



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE

# Programação com acesso a BD

- **Aula 12: Linguagem SQL: Consultas Complexas em Tabelas**

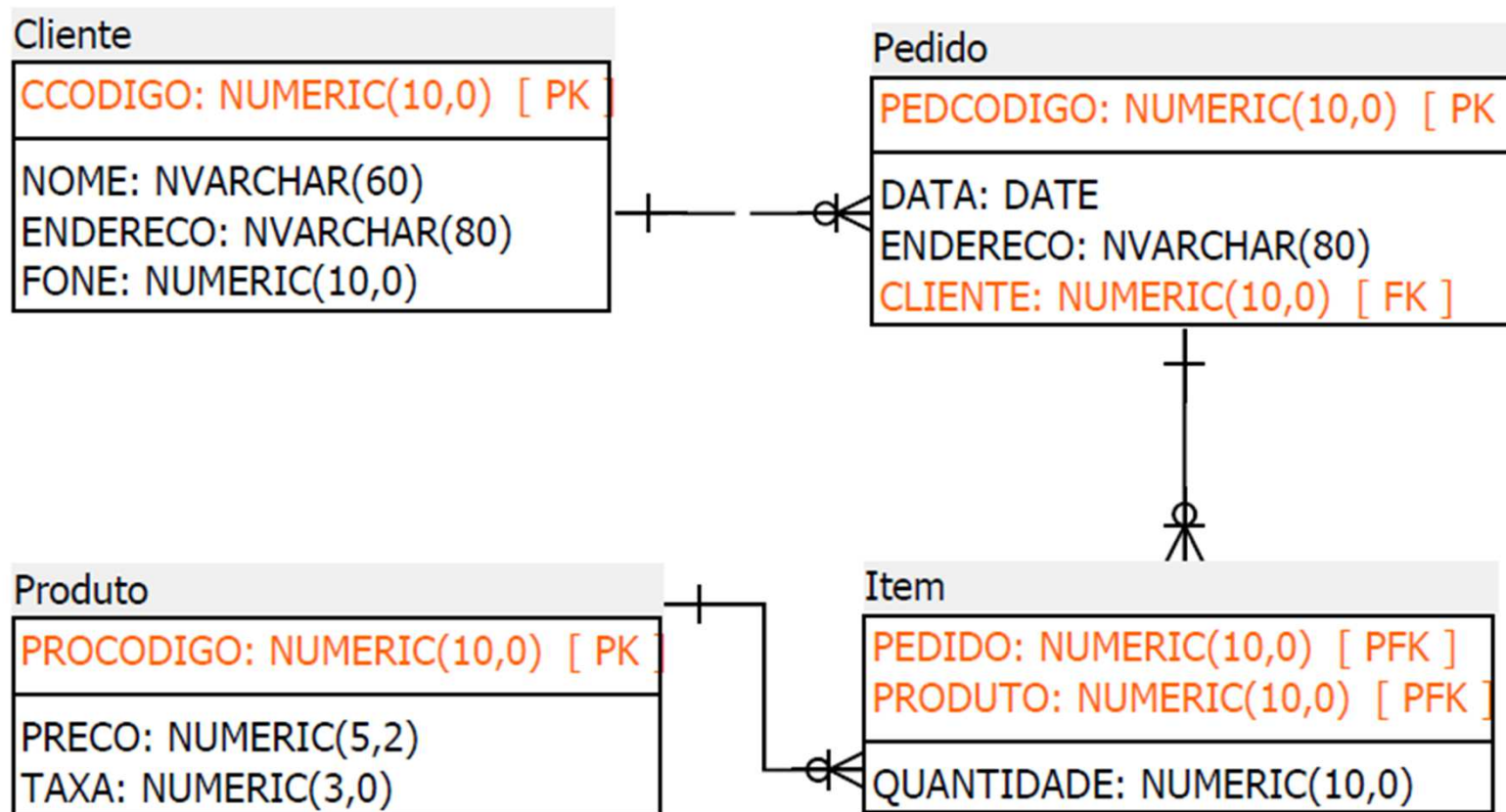
Prof.: Clayton Maciel Costa  
clayton.maciell@ifrn.edu.br

# Agenda

- Consultas SQL envolvendo NULL;
- Consultas Aninhadas IN;
- Funções EXISTS e NOT EXISTS;
- Funções de Agregação:
  - GROUP BY, HAVING.
- Operações com conjuntos:
  - UNION, INTERSECT, MINUS, EXCEPT.

# Banco de Dados Exemplo

- Considere o esquema de banco de dados:



# Comparações envolvendo NULL

- A SQL permite consultas que chequem se o valor de um atributo é NULL;
- É utilizado IS ou IS NOT para a comparação;
- Exemplos:
  - Listar os pedidos que não tem cliente

```
Select *  
From Pedido  
Where cliente IS NULL
```

- Listar todos os clientes que possuem telefone

```
Select *  
From Cliente  
Where fone IS NOT NULL
```

# Consultas Aninhadas

- Uma consulta SQL é aninhada quando ela está dentro de outra consulta SQL;
- Pode-se utilizar o comparador **IN**, que compara um valor  $v$  com um conjunto de valores  $V$  e evolui para verdadeiro, se  $v$  for um dos elementos de  $V$ .
  - Exemplo:
    - Listar os pedidos dos clientes do endereço Abolição II.

```
Select *  
From Pedido as p  
Where p.cliente IN (Select ccodigo  
                    From Cliente  
                    Where endereco = 'Abolição II')
```

# Consultas Aninhadas

- Mesmo exemplo utilizando junções:
  - Listar os pedidos dos clientes do endereço Abolição II

```
Select *  
From Pedido as p, Cliente as c  
Where p.cliente = c.ccodigo and p.endereco = 'Abolição II'
```

- Exemplos:
  - Listar os clientes dos pedidos 10, 20 ou 30

```
Select *  
From Cliente  
Where ccodigo IN (Select cliente  
                   From Pedido  
                   Where pedcodigo IN (10, 20, 30))
```

# Consultas Aninhadas

- Exemplo:
  - Listar os pedidos dos clientes que não moram no Abolição II.

```
Select *  
From Pedido  
Where cliente NOT IN (Select ccodigo  
                        From Cliente  
                        Where endereco = 'Abolição II')
```

# Funções Exists e Not Exists

- A função Exists e Not Exists da SQL é usada para verificar se o resultado de uma consulta aninhada correlacionada é vazio (não contém nenhuma tupla) ou não;
- Exemplos:
  - Listar os clientes que não possuem nenhum pedido

```
Select *  
From Cliente  
Where NOT EXISTS (Select *  
                  From Pedido  
                  Where cliente = ccodigo);
```



# Funções EXISTS e NOT EXISTS

- Exemplos:
  - Listar os clientes que possuem pelo menos um pedido

```
Select *  
From Cliente  
Where EXISTS (Select *  
              From Pedido  
              Where cliente = ccodigo);
```

# Funções de agregação – GROUP BY

- Listar o número de pedidos por cliente.

```
Select c.nome, COUNT(p.pedcodigo)
From Cliente as c, Pedido as p
Where c.ccodigo=p.cliente
Group By c.nome;
```

- Nota:
  - Atributos na cláusula `select` fora de funções de agregação têm de aparecer na lista `group by`
- Nota:
  - Se aparecer mais do que um atributo em `group by`, então cada grupo é formado pelas tuplas com valores iguais em todos esses os atributos

# Funções de agregação – HAVING

- Listar o número de pedidos por cliente, para clientes que tiverem mais de 5 pedidos

```
Select c.nome, COUNT(p.pedcodigo) as numeroPedidos  
From Cliente as c, Pedido as p  
Where c.ccodifgo=p.cliente  
Group By c.nome  
Having numeroPedidos > 5
```

- Nota:
  - predicados na cláusula `having` são aplicados depois da formação dos grupos, enquanto que os predicados na cláusula `where` são aplicados antes da formação dos grupos.

# Operações com Conjuntos

- As operações com conjuntos `union`, `intersect` e `except` (`minus` no Oracle) operam sobre relações e correspondem aos operadores  $\cup$ ,  $\cap$ ,  $-$ .
- Cada uma das operações anteriores elimina as duplicações automaticamente.
- Para reter duplicados deve-se utilizar as respectivas versões multiconjunto `union all`, `intersect all` e `except all`

# Operações com Conjuntos - Union

- Listar todos os clientes que realizaram pedidos com mais de 5 itens **ou** contendo algum produto de preço > R\$30,00

```
(Select c.nome  
  From Cliente as c, Pedido as p, Produto as pr  
  Where c.ccodifgo=p.cliente and  
        p.pedcodigo=i.pedido and  
        i.quantidade>5)
```

UNION

```
(Select c.nome  
  From Cliente as c, Pedido as p, Produto as pr  
  Where c.ccodifgo=p.cliente and  
        p.pedcodigo=i.pedido and  
        i.produto=pr.procodigo and  
        i.preço>30)
```

# Operações com Conjuntos - Intersect

- Listar todos os clientes que realizaram pedidos com mais de 5 itens e contendo algum produto de preço > R\$30,00

```
(Select c.nome  
From Cliente as c, Pedido as p, Produto as pr  
Where c.ccodifgo=p.cliente and  
      p.pedcodigo=i.pedido and  
      i.quantidade>5)
```

INTERSECT

```
(Select c.nome  
From Cliente as c, Pedido as p, Produto as pr  
Where c.ccodifgo=p.cliente and  
      p.pedcodigo=i.pedido and  
      i.produto=pr.procodigo and  
      i.preço>30)
```

# Operações com Conjuntos - Except

- Listar todos os clientes que realizaram pedidos com mais de 5 itens **mas não** contendo algum produto de preço > R\$30,00

```
(Select c.nome  
From Cliente as c, Pedido as p, Produto as pr  
Where c.ccodifgo=p.cliente and  
      p.pedcodigo=i.pedido and  
      i.quantidade>5)
```

EXCEPT

```
(Select c.nome  
From Cliente as c, Pedido as p, Produto as pr  
Where c.ccodifgo=p.cliente and  
      p.pedcodigo=i.pedido and  
      i.produto=pr.procodigo and  
      i.preço>30)
```

# FIM



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE