

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARAÍBA



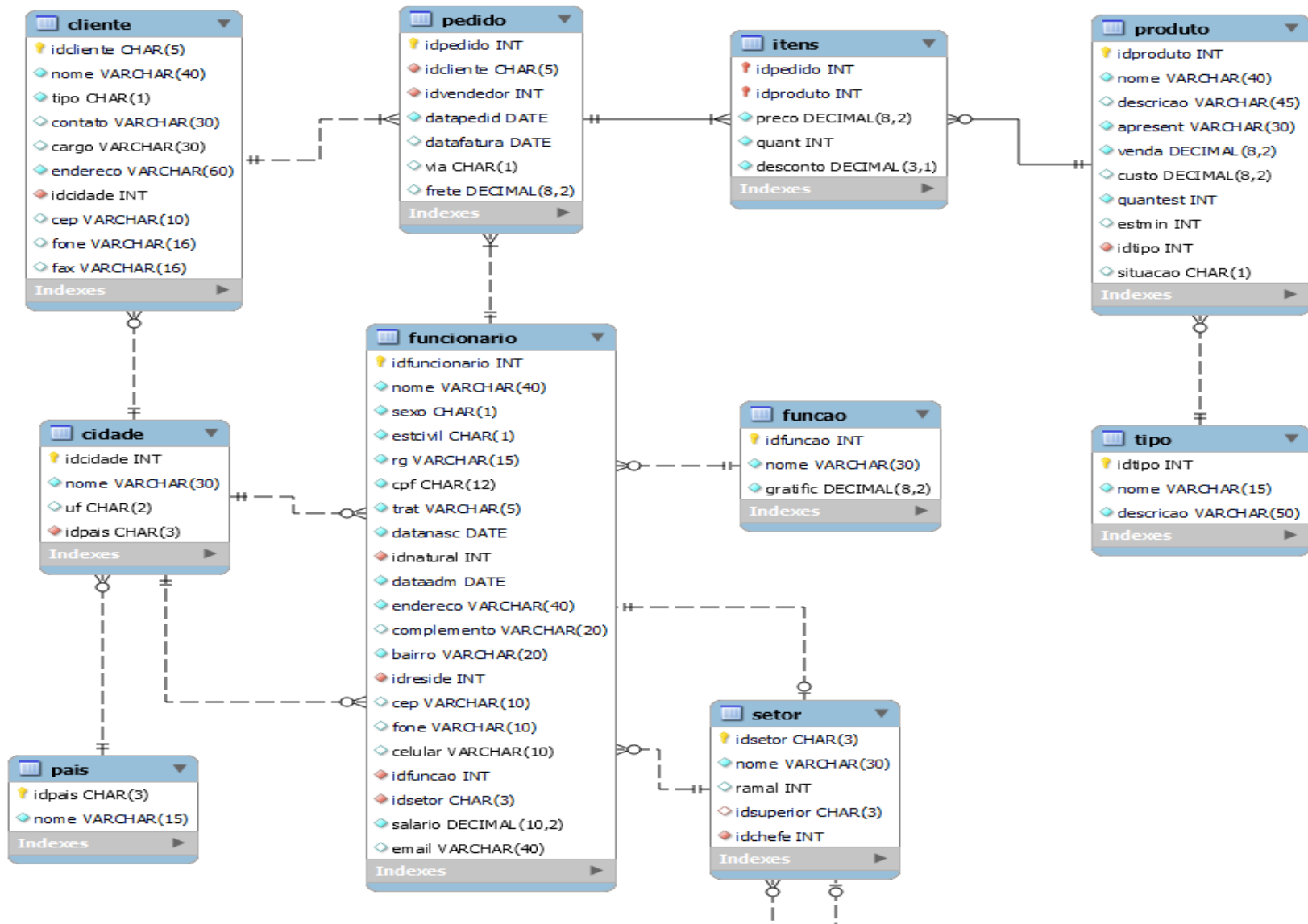
Linguagem SQL

DQL - Consultas Simples

Banco de Dados ContPedido



IFPB
Campus João Pessoa



O Comando SELECT



- O comando **SELECT** permite consultar dados em uma ou mais tabelas e retorna um conjunto de linhas e colunas, de acordo com as cláusulas especificadas.

A	B	C	D	E
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35

Tabela inicial



A	C	E
6	8	10
16	18	20
21	23	25
31	33	35

Tabela resultante

- O comando **SELECT** contém seis cláusulas:

SELECT	Indica as colunas que serão exibidas.
FROM	Indica a(s) tabela(s) de onde vem os dados.
WHERE	Restringe a seleção de linhas da tabela com base em uma expressão condicional.
GROUP BY	Agrupa as linhas selecionadas com base em um ou mais atributos.
HAVING	Restringe a seleção de linhas agrupadas com base em uma expressão condicional.
ORDER BY	Ordena as linhas selecionadas com base em um ou mais atributos.



Sintaxe:

```
SELECT * FROM <tabela>
```

Exemplo 1: Obter todos os dados de todos os clientes.

```
SELECT * FROM Cliente;
```

Exemplo 2: Obter todos os dados de todos os produtos.

```
SELECT * FROM Produto;
```

Sintaxe:

```
SELECT <lista de colunas>  
FROM <tabela>
```

Exemplo 3: Obter código, nome e telefone de todos os clientes.

```
SELECT idcliente, nome, fone  
FROM Cliente;
```

Exemplo 4: Obter código, data e valor do frete de todos os pedidos.

```
SELECT idpedido, datapedid, frete  
FROM Pedido;
```



Sintaxe:

```
SELECT <coluna> [AS] <alias>  
FROM <tabela>
```

Exemplo 5: Obter o código com o título “Produto” e o preço de venda com o título “Preço de venda” de todos os produtos.

```
SELECT idproduto AS 'Produto',  
       venda AS 'Preço de venda'  
FROM Produto;
```

Exemplo 6: Obter o nome com o título “Função” e o valor da gratificação com o título “Gratificação” de todas as funções.

```
SELECT nome AS 'Função', gratific AS 'Gratificação'  
FROM Funcao;
```



Sintaxe:

```
SELECT DISTINCT <lista de colunas>  
FROM <tabela>
```

Exemplo 7: Obter o código de todos os clientes que já fizeram pedido na empresa (sem repetições).

```
SELECT DISTINCT idcliente  
FROM Pedido;
```

Exemplo 8: Obter o bairro e o código da cidade de todos os funcionários da empresa (sem repetições).

```
SELECT DISTINCT bairro, idreside  
FROM Funcionario;
```




Sintaxe:

```
SELECT <lista de colunas, expressão numérica>  
FROM <tabela>
```

Exemplo 9: Obter código, quantidade em estoque, preço de venda e (quantidade x preço de venda) para cada produto.

```
SELECT idproduto, quantest, venda,  
       quantest * venda AS 'Valor'  
FROM Produto;
```

Exemplo 10: Obter código, nome, salário e o salário com 32% de aumento para todos os funcionários.

```
SELECT idfuncionario, nome, salario,  
       salario * 1.32 AS 'Sal.c/Aumento'  
FROM Funcionario;
```



Sintaxe:

```
SELECT <lista de colunas, string>  
FROM <tabela>
```

Exemplo 11: Obter código, nome e uma coluna contendo “classificado” para todos os clientes.

```
SELECT idcliente, nome, 'classificado' AS 'Resultado'  
FROM Cliente;
```

Exemplo 12: Obter código, nome e uma coluna contendo “num.ref” com o título “Referência” para todos os produtos.

```
SELECT idproduto, nome, 'num.ref.' AS 'Referência'  
FROM Produto;
```



Sintaxe:

```
SELECT <lista de colunas>  
FROM <tabela>  
ORDER BY <coluna> [ASC] [DESC]
```

Exemplo 13: Obter código, nome, contato e cargo de todos os clientes, ordenados pelo cargo em ordem ascendente.

```
SELECT idcliente, nome, contato, cargo  
FROM Cliente  
ORDER BY cargo;
```



Exemplo 14: Obter código, nome, estado civil e data de nascimento de todos os funcionários, ordenados pela data de nascimento em ordem descendente.

```
SELECT idfuncionario, nome, estcivil, datanasc  
FROM Funcionario  
ORDER BY datanasc DESC;
```

Exemplo 15: Obter código, nome, preço de custo, preço de venda e a diferença entre esses dois preços para todos os produtos, ordenados pela diferença em ordem descendente.

```
SELECT idproduto, nome, custo, venda,  
       venda - custo AS 'Diferença'  
FROM Produto  
ORDER BY venda - custo DESC;
```



Exemplo 16: Obter código, nome e tipo de todos os produtos, ordenados pelo tipo em ordem ascendente e pelo nome em ordem descendente.

```
SELECT idproduto, nome, idtipo  
FROM Produto  
ORDER BY idtipo ASC, nome DESC;
```

Exemplo 17: Obter código, nome, cidade onde reside, sexo e salário de todos os funcionários, ordenados por cidade e sexo em ordem ascendente e por salário em ordem descendente.

```
SELECT idfuncionario, nome, idreside, sexo, salario  
FROM Funcionario  
ORDER BY idreside, sexo, salario DESC;
```



- A filtragem deve ser feita utilizando-se a cláusula **WHERE**.
- A cláusula **WHERE** filtra as linhas com base em uma condição que deve ser especificada.
- A condição é representada por uma ou mais expressões condicionais , separadas pelos operadores lógicos **and**, **or** e **not**.
- O formato de cada expressão condicional é:
<coluna> <operador> <valor>
- Podem ser utilizados diversos operadores para compor as expressões condicionais utilizadas.



OPERADORES DE COMPARAÇÃO

=	Igual	quantidade = 20
		nome = 'Paulo'
		datanasc = '22/02/2011'
<	Menor que	preço < 20.00
<=	Menor ou igual	estoque <= 50
>	Maior que	dataadm > '22/02/2011'
>=	Maior ou igual	credito >= 10000.00
<> ou !=	Diferente	nome <> 'Paulo'



OPERADORES DE INCLUSÃO

IN	Dentro do conjunto	bairro IN ('Tambaú', 'Bessa')
NOT IN	Fora do conjunto	setor NOT IN ('COM', 'MKT', 'ALM')

OPERADORES DE INTERVALO

BETWEEN	Dentro do intervalo	preço BETWEEN 30.00 and 150.00
NOT BETWEEN	Fora do intervalo	quant NOT BETWEEN 10 and 200



OPERADORES DE NULO

IS NULL	O valor é nulo	telefone IS NULL
IS NOT NULL	O valor não é nulo	fax is NOT NULL

OPERADORES DE COMPARAÇÃO DE STRINGS

LIKE	Igual	nome LIKE 'M%'
NOT LIKE	Diferente	nome NOT LIKE 'A_D%'
%	Substitui um conjunto de caracteres	
_	Substitui um único caractere	



OPERADORES DE COMPARAÇÃO DE STRINGS

'A%'	String iniciando por 'A' .
'%Silva'	String terminando por 'Silva' .
'%Silva%'	String contendo 'Silva' em qualquer posição.
'A%Silva'	String iniciando por 'A' e terminando por 'Silva' .
'A_S'	String com três caracteres, iniciando por 'A' e terminando por 'S' .



OPERADORES LÓGICOS

AND	Ambas as condições são verdadeiras	valor > 1000.00 and quant <= 25
OR	Uma das condições é verdadeira	(valor > 1000.00 and quant <= 25) or (estmin <10 and tipo IN (2,4,7))
NOT	A condição é falsa	preço > 120.00 or not (quant >= 250)



Sintaxe:

```
SELECT <lista de colunas>  
FROM <tabela>  
WHERE <condição>
```

Exemplo 18: Obter código, nome, tipo, quantidade em estoque e preço de venda dos produtos que tenham quantidade em estoque maior que 20 ou preço de venda inferior a R\$ 50,00, ordenados pelo preço de venda em ordem decrescente.

```
SELECT idproduto, nome, idtipo, quantest, venda  
FROM Produto  
WHERE quantest > 20 or venda < 50.00  
ORDER BY venda DESC;
```



Exemplo 19: Obter todos os dados dos clientes que são do tipo pessoa jurídica e que não sejam da cidade de código igual a 14, ou que são do tipo pessoa física e da cidade de código igual a 14, ordenando pelo nome em ordem ascendente.

```
SELECT *  
FROM Cliente  
WHERE (tipo = 'J' and idcidade <> 14) or  
      (tipo = 'F' and idcidade = 14)  
ORDER BY nome;
```



Sintaxe:

```
SELECT <lista de colunas>  
FROM <tabela>  
WHERE <coluna> [NOT] IN (<lista de valores>)
```

Exemplo 20: Obter código, nome, endereço e bairro dos funcionários que residam em Manaíra, Tambaú ou Bessa, ordenados pelo nome do funcionário em ordem ascendente.

```
SELECT idfuncionario, nome, endereco, bairro  
FROM Funcionario  
WHERE bairro IN ('Manaíra', 'Tambaú', 'Bessa')  
ORDER BY nome;
```



Exemplo 21: Obter código, data e via de transporte de todos os pedidos que tenham sido transportados por via aérea ou marítima.

```
SELECT idpedido, datapedid, via
FROM Pedido
WHERE via IN ('A', 'M');
```

Exemplo 22: Obter código, nome e tipo de todos os produtos que não sejam do tipo 2 ou 4, ordenados pelo tipo em ordem descendente e pelo nome em ordem ascendente.

```
SELECT idproduto, nome, idtipo
FROM Produto
WHERE idtipo NOT IN (2, 4)
ORDER BY idtipo DESC, nome;
```



Sintaxe:

```
SELECT <lista de colunas>  
FROM <tabela>  
WHERE <coluna> [NOT] BETWEEN (<intervalo>)
```

Exemplo 23: Obter todos os dados dos produtos com quantidade em estoque entre 10 e 30 unidades, ordenados pelo nome do produto em ordem ascendente.

```
SELECT *  
FROM Produto  
WHERE quantest BETWEEN 10 AND 30  
ORDER BY nome;
```




Exemplo 24: Obter o nome e a data de nascimento dos funcionários que nasceram na década de 60, ordenados por data de nascimento em ordem decrescente e por nome em ordem ascendente.

```
SELECT nome, datanasc
FROM Funcionario
WHERE datanasc BETWEEN '1960/01/01' and '1969/12/31'
ORDER BY datanasc DESC, nome;
```

Exemplo 25: Obter código, cliente e data de todos os pedidos que não tenham sido realizados entre 1997 e 1999.

```
SELECT idpedido, idcliente, datapedid
FROM Pedido
WHERE YEAR(datapedid) NOT BETWEEN 1997 and 1999;
```



Sintaxe:

```
SELECT <lista de colunas>  
FROM <tabela>  
WHERE <coluna> IS [NOT] NULL
```

Exemplo 26: Obter código, nome e email dos funcionários que não tenham email, ordenando pelo nome em ordem ascendente.

```
SELECT idfuncionario, nome, email  
FROM Funcionario  
WHERE email IS NULL  
ORDER BY nome;
```



Exemplo 27: Obter tipo, código, nome e fax dos clientes que tenham fax, ordenados pelo tipo e pelo nome.

```
SELECT tipo, idcliente, nome, fax
FROM Cliente
WHERE fax IS NOT NULL
ORDER BY tipo, nome;
```

Exemplo 28: Obter código, nome, email e celular dos funcionários que não tenham email mas possuam telefone celular, ordenados pelo nome em ordem ascendente.

```
SELECT idfuncionario, nome, email, celular
FROM Funcionario
WHERE email IS NULL and celular IS NOT NULL
ORDER BY nome;
```



Sintaxe:

```
SELECT <lista de colunas>  
FROM <tabela>  
WHERE <coluna> LIKE <string>
```

Exemplo 29: Obter o código e o nome de todos os funcionários que tenham o nome começando por "M", ordenados pelo nome.

```
SELECT idfuncionario, nome  
FROM Funcionario  
WHERE nome LIKE 'M%'  
ORDER BY nome;
```



Exemplo 30: Obter todos os dados dos funcionários que tenham o nome terminando por "Silva", ordenados pelo nome.

```
SELECT *  
FROM Funcionario  
WHERE nome LIKE '%Silva'  
ORDER BY nome;
```

Exemplo 31: Obter todos os dados dos funcionários que tenham o sobrenome "Costa", ordenados pelo nome.

```
SELECT *  
FROM Funcionario  
WHERE nome LIKE '%Costa%'  
ORDER BY nome;
```



Exemplo 32: Obter todos os dados dos funcionários que tenham o nome iniciando por "Mar", seguido de um caractere qualquer, a letra "a" e quaisquer caracteres adicionais, ordenados pelo nome.

```
SELECT *  
FROM Funcionario  
WHERE nome LIKE 'Mar_a%'  
ORDER BY nome;
```

Exemplo 33: Obter os dados dos funcionários que tenham o nome iniciando por "Maria" e terminando por "Lima", ordenados pelo nome.

```
SELECT *  
FROM Funcionario  
WHERE nome LIKE 'Maria%Lima'  
ORDER BY nome;
```



Sintaxe:

```
SELECT <lista de colunas>  
FROM <tabela>  
LIMIT [<linha_inicial>,<quantidade_de_linhas>
```

Exemplo 34: Obter o código e o nome dos tipos de produtos, ordenado pelo código do tipo, apenas as 5 primeiras linhas.

```
SELECT idtipo, nome  
FROM Tipo  
ORDER BY idtipo  
LIMIT 5;
```



Exemplo 35: Obter o código e o nome dos tipos de produtos, ordenado pelo código do tipo, iniciando na 5ª linha e limitando a 3 linhas.

```
SELECT idtipo, nome  
FROM Tipo  
ORDER BY idtipo  
LIMIT 4,3;
```

Exemplo 36: Obter código do pedido, código do produto e quantidade dos 10 produtos com as maiores quantidades em um pedido, ordenados pela quantidade em ordem decrescente.

```
SELECT idpedido, idproduto, quant  
FROM Itens  
ORDER BY quant DESC  
LIMIT 10;
```