

# **Banco de Dados**

**José Antônio da Cunha**

**CEFET – RN**

# **Manipulando Dados**

**Sub-consulta(Subquery) – as subqueries permitem que você obtenha dados de uma tabela com base na existência desses dados em outra ou outras tabelas.**

# Manipulando Dados

## Sub-consultas

Suponha que você tenha uma tabela Pai e uma tabela Filho com a estrutura e os dados apresentados em seguida:

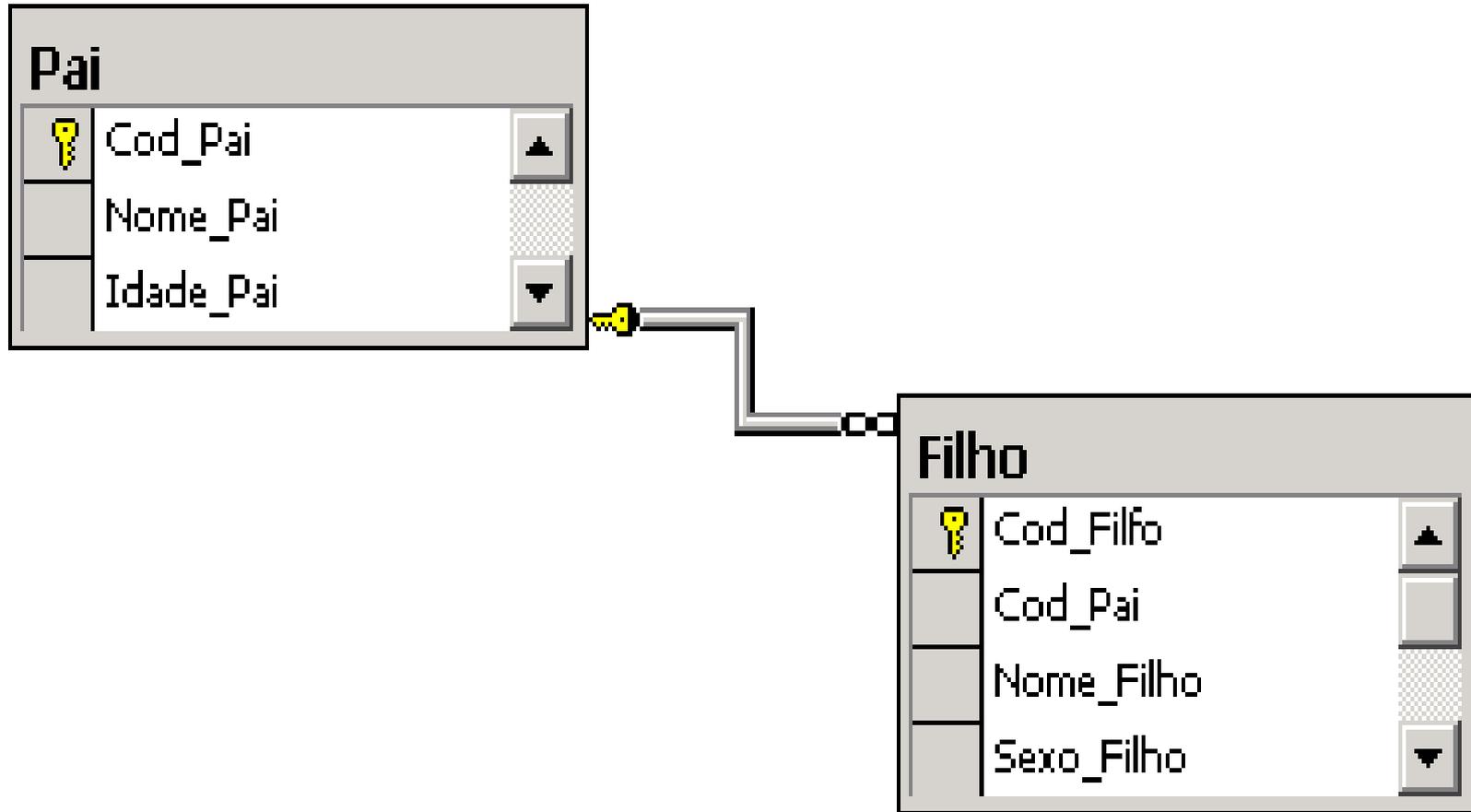
```
CREATE TABLE Pai
(
    Cod_Pai int identity not null Primary key,
    Nome_Pai char(30) not null,
    Idade_Pai tinyint not null
)
```

# Manipulando Dados

## Sub-consultas

```
CREATE TABLE Filho
(
  Cod_Filho int identity not null Primary key,
  Cod_Pai int          not null References Pai(Cod_pai),
  Nome_Filho char(30) not null,
  Sexo_Filho char(1)  not null Check(Sexo_Filho IN ('F', 'M'))
)
```

# Manipulando Dados



# Manipulando Dados

Cod_Pai	Nome_Pai	Idade_Pai
1	Daniel se Souza Leão Sobrinho	62
2	João Carlos da Silva	38
3	Fernando de Oliveira	36
4	Jairo de Oliveira Leão	32

Cod_Filho	Cod_Pai	Nome_Filho	Sexo_Filho
1	1	Renata de Oliveira Leão	F
2	1	Fernando de Oliveira Leão	M
3	1	Roberta de Oliveira Leão	F
4	1	Jairo de Oliveira Leão	M
5	2	Giovanna da Silva	F
6	3	Lucas Ribeiro Oliveira	M
7	3	Helder Ribeiro Oliveira	M

# Manipulando Dados

**Ex:** suponha que você precisasse obter da tabela Pai todos os dados do pai mais velho.

```
SELECT * FROM Pai  
WHERE Idade_Pai = (Select Max(Idade_Pai) FROM Pai)
```

```
SELECT * FROM Pai  
WHERE Idade_Pai IN (Select Max(Idade_Pai) FROM Pai)
```

# Manipulando Dados

**Ex:** Obtenha os dados do pais sem mostrar o pai mais velho.

```
SELECT * FROM Pai  
WHERE Idade_Pai <> (Select Max(Idade_Pai) FROM Pai)
```

```
SELECT * FROM Pai  
WHERE Idade_Pai < (Select Max(Idade_Pai) FROM Pai)
```

# Manipulando Dados

## Regras para Utilizar Subqueries:

1. As subqueries introduzidas com os sinais (>, >=, <, <=, =, <> ou !=) devem sempre retornar apenas um único valor;
2. Toda subquery deve ser escrita entre parênteses;
3. Toda subquery pode selecionar valor apenas de uma coluna.

# Manipulando Dados

Subqueries com os operadores **In** e **Not In** – as subqueries introduzidas com os operadores **IN** e **NOT IN** devem retornar sempre valores de apenas uma coluna, mas podem retornar mais de uma linha como resposta.

# Manipulando Dados

**Ex:** suponha que você precisasse obter da tabela Pai todas as colunas, apenas dos pais que possuem um ou mais filhos registrados na tabela Filho.

```
SELECT * FROM Pai
```

```
WHERE Cod_Pai IN (SELECT Cod_Pai FROM Filho)
```

# Manipulando Dados

**Ex:** suponha que você precisasse obter da tabela Pai todas as colunas, apenas dos pais que não possuem um ou mais filhos registrados na tabela Filho.

```
SELECT * FROM Pai
```

```
WHERE Cod_Pai NOT IN (SELECT Cod_Pai FROM Filho)
```

# Manipulando Dados

## Subqueries com os Operadores **Exists** e **Not Exists**

Ex: Obter todos os pais que possuem pelo menos um filho registrado na tabela filho.

```
SELECT * FROM Pai
```

```
WHERE EXISTS (SELECT Cod_Pai FROM Filho
```

```
WHERE Filho.Cod_Pai = Pai.Cod_Pai)
```

# Manipulando Dados

Observe que na subquery anterior, utiliza-se na cláusula WHERE a tabela Pai, que não foi citada na cláusula FROM dessa mesma subquery. A tabela Pai da cláusula Where da subquery está se relacionando com a Query. Portanto, para cada linha selecionada da tabela Pai pela query, o sistema tem que executar uma vez a subquery que obtém dados da tabela Filho. Assim, se houver na tabela Pai dez pais registrados, o sistema executa dez vezes a subquery correspondente.

Justamente porque essas subqueries estão “amarradas” com a query, ela são chamadas de **subqueries correlacionadas**.

**Nota:** As subqueries servem apenas como base de pesquisa para a query, portanto não será possível exibir seus dados.

# Manipulando Dados

## Substituindo Subqueries por outros comandos

Para selecionar todos os pais que possuem filhos, existem três possibilidades

*/\*1- Subquery correlacionada\*/*

```
SELECT * FROM Pai Where EXISTS
```

```
(SELECT Cod_Pai FROM Filho
```

```
WHERE Filho.CodPai = Pai.Cod_Filho)
```

# Manipulando Dados

## Substituindo Subqueries por outros comandos

Para selecionar todos os pais que possuem filhos, existem três possibilidades

*/\*2- Subquery não correlacionada\*/*

```
SELECT * FROM Pai
```

```
Where cod_pai IN (SELECT Cod_Pai FROM Filho)
```

# Manipulando Dados

## Substituindo Subqueries por outros comandos

Para selecionar todos os pais que possuem filhos, existem três possibilidades

```
/*3- Inner join*/
```

```
SELECT DISTINCT Pai.* FROM Pai INNER JOIN Filho  
ON (Pai.Cod_Pai = Filho.Cod_Pai)
```



**Melhor solução**

# Manipulando Dados

Para apresentar todos os dados da tabela Pai e a quantidade de filhos que cada pai possui, existem duas possibilidades:

```
/*Subquery correlacionada*/
```

```
SELECT *,
```

```
    (SELECT count(Cod_Pai) FROM Filho
```

```
    WHERE Filho.Cod_Pai = Pai.Cod_Pai ) As Qtd_Filho
```

```
FROM Pai
```

# Manipulando Dados

Para apresentar todos os dados da tabela Pai e a quantidade de filhos que cada pai possui, existem duas possibilidades:

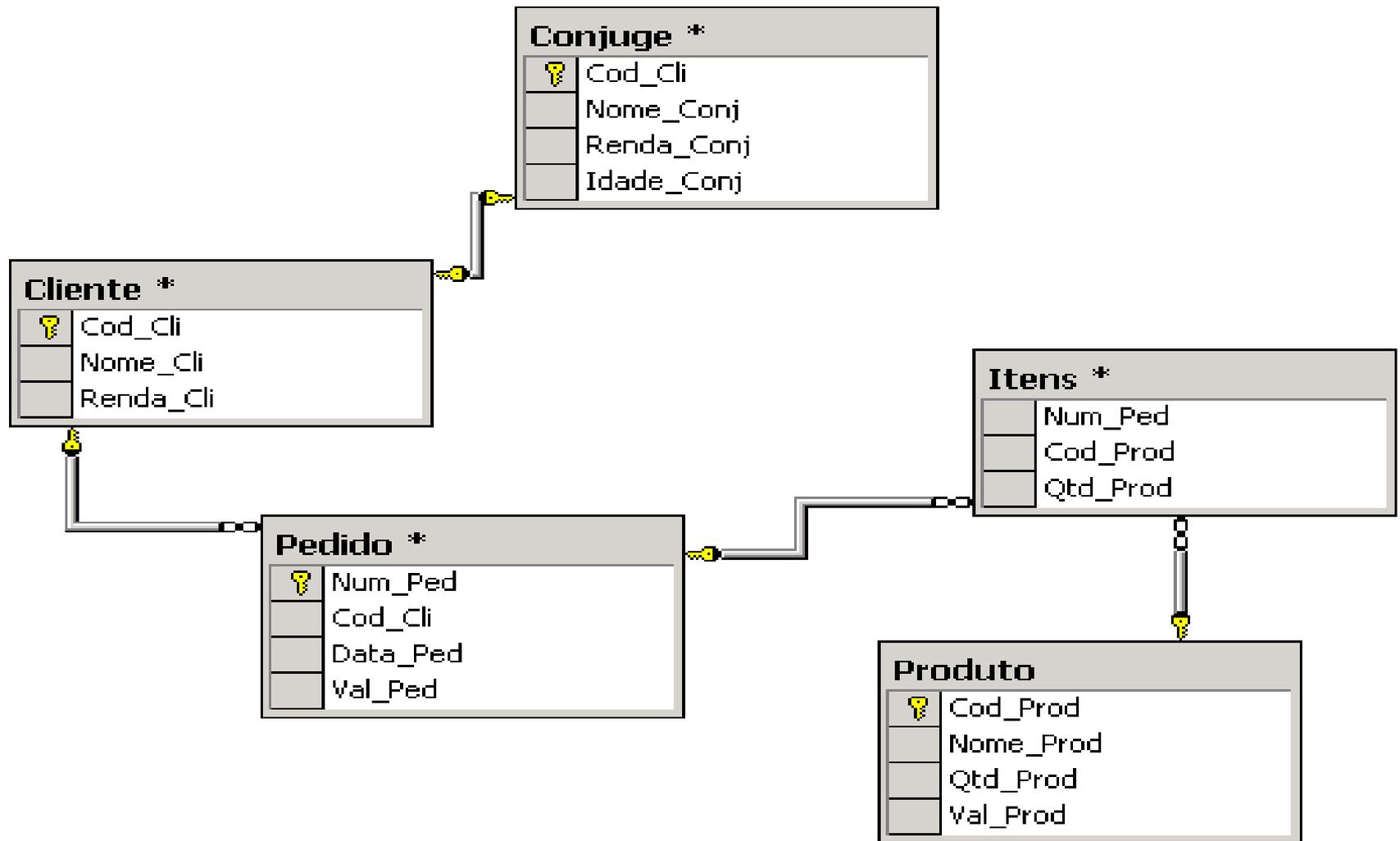
```
/*Left Join com Group By e a função Count() */
```

```
SELECT Pai.Cod_Pai, Pai.Nome_Pai, Count(Filho.Cod_Pai) AS  
Qtd_Filho  
FROM Pai LEFT JOIN Filho ON (Pai.Cod_Pai = Filho.Cod_Pai )  
Group By Pai.Cod_Pai, Pai.Nome_Pai
```

# Manipulando Dados

**Subqueries Aninhadas** – as subqueries são formadas de vários comandos Select. Vamos utilizar as tabelas do diagrama seguinte para exemplificarmos as subqueries aninhadas:

# Manipulando Dados



# Manipulando Dados

Para obter os dados de todos os clientes solteiros que já fizeram pelo menos um pedido.

```
SELECT * FROM Cliente
```

```
Where Cod_Cli NOT IN (SELECT Cod_Cli FROM Conjuge)
```

```
AND Cod_Cli IN (SELECT Cod_Cli FROM Pedido)
```

# Manipulando Dados

Para obter os dados de todos os clientes que compraram, por exemplo, o produto 1.

```
SELECT * FROM Cliente
```

```
Where Cod_Cli IN (SELECT Cod_Cli FROM Pedido
```

```
Where Num_Ped IN (SELECT Num_Ped FROM Itens
```

```
Where Cod_Prod = 1))
```

# Manipulando Dados

**Update com Subquery** – você pode alterar dados de uma tabela com base na existência de dados em outras tabelas utilizando subqueries. Por exemplo, você poderia alterar a renda dos clientes que compraram o produto de código 1.

```
UPDATE Cliente
```

```
SET Renda_Cli = Renda_Cli * 1.1
```

```
Where Cod_Cli IN (SELECT Cod_Cli From Pedido
```

```
    Where Num_Ped IN (SELECT Num_Ped From Itens
```

```
        Where Cod_Prod = 1))
```

# Manipulando Dados

**Delete com Subquery** – você pode excluir dados de uma tabela com base na existência de dados em outras tabelas utilizando subqueries. Por exemplo, você poderia excluir da tabela produto todos os produtos que nunca foram vendidos.

```
DELETE Produto
```

```
Where Cod_Prod NOT IN (SELECT Cod_Prod From Itens)
```

# Manipulando Dados

**Subquery com Join** – se for necessário, você pode utilizar joins com subqueries. Por exemplo, suponha que você precise selecionar todos os clientes e seus cônjuges, que compraram o produto de código 1.

```
SELECT *  
FROM Cliente INNER JOIN Conjuge  
ON (Cliente.Cod_Cli = Conjuge.Cod_Cli)  
Where Cod_Cli IN (SELECT Cod_Cli From Pedido  
Where Num_Ped IN (SELECT Num_Ped From Itens  
Where Cod_Prod = 1 ))
```