

Administrador de Banco de Dados

Módulo II

SQL Server: Consultas avançadas com Microsoft SQL Server 2019;

O que iremos aprender?



- A cláusula HAVING, para filtrar dados agrupados
- A cláusula CASE para classificar registros
- Como juntar tabelas usando diversos tipos de JOIN;
- Como unir consultas com os campos do mesmo tipo;
- Como usarmos Subconsulta;
- Views



Aula 9 – Having, Case e Juntando as tabelas

Having

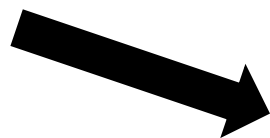
HAVING é uma condição (Filtro) que se aplica ao resultado de uma agregação.

A função **HAVING**, ela é aplicada para que nós possamos fazer um filtro sobre o resultado de um GROUP BY.

Having

X	Y
A	2
A	1
B	2
B	3
B	1
C	1
C	5
C	2
D	3

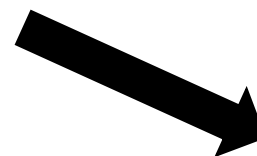
```
SELECT X, SUM(Y) FROM TABELA
GROUP BY X
```



X	Y
A	3
B	6
C	8
D	3

Quero só visualizar os campos cuja soma der maior que 6

```
SELECT X, SUM(Y) FROM TABELA
GROUP BY X
HAVING SUM(Y) >=6
```



X	Y
B	6
C	8

DESAFIO



Desafio

Quais são os clientes que fizeram mais de 2000 compras em 2016?

```
SELECT CPF, COUNT(*) AS 'QTD. DE COMPRAS' FROM [TABELA DE NOTAS FISCAIS]
WHERE YEAR(DATA) = 2016
GROUP BY CPF
HAVING COUNT(*) > 2000
```

CASE

CASE – Fazemos um teste em um ou mais campos e, dependendo do resultado, teremos um ou outro valor.

CASE

WHEN <CONDICAO 1> THEN <VALOR 1>

WHEN <CONDICAO 2> THEN <VALOR 2>

WHEN <CONDICAO 3> THEN <VALOR 3>

(...)

WHEN <CONDICAO N> THEN <VALOR N>

ELSE <VALOR_ELSE>

END

CASE

```
SELECT X,  
CASE
```

```
  WHEN Y >=8 AND Y <=10 THEN 'OTIMO'  
  WHEN Y >=7 AND Y <8   THEN 'BOM'  
  WHEN Y >=5 AND Y <7   THEN 'MEDIO'  
  ELSE 'RUIM'
```

```
END  
FROM TABELA
```

X	Y
CLIENTE 1	OTIMO
CLIENTE 2	MEDIO
CLIENTE 3	OTIMO
CLIENTE 4	OTIMO
CLIENTE 5	RUIM
CLIENTE 6	MEDIO
CLIENTE 7	BM
CLIENTE 8	BOM

X	Y
CLIENTE 1	8
CLIENTE 2	6
CLIENTE 3	9
CLIENTE 4	10
CLIENTE 5	4
CLIENTE 6	5
CLIENTE 7	7
CLIENTE 8	8

DESAFIO



Desafio

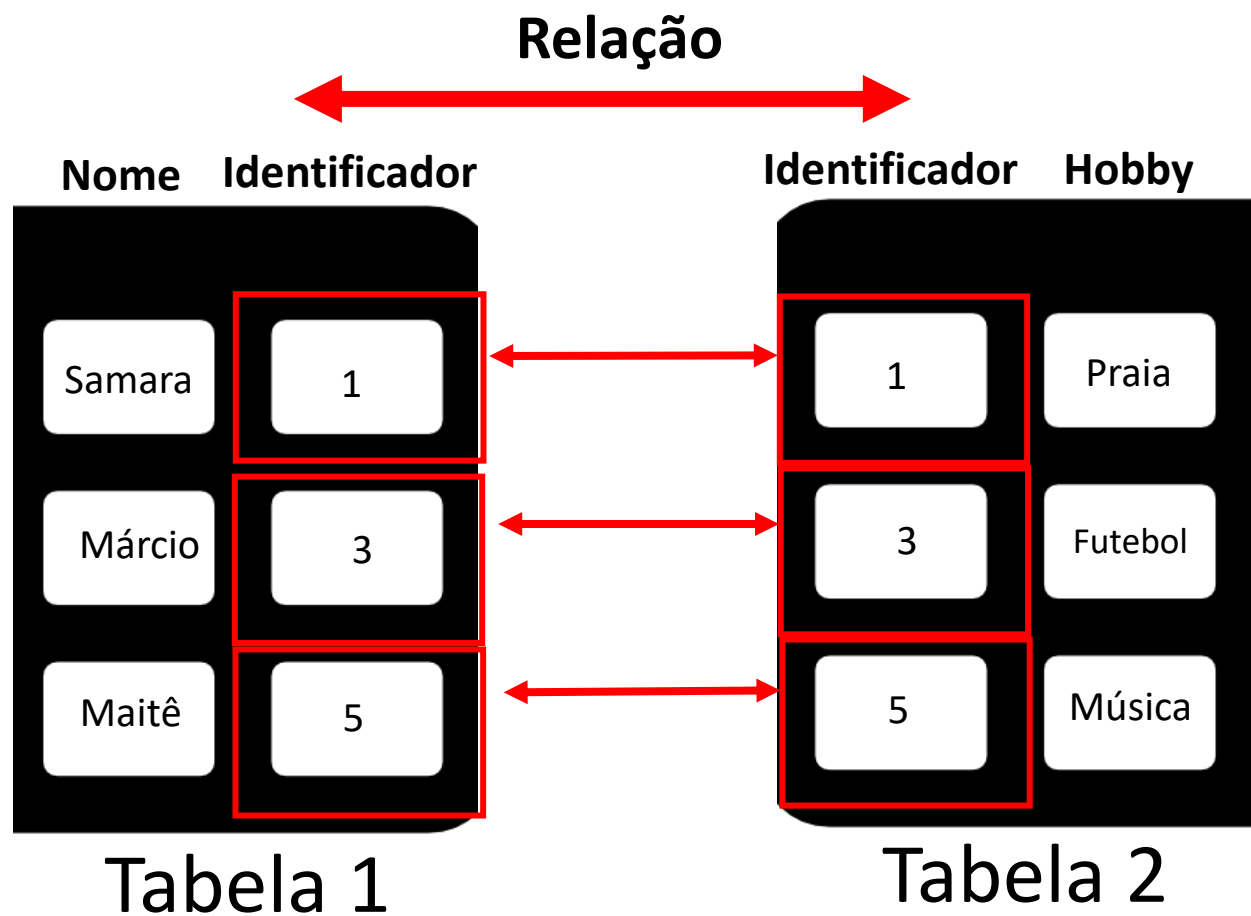
Veja o ano de nascimento dos clientes e classifique-os como: nascidos antes de 1990 são adultos, nascidos entre 1990 e 1995 são jovens e nascidos depois de 1995 são crianças.

Liste o nome do cliente e esta classificação.

```
SELECT [NOME],  
       CASE  
         WHEN YEAR([DATA DE NASCIMENTO]) < 1990 THEN 'Adulto'  
         WHEN YEAR ([DATA DE NASCIMENTO]) between 1990 and 1995 then 'Jovem'  
         ELSE 'Criança'  
       END AS 'CLASSIFICAÇÃO ETÁRIA'  
FROM [TABELA DE CLIENTES]
```

Usando JOIN

O JOIN nos traz a possibilidade de unir uma ou mais tabelas através de campo em comum.



- Eles não precisam ter o mesmo nome
- Eles tem que ser o mesmo tipo
- Independente da chave estrangeira
- Campo em comum entre as tabelas

Usando JOIN

O JOIN nos traz a possibilidade de unir uma ou mais tabelas através de campo em comum.

```
SELECT A.NOME, B.HOBBY FROM [TABELA ESQUERDA] A
INNER JOIN [TABELA DIREITA] B
ON A.IDENTIFICADOR = B.IDENTIFICADOR
```

Nome	Identificador
Samara	Praia
Márcio	Futebol

Nome	Identificador	Identificador	Hobby
Samara	1	1	Praia
Márcio	3	3	Futebol
Maitê	5	4	Música

Tabela Esquerda

Tabela Direita

DESAFIO



Desafio

Obtenha o faturamento anual da empresa. Leve em consideração que o valor financeiro das vendas consiste em multiplicar a quantidade pelo preço.

```
SELECT YEAR(DATA) AS 'ANO', SUM (QUANTIDADE * [PREÇO]) AS FATURAMENTO FROM [TABELA DE NOTAS FISCAIS] NF
INNER JOIN [TABELA DE ITENS NOTAS FISCAIS] INF
ON NF.NUMERO = INF.NUMERO
GROUP BY YEAR(DATA)
```

Usando LEFT JOIN

```
SELECT A.NOME, B.HOBBY FROM [TABELA ESQUERDA] A
LEFT JOIN [TABELA DIREITA] B
ON A.IDENTIFICADOR = B.IDENTIFICADOR
```

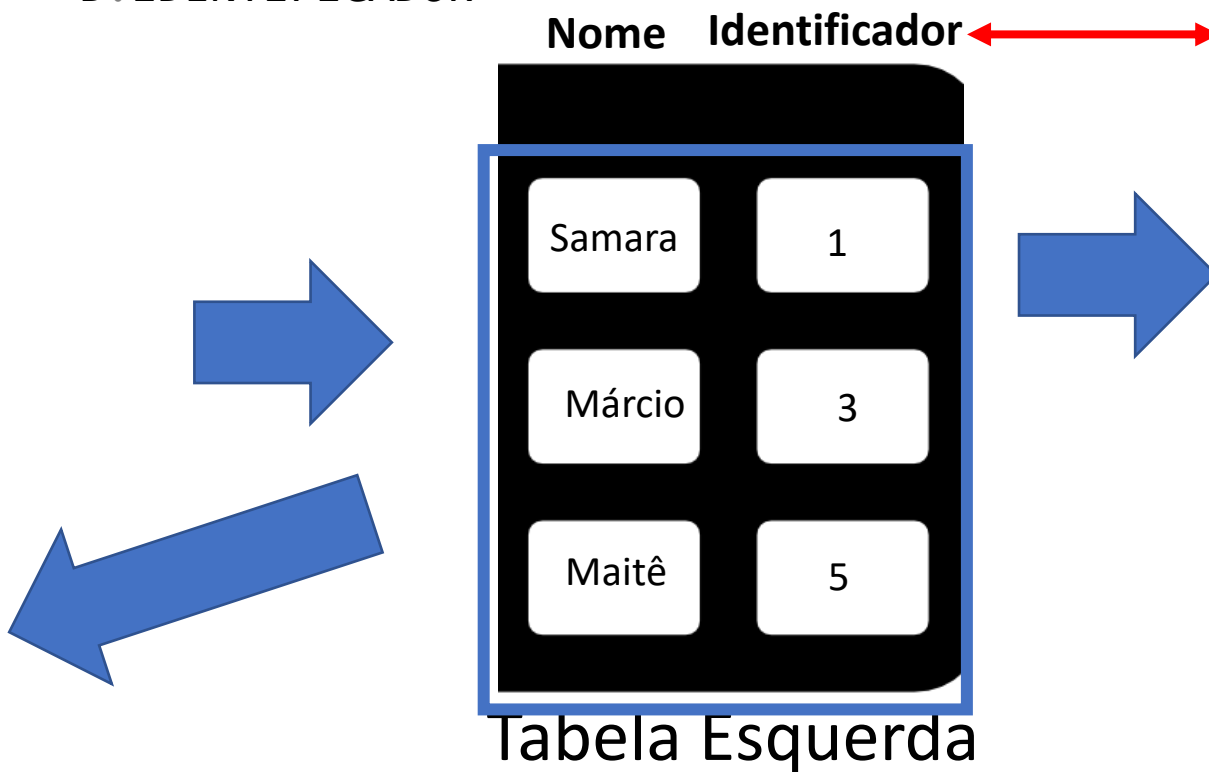
Nome	Identificador
Samara	Praia
Márcio	Futebol
Maitê	NULL

Nome	Identificador
Samara	1
Márcio	3
Maitê	5

Tabela Esquerda

Identificador	Hobby
1	Praia
3	Futebol
4	Música

Tabela Direita



Usando RIGHT JOIN

```
SELECT A.NOME, B.HOBBY FROM [TABELA ESQUERDA] A
RIGHT JOIN [TABELA DIREITA] B
ON A.IDENTIFICADOR = B.IDENTIFICADOR
```

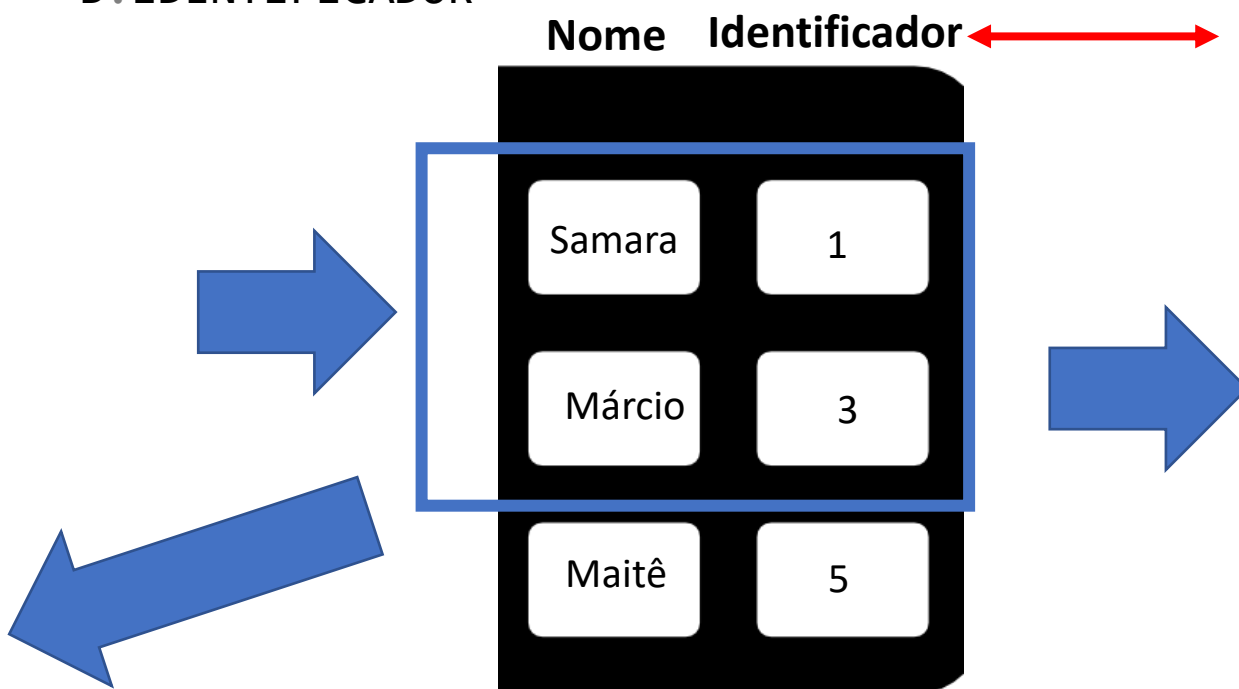
Nome	Identificador
Samara	Praia
Márcio	Futebol
NULL	Música

Nome	Identificador
Samara	1
Márcio	3
Maitê	5

Tabela Esquerda

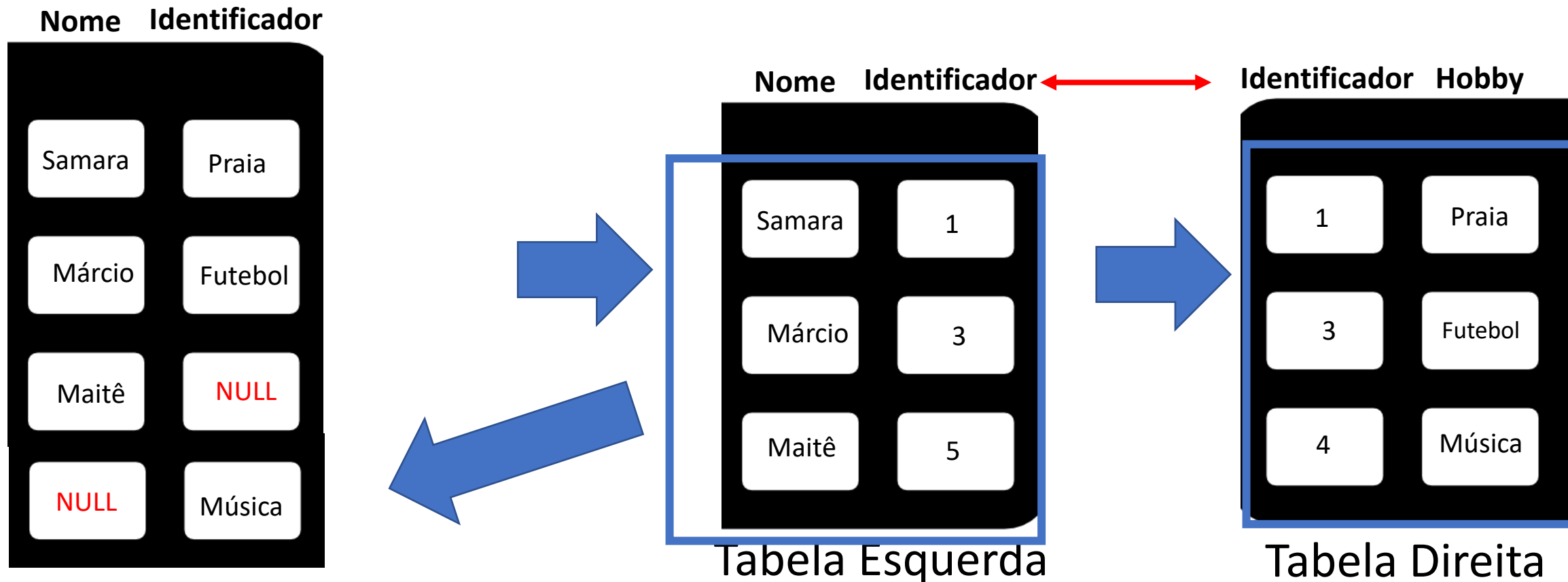
Identificador	Hobby
1	Praia
3	Futebol
4	Música

Tabela Direita



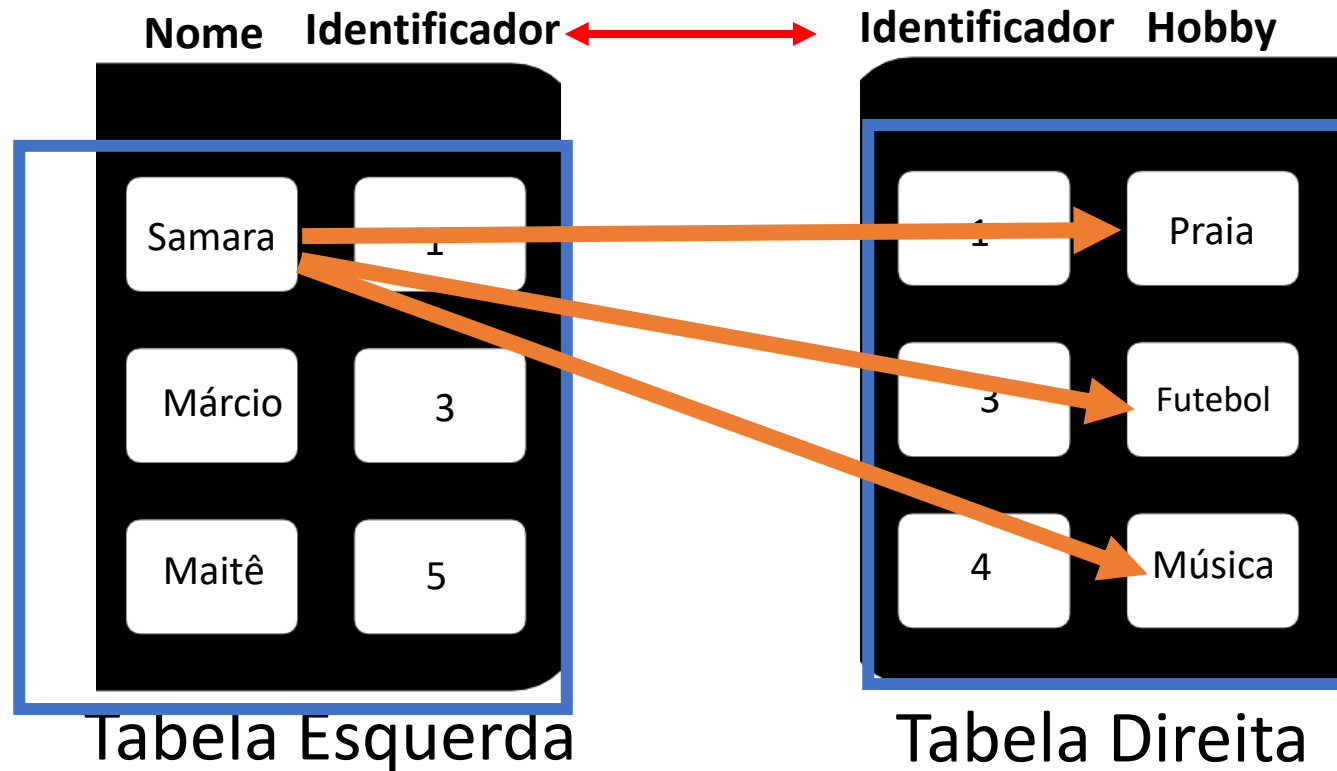
Usando FULL JOIN

```
SELECT A.NOME, B.HOBBY FROM [TABELA ESQUERDA] A
FULL JOIN [TABELA DIREITA] B
ON A.IDENTIFICADOR = B.IDENTIFICADOR
```



Usando CROSS JOIN

```
SELECT A.NOME, B.HOBBY FROM
[TABELA ESQUERDA] A CROSS JOIN [TABELA DIREITA] B
```



Usando UNION



Union – Faz união entre duas tabelas.

- Tabelas do banco de dados
- Resultantes de um JOIN
- Campos tem que ser do mesmo tipo
- Os campo tem que está na mesma ordem



Union – Na verdade o UNION junta duas consultas

Usando UNION

UNION –Ele aplica DISTINCT sobre os resultado final da consulta

Identificador	Hobby
1	Praia
3	Futebol
5	Fotografia
8	Artesanato






Identificador	Hobby
1	Praia
14	Computador
15	TV
18	Tênis



Identificador	Hobby
1	Praia
3	Futebol
5	Fotografia
8	Artesanato
14	Computador
15	TV
18	Tênis

Usando UNION ALL

UNION ALL – Não aplica DISTINCT sobre os resultado final da consulta

Identificador	Hobby
1	Praia
3	Futebol
5	Fotografia
8	Artesanato

Identificador	Hobby
1	Praia
14	Computador
15	TV
18	Tênis

Identificador	Hobby
1	Praia
3	Futebol
5	Fotografia
8	Artesanato
1	Praia
14	Computador
15	TV
18	Tênis

Subconsulta

Subconsulta – Podemos usar uma subconsulta dentro de uma consulta

X	Y
A	2
A	1
B	2
B	3
B	1
C	1
C	5
C	2

Z
1
2
3



4



5



```
SELECT X,Y FROM TAB1
WHERE Y IN(1,2)
```

```
SELECT X,Y FROM TAB1
WHERE Y IN(1,2,3)
```

```
SELECT X,Y FROM TAB1
WHERE Y IN(1,2,3,4)
```

```
SELECT X,Y FROM TAB1
WHERE Y IN(1,2,3,4,5)
```

Subconsulta

Subconsulta – Podemos usar uma subconsulta dentro de uma consulta

X	Y
A	2
A	1
B	2
B	3
B	1
C	1
C	5
C	2

Z
1
2
3

```
SELECT X,Y FROM TAB1
WHERE Y IN (SELECT Z FROM TAB2)
```

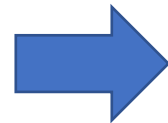


Subconsulta

Subconsulta – Podemos usar uma subconsulta dentro de uma consulta

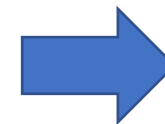
X	Y
A	2
A	1
B	2
B	3
B	1
C	1
C	5
C	2
D	3

```
SELECT X, SUM(Y) AS NEW_Y FROM TAB1
GROUP BY X
```



X	NEW Y
A	3
B	6
C	8
D	3

```
SELECT Z.X, Z.NEW_Y FROM (SELECT X, SUM(Y)
AS NEW_Y FROM TAB1 GROUP BY X)Z
WHERE Z.NEW_Y = 3
```



X	NEW Y
A	3
D	3

DESAFIO



Desafio

Qual seria a consulta, usando subconsulta, que seria equivalente a:

```
SELECT CPF, COUNT(*) FROM [TABELA DE NOTAS FISCAIS]
WHERE YEAR(DATA) = 2016
GROUP BY CPF
HAVING COUNT(*) > 2000
```



```
SELECT X.CPF, X.CONTAGEM FROM
(SELECT CPF, COUNT(*) AS 'CONTAGEM' FROM [TABELA DE
NOTAS FISCAIS]
WHERE YEAR(DATA) = 2016
GROUP BY CPF) X
WHERE X.CONTAGEM > 2000
```

View (Visão)

View – É uma tabela lógica, resultado de uma consulta que pode ser usada depois e, qualquer outra consulta

X	Y
A	2
A	1
B	2
B	3
B	1
C	1
C	5
C	2
D	4

```
SELECT X, SUM(Y) AS NEW_Y FROM TAB1
GROUP BY X
```

X	NEW Y
A	3
B	6
C	8
D	4

Criamos uma visão chamada
VW_VIEW

X	NEW Y
A	3
B	6
C	8
D	4

VW_VIEW

View (Visão)

```
SELECT * FROM VW_VIEW
```

X	NEW Y
A	3
B	6
C	8
D	4

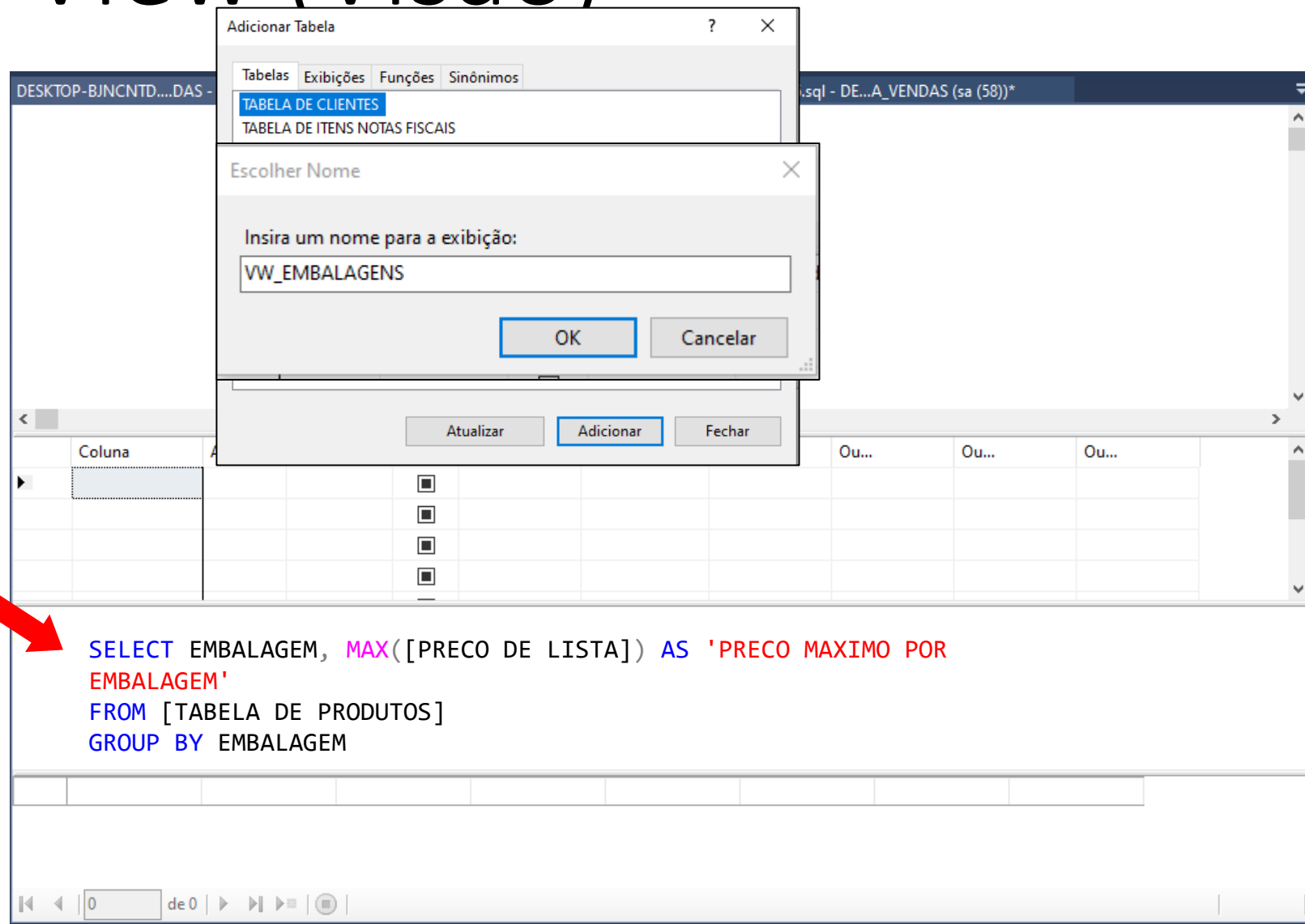
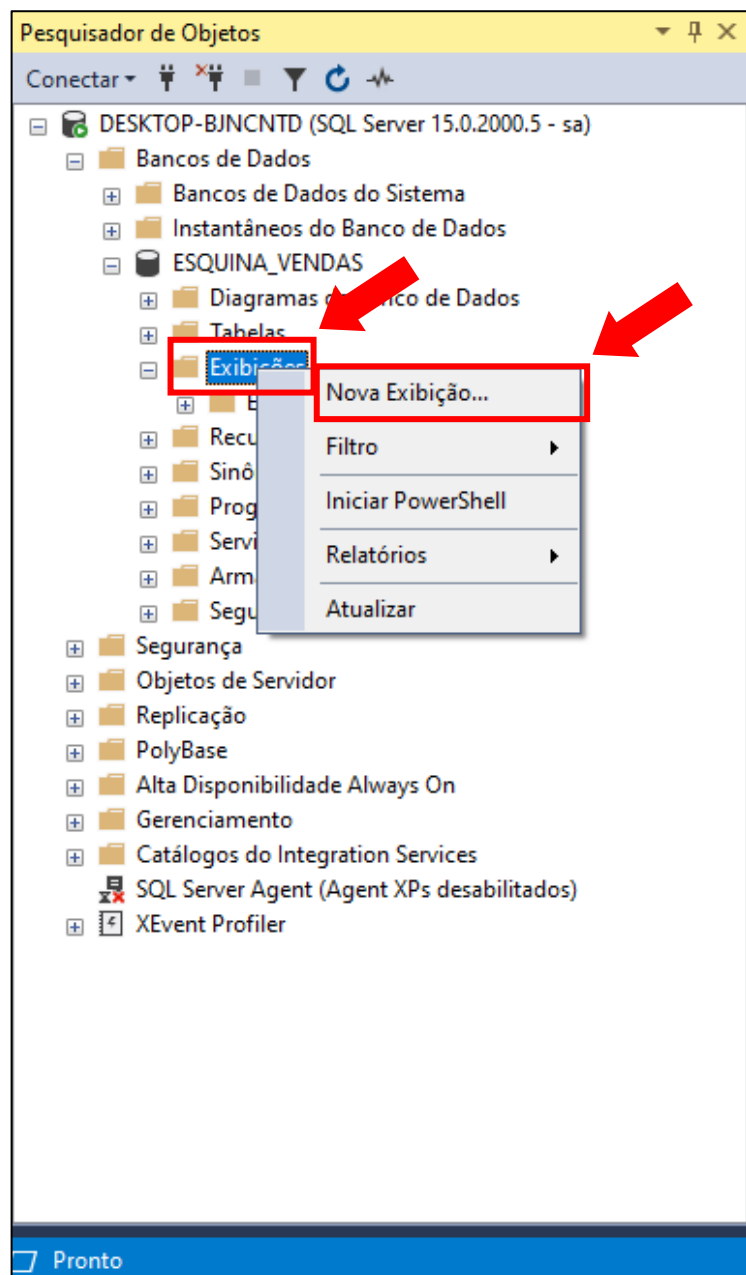
VW_VIEW

W	Y
F	3
G	6
H	9
I	9
J	3
K	6
L	3
M	3
N	4

TAB3

```
SELECT VW_VIEW.X, TAB3.W FROM VW_VIEW
INNER JOIN TAB3
ON VW_VIEW.NEW_Y = TAB3.Y
```

View (Visão)



View (Visão)

DESKTOP-BJNCNT...o.VW_EMBALAGENS* X DESKTOP-BJNCNTD....DAS - dbo.View_1 SQLQuery7.sql - DE...A_VENDAS (sa (59))*

TABELA DE PRODUTOS

- ☐ * (Todas as Colunas)
- ☐ CODIGO DO PRODUTO
- ☐ NOME DO PRODUTO
- ☒ EMBALAGEM $\{=$
- ☐ TAMANHO
- ☐ SABOR
- ☐ PRECO DE LISTA Σ

	Coluna	Alias	Tabela	Saída	Tipo de Class...	Ordem de Cla...	Agrupar por	Filtrar	Ou...	Ou...	Ou...
▶	EMBALAGEM		[TABELA D...	<input checked="" type="checkbox"/>			Agrupar por				
	[PRECO DE LI...	'PREC...	[TABELA D...	<input checked="" type="checkbox"/>			Max				
				<input type="checkbox"/>							
				<input type="checkbox"/>							

```

SELECT  EMBALAGEM, MAX([PRECO DE LISTA]) AS 'PRECO MAXIMO POR EMBALAGEM'
FROM    dbo.[TABELA DE PRODUTOS]
GROUP BY EMBALAGEM
    
```

de 0

View (Visão)

```
CREATE VIEW [NOME DA VIEW] AS
```

```
[CONSULTA A SER SALVA NA VIEW]
```


O que aprendemos nesta aula

- A cláusula CASE para classificar registros

- Como juntar tabelas usando diversos tipos de JOIN

- Como unir consultas com os campos do mesmo tipo

- A cláusula HAVING, para filtrar dados agrupados



- Como usarmos Subconsultas e Views