

Banco de Dados

Prof. Gale

[galesandro.capovilla@essj.com.br](mailto:galesandro.capovilla@essj.com.br)

# A LINGUAGEM SQL (*STRUCTURED QUERY LANGUAGE*)

- O SQL é uma linguagem estruturada para manipulação de dados. É padronizada para os bancos de dados relacionais, mas cada gerenciador pode possuir uma extensão própria dessa linguagem.

# Tipos de Campos

- **Alfanuméricos** → Contém cifras e letras. Apresentam uma longitude limitada (255 caracteres)
- **Numéricos** → Existem de vários tipos, principalmente, inteiros (sem decimais) e reais (com decimais).
- **Booleanos** → Possuem duas formas: Verdadeiro e falso (Sim ou Não)
- **Datas** → Armazenam datas facilitando posteriormente sua exploração. Armazenar datas desta forma possibilita ordenar os registros por datas ou calcular os dias entre uma data e outra.
- **Memos** → São campos alfanuméricos de longitude ilimitada. Apresentam o inconveniente de não poder ser indexados.
- **Auto-incrementáveis** → São campos numéricos inteiros que incrementam em uma unidade seu valor para cada registro incorporado. Sua utilidade é mais que evidente: Servir de identificador já que são exclusivos de um registro.

## ■ **COMANDO SELECT**

- Através do comando select, listamos os dados existentes no banco, de acordo com os critérios desejados:

**SELECT nome, telefone FROM cliente**

nome	telefone
João	445-0988
Alberto	465-9887
Maria	789-9877
Sônia	(null)
(4 row(s) affected)	



A utilização da cláusula WHERE faz com que o uso do comando SELECT seja dos mais freqüentes no dia a dia, pois através dela poderemos especificar condições de busca, as quais determinarão a quantidade de informações retornadas pelo servidor, ou, muitas vezes, trarão exatamente o que precisamos. Esta última característica, de obtermos exata e somente aquilo que necessitamos é que faz a grande diferença entre um servidor de arquivos, que envia pela rede o arquivo todo, e um gerenciador de bancos de dados, que envia somente o suficiente.

```
SELECT * FROM cliente WHERE telefone LIKE '4%'
```

# Operadores e cláusulas de Comparação

= igual  
<> diferente  
!= diferente  
> maior que  
> não maior (menor ou igual)  
< menor que  
!< não menor (maior ou igual)  
>= maior ou igual  
<= menor ou igual  
\* todos

**ORDER BY** define a ordem

**ORDER BY DESC** em ordem decrescente

**WHERE** onde

**BETWEEN** entre. Pertence ao intervalo declarado.

**NOT BETWEEN** não pertence ao intervalo declarado.

**GROUP BY** agrupar

**IS NULL** testa valores nulos. Verifica, por exemplo, se colunas não contém nenhum valor armazenado. Com conteúdo nulo.

**IS NOT NULL** testa valores não nulos. Sem conteúdo nulo.

**LIKE** o predicado LIKE procura por strings que se encontram dentro de um determinado padrão. Este predicado só pode ser usado com tipos de dados CHAR ou VARCHAR.

**IN** permite comparar o valor de uma coluna com um conjunto de valores. Normalmente, utilizamos o IN para substituir uma série de comparações seguidas da cláusula OR.

**NOT IN** Não pertence ao conjunto declarado

# Exemplos

- `Select * from nome_tabela;`

Quando desejo exibir todos os dados  
da tabela com todos os campos

- `Select nome,endereco from tabela01;`

Quando desejo exibir somente os campos  
nome e endereco da tabela toda

- `Select nome, idade where idade >= 18;`

Quando desejo exibir somente os campos  
nome e idade da tabela porém somente as idades superiores a 18 anos