazio!!

Espaço e NULL:
Coisas diferentes!!

Pessoa

| Codigo | Nome | Salário | Endereço |
|---------------|-------------|----------------|-----------------|
| 1 | Ana | 100 | |
| 2 | Juca | 0 | Av. Brasil |

Pessoa

| Codigo | Nome | Salário | Endereço |
|---------------|-------------|----------------|-----------------|
| 1 | Ana | 100 | NULL |
| 2 | Juca | NULL | Av. Brasil |

NULL

- ▶ Valores NULL indicam inexistência de valor
- ▶ Não indica vazio!!

Pessoa

| Codigo | Nome | Salário | Endereco |
|---------------|-------------|----------------|-----------------|
| 1 | Ana | 100 | |
| 2 | Juca | 0 | Av. Brasil |

Pessoa

| Codigo | Nome | Salário | Endereco |
|---------------|-------------|----------------|-----------------|
| 1 | Ana | 100 | NULL |
| 2 | Juca | NULL | Av. Brasil |

Zero e NULL:
Coisas diferentes!!

Exemplos

- Recuperar todos os dados dos pacientes cujo nome **começa** com A

```
SELECT *  
FROM paciente  
WHERE nome LIKE 'A%'
```

- Recuperar dados dos paciente cujo nome **contenha** 'Silva'

```
SELECT *  
FROM paciente  
WHERE nome LIKE '%Silva%'
```



Exemplos

- Pacientes cuja idade esteja entre 20 e 30

SELECT *

FROM paciente

WHERE idade BETWEEN 20 and 30

OU **WHERE** idade >= 20 AND idade <=30

WHERE idade >= 20 AND <=30 /*ERRADO*/

- Pacientes cuja idade esteja entre 20 e 30 e cuja cidade seja diferente de “Blumenau”

SELECT *

FROM paciente

WHERE (idade BETWEEN 20 and 30) AND cidade <>

▶ 'Blumenau'

Exemplos

- Pacientes cuja idade seja maior do que 20 e cuja cidade seja igual a Blumenau ou Joinville.

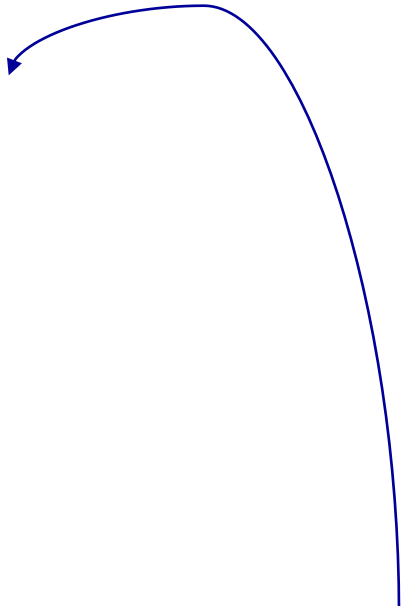
```
SELECT *  
FROM paciente  
WHERE idade > 20 AND  
      (cidade = 'Blumenau' OR  
       cidade = 'Joinville')
```

- Idade dos pacientes

```
SELECT idade  
FROM paciente
```

- Idade dos pacientes

```
SELECT DISTINCT idade  
FROM paciente
```



Sempre que houver o uso de **AND** e **OR** juntos, utilizar parênteses para indica a ordem de execução (como em operações Matemáticas)

Exemplos

- Pacientes cuja idade seja maior do que 20 e cuja cidade seja igual a Blumenau ou Joinville.

```
SELECT *  
FROM paciente  
WHERE idade > 20 AND  
      (cidade = 'Blumenau' OR  
       cidade = 'Joinville')
```

- Alternativa ao OR

```
SELECT *  
FROM paciente  
WHERE idade > 20 AND cidade IN ('Blumenau', 'Joinville')
```



Renomeação de tabelas

- É possível utilizar apelidos para **nomes de tabelas**.
 - ✓ Útil nos casos onde existam muitas tabelas na cláusula FROM, e principalmente quando o nome da tabela é muito grande

```
SELECT p.nome, p.idade
FROM paciente p
WHERE p.nome LIKE 'A%'
```



Renomeação de atributos

- É possível utilizar apelidos para **nomes de atributos do resultado**.
 - ✓ Útil nos casos onde existam muitas tabelas na cláusula FROM, e principalmente quando o nome da tabela é muito grande

```
SELECT p.nome, p.idade AS idade_pac
FROM paciente p
WHERE p.nome LIKE 'A%'
```



Exercícios

