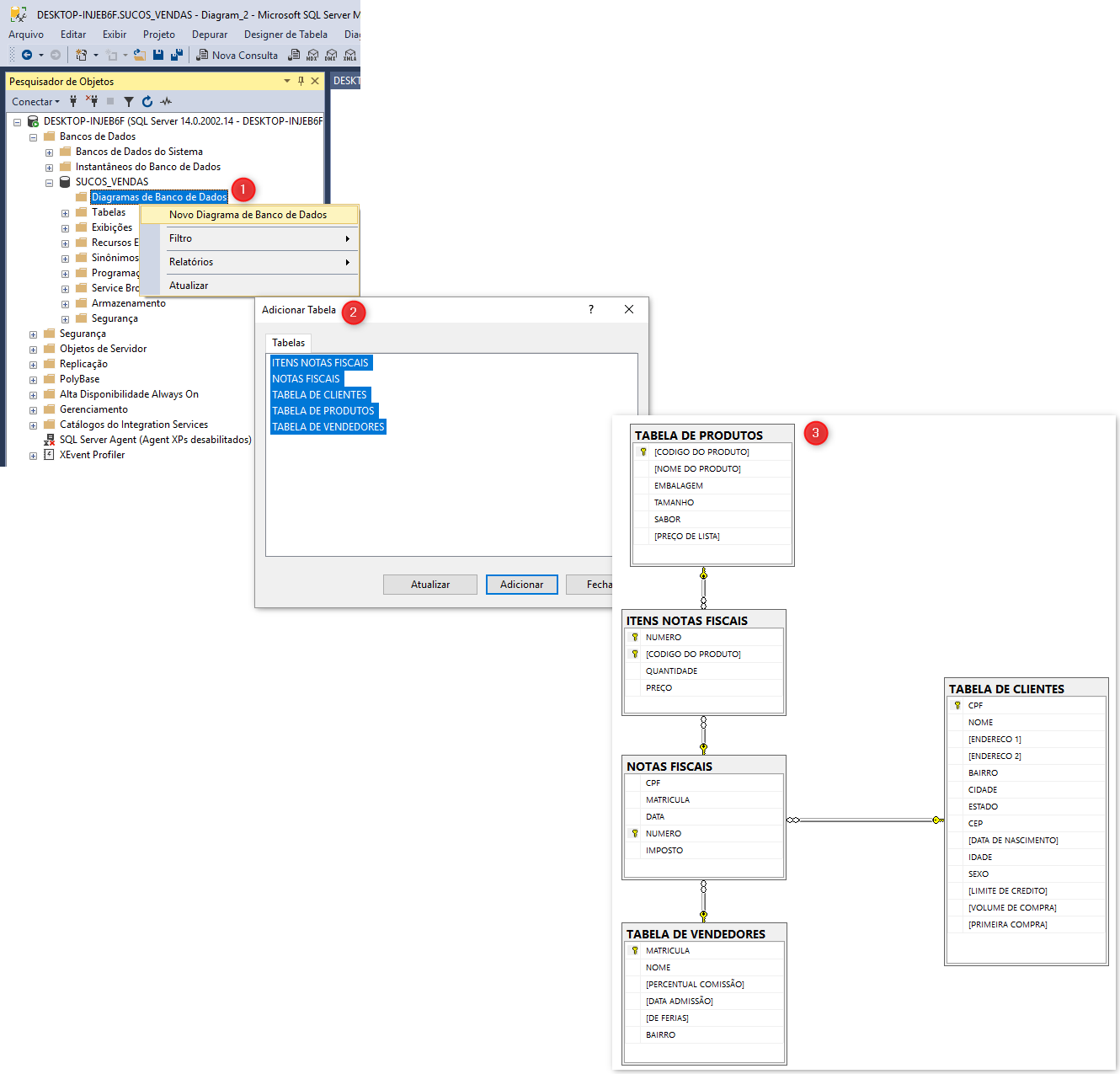
**SQL Server - Consultas Inteligentes**

**1 - CRIAR DIAGRAMA DE RELACIONAMENTO**

Conhecer o diagrama de classes é fundamental para saber como realizar as consultas.



**2 - FUNÇÕES AGREGADAS**

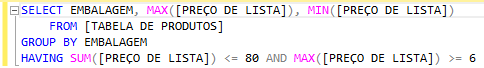
⦁ COUNT(): Apenas conta os valores

⦁ SUM(): Soma dos valores

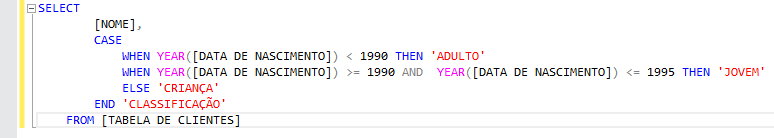
⦁ MAX(): Maior valor

⦁ MIN(): Menor valor

⦁ AVG(): Média dos valores



**3 - UTILIZANDO O CASE EM PROJEÇÕES**



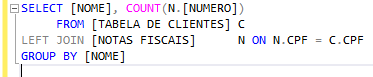
**4 - JOINS**

Relacionar as tabelas de acordo com os valores (geralmente CHAVE PRIMÁRIA).

**OBS**: a junção só pode ocorrer com atributos com **mesmo tipo**.

⦁ **INNER JOIN**: Junta APENAS os registros que existem nas duas tabelas.

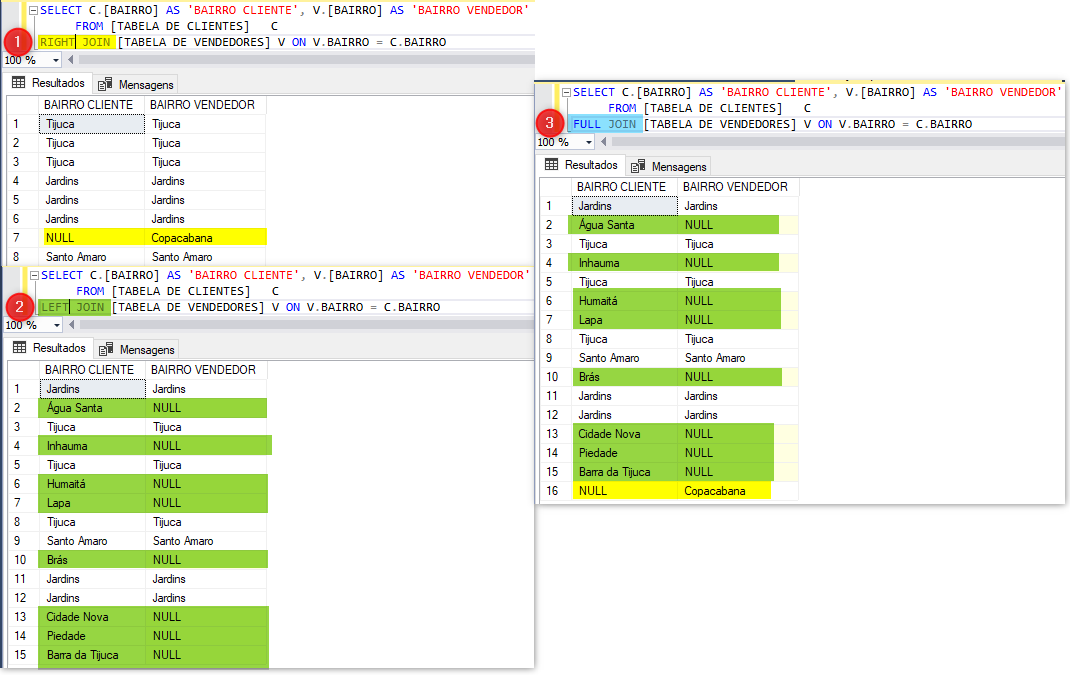
⦁ **LEFT JOIN / RIGHT JOIN**: Faz a junção mesmo que NÃO exista os registros de junção nas duas tabelas (ESQUERDA / DIREITA DA TABELA).



No exemplo acima:

* **LEITURA**: Preciso trazer TODOS os clientes, mesmo que não tenha nenhum registro desta tabela que está à ESQUERDA na tabela da direita.
* **OBS**: Ao realizar COUNT(), atentar-se ao atributo dentro da função: o que eu devo considerar para somar. No exemplo acima, devo considerar a PK da Nota Fiscal, uma vez que se um cliente não tiver nenhuma nota o COUNT me retornará ZERO para este cliente.

⦁ **FULL JOIN**: **RIGHT JOIN + LEFT JOIN** entre as tabelas. Quando não houver junção irá trazer NULL nos atributos tanto de uma tabela quanto da outra.



⦁ **CROSS JOIN**: Todas as combinações possíveis entre DUAS tabelas.

**4.1 - JUNTANDO CONSULTAS**

As consultas são operações de **Álgebra Relacional:** na teoria de conjuntos utilizar a UNIÃO, INTERSECÇÃO, EXCEÇÃO(DIFERENÇA) E O PRODUTO CARTESIANO (CROSS JOIN). Estas operações só podem ser utilizadas se:

* A **projeção** for **igual** em ambas as consultas;
* O **tipo** em cada **atributo** da projeção for **igual**.

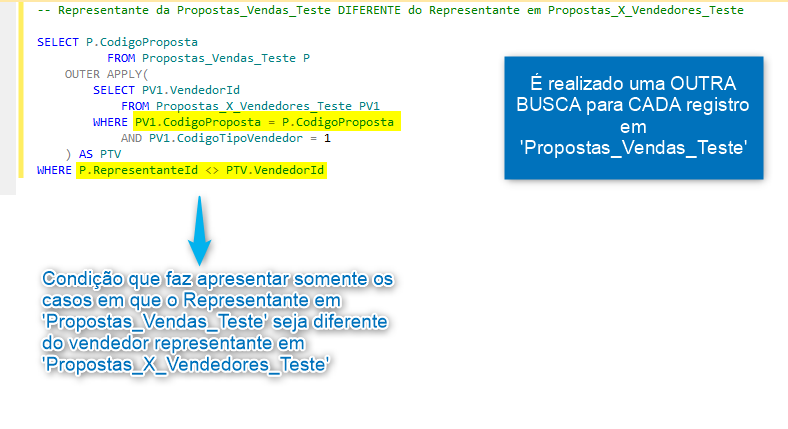
**UNION / UNION ALL**

* **UNION:** Faz a união de 2 consultas / *conjunto* em um resultado, **aplicando o DISTINCT.**
* **UNION ALL:** Faz a união de 2 consultas / *conjunto* em um resultado, **NÃO** **aplicando o DISTINCT.**

**INTERSECT:** Faz a intersecção entre 2 consultas / *conjuntos* em um resultado. Apresenta apenas o que existe nos DOIS conjuntos.

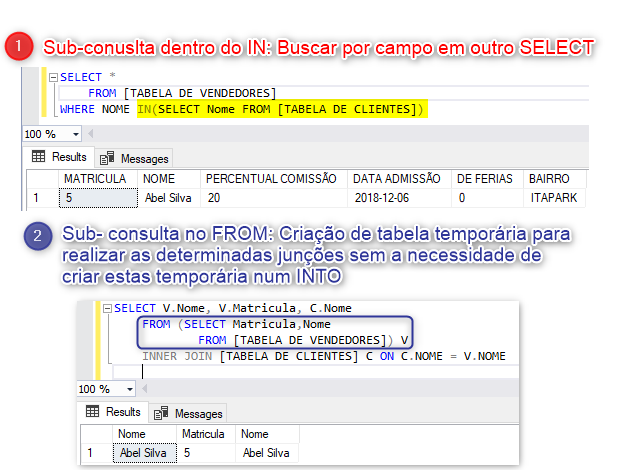
**EXCEPT**: Mostra o que existe apenas no PRIMEIRO conjunto. **Exclui** os registros que existem no SEGUNDA conjunto para exibir na consulta.

**OUTER APLLY:** Permite realizar uma outra consulta a partir da primeira.



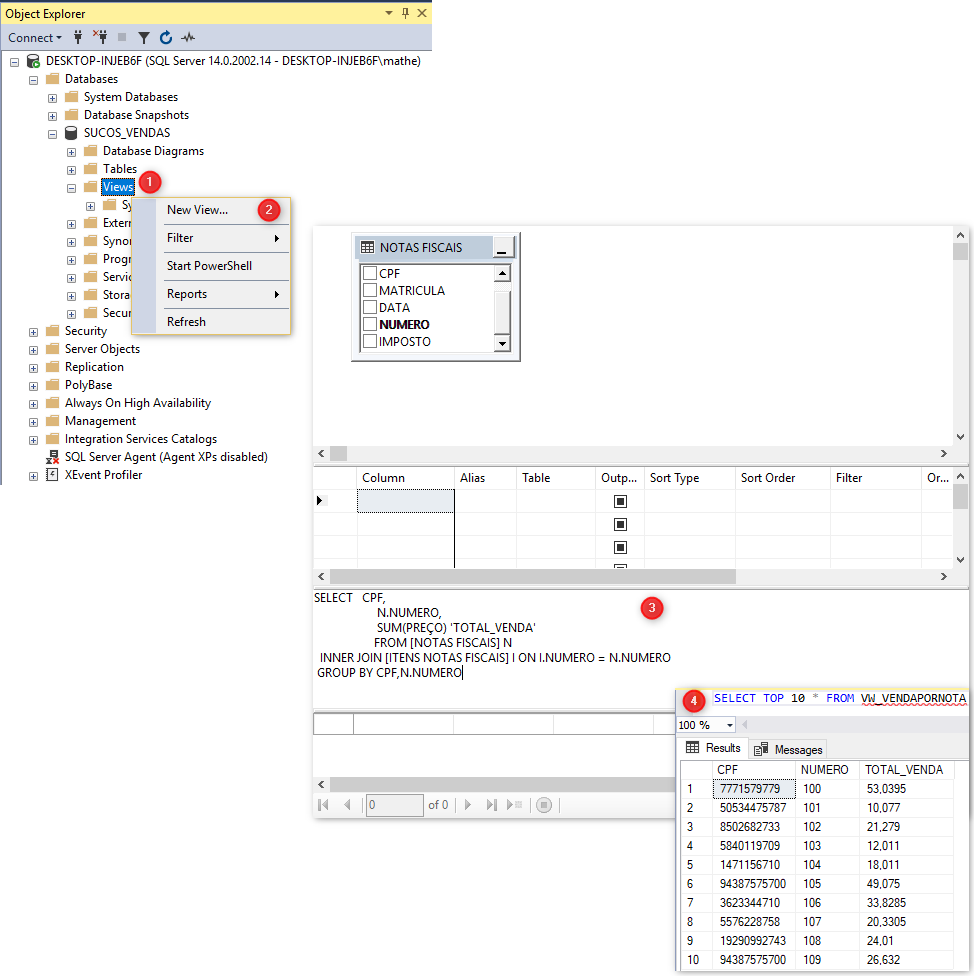
**4.2 - SUB-CONSULTAS**

Faz o papel de uma tabela temporária dentro de um **IN / FROM.**



**4.3 - VIEWS**

Consulta que é **criada e armazenada**.

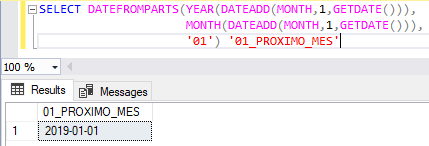


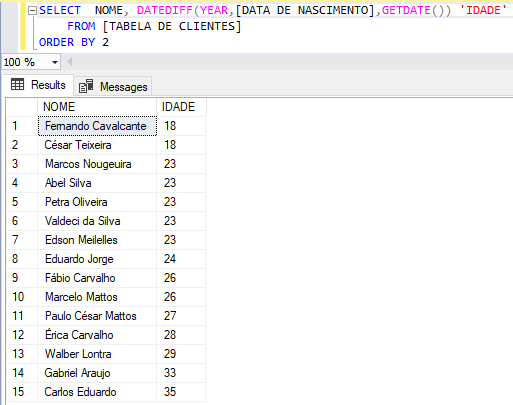
**5 - FUNÇÕES DO SQL SERVER**

**5.1 - Funções de STRING**

* **TRIM(expression):** Remove TODOS espaços em branco.
  + **LTRIM(expression):** Remove os espaços em branco À ESQUERDA
  + **RTRIM(expression):** Remove os espaços em branco À DIREITA
* **SUBSTRING(expression, start, length):** Retorna nova *string* a partir de uma expressão, o índice de início e o tamanho desejado
  + **LEFT (character, length):** Retorna nova *string*, a contar pela esquerda
  + **RIGHT (character, length):** Retorna nova *string*, a contar pela direita
* **LEN(expression):** Retorna o tamanho de uma expressão
* **UPPER(expression):** Retorna nova *string*, com letras em Maiúsculo
* **LOWER(expression):** Retorna nova *string,* com letras em Minúsculo
* **REPLACE(expession, old\_char, new\_char):** Retorna nova *string*, alterando os caracteres desejados
* **CONCAT(expression[],...):** Concatena uma série de *strings*  = ‘A’ + ‘B’(‘AB’) em um único campo.

**5.2 - Funções de DATA**

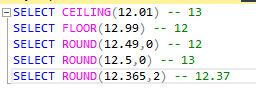




**OBS:** Arquivo em anexo na pasta com funções e explicações.

**5.3 - Funções MATEMÁTICAS**

* **CEILING(*expression*):** Arredonda para MAIOR inteiro
* **FLOOR(*expression*):** Arredonda para MENOR inteiro
* **ROUND(*expression, length*):** Arredonda de acordo com a expressão, levando em consideração o valor **mais próximo**. Recebe a **quantidade de casas decimais** desejadas.



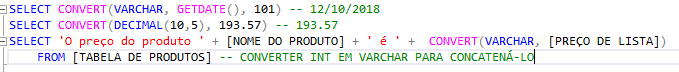
**5.4 - Conversão de Dados**

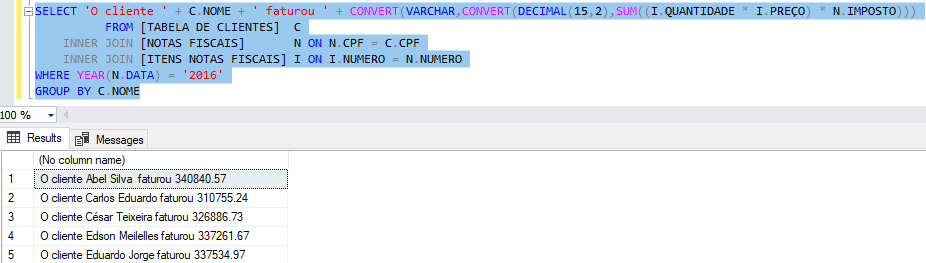
**CAST:** É padrão do SQL, existe no ORACLE/MYSQL.

**CONVERT:** Específico do SQL Server. Possui funcionalidades mais complexas.

<https://docs.microsoft.com/pt-br/sql/t-sql/functions/cast-and-convert-transact-sql?view=sql-server-2017>

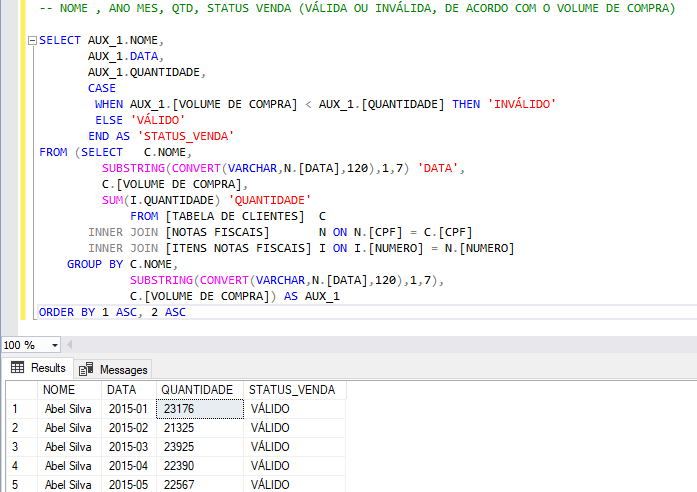
* CONVERT(<TIPO>,<CAMPO>,[<ESTILO (PARA DATA)>])



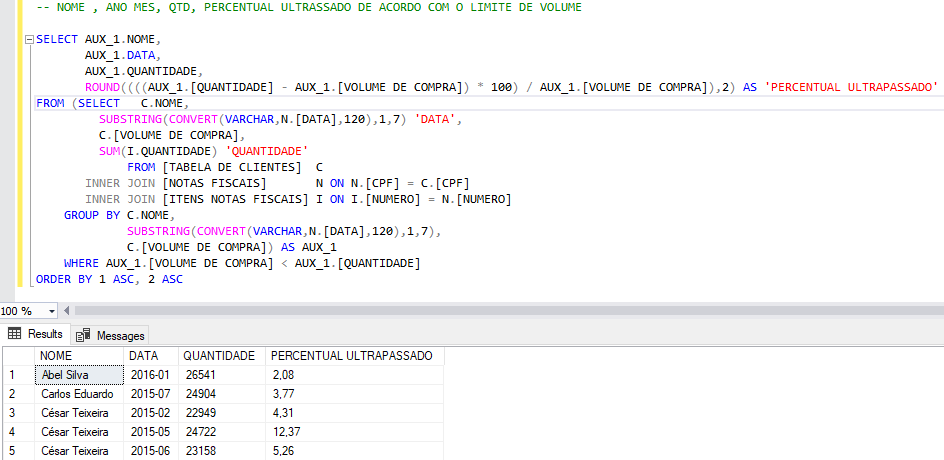


**6 – EXEMPLOS DE RELATÓRIOS**

**6.1 - NOME, ANO/MÊS DA VENDA, QUANTIDADE DE PRODUTOS NO MÊS, STATUS DA VENDA (VÁLIDO SE VOLUME DE COMPRA MAIOR OU IGUAL À QUANTIDADE)**



**6.2 - BUSCAR SOMENTE INVÁLIDOS E EXIBIR PERCENTUAL ULTRAPASSO NA QUANTIDADE**



**6.3 - SABOR, ANO, FATURAMENTO NO ANO E PARTICIPACAO DO DETERMINADO SABOR COMO PERCENTUAL**

