Sumário

[IMPORTANTE ! 2](#_Toc107719593)

[SELECT e FROM 2](#_Toc107719594)

[SELECT DISTINCT 2](#_Toc107719595)

[WHERE 2](#_Toc107719596)

[COUNT - Contagem 3](#_Toc107719597)

[TOP - Limita a contagem de retorno dos dados 3](#_Toc107719598)

[ORDER BY - Ordena os dados em Crescente ou Decrescente 3](#_Toc107719599)

[BETWEEN - Encontrar valor entre mínimo e máximo 4](#_Toc107719600)

[IN - Utilizado com Where para buscas/pesquisas 4](#_Toc107719601)

[LIKE – Realizar busca sem informações completas 4](#_Toc107719602)

[SUM – Realizar somas 5](#_Toc107719603)

[MIN – Menor Valor 5](#_Toc107719604)

[MAX – Maior Valor 5](#_Toc107719605)

[AVG – Média dos Valores 5](#_Toc107719606)

[GROUP BY – Divide o resultado da pesquisa em grupos 5](#_Toc107719607)

[HAVING – Utilizado com Group By para filtrar os dados que já estão agrupados 6](#_Toc107719608)

[AS – Nome/Apelido das colunas 6](#_Toc107719609)

[JOINS 6](#_Toc107719610)

[INNER JOIN 6](#_Toc107719611)

[LEFT JOIN // LEFT OUTER JOIN 7](#_Toc107719612)

[INNER JOIN 7](#_Toc107719613)

[Union – Unir duas consultas 7](#_Toc107719614)

[WHERE 8](#_Toc107719615)

[WHERE 8](#_Toc107719616)

[WHERE 8](#_Toc107719617)

[WHERE 8](#_Toc107719618)

[WHERE 8](#_Toc107719619)

[WHERE 8](#_Toc107719620)

[WHERE 8](#_Toc107719621)

[WHERE 8](#_Toc107719622)

[WHERE 8](#_Toc107719623)

[WHERE 8](#_Toc107719624)

# IMPORTANTE !

Este material tem os exemplos de acordo com a base de dados que está no link:

<https://drive.google.com/file/d/1LCofjYj-pV1asBLrxtgPDsbqMFRefHW5/view>

Link desse conteúdo:

[(92) Curso SQL Completo 2019 [Iniciantes] + Desafios + Muita Prática - YouTube](https://www.youtube.com/watch?v=rX2I7OjLqWE)

Este banco de dados também está na pasta:

SQL na Prática\Banco de Dados Curso

# SELECT e FROM

SELECT - seleciona uma ou mais tabelas

FROM - de onde irá vir a informação

# SELECT DISTINCT

Seleciona dados sem duplicidade de uma ou mais tabelas.

# WHERE

Condição para filtragem de dados.

--- Exemplos ---

SELECT \*

FROM Person.Person

WHERE LastName = 'miller' and FirstName = 'anna'

--

SELECT \*

FROM Production.Product

WHERE color = 'blue' or color = 'black'

--

SELECT \*

FROM Production.Product

WHERE ListPrice > 1500 and ListPrice < 5000

--

SELECT \*

FROM Production.Product

WHERE color != 'red' -- <> também serve para sinalizar 'diferente'

# COUNT - Contagem

--- Exemplos ---

SELECT count(\*) -- Contagem de Linhas

FROM Person.Person

SELECT count(title) -- contagem dos Títulos

FROM Person.Person

SELECT count(DISTINCT title) -- contagem dos sem duplicidades

FROM Person.Person

# TOP - Limita a contagem de retorno dos dados

--- Exemplos ---

SELECT TOP 10 \*

FROM Production.Product

# ORDER BY - Ordena os dados em Crescente ou Decrescente

--- Exemplos ---

SELECT \*

FROM Person.Person

ORDER BY FirstName asc -- OU desc

SELECT FirstName, LastName

FROM Person.Person

ORDER BY FirstName asc, LastName desc

# BETWEEN - Encontrar valor entre mínimo e máximo

--- Exemplos ---

SELECT \*

FROM HumanResources.Employee

WHERE HireDate between '2009/01/01' and '2010/01/01'

ORDER BY HireDate

SELECT \*

FROM Production.Product

WHERE ListPrice NOT between 100 and 1500 -- NOT para tudo 'exceto'

# IN - Utilizado com Where para buscas/pesquisas

--- Exemplos ---

SELECT \*

FROM Person.Person

WHERE BusinessEntityID IN (2,7,13)

# LIKE – Realizar busca sem informações completas

--- Exemplos ---

SELECT \*

FROM Person.Person

WHERE FirstName LIKE 'OVI%' -- Começa com OVI

--

SELECT \*

FROM Person.Person

WHERE FirstName LIKE '%to' -- Termina com TO

--

SELECT \*

FROM Person.Person

WHERE FirstName LIKE '%essa%' -- Que contém ESSA

Também pode ser usado o underline ('%ro\_'), que limite a busca do próximo caractere em apenas um.

SELECT \*

FROM Person.Person

WHERE FirstName LIKE '%ro\_'

# SUM – Realizar somas

--- Exemplos ---

SELECT TOP 10 SUM(LineTotal) AS "Soma"

FROM Sales.SalesOrderDetail

# MIN – Menor Valor

--- Exemplos ---

SELECT TOP 10 MIN(LINETOTAL) AS "Menor Valor"

FROM SALES.SalesOrderDetail

# MAX – Maior Valor

--- Exemplos ---

SELECT TOP 10 MAX(LINETOTAL) AS "Maior Valor"

FROM SALES.SalesOrderDetail

# AVG – Média dos Valores

--- Exemplos ---

SELECT TOP 10 AVG(LINETOTAL) AS "Maior Valor"

FROM SALES.SalesOrderDetail

# GROUP BY – Divide o resultado da pesquisa em grupos

--- Exemplos ---

-- Soma dos preços unitários por ID?

SELECT SpecialOfferID, SUM(UnitPrice) AS "SOMA"

FROM Sales.SalesOrderDetail

GROUP BY SpecialOfferID

-- Quantos produtos foram vendidos até hoje?

SELECT ProductId, COUNT (ProductId) AS "CONTAGEM"

FROM Sales.SalesOrderDetail

GROUP BY ProductID

-- Quantos nomes de cada nome temos no banco de dados?

SELECT FirstName, COUNT (FirstName) AS "Contagem"

FROM Person.Person

GROUP BY FirstName

-- Média de preços para os produtos que são pratas(silver)?

SELECT color, AVG (ListPrice) AS "PREÇO"

FROM Production.Product

WHERE Color = 'Silver'

GROUP BY Color

# HAVING – Utilizado com Group By para filtrar os dados que já estão agrupados

--- Exemplos ---

-- Quais produtos, no total de vendas, estão entre 162k e 500k

SELECT productId, SUM (linetotal) "TOTAL"

FROM Sales.SalesOrderDetail

GROUP BY ProductID

HAVING SUM (linetotal) BETWEEN 162000 AND 500000

-- Quais os nomes tem uma ocorrência maior que 10 vezes, porem somente somente onde o título é 'Mr."

SELECT FirstName, COUNT (FirstName) as "Quantidade"

FROM Person.Person

WHERE Title = 'Mr.'

GROUP BY FirstName

HAVING COUNT (FirstName) > 10

# AS – Nome/Apelido das colunas

--- Exemplos ---

SELECT TOP 10 AVG (ListPrice) AS "Preço Médio"

FROM Production.Product

# JOINS

Junção de dados entre tabelas diferentes.

## INNER JOIN

--- Exemplos ---

-- BusinessEntityId, FirstName, EmailAddress

SELECT TOP 10 \* FROM Person.Person

SELECT TOP 10 \* FROM Person.EmailAddress

/\*

Comparando as duas tabelas, a informação em comum é a coluna BusinessEntityId.

Utilizaremos ela para realizar a junção de informações que precisamos.

\*/

SELECT p.BusinessEntityId, p.FirstName, p.LastName, pe. EmailAddress

FROM Person.Person AS P

INNER JOIN Person.EmailAddress PE ON P.BusinessEntityID = PE.BusinessEntityID

--

-- Nomes dos Produtos e informações de suas subcategorias

--ListPrice, Nome do Produto, Nome da Subcategoria

SELECT TOP 10 \* FROM Production.Product

SELECT TOP 10 \* FROM Production.ProductSubcategory

/\*

Comparando as duas tabelas, a informação em comum é a coluna ProductSubcategoryID.

Utilizaremos ela para realizar a junção de informações que precisamos.

\*/

SELECT pr.ListPrice, pr.Name, pc.Name

FROM Production.Product AS Pr

INNER JOIN Production.ProductSubcategory AS PC ON PC.ProductSubcategoryID = Pr.ProductSubcategoryID

-- Juntas todas as informações de duas tabelas

SELECT TOP 10 \* FROM Person.BusinessEntityAddress

SELECT TOP 10 \* FROM Person.Address

/\* Coluna em comum é AddressID \*/

SELECT TOP 10 \*

FROM Person.BusinessEntityAddress BA

INNER JOIN Person.Address PA ON PA.AddressID = BA.AddressID

## LEFT JOIN // LEFT OUTER JOIN

--- Exemplos ---

-- Quais pessoas tem um cartão cadastrado?

SELECT \*

FROM Person.Person PP

LEFT JOIN Sales.PersonCreditCard PC

ON PP.BusinessEntityID = PC.BusinessEntityID

Condição para filtragem de dados.

## INNER JOIN

Condição para filtragem de dados.

# Union – Unir duas consultas

Union remove dados duplicados.

--- Exemplos ---

SELECT [ProductId], [Name], [ProductNumber]

FROM Production.Product

WHERE Name like '%Chain%'

UNION

SELECT [ProductId], [Name], [ProductNumber]

FROM Production.Product

WHERE Name like '%Decal%'

--

SELECT FirstName, Title, MiddleName

FROM Person.Person

WHERE Title = 'Mr.'

UNION

SELECT FirstName, Title, MiddleName

FROM Person.Person

WHERE MiddleName = 'A'

# WHERE

Consultar o link para ver padrões de manipulação de datas:

[DATEPART (Transact-SQL) - SQL Server | Microsoft Docs](https://docs.microsoft.com/pt-br/sql/t-sql/functions/datepart-transact-sql?view=sql-server-ver16)

-- Média de valor devido por mês?

SELECT AVG(TotalDue) as Medias, DATEPART(month, OrderDate) AS Mes

FROM Sales.SalesOrderHeader

GROUP BY DATEPART(month, OrderDate)

ORDER BY Mes

-- Média de valor devido por ano?

SELECT AVG(TotalDue) as Medias, DATEPART(year, OrderDate) AS Ano

FROM Sales.SalesOrderHeader

GROUP BY DATEPART(year, OrderDate)Condição para filtragem de dados.

# Manipulação de String

Link para consultar as funções:

[String Functions (Transact-SQL) - SQL Server | Microsoft Docs](https://docs.microsoft.com/en-us/sql/t-sql/functions/string-functions-transact-sql?view=sql-server-ver16)

-- Concatenar

SELECT CONCAT(FirstName, ' ', LastName)

FROM Person.Person

-- Maiúsculo, Minúsculo,

SELECT UPPER(FirstName), LOWER(LastName)

FROM Person.Person

-- Contagem de caracteres

SELECT FirstName, LEN(FirstName)

FROM Person.Person

-- Substring

SELECT FirstName, SUBSTRING(FirstName, 1, 3)

FROM Person.Person

-- Replace (Substituir)

SELECT ProductNumber, REPLACE(ProductNumber, '-', '#')

FROM Production.Product

# Operações Matemáticas

Link para consultar as funções:

[Funções matemáticas (Transact-SQL) - SQL Server | Microsoft Docs](https://docs.microsoft.com/pt-br/sql/t-sql/functions/mathematical-functions-transact-sql?view=sql-server-ver16)

-- Símbolos do cotidiano são os mesmos (+, -, \*, /)

-- Casas Decimais (Arredondamento)

SELECT LineTotal, ROUND(LineTotal, 2) AS 'Valor Arredondado'

FROM Sales.SalesOrderDetail

-- Raiz Quadrada

SELECT LineTotal, SQRT(LineTotal) AS 'Raiz Quadrada'

FROM Sales.SalesOrderDetail

Condição para filtragem de dados.

# WHERE

Condição para filtragem de dados.

# WHERE

Condição para filtragem de dados.

# WHERE

Condição para filtragem de dados.

# WHERE

Condição para filtragem de dados.

# WHERE

Condição para filtragem de dados.

# WHERE

Condição para filtragem de dados.

# WHERE

Condição para filtragem de dados.