Introdução às Consultas SQL

Mentor: Carlos Júnior

1 - Consultas SQL

- O que são?
- Quais são seus tipos?

2 - Consultas de Seleção

- O que são?
- Como funcionam?

3 - Consultas de Atualização

- O que são?
- Como funcionam?

4 - Consultas de Inserção

- O que são?
- Como funcionam?

5 - Consultas de Exclusão

- O que são?
- Como funcionam?

6 - Consultas Entre Tabelas

- O que são?
- Como funcionam?

7 - Group by, Order by, Offset e Limit

- O que são?
- Como funcionam?

Consultas SQL

Instruções que usam a linguagem SQL para recuperar, modificar ou excluir dados de um banco de dados relacional. São compostas de uma série de palavras-chave e operadores que instruem o banco de dados a realizar uma determinada tarefa. Existem quatro tipos principais de consultas SQL:

- **Consultas de seleção:** Recuperam dados de uma tabela.
- **Consultas de atualização:** Modificam dados em uma tabela.
- **Consultas de inserção:** Inserem dados em uma tabela.
- **Consultas de exclusão:** Excluem dados de uma tabela.

Consultas de Seleção

Sintaxe básica:

```
SELECT * -- todas as colunas
FROM <tbl_name>
[WHERE <condition>]
```

```
SELECT <col_name>, <col_name> -- apenas essas duas colunas
FROM <tbl_name>
[WHERE <condition>]
```

Consultas de Seleção

Apelidando coluna:

```
SELECT <col_name> AS <aliase>, <col_name> AS <aliase>
FROM <tbl_name>
```

Apelidando tabela:

```
SELECT *
FROM <tbl_name> AS <aliase>
```

Consultas de Seleção

Funções de agregação:

```
SELECT FUNC(<col_name>)
FROM <tbl_name>
```

sendo FUNC() a função de agregação que pode ser:

- **COUNT:** Conta o número de linhas de uma coluna
- **SUM:** Calcula a soma dos valores de uma coluna
- **AVG:** Calcula a média dos valores de uma coluna
- **MIN:** Retorna o valor mínimo de uma coluna
- **MAX:** Retorna o valor máximo de uma coluna

Consultas de Seleção

Adicionando condições:

```
SELECT <col_name> AS <aliase>, <col_name> AS <aliase>
FROM <tbl_name>
WHERE <col_name> <logic_op> <value>
```

sendo <logic_op> um operador lógico que pode ser, dentre outros:

- Igual: =
- Diferente: <>
- Maior que: >
- Menor que: <
- Maior ou igual que: >=
- Menor ou igual que: <=

Consultas de Atualização

Sintaxe básica:

```
UPDATE <tbl_name>
SET <col_name> = <value>, <col_name> = <value>
[WHERE <condition>]
```

o operador de atribuição = poderá ser acrescentado de outros operadores como no exemplo a seguir:

```
UPDATE <tbl_name>
SET <col_name> = <col_name> + <value>, -- somando <value> ao valor existente
    <col_name> = <col_name> - <value>, -- subtraindo <value> ao valor existente
    <col_name> = <col_name> * <value>, -- multiplicando <value> ao valor existente
    <col_name> = <col_name> / <value> -- dividindo <value> ao valor existente
[WHERE <condition>]
```

Consultas de Inserção

Sintaxe básica:

```
INSERT INTO <tb_name> (<col_name>, <col_name>)
VALUES (<value>, <value>), (<value>, <value>)
```

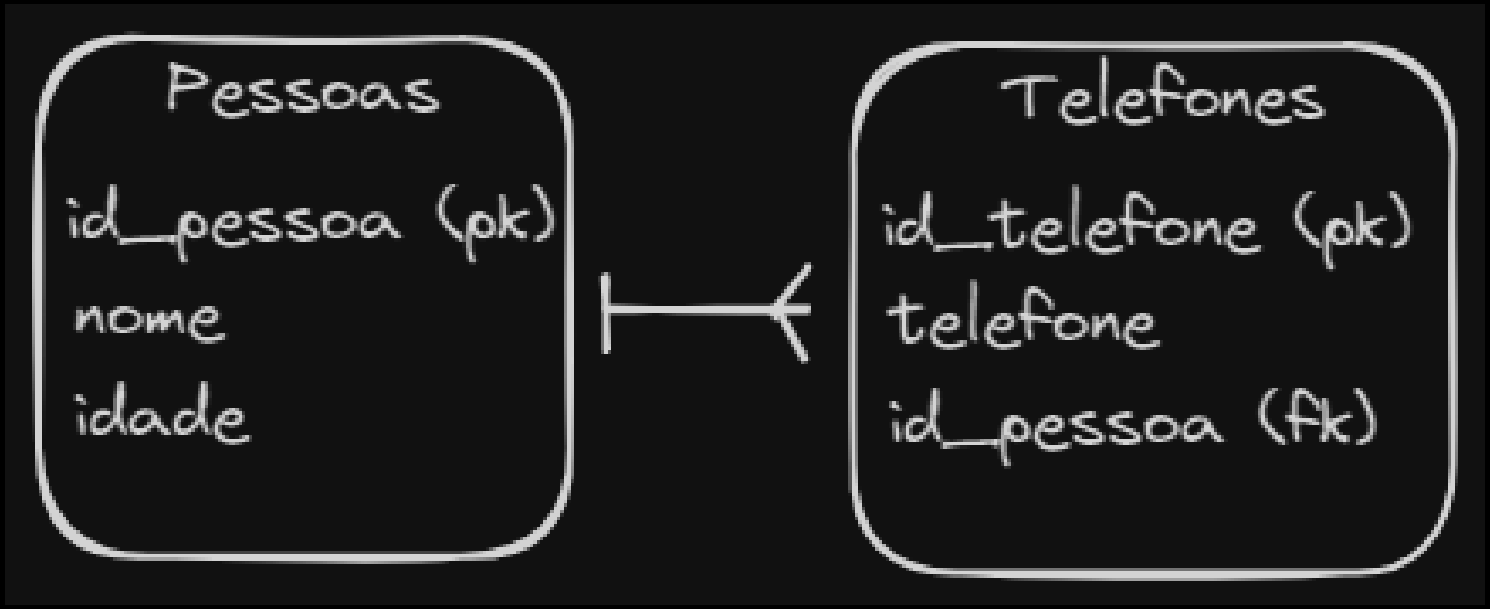
Consultas de Exclusão

Sintaxe básica:

```
DELETE FROM <tb_name>
[WHERE <condition>]
```

Consultas Entre Tabelas

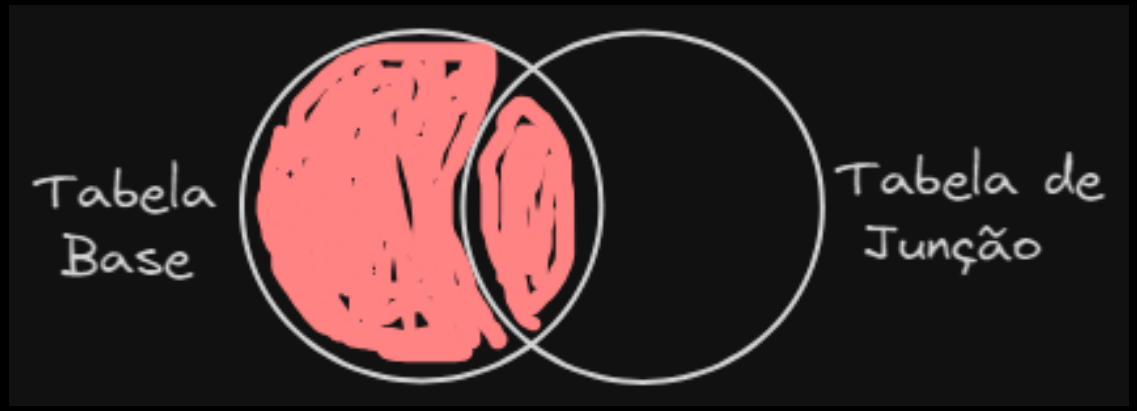
Até o momento vimos operações feitas em uma mesma tabela. E se precisarmos de informações que necessitam da combinação de duas ou mais tabelas? Por exemplo, em um banco de dados com as tabelas descritas abaixo, como eu poderia mostrar todos os dados de uma pessoa, incluindo os números?



Consultas Entre Tabelas

Sintaxe básica (Left Join):

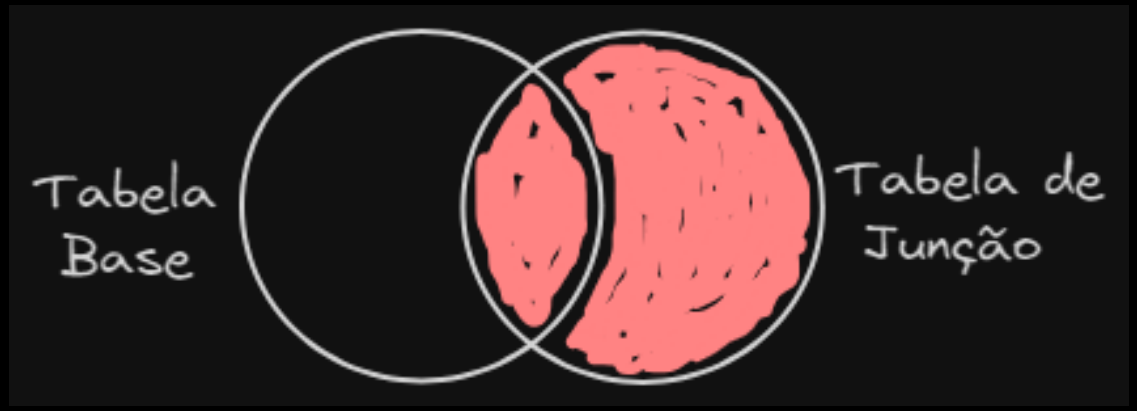
```
SELECT *
FROM <tbl_name> -- tabela base
LEFT JOIN <tbl_name> -- tabela de juncao
ON <tbl_name>.<col_name> = <tbl_name>.<col_name> -- o que sera comparado
```



Consultas Entre Tabelas

Sintaxe básica (Right Join):

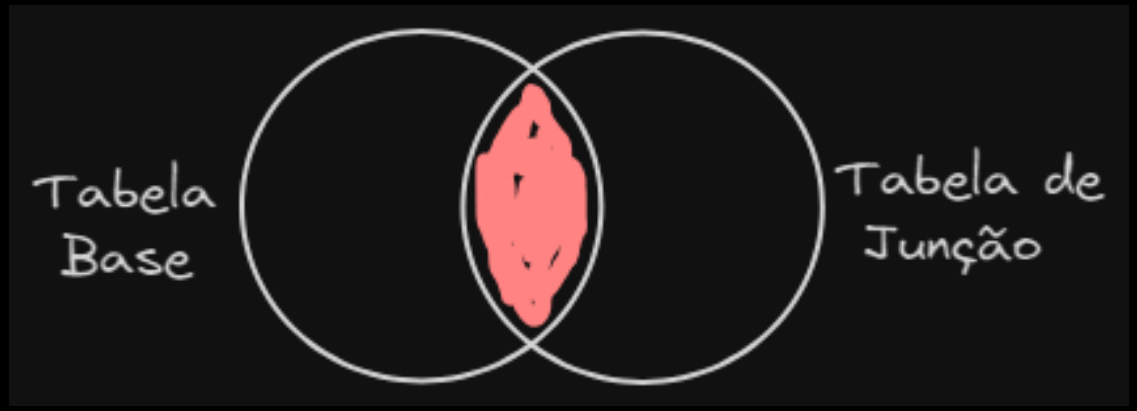
```
SELECT *
FROM <tbl_name> -- tabela base
RIGHT JOIN <tbl_name> -- tabela de juncao
ON <tbl_name>.<col_name> = <tbl_name>.<col_name> -- o que sera comparado
```



Consultas Entre Tabelas

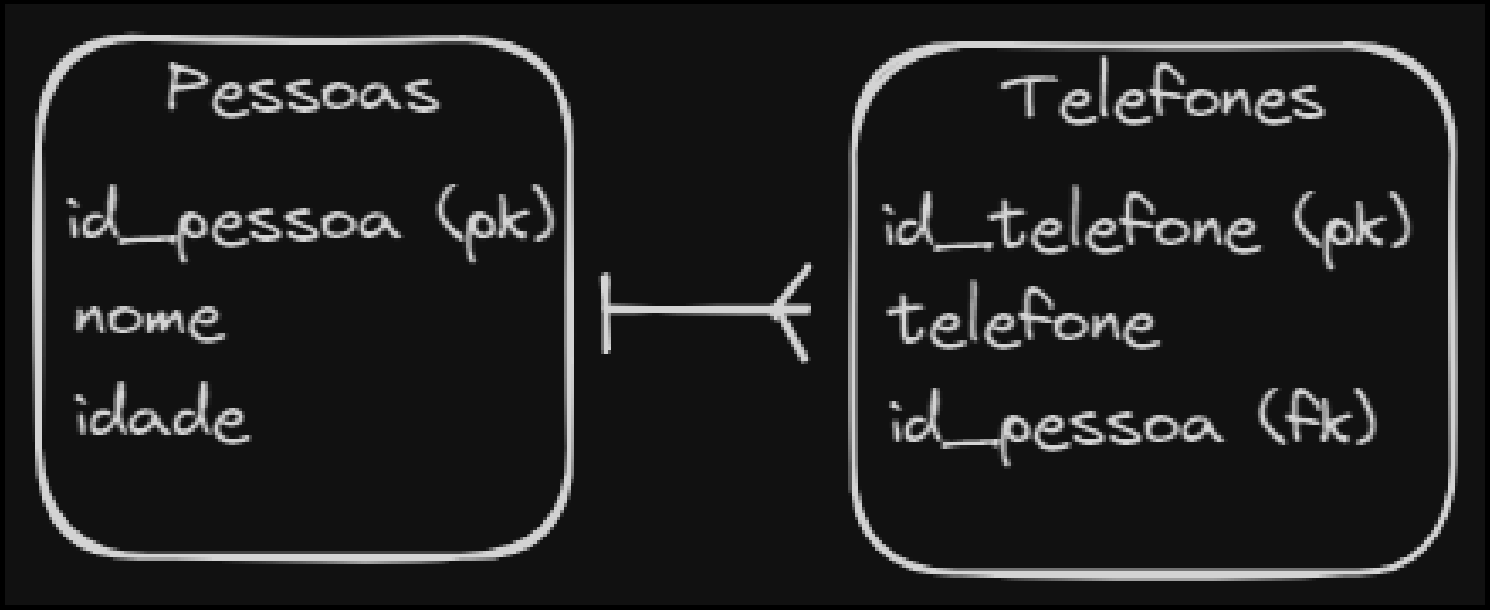
Sintaxe básica (Inner Join):

```
SELECT *
FROM <tbl_name> -- tabela base
INNER JOIN <tbl_name> -- tabela de juncao
ON <tbl_name>.<col_name> = <tbl_name>.<col_name> -- o que sera comparado
```



Group by, Order by, Offset e Limit

Agora imagina que você precisa fazer uma consulta que mostre quantos telefones cada usuário possui. Além disso, a consulta deve mostrar as pessoas ordenadas **decrescentemente** pelo número de telefones. Como poderíamos fazer isso?



Group by, Order by, Offset e Limit

Sintaxe básica (Group By):

```
SELECT *
FROM <tbl_name>
GROUP BY <col_name>, <col_name> -- colunas que serao agrupadas
```

Sintaxe básica (Order By):

```
SELECT *
FROM <tbl_name>
ORDER BY <col_name> [<ordering>], <col_name> [<ordering>] -- colunas que serao ordenadas
```

onde <ordering> pode ser ASC (crescente), que é o padrão, e DESC (decrescente).

Group by, Order by, Offset e Limit

Sintaxe básica (Offset):

```
SELECT *
FROM <tbl_name>
OFFSET <num_lines> -- numero de linhas que serao saltadas
```

Sintaxe básica (Limit):

```
SELECT *
FROM <tbl_name>
LIMIT <num_lines> -- numero de linhas que serao mostradas
```

Sua vez!

Utilize a mesma base de dados criada no exercício da aula anterior (a que possui duas tabelas: Pessoas e Telefones).

1. **Use uma consulta de inserção para inserir os seguintes registros:**
    - Calisto Silva, 30 anos, telefones: 40028922, 40068966 e 40078977
    - Allan Silva, 25, telefone: 40038933
    - David Dauli, 28, telefone: 40048944
    - Adailton Aveiro, 17, telefones: 40058955 e 40088988
2. **Use uma consulta de seleção para recuperar todas as linhas da tabela pessoas.**

Sua vez!

3. Use uma consulta de seleção para recuperar apenas os nomes das pessoas.
4. Use uma consulta de seleção para recuperar a soma, a média, a menor e a maior idade das pessoas. Nomeie as colunas resultantes da consulta de forma que fique claro o que são seus valores.
5. Use uma consulta de seleção para recuperar todas as pessoas cujo nome começa com a letra 'A'.
6. Use uma consulta de seleção para recuperar todos as pessoas cujo nome tenha a substring 'av', não importando se as letras estarão em caixa alta ou em caixa baixa.
7. Use uma consulta de seleção para recuperar todas as pessoas cuja idade é maior ou igual a 25.

Sua vez!

9. **Use uma consulta de atualização para adicionar 5 anos à idade de todas as pessoas.**
10. **Use uma consulta de atualização para adicionar 2 anos à idade de uma pessoa específica. Justifique o atributo que escolheu para selecionar a pessoa em questão.**
11. **Use uma consulta de exclusão para remover uma pessoa específica. Justifique o atributo que escolheu para selecionar a pessoa em questão.**
12. **Crie uma consulta de seleção para recuperar o nome, idade e a quantidade de números de telefones de todas as pessoas que possuem mais de 24 anos. O resultado também deve ser mostrado de maneira decrescente por idade.**

Sua vez!

13. Crie uma consulta de seleção que mostre os registros da tabela Pessoas de maneira paginada. Cada página deve mostrar apenas 2 registros. Tente pensar em uma lógica imaginando que você receberá um número que representa a página e você só poderá alterar esse número na consulta.
14. Pesquise sobre a funcionalidade HAVING que pode ser utilizada em parceria com o GROUP BY. Em seguida, explique o que essa palavra reservada faz, qual a diferença dela para o WHERE e exemplifique uma situação que ela pode ser utilizada.