Sumário

[Instalação e Configurações. 2](#_Toc118383101)

[Instalando um banco de dados de outras pessoas. 2](#_Toc118383102)

[Como comentar? 2](#_Toc118383103)

[SELECT – SELECIONADOR DE DADOS 2](#_Toc118383104)

[DISTINCT – DADOS UNICOS 3](#_Toc118383105)

[WHERE - CONDICIONAIS 4](#_Toc118383106)

[COUNT – Contador 5](#_Toc118383107)

[TOP – Retornar uma quantidade limitada de dados. 5](#_Toc118383108)

[ORDER BY – Ordenar dados por alguma coluna 5](#_Toc118383109)

[BETWEEN – Encontrar valores entre 6](#_Toc118383110)

[IN – Serve para selecionar um vetor de dados 6](#_Toc118383111)

[LIKE – Procurar combinações 7](#_Toc118383112)

[Min, Max, SUM, AVG 7](#_Toc118383113)

[AS - Como colocar apelido numa coluna. 7](#_Toc118383114)

[GROUP BY 8](#_Toc118383115)

[Having 9](#_Toc118383116)

[As – Serve apenas para renomear as colunas 9](#_Toc118383117)

[JOIN – Juntar dados 9](#_Toc118383118)

[UNION 12](#_Toc118383119)

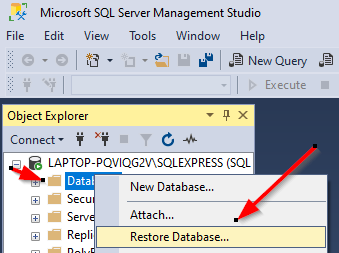
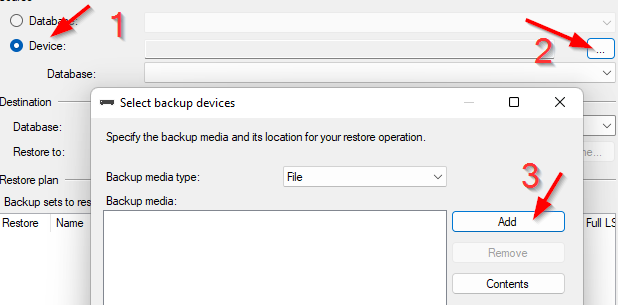
[SELF-JOIN 13](#_Toc118383120)

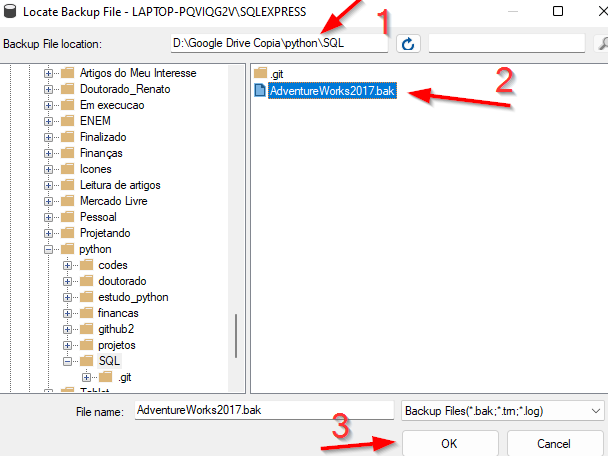
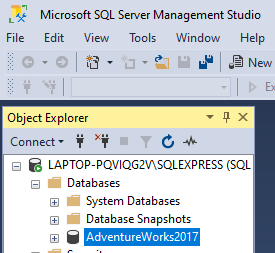
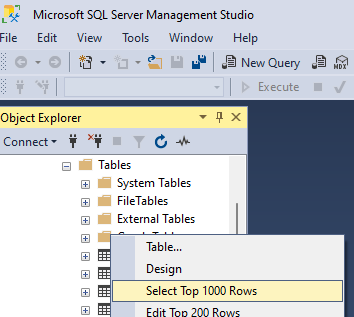
# Instalação e Configurações.

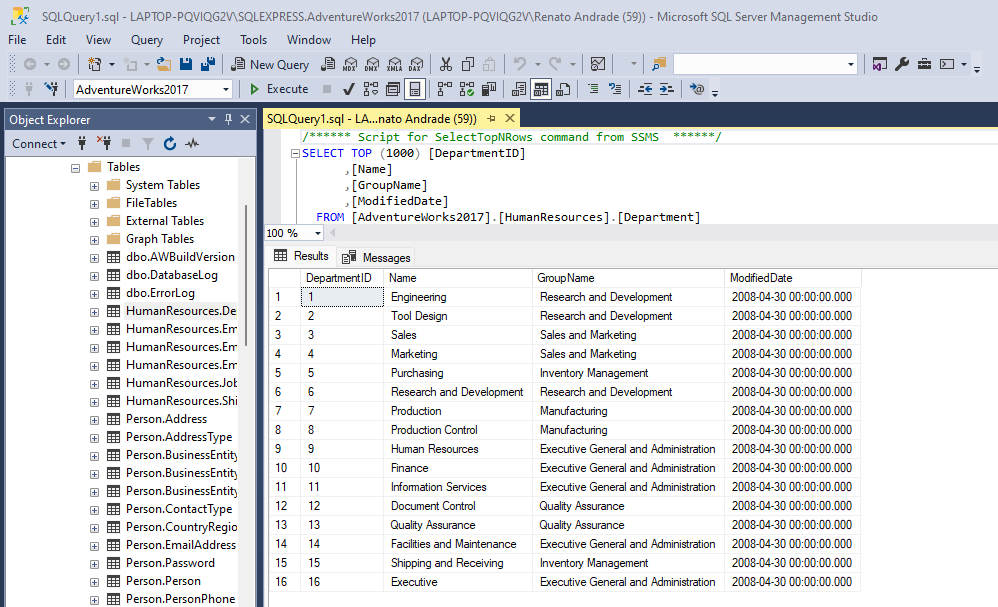
Caso não esteja conectando pode colocar esse código:

Localhost\SQLEXPRESS

# Instalando um banco de dados de outras pessoas.



# Como comentar?

-- ou /\*

# SELECT – SELECIONADOR DE DADOS

SELECT coluna1, coluna2

FROM tabela

tabela = nome da tabela

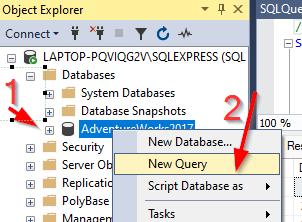
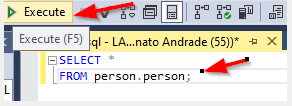
coluna é

SELECT \*

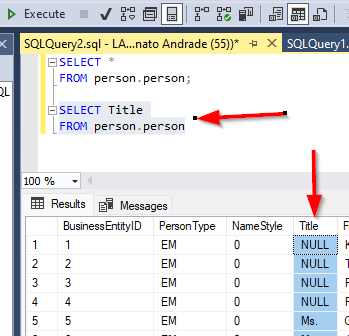
FROM tabela

Retorna todas colunas do tabela

Como proceder?

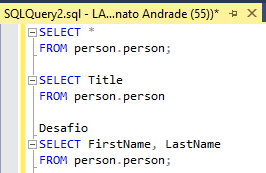
 

Caso queira apenas algumas colunas:



Se tiver mais de um código selecione o qual deseja rodar e pressione F5.

Desafio - A equipe de marketing precisa de fazer uma pesquisa sobre os nomes mais comuns de seus clientes e precisa do nome e sobre de todos os clientes que estão cadastrados no sistema:



# DISTINCT – DADOS UNICOS

Esse comando serve para pegar dados da tabela únicos, o que significa omitir os duplicados.

SELECT DISTINCT coluna1, coluna2

FROM tabela

Desafio - Quantos sobrenomes únicos temos na tabela person.person?

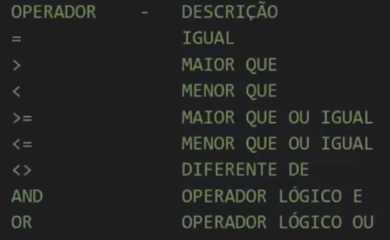


# WHERE - CONDICIONAIS

SELECT coluna1, coluna\_n

From tebla

WHERE condição



Retorna onde o laste name for miller:

SELECT \*

FROM person.Person

WHERE LastName = 'miller'

USO do “AND” Retorna todas as annas que tem sobrenome miller:

FROM Person.Person

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

Uso do OR:

Retorna todos produtos vermelhos ou pretos

SELECT \*

FROM Production.Product

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

Uso do > <:

Seleciona produtos com preços entre 1500 à 5000

SELECT \*

FROM Production.Product

WHERE ListPrice > 1500 and ListPrice < 5000

Uso do <>:

Seleciona produtos DIFERENTE da condição

SELECT \*

FROM Production.Product

WHERE color <> 'red'

Desafio - A equipe de produção precisa do nome de todas as peças que pesam mais de que 500 kg e menos que 700 kg para inspeção:

SELECT Name

FROM Production.Product

WHERE 500 < Weight and Weight <= 700

Desafio – O marketing quer saber uma relação entre os empregados (employees) que são casados (single = solteiro, married = casado) e são assalariados (salaried)

SELECT \*

FROM HumanResources.Employee

Where MaritalStatus = 'M' and SalariedFlag = '1'

Desafio – O usuario Peter Krebs está devendo um pagamento, consiga o e-mail dele para enviar uma cobraça!

Passo 1

SELECT \*

FROM person.Person

WHERE FirstName = 'Peter' and LastName = 'Krebs'

Passo 2

SELECT EmailAddress

FROM person.EmailAddress

WHERE BusinessEntityID = 26

# COUNT – Contador

Contar o numero de pronomes de tratamento (Ms, Mr... – tabela = Title que está person.person):

SELECT count(DISTINCT Title)

FROM person.Person

Retorno 6

Desafio – Quantos produtos temos cadastrado na tabela production.product

SELECT count(\*)

FROM production.Product

Desafio – Quantos tamanhos de produtos temos cadastrado na tabela production.product

SELECT count(size)

FROM production.Product

Desafio – Quantos tamanho diferentes de produtos temos cadastrado na tabela production.product

SELECT count(DISTINCT size)

FROM production.Product

# TOP – Retornar uma quantidade limitada de dados.

SELECT TOP N \*

FROM tabela

Retonar as 10 primeiras linhas da tabela production:

SELECT TOP 10 \*

FROM Production.Product

# ORDER BY – Ordenar dados por alguma coluna

SELECT coluna1, coluna2

FROM tabela

ORDER BY coluna1 asc ou desc

Ordenando todas as colunas da tabela person.person pelo primeiro nome de forma ascendente:

SELECT \*

FROM person.Person

ORDER BY FirstName asc

Ordenar as colunas da tabela person.person pelo primeiro nome de forma ascendente e o ultimo nome de forma descendente:

SELECT FirstName, LastName

FROM person.Person

ORDER BY FirstName asc, LastName desc

Significa dizer que primeiro vai ordenar pelo primeiro nome e depois vai ordenar pelo segundo nome.

Desafio – Encontra o ProductID ordenado de forma decrescente dos 10 produtos mais caros da tabela production.product

SELECT TOP 10 ProductID

FROM Production.Product

ORDER BY ListPrice desc

Desafio – Obter o nome e numero dos produtos das 4 primeiras linhas ordenados em ordem crescente pelo ProductID

SELECT TOP 4 name, ProductNumber

FROM Production.Product

ORDER BY ProductID asc

# BETWEEN – Encontrar valores entre

Parecido com valor >= minimo and valor <= máximo;

Encontrar produto com valor entre 1000 e 1500:

SELECT \*

FROM Production.Product

WHERE ListPrice between 1000 and 1500;

Não encontrar produtos com valor entre 1000 e 1500:

SELECT \*

FROM Production.Product

WHERE ListPrice not between 1000 and 1500;

Pessoas contratadas entre uma data e outra ordenada pela data:

SELECT \*

FROM HumanResources.Employee

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

Order By HireDate

# IN – Serve para selecionar um vetor de dados

Selecionando as pessoas com BusinessEntityID = 2, 7 e 13

SELECT \*

FROM Person.Person

Where BusinessEntityID IN (2, 7, 13)

Substitui:

SELECT \*

FROM Person.Person

Where BusinessEntityID = 2

OR BusinessEntityID = 7

OR BusinessEntityID = 13

# LIKE – Procurar combinações

Nomes que iniciam com ovi:

SELECT \*

FROM person.person

WHERE FirstName LIKE 'ovi%'

DESAFIO – Quantos produto temos cadastrados que custam mais que 1500 dolares?

SELECT count(ListPrice)

FROM Production.Product

WHERE ListPrice > 1500

DESAFIO - Quantas pessoas inciam com a letra P?

SELECT count(FirstName)

FROM Person.Person

WHERE FirstName LIKE '%P'

DESAFIO – Em quantas cidades nossos clientes moram?

SELECT count(DISTINCT City)

FROM Person.Address

Desafio – Quais são as cidades que temos clientes castrados?

SELECT DISTINCT(City)

FROM Person.Address

Desafio – Quantos produtos vermelhos temos entre 500 a 1000 dolares?

SELECT count(\*) AS 'Produtos vermelho que custam de 500 à 1000 dolares'

FROM Production.Product

Where color = 'red' AND ListPrice Between 500 and 1000

Desafio – Quantos produtos cadastrados tem a palavra “road” no nome deles?

SELECT count(\*)

FROM Production.Product

Where Name Like '%road%';

# Min, Max, SUM, AVG

SUM - Somar os dados de conjunto de dados:

SELECT TOP 10 sum(linetotal)

FROM Sales.SalesOrderDetail

MIN – Menor valor de uma coluna:

SELECT TOP 10 MIN(LINEtotal)

FROM Sales.SalesOrderDetail

AVG – Média de uma coluna:

SELECT TOP 10 AVG(LINEtotal)

FROM Sales.SalesOrderDetail

# AS - Como colocar apelido numa coluna.

SELECT TOP 10 sum(linetotal) AS "Soma"

FROM Sales.SalesOrderDetail

# GROUP BY

Dividir os resultados da sua pesquisa em grupos.

Exemplo somar as vendas de um dia específico:

SELECT ModifiedDate, SUM(UnitPrice) AS "SOMA"

FROM Sales.SalesOrderDetail

GROUP BY ModifiedDate

No caso o grupo é o dia da venda.

Exemplo: quantas vezes e quantos diferentes nomes, que tem ‘UST’, aparece na lista person.person:

SELECT FirstName, count(FirstName) AS "Vezes que aparece"

FROM Person.Person

WHERE FirstName LIKE '%ust%'

GROUP BY FirstName

Perceba que no SELECT tem 2 colunas que eu to chamando, os nomes e as contagens.

Média de preço dos produtos onde a cor é prata:

SELECT color, AVG(ListPrice) AS 'Preço'

FROM Production.Product

WHERE color like 'silver'

GROUP BY Color

Desafio – Quantas pessoas tem o mesmo MiddleName?

SELECT MiddleName, count(MiddleName) AS 'Soma'

FROM person.Person

GROUP BY MiddleName

Desafio - Quanto em media um produto é vendido na loja?

SELECT ProductID, AVG(OrderQty) AS 'QTDE'

FROM Sales.SalesOrderDetail

GROUP BY ProductID

Desafio confuse – Qual foram as 10 vendas que no total tiveram os maiores valores de venda (linetotal) por produto do maior valor para menor.

SELECT TOP 10 ProductID, MAX(LineTotal) 'Max'

FROM Sales.SalesOrderDetail

GROUP BY ProductID

ORDER BY 'Max' desc

Desafio – Saber quantos produtos e qual quantidade media de produtos que temos cadastrados nas nossas ordens de serviço (workorder), agrupado pelo productID.

elect ProductId, Count(productid) "contagem", AVG(orderqty) 'media'

FROM Production.WorkOrder

GROUP BY ProductID

# Having

Filtra dados que já estão agrupados.

SELECT coluna1, funcaoAgragracao(coluna2)

FROM tabela

GROUP BY coluna1

HAVING condição

Como saber a ocorrência de nomes que aparecem mais de 10x:

SELECT FirstName, count(FirstName) as "quantidade"

FROM person.Person

GROUP BY FirstName

HAVING count(FirstName) > 10

Como saber quais produtos tiveram vendas no total entre 162 k e 500 k:

SELECT productid, sum(LineTotal) as "total"

FROM Sales.SalesOrderDetail

GROUP BY ProductID

HAVING sum(lineTotal) between 162000 and 500000

Diferença entre where e having:

Como contar um nome que aparece Mr. Mais de 10 vezes?

Select FirstName , count(FirstName) as "qtde"

FROM person.Person

WHERE Title = 'Mr.'

GROUP BY FirstName

having count(firstname) > 10

Desafio – Identificar quais provincias aparecem mais que 1000x:

select StateProvinceID, count(StateProvinceID) as "Contagem"

FROM Person.Address

GROUP BY StateProvinceID

HAVING count(StateProvinceID) > 1000

ORDER BY Contagem desc

Desafio – Quais produtos não estão trazendo um número de vendas de no mínimo 1 milhão

# As – Serve apenas para renomear as colunas

# JOIN – Juntar dados

Juntar dados de tabelas diferentes

Sintaxe:

SELECT C.ClienteId, C.Nome, E.Rua, E.Cidade

FROM Cliente C

INNER JOIN Endereco e ON E.EnderecoId = C.EnderecoId

Pegar o email que esta numa tabela, baseado em dados de uma outra tabela:

-- Quero: BusinessEntityId, FisrtName, LastName, EmailAddress

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

From Person.Person as P

INNER JOIN Person.EmailAddress as PE on p.BusinessEntityID = pe.BusinessEntityID

-- Quero: ListPrice, Nome do produto, nome da subcategoria

Primeira coisa é visualizar as tabelas de dados

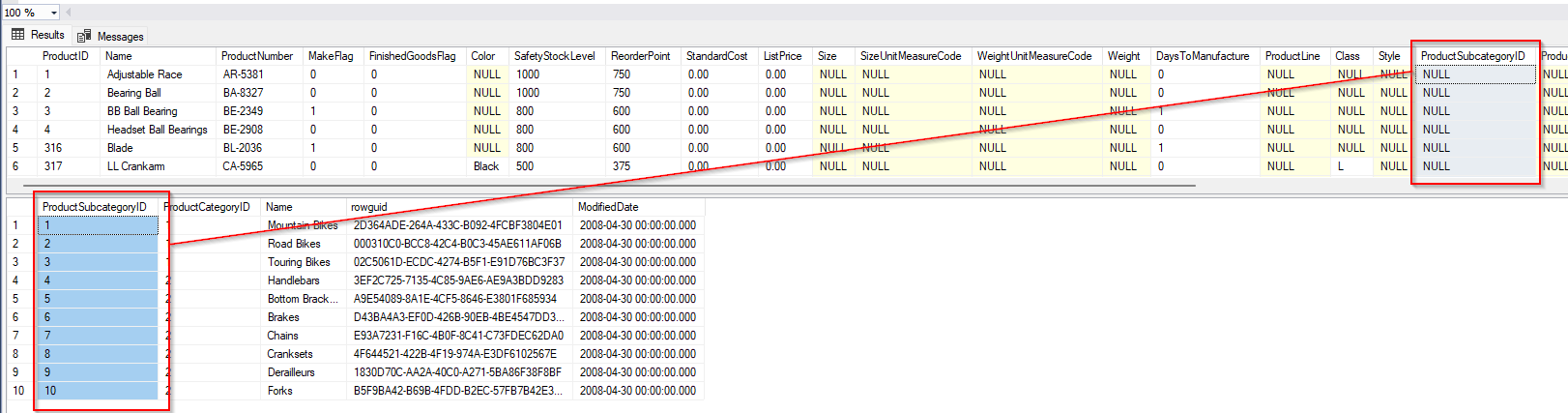
SELECT TOP 10 \*

FROM Production.Product

SELECT TOP 10 \*

FROM Production.ProductSubcategory

Segundo encontrar o dado em comum das tabelas:



3º Executar os codigos:

SELECT pr.listPrice, pr.Name, pc.Name

From Production.Product as pr

INNER JOIN Production.ProductSubcategory as PC on PC.ProductSubcategoryID = pr.ProductSubcategoryID

Desafio – Quero: BusinessEntityId, Name, PhoneNumberTypeId, PhoneNumber

De onde?

Select top 10 \*

From person.PhoneNumberType

Select top 10 \*

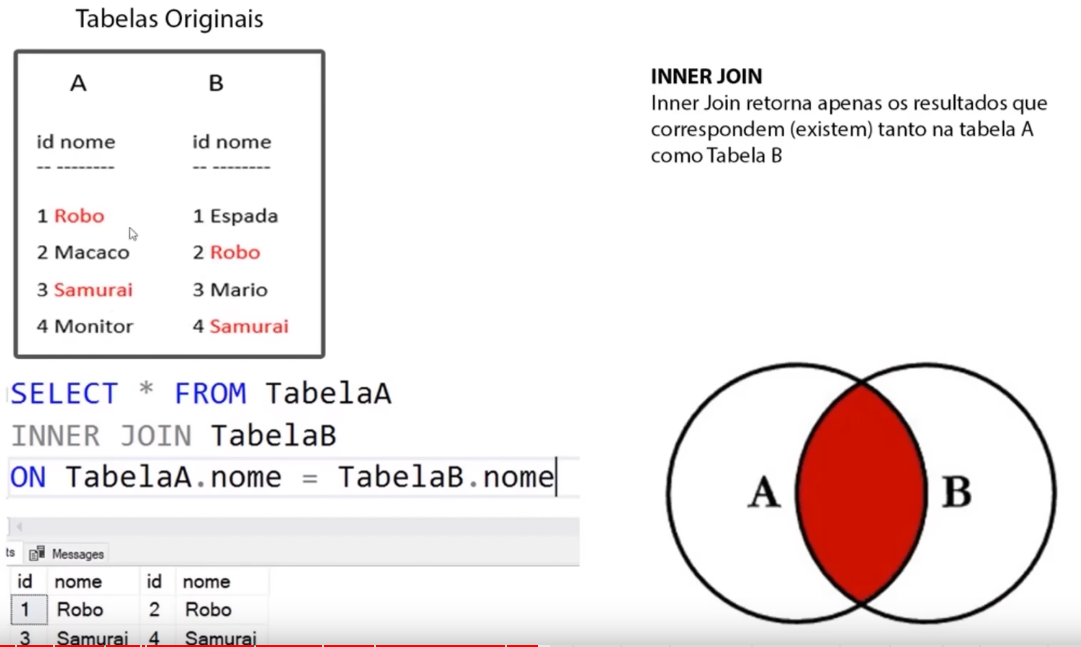
From person.PersonPhone

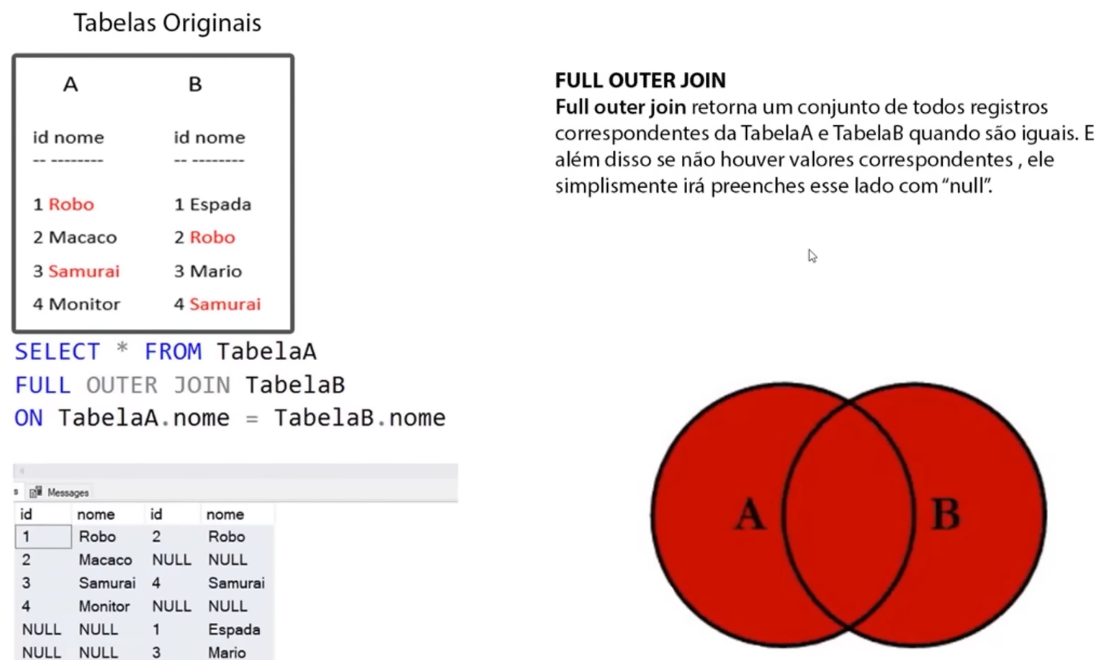
Resposta:

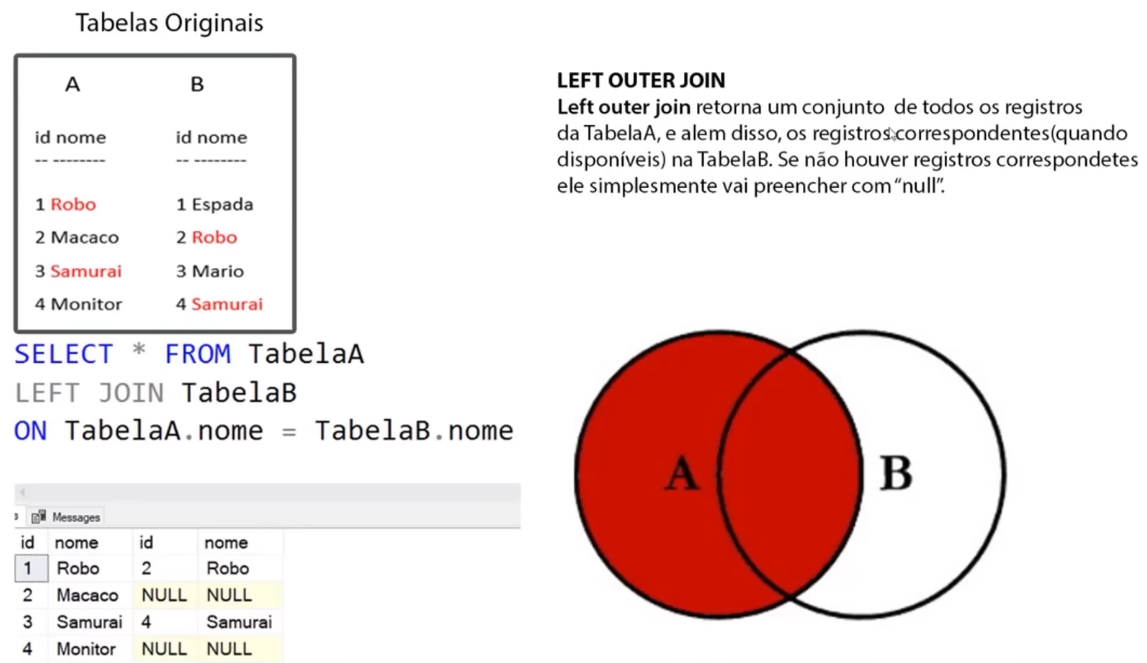
Select pp.BusinessEntityID, pe.Name, pp.PhoneNumberTypeID, pp.PhoneNumber

From person.PersonPhone as pp

INNER JOIN person.PhoneNumberType as pe on pe.PhoneNumberTypeID = pp.PhoneNumberTypeID







-- Descobrir quais pessoas possuem um cartao de credito registrado

-- Person.Person e Sales.PErsonCreditCard

Select \*

From Person.Person as Pessoas

LEFT JOIN Sales.PersonCreditCard as Cartoes

ON Pessoas.BusinessEntityID = Cartoes.BusinessEntityID

-- Descobrir quais pessoas possuem NÃO tem um cartao de credito registrado

-- Person.Person e Sales.PErsonCreditCard

Select \*

From Person.Person as Pessoas

LEFT JOIN Sales.PersonCreditCard as Cartoes

ON Pessoas.BusinessEntityID = Cartoes.BusinessEntityID

**WHERE Cartoes.BusinessEntityID IS NULL**

# UNION

SINTAX:

SELECT col1, col2

From Table1

UNION

SELECT col1, col2

From Table2

Exemplo 1:

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

From Production.Product

Where Name like '%Chain%'

UNION

Select ProductId, Name, ProductNumber

From Production.Product

where Name like '%Decal%'

ORDER BY Name desc

Na mesma table procura por nomes de produtos que tenha chain e Decal e junta tudo.

# SELF-JOIN

Exemplo 1:

--Sintax:

-- SELECT col1

-- FROM table1, table2

-- Where Condicao

--Clientes que moram na mesma regiao

Select A.ContactName, A.Region, B.ContactName, B.Region

From Customers as A, Customers B

Where A.Region = B.Region

Exemplo 2:

--Nome e data de contratacao de todos os funcionarios de um mesmo ano

-- Primeiro uma visao geral da tabela

Select \*

FROM Employees

-- Seperar os dados

Select A.FirstName, A.HireDate, B.FirstName, B.HireDate

From Employees A, Employees B

where DATEPART(Year, a.hiredate) = DatePart(Year, b.hiredate)

desafio

-- Quais produtos tem o mesmo percentual de desconto? [Order Details]

Select A.ProductId, A.Discount, B.ProductId, B.Discount

From [Order Details] A, [Order Details] B

where A.Discount = B.Discount

ORDER BY A.ProductID Desc

# SUBQUERY – É o select do select

Exemplo:

-- Monte um relatorio de todos produtos que tem preço acima da média

-- Visualize a table primero

Select \*

From Production.Product

-- Faça o codigo

Select \*

From Production.Product

where ListPrice > (SELECT AVG(listprice) from Production.Product)

Exemplo:

-- Nomes dos funcionarios que tem cargo de design Engineer

-- Visualize os dados

Select A.BusinessEntityID

From HumanResources.Employee A

Where JobTitle like 'design Engineer'

-- Crie o codigo

SELECT FirstName

FROM Person.Person

Where BusinessEntityID IN (Select A.BusinessEntityID

From HumanResources.Employee A

Where JobTitle like 'design Engineer')

-- Nota: É a partir do cargo que procura-se o nome. É so fazer um algorítimo mental

-- Outra forma de fazer o mesmo exemplo:

Select P.FirstName

FROM Person.Person as P

Inner Join HumanResources.Employee as E ON P.BusinessEntityID = E.BusinessEntityID AND E.JobTitle = 'Design Engineer'

Desafio

-- Encontre todos os endereço que estão no estado de "Alberta", Todas as informações

-- Usar Person.Address e Person.StateProvince

Select \*

From Person.Address

where StateProvinceID IN(

Select StateProvinceID

From Person.StateProvince

where [Name] = 'Alberta'

)