



UNIPÊ

Centro Universitário
de João Pessoa



Workshop Fábrica de Software 2024.1

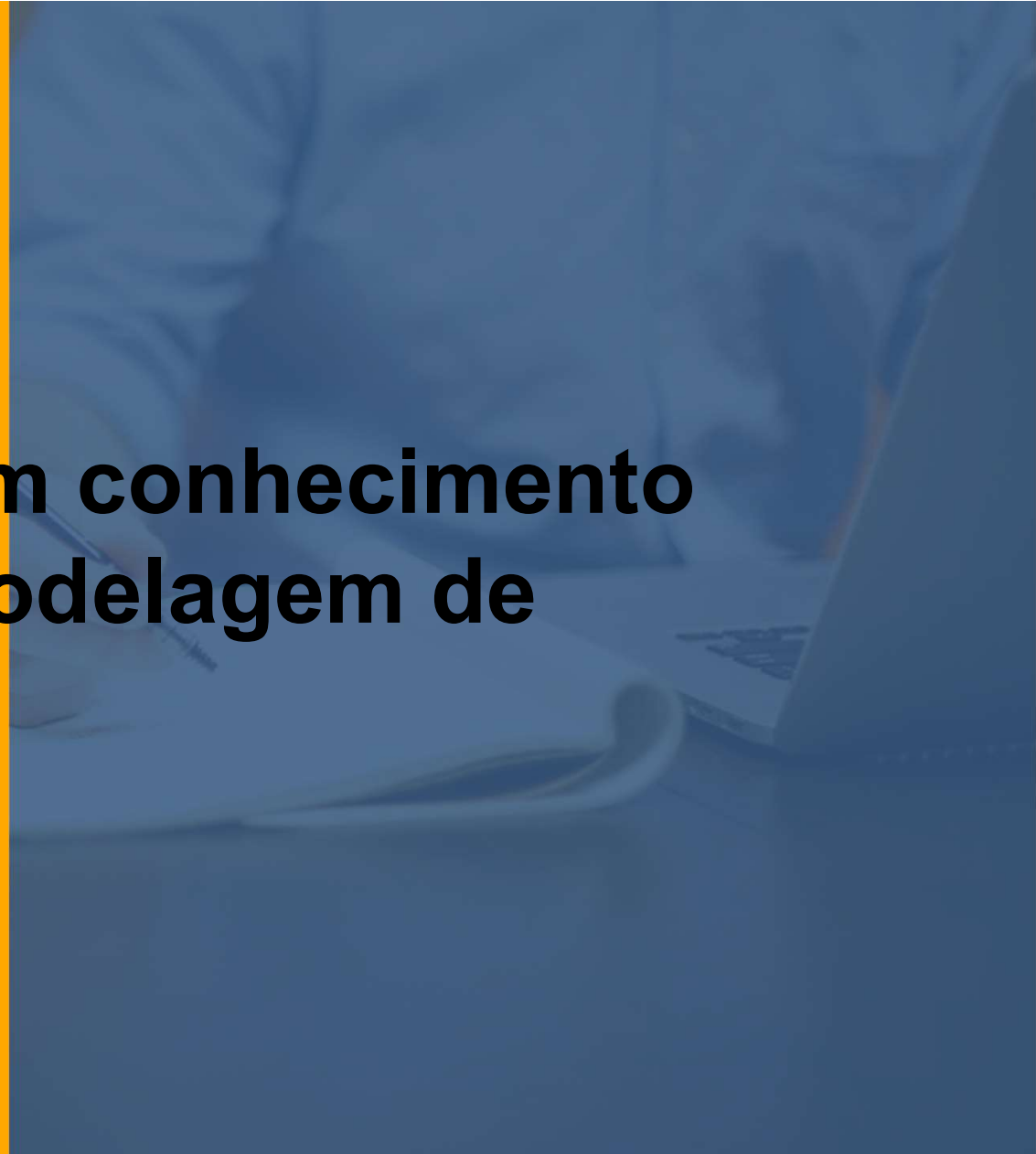


The background of the slide features a photograph of a person's back, seen from behind, in what appears to be a workshop or office setting. The person is wearing a light-colored sweater. The background is filled with various papers, diagrams, and sketches pinned to a wall. A semi-transparent blue overlay covers the entire image, and the text is placed on top of this overlay.

Olá!

**Sou:
Daniel Warella Pitsch**

**Já tem algum conhecimento
prévio de modelagem de
dados?**



**Já tem algum conhecimento
prévio de banco de dados?**



Lembrete

Me ajudem a falar
mais devagar!

Falar sobre tipos

Tempo por turma

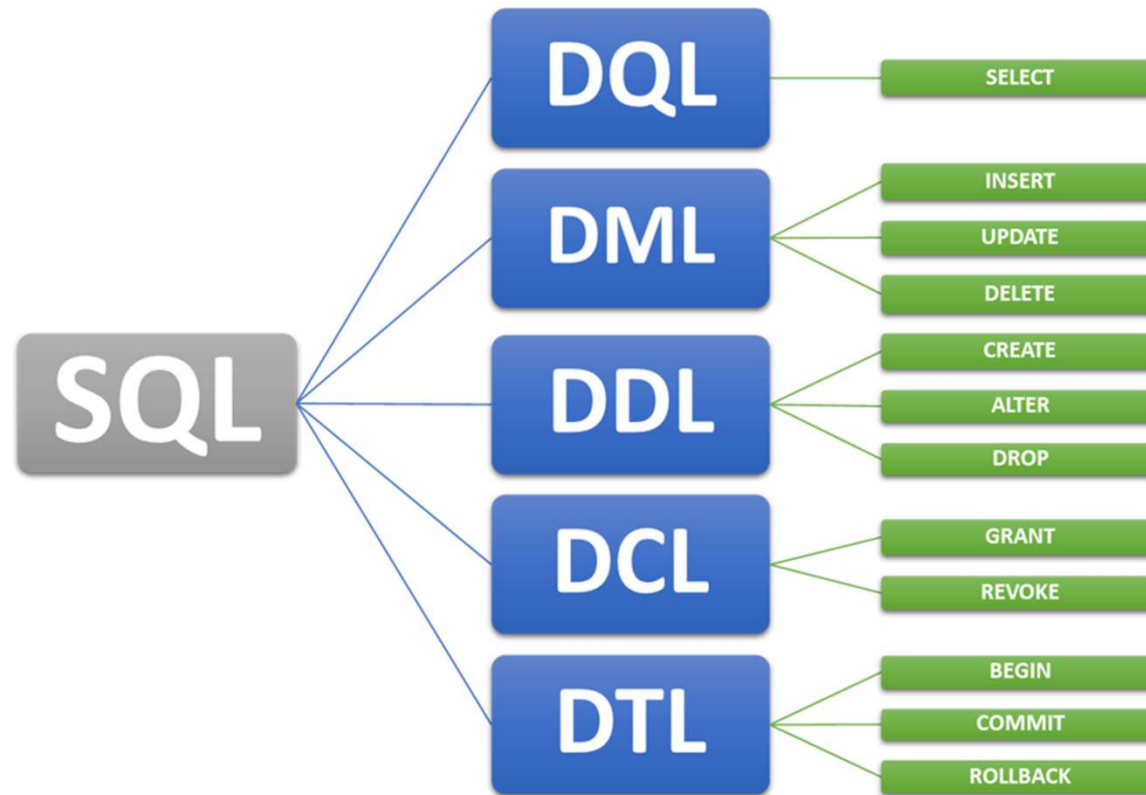
Ministrar workshop

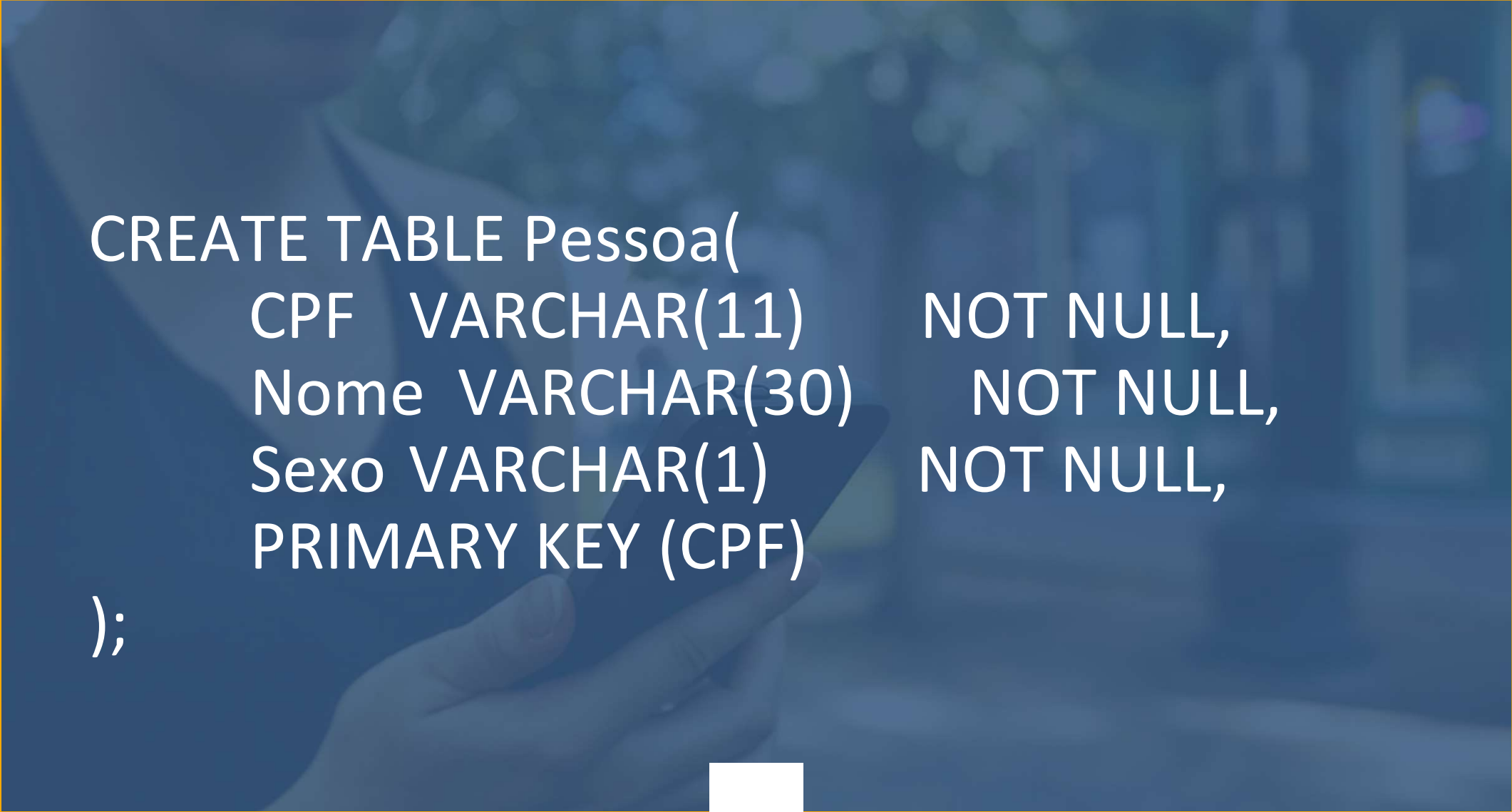
Ferramentas:



SQL

-
- significa ***Structured Query Language - Linguagem Estruturada de Pesquisa.***





```
CREATE TABLE Pessoa(  
    CPF VARCHAR(11) NOT NULL,  
    Nome VARCHAR(30) NOT NULL,  
    Sexo VARCHAR(1) NOT NULL,  
    PRIMARY KEY (CPF)  
);
```

Exemplo:

Departamento

codigo	nome
2B	Finanças
3C	Marketing

Funcionário

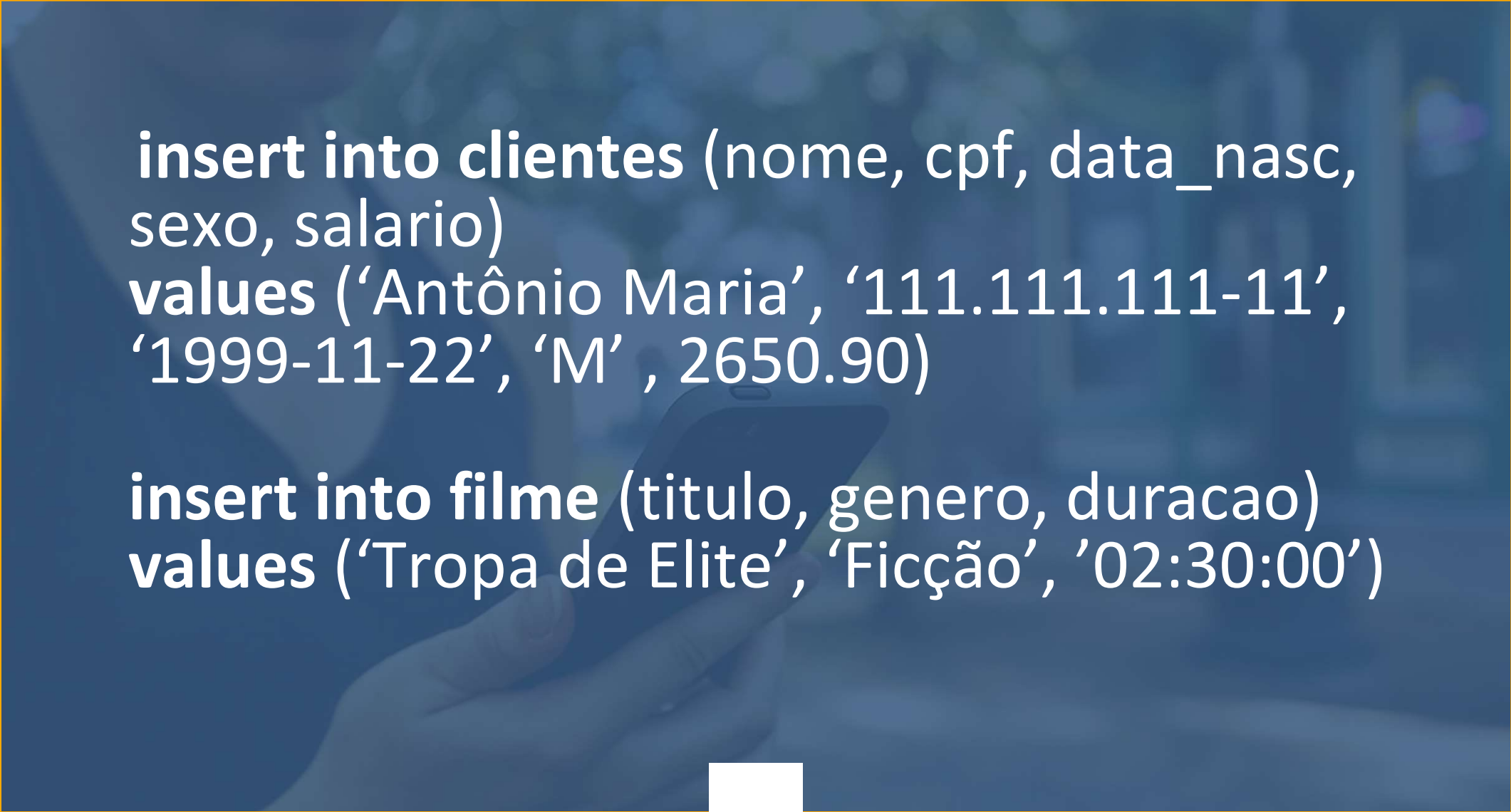
codigo	Nome	CPF	salario	codDep
1	Ana	111111111-11	3000,00	
2	Maria	222222222-22	5000,00	2B
3	João	333333333-33	2500,00	
4	José	444444444-44	3500,00	3C

ALTER TABLE funcionário ADD PRIMARY KEY (codigo)
// adiciona chave primária codigo na tabela funcionário.

ALTER TABLE funcionário ADD COLUMN categoria VARCHAR(10)
DEFAULT 'Júnior' NOT NULL
// adiciona a coluna categoria com o tipo varchar(10) e valor default
'Junior' na tabela funcionário


ALTER TABLE funcionario DROP COLUMN salario
// exclui a coluna salario da tabela funcionário.

ALTER TABLE funcionario MODIFY categoria CHAR(25)
// modifica o tipo da coluna categoria para char(25) na tabela
funcionário.



```
insert into clientes (nome, cpf, data_nasc,  
sexo, salario)  
values ('Antônio Maria', '111.111.111-11',  
'1999-11-22', 'M' , 2650.90)
```

```
insert into filme (titulo, genero, duracao)  
values ('Tropa de Elite', 'Ficção', '02:30:00')
```



```
UPDATE pessoa  
SET idade = 20  
WHERE nome = 'Alice Maria'
```

```
UPDATE cliente  
    SET salario = 5800  
    WHERE codigo = 11
```



DELETE FROM pessoa
WHERE nome = 'alibaba'

Todas as pessoas cujos nomes são
“alibaba” serão retiradas da tabela
pessoa.



```
SELECT lista_de_colunas FROM tabela
```

```
SELECT* from cliente;
```

```
SELECT codigo, nome FROM tb_fabricante;
```

```
SELECT codigo AS Produto, preco AS  
Preço_Produto FROM tb_produto;
```

Exibir o id, o nome e o bairro de todos o clientes ordenando pelo nome do cliente em ordem ascendente.

```
SELECT id, nome, bairro FROM tb_cliente ORDER BY nome;
```

```
SELECT formaPagamento, idCliente, dataEmbarque FROM tb_pedido ORDER BY dataEmbarque DESC;
```


Sintaxe:

SELECT lista_de_colunas **FROM** tabela **WHERE** coluna **LIKE** expressão

Exemplo : Mostrar todos os clientes cujos nomes começam a letra “A”.

```
SELECT * FROM tb_cliente WHERE nome LIKE 'A%';
```

Exemplo : Mostrar todos os produtos cujos nomes contém a sequência “or”.

```
SELECT * FROM tb_produto WHERE nome LIKE  
'%or%';
```

Exemplo : Mostrar todos os fabricantes cujos nomes começam com a letra “S” e possuem exatamente 4 letras.

```
SELECT * FROM tb_fabricante WHERE nome  
LIKE 'S _ _ _';
```



**SELECT lista_de_colunas FROM tabela NOT LIKE
expressão**

**Exemplo : Mostrar todos os clientes cujos nomes não
começam a letra “A”.**

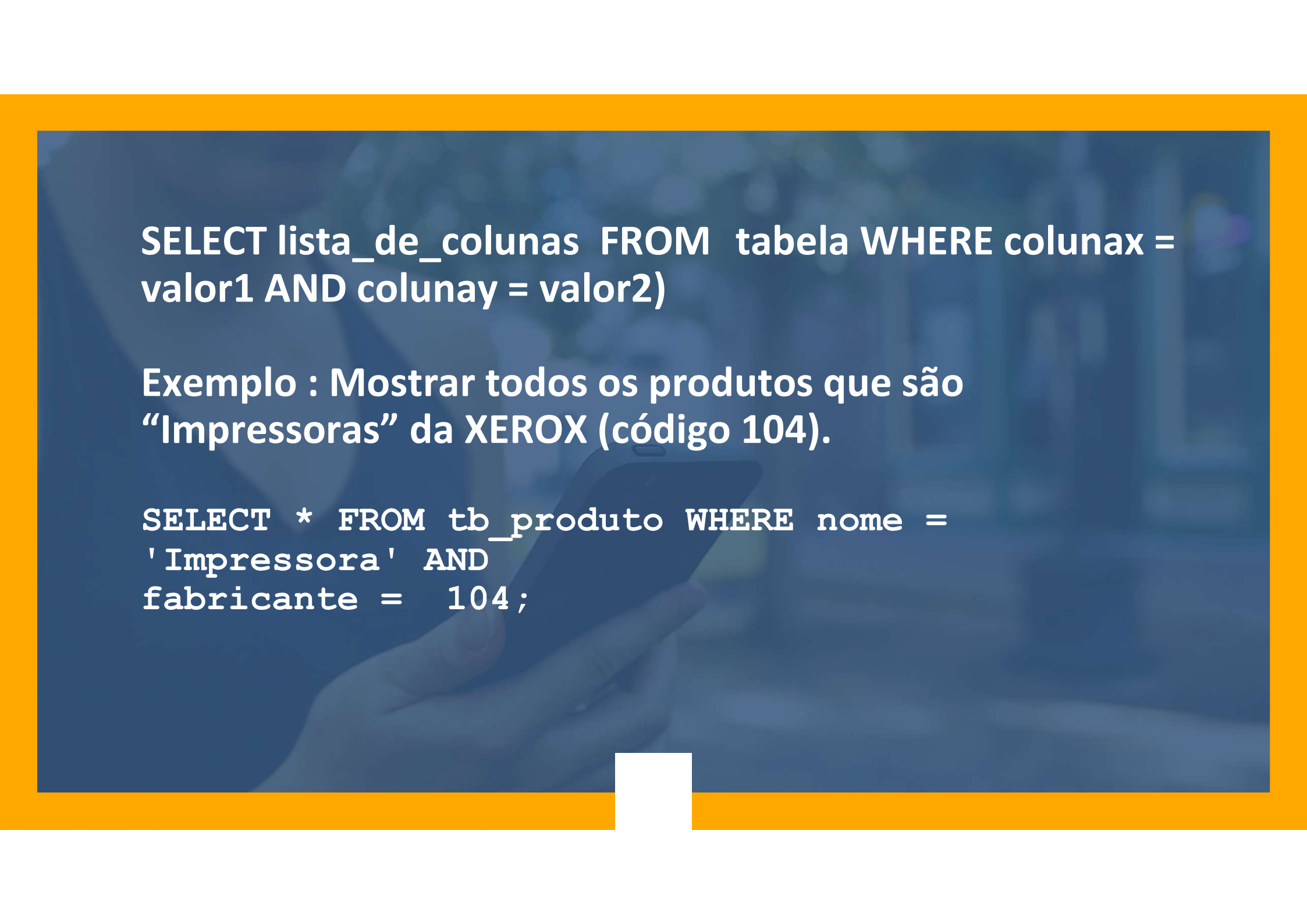
**SELECT * FROM tb_cliente WHERE nome NOT LIKE
'A%';**



SELECT lista_de_colunas FROM tabela WHERE coluna IS NULL

Exemplo : Mostrar todos os pedidos que não possuem desconto

SELECT * FROM tb_pedido WHERE desconto IS NULL;



```
SELECT lista_de_colunas FROM tabela WHERE colunax =  
valor1 AND colunay = valor2)
```

Exemplo : Mostrar todos os produtos que são
“Impressoras” da XEROX (código 104).

```
SELECT * FROM tb_produto WHERE nome =  
'Impressora' AND  
fabricante = 104;
```

SELECT lista de colunas **FROM** tabela **WHERE** coluna
[NOT] BETWEEN intervalo

Exemplo : Exibir todos os dados dos produtos com preço entre R\$ 1.000,00 e R\$ 5.000,00 ordenados pelo nome do produto em ordem ascendente.

```
SELECT * FROM tb_produto WHERE preco  
BETWEEN 1000 AND 5000 ORDER BY  
nome ASC;
```



```
SELECT COUNT(*) FROM tabela
```

Exemplo : Exibir em uma coluna chamada Quantidade o total de clientes cadastrados.

```
SELECT COUNT(*) AS Quantidade FROM  
tb_cliente;
```



```
SELECT SUM(coluna) FROM tabela
```

Exemplo : Exibir em uma nova coluna “Total” a soma dos valores de todos os produtos.

```
SELECT SUM(preco) AS Total FROM  
tb_produto;
```




```
SELECT AVG(coluna) FROM tabela
```

Exemplo : Exibir a “Média dos salários” dos vendedores da empresa.

```
SELECT AVG(salario) AS 'Média dos Salários'  
FROM tb_vendedor;
```



```
SELECT MAX(coluna) FROM tabela 4
```

Exemplo : Exibir o “Maior preço” do produto mais caro na empresa.

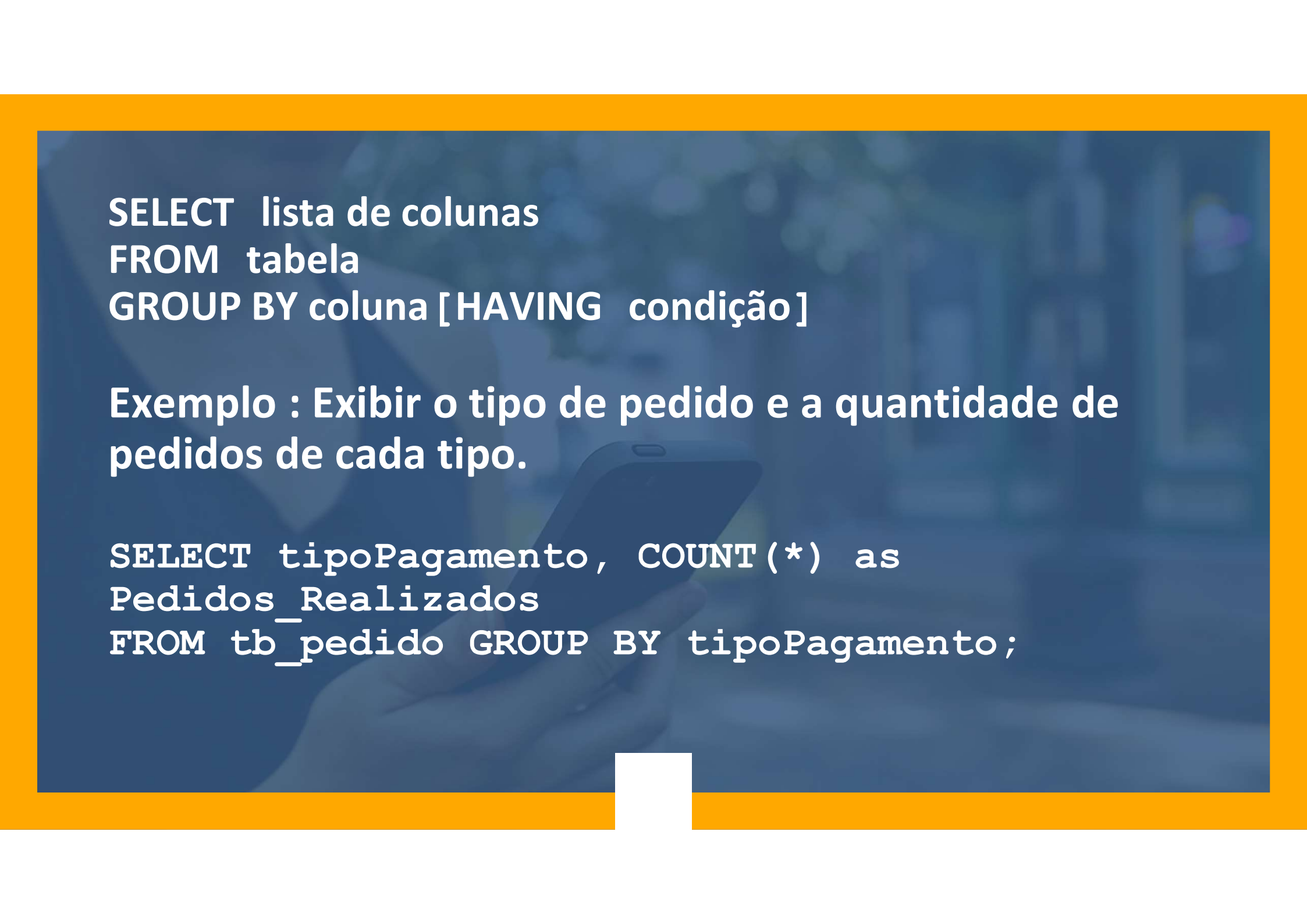
```
SELECT MAX(preco) AS 'Maior Preço' FROM  
tb_produto;
```



```
SELECT MIN(coluna) FROM tabela
```

Exemplo : Exibir o “Menor salário” pago a um vendedor na empresa.

```
SELECT MIN(salario) AS 'Menor salário' FROM  
tb_vendedor;
```



```
SELECT lista de colunas  
FROM tabela  
GROUP BY coluna [HAVING condição]
```

Exemplo : Exibir o tipo de pedido e a quantidade de pedidos de cada tipo.

```
SELECT tipoPagamento, COUNT(*) as  
Pedidos_Realizados  
FROM tb_pedido GROUP BY tipoPagamento;
```

Obrigado!

Perguntas?

Contatos:

 <https://www.linkedin.com/in/daniel-wp/>

 danielwarellapitsch@gmail.com

 <https://github.com/DanielWPitsch>