

# **Bacharelado em Sistemas de Informação**

## **Banco de Dados Aula 07 SQL – DML Parte 2**

Dr. Diego Buchinger  
[diego.buchinger@udesc.br](mailto:diego.buchinger@udesc.br)

## SQL

### Structured Query Language

---

DML – *Data Manipulation Language*  
*Consultas*

# MySQL – Consultas Básicas

- Para consultar dados de uma tabela utiliza-se

```
SELECT lista_de_colunas  
FROM tabela  
[WHERE condição];
```

- ☐ *lista\_de\_colunas* pode ser substituída por um asterisco (\*), que representa todos os atributos da tabela

- Exemplos:

```
SELECT * FROM pacientes;
```

```
SELECT * FROM pacientes  
WHERE idade > 18;
```

```
SELECT cpf, nome  
FROM pacientes;
```

```
SELECT cpf, nome  
FROM pacientes  
WHERE idade > 18;
```

# MySQL – Consultas Básicas

- Note que não há eliminação automática de duplicatas
  - ❑ Tabela  $\equiv$  coleção
  - ❑ Para eliminar duplicatas deve-se usar o termo *distinct*
- Exemplos:

```
SELECT DISTINCT doenca FROM consultas;
```

- É possível renomear os campos/colunas do seguinte modo:

```
SELECT codp AS codigo_paciente,  
      codm AS codigo_medico, data  
FROM consultas;
```

# MySQL – Consultas Básicas

- É possível utilizar operadores aritméticos e funções
  - ❑ Quantos grupos de 5 leitos podem ser formados em cada ambulatório?

```
SELECT nroa, capacidade/5 AS cap5  
FROM ambulatorios;
```

- ❑ Qual o salário líquido dos funcionários sabendo que há um desconto de 12,63% sobre o salário base?

```
SELECT nome, ROUND(salario * 0.8737, 2) AS  
    salario_liquido FROM funcionarios;
```

Função **ROUND**: parâmetros  
(valor : decimal, casas\_decimais : int | decimal)

# MySQL – Consultas Básicas

- A operação de seleção também permite o uso de funções de agregação / agrupamento:

- ☐ **COUNT**: contador de ocorrências [registros ou atributos]

**OBS:** conta os valores não nulos

```
SELECT COUNT(*) FROM medicos  
WHERE especialidade = 'ortopedia';
```

- ☐ **MAX / MIN**: valor máximo / mínimo de um atributo

```
SELECT MAX(salario) AS maior_salario FROM funcionários;
```

- ☐ **SUM**: soma os valores de um dado atributo
  - Qual é o gasto total com a folha de pagamento dos funcionários?

```
SELECT SUM(salario) FROM funcionários;
```

- ☐ **AVG**: contabiliza a média dos valores de um dado atributo
  - Qual é a média de idade dos funcionários de Florianópolis?

```
SELECT AVG(salario) FROM funcionários  
WHERE cidade = 'Florianópolis';
```

# MySQL – Consultas Básicas

- Pode-se misturar funções de agregação com `distinct`
  - ❑ Qual a quantidade de cargos distintos dos funcionários?

```
SELECT COUNT(DISTINCT especialidade)
FROM medicos;
```

- Não podem ser adicionados campos que não fazem parte da função de agregação
  - ❑ Qual a quantidade de cargos de cada cargo?

É possível fazer essa consulta.  
Veremos como depois

```
SELECT cargo, COUNT(cargo) FROM funcionarios;
```

- Pode-se realizar *casting* de tipos usando: `campo::tipo`

```
SELECT nome, cpf::text, idade::numeric(3,1)
FROM funcionarios;
```

# MySQL – Consultas Básicas

- Como procurar por valores nulos ou não nulos?
  - ❑ Deve-se utilizar a cláusula: `IS NULL` ou `IS NOT NULL`

```
SELECT cpf, nome  
FROM medicos WHERE nroa IS NULL
```

- Como procurar por intervalos de valores?
  - ❑ Utilizando `>`, `>=`, `AND`, `<` e `<=`
  - ❑ Utilizar a cláusula: `[NOT] BETWEEN valor1 AND valor2`

```
SELECT * FROM consultas  
WHERE hora BETWEEN '13:00' AND '18:00'
```



# MySQL – Consultas Básicas

- Procurar por existência de um valor em uma coleção

❑ Deve-se utilizar a cláusula: [NOT] IN

```
SELECT * FROM medicos
WHERE especialidade IN ('ortopedia', 'traumatologia');
```

- ❑ É possível combinar com subconsultas!

```
SELECT codm, codp, data FROM consultas
WHERE codm IN (
    SELECT codm FROM médicos
    WHERE idade > 40
);
```

Note que esta consulta irá retornar uma única coluna contendo entre zero a n registros de codm (ex: 1010, 2020, 3030)

# MySQL – Consultas Básicas

- Como procurar por padrões textuais?
  - ❑ Deve-se utilizar a cláusula: `[NOT] LIKE`
  - ❑ Esta cláusula entende os seguintes padrões:
    - `%` casamento com qualquer cadeia de caracteres
    - `_` casamento com um único caractere
    - `[a-d]` casamento com qualquer caractere entre as letras apresentadas (SQL-Server)
- ❖ Buscar médicos com nome iniciando por 'M'
- ❖ Buscar médicos com nome composto por três letras + sobrenome de qualquer tamanho

```
SELECT * FROM medicos  
WHERE nome LIKE 'M%'
```

```
SELECT * FROM medicos  
WHERE nome LIKE '_____%'
```

# MySQL – Consultas Básicas

- Como procurar por padrões textuais?
  - ❑ Deve-se utilizar a cláusula: `[NOT] LIKE`
  - ❑ Esta cláusula entende os seguintes padrões:
    - `%` casamento com qualquer cadeia de caracteres
    - `_` casamento com um único caractere
    - `[a-d]` casamento com qualquer caractere entre as letras apresentadas (SQL-Server)

- ❖ Buscar por consultas marcadas para o mês de julho (*casting*)

```
SELECT * FROM consultas
WHERE data::text LIKE '%-07-%';
```

**OBS:** padrão dos campos de data: 'aaaa-mm-dd'

- ❖ Buscar por pacientes cujo CPF termina com 20000 ou 30000

```
SELECT * FROM pacientes
WHERE cpf::text LIKE '%20000'
OR cpf::text LIKE '%30000';
```

# MySQL – Consultas Básicas

- Algumas funções básicas (variam entre SGBDs)
  - ❑ Verificar tamanho (# de caracteres) de um campo textual
    - ❖ Buscar cada doença e seu tamanho em letras

```
SELECT doença, length(doença) FROM consultas;
```

- ❑ Resgatar uma substring – `substring(texto, começo, #chars)`
  - ❖ Buscar apenas os meses da data de cada consulta

```
SELECT substring(data::text, 5, 2) FROM consultas;
```

- ❑ Alterar um texto específico em um campo textual  
`replace(texto, padrão_a_procurar, texto_substituto)`

```
SELECT replace(data::text, '-06-', '-07-')  
FROM consultas WHERE data::text LIKE '2024%';
```

# MySQL – Consultas Básicas

- MySQL implementa operações de União
  - ❑ Uma união “une” os registros retornados entre duas consultas
  - ❑ IMPORTANTE: a ordem dos tipos dos campos consultados precisam ser os mesmos ou compatíveis

```
SQL-Query-1    UNION    SQL-Query-2
```

- ❖ Buscar o nome e o CPF dos médicos e funcionários em uma única consulta

```
SELECT cpf, nome FROM medicos
UNION
SELECT cpf, nome FROM funcionarios
```

**OBS:** note que a ordem das colunas possuem os mesmos tipos de dados nas duas consultas realizadas

# Exercícios

---

1. Realize as seguintes consultas no BD:
  - a. Buscar o nome e o CPF dos médicos com menos de 40 anos ou com especialidade diferente de ortopedia
  - b. Buscar todos os dados das consultas marcadas no período da tarde (das 12h às 18h) após o dia 18/06/2024
  - c. Buscar o nome e a idade dos pacientes que não residem em Florianópolis
  - d. Buscar a data e a hora das consultas marcadas antes do dia 13/06/2024 e depois do dia 19/06/2024
  - e. Buscar o nome e a idade (em meses) dos pacientes
  - f. Buscar o menor e o maior salário dos funcionários de *Florianopolis*

# Exercícios

---

## 1. Realize as seguintes consultas no BD:

- g. Qual o horário da última consulta marcada para o dia 14/06/2024?
- h. Qual a média de idade dos médicos e o total de ambulatórios atendidos por eles?
- i. Buscar o código, o nome e o salário líquido dos funcionários que recebem mais do que \$ 1200. Para quem ganha acima de \$1200, há um desconto de 12% do salário bruto.
- j. Buscar o nome dos funcionários que terminam com a letra 'a'
- k. Buscar o nome e idade dos funcionários que possuem o número 6 em seu CPF
- l. Em quais cidades residem os funcionários e médicos da clínica? Mostrar o nome e a cidade (nesta ordem)