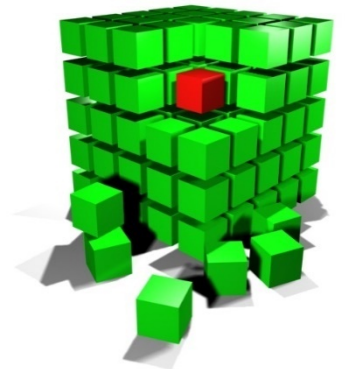


- **Aula 012 – Linguagem SQL**
 - Arquivos CSV.
 - Importe em massa.
 - JOINS.



■ Criação de arquivos CSV

- Para podermos realizar os **JOINS**, precisamos inserir dados nas demais tabelas de nosso banco de dados.
- Podemos utilizar um comando **INSERT** para realizar essa operação. Porém, dependendo da quantidade de dados, a utilização desse comando pode ser um pouco complicada. Dessa forma, utilizaremos um **importe em massa**.



■ Criação de arquivos CSV

- O formato **CSV**, ou **Comma-Separated Values**, é um formato simples de armazenamento, que agrupa informações em arquivos de texto, os quais podem ser utilizados para a troca de dados com banco de dados, planilhas, ou entre outros aplicativos.
- Cada linha em um texto CSV representa uma linha em uma planilha. Cada célula é geralmente separada por um caractere de tabulação, como uma vírgula, um espaço, dentre outros.



■ Listagem do nome das tabelas

- Podemos listar o nome das tabelas em nosso banco de dados utilizando:

-- Exibe o nome das tabelas que
-- existem no banco de dados em uso

SELECT name

FROM sys.tables

GO



- **Criação de arquivos CSV**
 - Para poder popular os dados de nossas tabelas, utilizaremos o **Microsoft Excel**.
 - Qualquer outro programa de planilha eletrônica pode ser utilizado. Também podemos realizar essa tarefa utilizando um editor de textos simples, como o **Bloco de Notas**, dentre outros.



■ Criação de arquivos CSV

- Como demonstração, será utilizada a tabela **DEPENDENTES**. Essa tabela é composta pelos seguintes campos:

1. Cod_Dependente
2. Nome_Dependente
3. Sexo_Dependente
4. Data_Nascimento

5. Cod_Funcionario



Chave Estrangeira



■ Tabela DEPENDENTES

-- Cria a tabela DEPENDENTES

```
CREATE TABLE DEPENDENTES (  
    Cod_Dependente    INT    PRIMARY KEY,  
    Nome_Dependente   VARCHAR(35),  
    Sexo_Dependente   CHAR(1),  
    Data_Nascimento   DATE,  
    ➡ Cod_Funcionario  INT    FOREIGN KEY REFERENCES  
    FUNCIONARIOS (ID)  
)  
  
GO
```



■ Verificando as restrições existentes

-- Exibe informações sobre as restrições da
-- tabela FUNCIONARIOS

```
EXEC sp_helpconstraint FUNCIONARIOS  
GO
```

-- Exibe informações sobre as restrições da
-- tabela DEPENDENTES

```
EXEC sp_helpconstraint DEPENDENTES  
GO
```



■ Verificando as restrições existentes

Object Name							
1	FUNCIONARIOS						
Captura Retangular							
	constraint_type	constraint_name	delete_action	update_act	status_enabled	status_for_replication	constraint_k
1	PRIMARY KEY (clustered)	PK_FUNCIONA__3214EC273F1C60BC	(n/a)	(n/a)	(n/a)	(n/a)	ID
Table is referenced by foreign key							
1	Aula09.dbo.DEPENDENTES: FK_DEPENDENT__Cod_F__4BAC3						
Object Name							
1	DEPENDENTES						
	constraint_type	constraint_name	delete_action	update_act	status_enabled	status_for_replication	constraint_keys
1	FOREIGN KEY	FK_DEPENDENT__Cod_F__4BAC3F29	No Action	No Action	Enabled	Is_For_Replication	Cod_Funcionario
2							REFERENCES Aula09.dbo.FUNCIONARIOS (ID)
3	PRIMARY KEY (clustered)	PK_DEPENDEN__ED37A040C6668A	(n/a)	(n/a)	(n/a)	(n/a)	Cod_Dependente



Observe que o comando `sp_helpconstraint` exibe as restrições de chaves primárias e estrangeiras, existentes entre as tabelas.



■ Dados – Tabela DEPENDENTES

dependentes.xlsx - Excel

Arquivo Página Inicial Inserir Layout da Página Fórmulas Dados Revisão Exibir Desenvolvedor Suplementos Ajuda Equipe Diga-me o que você deseja fazer Compartilhar

Recortar Colar Área de Transferência Recortar Copiar Pincel de Formatação

Calibri 11 Fonte

Quebrar Texto Automaticamente Alinhamento

Mesclar e Centralizar

Geral Número

Formatação Condicional Estilos

Formatar como Tabela Estilos de Célula

Inserir Excluir Formatar Células

AutoSoma Preencher Limpar

Classificar Localizar e Filtrar Selecionar Edição

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Cod_Dependente	Nome_Dependente	Sexo_Dependente	Data_Nascimento	Cod_Funcionario										
1															
2	1	Ana Clara	F	21/09/2015	9										
3	2	Maria Antonieta	F	14/10/2000	5										
4	3	João Pereira	M	12/05/2004	2										
5	4	Pedro Silva	M	12/12/1998	1										
6	5	Ana Cristina	F	16/06/2000	3										
7	6	Lucas Santos	M	26/04/1999	10										
8	7	Érika Santos	F	27/09/2001	10										
9	8	Marcelo Santos	M	11/11/2003	10										
10	9	Jorge Lucas	M	04/02/2007	7										
11	10	José Lima	M	30/09/2001	8										
12	11	Maria Lima	F	01/10/2013	8										
13	12	Pedro Lima	M	01/10/2013	8										
14	13	Carlos Antônio	M	03/08/2015	4										
15	14	Sheila Mariana	F	04/03/1997	3										
16	15	Lilian Maria	F	12/08/2003	3										
17	16	Melissa Pereira	F	06/10/2003	2										
18	17	Larissa Souza	F	07/05/1998	6										
19	18	Edson Souza	M	14/01/2000	6										
20	19	Miriam Muniz	F	09/11/2003	5										
21	20	Antônio Romão	M	10/10/1998	1										

Dependentes

Pronto

110%

■ Criação de arquivos CSV

Salvar como

Recente

OneDrive

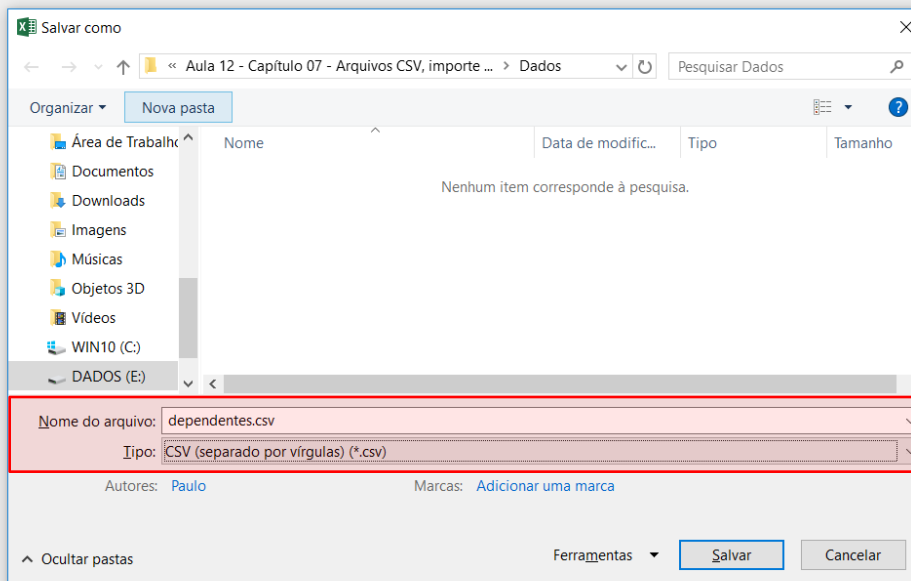
Este PC

Adicionar um Local

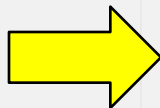
Procurar

Documentos

Área de Trabalho

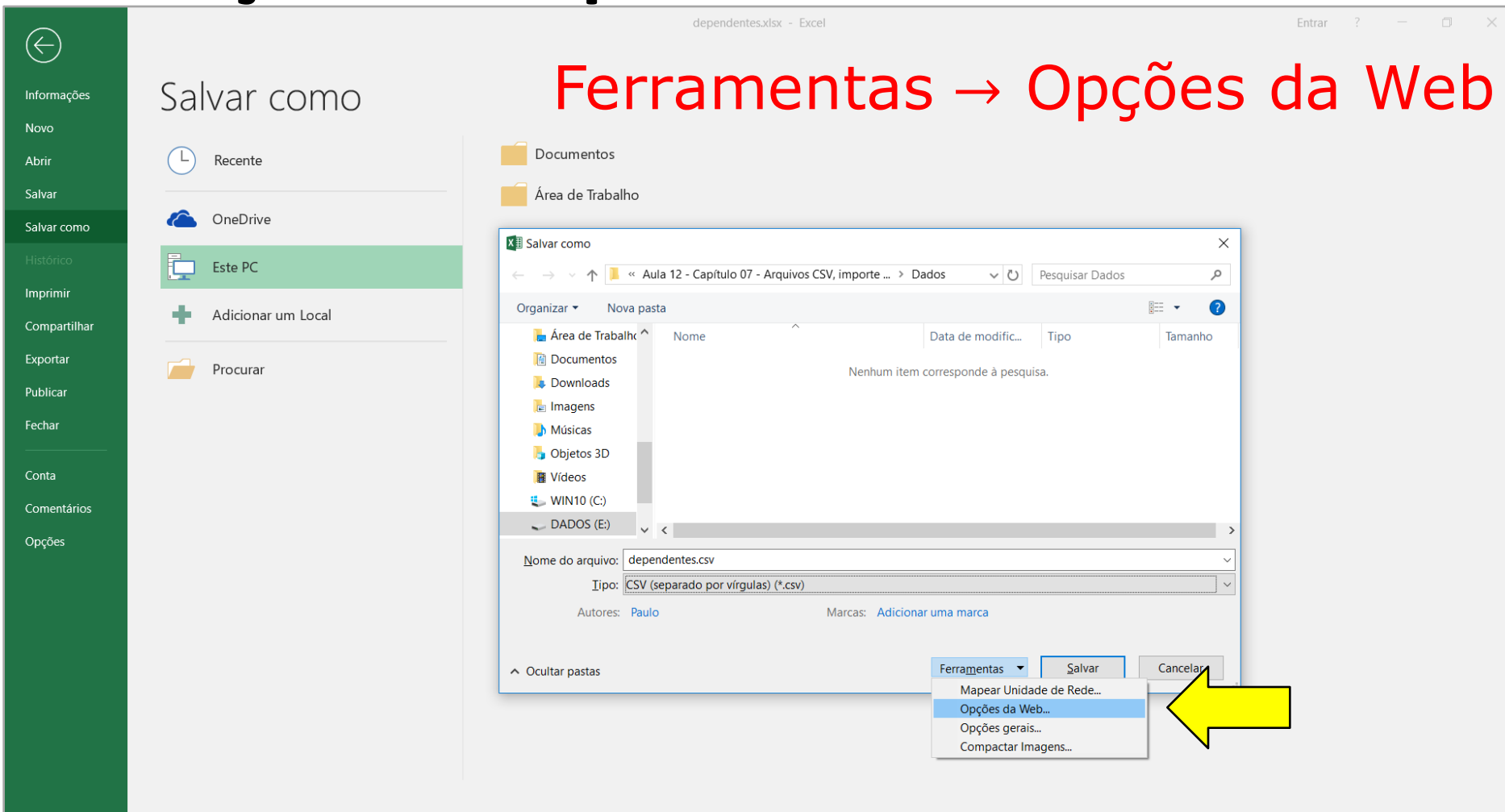


CSV



■ Criação de arquivos CSV

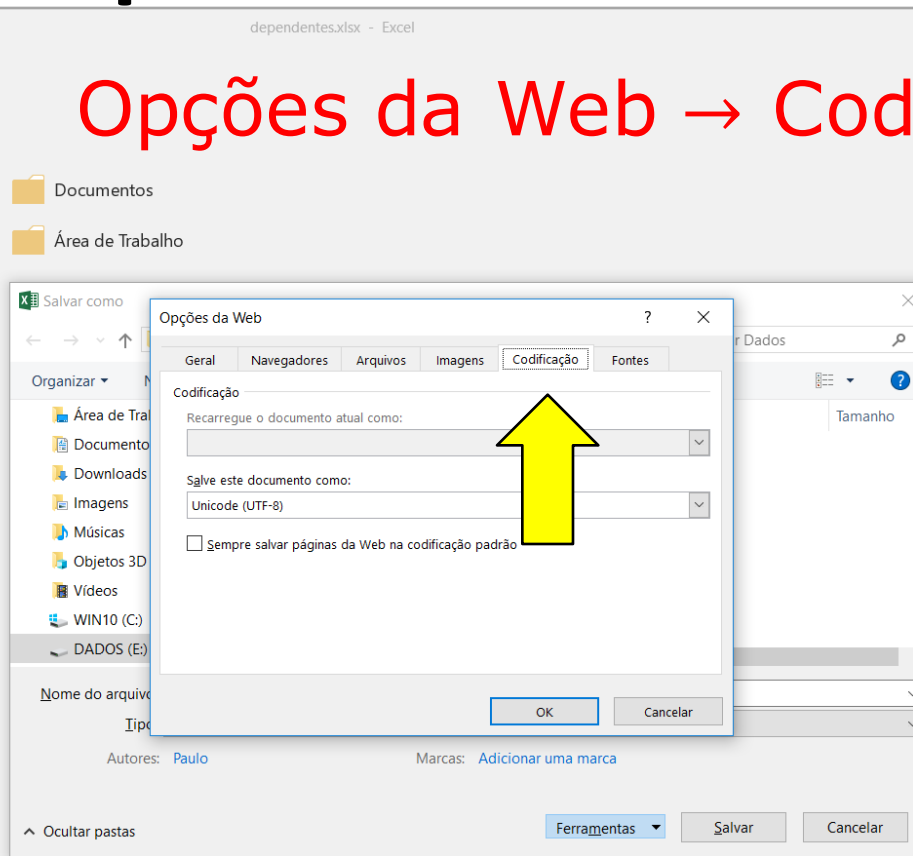
Ferramentas → Opções da Web



The screenshot shows the Microsoft Excel interface with the 'dependentes.xlsx' file open. The 'Salvar como' (Save As) dialog box is displayed, showing the file path 'Aula 12 - Capítulo 07 - Arquivos CSV, importe ... > DADOS'. The file name is 'dependentes.csv' and the type is 'CSV (separado por vírgulas) (*.csv)'. A yellow arrow points to the 'Ferramentas' (Tools) dropdown menu, which is open, showing the option 'Opções da Web...' (Web Options...).

■ Criação de arquivos CSV

Opções da Web → Codificação



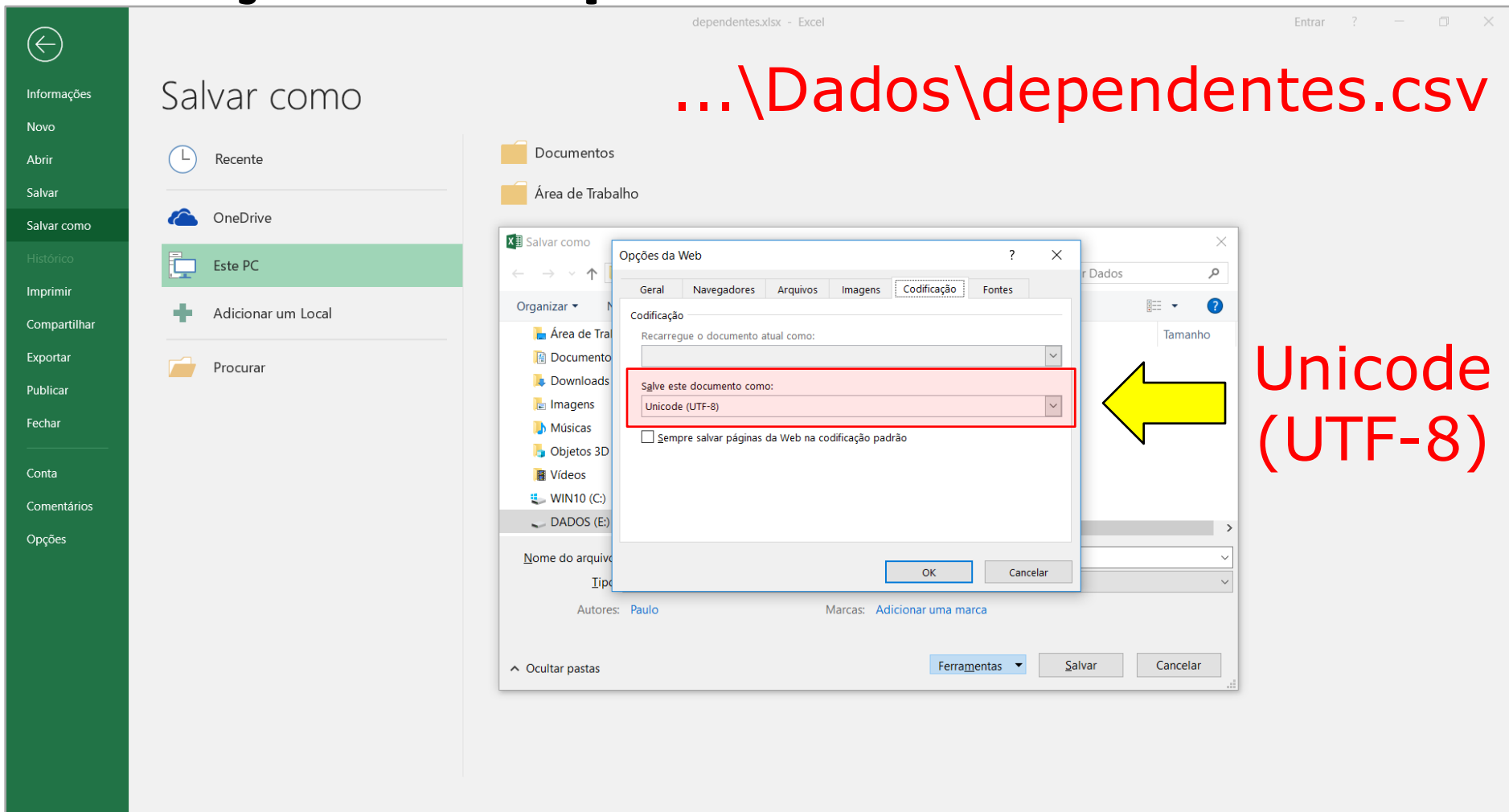
■ Criação de arquivos CSV

dependentes.xlsx - Excel

Entrar ? - + X

... \Dados\dependentes.csv

Unicode (UTF-8)



Salvar como

Recente

OneDrive

Este PC

Adicionar um Local

Procurar

Organizar

Área de Trabalho

Documentos

Downloads

Imagens

Músicas

Objetos 3D

Vídeos

WIN10 (C:)

DADOS (E:)

Nome do arquivo

Ícone

Autores: Paulo

Marcas: Adicionar uma marca

Ocultar pastas

Ferramentas

Salvar

Cancelar

Opções da Web

Geral Navegadores Arquivos Imagens Codificação Fontes

Codificação

Recarregue o documento atual como:

Salve este documento como:

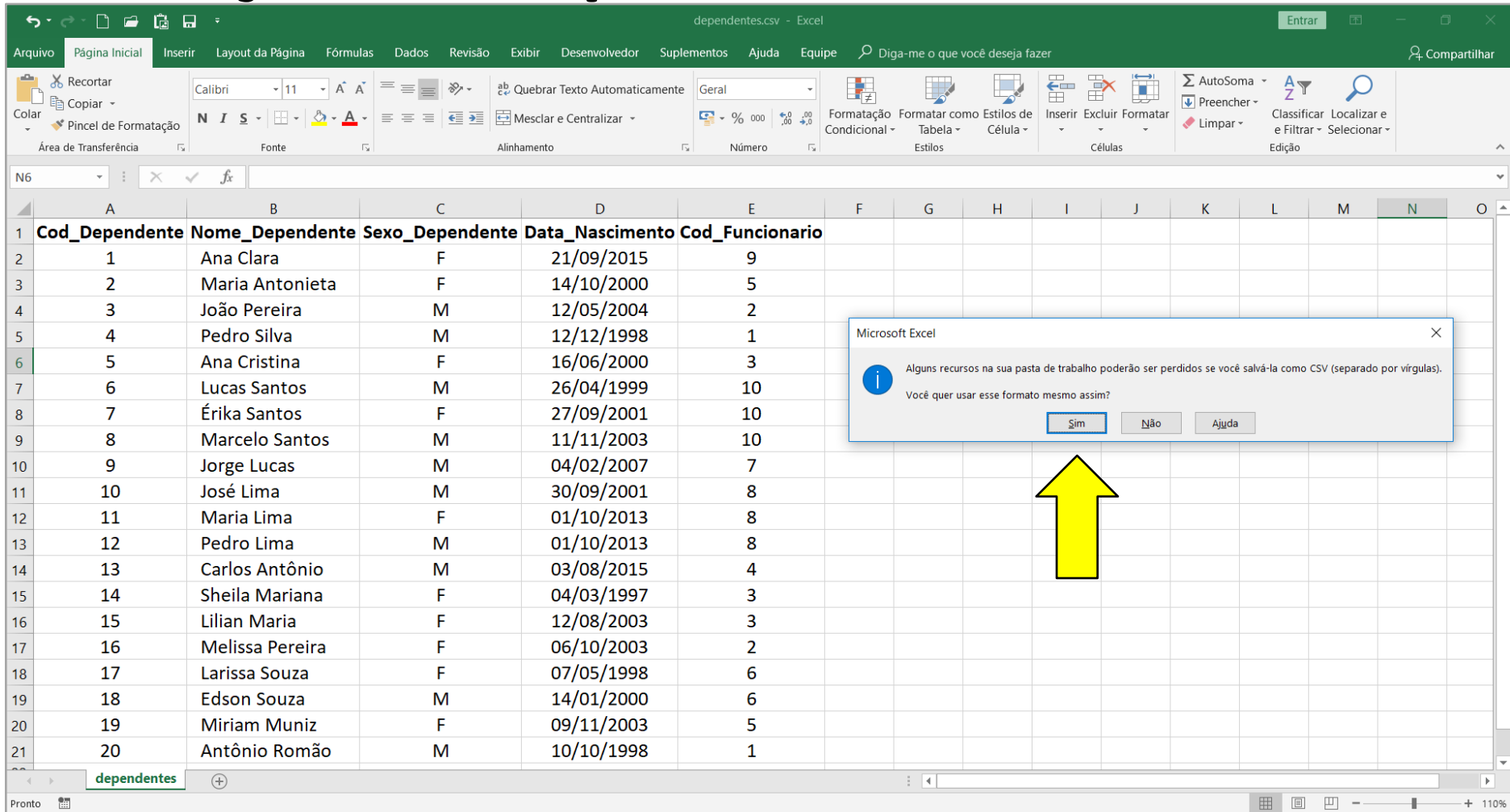
Unicode (UTF-8)

☐ Sempre salvar páginas da Web na codificação padrão

OK

Cancelar

■ Criação de arquivos CSV



The screenshot shows the Microsoft Excel interface with a file named 'dependentes.csv'. The data is organized in a table with the following columns:

Cod_Dependente	Nome_Dependente	Sexo_Dependente	Data_Nascimento	Cod_Funcionario
1	Ana Clara	F	21/09/2015	9
2	Maria Antonieta	F	14/10/2000	5
3	João Pereira	M	12/05/2004	2
4	Pedro Silva	M	12/12/1998	1
5	Ana Cristina	F	16/06/2000	3
6	Lucas Santos	M	26/04/1999	10
7	Érika Santos	F	27/09/2001	10
8	Marcelo Santos	M	11/11/2003	10
9	Jorge Lucas	M	04/02/2007	7
10	José Lima	M	30/09/2001	8
11	Maria Lima	F	01/10/2013	8
12	Pedro Lima	M	01/10/2013	8
13	Carlos Antônio	M	03/08/2015	4
14	Sheila Mariana	F	04/03/1997	3
15	Lilian Maria	F	12/08/2003	3
16	Melissa Pereira	F	06/10/2003	2
17	Larissa Souza	F	07/05/1998	6
18	Edson Souza	M	14/01/2000	6
19	Miriam Muniz	F	09/11/2003	5
20	Antônio Romão	M	10/10/1998	1

A dialog box titled 'Microsoft Excel' is open, displaying the message: 'Alguns recursos na sua pasta de trabalho poderão ser perdidos se você salvá-la como CSV (separado por vírgulas). Você quer usar esse formato mesmo assim?'. The dialog box has three buttons: 'Sim', 'Não', and 'Ajuda'. A large yellow arrow points to the 'Sim' button.

■ Criação de arquivos CSV

Cod_Dependente;Nome_Dependente;Sexo_Dependente;Data_Nascimento;Cod_Funcionario

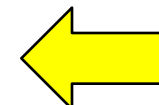
1;Ana Clara;F;21/09/2015;9
2;Maria Antonieta;F;14/10/2000;5
3;João Pereira;M;12/05/2004;2
4;Pedro Silva;M;12/12/1998;1
5;Ana Cristina;F;16/06/2000;3
6;Lucas Santos;M;26/04/1999;10
7;Érika Santos;F;27/09/2001;10
8;Marcelo Santos;M;11/11/2003;10
9;Jorge Lucas;M;04/02/2007;7
10;José Lima;M;30/09/2001;8
11;Maria Lima;F;01/10/2013;8
12;Pedro Lima;M;01/10/2013;8
13;Carlos Antônio;M;03/08/2015;4
14;Sheila Mariana;F;04/03/1997;3
15;Lilian Maria;F;12/08/2003;3
16;Melissa Pereira;F;06/10/2003;2
17;Larissa Souza;F;07/05/1998;6
18;Edson Souza;M;14/01/2000;6
19;Miriam Muniz;F;09/11/2003;5
20;Antônio Romão;M;10/10/1998;1

Conteúdo do arquivo, visualizado dentro do software Notepad++.



■ Criação de arquivos CSV

Cod_Dependente;Nome_Dependente;Sexo_Dependente;Data_Nascimento;Cod_Funcionario



Cabeçalho

1;Ana Clara;F;21/09/2015;9

2;Maria Antonieta;F;14/10/2000;5

3;João Pereira;M;12/05/2004;2

4;Pedro Silva;M;12/12/1998;1

5;Ana Cristina;F;16/06/2000;3

6;Lucas Santos;M;26/04/1999;10

7;Érika Santos;F;27/09/2001;10

8;Marcelo Santos;M;11/11/2003;10

9;Jorge Lucas;M;04/02/2007;7

10;José Lima;M;30/09/2001;8

11;Maria Lima;F;01/10/2013;8

12;Pedro Lima;M;01/10/2013;8

13;Carlos Antônio;M;03/08/2015;4

14;Sheila Mariana;F;04/03/1997;3

15;Lilian Maria;F;12/08/2003;3

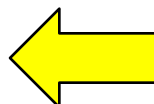
16;Melissa Pereira;F;06/10/2003;2

17;Larissa Souza;F;07/05/1998;6

18;Edson Souza;M;14/01/2000;6

19;Miriam Muniz;F;09/11/2003;5

20;Antônio Romão;M;10/10/1998;1



Linhas da Planilha



■ Importe em massa

- O **Microsoft SQL Server** oferece suporte à exportação e à importação de dados em massa.
- A **exportação em massa** se refere à cópia de dados de uma tabela do SQL Server, para dentro de um arquivo de dados.
- A **importação em massa** se refere ao carregamento de dados de um arquivo, para dentro de uma tabela do SQL Server.



■ Importe em massa

Método	Descrição
<code>bcp</code>	Um utilitário de linha de comando que exporta e importa dados em massa e gera arquivos de formato.
<code>BULK INSERT</code>	Uma instrução T-SQL que importa dados diretamente de um arquivo de dados para uma tabela de banco de dados.
<code>INSERT... SELECT * FROM OPENROWSET (BULK...)</code>	Uma instrução T-SQL que usa o provedor de conjunto de linhas em massa <code>OPENROWSET</code> para importação em massa dos dados para uma tabela do SQL Server especificando a função <code>OPENROWSET (BULK...)</code> para selecionar dados em uma instrução <code>INSERT</code> .



■ O comando BULK INSERT

BULK INSERT

```
[ database_name . [ schema_name ] . | schema_name . ] [ table_name | view_name ]
FROM 'data_file'
[ WITH
(
[ [ , ] BATCHSIZE = batch_size ]
[ [ , ] CHECK_CONSTRAINTS ]
[ [ , ] CODEPAGE = { 'ACP' | 'OEM' | 'RAW' | 'code_page' } ]
[ [ , ] DATAFILETYPE =
{ 'char' | 'native' | 'widechar' | 'widenative' } ]
[ [ , ] FIELDTERMINATOR = 'field_terminator' ]
[ [ , ] FIRSTROW = first_row ]
[ [ , ] FIRE_TRIGGERS ]
[ [ , ] FORMATFILE = 'format_file_path' ]
[ [ , ] KEEPIDENTITY ]
[ [ , ] KEEPNULLS ]
[ [ , ] KILOBYTES_PER_BATCH = kilobytes_per_batch ]
[ [ , ] LASTROW = last_row ]
[ [ , ] MAXERRORS = max_errors ]
[ [ , ] ORDER ( { column [ ASC | DESC ] } [ ,...n ] ) ]
[ [ , ] ROWS_PER_BATCH = rows_per_batch ]
[ [ , ] ROWTERMINATOR = 'row_terminator' ]
[ [ , ] TABLOCK ]
[ [ , ] ERRORFILE = 'file_name' ]
) ]
```

← Sintaxe



■ O comando BULK INSERT

- **CODEPAGE** – define a página de códigos a ser usada na importação e, por consequência, se os caracteres vêm acentuados ou não;
- **MAXERRORS** – define a quantidade máxima de erros que pode ocorrer numa importação de dados;
- **FIELDTERMINATOR** – caractere usado para separar uma coluna da outra (o default é o caractere **TAB**, representado por `\t`);
- **ROWTERMINATOR** – caractere usado para separar uma linha da outra (o default é o **ENTER**, representado por `\n`);
- **FIRE_TRIGGERS** – se informado, dispara as triggers na execução do comando;
- **KEEP_IDENTITY** – se informado, impede que a numeração automática (coluna **IDENTITY** da tabela) seja usada;
- **FORMATFILE** – define qual o arquivo de formato será usado na importação;
- **FIRSTROW** – define qual será a primeira linha importada;
- **LASTROW** – define qual será a última linha importada.



■ Tipos suportados pelo BULK INSERT

DATAFILETYPE	Descrição
<code>char</code>	Formato de caractere. É o tipo de dados padrão.
<code>native</code>	Tipos de dados nativo. O valor <code>native</code> oferece uma alternativa de alto desempenho ao valor <code>char</code> .
<code>widechar</code>	Permite o uso de caracteres do tipo Unicode.
<code>widenative</code>	Tipos de dados nativos, exceto em colunas do tipo <code>CHAR</code> , <code>VARCHAR</code> , e <code>TEXT</code> , na qual os dados são armazenados como Unicode. O valor <code>widenative</code> oferece uma alternativa de alto desempenho para <code>widechar</code> . Se o arquivo de dados contiver caracteres estendidos no padrão ANSI, especifique <code>widenative</code> .

■ Códigos de página do BULK INSERT

CODEPAGE	Descrição
ACP	Colunas de CHAR , VARCHAR , ou TEXT são convertidos a partir da página de código do Windows (ISO 1252) para a página de código do SQL Server.
OEM	Colunas de CHAR , VARCHAR , ou TEXT são convertidos da página de código OEM do sistema para a página de código do SQL Server. É o valor padrão.
RAW	Nenhuma conversão de uma página de código em outra ocorre. Essa é a opção mais rápida.
code_page	Número de página de código específico, por exemplo, 850.



■ O comando BULK INSERT

-- Altera o formato de data do SQL Server

SET DATEFORMAT DMY

GO

Para não obtermos resultados indesejados, devemos alterar o formato de entrada das datas, quando for necessário.

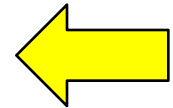


■ O comando BULK INSERT

- Insere os dados na tabela DEPENDENTES, utilizando os
- dados provenientes do arquivo 'dependentes.csv'

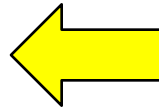
BULK INSERT DEPENDENTES

FROM N'E:\BD2\...\Dados\dependentes.csv'



WITH (

FIRSTROW = 2,



DATAFILETYPE = 'char',

FIELDTERMINATOR = ';',

CODEPAGE = 'ACP'

)

GO



■ O comando BULK INSERT

Cod_Dependente	Nome_Dependente	Sexo_Dependente	Data_Nascimento	Cod_Funcionario
1	Ana Clara	F	2015-09-21	9
2	Maria Antonieta	F	2000-10-14	5
3	João Pereira	M	2004-05-12	2
4	Pedro Silva	M	1998-12-12	1
5	Ana Cristina	F	2000-06-16	3
6	Lucas Santos	M	1999-04-26	10
7	Érika Santos	F	2001-09-27	10
8	Marcelo Santos	M	2003-11-11	10
9	Jorge Lucas	M	2007-02-04	7
10	José Lima	M	2001-09-30	8
11	Maria Lima	F	2013-10-01	8
12	Pedro Lima	M	2013-10-01	8
13	Carlos Antônio	M	2015-08-03	4
14	Sheila Mariana	F	1997-03-04	3
15	Lilian Maria	F	2003-08-12	3
16	Melissa Pereira	F	2003-10-06	2
17	Larissa Souza	F	1998-05-07	6
18	Edson Souza	M	2000-01-14	6
19	Miriam Muniz	F	2003-11-09	5
20	Antônio Romão	M	1998-10-10	1

```
SELECT *
FROM DEPENDENTES
GO
```

(20 row(s) affected)



■ Junções de tabelas – JOINS

- A capacidade de combinar os dados de diferentes tabelas, utilizando atributos comuns entre elas, talvez seja a distinção mais importante entre o banco de dados relacional e os outros modelos de banco de dados.
- Para realizar a operação de junção de tabelas inserimos seus nomes na cláusula **FROM** de um comando **SELECT** e realizamos comparações entre seus atributos comuns, geralmente a chave primária e a chave estrangeira, utilizando condições de junção, ou **JOINS**.



■ Junções de tabelas – JOINS

- O **JOIN** é uma cláusula da linguagem SQL que permite criar consultas combinando resultados de uma ou mais tabelas, por meio da comparação dos valores comuns existentes entre as colunas de cada uma delas.
- No caso de uma única tabela, a operação de junção costuma ser denominada de **SELF JOIN**.



■ Junções de tabelas – JOINS

- Para entender melhor os JOINS e sua aplicação, podemos visualizar as tabelas como conjuntos e os registros como elementos de cada um desses conjuntos.
- Por meio dos JOINS, podemos consultar os elementos de cada um desses conjuntos e as respectivas **intersecções**, **uniões** e **diferenças** existentes entre eles.



■ Junções de tabelas – JOINS

- A sintaxe básica de um JOIN é dada por:

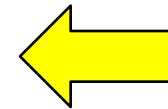
SELECT Campo_1,

... ,

Campo_N

Tipo de JOIN

FROM Tabela_1 **JOIN** Tabela_2



ON Tabela_1.Campo_Comum =
Tabela_2.Campo_Comum

GO



■ Junções de tabelas – JOINS

- A sintaxe básica de um JOIN é dada por:

SELECT Campo_1,

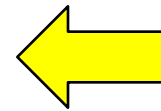
... ,

Campo_N

Campos utilizados
para a comparação

FROM Tabela_1 **JOIN** Tabela_2

ON Tabela_1.Campo_Comum =
Tabela_2.Campo_Comum



GO



■ Junções de tabelas – JOINS

- A operação de junção relacional utiliza o conteúdo das linhas de duas tabelas e retorna aquelas com base em uma das seguintes condições:
 - Tenham valores comuns em colunas comuns (**junção natural**);
 - Atendam a uma determinada condição de junção (**igualdade** ou **desigualdade**);
 - Tenham valores comuns em colunas comuns ou não tenham valores correspondentes (**junção externa**).



■ Junções de tabelas – JOINS

- As operações de junção realizadas em tabelas relacionais podem ser classificadas de duas maneiras: **internas** ou **externas**.
- A **junção interna** é a operação tradicional, que retorna somente as linhas cujo atributos utilizados para a comparação atendam a determinados critérios.
- A **junção externa** retorna todas as linhas, independentemente do critério de comparação ser correspondido.



■ Junções de tabelas – JOINS

- Os critérios de comparação utilizados em uma junção podem ser uma condição de igualdade, também chamada de junção natural (**natural join**) ou equitativa (**equijoin**), ou uma condição de desigualdade, também chamada de junção teta (**theta join**).
- O padrão SQL também introduz um tipo especial de junção, que retorna o resultado do produto cartesiano entre tabelas (**cross join**).



■ Junções de tabelas – JOINS

Classificação	Tipo	Exemplo	Descrição
CROSS (Cruzada)	CROSS JOIN (Antigo)	SELECT * FROM T1, T2	Retorna o produto cartesiano de T1 e T2.
	CROSS JOIN	SELECT * FROM T1 CROSS JOIN T2	Retorna o produto cartesiano de T1 e T2.



■ Junções de tabelas – JOINS

Classificação	Tipo	Exemplo	Descrição
INNER (Interna)	JOIN (Antigo)	<pre>SELECT * FROM T1, T2 WHERE T1.C1 = T2.C2</pre>	Retorna as linhas que atendam à condição de junção na cláusula WHERE . São selecionadas apenas as linhas com valores iguais dentro das colunas comparadas.



■ Junções de tabelas – JOINS

Classificação	Tipo	Exemplo	Descrição
INNER (Interna)	NATURAL JOIN	<pre>SELECT * FROM T1 NATURAL JOIN T2</pre>	Retorna somente linhas com valores correspondentes nas colunas das tabelas T1 e T2. Essas colunas devem ter os mesmos nomes e tipos de dados similares.



■ Junções de tabelas – JOINS

Classificação	Tipo	Exemplo	Descrição
INNER (Interna)	JOIN USING	<pre>SELECT * FROM T1 JOIN T2 USING (C1)</pre>	Retorna apenas as linhas com valores correspondentes nas colunas indicadas pela cláusula USING . No caso, a coluna C1 deve existir em ambas as tabelas.



■ Junções de tabelas – JOINS

Classificação	Tipo	Exemplo	Descrição
INNER (Interna)	JOIN ON	<pre>SELECT * FROM T1 JOIN T2 ON T1.C1 = T2.C2</pre>	Retorna apenas as linhas com valores correspondentes nas colunas indicadas pela cláusula ON .



■ Junções de tabelas – JOINS

Classificação	Tipo	Exemplo	Descrição
OUTER (Externa)	LEFT JOIN (Junção externa à esquerda)	<pre>SELECT * FROM T1 LEFT OUTER JOIN T2 ON T1.C1 = T2.C2</pre>	Retorna linhas com valores correspondentes e inclui todas as colunas da tabela à esquerda (T1), sem que haja correspondências de seus valores dentro da tabela T2.



■ Junções de tabelas – JOINS

Classificação	Tipo	Exemplo	Descrição
OUTER (Externa)	RIGHT JOIN (Junção externa à direita)	<pre>SELECT * FROM T1 RIGHT OUTER JOIN T2 ON T1.C1 = T2.C2</pre>	Retorna linhas com valores correspondentes e inclui todas as colunas da tabela à direita (T2), sem que haja correspondências de seus valores dentro da tabela T1.



■ Junções de tabelas – JOINS

Classificação	Tipo	Exemplo	Descrição
OUTER (Externa)	FULL JOIN (Junção completa)	<pre>SELECT * FROM T1 FULL OUTER JOIN T2 ON T1.C1 = T2.C2</pre>	Retorna linhas com valores correspondentes ou não, entre as colunas C1 e C2 de ambas as tabelas



■ Junções de tabelas – JOINS

- Para realizar junções em mais de uma tabela, temos:

```
SELECT Campo_1,
```

```
    . . . ,
```

```
    Campo_N
```

```
FROM Tabela_1 JOIN Tabela_2
```

```
ON Tabela_1.Campo_Comum =  
   Tabela_2.Campo_Comum
```

```
JOIN Tabela_3
```

```
ON Tabela_1.Campo_Comum =  
   Tabela_3.Campo_Comum
```

← Primeiro subconjunto

GO



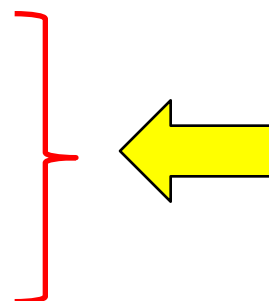
■ Junções de tabelas – JOINS

- Para realizar junções em mais de uma tabela, temos:

```
SELECT Campo_1,  
    . . . ,  
    Campo_N  
FROM Tabela_1 JOIN Tabela_2  
    ON Tabela_1.Campo_Comum =  
        Tabela_2.Campo_Comum  
JOIN Tabela_3  
    ON Tabela_1.Campo_Comum =  
        Tabela_3.Campo_Comum
```

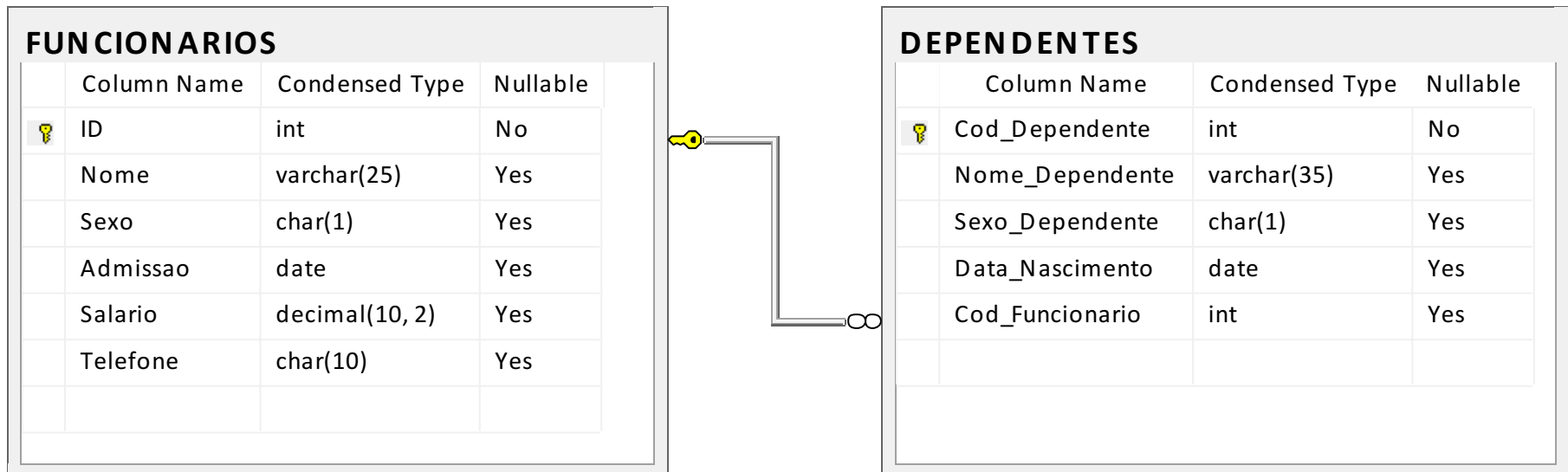
GO

Segundo
subconjunto

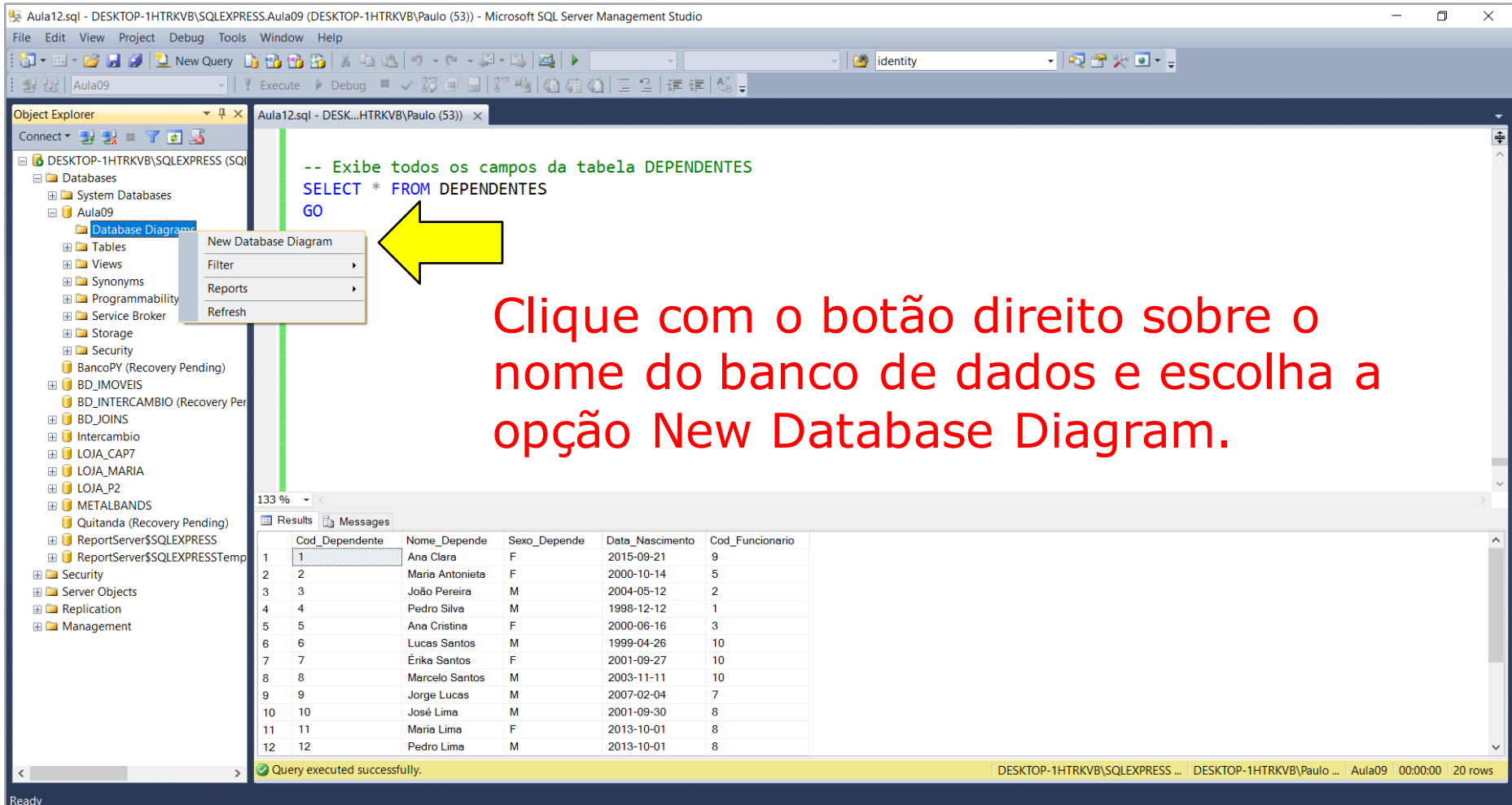


■ Junções de tabelas – JOINS

- Para entender melhor os JOINS e sua aplicação, iremos utilizar o conteúdo das tabelas **FUNCIONARIOS** e **DEPENDENTES**, cujo **Diagrama de Banco de Dados** e exibido logo abaixo:



■ Diagrama de banco de dados



The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left displays the database structure, with 'Database Diagrams' highlighted. A context menu is open over 'Database Diagrams', showing options like 'New Database Diagram', 'Filter', 'Reports', and 'Refresh'. A yellow arrow points to the 'New Database Diagram' option. The main query window shows the following SQL code:

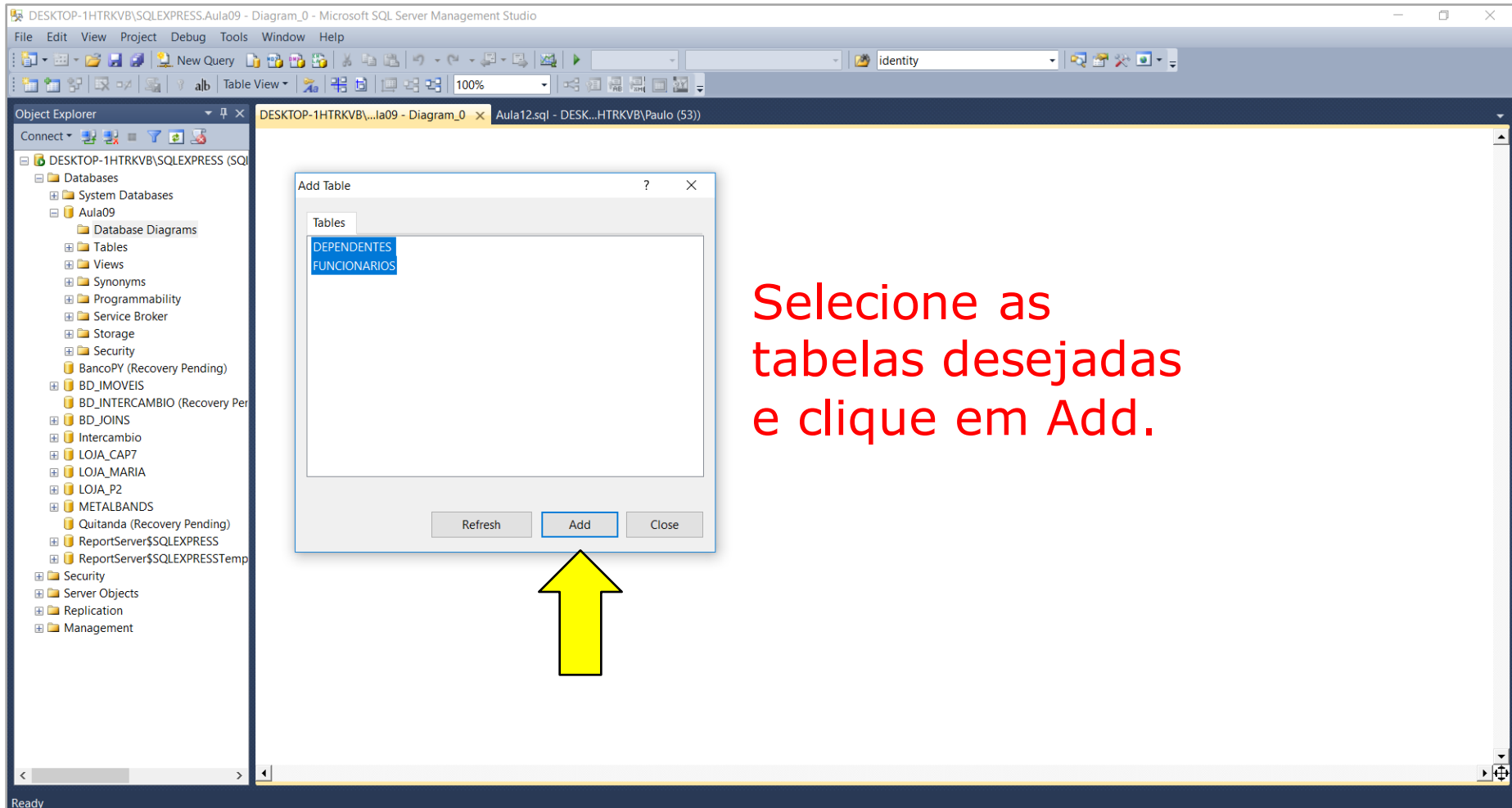
```
-- Exibe todos os campos da tabela DEPENDENTES  
SELECT * FROM DEPENDENTES  
GO
```

The 'Results' pane at the bottom displays the output of the query, which is a table with 12 rows and 5 columns. The status bar at the bottom indicates 'Query executed successfully.' and '20 rows'.

	Cod_Dependente	Nome_Dependente	Sexo_Dependente	Data_Nascimento	Cod_Funcionario
1	1	Ana Clara	F	2015-09-21	9
2	2	Maria Antonieta	F	2000-10-14	5
3	3	João Pereira	M	2004-05-12	2
4	4	Pedro Silva	M	1998-12-12	1
5	5	Ana Cristina	F	2000-06-16	3
6	6	Lucas Santos	M	1999-04-26	10
7	7	Érika Santos	F	2001-09-27	10
8	8	Marcelo Santos	M	2003-11-11	10
9	9	Jorge Lucas	M	2007-02-04	7
10	10	José Lima	M	2001-09-30	8
11	11	Maria Lima	F	2013-10-01	8
12	12	Pedro Lima	M	2013-10-01	8

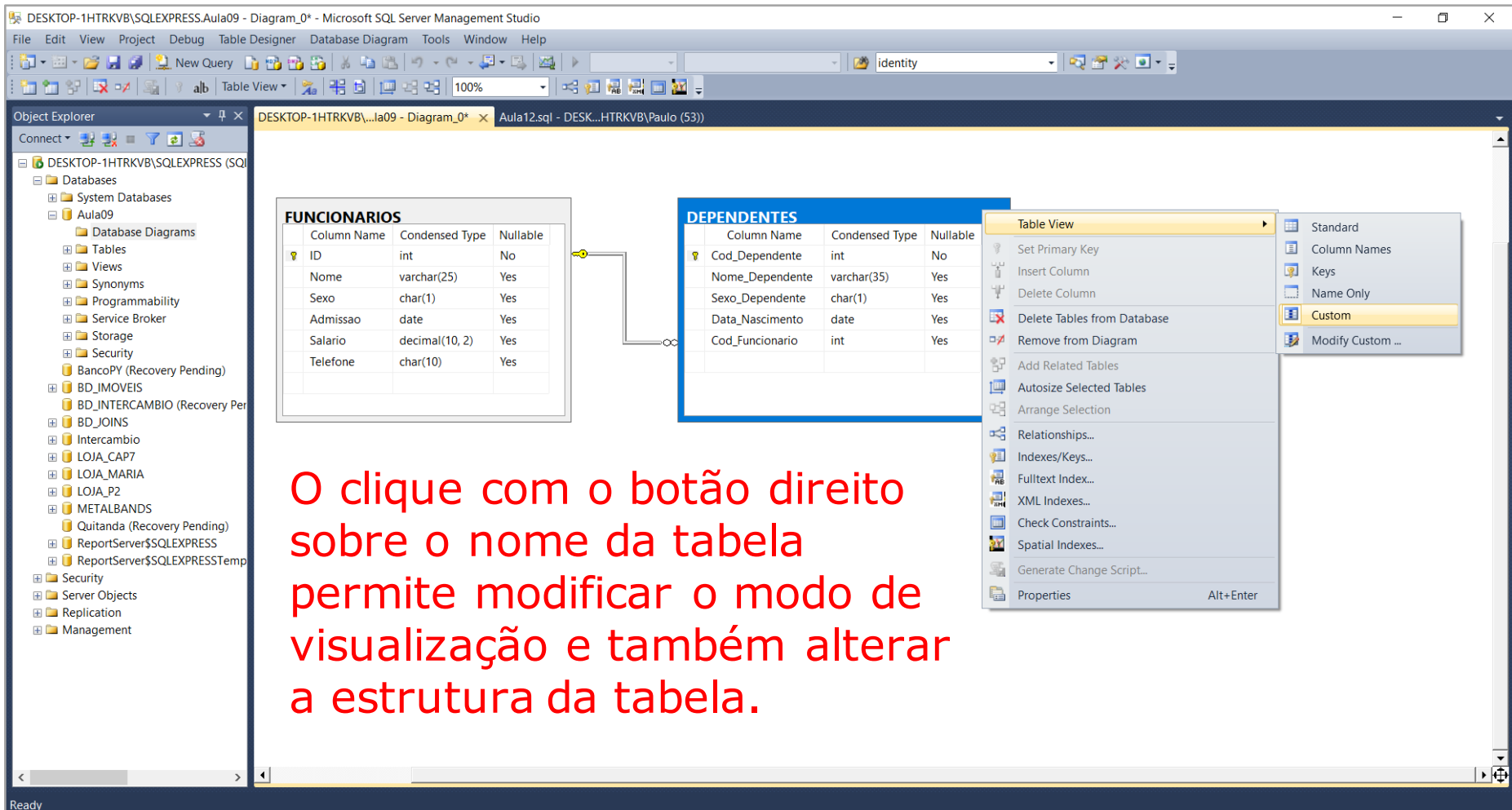
Clique com o botão direito sobre o nome do banco de dados e escolha a opção New Database Diagram.

■ Diagrama de banco de dados



Selecione as
tabelas desejadas
e clique em Add.

■ Diagrama de banco de dados



DESKTOP-1HTRKVB\SQLEXPRESS.Aula09 - Diagram_0* - Microsoft SQL Server Management Studio

File Edit View Project Debug Table Designer Database Diagram Tools Window Help

Object Explorer

DESKTOP-1HTRKVB\SQLEXPRESS (SQL)

Databases

System Databases

Aula09

Database Diagrams

Tables

Views

Synonyms

Programmability

Service Broker

Storage

Security

BancoPY (Recovery Pending)

BD_IMOVEIS

BD_INTERCAMBIO (Recovery Pending)

BD_JOINS

Intercambio

LOJA_CAP7

LOJA_MARIA

LOJA_P2

METALBANDS

Quitanda (Recovery Pending)

ReportServer\$SQLEXPRESS

ReportServer\$SQLEXPRESSTemp

Security

Server Objects

Replication

Management

DESKTOP-1HTRKVB\...Aula09 - Diagram_0* x Aula12.sql - DESK...HTRKVB\Paulo (53))

100%

identity

Table View

Table View

Standard

Column Names

Keys

Name Only

Custom

Modify Custom ...

Table View

Set Primary Key

Insert Column

Delete Column

Delete Tables from Database

Remove from Diagram

Add Related Tables

Autosize Selected Tables

Arrange Selection

Relationships...

Indexes/Keys...

Fulltext Index...

XML Indexes...

Check Constraints...

Spatial Indexes...

Generate Change Script...

Properties

Alt+Enter

FUNCIONARIOS

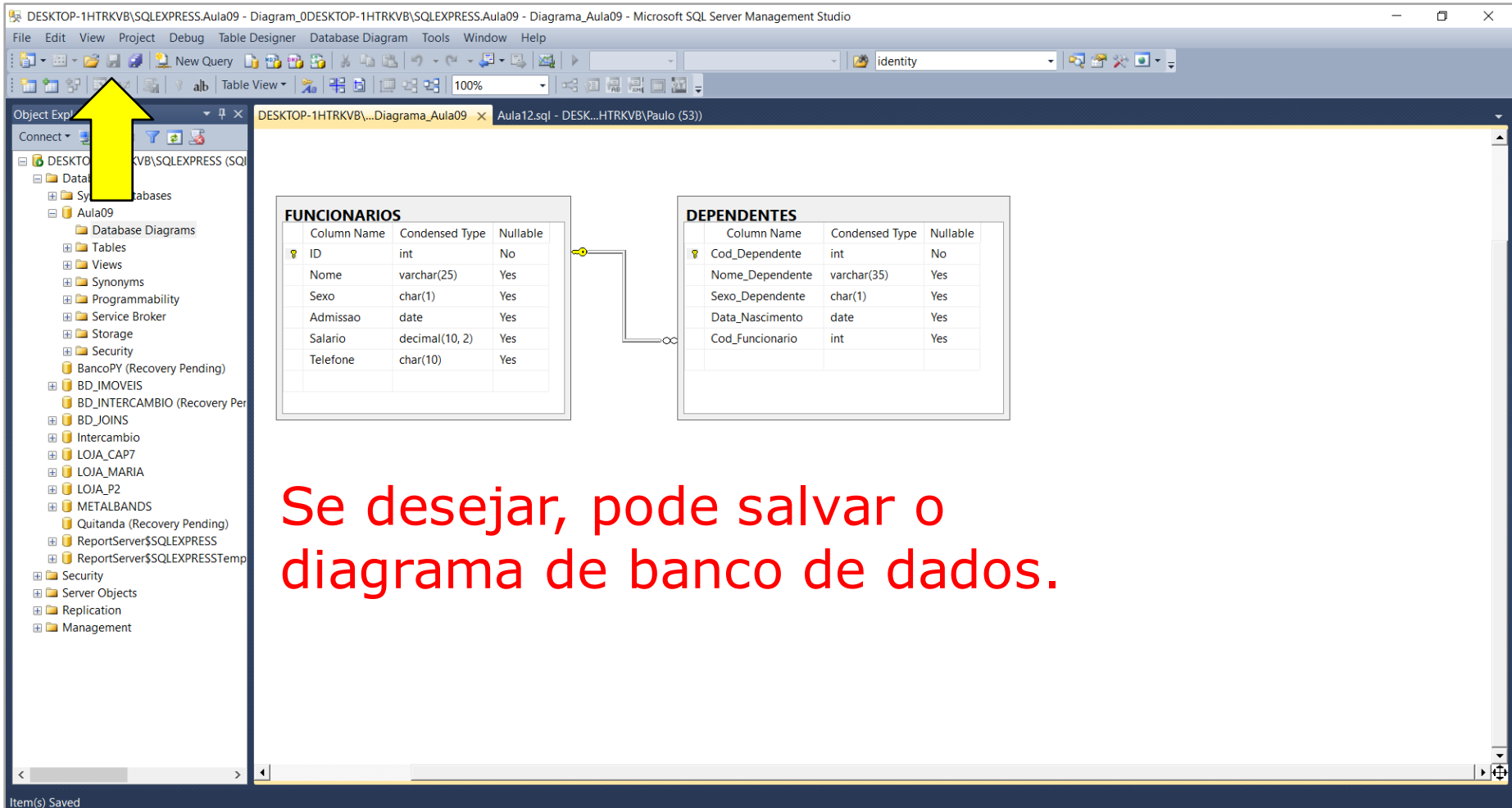
Column Name	Condensed Type	Nullable
ID	int	No
Nome	varchar(25)	Yes
Sexo	char(1)	Yes
Admissao	date	Yes
Salario	decimal(10, 2)	Yes
Telefone	char(10)	Yes

DEPENDENTES

Column Name	Condensed Type	Nullable
Cod_Dependente	int	No
Nome_Dependente	varchar(35)	Yes
Sexo_Dependente	char(1)	Yes
Data_Nascimento	date	Yes
Cod_Funcionario	int	Yes

O clique com o botão direito sobre o nome da tabela permite modificar o modo de visualização e também alterar a estrutura da tabela.

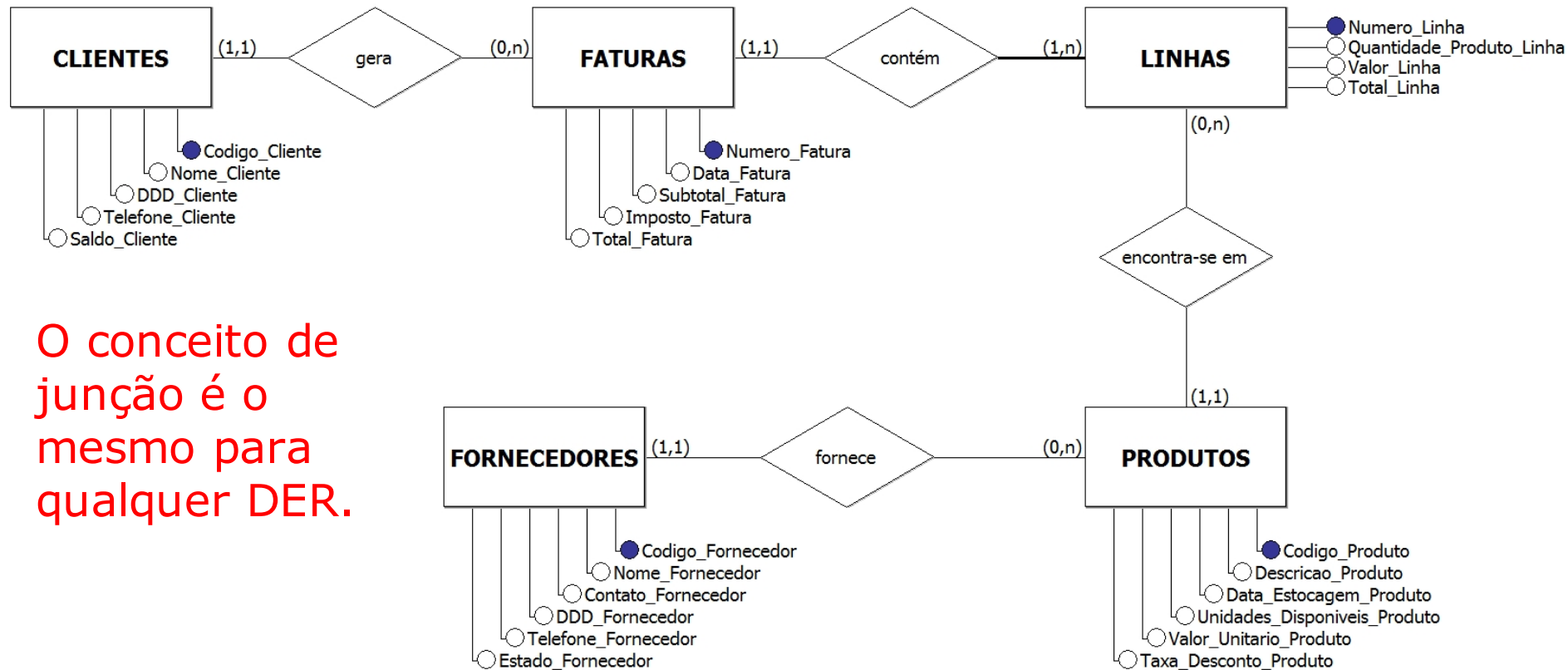
■ Diagrama de banco de dados



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure, with a yellow arrow pointing to the 'Database Diagrams' folder. The main window shows a diagram with two tables: 'FUNCIONARIOS' and 'DEPENDENTES'. The 'FUNCIONARIOS' table has columns: ID (int, primary key), Nome (varchar(25)), Sexo (char(1)), Admissao (date), Salario (decimal(10, 2)), and Telefone (char(10)). The 'DEPENDENTES' table has columns: Cod_Dependente (int, primary key), Nome_Dependente (varchar(35)), Sexo_Dependente (char(1)), Data_Nascimento (date), and Cod_Funcionario (int). A line connects the primary key of 'DEPENDENTES' to the 'Cod_Funcionario' column in 'DEPENDENTES', indicating a foreign key relationship.

Se desejar, pode salvar o diagrama de banco de dados.

■ Junções de tabelas – JOINS



■ Junções de tabelas – JOINS



A seguir, veremos como podemos trabalhar com as operações de junção, utilizando o nosso banco de dados de exemplo.



■ Cláusula CROSS JOIN

- Realiza a junção cruzada entre os dados
- das tabelas FUNCIONARIOS e DEPENDENTES
- Versão 1 - Sintaxe antiga

SELECT *

FROM FUNCIONARIOS, DEPENDENTES

GO



■ Cláusula CROSS JOIN

- Realiza a junção cruzada entre os dados
- das tabelas FUNCIONARIOS e DEPENDENTES
- Versão 2 - Sintaxe utilizando CROSS JOIN

SELECT *

FROM FUNCIONARIOS CROSS JOIN

DEPENDENTES

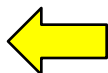
GO



■ Cláusula CROSS JOIN

ID	Nome	Sexo	Admissao	Salario	Telefone	Cod_Dependente	Nome_Dependente	Sexo_Dependente	Data_Nascimento	Cod_Funcionario
1	Maria da Silva	F	2018-01-10	2500.00	3668-0010	1	Ana Clara	F	2015-09-21	9
1	Maria da Silva	F	2018-01-10	2500.00	3668-0010	2	Maria Antonieta	F	2000-10-14	5
1	Maria da Silva	F	2018-01-10	2500.00	3668-0010	3	João Pereira	M	2004-05-12	2
1	Maria da Silva	F	2018-01-10	2500.00	3668-0010	4	Pedro Silva	M	1998-12-12	1
1	Maria da Silva	F	2018-01-10	2500.00	3668-0010	5	Ana Cristina	F	2000-06-16	3
1	Maria da Silva	F	2018-01-10	2500.00	3668-0010	6	Lucas Santos	M	1999-04-26	10
1	Maria da Silva	F	2018-01-10	2500.00	3668-0010	7	Érika Santos	F	2001-09-27	10
1	Maria da Silva	F	2018-01-10	2500.00	3668-0010	8	Marcelo Santos	M	2003-11-11	10
1	Maria da Silva	F	2018-01-10	2500.00	3668-0010	9	Jorge Lucas	M	2007-02-04	7
1	Maria da Silva	F	2018-01-10	2500.00	3668-0010	10	José Lima	M	2001-09-30	8
.....										
10	Maria Cristina	F	2017-10-10	1400.00	3662-1515	10	José Lima	M	2001-09-30	8
10	Maria Cristina	F	2017-10-10	1400.00	3662-1515	11	Maria Lima	F	2013-10-01	8
10	Maria Cristina	F	2017-10-10	1400.00	3662-1515	12	Pedro Lima	M	2013-10-01	8
10	Maria Cristina	F	2017-10-10	1400.00	3662-1515	13	Carlos Antônio	M	2015-08-03	4
10	Maria Cristina	F	2017-10-10	1400.00	3662-1515	14	Sheila Mariana	F	1997-03-04	3
10	Maria Cristina	F	2017-10-10	1400.00	3662-1515	15	Lilian Maria	F	2003-08-12	3
10	Maria Cristina	F	2017-10-10	1400.00	3662-1515	16	Melissa Pereira	F	2003-10-06	2
10	Maria Cristina	F	2017-10-10	1400.00	3662-1515	17	Larissa Souza	F	1998-05-07	6
10	Maria Cristina	F	2017-10-10	1400.00	3662-1515	18	Edson Souza	M	2000-01-14	6
10	Maria Cristina	F	2017-10-10	1400.00	3662-1515	19	Miriam Muniz	F	2003-11-09	5
10	Maria Cristina	F	2017-10-10	1400.00	3662-1515	20	Antônio Romão	M	1998-10-10	1



(200 row(s) affected)



■ Cláusula CROSS JOIN

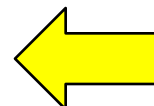
- Realiza a junção cruzada entre os dados das tabelas
- FUNCIONARIOS e DEPENDENTES. Utiliza um alias para
- nomear cada tabela.

```
SELECT F.ID      AS 'Código do Funcionário',  
       F.Nome    AS 'Nome do Funcionário',  
       D.Nome_Dependente AS 'Nome do  
Dependente',  
       D.Data_Nascimento AS 'Data de  
Nascimento'  
FROM  FUNCIONARIOS F CROSS JOIN DEPENDENTES D  
GO
```



■ Cláusula CROSS JOIN

Código do Funcionário	Nome do Funcionário	Nome do Dependente	Data de Nascimento
1	Maria da Silva	Ana Clara	2015-09-21
1	Maria da Silva	Maria Antonieta	2000-10-14
1	Maria da Silva	João Pereira	2004-05-12
1	Maria da Silva	Pedro Silva	1998-12-12
1	Maria da Silva	Ana Cristina	2000-06-16
1	Maria da Silva	Lucas Santos	1999-04-26
1	Maria da Silva	Érika Santos	2001-09-27
1	Maria da Silva	Marcelo Santos	2003-11-11
1	Maria da Silva	Jorge Lucas	2007-02-04
1	Maria da Silva	José Lima	2001-09-30
.....			
10	Maria Cristina	José Lima	2001-09-30
10	Maria Cristina	Maria Lima	2013-10-01
10	Maria Cristina	Pedro Lima	2013-10-01
10	Maria Cristina	Carlos Antônio	2015-08-03
10	Maria Cristina	Sheila Mariana	1997-03-04
10	Maria Cristina	Lilian Maria	2003-08-12
10	Maria Cristina	Melissa Pereira	2003-10-06
10	Maria Cristina	Larissa Souza	1998-05-07
10	Maria Cristina	Edson Souza	2000-01-14
10	Maria Cristina	Miriam Muniz	2003-11-09
10	Maria Cristina	Antônio Romão	1998-10-10



(200 row(s) affected)



■ Cláusulas NATURAL JOIN e USING

-- NATURAL JOIN -> Não existe no MICROSOFT SQL SERVER

SELECT *

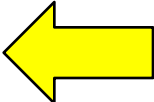
FROM FUNCIONARIOS NATURAL JOIN DEPENDENTES

GO

-- JOIN USING -> Não existe no MICROSOFT SQL SERVER

SELECT *

FROM FUNCIONARIOS F JOIN DEPENDENTES D

USING F.ID  A coluna ID tem que existir
nas duas tabelas

GO



■ Cláusula JOIN ON

-- Exibe o ID, nome e salário dos funcionários, junto com o
-- nome do dependente e o código do responsável por ele.

```
SELECT  F.ID           AS 'Código do Funcionário',  
        F.Nome        AS 'Nome do Funcionário',  
        F.Salario     AS 'Salário',  
        D.Nome_Dependente AS 'Dependente',  
        D.Cod_Funcionario AS 'Código do  
Responsável'
```

```
FROM FUNCIONARIOS F JOIN DEPENDENTES D  
    ON F.ID = D.Cod_Funcionario
```

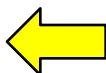
GO



■ Cláusula JOIN ON

Código do Funcionário	Nome do Funcionário	Salário	Dependente	Código do Responsável
9	Maria Cristina	1870.00	Ana Clara	9
5	Marcelo Augusto	2500.00	Maria Antonieta	5
2	Pedro Pereira	1089.00	João Pereira	2
1	Maria da Silva	2500.00	Pedro Silva	1
3	Maria Cristina	1320.00	Ana Cristina	3
10	Maria Cristina	1400.00	Lucas Santos	10
10	Maria Cristina	1400.00	Érika Santos	10
4	Antônio Carlos	1089.00	Carlos Antônio	4
3	Maria Cristina	1320.00	Sheila Mariana	3
3	Maria Cristina	1320.00	Lilian Maria	3
2	Pedro Pereira	1089.00	Melissa Pereira	2
6	Pedro Silva	1155.00	Larissa Souza	6
6	Pedro Silva	1155.00	Edson Souza	6
5	Marcelo Augusto	2500.00	Miriam Muniz	5
1	Maria da Silva	2500.00	Antônio Romão	1

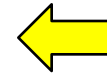
(20 row(s) affected)



■ Cláusula INNER JOIN

- Exibe os dados dos funcionários e de seus dependentes.
- Ordena o resultado pelo nome do funcionário e do dependente.
- Exibe somente os dependentes que nasceram a partir do ano 2000.

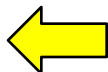
```
SELECT  F.ID           AS 'Código do Funcionário',  
        F.Nome        AS 'Nome do Funcionário',  
        F.Salario      AS 'Salário',  
        D.Nome_Dependente AS 'Dependente',  
        D.Data_Nascimento AS 'Data de Nascimento',  
        D.Cod_Funcionario AS 'Código do Responsável'  
FROM  FUNCIONARIOS F INNER JOIN DEPENDENTES D  
      ON F.ID = D.Cod_Funcionario  
WHERE YEAR(D.Data_Nascimento) >= 2000  
ORDER BY F.Nome, D.Nome_Dependente  
GO
```



■ Cláusula INNER JOIN

Código do Funcionário	Nome do Funcionário	Salário	Dependente	Data de Nascimento	Código do Responsável
4	Antônio Carlos	1089.00	Carlos Antônio	2015-08-03	4
5	Marcelo Augusto	2500.00	Maria Antonieta	2000-10-14	5
5	Marcelo Augusto	2500.00	Miriam Muniz	2003-11-09	5
9	Maria Cristina	1870.00	Ana Clara	2015-09-21	9
3	Maria Cristina	1320.00	Ana Cristina	2000-06-16	3
10	Maria Cristina	1400.00	Érika Santos	2001-09-27	10
3	Maria Cristina	1320.00	Lilian Maria	2003-08-12	3
10	Maria Cristina	1400.00	Marcelo Santos	2003-11-11	10
7	Mônica da Silva	3300.00	Jorge Lucas	2007-02-04	7
2	Pedro Pereira	1089.00	João Pereira	2004-05-12	2
2	Pedro Pereira	1089.00	Melissa Pereira	2003-10-06	2
6	Pedro Silva	1155.00	Edson Souza	2000-01-14	6
8	Tiago Lima	2500.00	José Lima	2001-09-30	8
8	Tiago Lima	2500.00	Maria Lima	2013-10-01	8
8	Tiago Lima	2500.00	Pedro Lima	2013-10-01	8

(15 row(s) affected)



■ Inserindo novos funcionários

-- Insere mais cinco novos funcionários

INSERT INTO FUNCIONARIOS VALUES

(11, 'Ana Cláudia', 'F', '12/09/2011', 4900.00,
'3663-9090'),

(12, 'André Lima', 'M', '05/11/2009', 2050.00,
'3664-8989'),

(13, 'Marcos Souza', 'M', '02/10/2009', 3800.00,
NULL),

(14, 'Mariana Gomes', 'F', '10/11/2010', 1750.50,
NULL),

(15, 'Cíntia Faria', 'F', '10/07/2016', 1750.50,
'3662-1212')

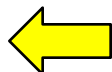


■ Inserindo novos funcionários

ID	Nome	Sexo	Admissao	Salario	Telefone
1	Maria da Silva	F	2018-01-10	2500.00	3668-0010
2	Pedro Pereira	M	2015-03-25	1089.00	3668-1550
3	Maria Cristina	F	2015-09-10	1320.00	NULL
4	Antônio Carlos	M	2015-05-15	1089.00	3664-5000
5	Marcelo Augusto	M	2017-12-09	2500.00	3664-2001
6	Pedro Silva	M	2015-11-15	1155.00	NULL
7	Mônica da Silva	F	2014-10-12	3300.00	3663-9000
8	Tiago Lima	M	2016-05-10	2500.00	NULL
9	Maria Cristina	F	2012-09-21	1870.00	3663-7000
10	Maria Cristina	F	2017-10-10	1400.00	3662-1515
11	Ana Cláudia	F	2011-09-12	4900.00	3663-9090
12	André Lima	M	2009-11-05	2050.00	3664-8989
13	Marcos Souza	M	2009-10-02	3800.00	NULL
14	Mariana Gomes	F	2010-11-10	1750.50	NULL
15	Cíntia Faria	F	2016-07-10	1750.50	3662-1212

```
SELECT *
FROM FUNCIONARIOS
GO
```

(15 row(s) affected)



■ Cláusula LEFT JOIN

- Exibe as informações de todos os funcionários,
- inclusive daqueles que não possuem nenhum dependente

```
SELECT  F.ID                AS 'ID',  
        F.Nome             AS 'Funcionário',  
        F.Admissao         AS 'Admissão',  
        F.Salario          AS 'Salário',  
        D.Nome_Dependente  AS 'Dependente',  
        D.Data_Nascimento  AS 'Data de Nascimento'  
FROM    FUNCIONARIOS F LEFT JOIN DEPENDENTES D  
        ON F.ID = D.Cod_Funcionario
```

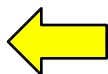
GO



■ Cláusula LEFT JOIN

ID	Funcionário	Admissão	Salário	Dependente	Data de Nascimento
1	Maria da Silva	2018-01-10	2500.00	Pedro Silva	1998-12-12
1	Maria da Silva	2018-01-10	2500.00	Antônio Romão	1998-10-10
2	Pedro Pereira	2015-03-25	1089.00	João Pereira	2004-05-12
2	Pedro Pereira	2015-03-25	1089.00	Melissa Pereira	2003-10-06
3	Maria Cristina	2015-09-10	1320.00	Ana Cristina	2000-06-16
3	Maria Cristina	2015-09-10	1320.00	Sheila Mariana	1997-03-04
.....					
10	Maria Cristina	2017-10-10	1400.00	Érika Santos	2001-09-27
10	Maria Cristina	2017-10-10	1400.00	Marcelo Santos	2003-11-11
11	Ana Cláudia	2011-09-12	4900.00	NULL	NULL
12	André Lima	2009-11-05	2050.00	NULL	NULL
13	Marcos Souza	2009-10-02	3800.00	NULL	NULL
14	Mariana Gomes	2010-11-10	1750.50	NULL	NULL
15	Cínthia Faria	2016-07-10	1750.50	NULL	NULL

(25 row(s) affected)



■ Cláusula LEFT JOIN

-- Exibe as informações de todos os funcionários
-- que não possuem nenhum dependente.

```
SELECT  F.ID          AS 'ID',  
        F.Nome        AS 'Funcionário',  
        F.Admissao    AS 'Admissão',  
        F.Salario     AS 'Salário',  
        D.Nome_Dependente AS 'Dependente',  
        D.Data_Nascimento AS 'Data de Nascimento'  
FROM  FUNCIONARIOS F LEFT JOIN DEPENDENTES D  
      ON F.ID = D.Cod_Funcionario  
WHERE D.Nome_Dependente IS NULL
```

GO



■ Cláusula LEFT JOIN

ID	Funcionário	Admissão	Salário	Dependente	Data de Nascimento
--	-----	-----	-----	-----	-----
11	Ana Cláudia	2011-09-12	4900.00	NULL	NULL
12	André Lima	2009-11-05	2050.00	NULL	NULL
13	Marcos Souza	2009-10-02	3800.00	NULL	NULL
14	Mariana Gomes	2010-11-10	1750.50	NULL	NULL
15	Cínthia Faria	2016-07-10	1750.50	NULL	NULL

(5 row(s) affected) 



■ Cláusula LEFT JOIN

ID	Funcionário	Admissão	Salário	Dependente	Data de Nascimento
11	Ana Cláudia	2011-09-12	4900.00	NULL	NULL
12	André Lima	2009-11-05	2050.00	NULL	NULL
13	Marcos Souza	2009-10-02	3800.00	NULL	NULL
14	Mariana Gomes	2010-11-10	1750.50	NULL	NULL
15	Cínthia Faria	2016-07-10	1750.50	NULL	NULL



```
...
FROM FUNCIONARIOS F LEFT JOIN DEPENDENTES D
    ON F.ID = D.Cod_Funcionario
WHERE D.Nome_Dependente IS NULL
...
```

Tabela à esquerda
do LEFT JOIN:

FUNCIONARIOS



■ Cláusula RIGHT JOIN

- Exibe as informações de todos os funcionários
- e de seus dependentes, utilizando um RIGHT JOIN.

```
SELECT  F.ID                AS 'ID',  
        F.Nome              AS 'Funcionário',  
        F.Admissao          AS 'Admissão',  
        F.Salario           AS 'Salário',  
        D.Nome_Dependente   AS 'Dependente',  
        D.Data_Nascimento   AS 'Data de Nascimento'  
FROM    DEPENDENTES D RIGHT JOIN FUNCIONARIOS F  
ON      F.ID = D.Cod_Funcionario
```



■ Cláusula RIGHT JOIN

ID	Funcionário	Admissão	Salário	Dependente	Data de Nascimento
1	Maria da Silva	2018-01-10	2500.00	Pedro Silva	1998-12-12
1	Maria da Silva	2018-01-10	2500.00	Antônio Romão	1998-10-10
2	Pedro Pereira	2015-03-25	1089.00	João Pereira	2004-05-12
2	Pedro Pereira	2015-03-25	1089.00	Melissa Pereira	2003-10-06
3	Maria Cristina	2015-09-10	1320.00	Ana Cristina	2000-06-16
3	Maria Cristina	2015-09-10	1320.00	Sheila Mariana	1997-03-04
.....					
10	Maria Cristina	2017-10-10	1400.00	Érika Santos	2001-09-27
10	Maria Cristina	2017-10-10	1400.00	Marcelo Santos	2003-11-11
11	Ana Cláudia	2011-09-12	4900.00	NULL	NULL
12	André Lima	2009-11-05	2050.00	NULL	NULL
13	Marcos Souza	2009-10-02	3800.00	NULL	NULL
14	Mariana Gomes	2010-11-10	1750.50	NULL	NULL
15	Cíntia Faria	2016-07-10	1750.50	NULL	NULL

(25 row(s) affected) 



■ Cláusula FULL JOIN

- Exibe as informações de todos os funcionários
- e de seus dependentes, utilizando um FULL JOIN.

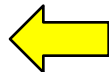
```
SELECT  F.ID                AS 'ID',  
        F.Nome              AS 'Funcionário',  
        F.Admissao          AS 'Admissão',  
        F.Salario           AS 'Salário',  
        D.Nome_Dependente   AS 'Dependente',  
        D.Data_Nascimento   AS 'Data de Nascimento'  
FROM  FUNCIONARIOS F FULL JOIN DEPENDENTES D  
      ON F.ID = D.Cod_Funcionario
```



■ Cláusula FULL JOIN

ID	Funcionário	Admissão	Salário	Dependente	Data de Nascimento
1	Maria da Silva	2018-01-10	2500.00	Pedro Silva	1998-12-12
1	Maria da Silva	2018-01-10	2500.00	Antônio Romão	1998-10-10
2	Pedro Pereira	2015-03-25	1089.00	João Pereira	2004-05-12
2	Pedro Pereira	2015-03-25	1089.00	Melissa Pereira	2003-10-06
.....					
10	Maria Cristina	2017-10-10	1400.00	Érika Santos	2001-09-27
10	Maria Cristina	2017-10-10	1400.00	Marcelo Santos	2003-11-11
11	Ana Cláudia	2011-09-12	4900.00	NULL	NULL
12	André Lima	2009-11-05	2050.00	NULL	NULL
13	Marcos Souza	2009-10-02	3800.00	NULL	NULL
14	Mariana Gomes	2010-11-10	1750.50	NULL	NULL
15	Cíntia Faria	2016-07-10	1750.50	NULL	NULL

(25 row(s) affected)



■ Consulta com vários JOINS

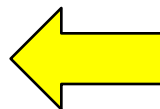
-- Exemplo de consulta mais complexa, envolvendo vários JOINS

```
SELECT C.Codigo_Cliente      AS 'ID do Cliente',
       C.Nome_Cliente        AS 'Nome do Cliente',
       C.Telefone_Cliente    AS 'Telefone do Cliente',
       F.Numero_Fatura        AS 'Número da Fatura',
       F.Data_Fatura          AS 'Data da Fatura',
       L.Numero_Linha         AS 'Número da Linha',
       P.Descricao_Produto    AS 'Nome do Produto',
       F2.Nome_Fornecedor     AS 'Nome do Fornecedor',
       P.Valor_Unitario_Produto AS 'Valor Unitário',
       P.Taxa_Desconto_Produto AS 'Taxa de Desconto',
       L.Quantidade_Produto_Linha AS 'Quantidade de Produto',
       L.Valor_Linha          AS 'Valor da Linha',
       L.Total_Linha          AS 'Total da Linha',
       F.Subtotal_Fatura      AS 'Subtotal da Fatura',
       F.Imposto_Fatura       AS 'Imposto da Fatura',
       F.Total_Fatura         AS 'Total da Fatura'

FROM CLIENTES C INNER JOIN FATURAS F
    ON C.Codigo_Cliente = F.Codigo_Cliente
INNER JOIN LINHAS L
    ON F.Numero_Fatura = L.Numero_Fatura
INNER JOIN PRODUTOS P
    ON L.Codigo_Produto = P.Codigo_Produto
INNER JOIN FORNECEDORES F2
    ON P.Codigo_Fornecedor = F2.Codigo_Fornecedor

ORDER BY C.Nome_Cliente,
         F.Data_Fatura,
         L.Numero_Linha,
         P.Descricao_Produto,
         F2.Nome_Fornecedor
```

Exemplo de consulta
com vários JOINS



■ Consulta com vários JOINS

ID do Cliente	Nome do Cliente	Telefone do Cliente	Número da Fatura	Data da Fatura	Número da Linha	Nome do Produto	Nome do Fornecedor	Valor Unitário	Taxa de Desconto	Quantidade de Produto	Valor da Linha	Total da Linha	Subtotal da Fatura	Imposto da Fatura	Total da Fatura
54	Alison Ribeiro	95 3898642	116	2016-02-04	1	Notebook Acer Aspire E5-574-307M	Comercial Lininha	1994.05	0.16	1	199.405	1675.00	2.445.37	0.49	3643.60
54	Alison Ribeiro	95 3898642	116	2016-02-04	2	Mochila Belkin FBK780N NCB1 Notebook	Armadem da Família	94.00	0.00	2	188.00	188.00	2.445.37	0.49	3643.60
54	Alison Ribeiro	95 3898642	116	2016-02-04	3	Pasta Para Notebook Multiuso	Mercearia do Zé	24.20	0.07	10	242.00	225.06	2.445.37	0.49	3643.60
54	Alison Ribeiro	95 3898642	116	2016-02-04	4	Jogo Cama Solt eiro Santista	Mercedinho Bazanella	72.92	0.02	5	364.60	357.31	2.445.37	0.49	3643.60
54	Alison Ribeiro	95 3898642	118	2016-03-29	1	Jogo Cama Solt eiro All ison	Galpões Correia	573.00	0.00	9	515.70	1015.93		0.49	1513.74
54	Alison Ribeiro	95 3898642	118	2016-03-29	2	Jogo Cama Solt eiro Santista	Mercedinho Bazanella	72.92	0.02	7	510.44	508.23	1015.93	0.49	1513.74
54	Alison Ribeiro	95 3898642	81	2016-06-25	1	Smartphone Samsung Galaxy S7 32 GB	Comercial Lininha	3199.00	0.08	4	12796.00	11772.32	12663.70	0.41	17855.82
54	Alison Ribeiro	95 3898642	81	2016-06-25	2	Mochila Multilaser Slim Notebook	Mercearia do Zé	116.55	0.00	6	699.30	699.30	12663.70	0.41	17855.82
54	Alison Ribeiro	95 3898642	81	2016-06-25	3	Pasta Para Notebook Multiuso	Mercearia do Zé	24.20	0.07	4	96.80	90.82	12663.70	0.41	17855.82
54	Alison Ribeiro	95 3898642	81	2016-06-25	4	Capa Stillo Com Bolso Para Notebook	Ca sas Bahi a	18.90	0.10	6	113.40	102.06	12663.70	0.41	17855.82
54	Alison Ribeiro	95 3898642	14	2016-08-25	1	Pasta Notebook Multilaser Casual	Armadem da Família	74.00	0.00	5	370.00	370.00		0.31	484.70
79	Almir Santos	98 7354797	180	2016-04-07	1	Pasta Para Notebook Multiuso	Mercearia do Zé	24.20	0.07	4	96.80	90.82	738.68	0.31	967.67
79	Almir Santos	98 7354797	180	2016-04-07	2	Capa Stillo Com Bolso Para Notebook	Ca sas Bahi a	18.90	0.10	1	18.90	17.81	738.68	0.31	967.67
79	Almir Santos	98 7354797	180	2016-04-07	3	Colcha Casal Dupla Face Santista	Armadem do Gaúcho	80.67	0.13	9	726.03	631.65	738.68	0.31	967.67
79	Almir Santos	98 7354797	248	2016-04-18	1	Smartphone Samsung Galaxy S7 32 GB	Comercial Lininha	3199.00	0.08	5	15995.00	14715.40		0.46	21484.48
79	Almir Santos	98 7354797	92	2016-05-21	1	Mochila G Notebook Acolchoada	Mercedinho Bazanella	79.90	0.33	4	319.60	214.13	355.71	0.00	355.71
79	Almir Santos	98 7354797	92	2016-05-21	2	Pasta Para Notebook Multiuso	Mercearia do Zé	24.20	0.07	1	24.20	22.51	355.71	0.00	355.71
79	Almir Santos	98 7354797	92	2016-05-21	3	Capa Stillo Com Bolso Para Notebook	Ca sas Bahi a	18.90	0.10	7	132.30	119.07	355.71	0.00	355.71
79	Almir Santos	98 7354797	236	2016-06-16	1	Jogo Cama Solt eiro Santista	Mercedinho Bazanella	72.92	0.02	2	145.84	142.92	1106.31	0.00	1106.31
79	Almir Santos	98 7354797	236	2016-06-16	2	Jogo Cama Solt eiro Santista	Mercedinho Bazanella	72.92	0.02	5	364.60	357.31	1106.31	0.00	1106.31
79	Almir Santos	98 7354797	236	2016-06-16	3	Capa Stillo Com Bolso Para Notebook	Ca sas Bahi a	18.90	0.10	8	151.20	136.08	1106.31	0.00	1106.31
79	Almir Santos	98 7354797	236	2016-06-16	4	Mochila Belkin FBK780N NCB1 Notebook	Armadem da Família	94.00	0.00	5	470.00	470.00	1106.31	0.00	1106.31
79	Almir Santos	98 7354797	295	2016-06-22	1	Capa Stillo Com Zipper Para Notebook	Comercial Lininha	16.80	0.10	7	117.60	105.84	248.76	0.35	335.83
79	Almir Santos	98 7354797	295	2016-06-22	2	Jogo Cama Solt eiro Santista	Mercedinho Bazanella	72.92	0.02	2	145.84	142.92	248.76	0.35	335.83
79	Almir Santos	98 7354797	292	2016-06-26	1	Jogo Cama Solt eiro Santista	Mercedinho Bazanella	72.92	0.02	10	729.20	714.62	1647.02	0.03	1696.43
79	Almir Santos	98 7354797	292	2016-06-26	2	Mochila Multilaser Slim Notebook	Mercearia do Zé	116.55	0.00	8	932.40	932.40	1647.02	0.03	1696.43
79	Almir Santos	98 7354797	484	2016-07-24	1	Jogo Cama Solt eiro Santista	Mercedinho Bazanella	72.92	0.02	8	583.36	571.69		0.07	611.71
81	Wesley Pereira	92 6731388	463	2016-02-01	1	Jogo Cama Solt eiro Santista	Mercedinho Bazanella	72.92	0.02	1	72.92	71.46		0.39	99.33
81	Wesley Pereira	92 6731388	164	2016-03-24	1	Jogo Cama Solt eiro Santista	Mercedinho Bazanella	72.92	0.02	2	145.84	142.92		0.24	177.22
81	Wesley Pereira	92 6731388	380	2016-03-25	1	Notebook Acer Aspire E5-574-307M	Comercial Lininha	1994.05	0.16	2	3988.10	3350.00		0.14	3896.57
81	Wesley Pereira	92 6731388	280	2016-03-25	2	Capa Stillo Com Bolso Para Notebook	Ca sas Bahi a	18.90	0.10	4	75.60	68.04		0.14	3896.57
81	Wesley Pereira	92 6731388	231	2016-05-20	1	Jogo Cama Solt eiro All ison	Galpões Correia	573.00	0.00	1	57.30	57.30	2791.05	0.01	2818.96
81	Wesley Pereira	92 6731388	231	2016-05-20	2	Capa Stillo Com Bolso Para Notebook	Ca sas Bahi a	18.90	0.10	2	37.80	34.02	2791.05	0.01	2818.96
81	Wesley Pereira	92 6731388	231	2016-05-20	3	Sofá 4 Lugares Linoforte Versa lhes	Armadem do Gaúcho	899.91	0.00	1	899.91	899.91	2791.05	0.01	2818.96
81	Wesley Pereira	92 6731388	231	2016-05-20	4	Sofá 4 Lugares Linoforte Versa lhes	Armadem do Gaúcho	899.91	0.00	2	1799.82	1799.82	2791.05	0.01	2818.96
81	Wesley Pereira	92 6731388	241	2016-06-29	1	Mochila G Notebook Acolchoada	Mercedinho Bazanella	79.90	0.33	5	399.50	267.67		0.14	910.02
81	Wesley Pereira	92 6731388	241	2016-06-29	2	Mochila G Notebook Acolchoada	Mercedinho Bazanella	79.90	0.33	3	239.70	160.60		0.14	910.02
81	Wesley Pereira	92 6731388	241	2016-06-29	3	Pasta Notebook Multilaser Casual	Armadem da Família	74.00	0.00	5	370.00	370.00		0.14	910.02
81	Wesley Pereira	92 6731388	346	2016-07-22	1	Capa Stillo Com Bolso Para Notebook	Ca sas Bahi a	18.90	0.10	9	170.10	153.09		0.18	349.30
81	Wesley Pereira	92 6731388	346	2016-07-22	2	Jogo Cama Solt eiro Santista	Mercedinho Bazanella	72.92	0.02	2	145.84	142.92		0.18	349.30
26	Yuri Silva	94 0653987	348	2016-01-12	1	Pasta Para Notebook Multiuso	Mercearia do Zé	24.20	0.07	4	96.80	90.82		0.29	847.02
26	Yuri Silva	94 0653987	348	2016-01-12	2	Jogo Cama Solt eiro Santista	Mercedinho Bazanella	72.92	0.02	4	291.68	285.85		0.29	847.02
26	Yuri Silva	94 0653987	348	2016-01-12	3	Colcha Casal Dupla Face Santista	Armadem do Gaúcho	80.67	0.13	4	322.68	288.73		0.29	847.02
26	Yuri Silva	94 0653987	317	2016-02-12	1	Notebook Acer Aspire E5-574-307M	Comercial Lininha	1994.05	0.16	2	3988.10	3350.00		0.17	8241.06
26	Yuri Silva	94 0653987	317	2016-02-12	2	Sofá 4 Lugares Linoforte Versa lhes	Armadem do Gaúcho	899.91	0.00	4	3599.64	3599.64		0.17	8241.06
26	Yuri Silva	94 0653987	317	2016-02-12	3	Mochila Belkin FBK780N NCB1 Notebook	Armadem da Família	94.00	0.00	1	94.00	7043.64		0.17	8241.06
26	Yuri Silva	94 0653987	316	2016-02-25	1	Jogo Cama Solt eiro All ison	Galpões Correia	573.00	0.00	4	229.20	229.20		0.05	291.08
26	Yuri Silva	94 0653987	212	2016-03-13	1	Sofá 4 Lugares Linoforte Versa lhes	Armadem do Gaúcho	899.91	0.00	5	4499.55	4499.55		0.05	5825.93
26	Yuri Silva	94 0653987	212	2016-03-13	2	Mochila Multilaser Slim Notebook	Mercearia do Zé	116.55	0.00	9	1048.95	5548.50		0.05	5825.93
26	Yuri Silva	94 0653987	187	2016-03-22	1	Capa Stillo Com Bolso Para Notebook	Ca sas Bahi a	18.90	0.10	3	56.70	51.83		0.47	75.01
26	Yuri Silva	94 0653987	38	2016-04-13	1	Mochila Belkin FBK780N NCB1 Notebook	Armadem da Família	94.00	0.00	8	752.00	752.00		0.46	1097.92
26	Yuri Silva	94 0653987	310	2016-05-02	1	Capa Stillo Com Bolso Para Notebook	Ca sas Bahi a	18.90	0.10	1	18.90	17.01		0.37	23.30

(999 row(s) affected)



■ Total de dependentes de cada funcionário

-- Exibe o total de dependentes de cada
-- funcionário.

```
SELECT F.ID                AS 'ID',  
       F.Nome              AS 'Funcionário',  
       COUNT(F.ID)         AS 'Total de Dependentes'  
FROM  FUNCIONARIOS F INNER JOIN DEPENDENTES D  
      ON F.ID = D.Cod_Funcionario  
GROUP BY F.ID, F.Nome  
GO
```



■ Total de dependentes de cada funcionário

ID	Funcionário	Total de Dependentes
---	-----	-----
1	Maria da Silva	2
2	Pedro Pereira	2
3	Maria Cristina	3
4	Antônio Carlos	1
5	Marcelo Augusto	2
6	Pedro Silva	2
7	Mônica da Silva	1
8	Tiago Lima	3
9	Maria Cristina	1
10	Maria Cristina	3

(10 row(s) affected) 



■ Total de dependentes de cada funcionário

-- Exibe o total de dependentes de cada
-- funcionário.

```
SELECT F.ID                AS 'ID',  
       F.Nome              AS 'Funcionário',  
       ➡ COUNT(F.ID)       AS 'Total de Dependentes'  
FROM FUNCIONARIOS F FULL JOIN DEPENDENTES D  
  ON F.ID = D.Cod_Funcionario  
GROUP BY F.ID, F.Nome  
GO
```

➡ Lógica incorreta!



■ Total de dependentes de cada funcionário

ID	Funcionário	Total de Dependentes
----	-------------	----------------------

---	-----	-----
-----	-------	-------

1	Maria da Silva	2
---	----------------	---

2	Pedro Pereira	2
---	---------------	---

.....		
-------	--	--

9	Maria Cristina	1
---	----------------	---

10	Maria Cristina	3
----	----------------	---

11	Ana Cláudia	1
----	-------------	---

12	André Lima	1
----	------------	---

13	Marcos Souza	1
----	--------------	---

14	Mariana Gomes	1
----	---------------	---

15	Cínthia Faria	1
----	---------------	---

← Lógica incorreta!

(15 row(s) affected) ←



■ Total de dependentes de cada funcionário

-- Exibe o total de dependentes de cada
-- funcionário.

```
SELECT F.ID                AS 'ID',  
       F.Nome              AS 'Funcionário',  
       ➡ COUNT(D.Cod_Dependente) AS 'Total de  
Dependentes'  
FROM FUNCIONARIOS F FULL JOIN DEPENDENTES D  
   ON F.ID = D.Cod_Funcionario  
GROUP BY F.ID, F.Nome  
GO
```

➡ Lógica correta!



■ Total de dependentes de cada funcionário

ID	Funcionário	Total de Dependentes
----	-------------	----------------------

1	Maria da Silva	2
---	----------------	---

2	Pedro Pereira	2
---	---------------	---

.....

9	Maria Cristina	1
---	----------------	---

10	Maria Cristina	3
----	----------------	---

11	Ana Cláudia	0
----	-------------	---

12	André Lima	0
----	------------	---

13	Marcos Souza	0
----	--------------	---

14	Mariana Gomes	0
----	---------------	---

15	Cínthia Faria	0
----	---------------	---

← Lógica correta!

(15 row(s) affected) ←



- **Na próxima aula veremos**
 - Consultas relacionais e subconsultas.

